



MANUALE SPRIX



SPRIX / COLLAGE

AVVERTENZE IMPORTANTI SULLA DOCUMENTAZIONE

Questo documento è il **Manuale** didattico ed operativo relativo al linguaggio di programmazione dei componenti **Sprix** e **Collage** contenuti nella piattaforma di sviluppo **PassBuilder** integrata nel **Gestionale Passepartout**.

La piattaforma è collocata nella voce '**App PassBuilder**' del menu '**Servizi**' ed è costituita da un aggregato di funzionalità e strumenti nativi per realizzare personalizzazioni e sofisticate procedure verticali.

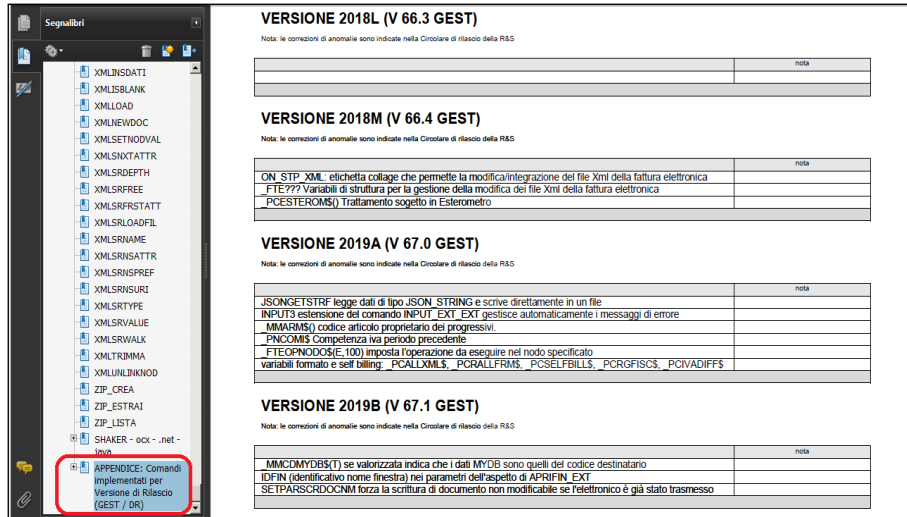
Fra questi strumenti di sviluppo ci sono quelli chiamati Sprix e Collage. E' sufficiente realizzare anche un singolo programma con uno di questi due strumenti a rappresentare un'**Applicazione Passepartout** realizzata con PassBuilder, pertanto, il presente manuale integra il "**Manuale PassBuilder**".

Il manuale Sprix/Collage è una componente del manuale PassBuilder in quanto contiene informazioni propedeutiche per sviluppare Applicazioni Passepartout. Per ragioni tecniche il manuale Sprix/Collage è mantenuto separato dal manuale PassBuilder, in quanto possiede funzionalità specifiche come i collegamenti all'Editor di programmazione Sprix/Collage (cioè può richiamare i dettagli di singole istruzioni, funzioni e variabili) e link interni agli esempi ed a sezioni di raggruppamenti per argomento.

Entrambi i manuali, infatti, sono disponibili on-line sia nell'Area Riservata del sito Edupass.it sia nell'Area Sviluppatori presente nell'Area Riservata del sito Passepartout.net (accessibile con credenziali user/password oppure dallo stesso Gestionale). Sebbene siano accessibili da punti diversi, questi documenti sono identici.

Si consiglia di scaricare il Manuale Sprix in formato PDF, dato che questo consente:

- la conservazione di questo documento per ciascuna versione del Gestionale, considerando che è anche suscettibile di modifiche in "tempo reale" per eventuali necessità contingenti; la complessità del contesto ambientale in cui si diffonde la natura tecnica di questo documento può richiedere utili e/o necessari interventi tempestivi ad ampia diffusione, al fine di aggiungere correzioni, chiarimenti, esempi, aggiornamenti d'appendice, ecc; le nuove informazioni, quindi, sono immediatamente fruibili e non passano necessariamente attraverso i periodici leggimi e le relative circolari, poichè questi ultimi due tipi di documenti riportano gli aggiornamenti di particolare rilevanza e correzioni di anomalie, pertanto si consiglia di riferirsi sempre ai leggimi elencati nell'Appendice in calce al manuale (v. figura). L'aggiornamento del manuale Sprix è indipendente dalle release del Gestionale, pertanto la data di pubblicazione è riportata nel titolo di questo documento, spesso equivalente alla data di ultima modifica del file PDF; la possibilità di scaricare in PDF consente quindi all'utente di collezionare e conservare tutte le versioni d'interesse e le cui differenze può confrontare con appositi programmi di comparazione; si consiglia di attendere qualche giorno dal rilascio dell'ultima release del Gestionale, in modo da conoscere anche le eventuali "ultime notizie";



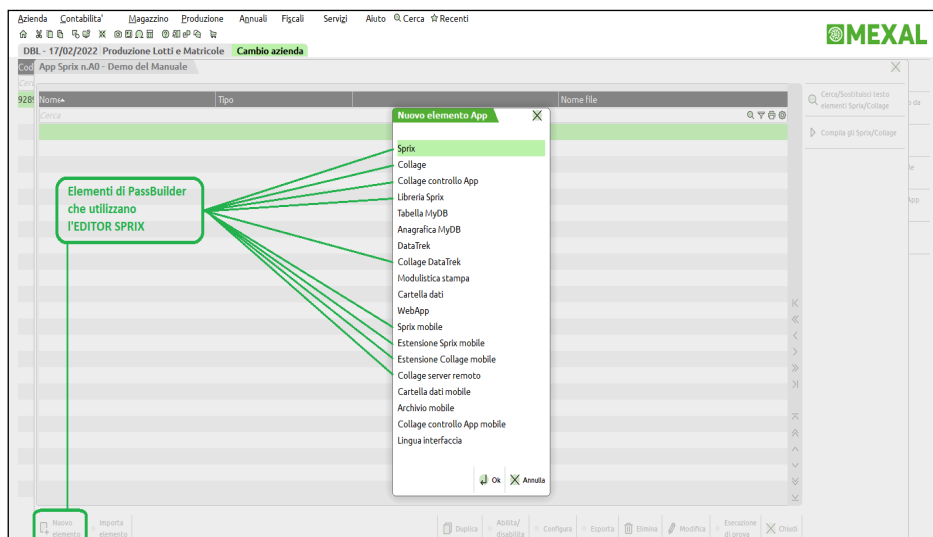
- la consultazione off-line e con un comodo "segnalibro";
- le ricerche più efficienti in base a parole "chiave"; il linguaggio Sprix è fatto di parole "chiavi" e a meno che non sia ancora previsto l'argomento è impossibile non trovare l'informazione desiderata;
- la visualizzazione di commenti e note a piè di pagina, spesso utili per informazioni ridondanti oppure storiche; una visualizzazione dei testi più utile per un manuale tecnico e più vicina alle esigenze dei Partner Sviluppatori;
- l'utilizzo di link "clicca qui" che collegano le singole voci ai paragrafi dei comandi raggruppati per argomento d'appartenenza;

Per tutti gli argomenti generali relativi alla piattaforma PassBuilder e che non sono menzionati (ripetuti) in questo manuale Sprix/Collage, si rimanda al manuale PassBuilder, di cui riportiamo due link:

[PassBuilder in Area Pubblica](#) (vedere il capitolo PassBuilder del menu Servizi nel manuale Passepartout).

[PassBuilder in Area Riservata](#) (il documento è il capitolo estratto dal manuale Passepartout).

Sebbene una singola Applicazione Passepartout possa essere realizzata in PassBuilder anche con uno solo fra i due strumenti di sviluppo Sprix e Collage, si ricorda che la piattaforma è composta anche da altri elementi e funzionalità (v. figura), pertanto anche i loro manuali (es. 'MyDB'), potrebbero essere mantenuti separati e collocati in altri contesti più idonei, ma ugualmente raggiungibili (ad esempio la 'Modulistica di Stampa' è utilizzata anche dalla procedura standard, pertanto il suo manuale è collocato nel manuale generale).



SPRIX

Inoltre, alcuni di questi elementi sfruttano anche specifiche funzionalità descritte in questo manuale Sprix/Collage (v. figura), ma che sono da integrare con le informazioni e gli esempi contenuti nei rispettivi manuali di ciascuno strumento (es. Mobile, Datatrek, ecc) oppure il medesimo PassBuilder o il Web Market PassStore; per la medesima motivazione, in Sprix/Collage si rimanda l'approfondimento di alcune funzionalità agli specifici manuali che le trattano (Xml, Web Service, Sql, ecc.).

Notare Bene:

I programmi Sprix e Collage sono capaci di leggere e scrivere dati, possono interagire a video e stampare in vari modi, si relazionano ad archivi e device esterni, ecc... Con Sprix e Collage è facile programmare, ma presuppone la conoscenza del contesto ambientale per ottenere i migliori risultati, pertanto le informazioni contenute nel [Manuale Utente \(clicca qui\)](#) sono **indispensabili** per potere conoscere l'ambiente in cui questi strumenti operano: caratteristiche del Sistema; configurazioni dei componenti Client e Server; remotizzazioni LAN (Locale) e WAN (Live Server Farm); accessi; restrizioni; funzionalità; regole; automatismi; controlli; semaforiche dati; ecc.. **Il Manuale Utente contiene frequenti riferimenti a SPRIX.** Solo a titolo di esempio: non è compito del Manuale Sprix descrivere quali siano i limiti di Sprix nell'Azienda di Prova o nell'Azienda Storizzata, ma è compito dei paragrafi del Manuale Utente dedicati alle funzioni "Azienda Prova" e "Storizzata" del menu "Apertura Azienda".

Ignorare il **Manuale Utente** oppure l'eventuale **Documentazione Specifica** della funzionalità in esame, quindi, comporta dei rischi di inefficacia e inefficienza dell'Applicazione, esempio: la **replica** di funzioni già esistenti oppure l'incontro di **vincoli o limiti** strutturali non considerati in sede di analisi. Il Manuale Sprix implementa lo stato dell'arte sui comandi SPRIX, ma **non riscrive** le informazioni contenute in altri manuali (anche quelli specializzati come la **Fatturazione Elettronica**) e **non descrive** tutto quello che si potrebbe fare oppure non fare con i comandi a disposizione, anche relativamente alla replica di funzioni del Gestionale che potrebbero avere vincoli commerciali. Il Manuale Sprix ripete e approfondisce talune argomentazioni quando le casistiche diventano importanti, frequenti, utili (workaround oppure "desiderata"), ma **ciò non significa che quanto non è espressamente dichiarato sia comunque consentito**, poichè occorrono sempre le necessarie verifiche in base a **tutta la documentazione coinvolta**, pena la deduzione di conclusioni affrettate oppure incomplete in sede d'analisi e conseguente rischio di intraprendere lo sviluppo di soluzioni in condizioni difficili o impossibili.

La voce [Allegati e Documenti](#) dell'Area Sviluppatori presente nell'Area Riservata del sito Passepartout.net, contiene:

1. guida alla documentazione, le condizioni di utilizzo del servizio di supporto EasyPass, l'utilizzo del Forum;
2. template per la realizzazione della documentazione di una Applicazione (come realizzare il manuale di una App);
3. esempio Shaker XLS in VB;
4. archivio storico di esempi e documenti, ancora validi ma non più aggiornati (tracciati, tabelle, relazioni, diagrammi, mirroring);
5. esempi di App didattiche, principalmente in Sprix e Collage;

Considerare, infine, anche la [Composizione Prodotti](#) della sezione Commerciale nell'Area Riservata, in quanto riporta dei vincoli commerciali, ad esempio: i livelli di prodotto che non sono contabili hanno, ovviamente, dei limiti di gestione della primanota.

Se sei un nuovo collaboratore puoi ignorare la seguente tabella commerciale, che conclude queste avvertenze relative alla documentazione:

POLITICHE STRUMENTI DI SVILUPPO

Il presente paragrafo è soggetto ad aggiornamenti e sostituisce le seguenti obsolete politiche relative agli strumenti di sviluppo:

- Politica *Sprix Lite, Sprix Full, MDS, SDK* – versione di riferimento **2003G**, ex chiarimento **18916** supporto online (**obsoleto**)
- Politica *Sprix Pubblico, Sprix Privato* – versione di riferimento **2011C**, ex chiarimento **26612** supporto online (**obsoleto**)

MDS / PASSBUILDER (versione di riferimento **2016H** - in vigore)

MDS (liv. 4)	Pass Builder (liv. 5)	Funz.tà dalla 2016H	Sprix Collage	Modalità	Stampe Pers.zzate	MyDB	MexalDB	Shaker	Esecuz. App	Sviluppo App	Sprix Mobile	Data Trek
N	N	N	S	Pubblica	S	S	N	N	N	N	N	N
S	N	N	S	Privata	S	S	S	S	N	N	N	N
N	S	S	S	Pubblica	S	S	N	N	S	S	S	S
S	S	S	S	Privata	S	S	S	S	S	S	S	S

Il livello di licenza di sviluppo disponibile si verifica premendo i tasti [F1+F1+F2] nel Gestionale; valutare ciascun caso (configurazione) in base alle vigenti condizioni commerciali, comunicate attraverso appositi canali di comunicazione e circolari di Area Riservata (es: circolari del 1 luglio e del 1 agosto 2016).

MDS (Mexal Developer Suite)

La personalizzazione consente di non dovere collocare il sorgente nell'installazione, cioè la personalizzazione può operare nella cosiddetta modalità "*privata*" anziché in quella predefinita cosiddetta "*pubblica*" (*). Nelle installazioni Live nella **Server Farm Passepartout**, lo Shaker è disponibile con il solo componente ".Net"; in ambiente Live l'utilizzo dei restanti componenti **Shaker, MexalDB, RDBMS** oppure software diversi dal Gestionale che richiedono configurazioni particolari (es: operazioni SQL su file Xls/Mdb via ODBC) **non sono disponibili**. Shaker non occupa un terminale "virtuale" (MDI), piuttosto occupa un terminale reale, come previsto dalla licenza. Dalla versione 2020A (v750), MDS consente la replica DB Datatrek in chiaro.

Terminali Utenti e Terminali di Servizio (di tipo "L" limitato fino a 3 utenti, di tipo "I" indefinito fino a quanti possibili)
Per ulteriori informazioni consultare le **[Nozioni Generali - Configurazione]** nel Manuale Utente

SHELLSPRIX	E' terminale di Servizio di tipo "I" ed eseguita dall'utente attivo (internamente al Gestionale)
SPXREMOTO\$	E' terminale di Servizio di tipo "I" solo quando il chiamante è classificato Assistente (Passcom studio)
MXRS	E' terminale Utente con interfaccia eseguita da riga di comando esterna
MXRSBATCH	E' terminale Utente senza interfaccia eseguita da riga di comando esterna
SHAKER	E' terminale di Servizio di tipo "L"

Il Terminale è Utente quando rappresenta il punto d'ingresso nel Gestionale ed attiva un processo padre (non un "clone" di Servizio). Non è possibile con Sprix e Collage controllare il tipo dei terminali aperti, in quanto tale possibilità si avrebbe solo all'apertura del Gestionale e Sprix e Collage si attivano solo dopo l'accesso in un'azienda, pertanto occorre un controllo personale all'apertura del programma.

PASSBUILDER

E' la piattaforma di sviluppo e di gestione delle personalizzazioni cosiddette "App"; l'attivazione di PassBuilder non pregiudica la configurazione di personalizzazioni preesistenti alla politica PassBuilder, per le quali, restano in vigore le politiche precedenti alla versione 2016H. Le personalizzazioni preesistenti possono coesistere con la possibilità di generare "App" e sono convertibili in "App". (**)

Per sviluppare la cosiddetta **App Utente** è necessario che l'installazione attivi sia **MDS** che **PassBuilder**.

Per installare ed eseguire la **App Utente** è necessario che l'installazione attivi **PassBuilder**; se Cliente **non ha MDS attivo**:

- Il Cliente può eseguire la App Utente realizzata dal suo Partner.
- Il Partner autore della App Utente può pubblicarla nella vetrina **PassStore** come **PassApp**; indipendentemente dall'autore il Cliente può eseguire qualsiasi PassApp.
- MDS consente il trasferimento di App fra Partner tramite marcatura del Partner che la riceve.

PASSSTORE

La pubblicazione di App nella vetrina online PassStore è un'opportunità che consente di distribuire applicazioni gratuite e non; la distribuzione di una medesima PassApp a più di un'installazione comporta, comunque, il conteggio per singola installazione.

Casi possibili

Cambio Dealer/Codice Contratto: se il cliente **non dispone di MDS**, nell'eventualità che avvenga una modifica del Dealer (es. cambio di Partner o di Ragione Sociale e p.Iva) è necessario che l'App sia ricompilata dall'installazione con il nuovo Dealer. Se la trasformazione dei programmi Sprix / Collage / MyDB in Applicazioni PassBuilder **non è ancora avvenuta** (per esempio, tramite la funzione di trasformazione disponibile ad aziende non aperte), prima di eseguire la trasformazione si consiglia di procedere alla modifica del Dealer/Codice Contratto; se la trasformazione in App è già avvenuta, la modifica del Dealer/Codice Contratto può invalidare l'App (App non valida) e richiederne una nuova compilazione che la codifichi con il nuovo dealer; la ricompilazione potrebbe richiedere il preventivo export dei programmi dall'installazione del cliente. In ogni caso, l'assenza dei sorgenti richiede la licenza MDS, altrimenti, la presenza dei sorgenti consente di ricostruire ex-novo l'App, in modo che questa si generi con il codice installazione che li ha importati, compilati e applicate le modifiche.

Trasferimenti di Applicazioni fra Partner: premessa l'esistenza dei sorgenti nell'app, oppure in alternativa, di licenza MDS o pubblicazione PassStore, nel passaggio di App fra Partner, l'App in oggetto dev'essere ricompilata dal Partner ricevente prima di essere importata

nell'installazione a cui si riferisce come dealer di compilazione.

(*) La modalità "pubblica" significa che l'applicazione può funzionare solo se il sorgente ed il suo relativo eseguibile sono entrambi collocati nell'installazione; la modalità "privata", invece, significa che l'applicazione può funzionare anche con il solo eseguibile, ma richiede l'attivazione del livello MDS (per l'esecuzione "privata" senza livello MDS contattare la divisione commerciale Passepartout S.p.A.). **Attenzione:** *i vecchi eseguibili compilati prima del rilascio di MDS (versione 2003G), se ricompilati richiedono la presenza del sorgente oppure la licenza di sviluppo MDS; a tal proposito si ricorda che in mancanza del sorgente non è consentito il reverse engineering non autorizzato dalla dichiarazione espressa e sottoscritta da parte del titolare di tutti i riservati diritti. **Attenzione:** l'utilizzo della funzione "elimina sorgenti sprix collage app" presente nell'ambiente PassBuilder senza disporre del necessario livello MDS invalida l'Applicazione e la rende non utilizzabile.;*

(**) chiarimenti sul passaggio di Sprix e MyDB in PassApp sono descritti nella circolare "CC_SG-PAC_PassaggioInPassApp_170529" e nella circolare "CC_SG-PAC_NovitaMeeting_170517".



SPRIX/COLLAGE

17/04/2024

STRUTTURA DEL MANUALE

1. INTRODUZIONE (Ambiente di Sviluppo e Gestione dei Programmi)

Questa prima parte è dedicata ai nuovi collaboratori che desiderano sviluppare Applicazioni Passepartout attraverso gli elementi di programmazione Sprix e Collage presenti fra quelli disponibili nella piattaforma PassBuilder integrata nel Gestionale Passepartout. Si tratta di un capitolo dedicato ad un esempio dimostrativo per descrivere l'ambiente di sviluppo Sprix/Collage integrato nel Gestionale Passepartout. Il capitolo riporta, infine, delle indicazioni sulla gestione dei programmi ed in cui la cui nomenclatura dei file è differente dalla gestione precedente solo a livello terminologico. Le suddette informazioni possono essere utili anche agli utenti esperti.

2. COMANDI PER ARGOMENTO (Linguaggio, Sintesi dei comandi con Regole, Esempi e Casi)

Questa seconda parte aggrega i comandi del linguaggio Sprix/Collage in tabelle per categoria di argomento. Ciascuna tabella elenca i comandi in modo sintetico e, finchè possibile, di rispettarne l'ordine logico di utilizzo. Le tabelle riportano piccoli esempi, anche tramite link e casi d'uso.

Nota: alcuni esempi potrebbero non essere più riproducibili in quanto non più allineati all'evoluzione grafica e/o funzionale del Gestionale, oppure, sebbene ancora funzionanti potrebbero non essere più efficienti e superati da nuove funzioni più avanzate di Sprix/Collage e/o del Gestionale. Alcuni di questi esempi e casi d'uso provengono dall'attività di assistenza on-line (Supporto EasyPass) e restano documentati, **ma non modificati**, sia per ragioni di "evoluzione storica" sia perchè i programmi Sprix/Collage sono portabili nel tempo e nelle versioni, fino a quanto è più possibile. Le correzioni possono avvenire, ma solo in caso di evidenti errori. Gli esempi possono avere font e formattazioni differenti: il "copia e incolla" del testo potrebbe interpretare e riportare dei caratteri speciali che potrebbero "sporcare" il sorgente ed impedirne la compilazione e/o il funzionamento (per esempio i doppi apici di taluni font non vengono riconosciuti e si "tramutano" in altro); in tali casi è opportuno fare transitare il codice copiato in un editor ASCII puro ed eventualmente apportare delle modifiche massive a tali caratteri in modo da uniformarli a quanto richiesto dall'Editor Sprix.

3. COMANDI IN ORDINE ALFABETICO (Dettaglio dei comandi con relativi Parametri)

Questa terza parte contiene la libreria di tutti i comandi elencati in ordine alfabetico, con il dettaglio dei parametri per ciascuna voce e le particolarità delle condizioni di utilizzo.

4. MDS - SHAKER (linee guida per sviluppare applicazioni esterne sfruttando comandi Sprix, ma tramite altri linguaggi)

La quarta parte riguarda Shaker. Consiste in una suite di componenti di sviluppo che consente di sviluppare personalizzazioni del Gestionale che sfruttano il motore SPRIX ma senza usarne il linguaggio, ovvero, tramite i linguaggi di altri ambienti di sviluppo (OCX, .NET, JAVA). Richiede apposita Licenza MDS (Mexal Development suite).

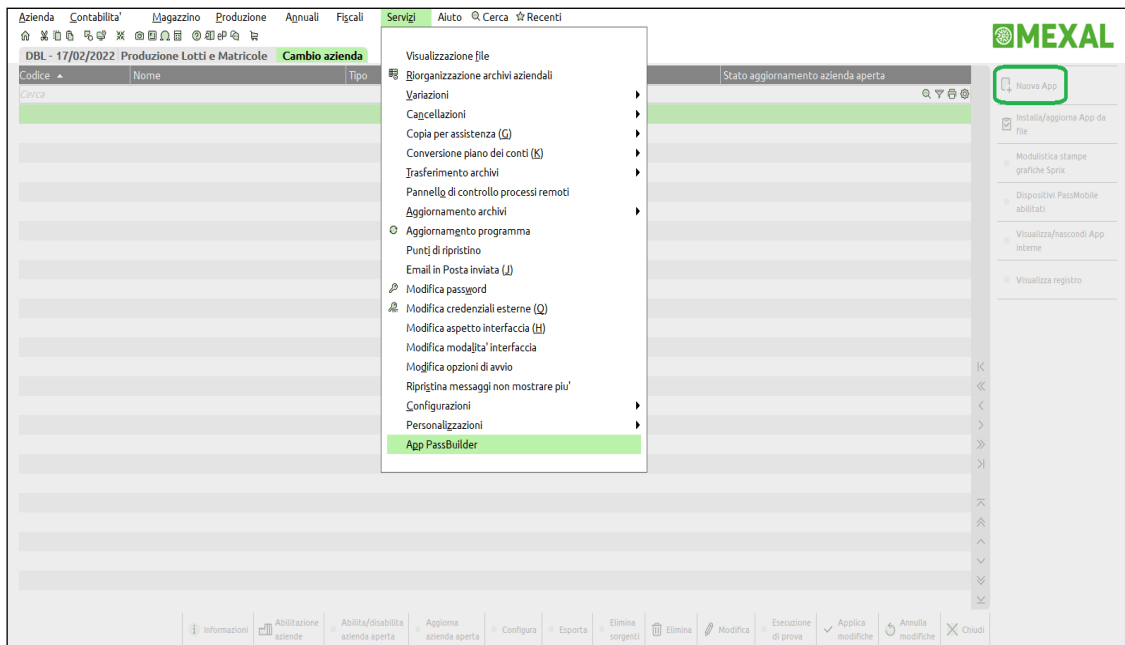
5. APPENDICE: implementazioni per versione di rilascio

La quinta e ultima parte è un'appendice sulle implementazioni delle versioni di rilascio più recenti; è una sintesi degli ultimi leggimi, i dettagli e gli approfondimenti sono riportati nella sezione 2^ e sezione 3^.

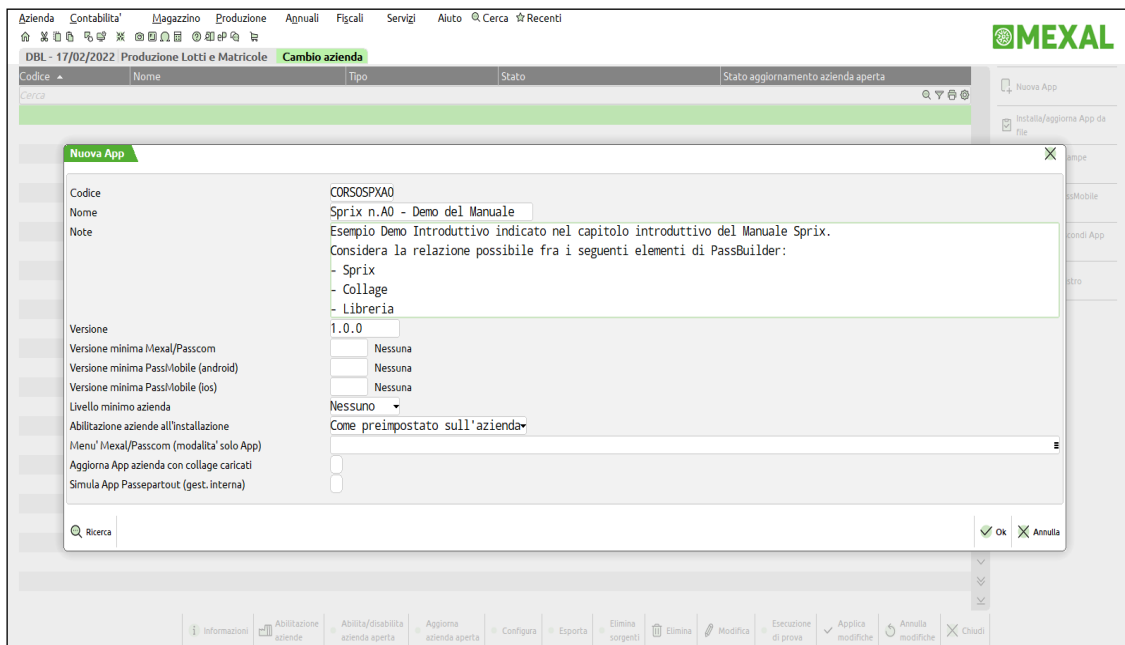
INTRODUZIONE: AMBIENTE DI SVILUPPO

ACCENNI SULLA PIATTAFORMA

L'ambiente di sviluppo PassBuilder ti consente di realizzare Applicazioni Passepartout: accedi alla voce "App PassBuilder" collocato nella voce di menu "Servizi", quindi seleziona "Nuova App".



Inserisci i dati identificativi dell'Applicazione.



Menu: Azienda Contabilità Magazzino Produzione Annuali Fiscali Servizi Aiuto Cerca Recenti

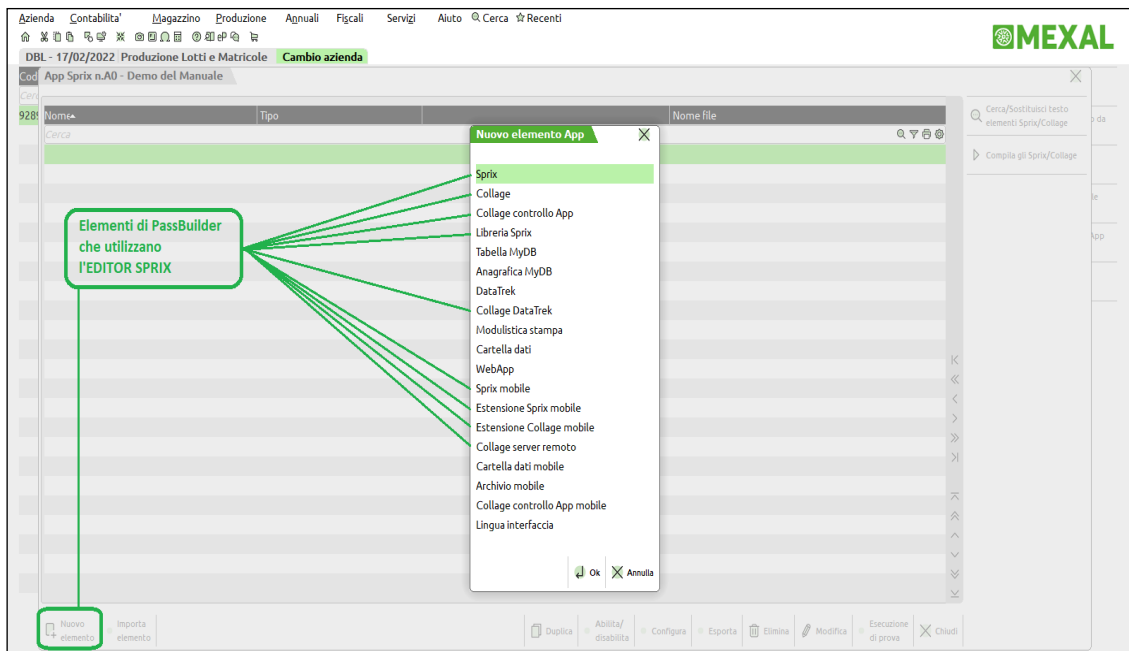
DBL - 17/02/2022 Produzione Lotti e Matricole **Cambio azienda**

Codice	Nome	Tipo	Stato	Stato aggiornamento azienda aperta
928900CORSOSPXA0	Sprix n.A0 - Demo del Manuale	App Utente	Nuova	Azienda DBL App non abilitata

Informazioni | Abilitazione aziende | Abilita/disabilita azienda aperta | Aggiorna azienda aperta | Configura | Esporta | Elimina sorgenti | Elimina | **Modifica** | Esecuzione | Applica modifiche | Annulla modifiche | Chiudi

Moduli: Nuova App, Installa/Aggiorna App da file, Modifica stampe grafiche Sprix, Dispositivi PassMobile abilitati, Visualizza/hasconci App interne, Visualizza registro

Successivamente puoi scegliere gli elementi di sviluppo che servono per generare i contenuti dell'Applicazione: sono elencati e selezionabili, hanno diversa natura e funzione in base a quello che l'Applicazione deve fare.



Questo manuale tratta principalmente tre elementi: "Sprix", "Collage" e "Libreria Sprix"; ciascuno di questi consente di generare files sorgenti contenenti codice scritto in linguaggio Sprix tramite un apposito Editor. Il linguaggio Sprix è di tipo sequenziale e procedurale, elaborato riga per riga, e dispone di un vasto insieme di comandi, funzioni, istruzioni e variabili per interagire con il Gestionale; alcuni comandi riguardano altri elementi dell'Applicazione, quindi anche questi componenti usano lo stesso Editor, per esempio "Sprix mobile" e "Collage Server Remoto". (v. immagine).

Elementi come la "Anagrafica MyDB" e la "Modulistica stampa", invece, sebbene il linguaggio Sprix abbia specifici comandi per utilizzarli non si costruiscono con lo stesso Editor, ma tramite appositi moduli. Esempio, un programma Sprix che scrive dati su tracciati personalizzati di una "Anagrafica MyDB" dispone di chiamate alla base dati MyDB ed istruzioni per scrivere nei suoi record, ma la natura e la struttura dei record non si costruisce con l'Editor, poichè si crea con appositi moduli e tabelle.

Un'Applicazione può essere costituita da un insieme di vari programmi "Sprix" e "Collage" fra essi combinati e relazionati, ma può essere anche costituita da uno solo di questi. L'esempio introduttivo che seguirà è composto da tre elementi: un programma che opera ad "eventi" (Collage) contenente un codice sorgente "riutilizzabile" (Libreria) impostato con un comando di tipo shell per lanciare ed eseguire un ulteriore programma (Sprix) come se questo fosse un modulo o una funzione di servizio.

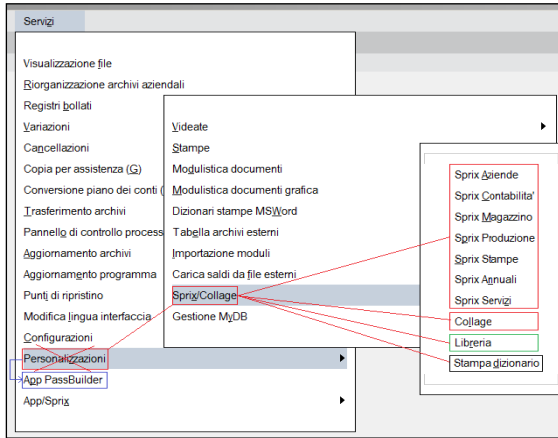
Il Collage è un programma contestualizzato nei moduli standard del Gestionale e viene eseguito quando si verificano determinati eventi; quando questi eventi accadono, il modulo standard del Gestionale cede il controllo al programma Collage, ma poi lo riprende quando il Collage ha finito il suo compito. Collage, quindi, non può "corrompere" la struttura originaria del Gestionale, poichè semplicemente nasce e ci muore "sopra", per gestire l'Input (per controllare, bloccare, aggiungere, automatizzare l'operatività utente), l'Output (per leggere i dati a video, mascherare il client, modificare le stampe e la modulistica dei documenti) per eseguire dei processi e per dialogare con database, applicativi e device esterni.

Un programma Sprix, invece, rispetto a Collage può essere eseguito in modo autonomo e indipendente come modulo aggiuntivo del Gestionale e consente di accedere alla sua base dati in lettura diretta e/o con cicli tabellari, ma anche in modifica. Essendo indipendente, anche Sprix non è in grado di "corrompere" l'impianto standard del Gestionale.

Entrambi i programmi Sprix e Collage possono sfruttare porzioni di codice in comune (Libreria) e possono lanciare ulteriori programmi Sprix come processi aggiuntivi e separati, ma un programma Collage non può intervenire su un programma Sprix, poichè non esistono "eventi" Sprix.

Per utilizzare i programmi Sprix e Collage di un'Applicazione occorre anzitutto identificarli e collocarli nel contesto del Gestionale che si desidera personalizzare o verticalizzare. Prima di spiegare come avviene, può essere utile un breve sguardo al passato.

Prima della nascita della piattaforma di sviluppo PassBuilder i programmi di personalizzazione e verticalizzazione si creavano accedendo nella specifica voce di menu [Servizi-Personalizzazioni-Sprix/Collage] dedicata a questi due strumenti di sviluppo, considerabili oggi come i precursori di PassBuilder, dato che tutt'ora sono il mezzo di relazione principale con gli archivi e moduli del Gestionale. L'obsoleta voce "Personalizzazioni" è stata sostituita dalla voce "App PassBuilder".



Nota: l'elenco [Sprix/Collage] è ancora impostabile come "visibile", ma non è accessibile (v. flag "Disabilita Sprix/Collage/MyDB non App" nell'elenco delle Configurazioni relative alla funzione [F11] "Impostazioni di configurazione predefinite" del menu "ZCU" [Servizi-Configurazioni-Amministrazione Utenti]).

Le prime sette voci del sottomenu "Sprix/Collage" rappresentavano i menu del Gestionale in cui collocare, in fondo a ciascuno di essi, fino a 99 programmi Sprix come moduli aggiuntivi indipendenti.

La voce Collage, invece, consentiva di collocare fino a 99 programmi Collage nell'intero Gestionale o in sue determinate "zone", dall'intero menu/sottomenu fino alla specifica etichetta di un campo di input del singolo modulo standard.

La differenza principale fra il programma Sprix e quello Collage è che Sprix è un modulo aggiuntivo integrato in coda ai menu principali del Gestionale ed eseguibile in modo "a se stante". Collage, invece, è un programma ancora più integrato in quanto funziona direttamente nelle "zone" dei moduli standard quando accade un determinato "evento".

Nella voce di Libreria si collocavano fino a 99 sorgenti in cui scrivere routines di codice Sprix potenzialmente in comune a più programmi Sprix e Collage, ottimizzando così la quantità di codice sviluppato in quanto "riutilizzabile".

La voce "Stampa Dizionario" è conservata e spostata di un livello nel menu [Servizi - Personalizzazioni]. Consente di consultare tutta la struttura della base dati del Gestionale.

I suddetti concetti espressi nella tabella permangono in PassBuilder, poiché è cambiato solamente il modo con cui dichiarare e collocare questi elementi nel Gestionale, ovvero, attraverso maschere di input al posto della vecchia rappresentazione ad elenchi¹ del sottomenu [Sprix/Collage] e che non è più operativa eccetto la possibilità di attivare la sola visualizzazione dei programmi in elenco.

Per realizzare un programma Sprix oppure Collage occorre accedere nell'apposita voce "App PassBuilder" e seguire le prime indicazioni del manuale PassBuilder (contenuto nel [manuale utente](#) e che in questo contesto riprendiamo e sintetizziamo) su come generare un'Applicazione (tasto [Nuova App]) con attribuzione di Codice, Nome, Versione, ecc.; dopo questo "battesimo" c'è la fase di costruzione (tasto [Modifica App]) che avviene scegliendo gli elementi e le funzionalità che la costituiranno, attraverso il tasto [Nuovo elemento].

¹ Ricordiamo che alcune immagini potrebbero non essere aggiornate e quindi potrebbero riferirsi alla specifica (obsoleta) voce di menu dedicata all'ambiente di sviluppo [Sprix/Collage]. Questo intero ambiente è attualmente collocato fra gli elementi della piattaforma di sviluppo PassBuilder a cui si può accedere dalla voce 'App PassBuilder' del menu 'Servizi'. Le caratteristiche di utilizzo, fruibilità, funzionalità e concetti dell'originario ambiente (Editor Sprix/Collage/Libreria), però, **non cambiano e restano inalterate**, nonostante alcune visualizzazioni e collocazioni di descrizioni e campi input o terminologie e nomenclature dei file, si presentino apparentemente diversi. Nella dimostrazione dell'esempio, quindi, possono essere presenti immagini relative al vecchio stile di rappresentazione ad elenchi dei programmi Sprix, Collage e Libreria, ma i campi delle maschere di input degli elementi Sprix, Collage e Libreria di PassBuilder rendono intuibile come questi siano stati sostituiti ai suddetti elenchi, e cioè, quanto nel menu era prima rappresentato da un elenco adesso nelle maschere di PassBuilder è rappresentato da dei campi alfanumerici per la descrizione del programma, la relativa collocazione nei menu, l'ordinamento di esecuzione, ecc...

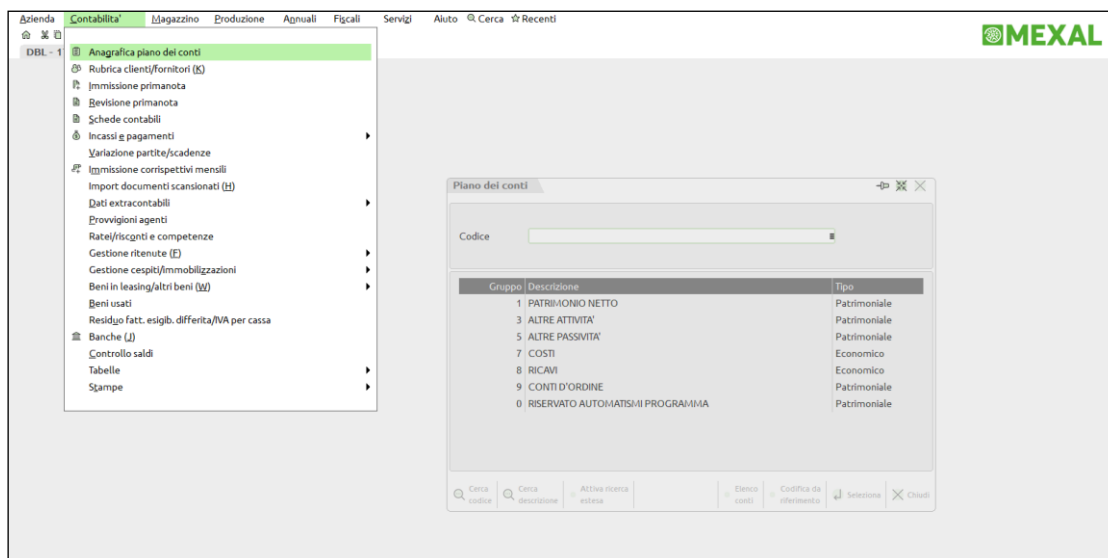
DEMO: COLLAGE + SPRIX + LIBRERIA

Vediamo un piccolo esempio di combinazione di questi tre elementi componenti (Sprix + Collage + Libreria), al fine di ottenere una semplice introduzione a questo ambiente e per dimostrarne le potenzialità, non dimenticando che il contesto in cui si opera è ben più ampio ed articolato, cioè quello del Gestionale: << all'ingresso del modulo "Anagrafica piano dei conti" del menu "Contabilità", un determinato "evento" deve attivare l'esecuzione di un altro programma per la visualizzazione di un messaggio e una successiva stampa >>

Attenzione: generalmente si dovrebbe iniziare a parlare da Sprix (in quanto è programma indipendente), poi si passa all'utilizzo della Libreria e infine Collage. Essendo questo solo un esempio demo (per combinarli tutti e tre insieme), risulta più opportuno iniziare a trattare da Collage.

Inizio a costruire l'Applicazione partendo dall'impostazione del programma che funziona ad "eventi", cioè "Collage". A differenza del programma Sprix, il Collage è un programma che funziona quando avviene qualcosa ad opera dell'utente, pertanto, svilupperò un Collage che sfrutta una Libreria, la quale esegue uno Sprix quando si manifesta una determinata condizione durante l'utilizzo dei moduli da parte dell'utente (condizione diversa rispetto all'esecuzione dello Sprix in modo autonomo e indipendente da una voce menu).

Inizialmente creo un Collage da collegare in un punto della "Anagrafica piano dei conti". Prendo nota del fatto che questo percorso di menu ha un codice che corrisponde alla sigla "CG" ovvero l'insieme in maiuscolo delle lettere sottolineate delle voci di menu "C"ontabilità e "A"nagrafica piano dei conti". Queste lettere sono utili come "tasti di selezione rapida dei menu" [Alt+C+G], ma possono corrispondere allo "ID Menu" che serve per "installare" il Collage su una determinata zona del Gestionale.



Decido che l'evento deve avvenire nel campo di input "Codice". Dal menu di Passepartout seleziono **Aiuto** e poi **Informazioni su...**, che apre una finestra contenente dati relativi allo "ID Menu", allo "ID Finestra" e "ID campo" (e tante altre informazioni utili per le richieste di assistenza), ma anche un pulsante **Informazioni Collage** che consente di conoscere il nome del campo e gli eventi attivi su di esso. La sequenza di tasti per aprire velocemente queste finestre è [F1]+[F2]+[ShiftF6].

Azienda Contabilità Magazzino Produzione Annuali Fiscali Servizi **AIUTO** Cerca Recenti

DBL - 17/02/2022 Produzione Lotti e Matricole **Cambio azienda**

MEXAL

- AIuto campo attuale
- Manuale utente
- EduPass
- EduPass area riservata
- Richiesta assistenza (EasyPass)
- PassStore
- Passepartout sul Web
- Documentazione di versione
- Tasti funzione e simboli
- Calcolatrice
- Indagine soddisfazione cliente
- Informazioni su Passepartout Mexal**

Piano dei conti

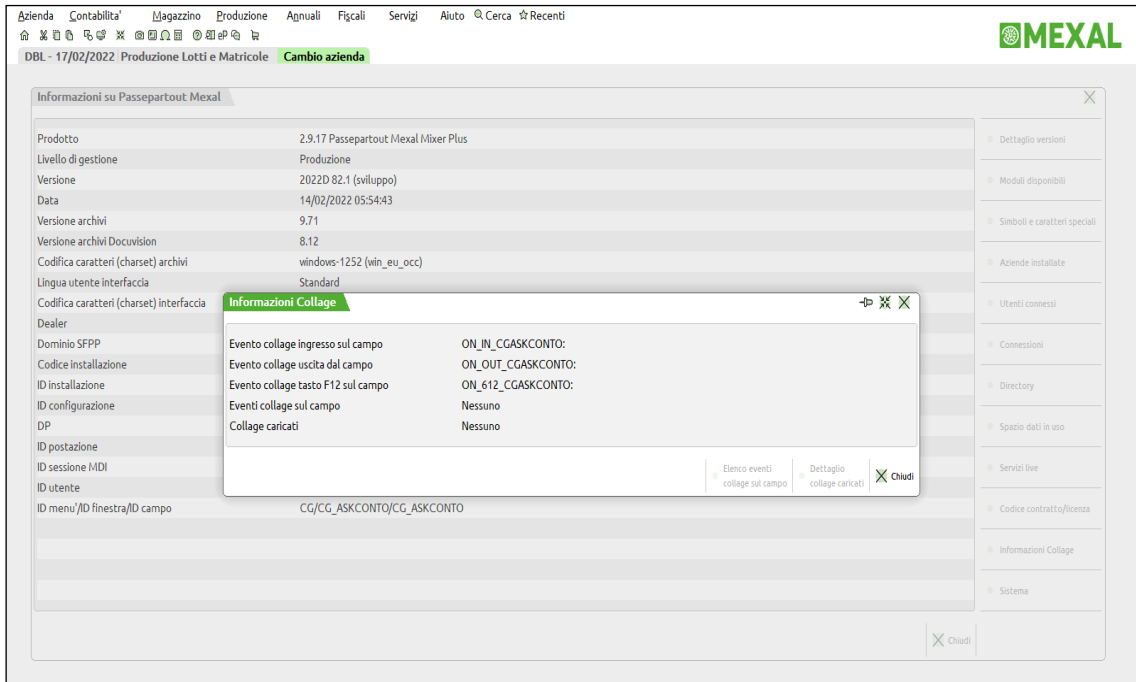
Codice

Gruppo	Descrizione	Tipo
1	PATRIMONIO NETTO	Patrimoniale
3	ALTRE ATTIVITA'	Patrimoniale
5	ALTRE PASSIVITA'	Patrimoniale
7	COSTI	Economico
8	RICAVI	Economico
9	CONTI D'ORDINE	Patrimoniale
0	RISERVATO AUTOMATISMI PROGRAMMA	Patrimoniale

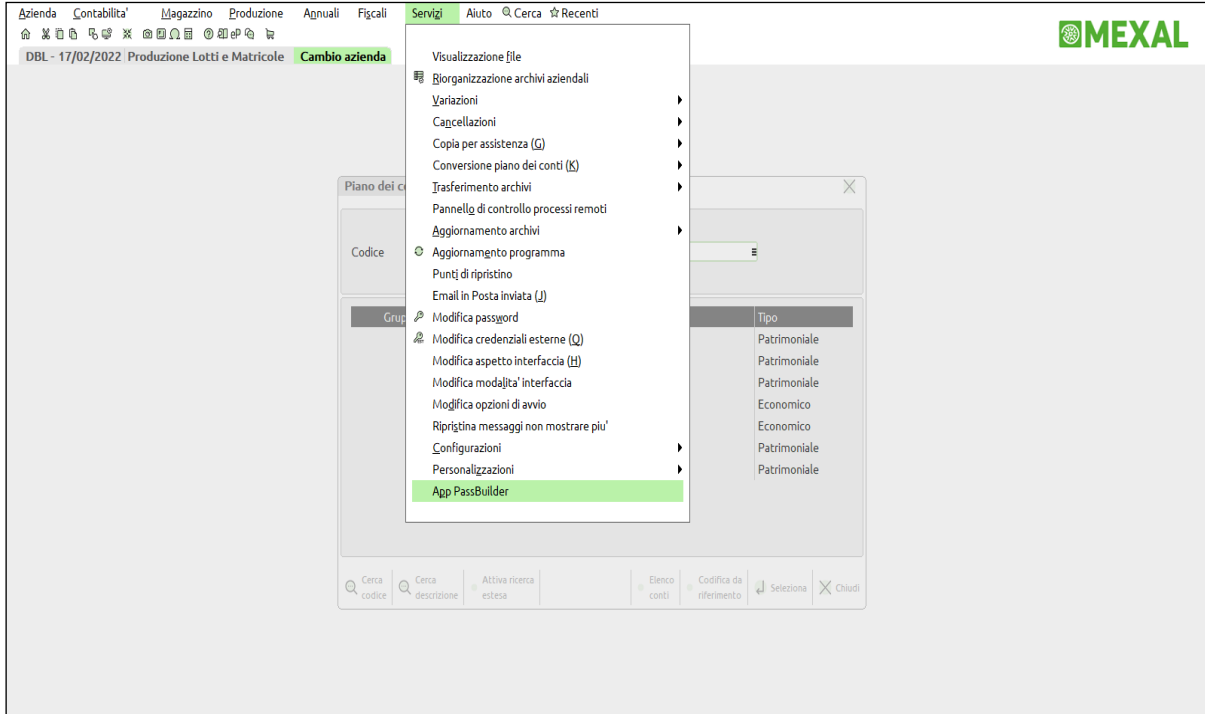
Cerca codice
 Cerca descrizione
 Attiva ricerca estesa
 Elenco conti
 Codifica da riferimento
 Selezione
 Chiudi

Il campo si chiama "CG_ASKCONTO", cioè "CG" come "Contabilità Anagrafica" e "ASKCONTO" come "Ricerca Codice". L'unione delle sigle senza carattere "_" forma il codice (ID) "CGASKCONTO" a cui sono collegati eventi: all'ingresso (ON_IN); all'uscita (ON_OUT); alla pressione del tasto funzione [F12] (ON_612).

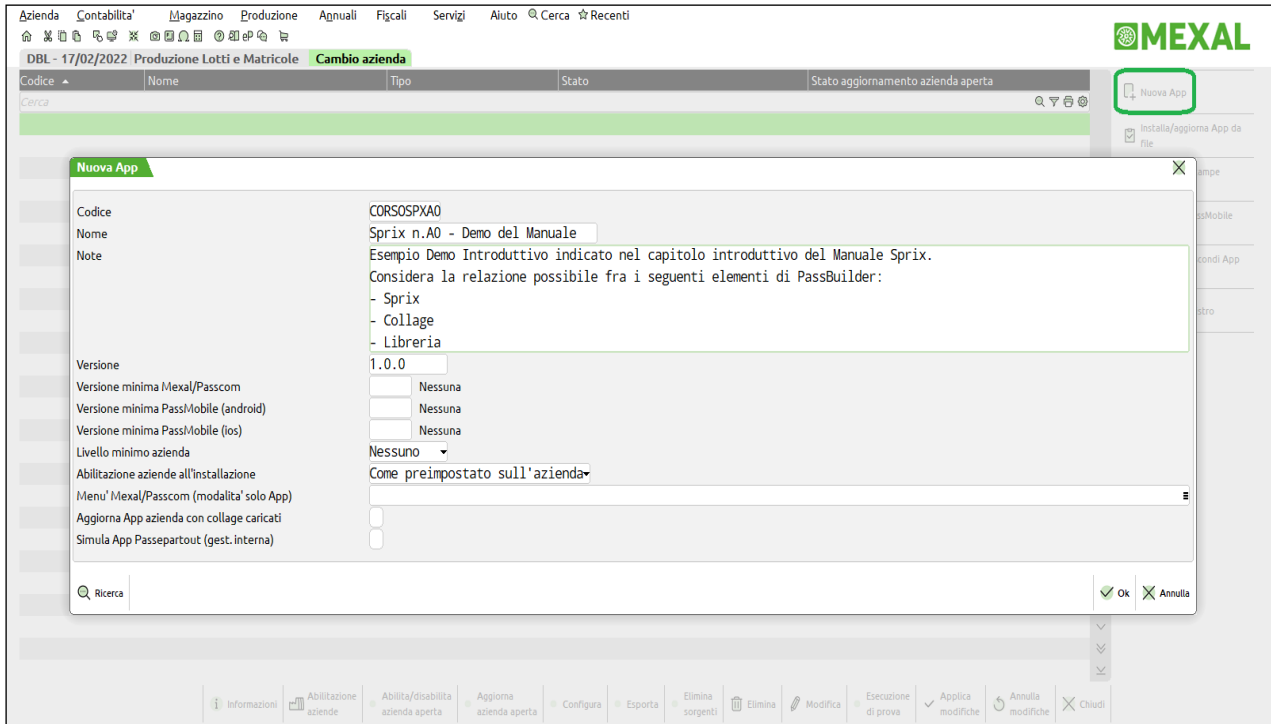
Decido che l'evento in cui dovrà avvenire l'esecuzione del Collage sarà "alla pressione del tasto F12", quindi, ON_612_CGASKCONTO:.



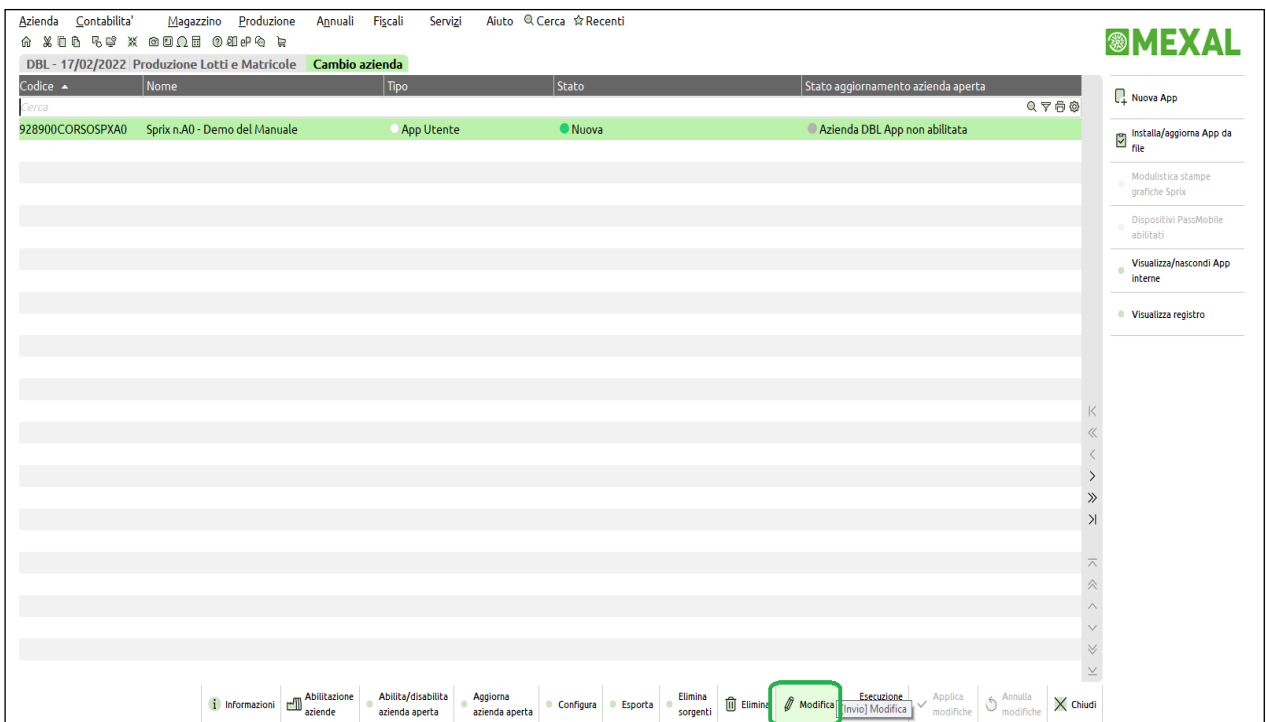
Inizio a costruirmi l'Applicazione che sarà composta da tre sorgenti (Collage + Sprix + Libreria) accedendo dal menu "App PassBuilder".

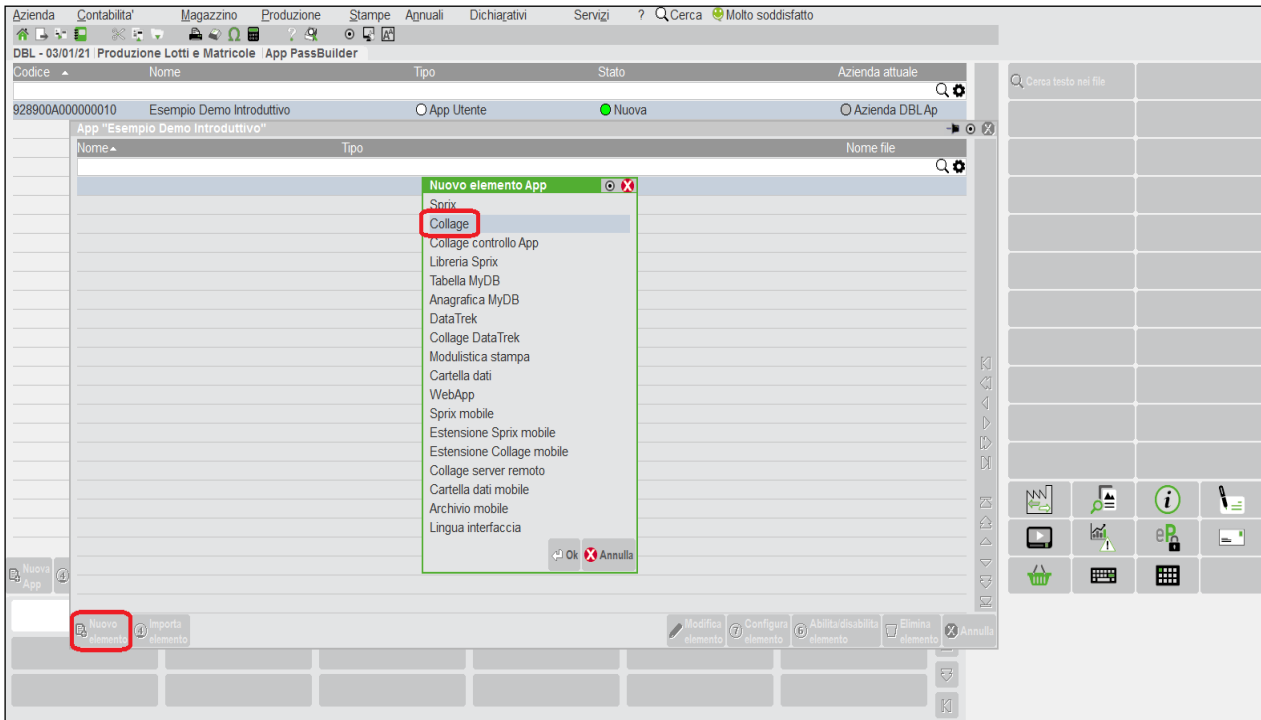


Premi [F4] "Nuova App" e ne inserisco gli elementi identificativi e le eventuali compatibilità con versioni e livello di dati gestiti.



Dopo la conferma con tasto [F10] "Ok", premi [Invio] "Modifica App" e poi [F4] "Nuovo elemento", infine, scegli l'elemento "Collage" per entrare nell'Editor per comporre il codice sorgente del Collage utilizzando il linguaggio Sprix.

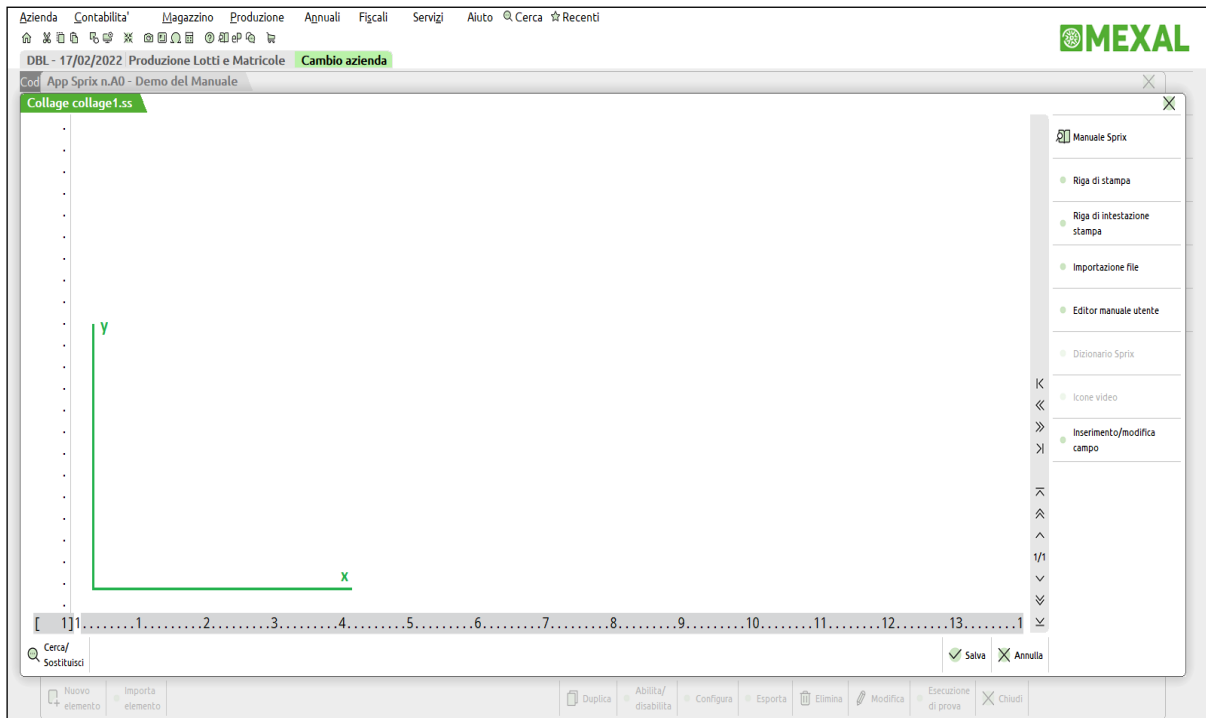




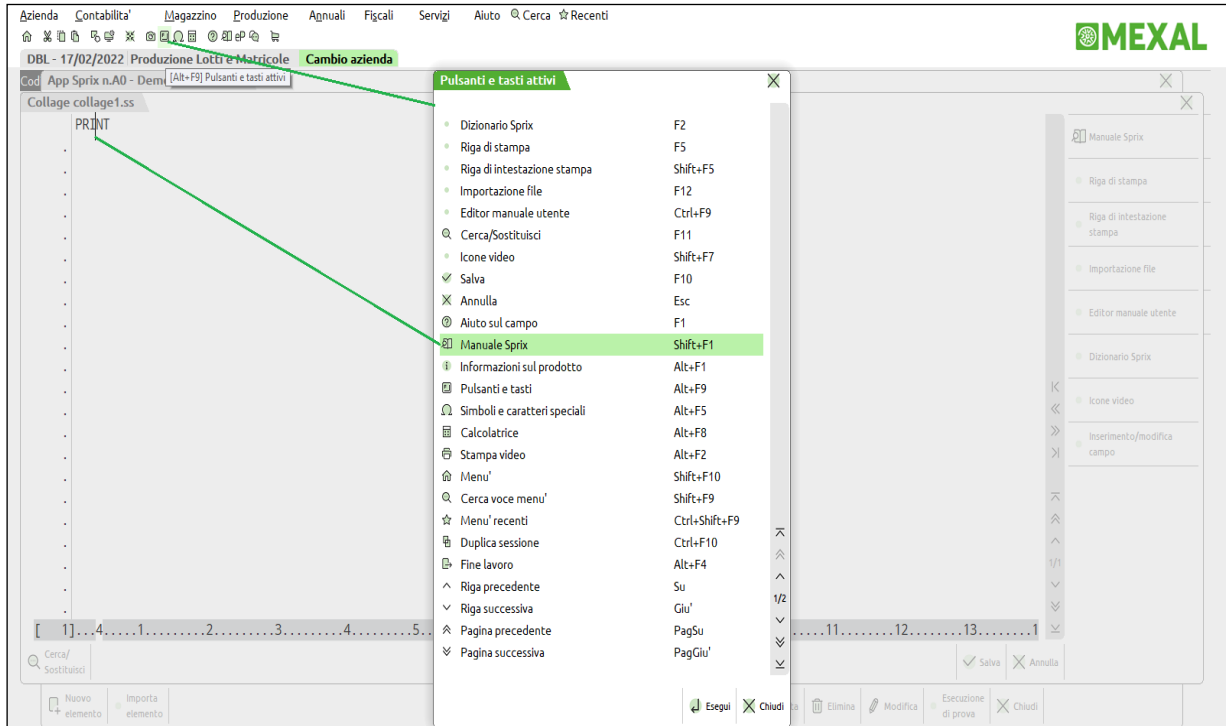
L'EDITOR del codice sorgente è un normale editor ASCII identico per tutti i programmi Sprix, Collage e Libreria; la modalità di realizzazione del codice non cambia rispetto al passato.

Il formato è lungo 3000 righe per 220 caratteri (ASCII 32-126 più una stringa di fine riga); questi "limiti" sono superabili sfruttando la Libreria.

Nella parte inferiore è visualizzata la posizione di riga e di colonna in cui si trova il cursore. La zona a sinistra dell'asse verticale caratterizza la tipologia della riga: il punto "." significa riga "vuota", ma puoi eliminarlo premendo [Invio] iniziando a scrivere trasformandola in operativa, oppure in un commento, oppure una riga di stampa.



E' disponibile il tasto funzione [Alt+F9] che visualizza due pagine di tasti funzione attivi.



Oltre ai tasti funzione specifici per determinate funzionalità (verranno esaminate in seguito), ci sono tasti strettamente dedicati all'editor per:

- **apertura di informazioni dettagliate sul singolo comando indicato [Shift+F1]:** come l'esempio in figura, premendo tale tasto funzione sopra un comando (un'istruzione, una funzione o una struttura di variabili), si apre questo Manuale Sprix alla corrispondente voce di libreria; attenzione: **non specificare alcun carattere "\$" o "(" anche se previsto dal comando**. Questo automatismo "a chiavi" sul manuale è uno dei motivi principali per cui il manuale Sprix è indipendente e separato dagli altri manuali che compongono PassBuilder. La pressione di [F1] è invece l'help di campo dove previsto
- **navigazione, ricerca e sostituzione di testo [F11]:** ricerche di tipo case sensitive o insensitive, per parola intera e ultime ricerche; sono previsti ulteriori tasti per ricercare "tutte" le parole oppure la "precedente" - "successiva" e che restano attivi anche dopo la chiusura della finestra di ricerca per proseguire sulle restanti occorrenze; anche l'elenco degli elementi dell'App ha questa funzione (potenziata dal campo di ricerca estesa full text) in cui, posizionandosi sulle righe dei riferimenti rilevati, è possibile aprire il file sorgente in cui il testo è stato trovato.
- **la conversione del testo:** dal carattere minuscolo a quello maiuscolo e viceversa (anche in sequenza mantenendo premuto il relativo tasto funzione [Shift+F11]);
- **tabulazione di più righe selezionate e la selezione a colonne del testo:** per rapide indentazioni e gestioni del testo, si esegue con la combinazione dei tasti [Shift+Alt] e con le frecce di navigazione e dei tasti per gestire il testo: [Shift+F11], [Tab], [Shift+Tab], [Canc], [BackSpace], [Ctrl+C/V/Z/X], [Invio], [Caratteri]. Note:
 - durante la selezione della colonna [Shit+Alt+Freccia] nel editor compare una colonna "grigia" che indica il posizionamento verticale di gestione del testo, cioè il punto d'inserimento o cancellazione multiplo. La successiva selezione del testo con le frecce di navigazione diviene in reverse "blu" poiché è pronto ad essere gestito, anche oltre il fine riga;
 - [BackSpace] è diverso da [Canc] in quanto non cancella la riga precedente, ma assume la riga attuale in quella precedente;
 - La memorizzazione [Ctrl+C] nella clipboard non consente la copia di tutti gli elementi di codice (es. campi di stampa);
 - L'import di file [F12] oltre ad essere di tipo testo può acquisire un altro sorgente (da locale o da remoto), ma non vengono importati i parametri di esecuzione [F6].

Alcune vecchie versioni del Gestionale consentivano di selezionare il "Cambio modalità editor" alla versione più obsoleta di editor; nelle versioni più recenti ciò non è più disponibile (il default è fisso al nuovo editor) pertanto *Passbuilder non consente più passaggi di editor e l'unico modo per apportare modifiche è quello di modificare il file pxconf.ini impostando la seguente voce oppure cancellando entrambe le seguenti due righe (per ambienti Live occorre segnalare in EasyPass una richiesta di assistenza Sistemistica):*

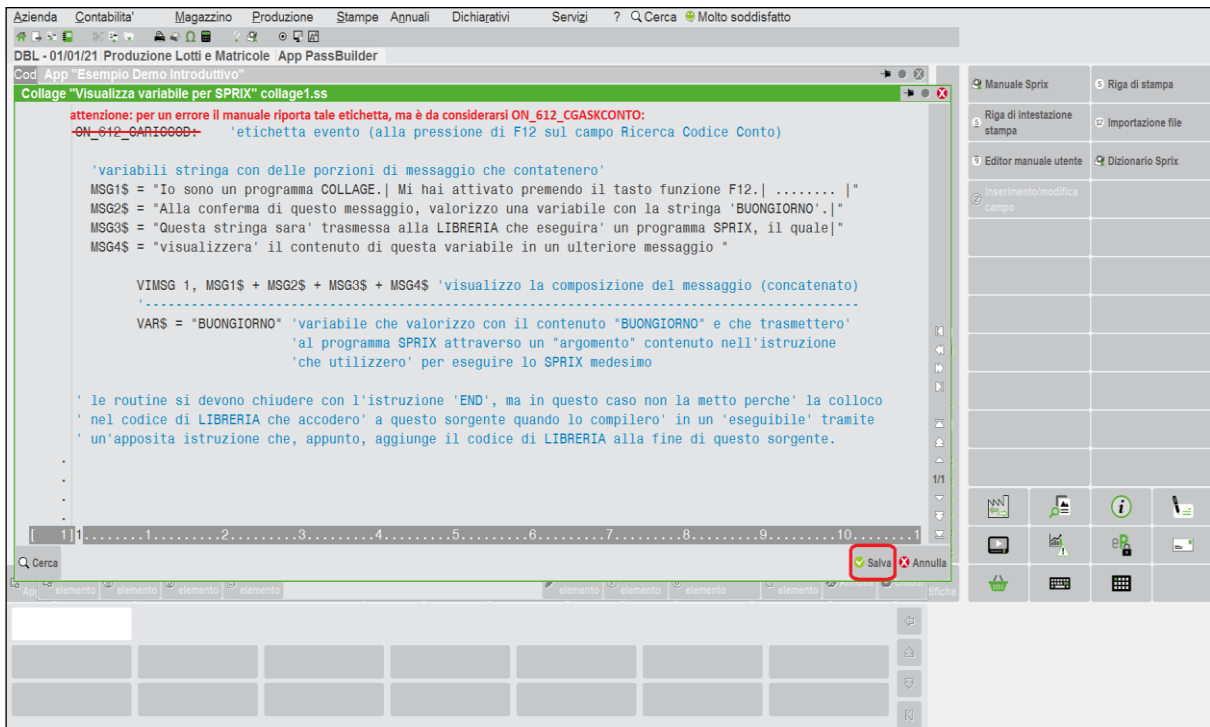
[SPRIX]

Neweditor=1

Nota: l'import con [Shift+F4] nell'applicazione di un elemento sorgente ".ss" datato può comportare una non perfetta compatibilità del vecchio file sorgente al nuovo editor, con possibili ripercussioni sull'editabilità del ciclotabellare; piuttosto che riscrivere il ciclotabellare è possibile importare (per "rinnovare") il medesimo codice con [F12] dal medesimo sorgente.

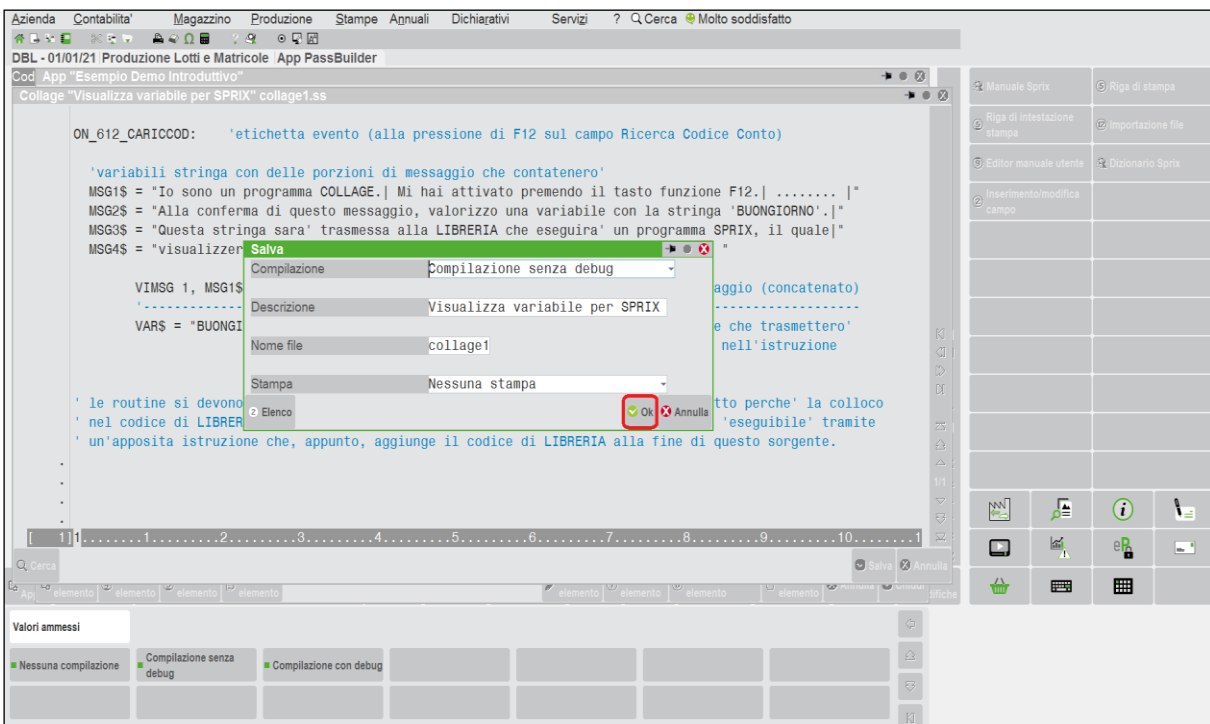
Se possibile, abituati alla tastiera e non al mouse, difficilmente trovi cose che puoi fare solo con il mouse. Apparentemente, il Passepartout è un ambiente "tradizionale", ma grazie a questi sistemi esperti di sviluppo, consente di realizzare particolari micro e macro funzionalità.

Torniamo a parlare della composizione del programma Collage. Inserisco il codice sorgente: le righe blu prefissate dall'apice sono dei commenti non operativi, quelle nera sono il codice vero e proprio. Traduzione del codice dell'immagine che segue: alla pressione del tasto [F12] inizializzo 4 variabili locali di tipo stringa (un insieme di caratteri alfanumerici) e le concateno in un messaggio da visualizzare. Infine, inizializzo una variabile locale di tipo stringa (alfanumerica) con il valore "BUONGIORNO". Attenzione, le immagini che seguiranno riportano erroneamente l'etichetta ON_612_CARICCOD:, ma è da considerarsi ON_612_CGASKCONTO:. Le immagini saranno aggiornate prima possibile.

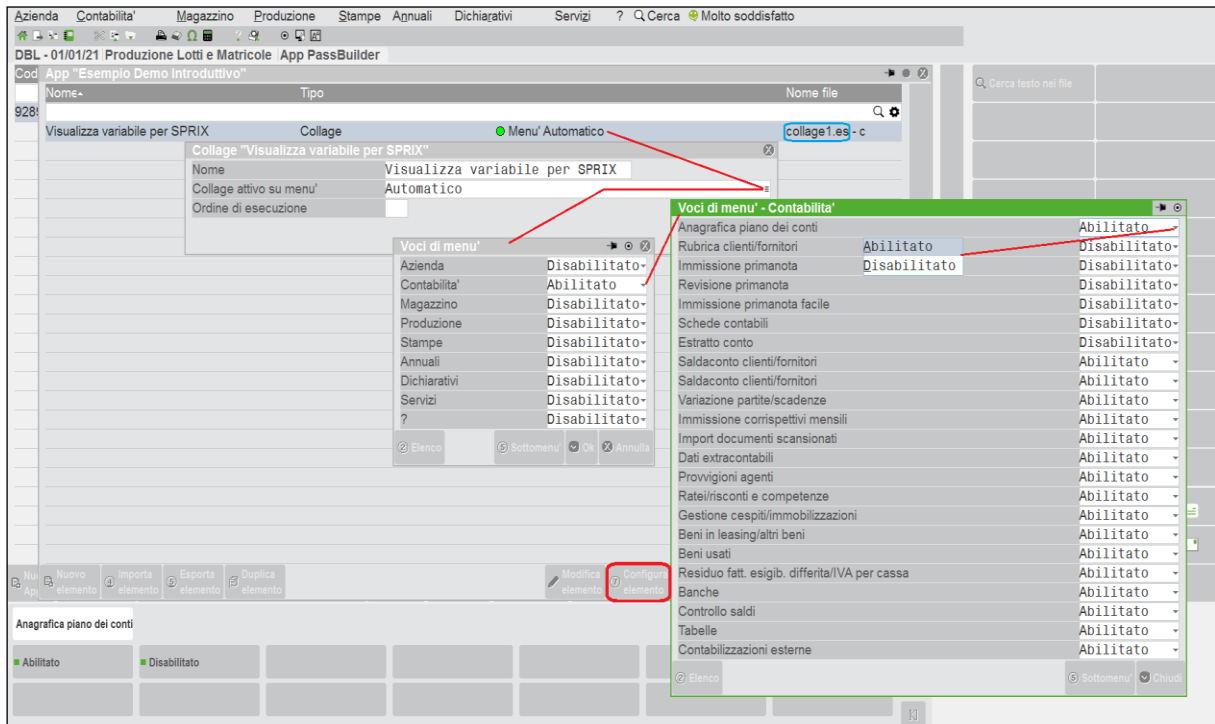


Salvando il programma l'Editor mi visualizza le informazioni sulle caratteristiche identificative e di archiviazione del file sorgente. La maschera di salvataggio propone dei valori di default e mi limito ad inserire solo la descrizione del programma Collage. Sono possibili anche altre azioni, ad esempio la possibilità di avviare un debug oppure di modificare il nome al file del programma oppure di stampare il codice sorgente in vari modi, ma per il momento rimandiamo questi argomenti.

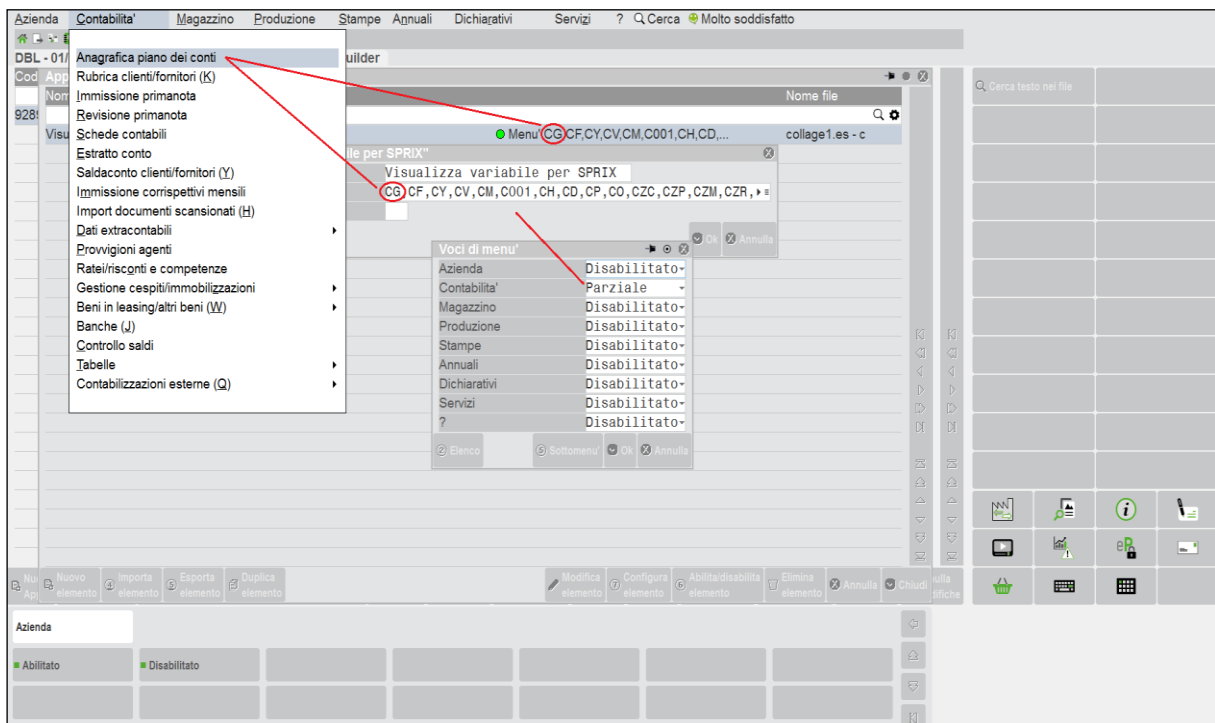
Premi "Ok" per compilarlo e salvarlo definitivamente: se ho scritto il codice in modo corretto il compilatore esegue la compilazione, altrimenti mi blocca perchè controlla e segnala errori di sintassi. Se volessi semplicemente salvare un lavoro temporaneo senza compilare, dovrei scegliere il salvataggio senza la compilazione.



Alla conferma l'elemento Collage è archiviato e compilato nell'Applicazione. Essendo Collage un programma che si attiva ed esegue il codice in base ad "eventi", dovrei configurare l'ambito di applicazione nel Gestionale.



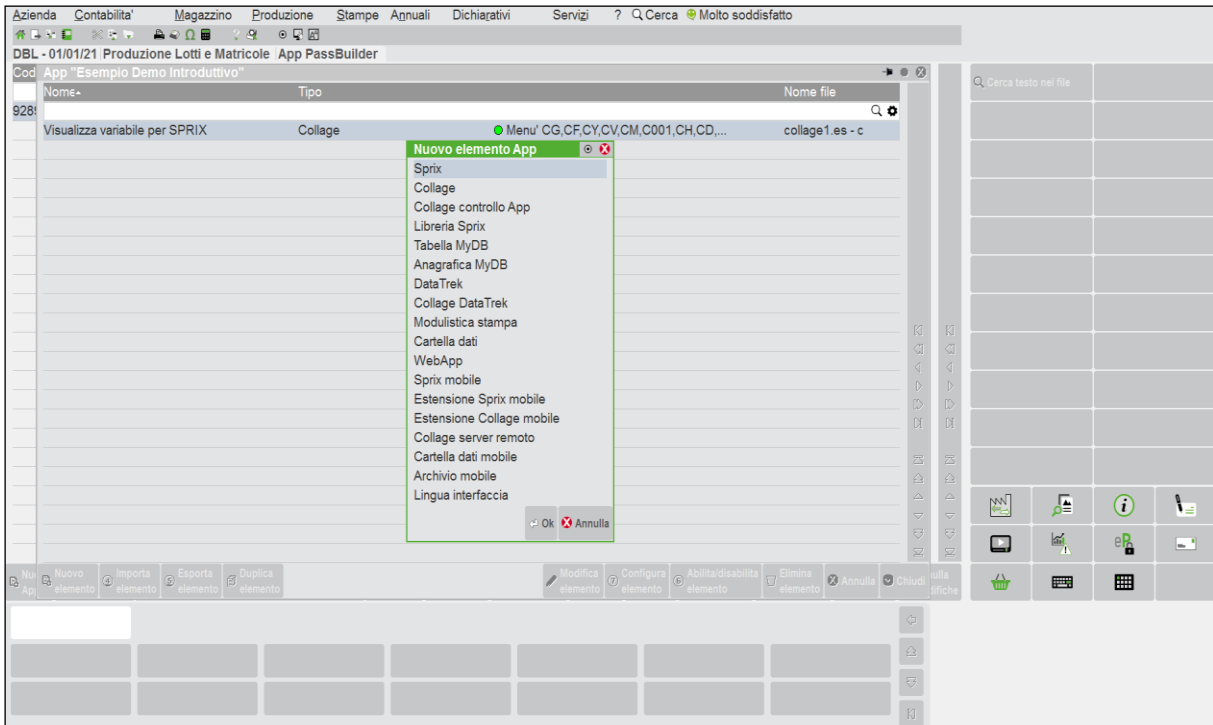
Se lasciassi "Automatico" il Collage sarebbe in "ascolto" sull'intera installazione, mentre invece, mi interessa solo una parte e quindi scelgo che sia attivo solo sulla parte di "contabilità" che mi interessa, fra cui il menu precedentemente indicato, ovvero "CG".



Ripeto: questo Collage che ho realizzato, stabilisce che alla pressione del tasto [F12] nel campo "Codice" dell'Anagrafica Piano dei Conti (menu CG) visualizza un messaggio e nel "frattempo", valorizza con la stringa "BUONGIORNO" la variabile VAR\$. Voglio assegnare questa variabile ad una Libreria e da questa, poi, ad un programma Sprix, ma prima di realizzare la Libreria, quindi, devo realizzare il programma Sprix a cui questa passerà il dato in questione.

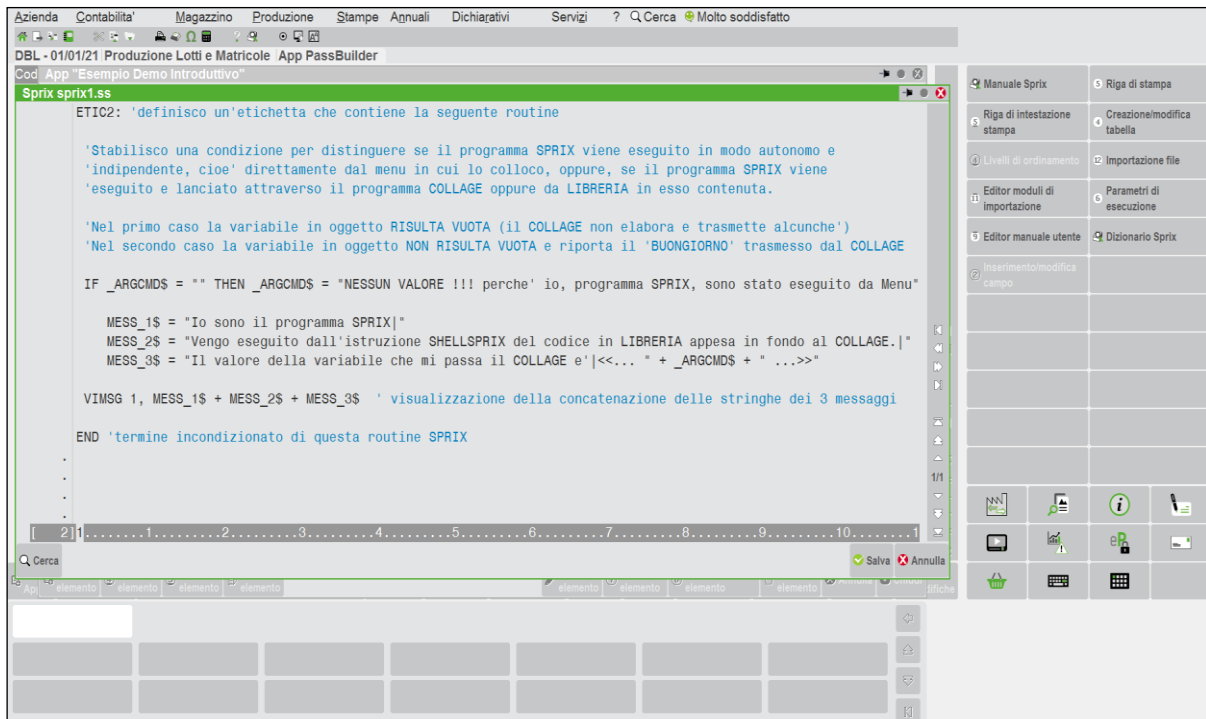
Nota Bene: l'utilizzo del codice di menu di selezione rapida (nell'esempio 'CG') è solo a titolo esemplificativo, poichè in linea generale corrisponde alla radice (ID menu) indicata nelle informazioni su Collage [F1+F1+ShF6] ed è questa quella da utilizzare.

Adesso devo realizzare il programma Sprix e deciderò di pubblicarlo come un sottomenu del menu "Azienda", ovvero, l'App che conterrà il mio Sprix si troverà nella voce di menu "Azienda". Come ho già operato per creare il Collage, accedo agli elementi dell'App e quindi all'Editor, ma questa volta dell'elemento "Sprix".

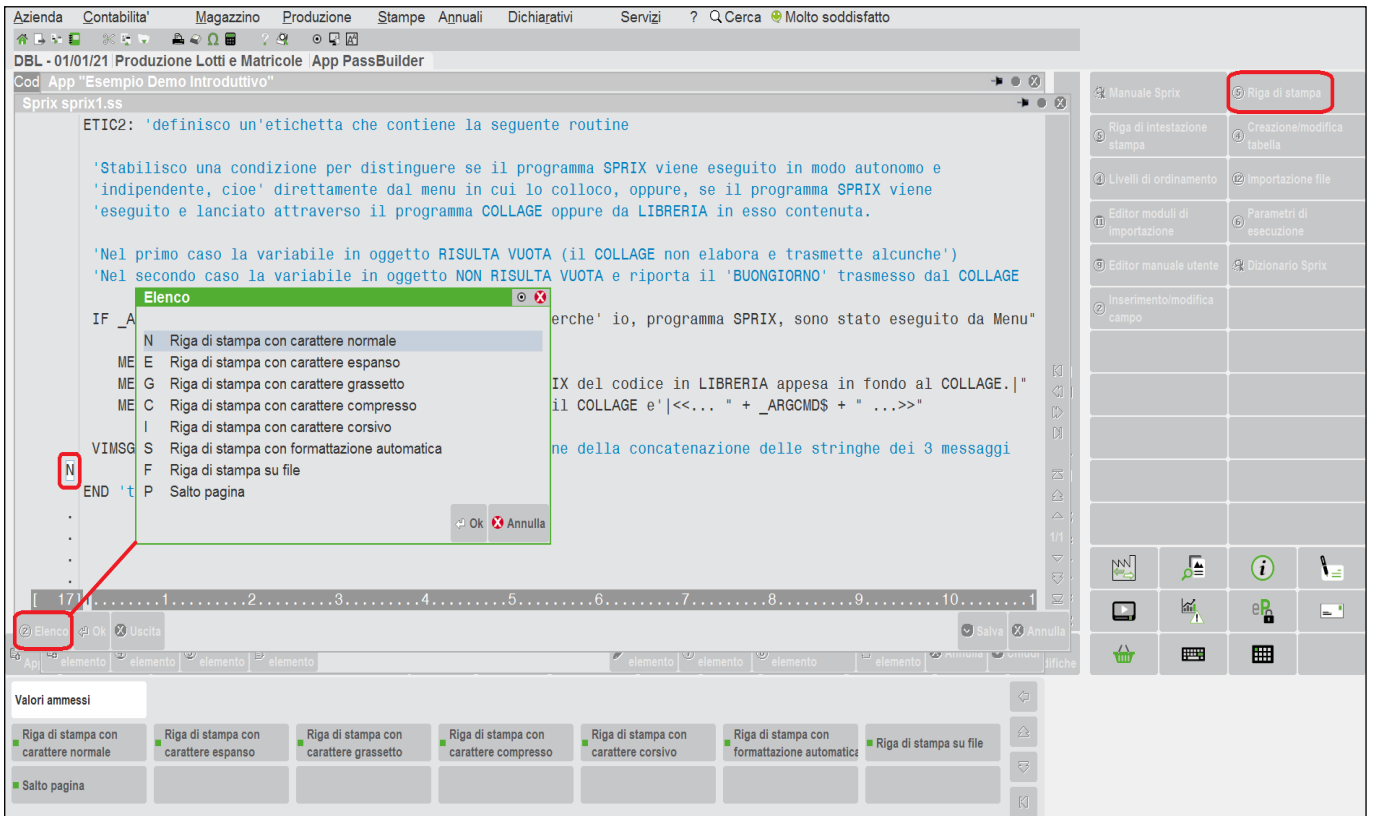


Inserisco il mio codice sorgente Sprix. Traduzione del codice: stabilisco la condizione secondo la quale, se la variabile `_ARGCMD$` è vuota, questa sia valorizzata con il messaggio "NESSUN VALORE...", dopo di che, inizializzo tre variabili locali di tipo stringa di cui la terza a sua volta contiene il valore della variabile e le concateno in un messaggio da visualizzare, infine, concludo il programma chiudendo la routine iniziata con l'etichetta che ho arbitrariamente battezzata "ETIC2".

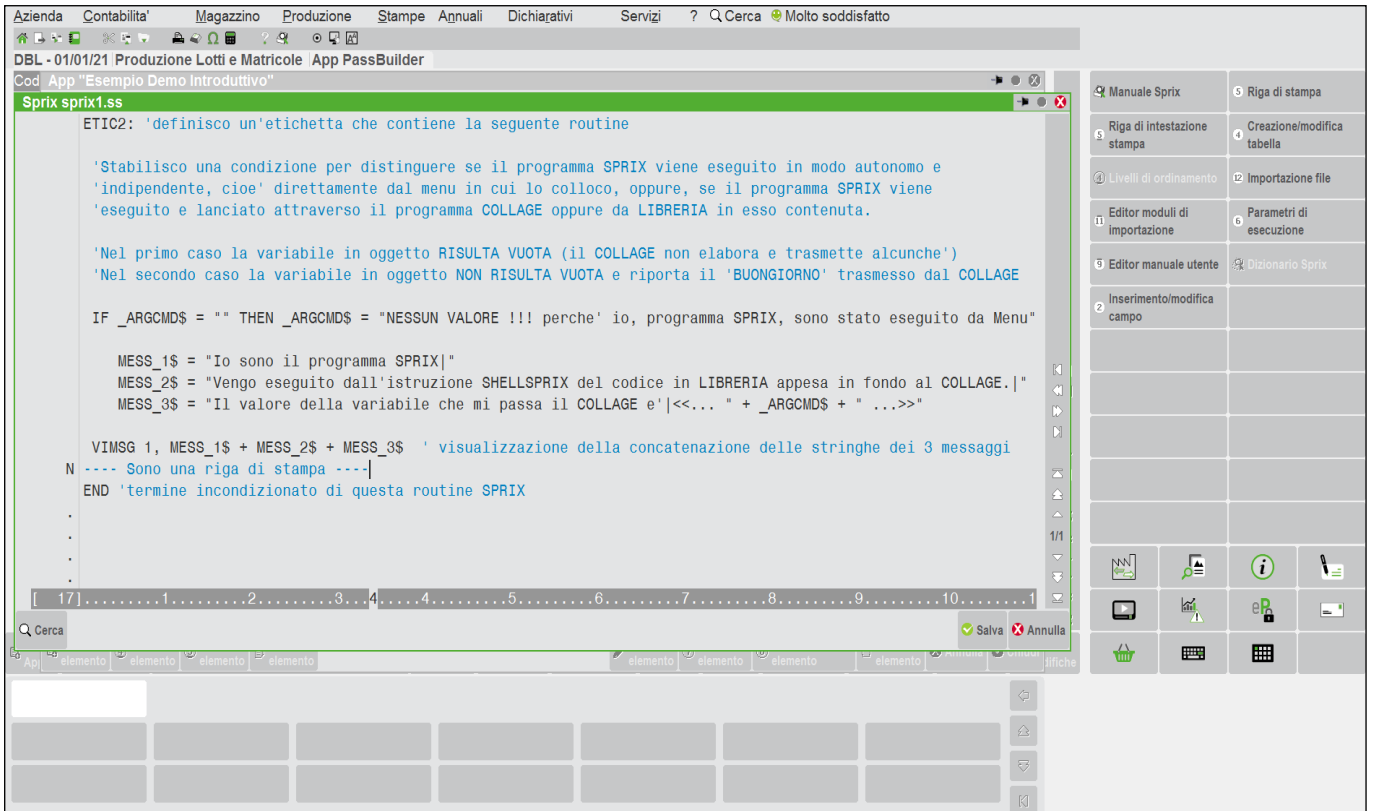
Il messaggio cambierà a seconda del valore che è passato alla variabile `_ARGCMD$`, cioè, "BUONGIORNO" se proviene da Collage tramite una variabile che "transita" per la Libreria (la quale, a sua volta, esegue lo Sprix), oppure, "NESSUN VALORE" se la variabile è vuota, poiché potrei eseguire lo Sprix in modo indipendente dal menu "Azienda" in cui è collocato, cioè senza farlo eseguire dalla Libreria del Collage, e di conseguenza lo Sprix si comporterà diversamente, poiché avrà la variabile non valorizzata, cioè null (""). Notare il comando di condizione "IF".



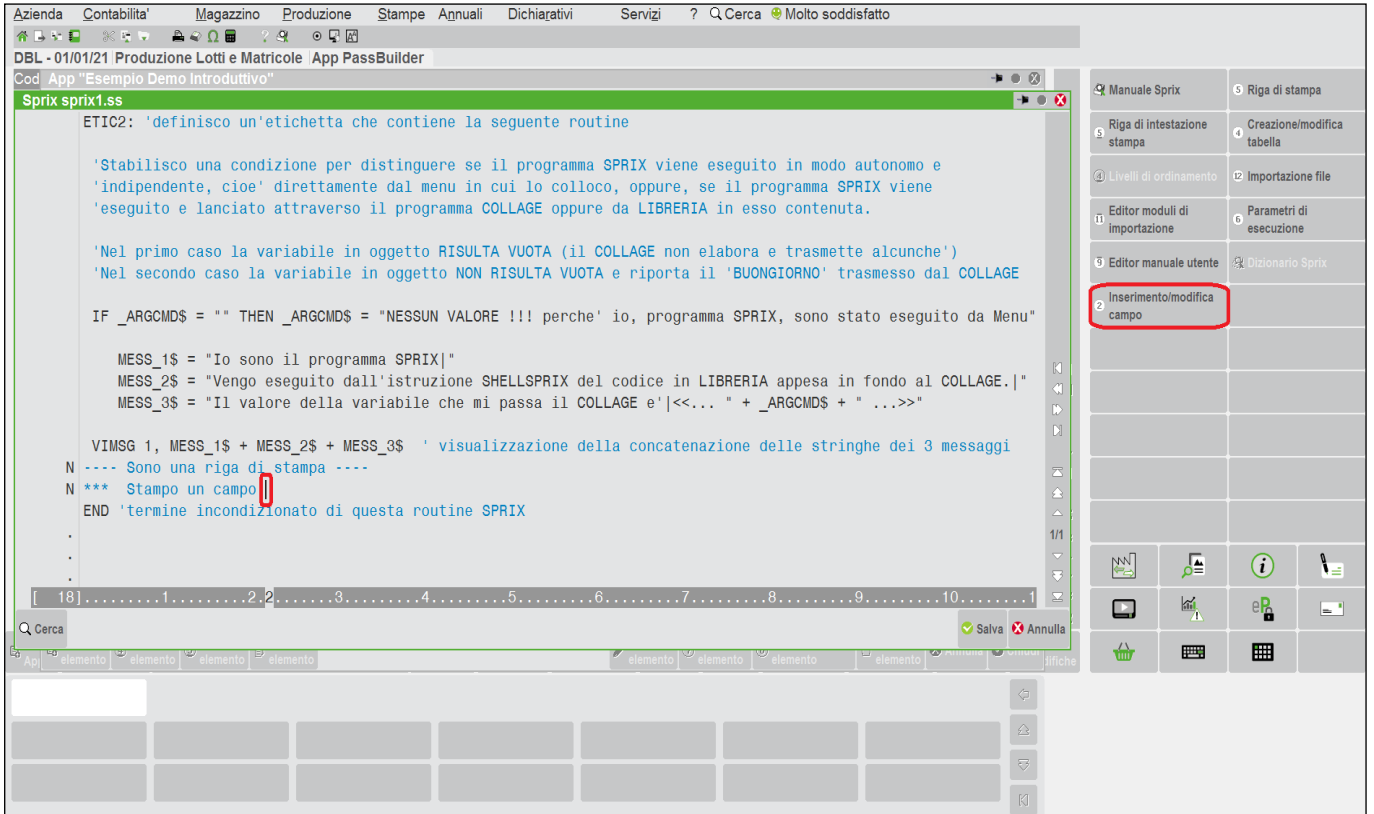
In aggiunta alla visualizzazione del messaggio, nello Sprix stabilisco anche l'esecuzione di una riga di stampa; l'Editor infatti è anche un motore di stampa con un proprio formato e caratteristiche di riga, contrassegnate a sinistra dell'ordinata cartesiana dell'Editor e distinte da quelle di codice. Anticipo, che esistono anche altri **comandi di stampa più avanzati**, utili per specifiche condizioni e **report più raffinati** in alternativa alle stampe "comprese" fornite dai ciclitabellari tramite l'editor.



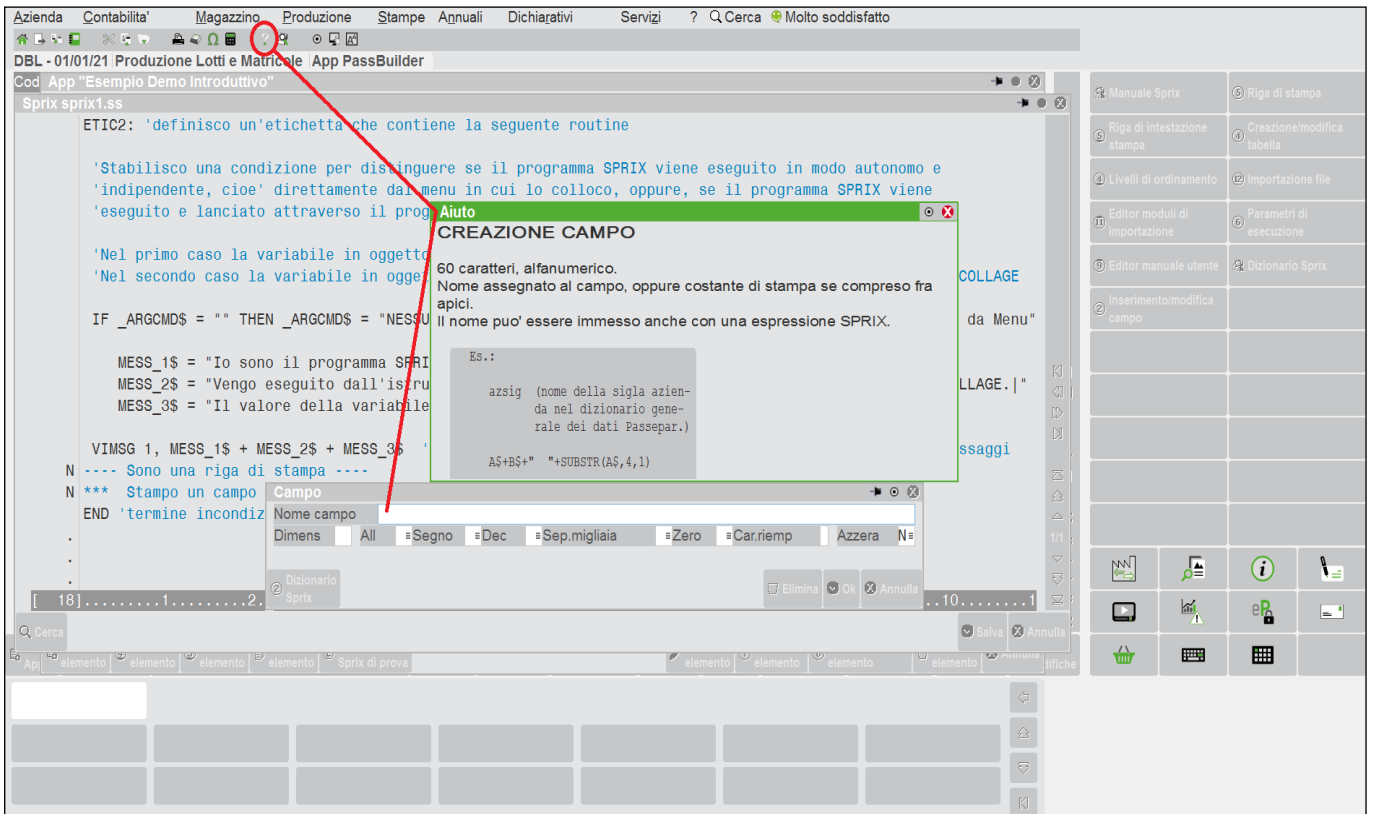
Inserisco la riga che verrà stampata: la faccio di tipo "N" che significa "Normale", per come la interpreta questo Editor capace di stampare.



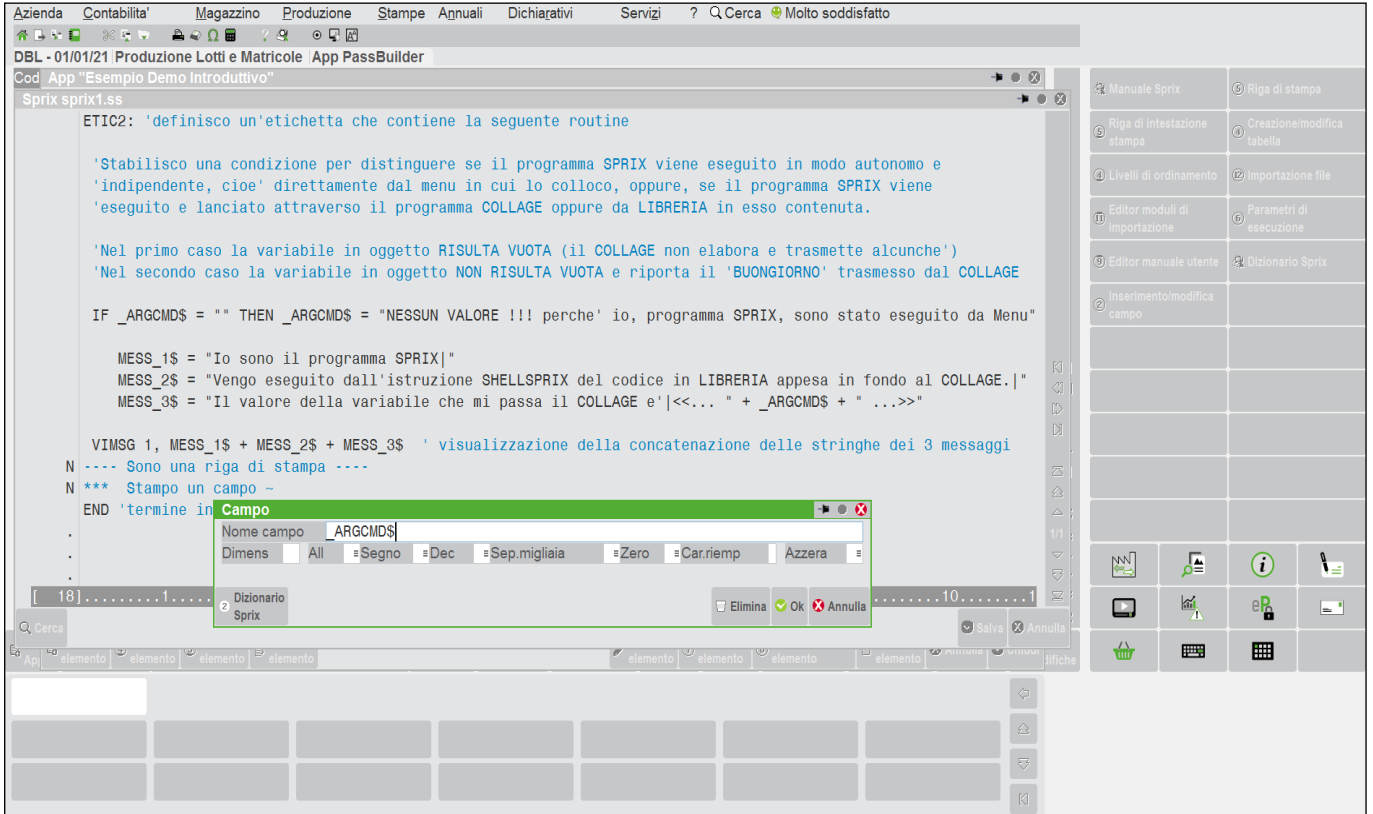
Premendo Invio aggiungo un'altra riga di stampa, in cui stampare un "Campo di stampa" con il valore della variabile `_ARGCMD$`.



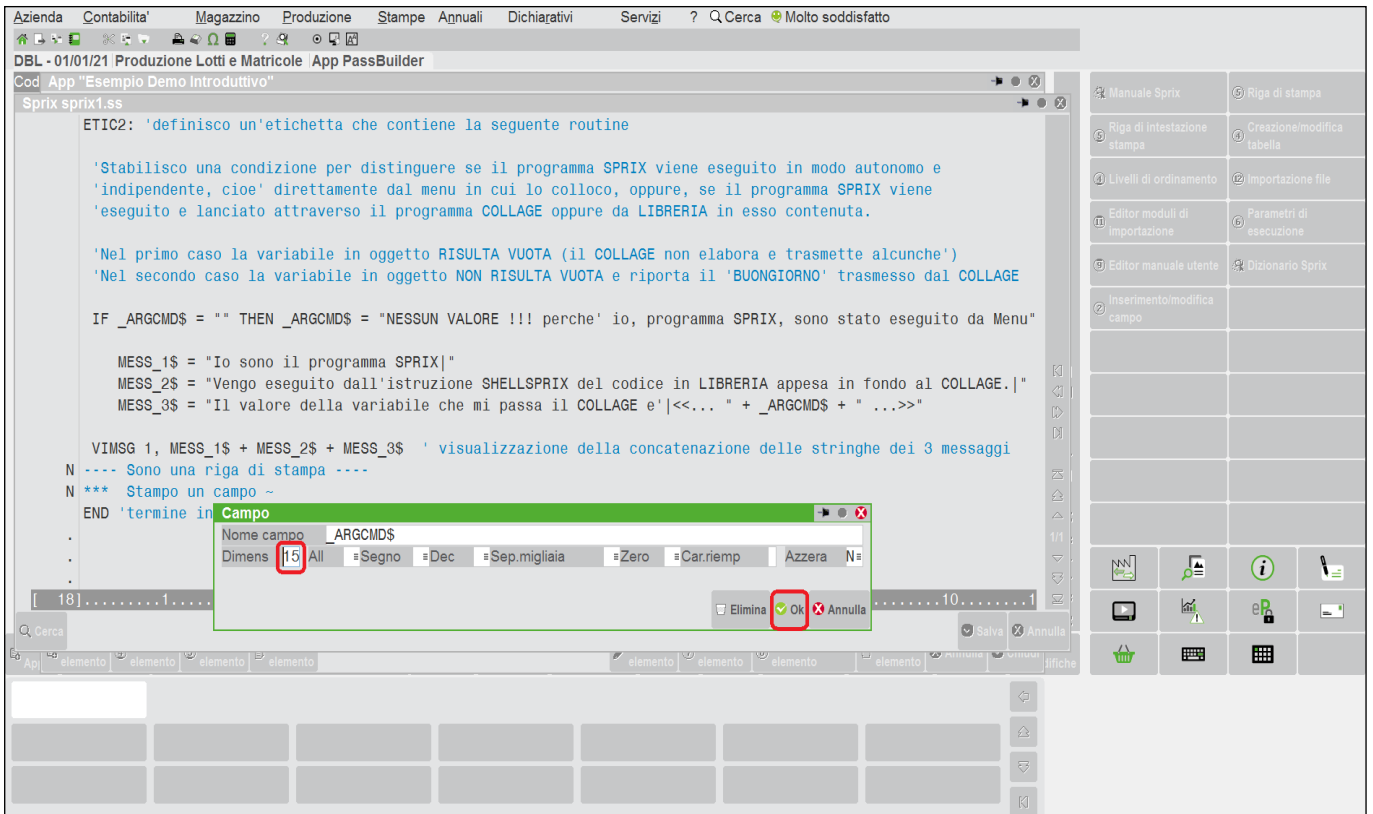
L'Editor visualizza una finestra con tutte le informazioni sulle caratteristiche del "Campo di stampa" che puoi facilmente intuire e stabilire. Se premi l'help [F1] su ciascun campo, ne otterrai il significato e la funzionalità; come nel Gestionale questo help di campo è presente anche in altri campi dell'Editor e relative funzionalità.



Adesso, inserisco la variabile `_ARGCMD$` e definisco la lunghezza del campo, in modo che l'intero contenuto sia stampato.



Alla conferma con "Ok", infatti, il campo è proposto lungo 15 caratteri, ma in realtà, data la lunghezza della stringa che potrei ottenere nella stampa, ovvero *"NESSUN VALORE !!! perché io, programma SPRIX, sono stato eseguito da Menu"*, avrò bisogno di almeno 60 caratteri.



Adeguo, pertanto, la lunghezza del campo a 60 caratteri.

The screenshot shows the SPRIX software interface. The main window is titled "Sprix sprix1.ss" and contains a code editor with the following text:

```
ETIC2: 'definisco un'etichetta che contiene la seguente routine

'Stabilisco una condizione per distinguere se il programma SPRIX viene eseguito in modo autonomo e
'indipendente, cioe' direttamente dal menu in cui lo colloco, oppure, se il programma SPRIX viene
'execuito e lanciato attraverso il programma COLLAGE oppure da LIBRERIA in esso contenuta.

'Nel primo caso la variabile in oggetto RISULTA VUOTA (il COLLAGE non elabora e trasmette alcunche')
'Nel secondo caso la variabile in oggetto NON RISULTA VUOTA e riporta il 'BUONGIORNO' trasmesso dal COLLAGE

IF _ARGCMD$ = "" THEN _ARGCMD$ = "NESSUN VALORE !!! perche' io, programma SPRIX, sono stato eseguito da Menu"

MESS_1$ = "Io sono il programma SPRIX|"
MESS_2$ = "Vengo eseguito dall'istruzione SHELLSPRIX del codice in LIBRERIA appesa in fondo al COLLAGE.|"
MESS_3$ = "Il valore della variabile che mi passa il COLLAGE e'|<<... " + _ARGCMD$ + " ...>>"

VIMSG 1, MESS_1$ + MESS_2$ + MESS_3$ ' visualizzazione della concatenazione delle stringhe dei 3 messaggi
N ---- Sono una riga di stampa ----
N *** Stampa un campo -
END 'termine in Campo
```

A dialog box titled "Campo" is open over the code editor. It contains the following fields and options:

- Nome campo: _ARGCMD\$
- Dimens: 60
- All:
- Segno:
- Dec:
- Sep.migliaia:
- Zero:
- Car.riemp:
- Azzerà:
- N:

The dialog box has three buttons at the bottom: "Elimina", "Ok", and "Annulla". The "Ok" button is highlighted with a red box.

Salvo il codice del programma.

The screenshot shows the SPRIX software interface. The main window is titled "Sprix sprix1.ss" and contains the same code editor as in the previous screenshot. The code is now complete with the following text:

```
ETIC2: 'definisco un'etichetta che contiene la seguente routine

'Stabilisco una condizione per distinguere se il programma SPRIX viene eseguito in modo autonomo e
'indipendente, cioe' direttamente dal menu in cui lo colloco, oppure, se il programma SPRIX viene
'execuito e lanciato attraverso il programma COLLAGE oppure da LIBRERIA in esso contenuta.

'Nel primo caso la variabile in oggetto RISULTA VUOTA (il COLLAGE non elabora e trasmette alcunche')
'Nel secondo caso la variabile in oggetto NON RISULTA VUOTA e riporta il 'BUONGIORNO' trasmesso dal COLLAGE

IF _ARGCMD$ = "" THEN _ARGCMD$ = "NESSUN VALORE !!! perche' io, programma SPRIX, sono stato eseguito da Menu"

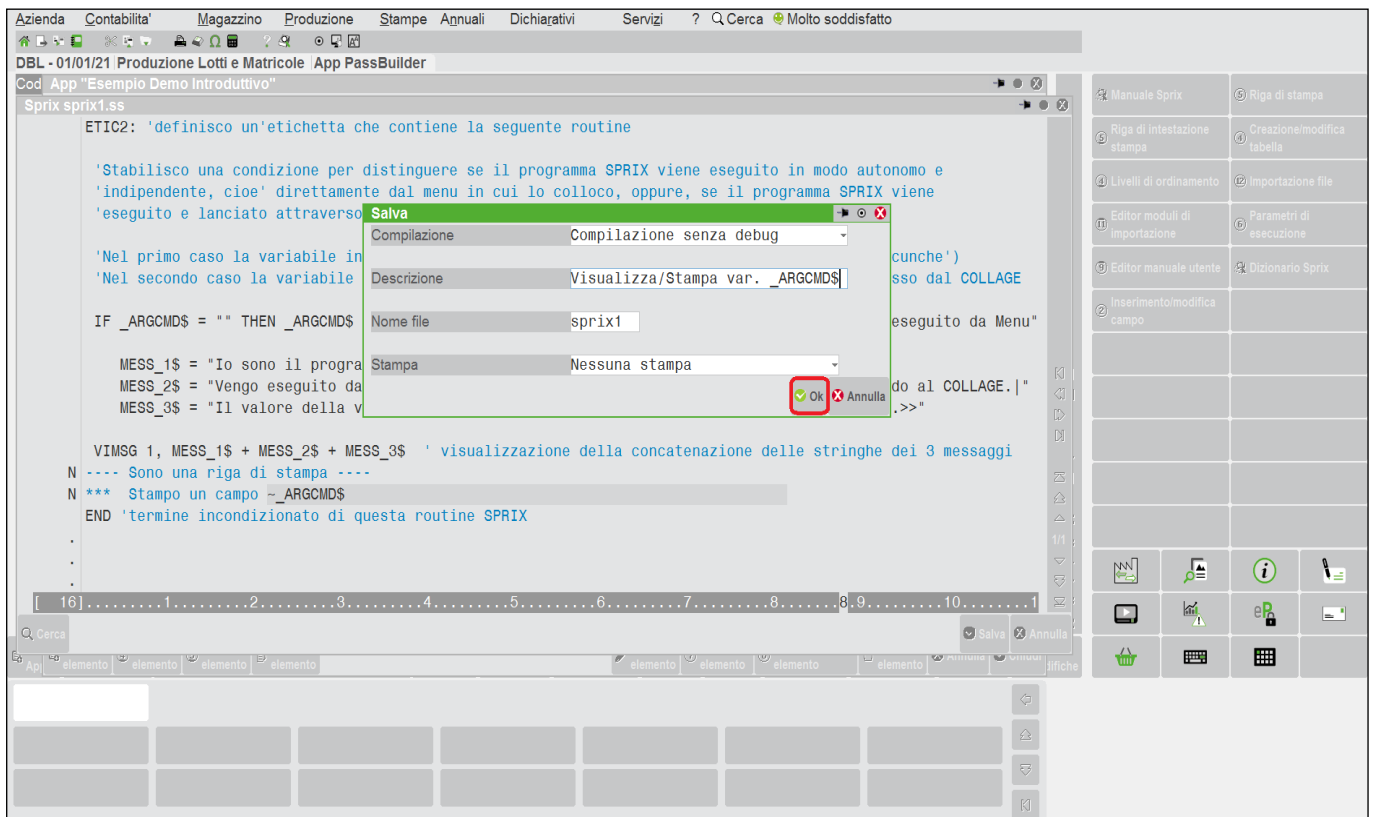
MESS_1$ = "Io sono il programma SPRIX|"
MESS_2$ = "Vengo eseguito dall'istruzione SHELLSPRIX del codice in LIBRERIA appesa in fondo al COLLAGE.|"
MESS_3$ = "Il valore della variabile che mi passa il COLLAGE e'|<<... " + _ARGCMD$ + " ...>>"

VIMSG 1, MESS_1$ + MESS_2$ + MESS_3$ ' visualizzazione della concatenazione delle stringhe dei 3 messaggi
N ---- Sono una riga di stampa ----
N *** Stampa un campo -_ARGCMD$
END 'termine incondizionato di questa routine SPRIX
```

The dialog box is no longer present. The "Salva" button in the bottom right corner of the code editor is highlighted with a red box.

Confermo la "consueta" maschera di compilazione (già vista anche in Collage), la quale visualizza le informazioni di default sulle caratteristiche identificative e di archiviazione del file sorgente. Inserisco la descrizione e non modifico altro, premo Ok per salvare definitivamente il sorgente (nome file con estensione ".ss") con la compilazione per generare anche il relativo eseguibile (nome file con estensione ".es").

Dalle immagini di questo esempio si noterà che i nomi file sono "sprix1.ss" e "sprix1.es". Lo stesso meccanismo vale per Collage, mentre la Libreria ha solamente il sorgente poichè non si compila. Passepartout ha un'automatica sintassi e numerazione dei programmi Sprix, Collage e Libreria, ma comunque è personalizzabile.



Come in Collage, rientrando nella composizione degli elementi dell'App ho la possibilità di configurare lo Sprix e stabilire in quale menu del Gestionale collocare il programma. In questo caso, scelgo di posizionarlo nella voce di menu "Azienda". Se consentito dall'ambiente, la presenza di una lettera maiuscola nel "Nome" Sprix, attiva i tasti di selezione rapida rapida (e sono sconsigliati i prefissi con caratteri speciali).

Azienda Contabilita' Magazzino Produzione Stampe Annuali Dichiarativi Servizi ? Q Cerca Molto soddisfatto

DBL - 01/01/21 | Produzione Lotti e Matricole | App PassBuilder

Cod App "Esempio Demo Introduttivo"

Nome-	Tipo	Nome file
Visualizza variabile per SPRIX	Collage	collage1.es - c
Visualizza/Stampa var. _ARGCMD\$	Sprix	spr1.es - spr

Sprix "Visualizza/Stampa var. _ARGCMD\$"

Nome Visualizza/Stampa var. _ARGCMD\$

Esecuzione da menu' Azienda

Posizione menu' Nessuno (0)

Azienda (1)

Contabilita' (2)

Magazzino (3)

Produzione (4)

Stampe (5)

Annuali (6)

Servizi (8)

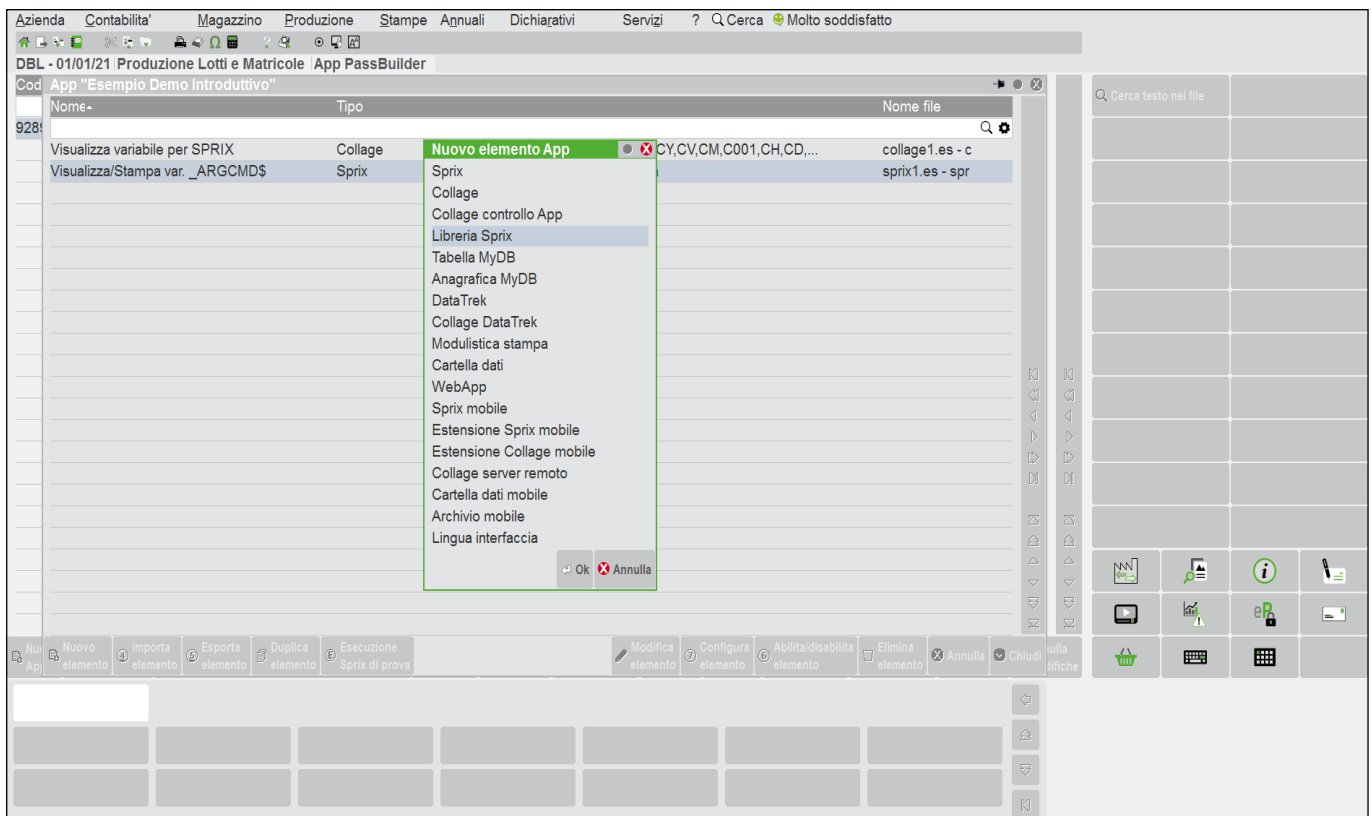
Ok Annulla

Cerca testo nei file

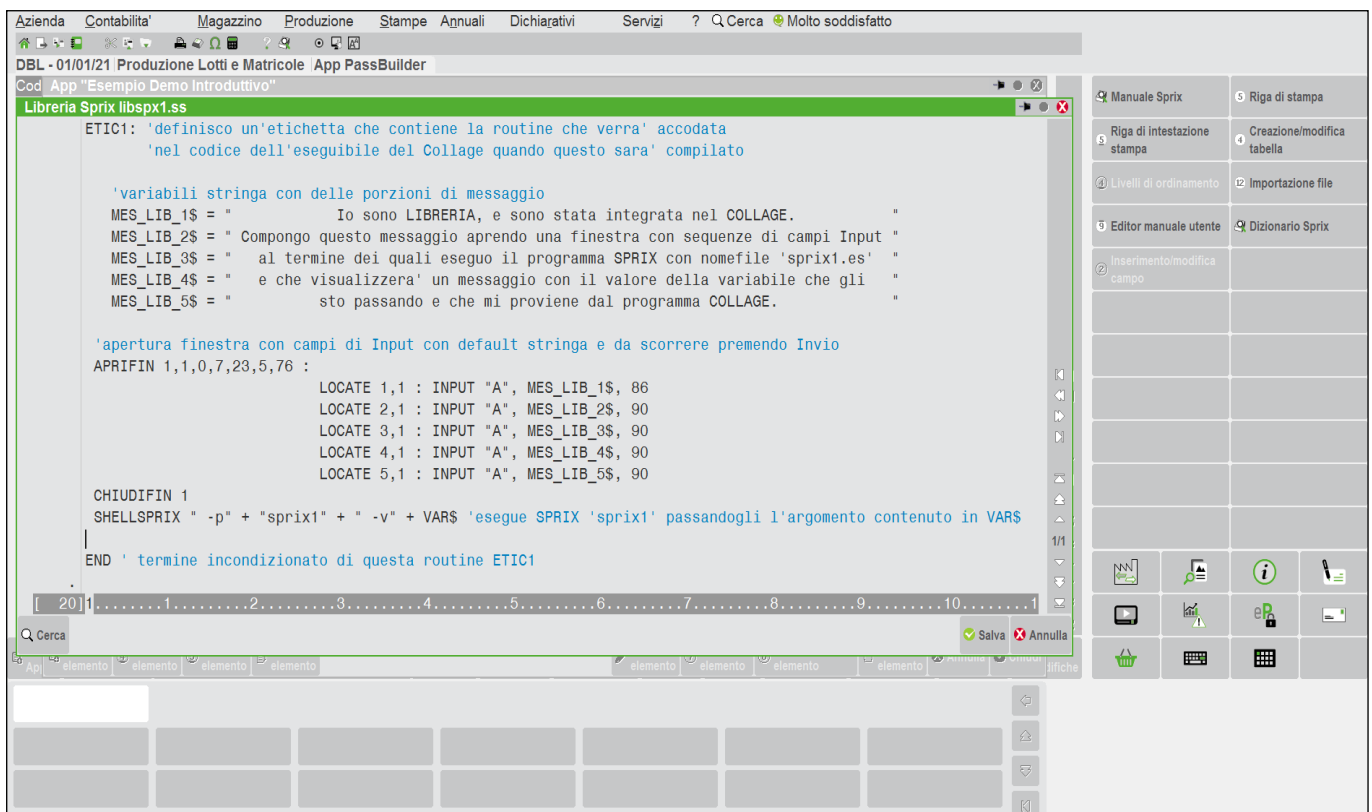
Esecuzione da menu'

<input checked="" type="checkbox"/> Nessuno	<input checked="" type="checkbox"/> Azienda	<input checked="" type="checkbox"/> Contabilita'	<input checked="" type="checkbox"/> Magazzino	<input checked="" type="checkbox"/> Produzione	<input checked="" type="checkbox"/> Stampe	<input checked="" type="checkbox"/> Annuali
<input checked="" type="checkbox"/> Servizi						

Adesso, manca la parte di codice della Libreria e che voglio inserire nel Collage, quindi lo devo creare. L'operatività si ripete, l'ambiente e le metodologie sono sempre le stesse, tutto è uniforme salvo alcuni dettagli che distinguono Sprix da Collage, e questi, da Libreria.



Traduzione del codice: definisco una routine che va dalla etichetta ETIC1 alla chiusura di routine con END; dentro questa routine costruisco il messaggio e con il comando di lancio Sprix SHELLSPRIX, tramite il parametro "-p" eseguo lo Sprix "sprix1" e con il parametro "-v" gli passo il contenuto della variabile VAR\$, cioè la stringa "BUONGIORNO" proveniente dal Collage che sarà assunto, poi, da _ARGCMD\$.



Quando salvo, il messaggio mi informa che non eseguirà la compilazione del formato poiché è solo una Libreria. I controlli di compilazione, infatti, avverranno quando inserirò questa Libreria compilando il Collage. Dimostro come....

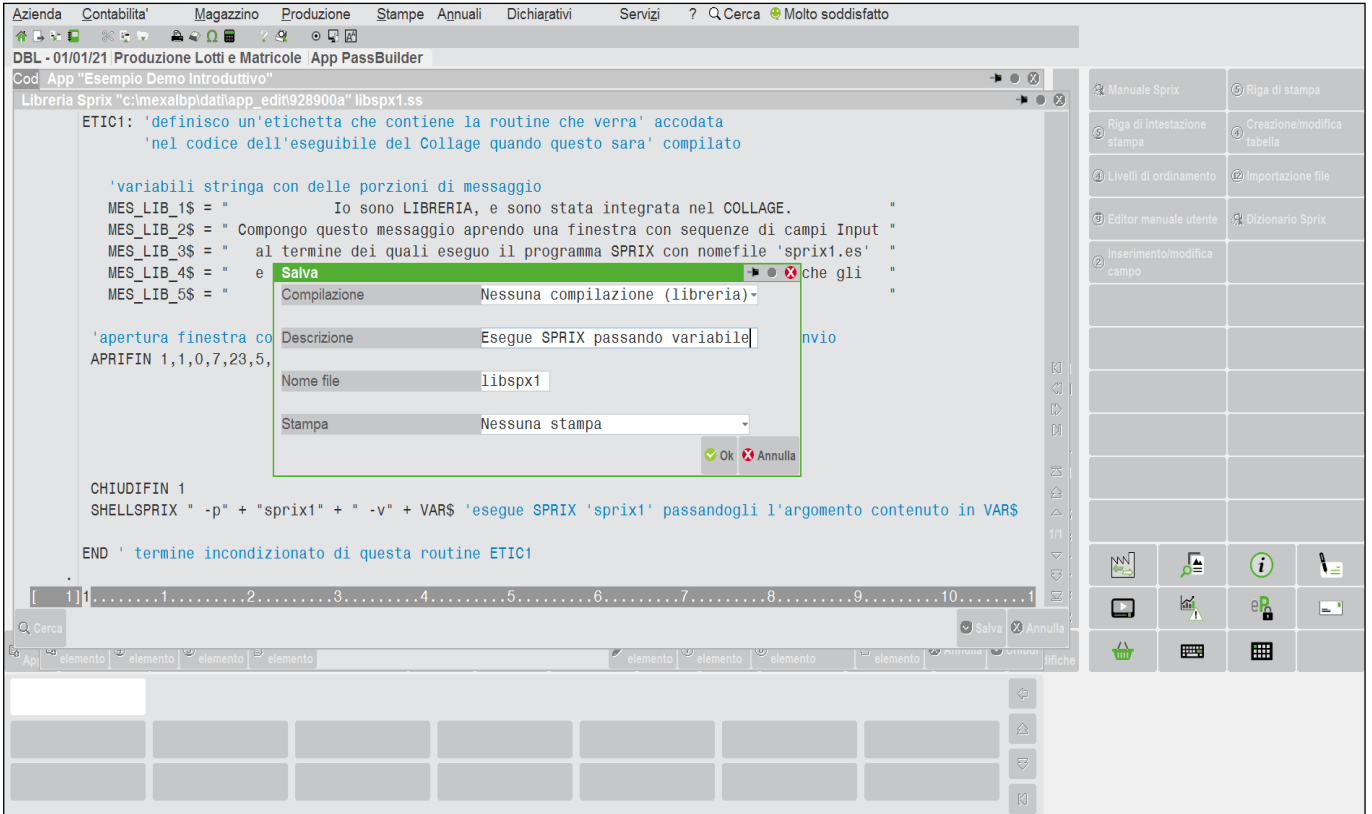
The screenshot shows the 'Libreria Sprix libspx1.ss' editor. A dialog box titled 'Salva' is open, allowing the user to configure the new library element. The 'Compilazione' dropdown menu is set to 'Nessuna compilazione (libreria)'. The 'Stampa' dropdown menu is also set to 'Nessuna stampa'. The 'Nome file' field contains 'libspx1'. The 'OK' button is highlighted with a red box, indicating the user's confirmation.

Dopo la conferma, Passepartout torna all'elenco degli elementi costituenti l'App, fra cui quella appena creata con il nome file "libspx1.ss". Se non indico alcuna descrizione, il default è il percorso in cui si trova la libreria, ma posso modificarlo salvando nuovamente il sorgente poiché non c'è la possibilità di potere configurare l'elemento.

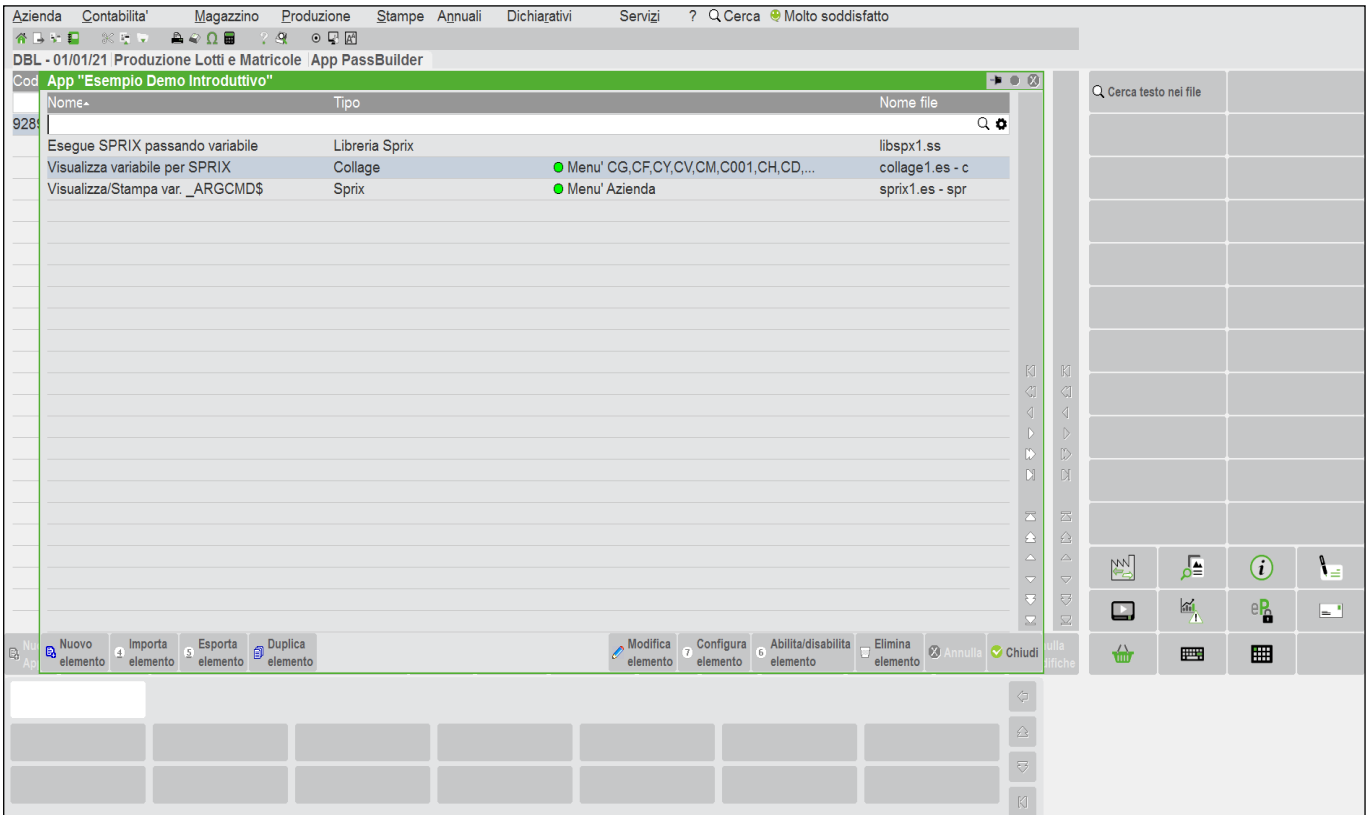
The screenshot shows the 'App "Esempio Demo Introduativo"' list of elements. The table below shows the elements in the application:

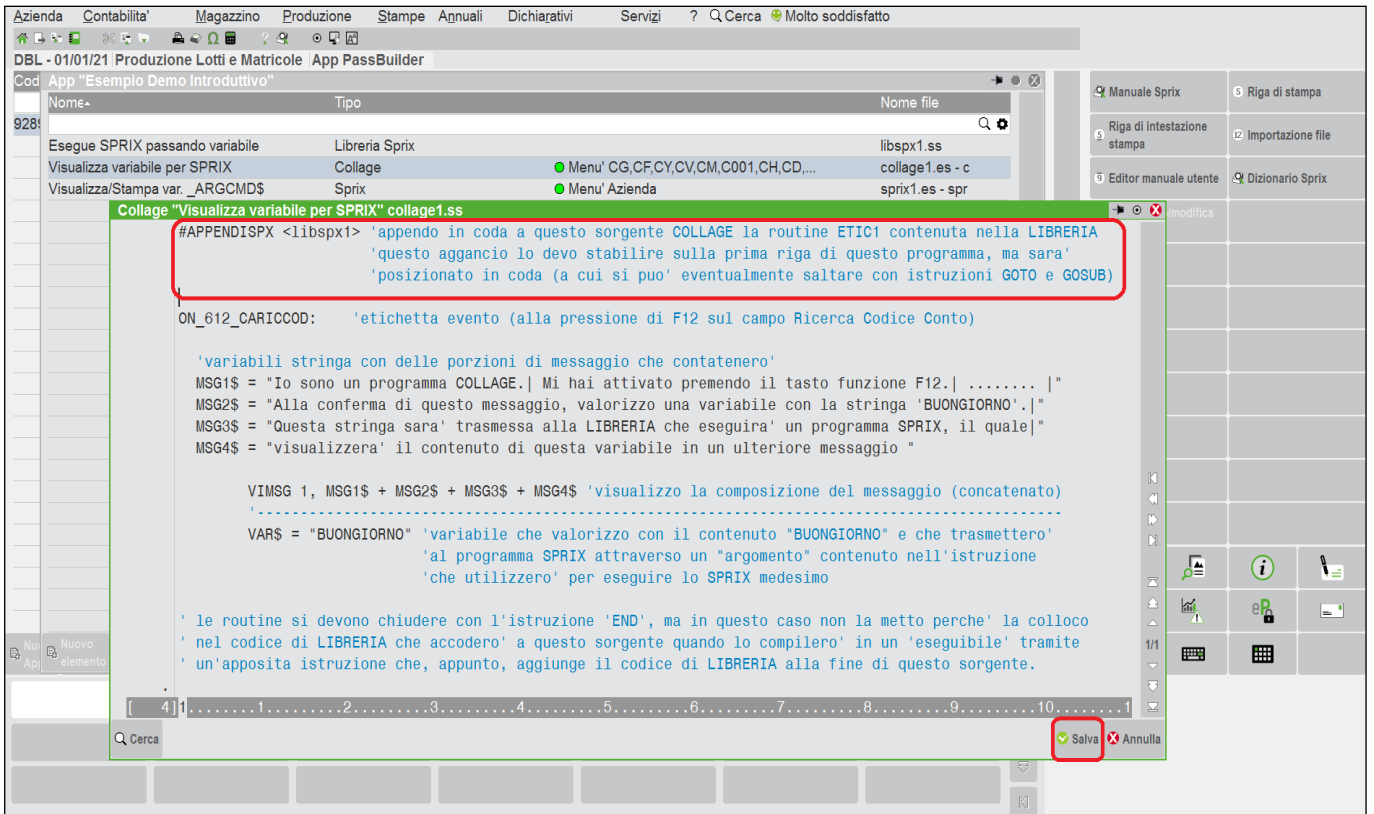
Nome	Tipo	Nome file
c:\mexal\p\dati\app_edit\928900a	Libreria Sprix	libspx1.ss
Visualizza variabile per SPRIX	Collage	collage1.es - c
Visualizza/Stampa var. _ARGCMD\$	Sprix	sprix1.es - spr

The path 'c:\mexal\p\dati\app_edit\928900a' and the file name 'libspx1.ss' are highlighted with red boxes in the original image.



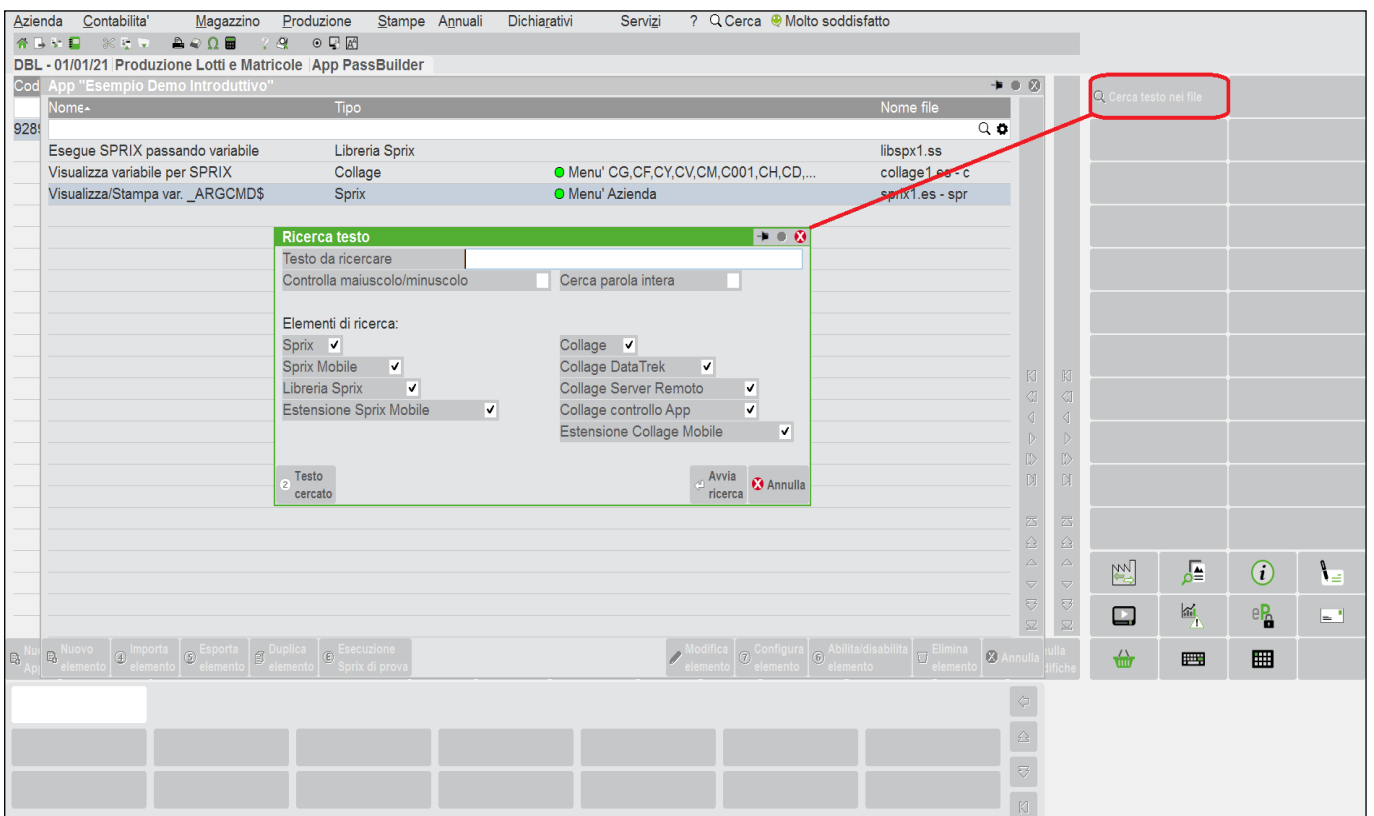
Adesso aggiungo questa Libreria dentro il Collage. Rientro dentro il Collage ed inserisco un apposito comando nella prima riga del codice, cioè l'istruzione che mi richiama Libreria "libsp1".



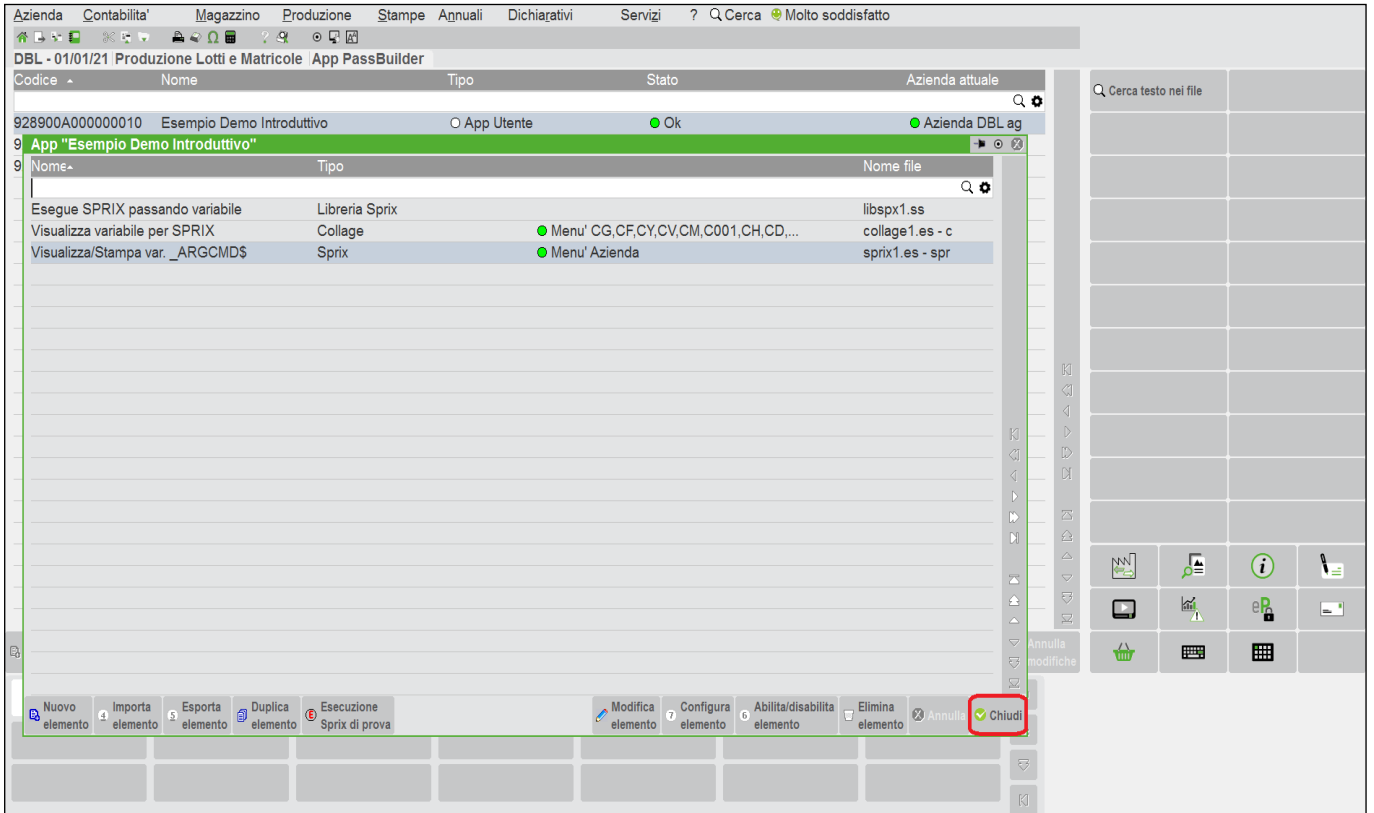


Infine salvo e compilo. La possibilità di richiamare routine di libreria, quindi, consente di creare sorgenti leggeri, sfruttando opportuni comandi GOSUB che richiamano routine di Libreria (riutilizzabili da vari programmi) le quali tornano a chiudere precedenti GOSUB tramite opportuni comandi RETURN.

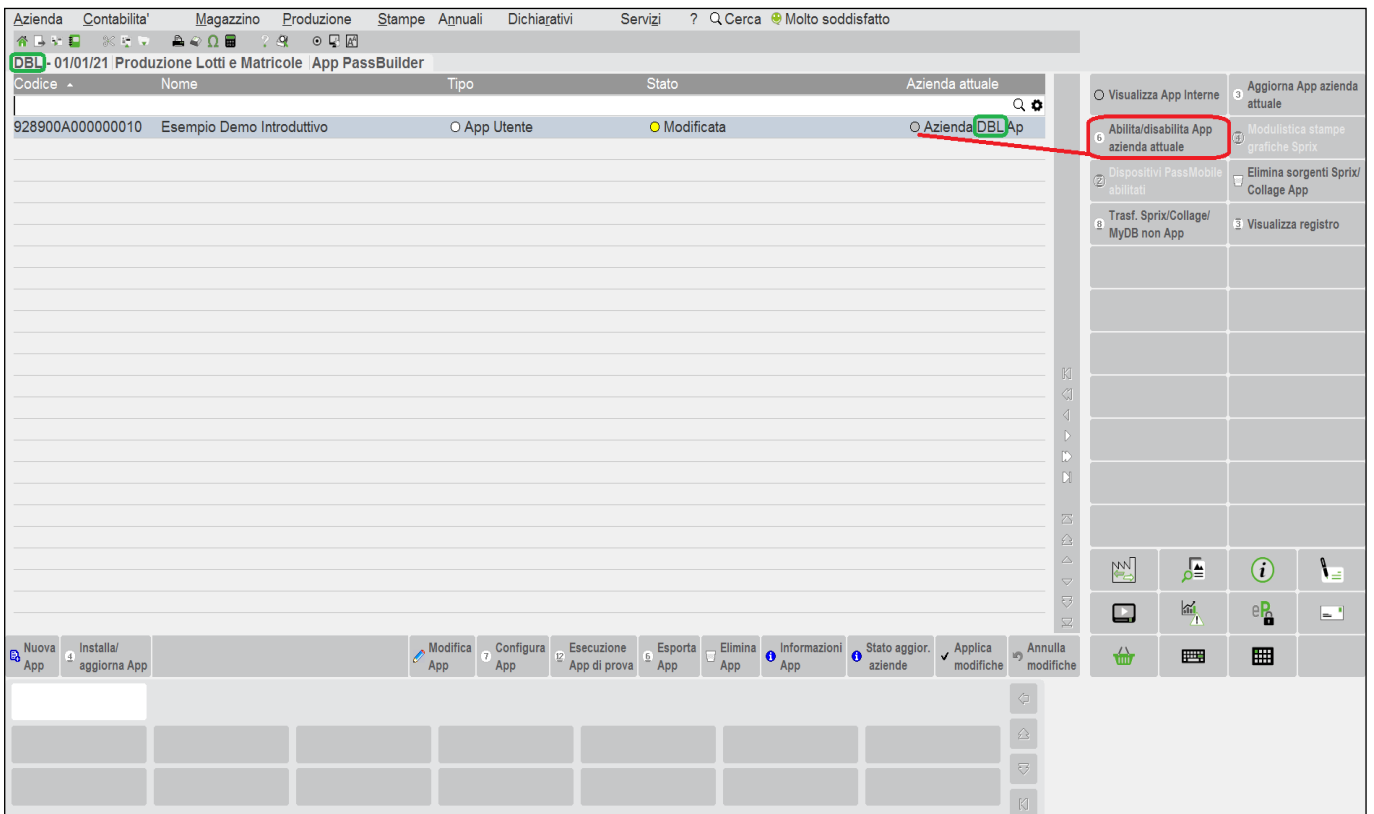
La compilazione controllerà la correttezza e la coerenza del codice dei sorgenti coinvolti per fonderli e generare un unico file eseguibile Collage, al termine del quale si trova la porzione di codice di Libreria. La compilazione, quindi, controlla che il codice di Libreria che sto aggiungendo nell'eseguibile Collage sia corretto e coerente all'intero codice. Per esempio se nel codice del Collage inserissi GOSUB ad un'etichetta di routine non esistente nella Libreria, il compilatore bloccherebbe la compilazione costringendo alla correzione (tranne nel salvataggio del sorgente senza alcuna compilazione). Comunque, sebbene il compilatore mi indichi le zone di errore nel codice, c'è una funzione che mi consente di cercare in vari modi nel testo dei vari codici sorgente.



Dopo il completamento e la compilazione dei programmi, chiudo l'elenco degli elementi dell'Applicazione.



Verifico che per l'azienda in uso l'App non sia disattivata, altrimenti mi basta abilitarla.



Applico tutte le modifiche in modo da cambiare lo stato dell'intera App (in "Ok") per poterla eseguire.

Nota: In presenza di una modifica all'interno di una App contenente un Collage, se questo è stato caricato (accesso da un menu) da un altro terminale, anche se l'evento non è in esecuzione, viene inibito l'allineamento "a caldo" con il pulsante applica modifiche.

The screenshot shows the application management interface. A table lists applications, with one row highlighted: **928900A000000010** | **Esempio Demo Introduttivo** | **App Utente** | **Modificata** | **Azienda DBL ag**. A confirmation dialog box titled "Applicazione modifiche" is displayed, asking: "Tutte le modifiche alla App 928900A000000010 'Esempio Demo Introduttivo' verranno applicate. Confermi operazione?". The dialog has "No" and "Sì" (Yes) buttons. A red arrow points from the "Sì" button to the "Applica modifiche" button in the bottom toolbar.

Posso controllare la "carta d'identità" dell'Applicazione.

The screenshot shows the application management interface with the "Informazioni App" dialog box open. The dialog displays the following information:

- Informazioni App**
 - Produttore/Codice: 928900A000000010
 - Tipo: App Utente - Autore di creazione "Installazione 928900"
- In uso**
 - Nome: Esempio Demo Introduttivo
 - Versione: 1.0.0
 - Versione minima Mexal/Passcom: 63.2
 - Livello minimo azienda: Nessuno
 - Autore ultima modifica: Installazione 928900 - Dealer 10993 - 21/03/2019 11:10:10
 - Stato: Ok
 - Azienda attuale: Azienda DBL aggiornata
- Modificata/In modifica**
 - Nome: Esempio Demo Introduttivo
 - Versione: 1.0.0
 - Versione minima Mexal/Passcom: 63.2
 - Livello minimo azienda: Nessuno
 - Autore ultima modifica: Installazione 928900 - Dealer 10993 - 21/03/2019 11:34:34
 - Stato: Ok

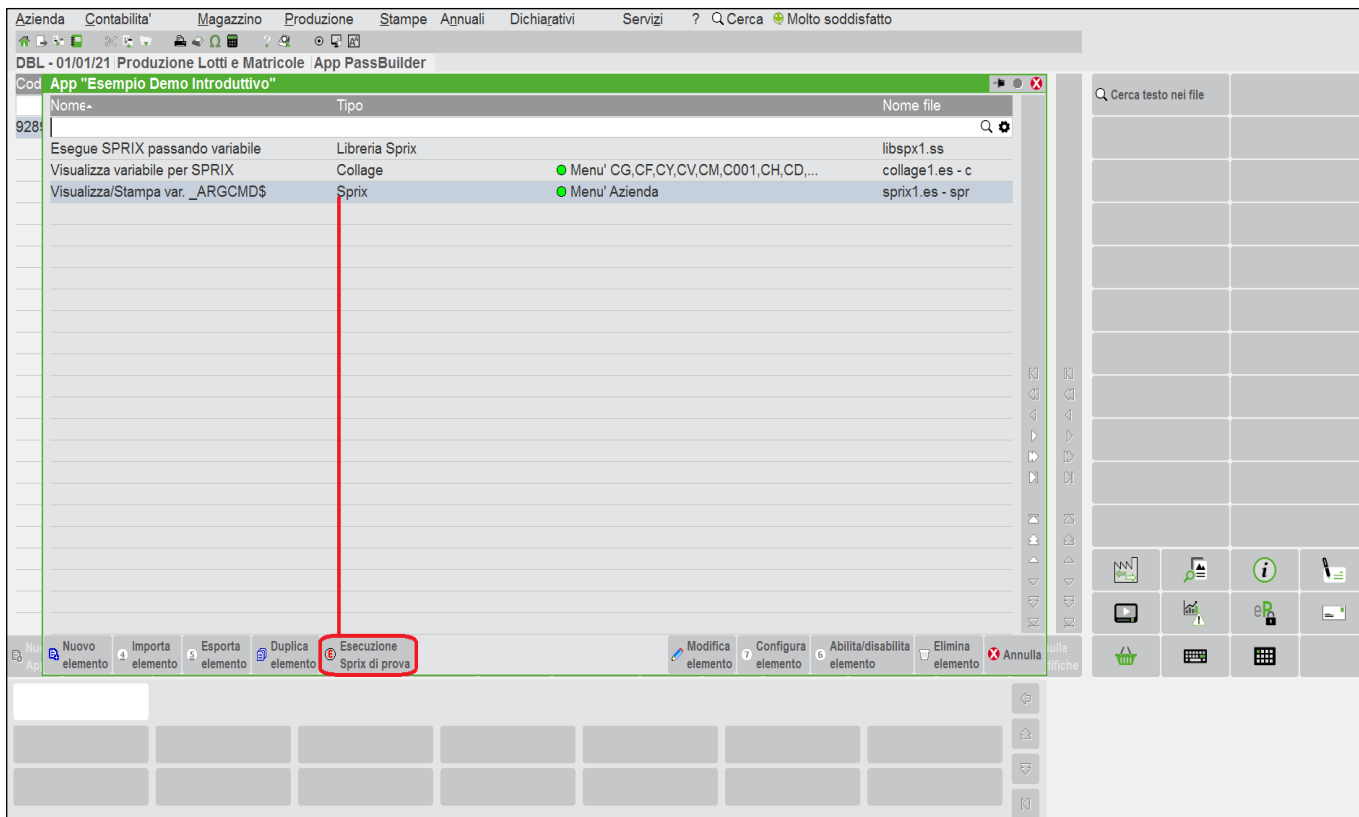
A red arrow points from the "Informazioni App" button in the bottom toolbar to the dialog box.

Se tutto è corretto, dallo stesso ambiente di sviluppo posso provare direttamente l'App grazie all'apertura di una sessione di prova e che posso poi chiudere. **Attenzione:** la "prova" significa che l'esecuzione avviene da PassBuilder e non dai moduli del Gestionale, pertanto la prova è a tutti gli effetti "reale" e l'applicazione agisce realmente. Se vuoi fare una "prova" in senso stretto, invece, utilizza archivi di prova su ambiente di prova.

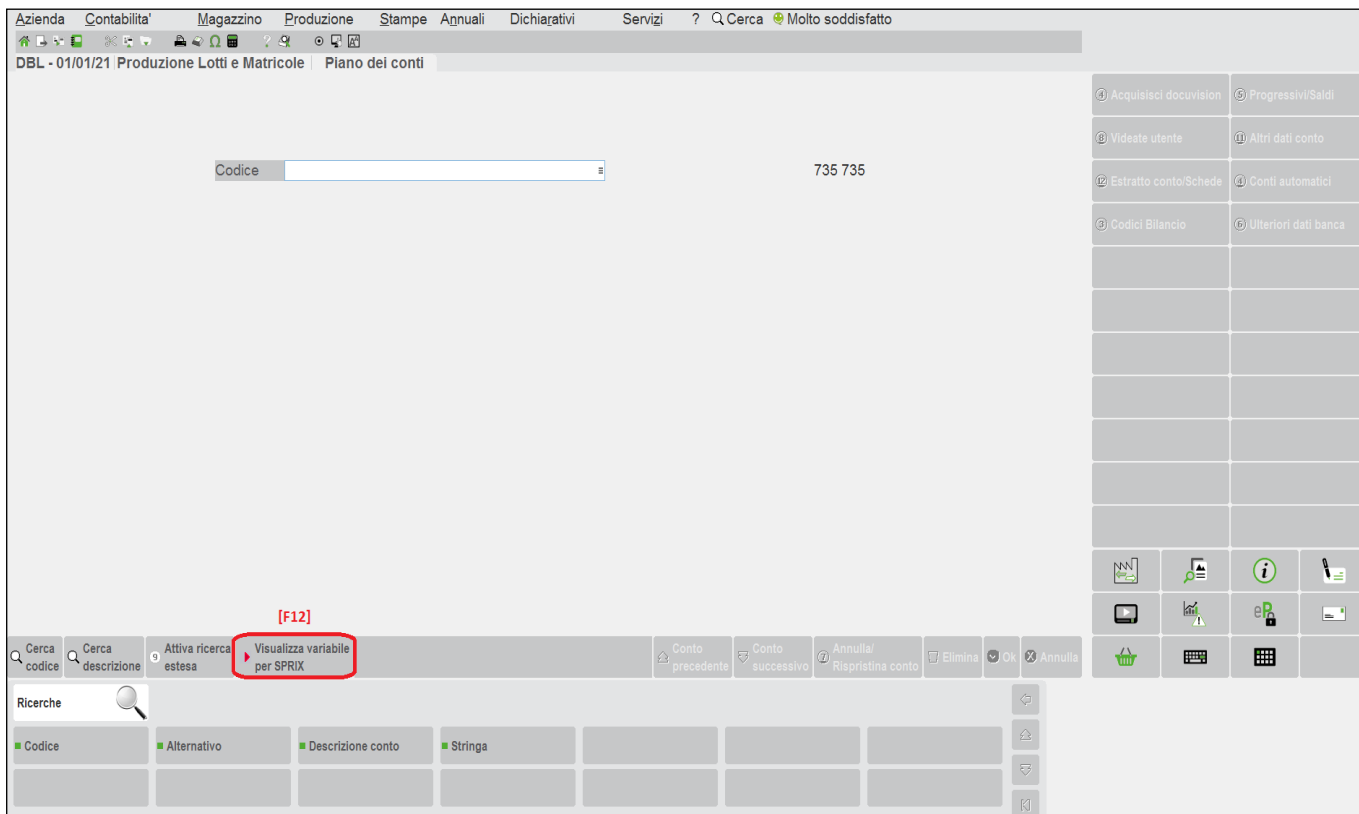
The screenshot shows the PassBuilder application interface. At the top, there is a navigation bar with tabs: Azienda, Contabilita', Magazzino, Produzione, Stampe, Annuali, Dichiarativi, Servizi, and a search bar. Below this is a header for 'DBL - 01/01/21 | Produzione Lotti e Matricole | App PassBuilder'. The main area contains a table with columns: Codice, Nome, Tipo, Stato, and Azienda attuale. The first row is highlighted and contains: 928900A000000010, Esempio Demo Introduttivo, App Utente, Ok (circled in red), and Azienda DBL ag (circled in red). To the right of the table is a sidebar with various options like 'Visualizza App Interne', 'Aggiorna App azienda attuale', etc. At the bottom, there is a toolbar with buttons: Nuova App, Installa/aggiorna App, Modifica App, Configura App, Esecuzione App di prova (circled in red), Esporta App, Elimina App, Informazioni App, Stato aggiorn. aziende, Applica modifiche, and Annulla modifiche.

The screenshot shows the PassBuilder application interface with a menu open. The menu items are: Apertura azienda, Anagrafica azienda, Chiusura azienda aperta, Parametri di base, AD - Analisi dati, Agenda (R), Clicca e Posta, Controllo di gestione, Docuision, Messaggi azienda, Cassetto Fiscale (Y), App (X) (circled in red), and Chiudi sessione di prova esecuzione App (F) (circled in red). The 'App (X)' item has a sub-menu with 'Esempio Demo Introduttivo' and 'Visualizza/Stampa var._ARGCMD\$'. The 'Chiudi sessione di prova esecuzione App (F)' item is also highlighted. The background shows the same interface as the previous screenshot, but with the menu open.

Se compilato correttamente, anche il singolo elemento Sprix è eseguibile direttamente in prova; questa possibilità non vale per Collage, poiché richiede sessione di prova.

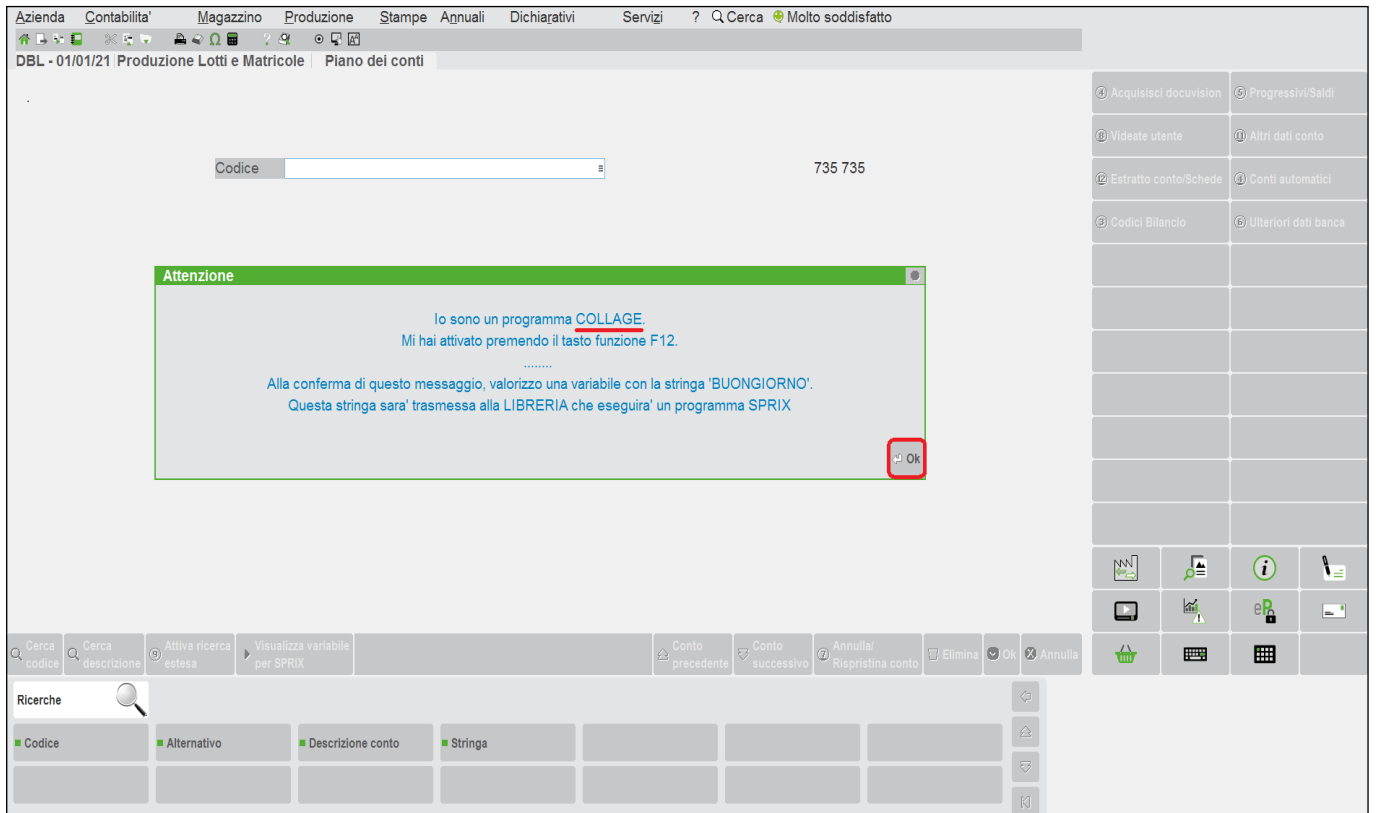


Dopo che ho salvato e compilato "il tutto" posso anche eseguire l'intera App in via ufficiale, cioè non in prova. Sul campo che ho scelto (Codice del piano dei conti in contabilità) premo F12 per vedere che cosa succede. Noto che in basso si è attivato un tasto chiamato "Collage". Si è attivato nuovo tasto che è [F12], relativo all'evento ON_612_CARICCOD che ho scelto, quindi lo clicco. **Notare:** nell'immagine il tasto F12 è disegnato in rosso per evidenziare, ma in realtà lo si vede tramite tooltip.



Di conseguenza, mi si avvia una serie di messaggi eseguiti dal comando VIMSG stabilito nei relativi codici sorgenti e che vengono eseguiti nel seguente ordine.

Eseguibile Collage:



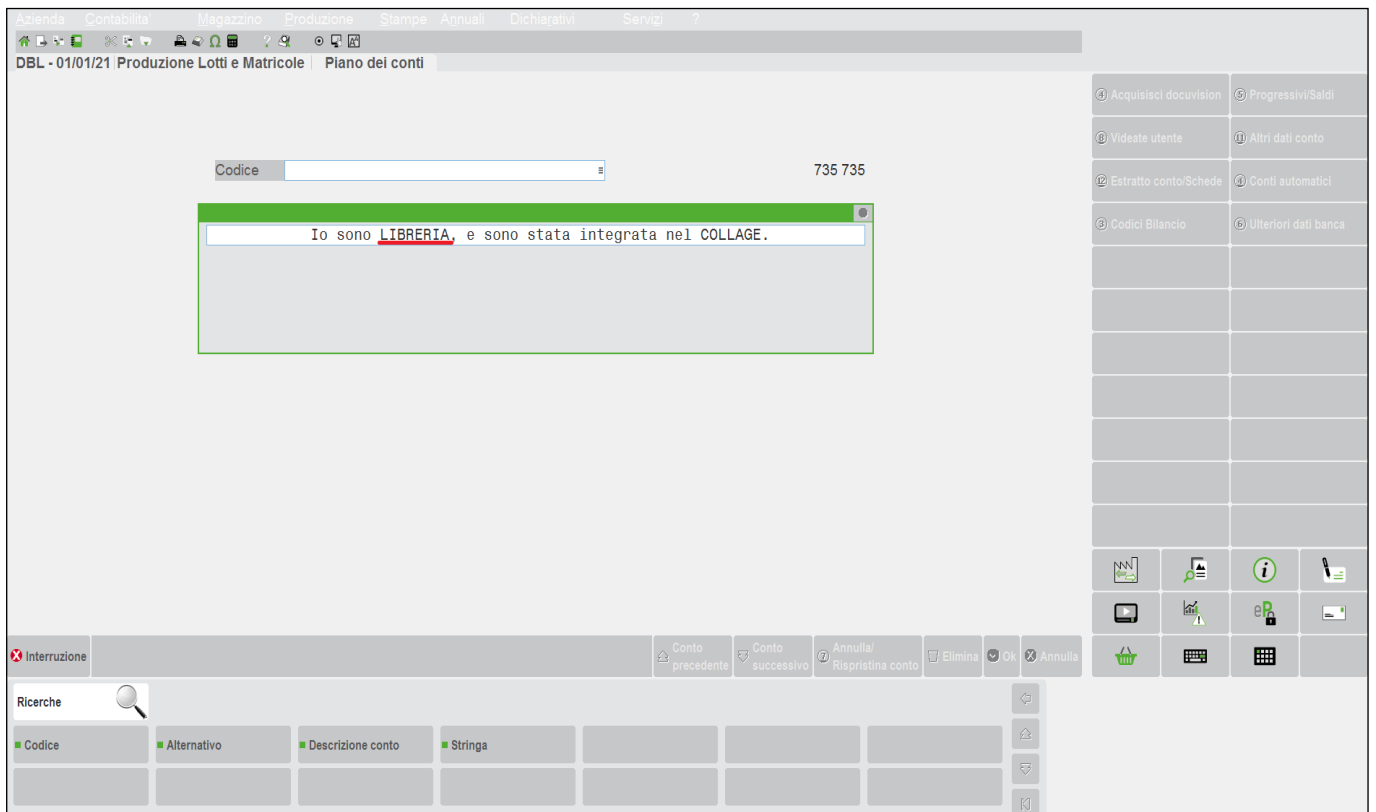
The screenshot shows the SPRIX software interface. At the top, there is a menu bar with options: Azienda, Contabilita', Magazzino, Produzione, Stampe, Annuali, Dichiarativi, Servizi, and a search icon. Below the menu bar, the current session is identified as 'DBL - 01/01/21 | Produzione Lotti e Matricole | Piano dei conti'. A search bar contains the code '735 735'. A central message box titled 'Attenzione' (Attention) contains the following text:

Io sono un programma COLLAGE
 Mi hai attivato premendo il tasto funzione F12.

 Alla conferma di questo messaggio, valorizzo una variabile con la stringa 'BUONGIORNO'.
 Questa stringa sara' trasmessa alla LIBRERIA che eseguirà un programma SPRIX

An 'Ok' button is highlighted with a red box in the bottom right corner of the message box. On the right side of the interface, there is a vertical toolbar with various icons for navigation and actions. At the bottom, there is a search area with a magnifying glass icon and a table with columns for 'Codice', 'Alternativo', 'Descrizione conto', and 'Stringa'.

Eseguibile Collage, ma proveniente da Libreria acquisita e registrata dentro il Collage:



This screenshot shows the same SPRIX software interface as above. The central message box now displays the text:

Io sono LIBRERIA, e sono stata integrata nel COLLAGE.

The 'Ok' button is no longer visible. The rest of the interface, including the menu bar, search bar, and bottom toolbar, remains the same as in the previous screenshot.

The screenshot shows the SPRIX interface with a message box containing the following text:

```

Io sono LIBRERIA, e sono stata integrata nel COLLAGE.
Compongo questo messaggio aprendo una finestra con sequenze di campi Input
al termine dei quali eseguo il programma SPRIX con nomefile 'sprix1.es'
e che visualizzera' un messaggio con il valore della variabile che gli
sto passando e che mi proviene dal programma COLLAGE.
    
```

A red arrow points from the text to a keyboard layout at the bottom of the screen, with the following annotation:

Ad ogni Invio si apre ciascun successivo campo di Input con il relativo contenuto default

The interface also shows a top menu bar with options like 'Azienda', 'Contabilità', 'Magazzino', 'Produzione', 'Stampe', 'Annuali', 'Dichiarativi', and 'Servizi'. A right-hand sidebar contains various icons for account management. At the bottom, there is a search bar and a grid of buttons for account selection.

Viene lanciato l'eseguibile Sprix:

Noto che sta valorizzando il valore della variabile con "BUONGIORNO". Quindi tutto funziona.

The screenshot shows the SPRIX interface with a confirmation message box titled 'Attenzione' containing the following text:

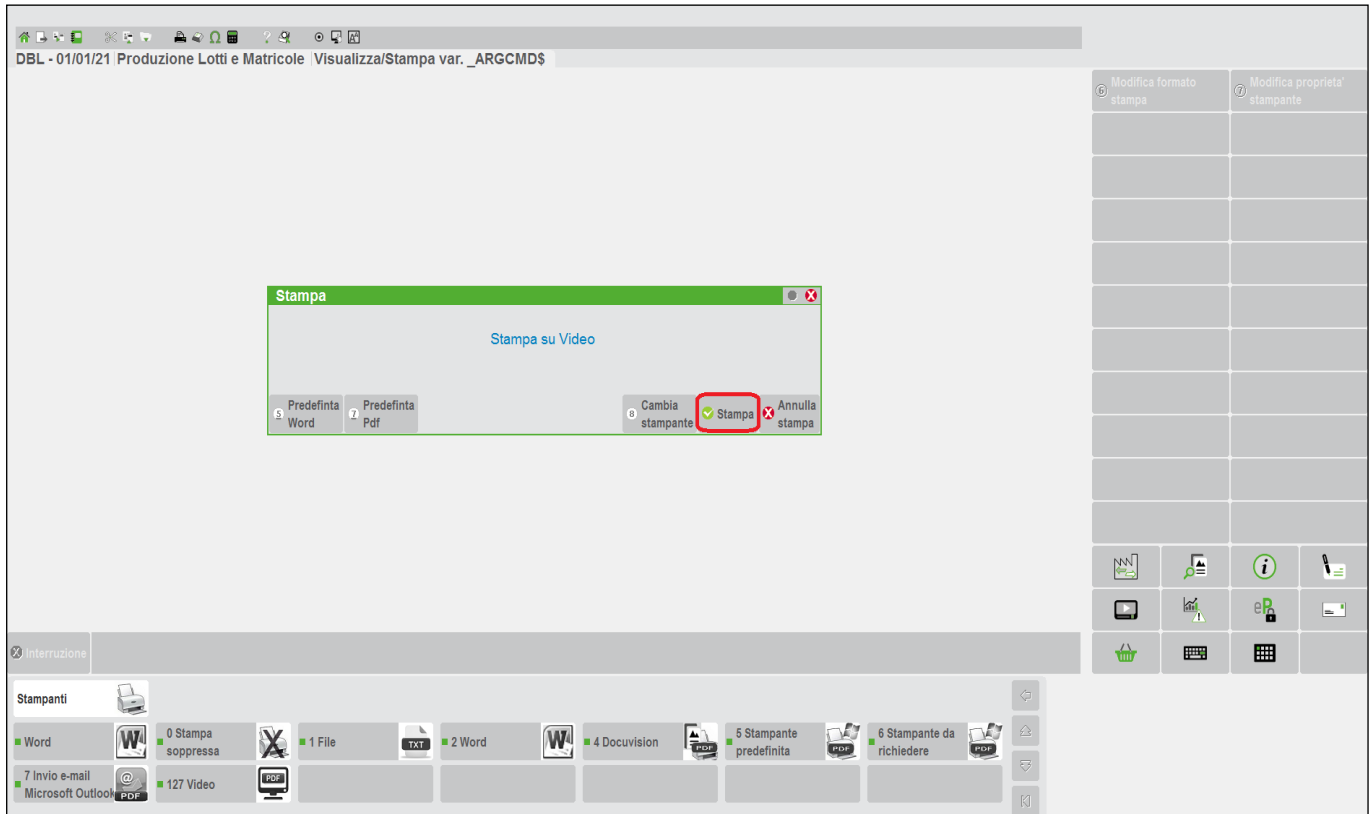
```

Io sono il programma SPRIX
Vengo eseguito dall'istruzione SHELLSPRIX del codice in LIBRERIA appesa in fondo al COLLAGE.
Il valore della variabile che mi passa il COLLAGE e'
<<... BUONGIORNO ...>>
    
```

An 'Ok' button is highlighted with a red box in the bottom right corner of the message box.

The interface also shows the same top menu bar and right-hand sidebar as in the previous screenshot. At the bottom, there is a search bar and a grid of buttons for account selection.

Nello Sprix, però, avevo aggiunto anche l'esecuzione di una stampa. Verifico com'è.

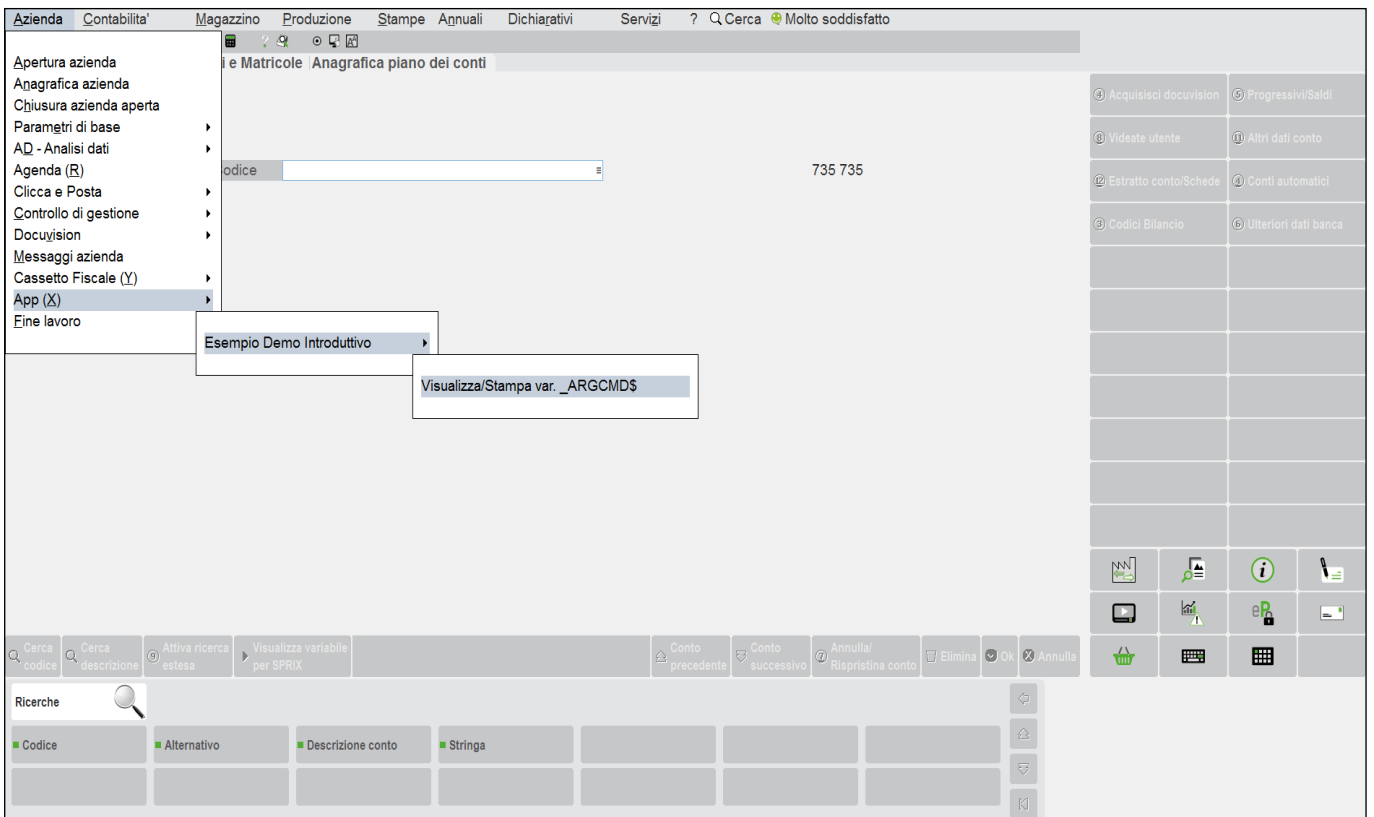


Anche qui, noto che il campo è correttamente valorizzato. La stampa rispecchia il formato dell'Editor, ma nonostante che io abbia stabilito una dimensione del font Normale, la stampa è **compressa** in quanto è predisposta per stampare sempre **220 caratteri**; l'utilizzo di una stampante di tipo "W", infatti, considera tutti i 220 caratteri e per impostarli nelle stampe Sprix il driver usa un font ridotto.

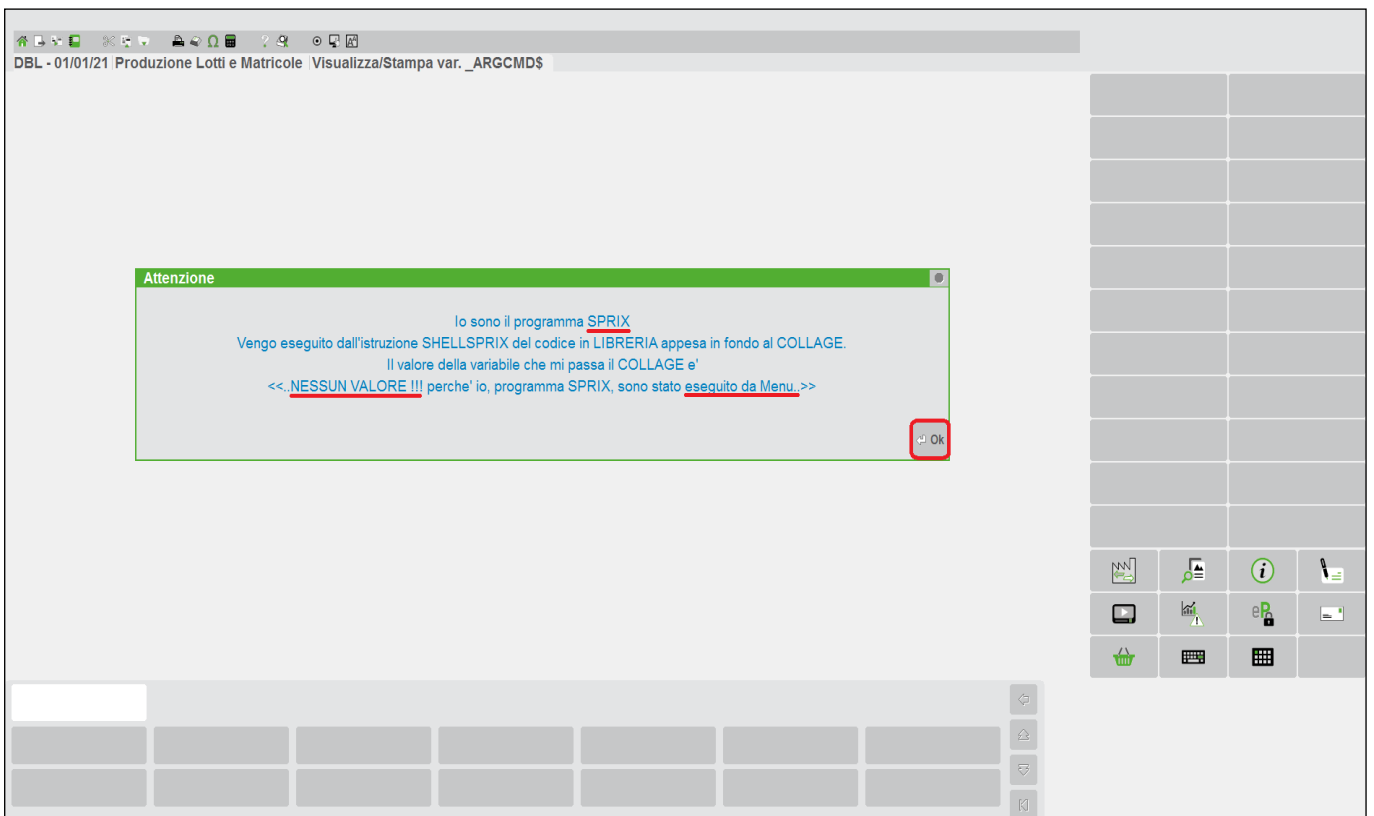
Attenzione: l'immagine seguente è **molto ingrandita**, rappresenta la porzione superiore sinistra di un foglio A4, poiché l'intera stampa avrebbe caratteri piccoli, pertanto, se per la variabile avessi predisposto e riempito un campo di stampa di quasi 200 caratteri avrei scritto fino alla fine del formato. Se volessi sfruttare questo metodo di stampa basato sull'Editor, per avere una stampa non compressa dovrei configurare una stampante nel menu Servizi con relativo tipo di font e dimensioni, oppure dovrei usare comandi di stampa parametrizzati in base al formato del report che vorrei ottenere; questa seconda modalità è molto utile nei ciclitabellari, in quanto la stampa eseguita con l'editor risulterebbe limitata, mentre invece caricando in memoria i dati elaborati dal ciclo per poi eseguirne delle stampe più "avanzate" con appositi comandi di stampa (ad esempio "APRISTAMP") e relativi moduli grafici personalizzati.



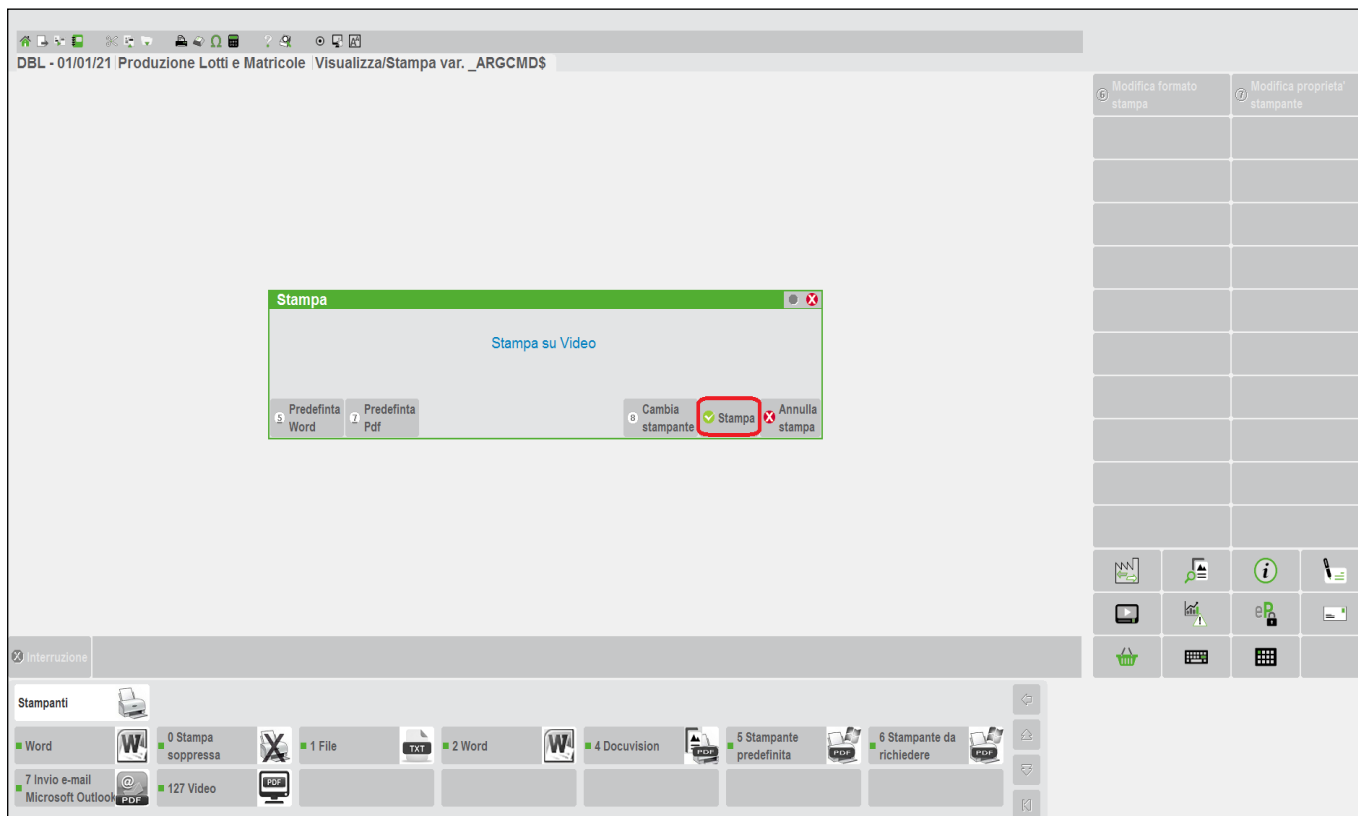
In precedenza, ho accennato al fatto che lo Sprix può essere anche eseguito separatamente, in modo indipendente, attraverso la voce di menu in cui lo ho inserito. Lo eseguo e vedo che cosa succede.



Anche se le prime tre righe non sono più veritiere (perché il codice dello Sprix prevede di stamparle comunque), noto che il contenuto della variabile è cambiato.



Verifico in stampa che cosa succede.



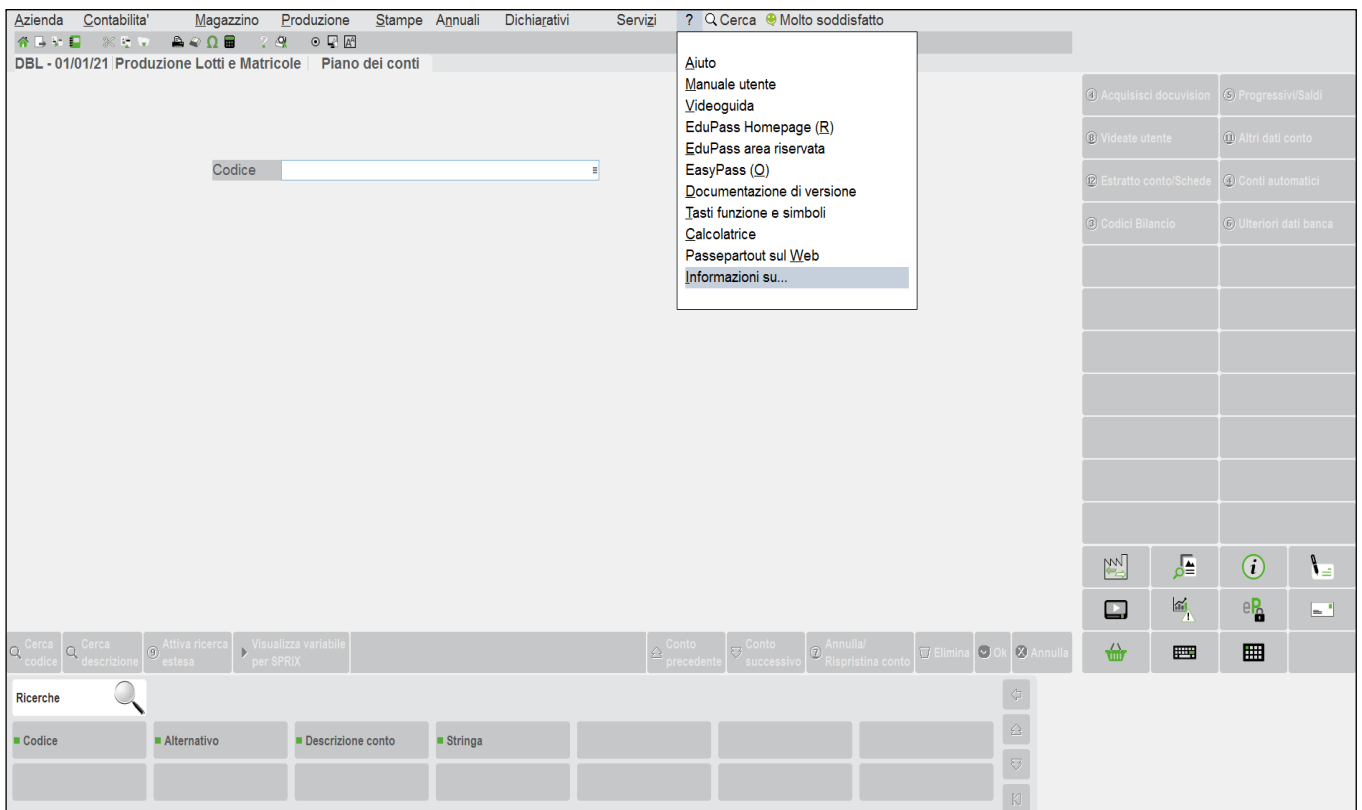
Anche il contenuto della stampa è cambiato.



Questo esempio è terminato. Se vuoi, puoi esercitarti con questo simpatico esempio ([clicca qui](#)) relativo al comando **PRINTICONA**, cioè una funzione che trovi nel gruppo di comandi per gestire il video: è uno Sprix che puoi copiare e incollare ed eseguire in uno qualsiasi dei menu (se perdi l'indentazione del codice, nessun problema, il programma funziona lo stesso). *Ti ricordo che nel manuale ci sono esempi che possono avere font e formattazioni differenti: il "copia e incolla" del testo potrebbe interpretare e riportare dei caratteri speciali che potrebbero "sporcare" il sorgente ed impedirne la compilazione e/o il funzionamento (per esempio i doppi apici di taluni font non vengono riconosciuti e si "tramutano" in altro); in tali casi è opportuno fare transitare il codice copiato in un editor ASCII puro ed eventualmente apportare delle modifiche massive a tali caratteri in modo da uniformarli a quanto richiesto dall'Editor Sprix.*

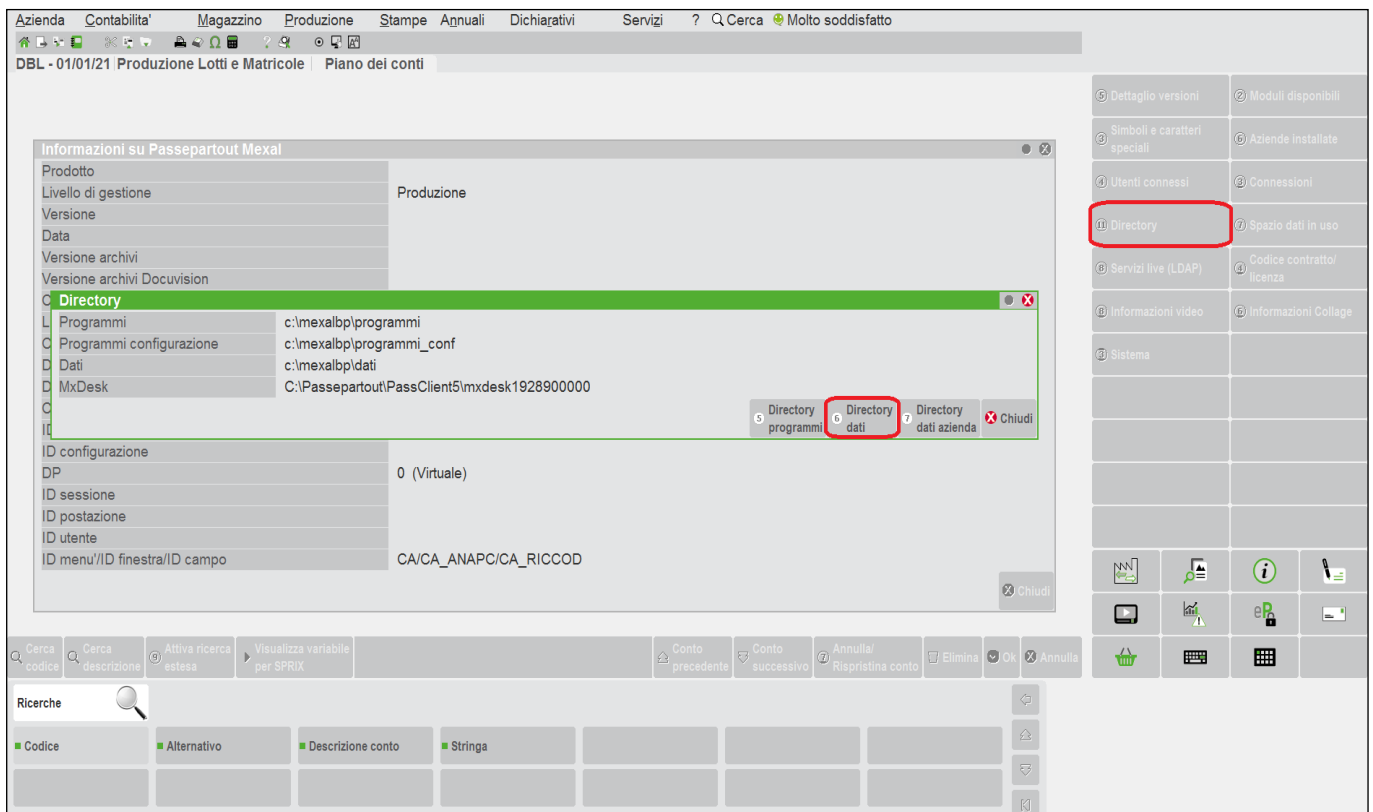
Questo esempio dei messaggi e delle stampe ha dimostrato che ci può essere una relazione fra i due componenti d'ambiente Sprix e Collage. Come accennato in precedenza, analogamente potrei sviluppare uno Sprix che sfrutta una Libreria oppure uno Sprix che esegue un altro Sprix, sempre tramite SHELLSPRIX, ma sarebbe ripetitivo.

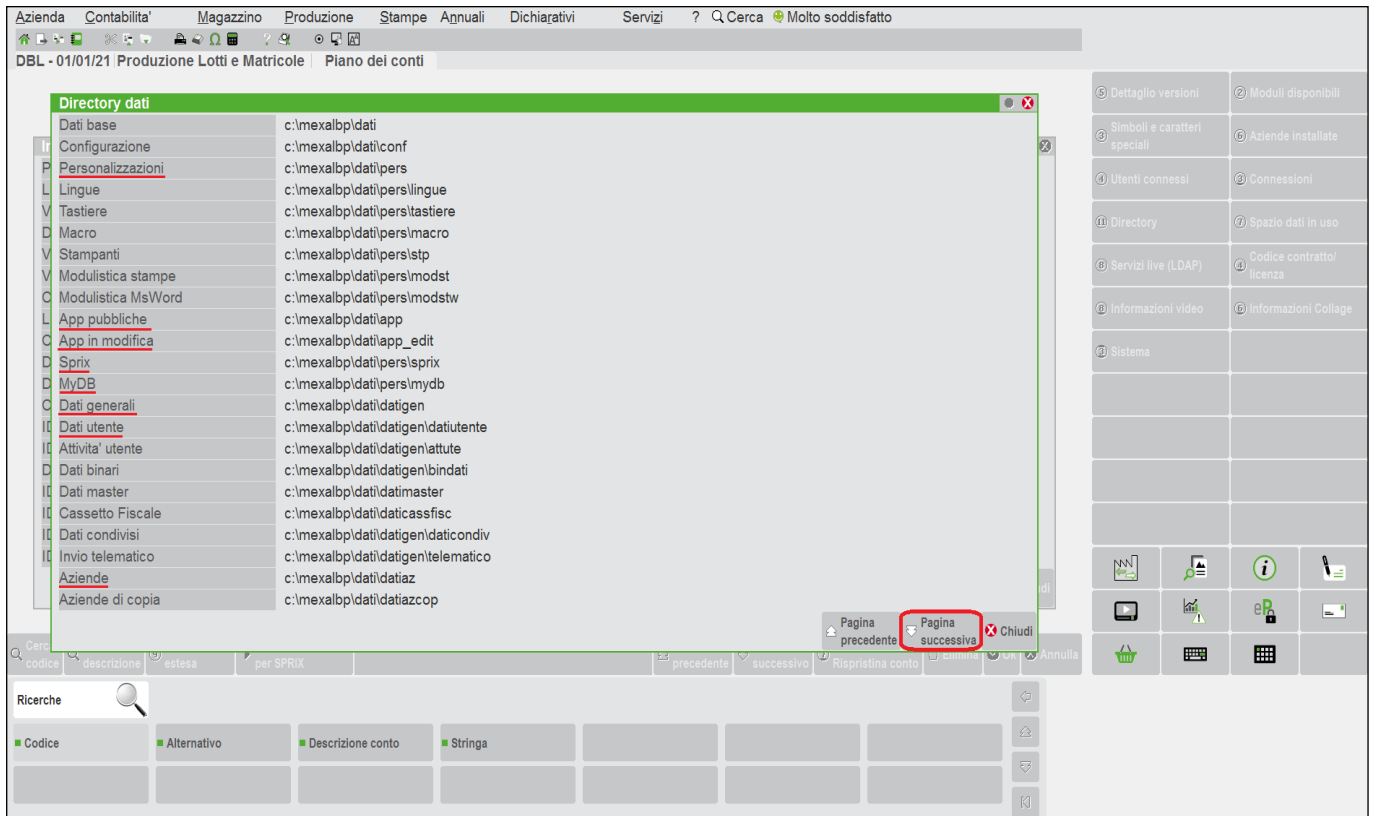
Dopo che ho sviluppato un esempio sui miei tre sorgenti (Collage, Libreria, Sprix) verifico dove risiedono tutti i file che riguardano l'Applicazione, ovvero da quelli dei programmi a quelli dei dati e file personali. La voce di menu "informazioni su" consente di verificare i percorsi disponibili:



Selezione le Directory per verificare i percorsi dell'installazione, dai **dati generali** a quelli specifici dell'**azienda**.

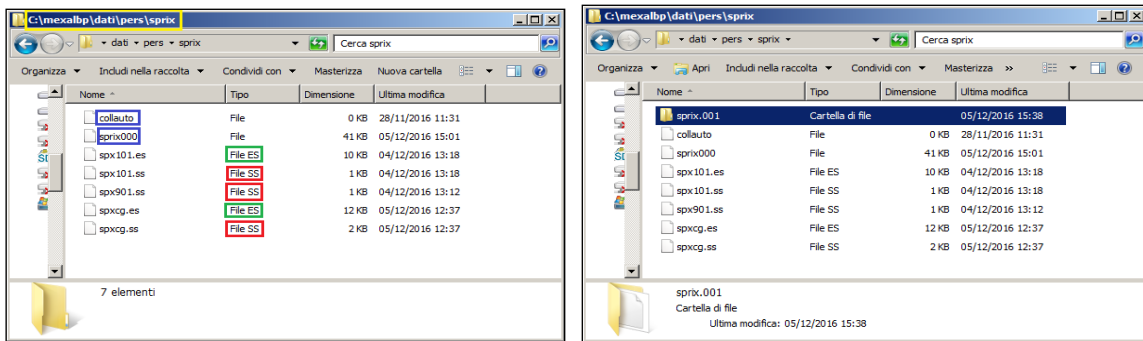
Attenzione: le seguenti immagini sui percorsi sono visibili solo nelle installazioni in **Locale** e non nelle installazioni Live (Server Farm), poiché in tale ambiente gli unici percorsi accessibili sono quelli di tipo "dato utente"; inoltre, possono riportare nomenclature obsolete in relazione alle implementazioni, pertanto, riferitevi sempre ai percorsi aggiornati che avete a video.



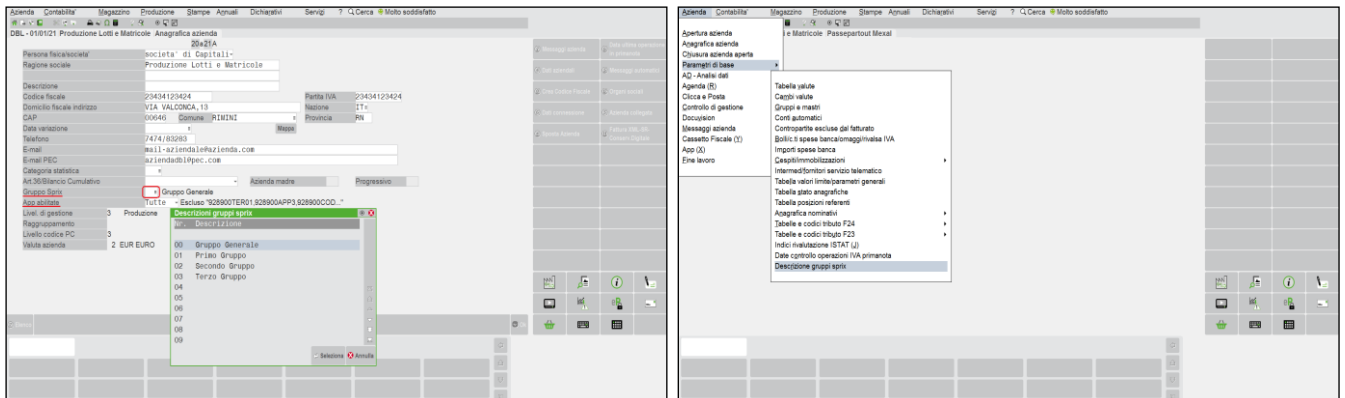


Nei percorsi dei dati Generali dell'installazione viene mantenuto anche il percorso della obsoleta organizzazione dei programmi Sprix e Collage ("...dati\pers\sprix") non più oggetto di trattazione² poiché, come detto in precedenza, attualmente è utile alla sola visualizzazione degli elenchi dei programmi ed è sostituita da quella delle Applicazioni "definitive" ("...dati\app") e delle Applicazioni in corso di "sviluppo" ("...dati\app_edit").

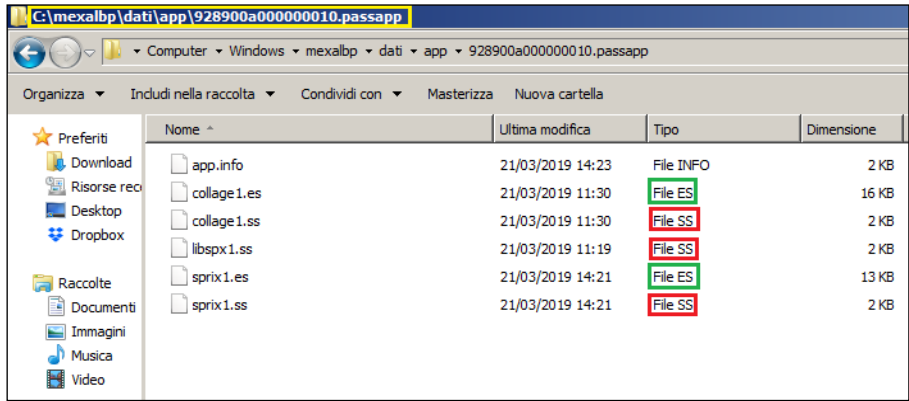
² Uno sguardo al passato. La tradizionale gestione di programmi "ad elenchi" di Sprix e Collage, depositata nel percorso dei dati generali "...dati\pers\sprix" i relativi files dei programmi; consisteva in una cartella che raccoglieva tutti i file sorgenti (con estensione ".ss") ed eseguibili (con estensione ".es") dei programmi Sprix e Collage relativi all'installazione, fra i quali anche quelli per la gestione nei vari menu ed eventi collegati agli elenchi dei programmi (ovvero i cosiddetti file "sprix000" e file "collauto").



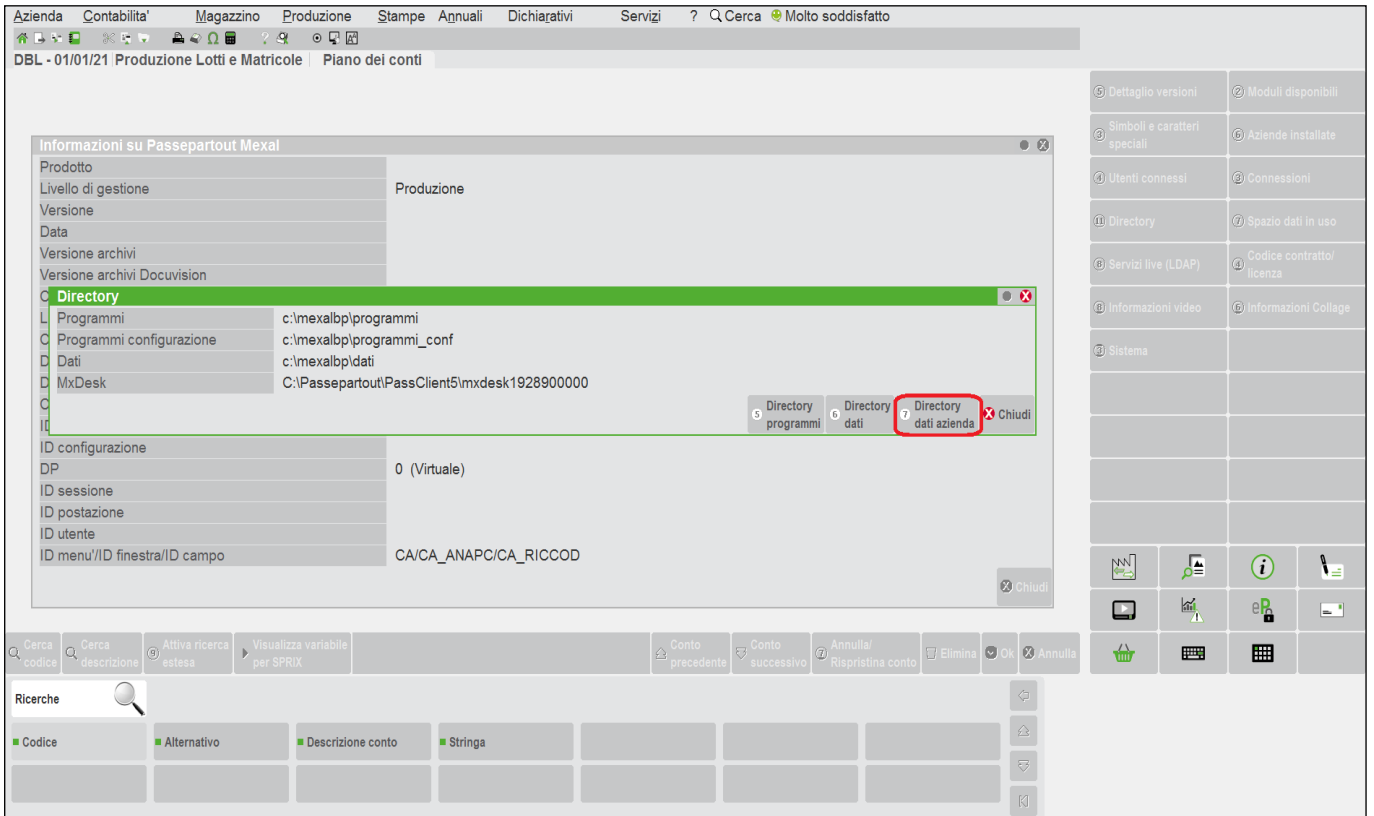
Il percorso dei programmi Sprix e Collage era dunque "...dati\pers\sprix", generale per tutta l'installazione e gestibile da sistema operativo solo in ambiente Locale. La cartella dei programmi era Generale per tutta l'installazione poiché, verificando nell'anagrafica dell'azienda, si poteva notare che poteva non avere alcun numero di "gruppo di programmi" Sprix e Collage, cioè poteva equivalere a quello cosiddetto "Generale", a meno che non si utilizzassero ripartizioni in gruppi con relativa tabella di descrizioni dei gruppi.



Relativamente ai programmi Sprix, Collage e Librerie, in entrambi i percorsi delle Applicazioni si trovano i file sorgenti e relativi eseguibili degli elementi Sprix, Collage e Librerie dell'intera Applicazione. In riferimento all'esempio fatto sono quelli rappresentati nella seguente immagine:



Ci sono i percorsi dei specifici dati Aziendali.



Quindi, a meno che non si stabilisse un apposito numero di gruppo di programmi dedicato alla singola azienda e che si presentava in una specifica sottocartella con estensione equivalente al numero di gruppo, fino a un numero di 99 gruppi e di conseguenza 99 relativi percorsi. Esempio: il gruppo numero "1" equivaleva alla sottocartella "sprix.001" e questo gruppo si poteva dedicare ad una o più aziende della stessa installazione. Questo significava che l'azienda poteva utilizzare un determinato percorso dove venivano generati i programmi; il file sprix000 era l'indice dei programmi negli elenchi (e se si fosse perduto o rovinato bastava ricostruirlo attraverso la re-imputazione dei file dagli elenchi a video dei programmi); il file collauto, invece, veniva generato in automatico dalla procedura. I restanti file erano i programmi in formato sorgente ".ss", cioè il codice in chiaro visibile nell'Editor, e gli omonimi formati compilati eseguibile ".es", cioè quelli che eseguono il programma: l'esecuzione poteva avvenire da parte di più utenti contemporaneamente, mentre l'aggiornamento del sorgente no, poteva essere in capo ad un solo utente per volta. Il file di Libreria, invece, non aveva il corrispondente (.es), poiché non si compila. Nota: come qualsiasi altro file, anche gli eseguibili possono corrompersi per vari motivi e quindi provocare un errore di tipo "errore non correggibile", pertanto, occorre anzitutto ripristinare delle copie o rigenerare il file eseguibile tramite la compilazione. Non mancava, infine, la possibilità che ciascun gruppo potesse trasformarsi automaticamente in App dell'ambiente PassBuilder e, di conseguenza, si generava una sottocartella dedicata all'azienda con le sue personalizzazioni.

Azienda Contabilita' Magazzino Produzione Stampe Annuali Dichiarativi Servizi ? Cerca Molto soddisfatto

DBL - 01/01/21 Produzione Lotti e Matricole Piano dei conti

Informazioni su Passepartout Mexal

Prodotto

L	Directory dati azienda	
V	Azienda	c:\mexalbp\dati\dataz\dbl
D	Dati utente azienda	c:\mexalbp\dati\dataz\dbl\datiutente
V	App	c:\mexalbp\dati\dataz\dbl\app
V	Dati App	c:\mexalbp\dati\dataz\dbl\datiapp
C	Dati MyDB	c:\mexalbp\dati\dataz\dbl\datimyb
L	Riclassificati	c:\mexalbp\dati\dataz\dbl\rcdir
C	Primanota facile in	c:\mexalbp\dati\dataz\dbl\prmf_in
D	Primanota facile out	c:\mexalbp\dati\dataz\dbl\prmf_out
D	Cataloghi prodotti	c:\mexalbp\dati\dataz\dbl\catalogart
C	Immagini articoli	c:\mexalbp\dati\dataz\dbl\imgart
IT	Allegati DV	c:\mexalbp\dati\dataz\dbl\vallegati
IT	Volumi DV	c:\mexalbp\dati\dataz\dbl\divvolumi
D	Sost. import moduli	c:\mexalbp\dati\dataz\dbl\imsts
IT	Ricevute	c:\mexalbp\dati\dataz\dbl\ricevute
IT	Fatture	c:\mexalbp\dati\dataz\dbl\fatture

Cerca codice Cerca descrizione Attiva ricerca estesa Visualizza variabile per SPRIX

Conto precedente Conto successivo Annulla/Rispristina conto Elimina OK Annulla

Ricerche

Codice Alternativo Descrizione conto Stringa

5 Dettaglio versioni 2 Moduli disponibili

3 Simboli e caratteri speciali 6 Aziende installate

4 Utenti connessi 8 Connessioni

11 Directory 7 Spazio dati in uso

10 Servizi live (LDAP) 9 Codice contratto/licenza

8 Informazioni video 10 Informazioni Collage

8 Sistema

Ci sono i percorsi dei programmi.

Azienda Contabilita' Magazzino Produzione Stampe Annuali Dichiarativi Servizi ? Cerca Molto soddisfatto

DBL - 01/01/21 Produzione Lotti e Matricole Piano dei conti

Informazioni su Passepartout Mexal

Prodotto

Livello di gestione Produzione

Versione

Data

Versione archivi

Versione archivi Docuision

C	Directory	
L	Programmi	c:\mexalbp\programmi
C	Programmi configurazione	c:\mexalbp\programmi_conf
D	Dati	c:\mexalbp\dati
D	MxDesk	C:\Passepartout\PassClient5\mxdesk1928900000

5 Directory programmi 6 Directory dati 7 Directory dati azienda Chiudi

ID configurazione

DP 0 (Virtuale)

ID sessione

ID postazione

ID utente

ID menu/ID finestra/ID campo CA/CA_ANAPC/CA_RICCOD

Cerca codice Cerca descrizione Attiva ricerca estesa Visualizza variabile per SPRIX

Conto precedente Conto successivo Annulla/Rispristina conto Elimina OK Annulla

Ricerche

Codice Alternativo Descrizione conto Stringa

5 Dettaglio versioni 2 Moduli disponibili

3 Simboli e caratteri speciali 6 Aziende installate

4 Utenti connessi 8 Connessioni

11 Directory 7 Spazio dati in uso

10 Servizi live (LDAP) 9 Codice contratto/licenza

8 Informazioni video 10 Informazioni Collage

8 Sistema

Azienda Contabilita' Magazzino Produzione Stampe Annuali Dichiarativi Servizi ? Q Cerca Molto soddisfatto

DBL - 01/01/21 Produzione Lotti e Matricole Piano dei conti

Directory programmi

Programmi base	c:\mexalbp\programmi
Programmi configurazione	c:\mexalbp\programmi_conf
Programmi	c:\mexalbp\programmi\prog
WebDesk Server	c:\mexalbp\programmi\progwebdesksrv
WebDesk JRE	c:\mexalbp\programmi\progwebdeskjre
Client MxDesk	c:\mexalbp\programmi\prog\mxwin
Client MxMobiledesk	c:\mexalbp\programmi\prog\mxmobdesk
Client ADP	c:\mexalbp\programmi\prog\mxadp
Client Sign	c:\mexalbp\programmi\prog\mxsign
Client Ecr	c:\mexalbp\programmi\prog\mxecr
Client PassCommunicator	c:\mexalbp\programmi\prog\mxvc
Client AD	c:\mexalbp\programmi\prog\mxad
Client MxWebApp	c:\mexalbp\programmi\prog\mxwebapp
Manuale	c:\mexalbp\programmi\manuale\pass
Manuale html	c:\mexalbp\programmi\manuale\html
Binari	c:\mexalbp\programmi\binloc
Binari master	c:\mexalbp\programmi\bindati
Binari master DR	c:\mexalbp\programmi\bindatidr
PC Pass master	c:\mexalbp\programmi\binloc\masterpcpass
Modulistica stampe	c:\mexalbp\programmi\binloc\modstn
Modulistica MsWord	c:\mexalbp\programmi\binloc\modstnw
Moduli sprix	c:\mexalbp\programmi\binloc\modsp
Modelli sprix	c:\mexalbp\programmi\binloc\modsp\importmoduli

Pagina precedente Pagina successiva Chiudi

Cerca codice descrizione estesa per SPRIX precedente successivo Rispristina conto Annulla

Ricerche

Codice	Alternativo	Descrizione conto	Stringa		

5 Dettaglio versioni 2 Moduli disponibili

3 Simboli e caratteri speciali 6 Aziende installate

4 Utenti connessi 3 Connessioni

11 Directory 2 Spazio dati in uso

8 Servizi live (LDAP) 4 Codice contratto/licenza

8 Informazioni video 6 Informazioni Collage

8 Sistema

Azienda Contabilita' Magazzino Produzione Stampe Annuali Dichiarativi Servizi ? Q Cerca Molto soddisfatto

DBL - 01/01/21 Produzione Lotti e Matricole Piano dei conti

Directory programmi

App	c:\mexalbp\programmi\binloc\appn
Tastiere	c:\mexalbp\programmi\binloc\tastieren
Icone	c:\mexalbp\programmi\binloc\iconegrafiche
Web App	c:\mexalbp\programmi\binloc\webapp
Dati Redditi Web	c:\mexalbp\programmi\binloc\webappdati\redditiweb
Stampe laser	c:\mexalbp\programmi\laser
Gerico	c:\mexalbp\programmi\gerico
Setup	c:\mexalbp\programmi\setup

Pagina precedente Pagina successiva Chiudi

Cerca codice descrizione estesa per SPRIX precedente successivo Rispristina conto Annulla

Ricerche

Codice	Alternativo	Descrizione conto	Stringa		

5 Dettaglio versioni 2 Moduli disponibili

3 Simboli e caratteri speciali 6 Aziende installate

4 Utenti connessi 3 Connessioni

11 Directory 2 Spazio dati in uso

8 Servizi live (LDAP) 4 Codice contratto/licenza

8 Informazioni video 6 Informazioni Collage

8 Sistema

In sintesi, riguardo ai percorsi di gestione dei file relativi ai programmi ed ai file personali, si può dire quanto segue sulle nomenclature:

- file personali è "...\\dati\\datigen\\" (dizionario 'sxdir') per configurazioni Server in LOCALE
- file personali è "...\\datiaz\\azi\\" (dizionario 'azdir') per configurazioni Server in LOCALE
- file personali è "...\\dati\\datigen\\datiutente\\" (abbreviabile con "I") per configurazioni Live in SERVER FARM
- file personali è "...\\dati\\datiaz\\azi\\datiutente\\" (abbreviabile con "J") per configurazioni Live in SERVER FARM; l'Idesk di uno Studio accede al solo percorso "J".
- file personali è "...\\dati\\datiaz\\azi\\app\\(nomeapp)\\nomecartella.bindatI\\" (abbreviabile con nomecartella\$nomefile)
- file personali è "...\\dati\\datiaz\\azi\\datiapp\\(nomeapp)\\datiutenteI\\" (abbreviabile con %nomefile)

Per accedere alle cartelle dati, utilizzare la sintassi NomeCartella\$NomeFile.est (Il percorso viene risolto con "I" finale dopo il nome cartella) esempio fileapp\$I\\immagine.jpg. Per ulteriori approfondimenti si rimanda al manuale PassBuilder.

Ad ogni modo, attraverso il modulo dei Servizi "Visualizzazione File" posso verificare a quali percorsi dati posso accedere, inserendo i seguenti percorsi abbreviati "I" (che significa "Dati Utente") oppure "J" (che significa "Dati Utente Azienda") e navigare nei percorsi premendo Elenco File [F3] o Elenco Archivi [F2].

Attenzione: l'utilizzo di caratteri speciali (esempio: ù, ü, à, ecc..) nei nomi file e/o nei percorsi Server accessibili, può impedire l'aggiornamento degli archivi dell'azienda che ne fa uso e ne rilascia il conseguente messaggio "punto di ripristino non creato". Rimuovere i caratteri e ripetere l'aggiornamento.

L'installazione Locale consente una maggiore libertà di navigazione dei percorsi rispetto all'installazione Live, che ha restrizioni d'ambiente a protezione della Server Farm, pertanto, si può inserire solo "I" oppure "J" per accedere al percorso <DATI GENERALI> oppure <DATI AZIENDA>, gli unici in cui si può gestire file a livello di installazione generale oppure della specifica azienda. Si può verificare, inoltre, che Sprix non può accedere alla cartella "prog" del Server del Gestionale.

Notare Bene! Nelle migrazioni dall'ambiente locale verso l'ambiente Hosting (Server Farm), prestare sempre particolare attenzione alle informazioni fornite dalla documentazione di Passaggio Hosting, sulla gestione dei percorsi utilizzati dalle personalizzazioni. In caso di dubbi o incertezze, chiedere sempre chiarimenti e approfondimenti prima di fornire all'Hosting le cartelle dati per la migrazione.

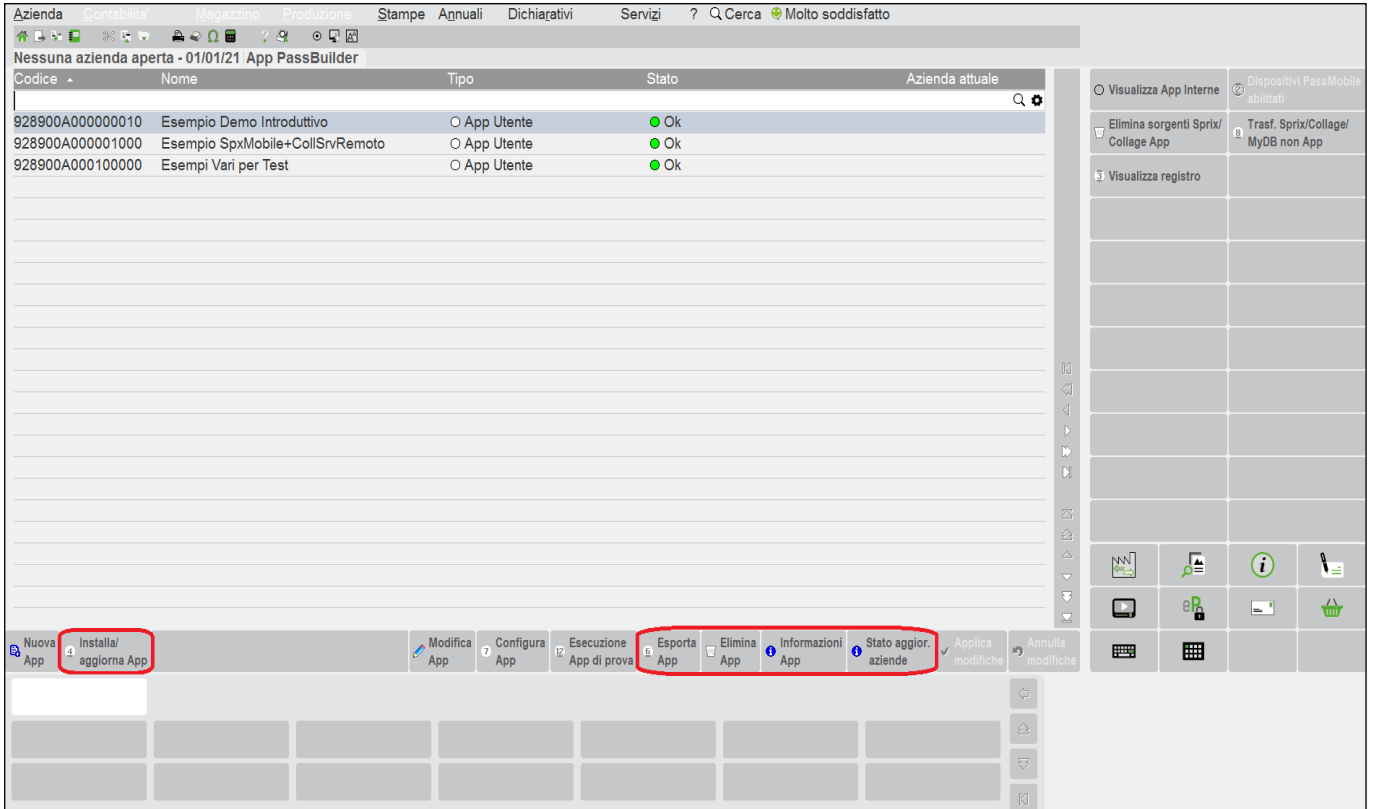
Per ulteriori indicazioni relative ai percorsi dei nomi file si rimanda alla [Nomenclatura File](#) presente nella sezione "DATABASE – FILE – PORTA COM – SHELL" del capitolo dei Comandi per Argomento.

Un concetto importante: i programmi Sprix e Collage sono solo apparentemente eseguiti dal Client perchè, in realtà, tutto viene eseguito dal Server; per ottenere effetti anche lato Client occorre re-dirigere i comandi tramite il seguente prefisso "<" da aggiungere al percorso del Client (es. "<C:..."). I comandi che sfruttano delle librerie di terze parti (quindi non Passepartout) per la gestione di alcune specifiche funzionalità (es. i formati compressi "ZIP") potrebbero non riconoscere la sintassi del reindirizzamento (redirezione Client).

La personalizzazione può, inoltre, eseguire Sprix in un Server di un'altra installazione attraverso il comando SPXREMOTO\$.

GESTIONE DELL'APPLICAZIONE E DEI SUOI ELEMENTI

Le funzioni standard del Gestionale Passepartout non possono essere compromesse dalle Applicazioni. I moduli standard funzionano sempre e comunque, poichè non sono vincolati dalla procedura applicativa, dato che, come già accennato, **può essere aggiornata, disattivata, rimossa o sostituita senza ripercussione sull'integrità dei dati e funzionalità standard**, nemmeno cambiando il sistema operativo, configurazioni o livello di prodotto. Le Applicazioni sono disattivabili anche senza entrare nell'azienda, proprio per consentire di sbloccare le funzionalità standard dagli eventuali malfunzionamenti delle personalizzazioni. In ogni caso, le esecuzioni sugli ambienti di prova e in aggiunta ai backup, mettono sempre lo sviluppatore previdente al riparo da qualsiasi evenienza.



Codice	Nome	Tipo	Stato	Azienda attuale
928900A000000010	Esempio Demo Introduttivo	<input type="radio"/> App Utente	● Ok	
928900A000001000	Esempio SpxMobile+CollSrvRemoto	<input type="radio"/> App Utente	● Ok	
928900A000100000	Esempi Vari per Test	<input type="radio"/> App Utente	● Ok	

Azienda Contabilità Magazzino Produzione Stampe Annuali Dichiarativi Servizi ? Cerca Molto soddisfatto

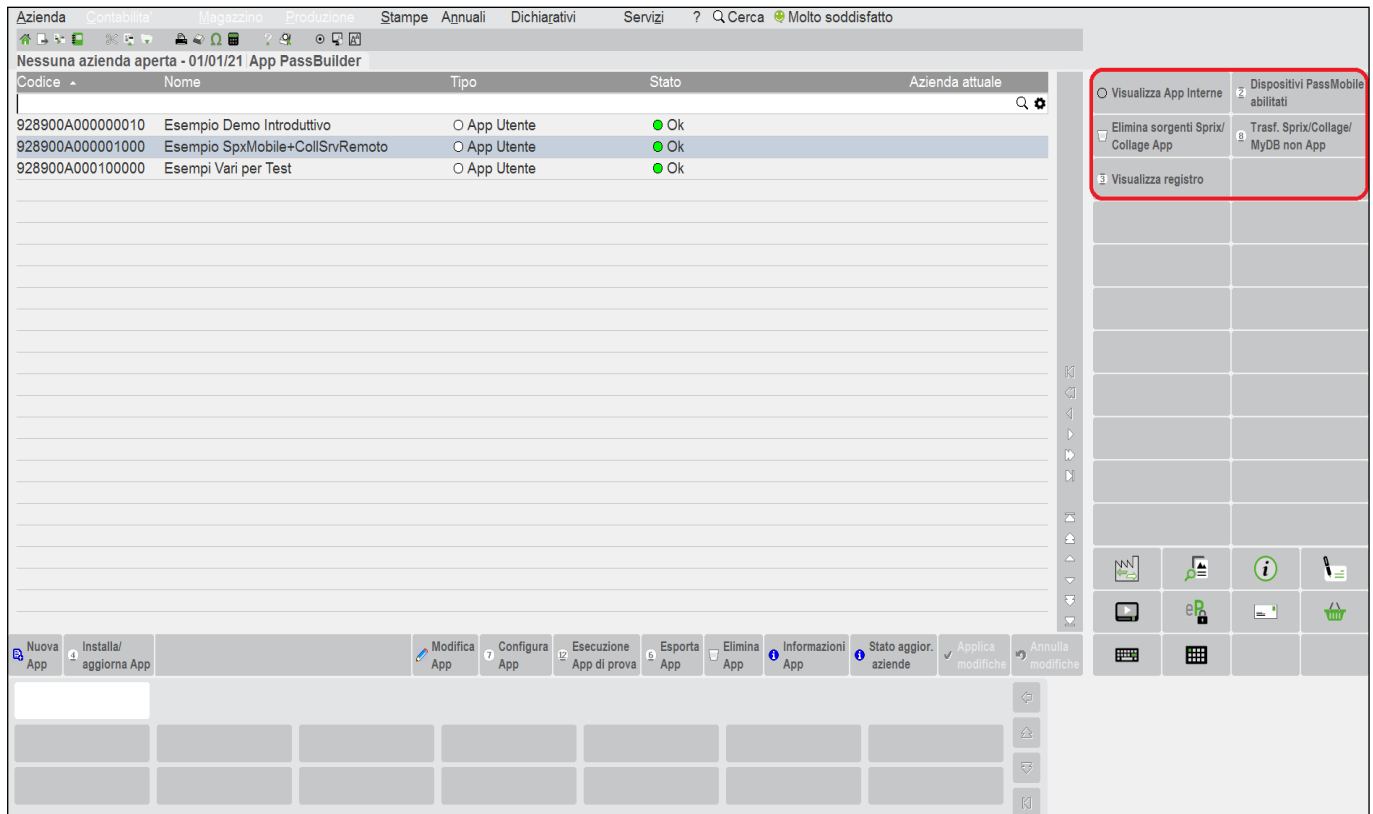
Stato aggiornamento aziende 928900A000000010 "Esempio Demo Introduttivo"

Nessuna azienda aperta - 01/01/21 | App PassBuilder

Codice	Nome	Sigla	Nome azienda	App abilitata	Stato aggiornamento azienda
928900A000000010	Esempio Demo Introduttivo	ALM			⚠ Azienda ALM stato aggiornamento non determinabile
928900A000001000	Esempio SpxMobile+CollSrvRemot	BIK			⚠ Azienda BIK stato aggiornamento non determinabile
928900A000100000	Esempi Vari per Test	CDG			⚠ Azienda CDG stato aggiornamento non determinabile
		DBL	Produzione Lotti e Matricole	<input type="radio"/> No	<input type="radio"/> Azienda DBL App disabilitata
		FIF			⚠ Azienda FIF stato aggiornamento non determinabile
		FRA			⚠ Azienda FRA stato aggiornamento non determinabile
		GAM			⚠ Azienda GAM stato aggiornamento non determinabile
		IEG			⚠ Azienda IEG stato aggiornamento non determinabile
		LIM			⚠ Azienda LIM stato aggiornamento non determinabile
		PAL			⚠ Azienda PAL stato aggiornamento non determinabile
		PAO			⚠ Azienda PAO stato aggiornamento non determinabile
		PMF			⚠ Azienda PMF stato aggiornamento non determinabile
		SER			⚠ Azienda SER stato aggiornamento non determinabile
		TIP			⚠ Azienda TIP stato aggiornamento non determinabile
		VIN			⚠ Azienda VIN stato aggiornamento non determinabile
		ZUC			⚠ Azienda ZUC stato aggiornamento non determinabile
		ZZ1			⚠ Azienda ZZ1 stato aggiornamento non determinabile
		ZZ2			⚠ Azienda ZZ2 stato aggiornamento non determinabile

Sono disponibili ulteriori funzioni per configurare la natura dell'App (esempio la presenza di dispositivi "Mobile"), la consultazione dello specifico registro (senza dimenticare che ci sono altri importanti file di log dell'installazione in generale), la "riservatezza" dei sorgenti con la possibilità di eliminarli dall'App (e ciò richiede apposita licenza chiamata "MDS"), la possibilità di "Trasferire Sprix/Collage/MyDB non App" cioè la funzione che consente di convertire il "Gruppo Sprix" (di cui abbiamo già trattato, ovvero la cartella "sprix.nnn" contenuta nel percorso "...pers\sprix\") in un'unica Applicazione. La conversione dei programmi può avvenire in modo massivo in un'unica App oppure tante App per quanti sono i gruppi. Per approfondimenti sulla trasformazione consultare il capitolo "APP PASSBUILDER" del manuale PassBuilder all'interno del Menu Servizi del Manuale Utente.

Riguardo alla possibilità di eliminare i sorgenti significa che vengono tolti dall'App i vari file con estensione ".ss" e restano installati i soli file eseguibili ".es": ciò consente l'esecuzione del programma senza la contemporanea presenza del sorgente ".ss", per non rilasciare il codice. È superfluo ricordare di conservare copie integre del sorgente, non solo per sicurezza, ma anche per fini pratici, poiché nella particolarità del funzionamento ad eventi che caratterizza il Collage, potrebbe essere utile per sbloccare un eseguibile Collage che rimane caricato in memoria in seguito a manipolazioni sui file in presenza d'attivazioni automatiche oppure loop di codice; questi tipi di accadimenti erano possibili in passato³, ma PassBuilder ha risolto e semplificato molto le condizioni.



The screenshot shows the PassBuilder application interface. At the top, there are menu items: Azienda, Contabilità, Magazzino, Produzione, Stampe, Annuali, Dichiarativi, Servizi, and a search bar. Below the menu, a status bar indicates "Nessuna azienda aperta - 01/01/21 App PassBuilder". The main area contains a table with the following data:

Codice	Nome	Tipo	Stato	Azienda attuale
928900A000000010	Esempio Demo Introduttivo	○ App Utente	● Ok	
928900A000001000	Esempio SpxMobile+CollSrvRemoto	○ App Utente	● Ok	
928900A000100000	Esempi Vari per Test	○ App Utente	● Ok	

On the right side, there is a sidebar with several options, some of which are highlighted with a red box:

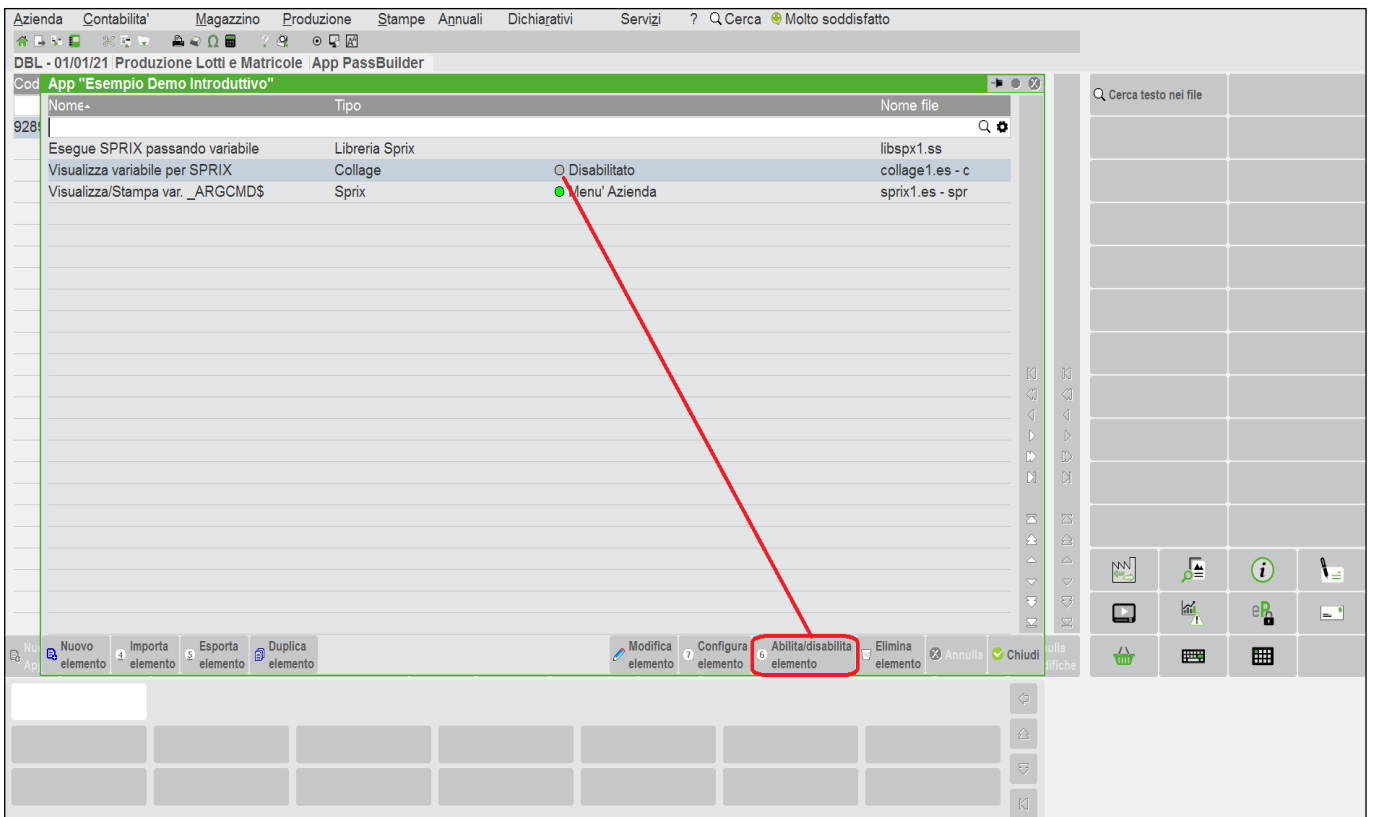
- Visualizza App Interne
- Dispositivi PassMobile abilitati
- Elimina sorgenti Sprix/Collage App
- Trasf. Sprix/Collage/MyDB non App
- Visualizza registro

At the bottom, there is a toolbar with various icons for actions like Nuova App, Installa/aggiorna App, Modifica App, Configura App, Esecuzione App di prova, Esporta App, Elimina App, Informazioni App, Stato aggiorn. aziende, Applica modifiche, and Annulla modifiche.

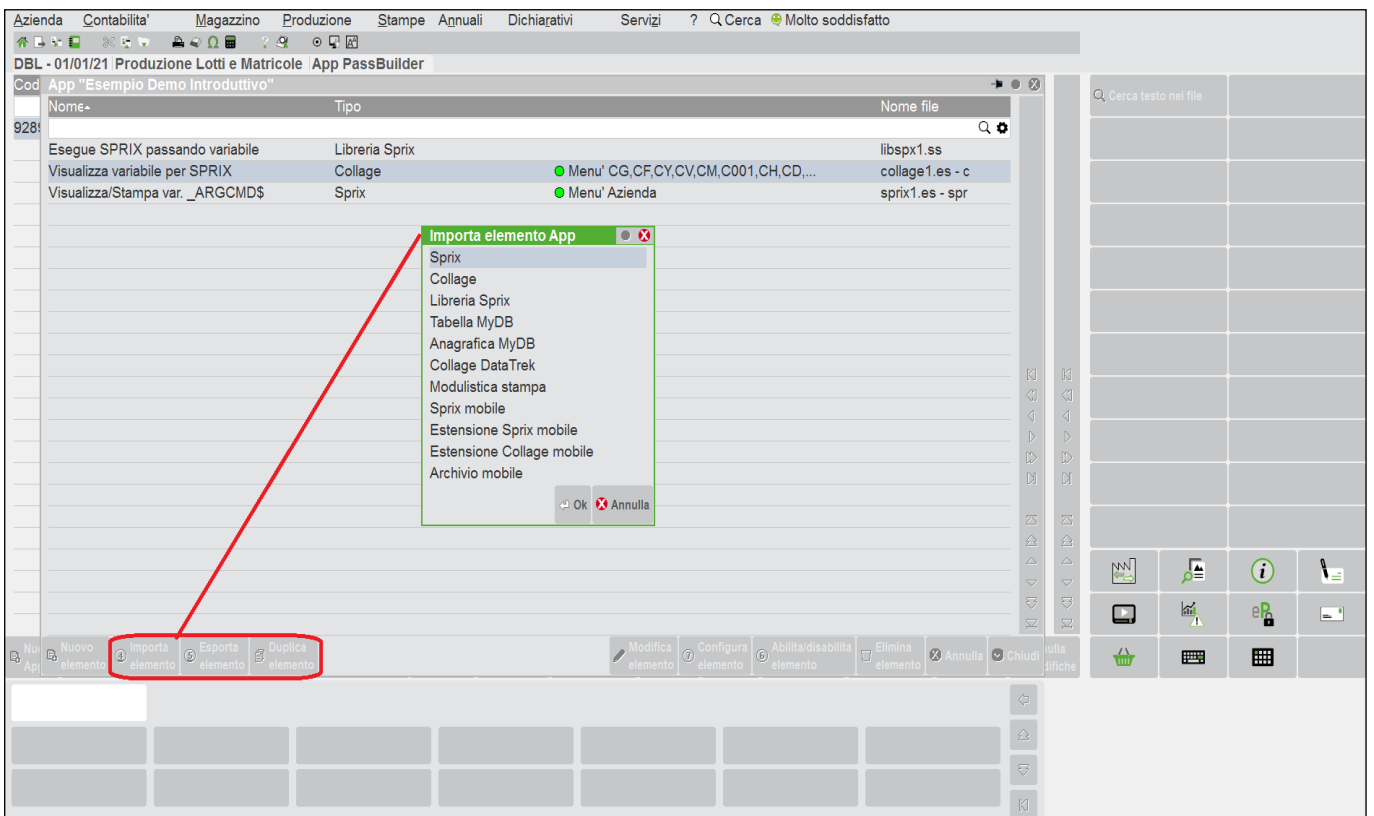
³ In passato occorre il reboot del intero Gestionale dopo l'azzeramento del file "collauto". In seguito a manipolazioni sui file (esempio la cancellazione del sorgente), può capitare che l'eseguibile Collage resti caricato in memoria, oppure, nonostante sia cancellato l'eseguibile venga ancora cercato e non trovato (specialmente se è stato compilato su menu AUTO e senza averlo disattivato con [F6] nell'elenco dei Collage). In tal caso, non è sufficiente far ripartire l'intero Gestionale, ma occorre azzerare (cancellare) il file "collauto".

In passato alcune operazioni sugli elenchi dei file sprix potevano rilasciare il messaggio "sprix "<nomesprix->" (spxNNN.es) non eseguibile da questo menu" è causato dallo spostamento dell'eseguibile dalla sua originaria posizione numerata nell'elenco di menu (per esempio quando si copia / sovrascrive un eseguibile su uno precedente); quando la nuova posizione dell'eseguibile è diversa rispetto a quella che lo stesso aveva quando è stato generato dall'ultima compilazione (viene marcato in base alla posizione), si genera questo messaggio. Esempio: i file "spx301.ss" e "spx301.es" sono originariamente collegati alla posizione 301, ma poi si sposta il solo eseguibile "spx301.es" nella posizione 302 (senza portare il sorgente "spx301.ss" per la corretta ricompilazione). Di conseguenza, l'eseguibile continua ad essere marcato con posizione 301, ma non consente l'esecuzione dalla posizione 302, nemmeno tramite lancio da comando SHELLSPRIX. Soluzione (a): ricompilare il sorgente nella nuova posizione di menu per generare il nuovo aggancio Soluzione (b): riagganciare l'eseguibile nella posizione d'origine, come si fa con l'installazione dell'eseguibile senza il relativo sorgente.

Nemmeno i malfunzionamenti degli Elementi dell'Applicazione possono bloccare le procedure standard, dato che anche questi elementi sono disattivabili⁴ oppure non eseguibili e senza alcuna ripercussione sui dati e sulle funzioni standard.

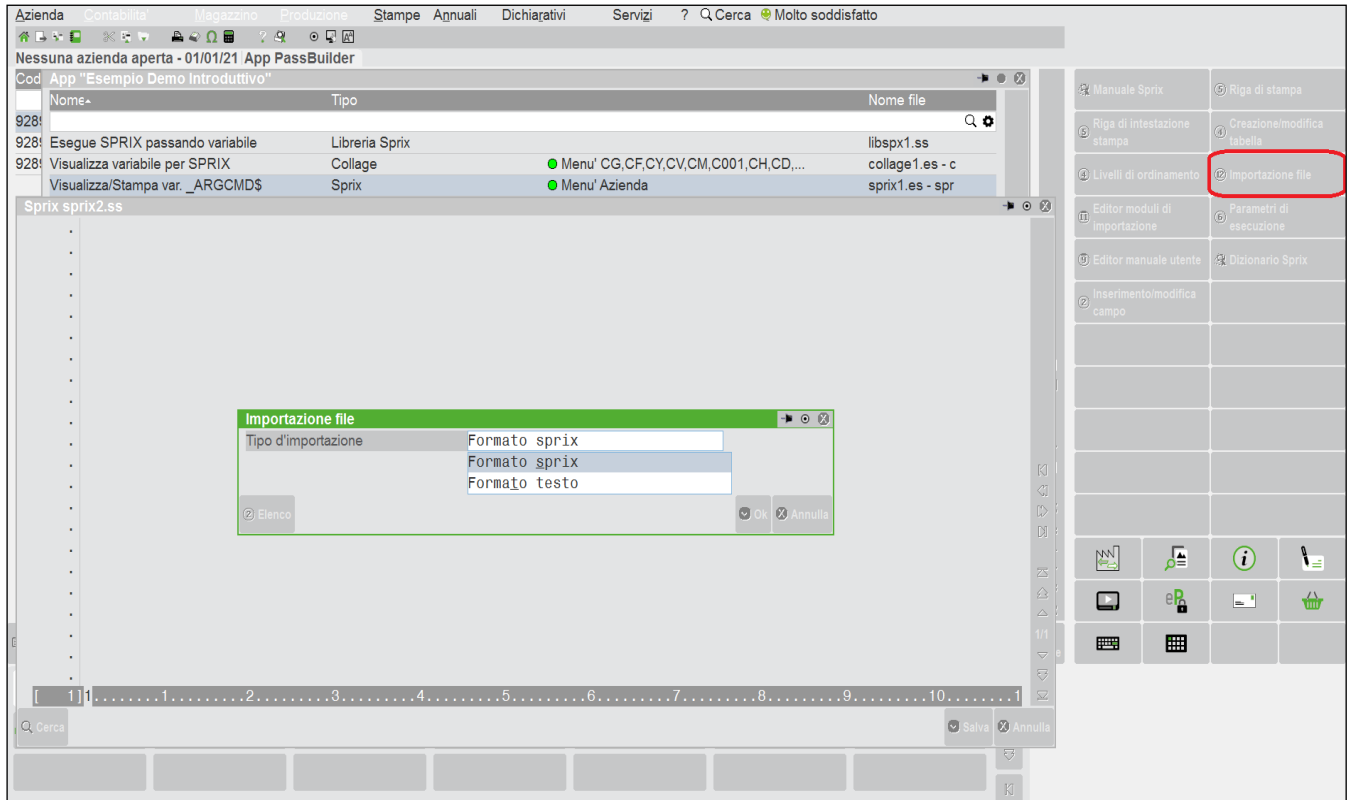


L'App nel suo complesso è aggiornabile ed esportabile, ma è anche possibile importarvi, esportare e duplicarne gli elementi. Se lo sprix viene importato dalla gestione fuori App, questo è riportato senza la descrizione in quanto l'informazione non è presente nel sorgente, ma in un file separato; stessa cosa avviene per Collage, che inoltre, viene inserito di default nel menu AUTO.



⁴ Questa possibilità consente di disattivare Collage nel caso in cui più Applicazioni sulla stessa installazione con molteplici Aziende contengano dei Collage che possano attivarsi sulle Aziende ed in modo non desiderato, ma non si abbia, però, l'intenzione di disattivare l'intera Applicazione per le Aziende in questione; in alternativa, occorrerebbe sabilire nello stesso sorgente del Collage delle condizioni (IF) che ne consentano l'esecuzione in caso dell'azienda di interesse (azsig), oppure, se l'installazione è locale, sarebbe possibile stabilire dei gruppi di Applicazioni da gestire a livello di cartella su percorso "\dat\app".

Anche dall'interno dell'Editor si può importare del codice sia in formato sorgente ".ss" che in formato testo "txt". Il caricamento avviene indicando il percorso da cui caricare il file. Nei sistemi Live è consentito prelevare da percorso del medesimo Server con default sui percorsi consentiti e/o accessibili oppure è consentito acquisire dettagliando un percorso ridiretto su Client remoto ("<C:\Tmp\..."). Nei sistemi Locali il modo è analogo, ma l'accesso ai percorsi del sistema è maggiore secondo i criteri stabiliti da Administrator, che siano in Lan o Localhost. Per procedure upload di file da dispositivo/sistema operativo locale Client WebDesk consultare il manuale generale sez. "Nozioni Generali", mod. "WebDesk Client", cap. "Utilizzo", par. "Acquisizione file per invio al server". Si ricorda che l'import non considera i file di Libreria e neppure il comando del tasto funzione dell'Editor [F6] Parametri di esecuzione (consultare la relativa voce nei comandi), pertanto, anche la Libreria va importata a parte ed il comando [F6] (una macro di input variabili) va interamente ricostruito.



Per quanto riguarda l'import in formato sorgente si può prelevare dal percorso accessibile e/o consentito...

Azienda Contabilità Magazzino Produzione Stampe Annuali Dichiarativi Servizi ? Cerca Molto soddisfatto

Nessuna azienda aperta - 01/01/21 App PassBuilder

Cod App "Esempio Demo Introduttivo"

Sprix sprix2.ss

Importazione file

Tipo d'importazione

File sprix da importare

Elenco file in c:\mexalbp\dati\app_edit\928900a000000010.passapp [*.]:

- ..
- app.info
- collage1.es
- collage1.ss
- libspx1.ss
- sprix1.es
- sprix1.ss
- sprix2.ss

1 1 2

Q Cerca

Filtro Seleziona Seleziona Directory Annulla Salva Annulla

Drive Directory client

Proprieta' directory Dimensione

Livelli di ordinamento Importazione file

Editor moduli di importazione Parametri di esecuzione

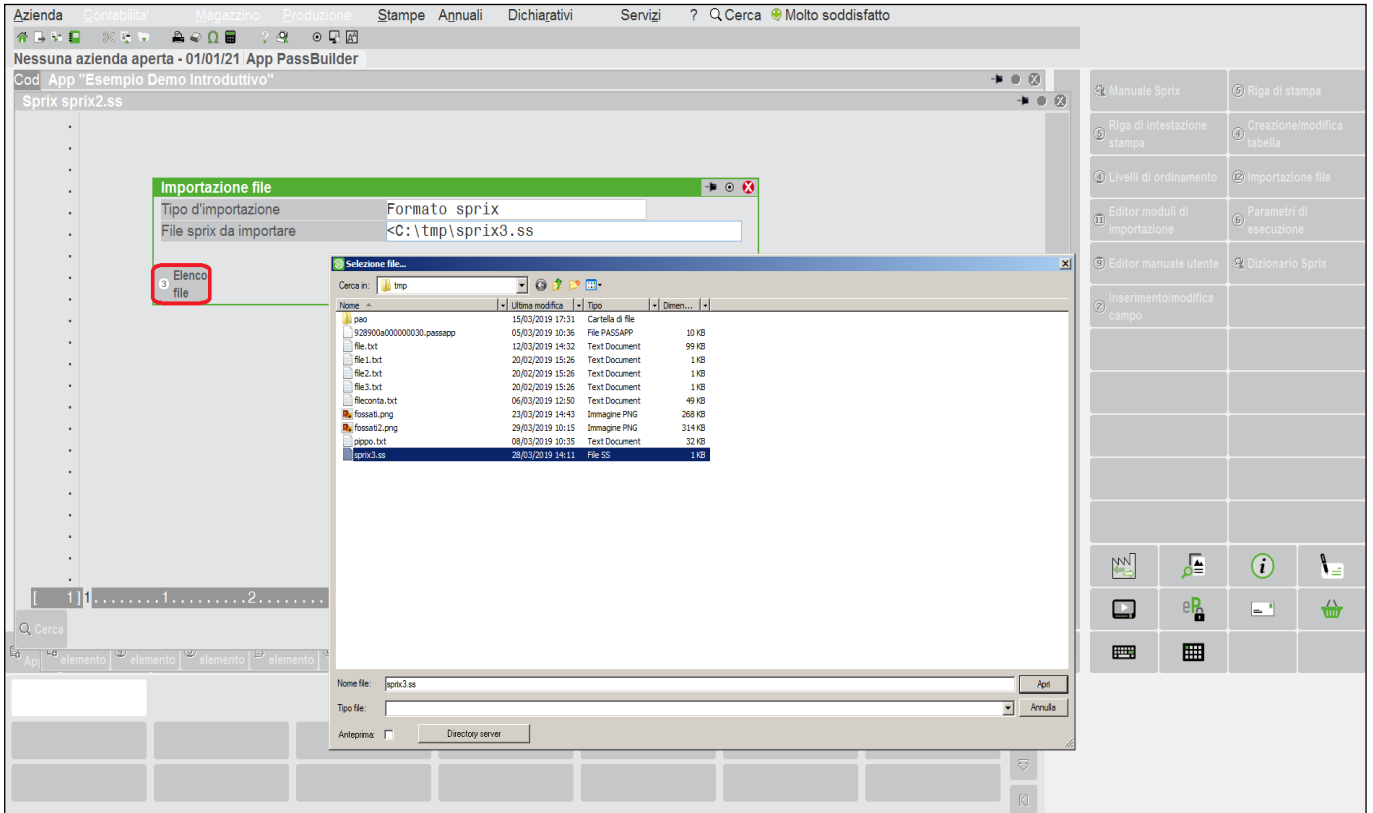
Editor manuale utente Dizionario Sprix

Inserimento/modifica campo

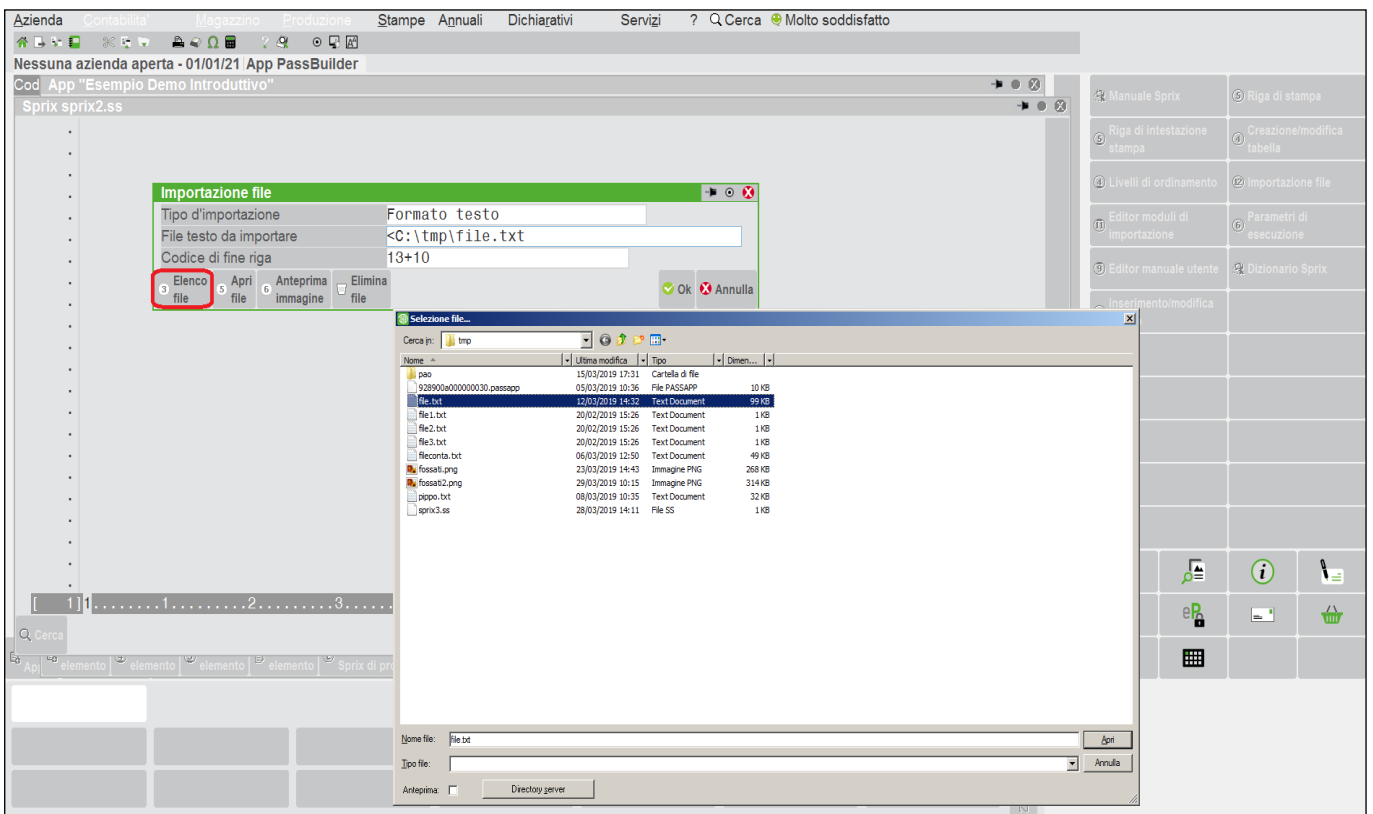
1/1

App elemento elemento elemento elemento Sprix di prova elemento elemento elemento elemento

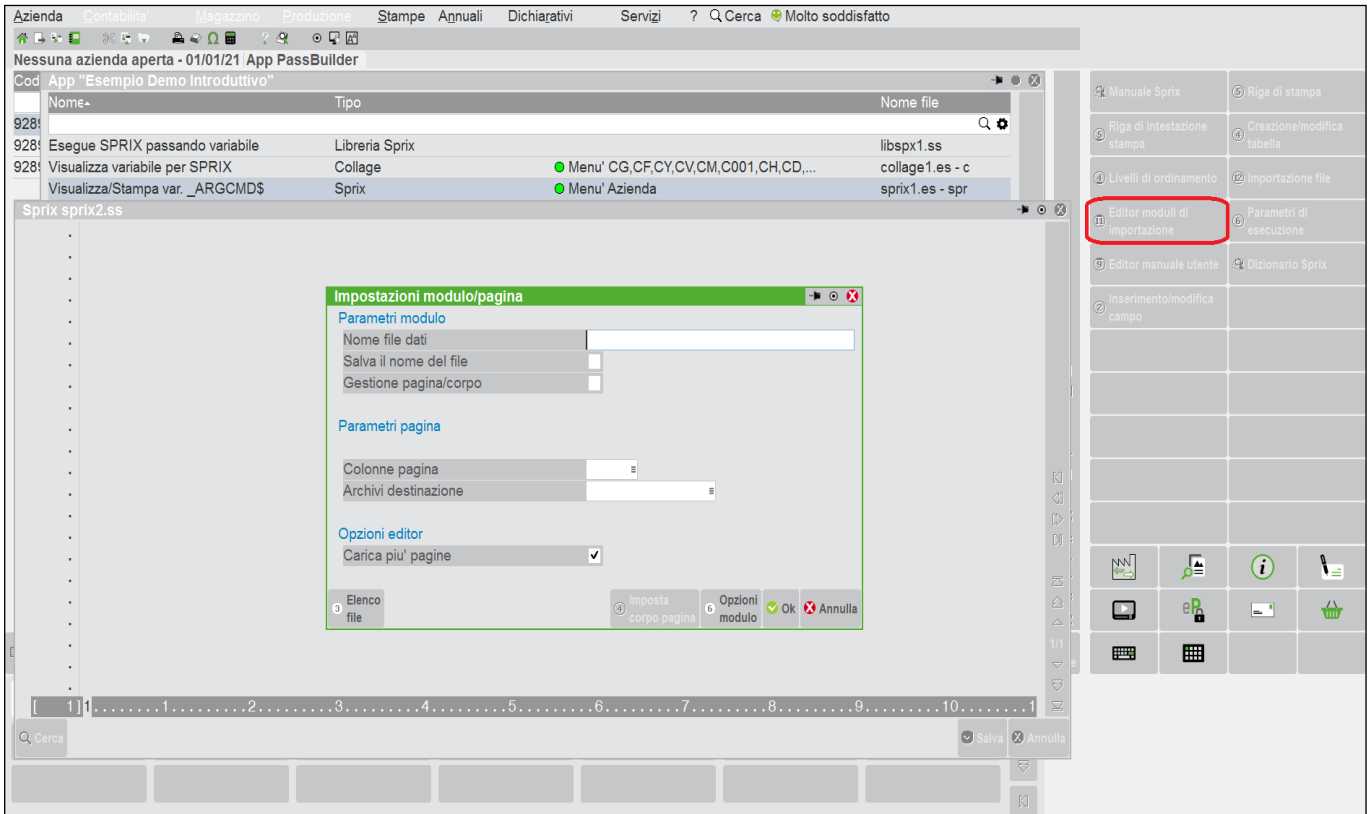
...oppure si può prelevare dal Client Locale (tramite il comando "<") per caricarlo, quindi, nel Server Remoto.



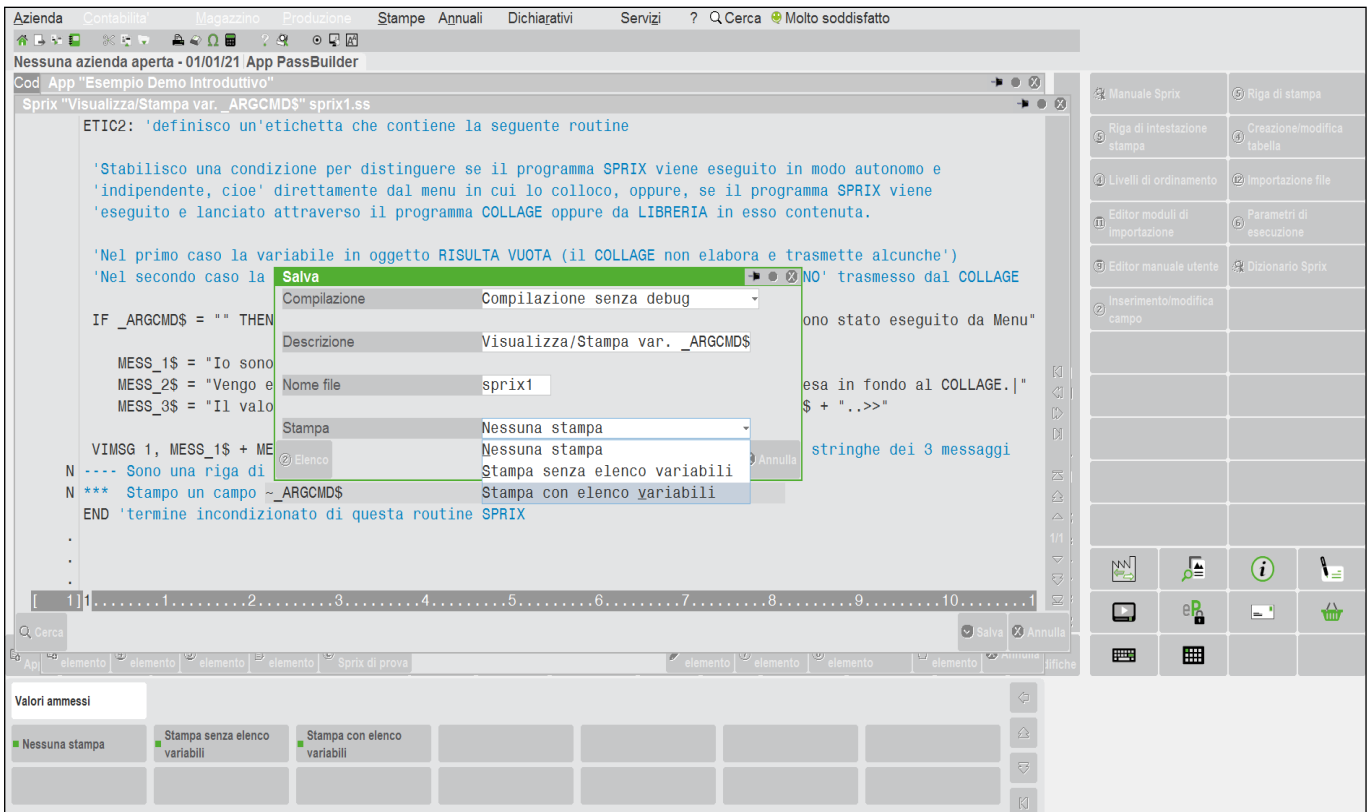
Per quanto riguarda l'import in formato testo, come per il formato sorgente si può prelevare dal percorso accessibile e/o consentito in Locale e/o Remoto. L'import di file testo richiede un percorso esteso e con l'indicazione di codici ASCII dei Caratteri Fine Record: il default Windows è 13+10 (CRLF:CarriageReturn+LineFeed), in Linux è 10; questi caratteri consentono la prosecuzione del import su una nuova riga, specialmente quando il testo eccede la lunghezza Editor. I caratteri non consentiti sono sostituiti da blank (32). Dalle "Informazioni su" del menu selezioni la char set dei "Simboli e caratteri speciali".



Nell'Editor si può generare codice disegnando un modulo. Questa funzionalità è relativa ai comandi IMOD che consentono le importazioni di dati attraverso codice che viene generato disegnando e impostando moduli. L'acronimo IMOD significa "Import Moduli". Lo si può trovare nell'elenco dei MODULI E ARCHIVI DEL GESTIONALE di questo manuale.



La finestra di Salvataggio e Compilazione consente di gestire alcune operazioni sul sorgente, per esempio la stampa in formato testo, anche con la collocazione delle etichette e delle variabili.



L'eventuale successivo import in formato testo, però, interpreta il campo di stampa come una costante (come indicata nell'Editor). L'import testo, quindi, avviene con caratteristiche della riga (stampa / operativa / commento) compatibilmente a quelle dell'Editor. L'import di testo è consentito anche con Copia (Ctrl+C) e Incolla (Ctrl+V), ma anche questo metodo non importa campi di stampa, né i Parametri di esecuzione [F6], poiché il "copia/incolla" avviene tramite la "clipboard".

```
(Generale): 29/3/2019 16:27

1      ETIC2: 'definisco un'etichetta che contiene la seguente routine
2
3      'Stabilisco una condizione per distinguere se il programma SPRIX viene eseguito in modo autonomo e
4      'indipendente, cioe' direttamente dal menu in cui lo colloco, oppure, se il programma SPRIX viene
5      'eseguito e lanciato attraverso il programma COLLAGE oppure da LIBRERIA in esso contenuta.
6
7      'Nel primo caso la variabile in oggetto RISULTA VUOTA (il COLLAGE non elabora e trasmette alcunche')
8      'Nel secondo caso la variabile in oggetto NON RISULTA VUOTA e riporta il 'BUONGIORNO' trasmesso dal COLLAGE
9
10     IF _ARGCMD$ = "" THEN _ARGCMD$ = "NESSUN VALORE !!! perche' io, programma SPRIX, sono stato eseguito da Menu"
11
12     MESS_1$ = "Io sono il programma SPRIX|"
13     MESS_2$ = "Vengo eseguito dall'istruzione SHELLSPRIX del codice in LIBRERIA appesa in fondo al COLLAGE.|"
14     MESS_3$ = "Il valore della variabile che mi passa il COLLAGE e'|<<..\" + _ARGCMD$ + \"..>>"
15
16     VIMSG 1, MESS_1$ + MESS_2$ + MESS_3$ ' visualizzazione della concatenazione delle stringhe dei 3 messaggi
17     N ---- Sono una riga di stampa ----
18     N *** Stampa un campo ~ _ARGCMD$
19     END 'termine incondizionato di questa routine SPRIX
```

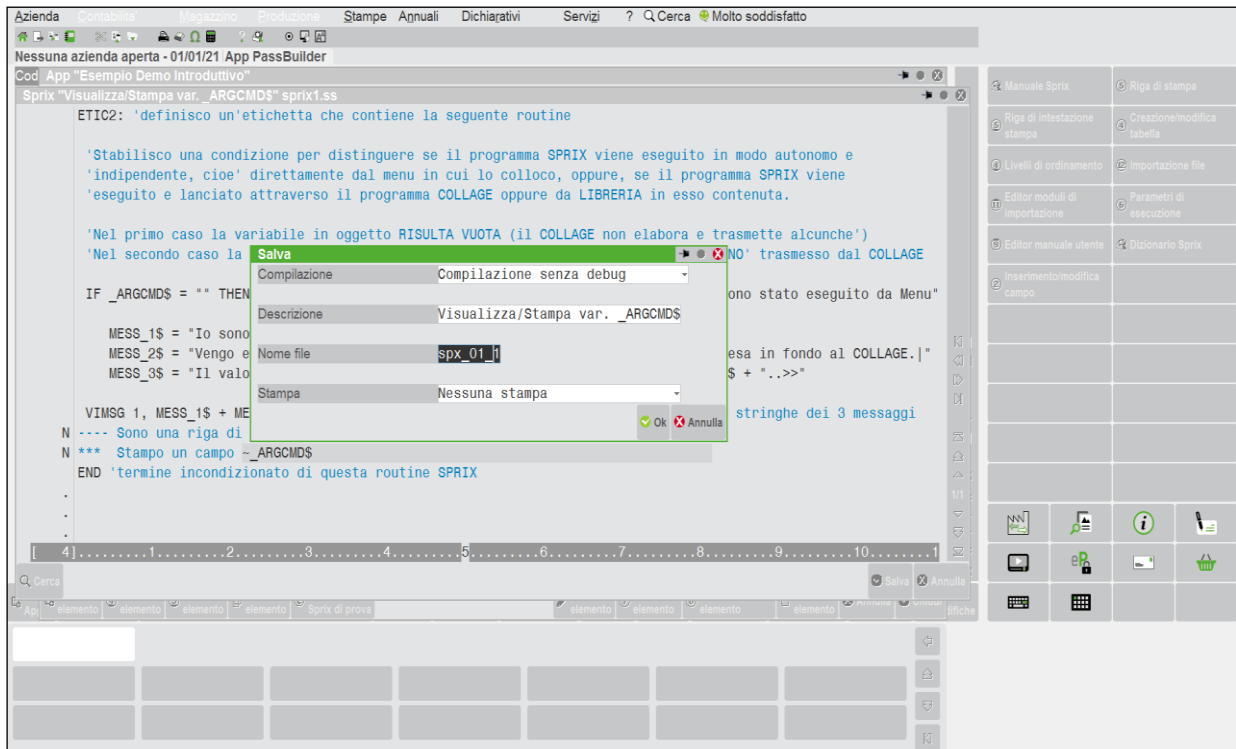
Lista variabili (nome / righe in cui compare)

MESS_1\$	12	16
MESS_2\$	13	16
MESS_3\$	14	16
_ARGCMD\$	10	10 14

Lista etichette (nome / righe in cui compare)

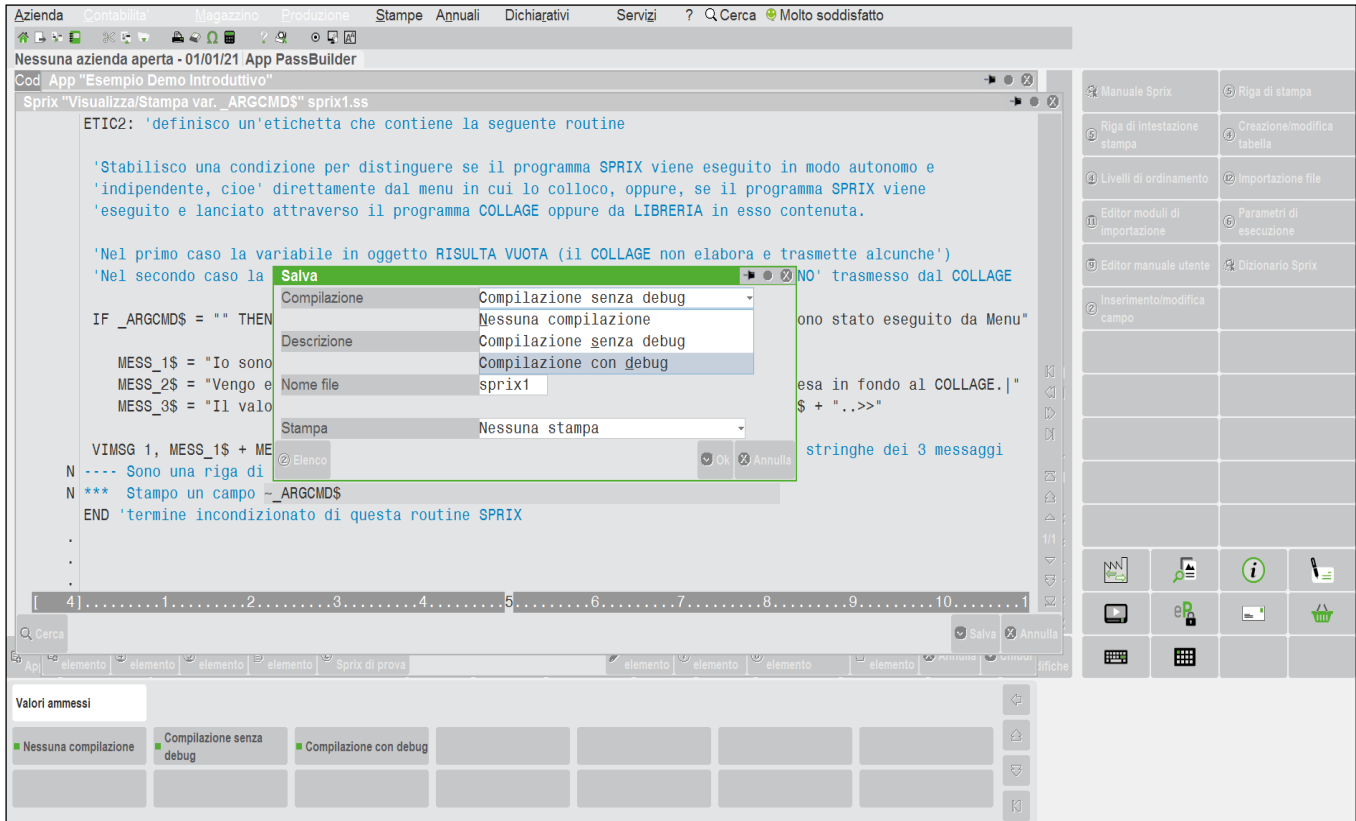
ETIC2	1
-------	---

Nell'eventuale personalizzazione del nome del file, non utilizzare caratteri speciali non accettati dal sistema operativo in uso (es: asterisco (*)), perchè potrebbe andare in errore il sistema.



Infine la possibilità di salvare semplicemente, compilare con eventuale debug molto utile. La personalizzazione è verificata dalla piattaforma con un compilatore integrato non vincolato al sistema operativo e viene resa trasparente al sistema, senza obbligo di manutenzioni a seguito di aggiornamenti del Passepartout.

La funzione di salvataggio ha, quindi, anche le prerogative di **compilatore**, esegue controlli di correttezza e validità del codice per generare il relativo eseguibile. Per salvare un codice incompleto si salva con “Nessuna compilazione” e serve a tutelare il sorgente aperto nell'Editor, poiché danneggiabile da eventuali crash di sistema.



In assenza di errori la compilazione genera l'eseguibile e il programma può essere eseguito dal relativo menu oppure anche da “Esecuzione di prova” con controlli di compilazione.

Azienda Contabilità Magazzino Produzione Stampe Annuali Dichiarativi Servizi ? Q Cerca Molto soddisfatto

Apertura azienda
 Anagrafica azienda
 Chiusura azienda aperta
 Parametri di base
 AQ - Analisi dati
 Agenda (R)
 Clicca e Posta
 Controllo di gestione
 Docuvision
 Messaggi azienda
 Cassetto Fiscale (Y)
 App (X)
 Fine lavoro


App PassBuilder

	Tipo	Nome file
riabile	Libreria Sprix	libsp1.ss
X	Collage	collage1.es - c
CMDS	Sprix	spr1.es - spr

Esempio Demo Introduttivo

Visualizza/Stampa var. _ARGCMD\$

Nuova App
 Nuovo elemento
 Importa elemento
 Esporta elemento
 Duplica elemento
 Esecuzione Sprix di prova
 Modifica elemento
 Configura elemento
 Abilita/disabilita elemento
 Elimina elemento
 Annulla
 Chiudi

La gestione dei file (dall'accesso al sorgente alla compilazione ed esecuzione dell'eseguibile) talvolta potrebbe generare messaggi di errore che non consentono tale possibilità: "l'eseguibile è in creazione o in esecuzione in un altro terminale"; "operazione non effettuabile, l'eseguibile è in uso da un altro terminale"; "operazione non effettuabile. Il sorgente è in uso da un altro terminale". Le condizioni possono essere le più varie, attenuate dalla possibilità di avere una cartella di App "definitive" e una cartella di App in "edit" 

La compilazione consente di eseguire il debug in Sprix e in Collage. La "Esecuzione di prova" non è consentita in Collage (come già detto viene aperta una sessione di prova): l'apertura del debugger in Collage è all'evento che ho stabilito in un determinato punto del Passepartout. L'apertura in Sprix, invece, avviene eseguendo lo Sprix dal menu, oppure, dopo che ho salvato con debug e la "Esecuzione di prova".

Il debug viene eseguito se viene soddisfatta la condizione di compilazione, perché consente di cercare errori sul flusso logico della procedura attraverso l'esecuzione del programma **un'istruzione alla volta** con "Esegui un'istruzione", poiché il linguaggio ha un'elaborazione sequenziale riga per riga; ciò consente di verificarne il valore delle variabili e stabilire break point d'interruzione fino a una determinata riga. Debug apre la finestra d'esecuzione e si posiziona sulla prima istruzione che è pronta ad eseguire (visualizzata in reverse).

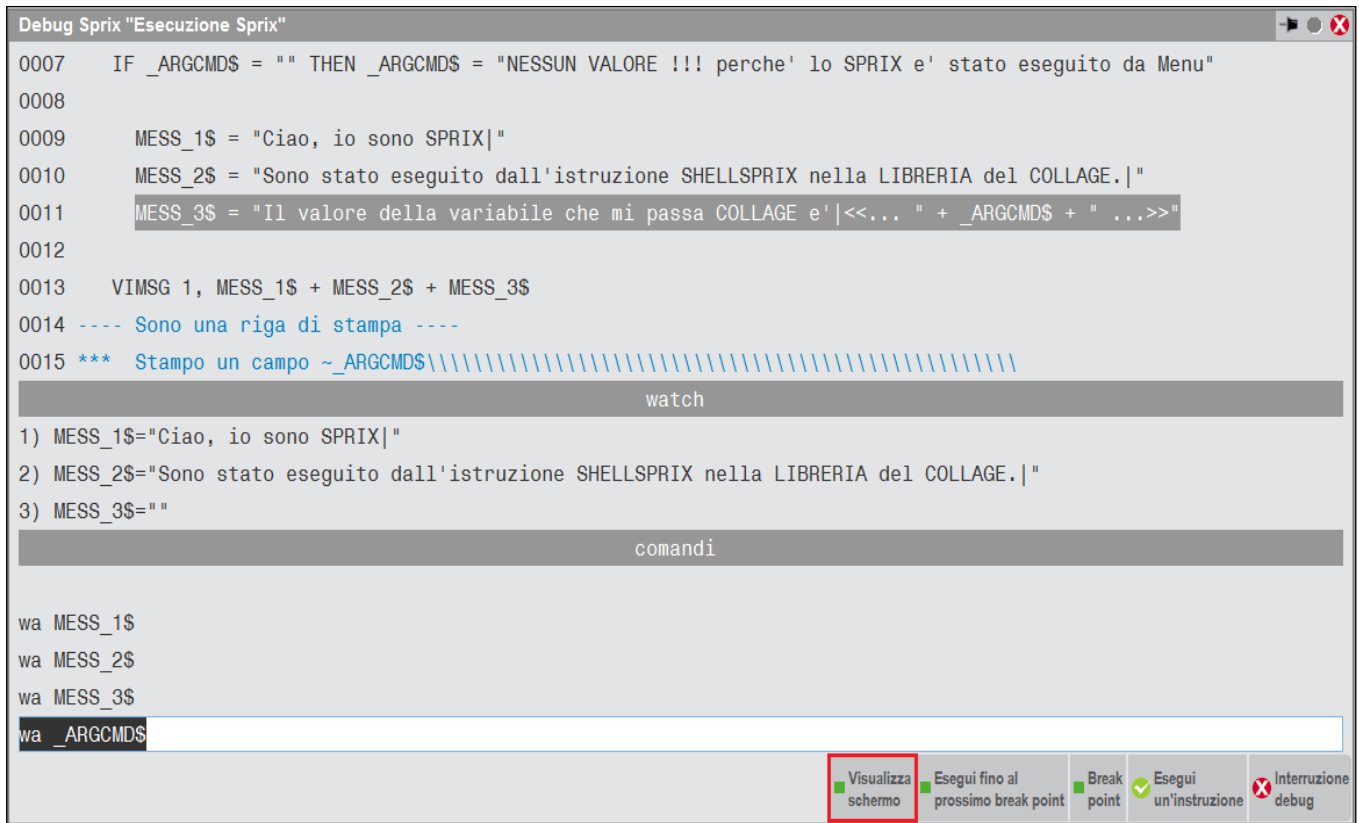


Il debug consente anche di processare l'eventuale parte di Libreria accodata al sorgente. La finestra di debugger ha due sezioni, una superiore con la porzione di codice scorribile e che visualizza in reverse quella che è prossima all'esecuzione, una inferiore per visualizzare i seguenti comandi debugger inseribili nel campo di immissione (e che sono richiamabili inserendo 'h' oppure 'help'):

'?' visualizza il valore attuale di variabile [se è un array l'elemento deve essere una costante, esempio: ? VAR\$(2)]
'wa' (watch) come '?', ma apre una sezione centrale nella finestra in cui numera fino a 7 watch e ne visualizza l'aggiornamento del valore ad ogni istruzione eseguita, esempio wa VAR\$(2)
'wc' cancella numero watch oppure '*' tutti i watch, esempio wc 1 oppure wc *
'bp' (breakpoint) imposta fino a 9 punti in cui l'esecuzione del programma si interrompe e viene aperta la finestra di debugger. I punti sono indicabili con etichetta seguita dal carattere ":" oppure con il numero di riga (esempio bp ETIC: oppure bp 194); il breakpoint visualizza in reverse i numeri di riga ed è attivabile/cancellabile con il pulsante "Break point" sulla prima riga visualizzata; con pulsante "Esegui fino al prossimo break point" si chiude il debugger ed esegue fino al breakpoint; il pulsante "Interruzione debug" interrompe e riapre il debugger (riapribile anche con la ripetizione del programma nel caso di "Esegui fino al prossimo break point" con assenza di breakpoint)
'bc' cancella 'bp', esempio bc ETIC: , bc 194 , bc * (* = tutti bp)
'bl' visualizza la lista di breakpoint (numeri riga o etichette)
'tl' visualizza la lista dei tasti in sospeso, in attesa di essere letti, inviati con PUTTAST

⁵ Il messaggio "l'eseguibile è in creazione o in esecuzione in un altro terminale" significa che non si può compilare in quanto il programma è in uso da un altro terminale; il messaggio "operazione non effettuabile. L'eseguibile è in uso da un altro terminale", significa che non è possibile accedere da un terminale al sorgente di un eseguibile in esecuzione su un altro terminale (è necessario disattivare l'eseguibile); il messaggio "operazione non effettuabile. Il sorgente è in uso da un altro terminale", significa che non si può accedere contemporaneamente allo stesso sorgente da due terminali differenti. In ambiente PassBuilder, Per applicare modifiche definitive all'App l'azienda deve essere ad utilizzo esclusivo del terminale, poiché il programma conosce le modifiche solo alla conferma delle variazioni, ma ciò richiede la certezza che nessun elemento dell'app sia in uso. Il terminale dedicato è richiesto anche quando l'App richiede allineamento dati di alcuni suoi elementi (es. MyDB) a seguito di aggiornamento archivi per nuova versione del gestionale. Ciò comporta che le modifiche e test vanno fatti con l'esecuzione di prova e sulla copia dell'azienda se sussiste il rischio di incidere sulla struttura dei dati.

La finestra del debugger copre l'esecuzione del programma, ma questa è anche temporaneamente chiudibile con "Visualizza schermo" in modo da verificare il sottostante risultato a video, analogamente a quanto accade se il programma prevedesse una immissione da tastiera posizionando il cursore dal debugger alla finestra d'immissione (attiva).



Il debugger non solo consente di verificare la corretta esecuzione, bensì consente ai nuovi collaboratori di capire come funziona la sequenza del linguaggio Sprix: un ottimo metodo è quello di installare un programma funzionante ([esempio](#)) e seguirne passo a passo l'esecuzione in debug. Un assunto: la compilazione del codice scritto nell'editor ha "sempre ragione" e occorre cercare l'errore, sia logico che di sintassi; l'errore non è necessariamente relazionata alla riga che viene segnalata, poiché potrebbe essere solamente la risultanza di tutto quello che sta a monte nella procedura.

La compilazione può essere massiva su tutti gli elementi Sprix/Collage dell'App in PassBuilder (pulsante funzione tasto [F12] "Compila elementi sprix/collage" presente nell'elenco degli elementi); consente la compilazione globale anche in debug e se è attivo un filtro di ricerca estesa (v. campo di ricerca full text) avviene solo sugli elementi filtrati.

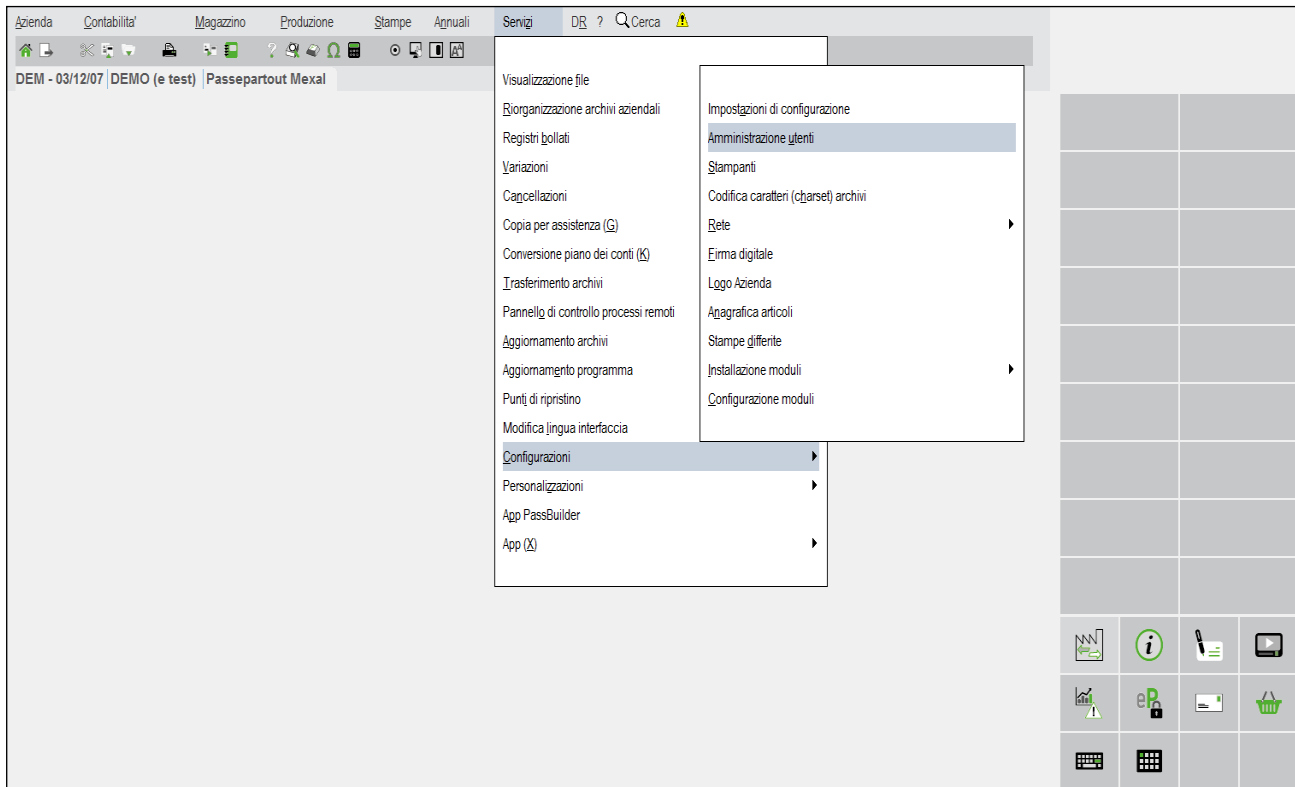
Attenzione: a seguito dell'introduzione della portabilità delle App tra versioni differenti del gestionale si è resa necessaria l'introduzione della retrocompatibilità dei programmi Sprix e Collage, altrimenti, una loro applicazione su versione più vecchia di quella che li ha compilati provocava il messaggio di errore per numero variabili system non valido (ricompilare). Questo comporta che l'esecuzione del compilato debug sia più rapida del compilato normale. Nel compilato debug viene creato un ambiente più grande che conosce già tutte le variabili mentre, invece, nell'eseguibile le variabili non sono più conosciute (come avveniva in precedenza) e quindi all'esecuzione vengono ogni volta inizialmente ricercate. Per ripristinare la velocità di esecuzione è necessario fare conoscere al programma le variabili coinvolte dichiarandole all'inizio sorgente (main), ponendo a zero (0) quelle numeriche e null ("") quelle alfanumeriche. Essendo queste variabili dichiarate all'inizio non incide sul programma che, invece, riceve il beneficio di conoscerle. Per esempio, nella lettura GETMULTILOT per tutti i lotti di un articolo, se vi fossero decine di migliaia di referenze, per ciascuna variabile coinvolta si avrebbe una quantità enorme di singole letture e con conseguente rallentamento nei tempi di elaborazione. In altri termini, per ogni informazione che viene letta l'ambiente deve decodificare il nome della relativa variabile, al fine di individuare quale sia la variabile in cui deve collocare il dato trovato. In questo esempio, per ciascuna _LOT???_M avviene l'allocatione per decine di migliaia di volte. Se all'inizio del programma, invece, si definissero le variabili coinvolte la compilazione le farebbe conoscere all'eseguibile il quale non le deve cercare poiché utilizza informazioni già conosciute.

Esempio: _LOTID_M(1)=0 : _LOTSTAT_M\$(1)=" : _LOTMESG_M\$(1)=" : _LOTMMSG_M\$(1)=" : _LOTTCCA_M\$(1)=" : _LOTCART_M\$(1)=" : _LOTCTPL_M\$(1)=" : _LOTCFOR_M\$(1)=" : _LOTDTC_M\$(1)=" : _LOTDIVA_M\$(1)=" : _LOTDTCR_M\$(1)=" : _LOTDTUM_M\$(1)=" : _LOTCTUTE_M\$(1)="

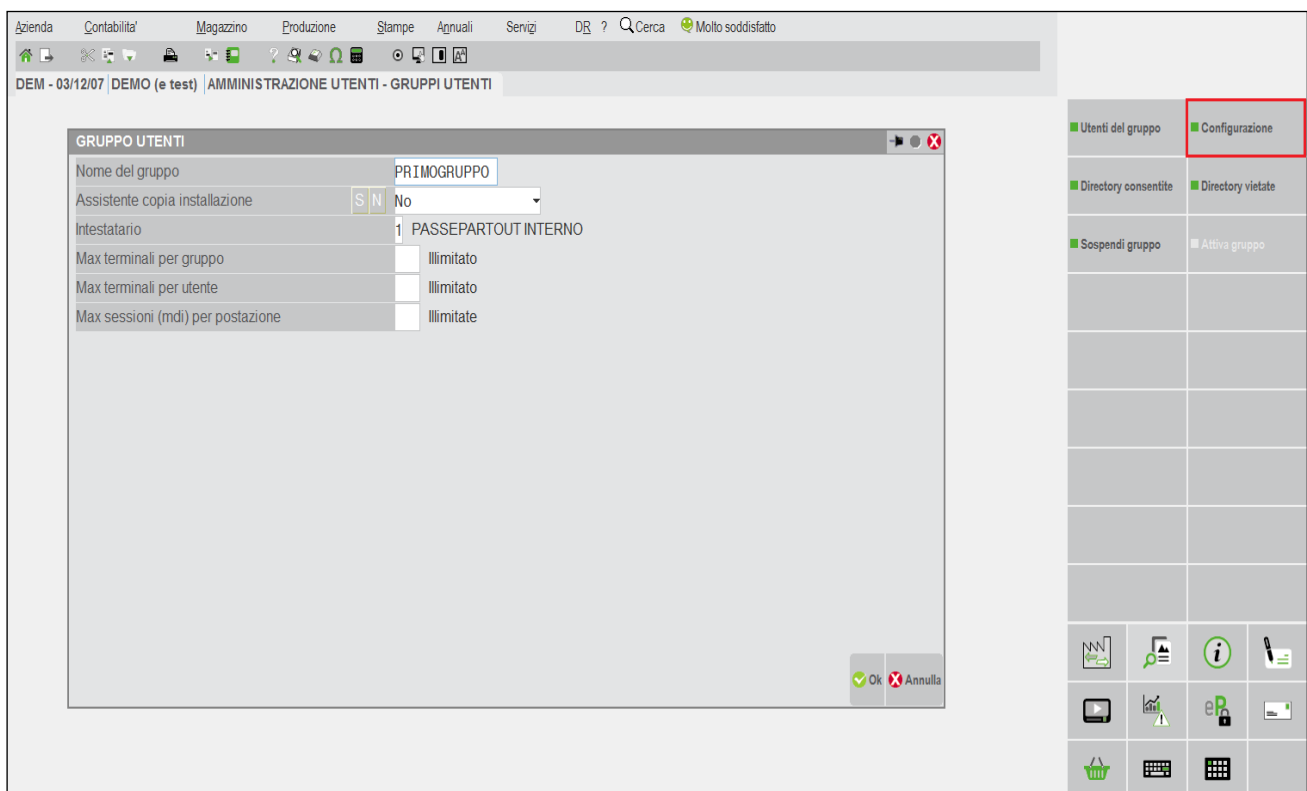
Può accadere, inoltre, che le App da installare presso il cliente possano rilevare il seguente messaggio (dove "xxxxxx.es" è il nome del file eseguibile Sprix o Collage): "Errore il file xxxxxx.es non è valido (errspr01). App non installabile". Le motivazioni possono essere le seguenti:

- la versione del gestionale del cliente è precedente a quella usata per sviluppare l'App. Soluzione: aggiornare il cliente.
- l'App non contiene i sorgenti ed il Cliente non ha attivato il modulo MDS. Soluzione: attivare il modulo MDS.
- l'App non è stata compilata correttamente in tutti i suoi componenti, potrebbero essere presenti delle compilazioni in debug. Soluzione: ricompilare i singoli sorgenti di cui è composta l'App.

Per concludere questa sezione dedicata all'Ambiente di sviluppo, descrivo le funzioni con le quali l'Amministratore può configurare "quale utente può eseguire cosa e/o come". L'Amministratore può accedere alla voce di menu "Amministrazione utenti", quindi, prestare attenzione anche alle informazioni del Manuale Generale dell'utente⁶.



L'Amministratore accede alla configurazione del gruppo o dei relativi utenti, per selezionare una serie di parametri disposti su più pagine.



⁶ nel **Manuale Generale dell'utente** sono presenti molti argomenti che riportano informazioni per l'ambiente Sprix, in particolare le **Nozioni Generali** (es. componente client / server, configurazione utenti e terminali, rdbms e operazioni batch), l'**Azienda** e **Servizi, Configurazioni, ...**

Si dispone di un parametro per visualizzare un'interfaccia del Passepartout "vecchio stile", in modo da consentire l'esecuzione di Collage preesistenti alla nuova interfaccia, poiché il Collage lavora principalmente sulle condizioni a video.

Configurazione gruppo PRIMOGRUPPO

Parametro	Valore	Stato
Avvio		
Livello di gestione	Predefinito	Da richiedere
Data di partenza	Predefinito	Ultima usata
Menu' di partenza	Predefinito	Nessuno
Azienda di partenza	Predefinito	Nessuna
Sottoazienda di partenza	Predefinito	Prevalente
Aspetto		
Aspetto interfaccia	Predefinito	Per questa postazione di lavoro
Lingua interfaccia	Predefinito	Standard
Modalità compatibilità collage	Predefinito	No
Accesso emis./revis. doc. a lista	Predefinito	Si
Griglia finestre elenco	Predefinito	No
Configurazioni		
Operatore predefinito	Predefinito	1
Stampante predefinito	Predefinito	Video
Categoria Docuvision	Modificato	2-PRIMO_1
Moduli		
AD Analisi di Bilancio	Predefinito	Si
AD Analisi Commerciale	Predefinito	Si

1/3

Ok Annulla

Si dispone di un parametro per consentire o non consentire al gruppo/utente l'esecuzione di determinati Sprix; nell'esempio è raffigurato il divieto (con il carattere meno "-") d'uso dello spx101.es/ss, spx102.es/ss, 103.es/ss. Per vedere quale utente utilizza uno Sprix è sufficiente premere in sequenza [F1+F1+F4] per accedere alla lista degli utenti che utilizza uno Sprix (e sono quelli con voce di menu composta dalla lettera di menu + "X").

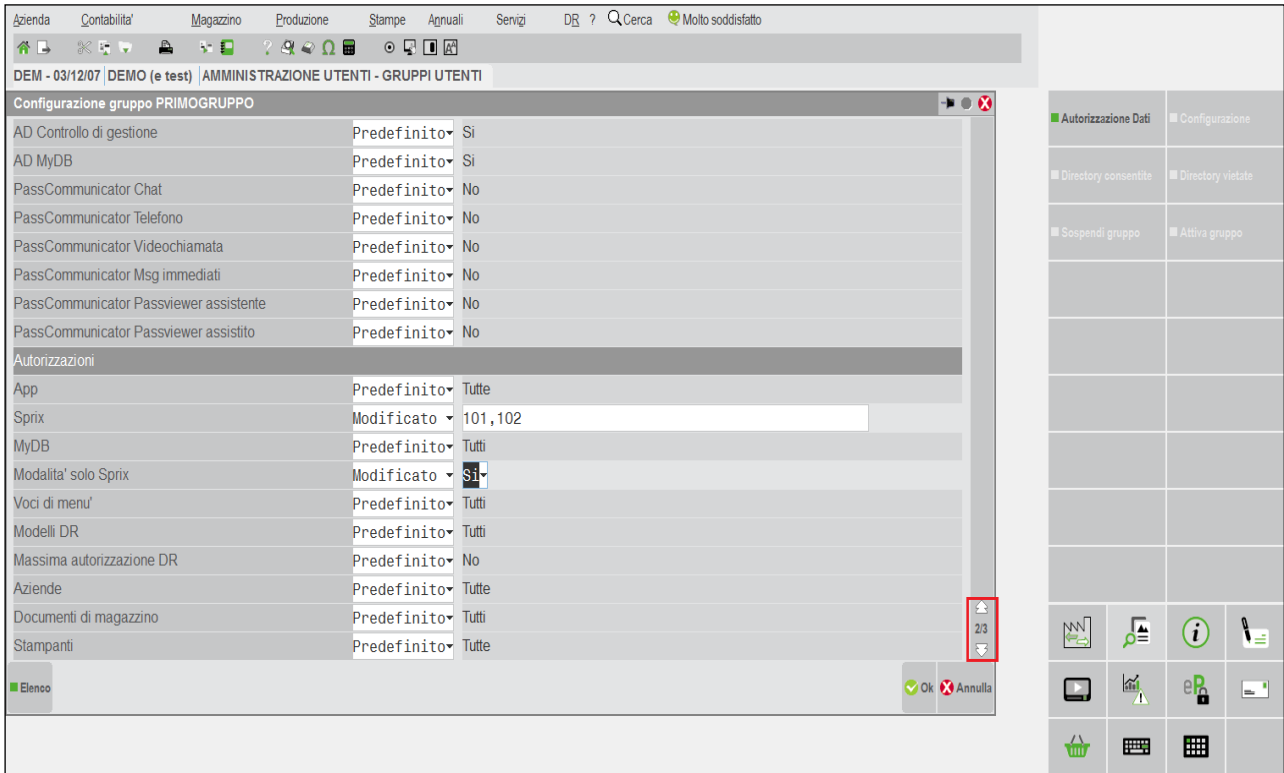
Configurazione gruppo PRIMO

Parametro	Valore	Stato
Autorizzazioni		
App	Predefinito	Tutte
Sprix	Modificato	-101,102,103...
MyDB	Predefinito	Tutti
Modalità solo Sprix	Predefinito	No
Voci di menu'	Predefinito	Tutti
Modelli DR	Predefinito	Tutti
Massima autorizzazione DR	Predefinito	No
Aziende	Predefinito	Tutte
Documenti di magazzino	Predefinito	Tutti
Stampanti	Predefinito	Tutte

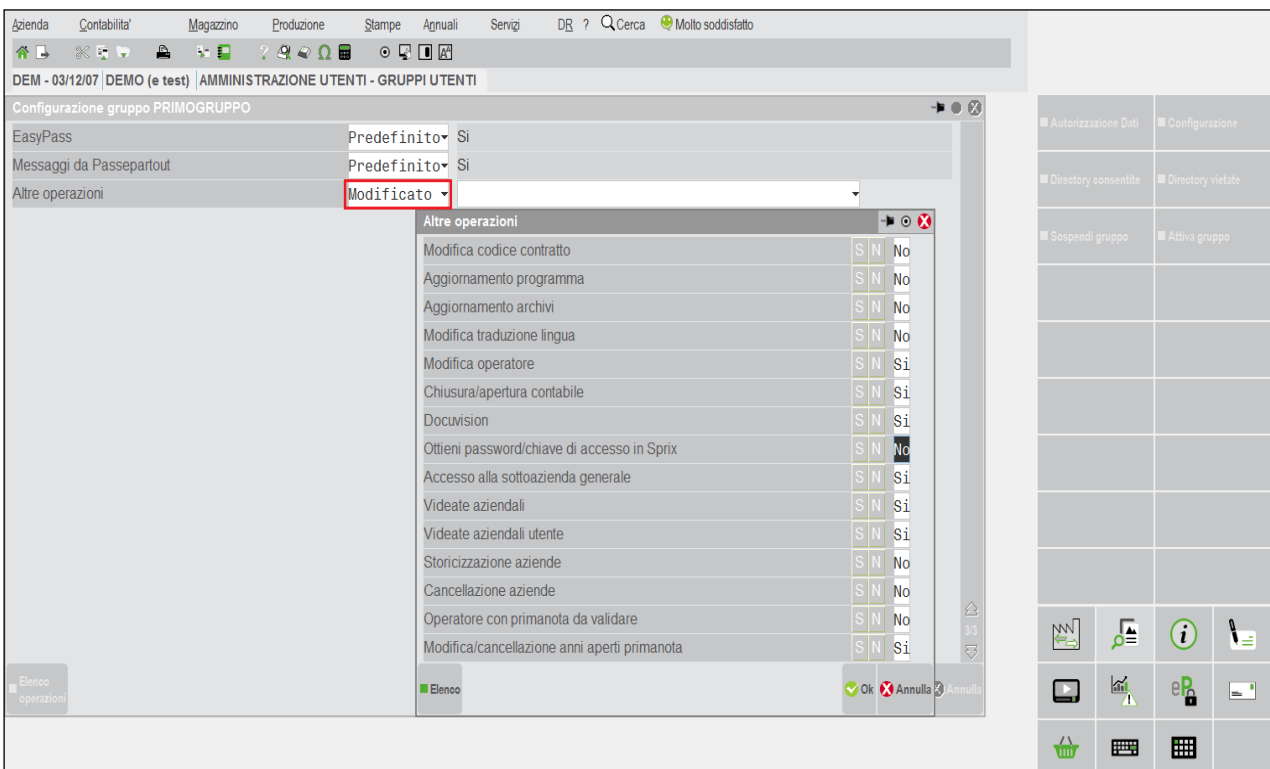
Tutti Nessuno

Ok Annulla

Si dispone di un parametro per imporre al gruppo/utente la sola esecuzione di taluni Sprix in una determinata Azienda⁷ senza poter entrare in altri menu, per esempio, i soli programmi spx101.es e spx102.es. La finestra consente di impedire all'utente l'accesso alle voci di menu e/o alle Aziende, ma a differenza di queste, l'impedimento ai menu non impedisce all'utente di eseguire i comandi Sprix relativi a quei menu (es. l'utente può registrare in magazzino con Sprix anche se non può accedere al menu magazzino), pertanto, l'eventuale impedimento all'utente va previsto all'interno del codice sorgente stesso. Questo esempio è la dimostrazione che Sprix/Collage, come già accennato in precedenza, non ha la stessa potenza e gli stessi controlli standard del Gestionale.



Si dispone di un parametro per impedire al gruppo/utente l'accesso all'ottenimento di talune "informazioni sensibili" attraverso lo stesso dizionario Sprix. In particolare, la condizione riguarda informazioni di sistema sulle password, qui indicate con le cosiddette variabili Dizionario: "sxpas", "akdpa", "akcpa", "akipa", "ajpas" ed eventuali altre simili.



⁷ La possibilità di disabilitare/vietare l'esecuzione di Collage al gruppo/utente è consentito tramite controlli nello stesso sorgente in relazione al dizionario "sxgrp" e/o "sxute"

INDICAZIONI PER LO SVILUPPO DI PROGRAMMI IN SPRIX E COLLAGE

Fatto salvo quanto espressamente richiesto per la realizzazione di PassApp attraverso l'ambiente di sviluppo PassBuilder (consultare la manualistica pubblicata in passstore.passepartout.net), si indicano alcune regole da rispettare nello sviluppo di personalizzazioni realizzate con Sprix e Collage.

1. Per ogni progetto è opportuno elaborare l'analisi tecnica del requisito richiesto prima di procedere allo fase di sviluppo.
2. Utilizzare la seguente nomenclatura delle versioni dei programmi realizzati:
 - 1.0: la versione iniziale;
 - 1.0a (b, c, d, ecc.): versioni ottenute a fronte di correzioni e/o modifiche di piccola entità; arrivati alla lettera "z" si prosegue con la versione 1.1a (b, c, d, ecc.);
 - si passa alla 2.0 per interventi di modifica al tracciato record o aggiunta di moduli;
3. Nei sorgenti non fare alcun riferimento alle videate aggiuntive e riportabili presenti nel gestionale, poiché non sono univoche per ogni azienda; bisogna quindi fare riferimento a file aggiuntivi oppure MYDB.
4. In tutti i campi presenti nella procedura deve essere gestito l'help in linea, da prevedere già in fase di analisi (es. per Sprix e Collage utilizzare la funzione "SETMAN").
5. Per quanto possibile tutte le elaborazioni devono essere eseguite in memoria (es.: array, buffer, ...) per evitare rallentamenti con continui "swap" su disco (evitare ove possibile l'utilizzo di file temporanei).
6. Qualora la personalizzazione richieda l'utilizzo di archivi aggiuntivi, devono valere le regole seguenti:
 - creare gli archivi aggiuntivi nella stessa cartella degli archivi dell'azienda o in una sotto-cartella della stessa; le cartelle accessibili sono state spiegate in precedenza (es. ... \datiaz\azi\);
 - gli archivi aggiuntivi devono avere il nome su disco lungo almeno 8 caratteri
7. Deve essere prodotto il Manuale del programma realizzato, esplicativo delle funzionalità implementate; il Manuale dovrà contenere anche le specifiche relative a:
 - eventuali file di "log", file ".ini" utilizzati dalla personalizzazione;
 - utilizzo eventuale dei comandi SHELL, DBMIRROR e collegamenti ODBC;
 - eventuali artifici utilizzati per superare limiti di Sprix e Collage e/o del Gestionale; segnalare anche gli eventuali limiti della personalizzazione;
 - gli eventuali protocolli e/o driver utilizzati e/o certificati;
 - gli eventuali dispositivi collegati a porte di comunicazione (registratori di cassa, lettori ottici per codici a barre, stampanti termiche, ecc..) comprensiva dei riferimenti del produttore, della versione e dell'anno di installazione;
8. Per ogni aggiornamento della personalizzazione occorre aggiornare il relativo Manuale di utilizzo.
9. I sorgenti delle varie personalizzazioni devono essere conservate sul Server aziendale (regolarmente sottoposto a backup) suddivise per Cliente.
10. Per ogni cliente con personalizzazione deve essere mantenuta traccia della versione della personalizzazione consegnata ed installata.
11. In tutti i sorgenti deve essere gestita la semaforica per evitare eventuali conflitti di accesso ai file suddetti; si presume che tutte le installazioni Mexal siano multi utente.
12. In eventuali messaggi presenti nella procedura si devono utilizzare i canoni di scrittura delle videate in stile Passepartout, quindi si devono utilizzare i colori appropriati per la visualizzazione dei messaggi a video. Esempio: in caso di errore la finestra deve essere di colore rosso, in caso di messaggio la finestra deve essere di colore verde, ecc..
13. La modalità operativa di gestione dei campi in Sprix e Collage deve essere la stessa del programma Passepartout. Esempio: in caso di variazione, inserimento o cancellazione di un dato, se questo è già presente, deve essere immediatamente visualizzato sul campo INPUT. Si deve quindi evitare la classica gestione dei menù inserimento variazione, ecc..
14. Tutte le variabili ed i flag utilizzati in Sprix e Collage devono essere dichiarati e commentati all'inizio del programma stesso in una sezione opportuna (Esempio: programmatore, data di creazione, data di ultima modifica, 'AMF=0' serve per ...; gestita nel paragrafo ...; azzerata da ...; sommata a; indicazioni sulle regole di semaforica utilizzate; note su variabili utilizzate delle librerie richiamate con "APPENDISPX").
15. Se la personalizzazione è composta da più Sprix/Collage, messaggi routine e/o procedure comuni devono essere scritti in Libreria.

COMANDI PER ARGOMENTO (Sintesi, Esempi e Casi)

LINGUAGGIO SPRIX E COLLAGE

La sintassi del **linguaggio** Sprix/Collage è un codice basico che viene elaborato in modo sequenziale riga per riga, composto da una sequenza di porzioni di programma (**routine**).

La sequenza di esecuzione delle routine è indirizzabile con la dichiarazione di **label**, ovvero libere etichette di max 29 caratteri maiuscoli terminati dal carattere ":", esempio ETIC1:, INIZIO:, RIPETI:, FINE:). La label non è duplicabile, definisce una routine ed è da chiudere con il comando END.

La label consente di accodare codice di **libreria** al sorgente in uso: la routine di libreria deve iniziare con una label eseguita con il comando GOSUB presente nel programma che la richiama, ed infine, terminare con il comando RETURN che restituisce il controllo al programma che la richiama. Per la stessa ragione è bene terminare anche lo Sprix chiamante con END.

In ciascuna **riga operativa** (cioè non un commento o una stampa) puoi inserire **comandi** (istruzioni, funzioni, variabili, costanti, espressioni, ecc.) interpretati dal sorgente in sequenza, quando separate dal carattere ":".

La rappresentazione di un valore avviene tramite una **variabile**, utilizzabile al momento opportuno senza doverla dichiarare all'inizio codice del programma. La variabile, può infatti, assumere valore stringa o valore numerico, entrambi anche sotto forma di array.

La variabile **stringa** ha un nome che termina con "\$", di max 12 caratteri A-Z maiuscoli e 0-9, non può iniziare con "_" o numero, non può essere un nome di un comando riservato al linguaggio. Il suo valore alfanumerico si indica con max 255 caratteri, fra doppi apici. Esempio: ABC_123\$ = "Sprix:1". Alloca memoria di 4 byte + la lunghezza della variabile; se non è inizializzata ha valore null (A\$ = "").

La variabile **numerica** ha un nome come la stringa, ma non termina con "\$" e il suo valore è un numero reale positivo o negativo, di max 6 decimali separati da un punto e senza arrotondamento della parte che è eccedente all'ultimo decimale stabilito. Esempio: A = 2345.61 : B = -12 : NUM_01 = -45.123456. Alloca memoria di 8 byte; se non è inizializzata ha valore zero (N = 0).

La variabile **array** è un vettore multi-dimensione composto da **elementi**, ovvero valori in sequenza del medesimo tipo (stringa/numero), ciascuno dei quali è una variabile accessibile tramite l'**indice** relativo all'elemento. Consente fino a matrici tridimensionali, dispone di comandi per dimensionare (DIM) e azzerare l'array (AZZARRAY). Le variabili numeriche e stringa, quindi, possono anche essere in forma di matrice da una, due oppure tre dimensioni da dichiarare tramite il comando DIM seguito dal nome della variabile e fra parentesi tonde ciascuna dimensione separata da virgola e la quantità di elementi (numeri interi costanti) di ciascuna dimensione. Se la prima dimensione è lasciata vuota, si assume come dimensione "automatica", consente risparmio di memoria, ma con minori prestazioni.

	una dimensione	due dimensioni	tre dimensioni
Non automatici:	DIM N(n1) DIM A\$(n1)	DIM N(n1,n2) DIM A\$(n1,n2)	DIM N(n1,n2,n3) DIM A\$(n1,n2,n3)
Automatici:	DIM N() DIM A\$()	DIM N(,n2) DIM A\$(,n2)	DIM N(,n2,n3) DIM A\$(,n2,n3)

Si accede a ciascun elemento tramite **Indice** numerico (costante o variabile), esempio: DIM A\$(,5) : I=4 : A\$(I,3)="spx" : AZZARRAY A\$

Esistono dei limiti dimensionali: fino a un massimo di 262080 per l'array stringa e di 131040 per quello numerico, entrambi ricavati dal prodotto degli elementi di ciascuna dimensione. Nota: essendoci limiti dimensionali, quindi, se questi non vengono rispettati rispetto alle dimensioni dichiarate o si impiegano indici nelle variabili non congruenti alle dimensioni dichiarate e/o consentite, si provocano messaggi di tipo "Indice dell'array fuori dai limiti". Se la dimensione non è automatica, ma costante, nella dichiarazione della dimensione non si possono utilizzare valori superiori a 32767 elementi in quanto non esegue la compilazione e dichiara "costante numerica troppo grande per la dimensione". (vedere DIM)

La **variabile** può assumere un valore assegnato costante [es. V = 3] oppure un valore diverso a seconda delle circostanze, sia valorizzandola con gli archivi di **struttura** [es. la descrizione del conto X del piano dei conti DES\$ = _PCDES\$], sia valorizzandola con archivi **dizionario** [esempio: DES\$ = pcdes(X)], sia valorizzandola con la risultanza di un'**espressione** di valori omogenei (alfanumerici o numerici) che fra essi combinati danno origine a una espressione numerica o alfanumerica; Es: N = 6*(18+20) : C\$ = A\$+B\$ (dove A\$ = "stringa 1" : B\$ = "stringa2"). Devono essere omogenei altrimenti non si combinano, salvo che siano **"trattati"** con opportuni comandi per renderli tali (es. conversione da numero in stringa o viceversa), compreso i valori di data e tempo.

Un consiglio generale: le Istruzioni/Funzioni che consentono di leggere/scrivere/cancellare singoli record negli archivi (esempio un documento) dispongono di parametri utili per individuare il record in oggetto; il parametro che rappresenta la chiave ("Id record") può essere di semplice (es: un numero) oppure articolato (es: una stringa concatenata di più elementi, cioè, sigla documento, serie documento, numero documento, classe documento, ecc...); in quest'ultimo caso, per individuare con precisione il record si rende utile/necessaria l'attribuzione al parametro della chiave concatenata tramite una **variabile d'appoggio**. Esempio: il Gestionale registra il documento di magazzino (es. record fattura) e ne archivia il relativo documento nel documentale Docuvision (es. record file Xml), creando quindi un'associazione fra i due record presenti su due diversi

archivi (quello del magazzino e del documentale); supponiamo di cancellare il solo record del documentale tramite DELDVDOC (acronimo dell'istruzione "Cancella Docuvision Documento"), ma per farlo occorre anche conoscere la relativa associazione tramite GETASSOCDV (acronimo dell'istruzione "Leggi Associazione Docuvision"). La GETASSOCDV dispone dei seguenti parametri [**<tipo>**,**<keypass>**,**<iddoc>**], in cui il parametro **<tipo>** rappresenta "il tipo di documento" (fattura di magazzino, cioè di tipo 1400), il parametro **<iddoc>** rappresenta lo "ID del documento Docuvision" ed infine **<keypass>**, il parametro che rappresenta la chiave della associazione ed è così composta: "**<sig\$>**;**<ser>**;**<num>**;**<clifor\$>**;**<ddoc\$>**", ovvero **<sig\$>** (sigla documento); **<ser>** (serie del documento); **<num>** (numero del documento); **<clifor\$>** (codice conto intestatario); **<ddoc\$>** (data documento). Ebbene, tutta questa concatenazione di parametri deve essere riassunta in un'unica variabile stringa di appoggio, es. VAR\$, da passare alla GETASSOCDV. Esempio: fattura emessa, sezionale 1, numero documento 10, cliente 501.00005, del primo marzo 2019. **VAR\$**="FT;1;10;501.00005;20190301" oppure

VAR\$=_MMSIG\$ +";"+ STR\$(_MMSER,0,0)+";"+STR\$(_MMNUM,0,0)+";"+_MMCLIS\$+";"+_MMDAT\$: GETASSOCDV 1400,**VAR\$**,0

Il valore di un'**espressione** è il risultato della combinazione di costanti e/o variabili tramite **operatori** (aritmetici, di relazione, logici), che hanno questa priorità di esecuzione decrescente:

NOT, - (*negativo*), ^, *, /, MOD, +, - (*sottrazione*), <, <=, >, >=, =, <>, **AND**, **OR**, ().

Quelli **aritmetici** su valori numerici: il meno unario (*negativo*) con singolo operando (-123) ha priorità rispetto al meno (*sottrazione*) con più operandi (14-20 = -6, C = 20-6, D=A-2-1, ecc); elevazione a potenza "A" con esponente tra 1 e 32.767 e intervallo di risultato +/- 999999999999 (3^4 = 81, E=10^2, F = E^2, ecc); moltiplicazione "*" (3*7 = 21, C = 10*2, D = B*2, ecc); divisione "/", moltiplicato zero è = a +/- 999999999999 (18/3 = 6, A=10/2, B=A/2); "MOD" resto della divisione e valore degli operandi compreso nell'intervallo +/- 2147483647 senza decimali (18 MOD 5 è = al resto di 18/5 con risultato 3, G=100 MOD 3, H=11 MOD 2, I=E MOD 2); somma "+" (2+3 = 5, A=10+4+3, B=A+1). L'espressione consente **concatenazioni** di stringhe costanti e/o variabili (A\$="Pass", B\$="Store" la concatenazione A\$+B\$+" 2016" = "PassStore 2016"); le parentesi "()" consentono d'invertire l'ordine d'esecuzione che, eccetto l'unario (-123), a parità di livello è da sinistra a destra, mentre l'assegnazione e/o la relazione (=) si raggruppano da destra. Esempi di espressione e relativo equivalente con le parentesi:

saldo - capitale - interessi	((saldo - capitale) - interessi)
saldo + -capitale + interessi	((saldo + (-capitale)) + interessi)
tasse - 10/100 * saldo	((tasse-10/100)*saldo)
primo = ultimo = 0	(primo = (ultimo = 0))

Gli operatori di **relazione** confrontano valori omogenei (stringhe/numeri) ottenendo un numero true (1) or false (0) che condiziona un comando IF. Il confronto fra le stringhe è in ordine alfabetico ("aa" < "b"), la più corta nell'intera misura è minore ("a" < "a "), infine, le minuscole > di maiuscole > di numeri. Gli operatori di relazione sono:

< (minore di)	IF X>60 AND Y<20 THEN Z=X*Y
<= (minore o uguale a)	C=4: IF H<=3 THEN Q=10*3
> (maggiore di)	IF A>10 OR B<0 THEN X=A/2
>= (maggiore o uguale a)	Y=3: IF X>=3 THEN Z=10*3
= (uguale a)	assegnazione a una variabile

Gli operatori **logici** confrontano relazioni, analogamente a come fanno gli operatori di relazione, per prendere una decisione. Sono:

NOT [che significa "non", esempio IF NOT (P=-1) THEN ...], **AND** [che significa "e"] e **OR** [che significa "oppure"]].

Se il comando prevede l'indicazione di parentesi "()" – obbligatorie – significa che è una funzione o una variabile array. Il comando possiede **parametri** di tipo **stringa\$** o **numerico**. Se nel manuale sono indicati fra parentesi "[...]" significa facoltativi, diversamente, se indicati fra "<...>" (oppure non sono previsti altri simboli) significa che sono obbligatori e se il loro valore è null oppure zero vanno comunque indicati con "" oppure 0 a seconda che siano stringa o numero. Esempio: date delle variabili (A\$="Ab12" ; A\$="" ; N=68 ; N=0) come parametri del **COMANDO** **<par\$>**, [par]: COMANDO A\$: COMANDO "Ab12" : COMANDO A\$, N : COMANDO "", 68 : COMANDO "Ab12", 0.

ETICHETTE COLLAGE

Il Collage è un programma che funziona a eventi, esegue e/o blocca le sue routine di codice in base a **etichette** presenti nel Passepartout, ma non può intervenire in un programma Sprix; Collage interagisce in base a determinati accadimenti nell'utilizzo di procedure del Passepartout, pertanto, lo sviluppo di un programma Collage richiede accurata conoscenza delle procedure Passepartout da modificare. Il Collage utilizza il linguaggio Sprix, ma possiede specifici comandi in quanto non dispone di tutti quelli che sono presenti in Sprix (per esempio [F6] Parametri di Esecuzione). Si ricorda, comunque, che Collage può eseguire uno Sprix, come anche uno Sprix può eseguire uno Sprix.

A differenza di Sprix il Collage:

- non accede direttamente alle strutture degli archivi dati in lettura, scrittura e cancellazione;
- accede in lettura e/o imputazione⁸ dei soli dati "caricati a video" nel momento in cui viene eseguito il relativo comando;
- non esegue letture sequenziali degli archivi dati nelle tabelle "dizionario" (il cosiddetto **ciclotabellare**), ma anche lui, come Sprix, può leggere il dizionario in modo diretto tramite parametri chiave del singolo record, quando consentito dal tipo dato; non disponendo del ciclo le correlazioni fra i dati possono essere eseguite con appoggio su file esterno popolato da Sprix eseguito tramite SHELLSPRIX.

Per quanto riguarda il riconoscimento di eventi e la formazione di etichette, le immagini dimostrative dell'esempio introduttivo nel manuale sono sufficienti a capire il contesto di quanto viene qui in seguito descritto. Altre immagini sono nel gruppo dei comandi per le stampe.

ATTENZIONE: quanto viene descritto in seguito si riferisce alla gestione di Collage precedente alla gestione delle App in PassBuilder e viene conservata per descrivere alcune logiche dell'ordine di sequenza di esecuzione tutt'ora valide per i Collage già consolidati. In PassBuilder l'ordinamento dei Collage in presenza di una sola App è in base al nome fisico del sorgente, se non è specificata alcun altro tipo di opzione. I Collage che vengono riportati in PassBuilder vanno rinominati come 001.ss , 002.ss , 003.ss, ecc. ed il nome del Collage deve rispettare l'ordine di esecuzione, oppure, se specificato l'ordine di esecuzione in base a tale ordine, nel nuovo campo presente dalla versione 2017J (come spiegato nella parte introduttiva di questo manuale)⁹.

Nella modalità precedente alla gestione di App in PassBuilder era presente la logica di sequenza di esecuzione basata sull'elenco dei Collage nell'elenco numerato del menu Collage.

Nell'elenco numerato del menu Collage, la prima sigla è il percorso della voce di menu a cui il programma si riferisce per eseguire le routine in esso contenute in base ad eventi (ETICHETTE) stabiliti nel codice stesso. Il Collage è caricato ed eseguito all'accesso della voce di menu a cui riferito. Premendo [CTRL+G] è possibile selezionare la funzione di "Stampa voci di menu", per esempio, la voce "CI" esegue routine in Contabilità – Immissione primanota.

Inserendo asterisco (*) nel campo del percorso di menu nell'elenco numerato dei Collage, significa che il Collage è riferito a tutte le voci del menu scelto, per esempio, (S*) riguarda tutte le voci del menu Stampe. Inserendo, invece, la parola (AUTO) significa che il Collage è installato ovunque sia consentito e si attiva al primo evento utile stabilito nel codice, ma questo, però, occupa sempre della memoria, per cui è da utilizzare solo quando è necessario.

In quasi tutti i punti del Gestionale viene effettuata la "chiamata" a routine Collage: se questa esiste viene eseguita, e quando termina riporta il controllo al Gestionale. Più Collage sono installabili sullo stesso menu e se contengono la stessa etichetta evento le rispettive routine sono caricate contemporaneamente sullo stesso punto, con priorità d'esecuzione a quella del Collage che nell'elenco è dichiarato prima degli altri Collage.

A parità d'etichetta su Collage diversi, il menu specifico ha la precedenza rispetto al programma Collage AUTO, anche se gerarchicamente di ordine inferiore nell'elenco dei Collage; l'ordine è talmente importante che l'elenco dispone di funzioni di "Cambia posto" fra i Collage e di "Attiva" e "Disattiva" Collage, marcando con carattere meno (–) quello disattivato. Selezionando nell'elenco la funzione "Percorso menu" si può modificare la voce di menu, ma deve essere mantenuta la coerenza con le etichette contenute nel programma.

Come già visto nel precedente esempio, selezionando "Informazioni su" dal menu di Passepartout si conoscono le informazioni sugli eventi ei Collage caricati: il motivo dell'attivazione di ciascun programma (da menu, da etichetta, da un altro Collage), lo stato del Collage attivo che rispetto alla specifica etichetta collegata allo specifico campo di input è considerato in memoria oppure ancora da caricare oppure disattivato da una specifica istruzione.

⁸ In Collage le variabili delle strutture archivi sono accessibili in sola lettura. Non è possibile impostare una variabile (ad esempio _AR???=VALORE\$ per anagrafica articoli) in quanto questa non viene memorizzata. L'unico modo per modificarne il valore originario è attraverso l'input automatico eseguito con apposite istruzioni e variabili, per esempio: PUTTAST per eseguire input da tastiera (per scrivere o spostarsi fra i campi); _SINP\$ per inserire il valore d'interesse nel campo attivo in quel momento (_SINP\$=VALORE\$).

⁹ Ordine di caricamento, esecuzione, collage in App

L'ordinamento dei Collage è dinamico, dipende da come è definito (AUTO, puntuale sul menu, ecc..) e dall'evento che richiama (ON_OUT_ALL, ON_OUT_nomecampo, ecc...). I collage vengono eseguiti in base a una serie di fattori che incidono non sull'ordinamento di esecuzione dei collage, ma sull'ordinamento di esecuzione del singolo evento, che prescinde dall'ordine di inserimento del collage stesso. I fattori che entrano in gioco sono:

se il collage è definito AUTO, oppure se è configurato per uno specifico menu (es. CK);

se l'evento è specifico (ON_IN_nomecampo) oppure generico (ON_IN_ALL);

se sono presenti collage disabilitati;

se vengono fatti partire dei collage con le istruzioni START e STOP COLLAGE e con che ordine;

dall'ordinamento alfabetico del codice dell'APP

dalla posizione specificata sul collage all'interno dell'app

Tutti questi fattori, alcuni presenti anche prima della nascita di PassBuilder, rendono non determinabile a priori l'esatto ordine con cui vengono eseguiti i collage e impossibile anche poterli pilotare manualmente. Con più APP, tale criterio non è applicabile.

Più Collage contemporaneamente attivi sulla stessa sessione di più terminali in rete, richiede opportuna gestione di semafori (blocchi) in caso di accessi ad una unica risorsa (file, porta COM, ecc.) i file dei Collage attivi sullo stesso menu sono letti dal alto verso il basso: l'accesso allo stesso dato comporta che, se in un determinato evento fosse eseguita una determinata operazione nel primo file, l'operazione eseguita precedentemente nel medesimo evento potrebbe essere persa nel successivo file.

Ciascuna routine Collage deve essere inizializzata con una cosiddetta ETICHETTA e terminata con il comando END. Alla chiamata della routine alcune variabili di sistema sono automaticamente settate (_SINP\$, _TF, _NCAM\$, _NFIN, _DES612\$, ecc.).

La sintassi dell'etichetta evento per l'esecuzione della routine in essa contenuta, si forma con modalità **predefinite, personalizzabili** e di **stampa** (queste ultime sono descritte nella tabella dei comandi raggruppati per argomento dedicata alle stampe). Per i dettagli sulle etichette si rimanda a ciascuna voce dell'elenco alfabetico dei comandi.

Le etichette **predefinite** sono: ON_START: (avvio della voce di menu), ON_END: (termine della voce di menu), ON_BEEP: (emissione di un suono). Le etichette predefinite relative agli archivi dati: ON_START?: (all'ingresso), ON_PUT?: (alla scrittura), ON_DEL?: (alla cancellazione), ON_END?: (all'uscita), in cui sostituire il carattere ? con le sigle dei moduli che gestiscono gli archivi dei dati.

Indichiamo solo alcune fra le sigle fra più utilizzate:

- PC (modulo piano dei conti)
- PN (modulo primanota)
- AR (modulo anagrafica articoli)
- MM (modulo emissione revisione documenti di magazzino)
- BL (modulo bolla di lavorazione)
- ecc...

Esempio, gli eventi del modulo piano dei conti sono ON_STARTPC:, ON_PUTPC:, ON_DELPC:, ON_ENDPC:. **Notare: a seconda dei periodi storici di implementazione possono esistere anche delle eccezioni a questa regola, ma debitamente documentate (vedere i comandi per argomento relativi all'anagrafica contatti).**

I tasti che attivano un determinato evento variano a seconda del modulo e delle circostanze dell'operazione; per esempio, l'uscita può essere effettuata con tasti uscita [Esc] oppure [Fine], tasto ok [Invio], tasto conferma [F10], ecc..., ma non con il cambio di menu [ShF10] !

Generalmente negli eventi PUT e DEL la semaforica è attiva ed il record è fisicamente già aggiornato, mentre in END la semaforica sul record è già disattivata; queste regole non sono universali, devono essere sempre verificate: esempio, ON_ENDMM in emissione documenti di magazzino scatta prima dello sblocco semaforica sul documento e quindi, altre operazioni lanciate al ON_ENDMM sullo stesso documento trovano il blocco dell'input da tastiera.

Le etichette **personalizzabili** si formano con un CODICE campo, che si riconosce selezionando **"Informazioni su"** dal menu del Gestionale, come visto in precedenza nell'esempio introduttivo. Queste informazioni riportano anche i possibili eventi: ON_IN_CODICE: (ingresso campo), ON_OUT_CODICE: (uscita campo), ON_ERR_CODICE: (alla visualizzazione di errore sul campo) e ON_612_CODICE: (alla pressione del tasto [F12] sul campo); un esempio di etichetta personalizzata sul tasto funzione [F12] è stata dimostrata nel precedente esempio. Esiste, inoltre anche la variabile Collage _NCAM\$ che contiene il CODICE del campo in cui si trova il cursore in quel momento. Sostituendo il CODICE con ALL, significa "evento su tutti i campi": ON_IN_ALL:, ON_OUT_ALL:, ON_ERR_ALL:, ON_612_ALL:. Con la variabile _DES612\$, infine, si può stabilire la descrizione del tasto [F12].

L'evento Collage associato ai tasti funzione riguarda il solo tasto funzione [F12]. Se si crea una etichetta "ON_612_[nome campo]:" e si digita [F12] quando il campo stesso è attivo il Collage sarà attivato. Di seguito è mostrato un esempio che mostra l'uso di F12 esteso a tutti i campi della videata che si vuole personalizzare. ON_612_ALL: VIMSG 1,"COLLAGE ATTIVATO!" : END

Notare Bene: l'evento predefinito potrebbe non manifestarsi necessariamente da tutti i punti del gestionale che ne consentono l'operazione associata (per esempio l'apertura di una finestra elenco da Magazzino piuttosto che la stessa finestra elenco, ma da Produzione). Altri, invece, richiedono l'uscita dal primo campo di input. Altri ancora come l'evento ON_ENDxx possono non essere eseguiti se si abbandona la procedura per passare ad una diversa azienda (CTRL+A) o si cambia programma da menu (SHIFT+F10). Le tipologie di evento e le condizioni dei dati a video in quel determinato momento condizionano grandemente le funzioni di lettura dei dati a video. Nelle annotazioni presenti nelle tabelle dei comandi raggruppati per argomenti relative ai moduli del Gestionale, possono essere riportate alcune casistiche sugli eventi più rilevanti; uno dei più importanti è collocato nel gruppo di comandi per la gestione del Documento di Magazzino, poiché in questo contesto è particolarmente evidente. [Clicca qui](#)

Approfondimento: l'utilizzo dei CODICI campo per la personalizzazione di etichette non impedisce che durante la gestione di un archivio si possa accedere anche ai dati di un altro archivio. Eccetto la possibilità consentita dall'utilizzo della SHELLSPRIX (per lanciare elaborazioni Sprix su altri archivi), in Collage è possibile accedere ai dati di un archivio pur eseguendo routine attivate dalle etichette di un altro archivio. Durante una registrazione è possibile, quindi, ottenere informazioni appartenenti ad archivi differenti fra loro. Per esempio, da Emissione Revisione Documenti - codice campo con prefisso 'MD' - si desidera conoscere i dati appartenenti all'anagrafica piano dei conti Clienti/Fornitori - 'PC' -, come la valuta, il fido, ecc. L'importante è avere a disposizione i parametri necessari alla ricerca in questi archivi e che possono essere ottenuti in vari modi. Un esempio, potrebbe consistere nella ricerca della valuta del cliente e valutare se è l'EURO, per cui, dato il codice conto del cliente (letto con _SINP\$ in fase di input del campo e quindi nell'evento ON_IN_MDRICOD:) ed attribuito ad una variabile stringa CONTO\$

è possibile risalire alla valuta presente nell'anagrafica di quest'ultimo (esempio: IF mutip(pcval(CONTO\$)) = "E" THEN...). Un ulteriore esempio, potrebbe consistere nella visualizzazione del fido del cliente: all'evento ON_OUT_MDRICOD: leggiamo il documento che stiamo scrivendo con GETATTMM e valorizziamo una variabile stringa, con il codice cliente presente nell'apposita variabile struttura archivi A\$=_MMCLI\$. Tale variabile stringa è utilizzabile come parametro da utilizzare nella specifica variabile dizionario del Piano dei Conti relativa al fido, in modo da valorizzare una variabile numerica contenente il fido (B=pcfid(A\$)); provvederemo infine a visualizzare questa variabile numerica (LOCATE 10,10 : PRINT STR\$(B,0,0)) ed terminare la routine (END)

Si consiglia di debuggare sempre per verificare le sequenze corrette o le ridondanze cicliche o per sfruttare eventi precedenti e/o successivi a quelli non utilizzabili; le sequenze degli eventi dipendono anche dal tipo di modulo e le eventuali opzioni e parametrizzazioni che nel Gestionale sono consentite. **Nota:** se il CODICE del campo non è previsto¹⁰ oppure non è in maiuscolo oppure non è relativo a un campo di input, l'evento potrebbe non esistere oppure essere riservato oppure essere vincolato¹¹. Esempio, non sono previsti degli eventi fino a che non si entra in un'azienda e la si conferma. Alcuni moduli particolarmente importanti come l'amministrazione della configurazione utenti (menu ZCU) non consentono di sviluppare dei Collage, ed il motivo è ben intuibile.

Per quanto riguarda l'impostazione delle tastiere personalizzate (tastiere c.d. "magic") nel gestionale Passepartout può essere impostata solo in presenza di campi input reali, cioè di campi in cui la procedura si aspetta la digitazione di valori da parte dell'utente. Le tastiere non possono quindi essere settate e di conseguenza utilizzate, in tutti gli eventi Collage in cui l'input non sia reale (ad esempio le finestre di ricerca articoli, di codici conto, ecc.)

Nota: le modifiche alle configurazioni del terminale e l'attivazione di determinate funzioni (specialmente quando nuove) possono influire sulla sequenza originaria stabilita nel codice del Collage, pertanto, potrebbe rendersi utile ripristinare la configurazione o adeguare il codice:

Esempio, in seguito ad ammodernamenti grafici all'aspetto del desk (estensione a wide screen, liste documenti, ecc.) si è reso indispensabile fornire anche dei parametri di configurazione utenti per consentire il mantenimento della compatibilità dei Collage (in stile tradizionale).

Esempio, l'attivazione o la disattivazione della funzione Ricerca Estesa nel documento di magazzino influenza il comportamento del tasto Invio, cioè determina l'apertura della lista di stringhe compatibili con il dato in Input oppure l'eventuale messaggio "Articolo non trovato".

In seguito a implementazioni possono nascere nuove etichette che sembra sostituiscano quelle già esistenti. In realtà si aggiungono in funzione dell'implementazione, mentre le precedenti mantengono la funzionalità se permangono le condizioni per le quali sono state utilizzate. Esempio: l'implementazione che prevede la contemporanea visualizzazione di alcuni archivi (anziché la visualizzazione in sequenza uno dopo l'altro come accadeva prima) ha richiesto l'aggiunta di alcune etichette; la procedura precedente, infatti, prevedeva la singola visualizzazione/selezione separata degli archivi delle vidate utente/aggiuntive con relative etichette (ZPVASELE da anagrafica articolo, ZPVCSELE da anagrafica conti e

¹⁰ Per gestire controlli Collage nelle finestre apribili tramite [F2], ma che non hanno CODICE, ovvero nelle finestre che si possono aprire tramite tasto [F2] ma che non hanno un'etichetta a cui agganciarsi (e che all'apertura fanno scattare l'evento ON_OUT_nomecampo, come ad esempio il tasto funzione [F2] sulla ricerca documenti da stampare in emissione differita documenti) si possono valutare alcune soluzioni alternative e che potrebbero rivelarsi utili anche in altri casi:

1) All'evento ON_IN_nomecampo (per esempio ON_IN_MEMODST) aprire una finestra "personalizzata" tramite APRIFIN e di selezione analoga a quella gestita dal programma, dove l'utente scorrendo i vari campi può scegliere il valore da passare al campo stesso (nell'esempio la sigla doc.); qui si possono inserire dei controlli (utili a pilotare/delimitare le scelte dell'utente); alla pressione dell'Invio sul campo di INPUT di appoggio al fianco di ogni valore di scelta (nell'esempio sopra di ogni sigla), si ricava il valore della variabile che si può passare al campo stesso tramite PUTTAST (più un Invio).

2) All'evento ON_OUT_nomecampo si può settare una variabile che dice se il campo è stato variato o meno; in questo caso al successivo ON_IN_nomecampo si può valutare se aprire o meno ancora la finestra personalizzata o passare al campo successivo con PUTTAST invio (rendendo così non possibile la scelta all'utente tramite [F2])

¹¹ Ulteriori considerazioni sull'intercettabilità degli eventi.

In passato c'era stato un periodo in cui si poteva intercettare anche l'uscita dal campo iva di riga del documento di magazzino, anziché, come di consueto dal campo sconto. Questo generò la necessità di scegliere un determinato tipo di configurazione attraverso il parametro "ModalitaAccessoCampolva" presente nella sezione [Impostazioni] del file di configurazione pxconf.ini che poteva influenzare l'avanzamento del cursore sulla riga del documento di magazzino (il default è = 1 e fa saltare a nuova riga dopo il campo sconto, se invece viene posto = 2 il cursore prosegue fino al campo del Valore Iva). Questa possibilità **non è più consentita** ed è rimasta in vigore la precedente operatività di default attraverso l'uscita dal campo sconto (MDSCON) anziché consentire la scelta sull'uscita dal campo Iva (MDALI).

Se l'evento Collage non è intercettato o la variabile non è valorizzata, occorre considerare alcuni aspetti relativi alle etichette. Determinati eventi non sono intercettabili se manca uno dei due elementi necessari ad attivare il trigger collegato: che esista il CODICE campo (verificabile selezionando "Informazioni su" dal menu del gestionale quando si è posizionati sul campo attivo) e che questo sia un campo di INPUT. Non è sufficiente che il campo sia elencato o scorribile/selezionabile a video; in tal caso è necessario avvalersi di altri eventi (precedenti, ad esempio sulla digitazione del tasto funzione che attiva l'evento in questione, oppure sui successivi se consentiti, ad esempio sulla digitazione del tasto che attiva l'uscita dall'evento stesso. Ad esempio alcuni eventi che non intercettabili sono: il menu aperto; il campo MDVISRDOD; la finestra degli ordini aperta in emissione revisione documenti o in anagrafica articoli [Sh+F7] (una soluzione consisterebbe nella ricostruzione di quanto le finestre consentono, attraverso la memorizzazione dei documenti su un file esterno, su cui effettuare interrogazioni al fine di attivare gli automatismi desiderati, VIMSG, LOCATE, ecc.); la prima finestra di accesso nel gestionale per l'attivazione immediata di Collage (Sprix e Collage si attivano solo dopo l'accesso in una determinata azienda, altrimenti ognuna non disporrebbe dei propri Sprix/Collage (l'ingresso nell'azienda o in un determinato menu è comunque automatizzabile attraverso la configurazione dell'utente). In alcune situazioni, per far scattare l'evento è necessaria un'ulteriore condizione: se il cursore non avanza almeno una volta dal primo campo in cui si trova quando si entra in una funzione del gestionale, non scatta l'evento ON_START e conseguentemente anche quelli successivi, ON_PUT? compreso; ad esempio, se in fase di revisione (non immissione) di un documento di magazzino, con il cursore non ci si sposta dal campo MDTDOC, l'evento ON_START non si attiva e conseguentemente, all'evento ON_PUTMM la variabile _NMM (numero righe corpo) non viene valorizzata da GETATTMM. In altri casi vi sono dei limiti ben precisi, ad esempio _PNPRO (importo complessivo provvigione) rimane vuoto all'ON_PUTPN su menu MD, se lo scadenario non è abilitato alla gestione della provvigione. La memorizzazione delle provvigioni viene gestita solo per determinate causali (es. FR,FE,NR,NE,RF), ci sono appositi controlli.

La condizione di cui sopra, può accadere soprattutto in caso di revisione di un documento e conferma immediata dello stesso; per aggirare tale condizione, potrebbe impostarsi il Collage in modo che si ripristini automaticamente, esempio:

```
ON_OUT_ALL:
IF X > 0 THEN END
IF _TF = 610 THEN PUTTAST "#501"
IF _TF = 610 THEN X=X+1
IF _TF = 610 THEN _TF=0
END
```

```
ON_IN_MDCAU:
IF X > 0 THEN _TF=610
IF X > 0 THEN PUTTAST "#610"
END
```

In altre determinate situazioni, la sequenza degli eventi può non essere quella che ci si aspetta; se il Collage è collegato ad una funzione che stampa modulistica, generalmente vengono eseguiti prima i trigger ON_PUT?? e poi i trigger ON_TST_MODULO_XXX??, ma se è collegato alla funzione Produzione/Bolle di lavorazione (BL) avviene il contrario.

Indichiamo, inoltre, un esempio di intercettazione del primo evento di ogni modulo e programma del gestionale, da installare nel percorso collage "AUTO":

```
ON_START:
F=1
END
ON_IN_ALL:
IF F=1 THEN VIMSG 4, "attivo": F=0
END
```

ZPVMSELE dai documenti); se non si selezionava alcuna videata, la procedura proseguiva ricercando fra gli archivi MyDB. L'implementazione della funzione di visualizzazione contemporanea delle videate e degli archivi MyDB (anch'essi visualizzati immediatamente), ha reso necessaria l'aggiunta di ulteriori etichette 'DATAAGG' (con assunzione di menu 'attuale', esempio, MADATAAGG in anagrafica articolo e CKDATAAGG in anagrafica conti). Questo è vero, però, solo se si definiscono anche archivi MyDB, poichè se non sono definiti (e sono definite le sole videate) continua a visualizzare la finestra di selezione in modo tradizionale e mantiene attive le precedenti etichette ZP? descritte sopra.

Possono presentarsi ulteriori simili condizioni, pertanto, per la creazione di codice Collage occorre considerare tutte le ipotesi d'uso da parte dell'utente e la scelta di determinate configurazioni, in particolare, le novità di versione. Passepartout ha da sempre operato per garantire la massima portabilità dei Collage, ma non può essere considerato un aspetto eternamente assoluto.

Collage può essere utile anche nei casi in cui non esistano funzioni Sprix per aggiornare Tabelle del Gestionale, ovvero, quando non sono disponibili le funzioni native in grado di farlo (nel Gestionale Sprix gestisce Anagrafiche e Archivi, non le Tabelle), lasciando quindi all'utente il popolamento manuale delle Tabelle; Collage, invece, è in grado di simulare questa procedura come una funzione di servizio.¹²

Dalla versione 2020A (v750) è consentito l'aggiornamento dell'App anche se i Collage sono caricati, ma non in esecuzione; eventuali variabili in memoria vengono sempre azzerate.

COLLAGE NELL'AZIENDA STD (azienda per le elaborazioni multiaziendali)

Lanciando dall'azienda STD un'elaborazione multiaziendale, i collage che vengono caricati sono quelli relativi all'azienda STD (compreso le voci sprix). I collage continuano a lavorare anche sulle aziende aperte, ma quello che è in esecuzione è pertinente all'azienda STD. L'apertura delle varie aziende potrebbe far supporre che i collage in esecuzione sono quelli delle singole aziende, ma in realtà sono quelli della STD anche se ci si trova su un'altra azienda (e che potrebbe non avere il collage in questione). Relativamente ai MyDB, sono disponibili i mydb della STD fino a che si resta nella medesima azienda, quando si passa ad un'altra azienda i MyDB non sono più disponibili.

12 . Esempio su tabella "Cambi valute" da 9 elementi: premendo [F12] su qualsiasi campo, apre la tabella del menu "APM" (Cambi valute), ne scrive il contenuto su 9 cambi, conferma e infine si posiziona sul menu di "Apertura Azienda". Il Collage potrebbe caricarsi i dati CAM?\$ leggendo un file esterno proveniente da siti di aggiornamento cambi.

ON_612_ALL:

ESECMENU("APM")

ON_IN_APMDCAM:

IF X=1 THEN END ELSE X=0

SETTF "501,505,610"

CAM1\$="1,20" : CAM2\$="1,30" : CAM3\$="1,30" : CAM4\$="1,40" : CAM5\$="1,50" : CAM6\$="1,60" : CAM7\$="1,70" : CAM8\$="1,80" : CAM9\$="1,90"

PUTTAST "#501" : PUTTAST CAM1\$: PUTTAST "#501" : PUTTAST CAM2\$: PUTTAST "#501" : PUTTAST CAM3\$: PUTTAST "#501" : PUTTAST CAM4\$

PUTTAST "#501" : PUTTAST CAM5\$: PUTTAST "#501" : PUTTAST CAM6\$: PUTTAST "#501" : PUTTAST CAM7\$: PUTTAST "#501" : PUTTAST CAM8\$

PUTTAST "#501" : PUTTAST CAM9\$: PUTTAST "#610"

X=1

ESECMENU("AA")

END

COMANDI GENERALI

Le seguenti tabelle raggruppano i **comandi Sprix e Collage per argomento**, specialmente quando sono riferiti all'architettura dei moduli, le funzioni e dati del Gestionale. Sono indicati parametri e vincoli per Livello/Configurazione (altre condizioni d'uso sono nell'introduzione di questo manuale) e la disponibilità in "S/C/Cr/Cd/Sm/Cm/Ca/Sh/Cc", cioè **Sprix (S)**, **Collage (C)**, **Collage Server Remoto (Cr)**, **Collage DataTrek (Cd)**, **Sprix Mobile (Sm)**, **Collage Mobile (Cm)**, **Collage Controllo App (Ca)**, **Shaker (Sh)**, **Customer Care (Cc)**.

Notare bene: gli esempi ed i casi sotto riportati sono esclusivamente a fini dimostrativi e non costituiscono in alcun modo del codice rivendibile o liberamente installabile e la cui applicabilità è da valutare in relazione alle circostanze dell'installazione (la versione e la configurazione del gestionale), in quanto potrebbero risultare non aggiornati o compatibili.

Generali		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
#APPENDISPX	Par: <nomefile\$1> [<nomefile\$2> <nomefile\$3>...]. Istruzione che consente di riutilizzare routine codice di Libreria, appese e compilate in coda al codice dell'eseguibile ".es". es.	S/C/Cr/Cd/Sm/Cm
ON_START:	Evento richiamato all'avvio del modulo; consultare Linguaggio Sprix/Collage (etichette);	C/Cr/Cd/Cm
REM o apice (')	Istruzione che trasforma tutta la riga in un commento; l'apice consente di commentare la riga partendo da una qualsiasi posizione sulla riga. es.	S/C/Cr/Cd/Sm/Cm
IF - THEN - ELSE	Par: <espress> THEN <comando1> [ELSE <comando2>]. Esegue condizione. es.	S/C/Cr/Cd/Sm/Cm
IFF - ELSEF - ENDF	Par: <espress> <comandi1> [ELSEF <comandi2>] ENDF. Esegue condizione. es.	S/C/Cr/Cd/Sm/Cm
GOTO	Par: <nometic>. Istruzione che esegue il salto ad una etichetta. es.	S/C/Cr/Cd/Sm/Cm
GOSUB - RETURN	Par: <nometic> (...+ RETURN). Esegue salto a subroutine. es. Caso: l'utilizzo nel ciclotabellare di GOSUB e RETURN potrebbe richiedere EXITTAB per uscire dal ciclotabellare. Caso: messaggio "troppi GOSUB in sospenso": Il limite di GOSUB in sospenso è 30. Entro questi limiti la routine deve arrivare a un RETURN nella nidificazione.	S/C/Cr/Cd/Sm/Cm
FOR - NEXT	Par: <varnum>=<x> TO <y> [STEP <z>] : '...' : NEXT. Esecuzione ciclica. es.	S/C/Cr/Cd/Sm/Cm
EXITTAB	Uscita forzata dalla tabella di lettura (v. Ciclotabellare)	S/Cr/Cd
AZZCACHE	Istruzione che azzerla la cache del Dizionario Non consentita nei Livelli di sviluppo obsoleti (F2, L3, F3)	S/C/Cr/Cd/Sh
END	Termine del programma / routine. es.	S/C/Cr/Cd/Sm/Cm
ON_END:	Evento richiamato al termine del modulo; consultare Linguaggio Sprix/Collage (etichette)	C/Cd
ABILCOLLAUTO	Par: <nomefile\$>,<abil>. Istruzione che abilita/disabilita caricamento Collage (*a)	C
STARTCOLLAGE	Par: <nomefile\$>. Istruzione che carica in memoria un nuovo Collage (*a)	C/Cd
STOPCOLLAGE	Par: <nomefile\$>. Istruzione che scarica da memoria il Collage specifico (*a)	C/Cd
CALL	Par: <nomefile\$>,<nomeetic\$>. Istruzione che esegue routine Collage (*a) es.	C
MAXCOLLAGE()	Par: (). Funzione che torna il numero di Collage attivi (*a)	S/C/Cr/Cd
NOMECOLLAGE\$()	Par: (<num>). Funzione che torna il nome del file Collage attivo (*a)	S/C/Cr/Cd
CARCOLLAGE()	Par: (<nomefile\$>). Funzione che torna il numero d'ordine del Collage (*a)	S/C/Cr/Cd
	(*a) Si ricorda che i Collage sono caricati dinamicamente in base a fattori che NON incidono sull'ordine di esecuzione dei collage, bensì sull'ordine di esecuzione del singolo evento, pertanto, l'ordine dei collage è ininfluente. Non è possibile scegliere l'ordine di esecuzione. I fattori che NON consentono di predeterminare, neppure di scegliere, l'ordine di esecuzione dei Collage, sono i seguenti: <ul style="list-style-type: none"> • se il Collage è configurato su menu AUTO oppure menu specifico (es. CK); • se l'evento è generico (ON_IN_ALL) oppure un evento specifico (ON_IN_nomecampo); più collage di App differenti che si attivano sulla stessa etichetta, si eseguono in base all'ordine dei codici delle applicazioni nell'elenco delle App di PassBuilder; • se presenti Collage disabilitati; • se eseguiti Collage con comando STARTCOLLAGE - STOPCOLLAGE e, in quale ordine; verificare sullo stesso Collage oppure più Collage sullo stesso menu (non AUTO), ma non la stessa etichetta; • se Collage è definito in una App PassBuilder oppure no; le singole App sono reciprocamente indipendenti (non si influenzano fra di loro), pertanto, non è consentito eseguire i comandi STOPCOLLAGE e STARTCOLLAGE per governare i Collage di altre App; 	
ESECMENU()	Par: (<menu\$>). Istruzione di esecuzione diretta e automatica menu	C
_VERSAPP	Versione dell'applicazione Nuova	S/C/Ca/Cd
_VERSAPP_PRE	Versione dell'applicazione Precedente	S/C/Ca/Cd
LIVLICAPP()	Funzione che ritorna il numero di livello di licenza acquistato sul PassStore per l'App. es.	S/C/Cr/Cd/Sm/Cm
ON_EVENTO_APP	Eventi per agire con Collage di controllo App sugli stati dell'App EVENTO equivale a "NUOVA", "MODIFICA", "ELIMINA", "POST_SYNC" Nell'Applicazione realizzata con PassBuilder è consentito un Collage di controllo applicazione. Vedere Pass Mobile	Ca
CREAUUID\$()	Par: (<pref\$>). Funzione che ritorna una stringa secondo le specifiche rfc4122; utile per "clicca & posta", ma anche per altre applicazioni.	S/C/Cr/Cd
NOMEPASSAPP\$()	Funzione che ritorna il nome completo della PassApp.	S/C/Cd
Semaforica		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
BLOCCASEM	Par: <nomesem\$>. Istruzione che blocca un semaforo. es.	S/C/Cr/Cd/Sh
SBLOCCASEM	Par: <nomesem\$>. Istruzione che sblocca un semaforo. es.	S/C/Cr/Cd/Sh
_ERRSEM	Codice risultato operazione BLOCCASEM. es.	S/C/Cr/Cd/Sh
Valore Numerico		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
DIM ()	Par:<num>([dim1,dim2,dim3) Funzione che inizializza le dimensioni dell'array numerico, si accede a ciascun elemento alla volta tramite l'indice fra parentesi tonde indicato in modo diretto o ciclico, anche tramite variabile. es.	S/C/Cr/Cd/Sm/Cm
AZZARRAY	Par: [num]. Istruzione che azzerla array. es.	S/C/Cr/Cd/Sm/Cm
_GLOBAL()	Array numerico globale Collage	C/Cd
CHR\$()	Par: (<num>). Funzione che ritorna il carattere del valore ASCII. La mappa di caratteri speciali è consultabile premendo in sequenza i tasti funzione [F1+F1+F3]. L'elenco dei caratteri estesi è in elenco alla presente tabella caratteri estesi . Esempio: X=87: risultato CHR\$(X) è "W". Esempio: X=128: risultato CHR\$(X) è "€". Nota: la stampa del simbolo Euro dipende dalla configurazione del dispositivo di stampa, pertanto, se è di tipo "W" la gestione dei caratteri speciali è affidata al sistema operativo, se è di tipo "S" dipende dalla stampante (cioè dal set di caratteri, driver o altro relativo alla configurazione della stampante). In alternativa, valutare	S/C/Cr/Cd/Sm/Cm

	<p>l'utilizzo del simbolo € contenuto in file immagine supportato da impiegare nella funzione VIMM (nella modulistica) e LPRINTIMG (in Sprix)</p> <p>La funzione CHR\$() è spesso utile e/o necessaria per gestire stringhe che presentano caratteri speciali. Può capitare che un formato testo CSV venga gestito e/o modificato con altri applicativi (per esempio Excel) e che il relativo salvataggio ne possa modificare i reali contenuti. In taluni casi, piuttosto che realizzare un file con estensione ".CSV" da aprire e poi gestire con Excel, per evitare potenziali interpretazioni che modificassero il tipo dato stringa è meglio sfruttare le stampe personalizzate da Sprix, creando un modulo di stampa che può essere reindirizzato in qualunque tipo di stampa desiderato.</p> <p>Per esempio, un CODICEARTICOLO in un file CSV viene salvato con una determinata impostazione di Excel, ma successivamente si scopre che viene anticipato da un prefisso costituito dal doppio apice ("CODICEARTICOLO"). Dato che il programma non gestisce questa sintassi (CODART\$="CODICEARTICOLO") si incorrerebbe in un errore se, invece, non si gestisse la stringa trattando il carattere (") con CHR\$(34), ovvero: CODART\$=CHR\$(34)+"CODICEARTICOLO"</p>	
ABS()	<p>Par: (<num>). Funzione che ritorna il valore assoluto del numero.</p> <p>Esempio: dati A=-1234; B=4 il risultato di ABS(A*B/2) o ABS(-1234*2) è 2468</p>	S/C/Cr/Cd/Sm/Cm
ARRO()	<p>Par: (<numero>,<tipo>,<arr>). Funzione che arrotonda il numero.</p> <p>Esempio: N=12500; ARRO(N,2,1000) vale 12000, ARRO(N,3,1000) vale 13000. In caso di numero con decimali (es N=12,536 da arrotondare con tipo 1 al centesimo) può adottarsi la forma: B=ARRO (N*100,1,1)/100</p>	S/C/Cr/Cd/Sm/Cm/Sh
ARRON()	<p>Par: (<numero>,<tipo>,<decimali>). Funzione che arrotonda in base ai decimali.</p> <p>Esempio: dato N=12,1256; ARRON(N,1,3) vale 12,125, ARRON(N,4,2) vale 12,13</p>	S/C/Cr/Cd/Sm/Cm/Sh
SETDECPIX	<p>Par: (<ndec>). Istruzione per numero decimali trattati da FIX automatico.</p> <p>Esempio: SETDECPIX 2: A=70003.5399911; B=9337.54; C=A+B; E=C=79341.07 (senza FIX a 6 dec. C=79341.079999)</p>	S/C/Cr/Cd/Sm/Sh
FIX()	<p>Par: (<num>,<numdec>). Funzione che tronca i decimali di un numero.</p> <p>Esempio: A=-12345.678; B=123.5; C=FIX(A,0); D=FIX(A,2); E=FIX(B,3); risultati: C=-12345; D=-12345,67; E=123,5</p>	S/C/Cr/Cd/Sm/Cm/Sh
CIFLET\$()	<p>Par: (<num>). Funzione che converte una cifra in lettere.</p> <p>Esempio: X=1234.88; CIFLET\$(X) risultato: MILLEDUECENTOTRENTAQUATTRO</p>	S/C/Cr/Cd/Sm/Cm/Sh
CIFLETDEC\$()	<p>Par: (<num>,<numdec>). Funzione che converte una cifra in lettere con decimali.</p> <p>Esempio: Y=6543.21; CIFLETDEC\$(Y,3) risultato: SEIMILACINQUECENTOQUARANTATRE/210</p>	S/C/Cr/Cd/Sm/Cm/Sh
STR\$()	<p>Par: (<num>,<numdec>,<punti>). Funzione che converte numero in stringa.</p> <p>Esempio: A=1234.567; B=-9876.543; STR\$(A,2,1); STR\$(B,1,0) = "1.234,56" e "-9876,5"</p>	S/C/Cr/Cd/Sm/Cm
MKN\$()	<p>Par: (<num>,<numbyte>,<numdec>). Funzione che converte numero in stringa (binario).</p> <p>Esempio: C\$=MKN\$(C,2,0); converte C (tra -32.768 e 32.767) in stringa 2 byte e 0 dec.</p>	S/C/Cr/Cd/Sh
CAMBIOVAL()	<p>Par: (<valp>,<vala>,<impo>,<dtcambio>,<numdec>). Funzione per il cambio valuta. La variabile _MMC/VF\$="F" imposta il cambio fisso per inserire un cambio arbitrario da memorizzare sul documento e consultabile tramite le rimanenti variabili dei cambi (v. nota *a della struttura _MM)</p>	S/C/Cr/Cd/Sh
Valore Stringa		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
DIM ()	<p>Par: <str\$>([dim1],dim2,dim3) Funzione che inizializza le dimensioni dell'array stringa, si accede a ciascun elemento alla volta tramite l'indice fra parentesi tonde indicato in modo diretto o ciclico, anche tramite variabile. es.</p>	S/C/Cr/Cd/Sm/Cm
AZZARRAY	<p>Par: [str\$]. Istruzione che azzerava array. es.</p>	S/C/Cr/Cd/Sm/Cm
GLOBAL\$()	<p>Array alfanumerico globale Collage</p>	C/Cd
ASC()	<p>Par: (<str\$>). Funzione che torna il valore ASCII del primo carattere della stringa</p> <p>Esempio: A\$="A1B" il risultato di ASC(A\$) è ASC("A") cioè 65</p>	S/C/Cr/Cd/Sm/Cm
LEN()	<p>Par: (<str\$>). Funzione che torna la dimensione di una stringa.</p> <p>Esempio: A\$=" A 1 B " il risultato di LEN(A\$) o LEN(" A 1 B ") è 7</p>	S/C/Cr/Cd/Sm/Cm
INSTR()	<p>Par: (<str\$>,<ricstr\$>). Funzione che cerca all'interno di una stringa.</p> <p>Esempio: A\$="spRix"; B\$="ri" il risultato di INSTR(A\$,B\$) o INSTR(A\$,"ri") è 3</p>	S/C/Cr/Cd/Sm/Cm
INSTR_CI()	<p>Par: (<str\$>,<ricstr\$>). Funzione che cerca sottostringa in stringa, tipo case insensitive.</p> <p>Esempio: A\$="spRix"; B\$="ri" il risultato di INSTR_CI(A\$,B\$) o INSTR_CI(A\$,"ri") è 3</p>	S/C/Cr/Cd/Sm/Cm/Sh
STRCMP_CI()	<p>Par: (<str\$1>,<str\$2>). Funzione per confrontare stringhe, tipo case insensitive.</p> <p>Esempio: A\$="A"; B\$="a"; STRCMP_CI(A\$,B\$) è 0</p>	S/C/Cr/Cd/Sm/Cm/Sh
VAL()	<p>Par: (<str\$>). Funzione che torna il valore numerico di una stringa.</p> <p>Esempio: A\$="1,3"; B\$="12ud"; C\$="1A9"; D\$="F14"; E\$="+3,99"; F\$="-,2"; G\$="1+2" Ris. A=VAL(A\$)=1,3; B=VAL(B\$)=12; C=VAL(C\$)=1; D=VAL(D\$)=0; E=VAL(E\$)=3,99; F=VAL(F\$)=-0,2; G=VAL(G\$)=1</p>	S/C/Cr/Cd/Sm/Cm
CVN()	<p>Par: (<str\$>,<numbyte>,<numdec>). Funzione che scompatta i campi trattati con MKN\$.</p> <p>Esempio: C=CVN(C\$,2,0) interpreta i primi 2 byte di C\$ per ottenere il numero C</p>	S/C/Cr/Cd/Sh
MKA\$()	<p>Par: (<str\$>,<numbyte>). Funzione che aggiunge ASCII 0 in fondo alla stringa.</p> <p>Esempio: A\$="Sprix"; B\$=MKA\$(A\$,7) risultato B\$="Sprix "</p>	S/C/Cr/Cd/Sh
CVA\$()	<p>Par: (<str\$>,<numbyte>). Funzione che elimina i caratteri ASCII 0 da stringa.</p> <p>Esempio: B\$="Sprix "; C\$=CVA\$(B\$,7) risultato C\$="Sprix"</p>	S/C/Cr/Cd/Sm/Cm/Sh
CVMAIU\$()	<p>Par: (<str\$>). Funzione che converte la stringa in maiuscolo.</p> <p>Esempio: A\$="Sprix"; B\$=CVMAIU\$(A\$) o B\$=CVMAIU\$("Sprix") risultato B\$="SPRIX"</p>	S/C/Cr/Cd/Sm/Cm/Sh
CVMINU\$()	<p>Par: (<str\$>). Funzione che converte la stringa in minuscolo.</p> <p>Esempio: C\$="SPRIX"; D\$=CVMINU\$(C\$) o D\$=CVMINU\$("SPRIX") risultato D\$="sprix"</p>	S/C/Cr/Cd/Sm/Cm/Sh
STRING\$()	<p>Par: (<numrip>,<str\$>). Funzione che ritorna una stringa ripetuta 'n' volte.</p> <p>Esempio: R\$=STRING\$(3,"a") o R\$=STRING\$(3,CHR\$(97)+CHR\$(45)) risultato R\$="a-a-a"</p>	S/C/Cr/Cd/Sm/Cm/Sh
LEFT\$()	<p>Par: (<str\$>,<numcar>). Funzione che estrae una stringa partendo da sinistra.</p> <p>Esempio: A\$="SPRIX"; B\$=LEFT\$(A\$,3) risultato B\$="Spr" Esempio: C\$="SPRIX"; D\$=LEFT\$(C\$+STRING\$(10," "),10); risultato D\$="SPRIX "</p>	S/C/Cr/Cd/Sm/Cm
MID\$()	<p>Par: (<str\$>,<posiniz>,<numcar>). Funzione che estrae la porzione di una stringa. Funzione utile anche per ottenere ordinamenti non previsti dal ciclotabellare.</p> <p>Esempio: A\$="sprixEcollage"; B\$=MID\$(A\$,4,5) risultato B\$="ixEco" Esempio: C\$="sprixEcollage"; D=4; E=LEN(C\$)-D+1; F\$=MID\$(C\$,D,E) ris. F\$="ixEcollage"</p>	S/C/Cr/Cd/Sm/Cm
RIGHT\$()	<p>Par: (<str\$>,<numcar>). Funzione che estrae una stringa partendo da destra.</p> <p>Esempio: E\$="Sprix"; F\$=RIGHT\$(E\$,3) ris. F\$="rix" Esempio: G\$="Sprix"; H\$=RIGHT\$(" "+G\$,7) risultato H\$=" Sprix"</p>	S/C/Cr/Cd/Sm
_LRECSORT	<p>Par: = N. Lunghezza stringa da ordinare. es.</p>	S/C/Cr/Cd/Sh
_SORT_STR_CI	<p>Par: = N. Ordinamento per caratteri estesi. es.</p>	S/C/Cr/Cd/Sh
PUTSORT	<p>Par: <str\$>. Istruzione che inserisce stringa in lista da ordinare. es.</p>	S/C/Cr/Cd/Sh
GETSORT	<p>Par: <str\$>. Istruzione che legge la prossima stringa ordinata. es.</p>	S/C/Cr/Cd/Sh

Data e Tempo		Cm/Ca/Sh/Cc
STRDATA\$()	Par: (<data\$>,<forma>,<separatore\$>). Funzione che converte la data. Esempio: D1\$="20090623" : D2\$=STRDATA\$(D1\$,3,"/") risultato D2\$="23/06/2009"	S/C/Cr/Cd/Sm/Cm/Sh
DIFDATA()	Par: (<data1\$>,<data2\$>). Funzione che torna la differenza tra due date. Esempio: OGGI\$="20090620" : IERIS\$="20090619" : D=DIFDATA(IERIS\$,OGGI\$) risultato D = -1	S/C/Cr/Cd/Sm/Cm/Sh
GIORSET()	Par: (<data\$>). Funzione che torna il giorno della settimana. Esempio: OGGI\$="20090618" : N = GIORSET(OGGI\$) risultato N=4 (giovedì)	S/C/Cr/Cd/Sm/Cm/Sh
PIUDATA\$()	Par: (<data\$>,<numgior>). Funzione che torna la data variata da un numero di giorni. Esempio: D\$="20090621" : B\$=PIUDATA\$(D\$,-1) : C\$=STRDATA\$(PIUDATA\$(B\$,10),2,"/") risultato C\$=30/06/09	S/C/Cr/Cd/Sm/Cm/Sh
AZZCRONO	Par: <nc>. Istruzione che resetta cronometro	S/C/Cr/Cd
CRONO()	Par: (<nc>). Funzione che legge cronometro Esempio: AZZCRONO 1 : CICLO: PRINT CRONO(1): IF CRONO(1)<10 THEN GOTO CICLO	S/C/Cr/Cd
SLEEP()	Par: (<n>). Funzione che sospende il processo per n millisecondi. Esempio: SLEEP(1500) sospende il programma per 1500 millisecondi (1,5 sec)	S/C/Cr/Cd

esempi e casi

Esempio1: comandi generali, numeri, stringhe, data, tempo, semaforica
(replicare -copia/incolla- delle librerie e del sorgente Sprix)

```
'Sorgente Sprix spxABC.ss
#APPENDISPX <spx901><lib2> 'libreria elaborazioni numeri e stringhe
#APPENDISPX <lib3> 'libreria elaborazioni date e tempo

'dichiaro Variabili
NUM_01=-45.123456 : ALFA_$="questa è la stringa 123"

'dimensiono, valorizzo e poi azzero degli Array
DIM ARR(3) 'numerico a una dimensione di 3 elementi
ARR(1)=7 : ARR(2)=28.5 : ARR(3)=1492.333
DIM ARR$(,10,20) 'alfanumerico di 3 dimensioni e la 1^ automatica
ARR$(1,1,1)="Sprix" : ARR$(2,1,1)="Collage" : ARR$(3,1,1)="MDS" etc.
AZZARRAY ARR : AZZARRAY ARR$ 'azzero gli array

T=0 'numeratore controllo esecuzione routine

INIZIO:
T=T+1 ' incremento il contatore
IF T>0 AND T<4 THEN GOSUB CONTROL ELSE IF T=0 OR T>=4 THEN GOSUB FINE
END

CONTROL:
IF T=1 THEN GOSUB ELABORANUMERO : GOTO INIZIO
IF T=2 THEN GOSUB ELABORASTRINGA : GOTO INIZIO
IF T=3 THEN GOSUB ELABORADATA : GOTO INIZIO
IF T=4 THEN GOTO INIZIO
RETURN

ELABORANUMERO:
'dichiaro variabili numeriche e effettuo delle elaborazioni in libreria
N1=252000 : N2=770.14453 : GOSUB ETIC1 : CLS

ELABORASTRINGA:
'dichiaro variabili stringa e le concateno
A1$="Sprix;Collage;" : A2$="MDS;SDK;2009" : A$=A1$+A2$
'spezzo la stringa concatenata A$ e dimensiono Array stringa automatico
PRINT " la lib2 spezza la stringa "+CHR$(34)+A$+CHR$(34) : PRINT ""
DIM AR$( ) : GOSUB ETIC2A:
FOR Z=1 TO I
PRINT STRING$(30, " ") + AR$(Z)
NEXT
VIMSG 1, "ho spezzato la stringa Sprix;Collage;MDS;SDK;2009" : CLS
GOSUB ETIC2B : CLS

ELABORADATA:
'dichiaro due date
DATA1$="20090303" : DATA2$="20090308" : GOSUB ETIC3
VIMSG 1, "ho trattato la DATA, adesso inizio a|contare il TEMPO" : CLS
GOSUB CRONOMETRO 'conto il tempo
VIMSG 1, "ora sospendo per 2,5 secondi" : CLS
GOSUB SOSPENDO :VIMSG 1,"ho sospeso per 2,5 secondi" : CLS

ORDINA:
DIM ARRAY$(,13,4) 'il mazzo ha 13 carte per 4 semi
PRINT "": PRINT "Quanti mazzi di carte ? (1-3): "; : INPUT "N",NUM,1
_LRECSORT=30
FOR M=1 TO NUM 'mazzi
FOR V=13 TO 1 STEP -1 'carte
FOR S=1 TO 4 'semi
IF S=1 THEN S$=" cuori "
IF S=2 THEN S$=" quadri "
IF S=3 THEN S$=" fiori "
IF S=4 THEN S$=" picche "
ARRAY$(M,V,S)=" Mazzo: "+STR$(M,0,0)+S$+" carta: "+STR$(V,0,0)
PRINT ARRAY$(M,V,S) : PUTSORT ARRAY$(M,V,S) : SLEEP 1.5
NEXT
NEXT
NEXT

VIMSG 1, "i semi del mazzo sono|DISORDINATI||ora li|RIORDINO" : CLS

RIPRENDI:
DIM SOSPESO$( ) : SS=0 : CC=0 'array sospeso per ordinare carte > 10

LEGGISORT:
IF CC=13 AND SS=4 THEN GOSUB STAMPAZ : GOTO RIPRENDI
GETSORT CARTA$ : CC=CC+1
IF CARTA$="" THEN VIMSG 1, "ho riordinato per seme": GOTO CTRL_SEM
IF VAL(MID$(CARTA$,26,2))<10 THEN PRINT CARTA$ ELSE GOSUB SOSPESO
GOTO LEGGISORT

SOSPESO:
SS=SS+1 : SOSPESO$(SS)=CARTA$ : RETURN
STAMPAZ:
FOR K=1 TO SS
PRINT SOSPESO$(K) :
NEXT
```

```

AZZARRAY SOSPESOS
RETURN

CTRL SEM:
BLOCCASEM "**SEM1"
IF ERRSEM<>0 THEN VIMSG 4,"Semaforo non bloccabile" ELSE GOTO SCRIVI
SLEEP 1500 : GOTO CTRL_SEM

SCRIVI:
CLS: VIMSG 1,"il controllo sul blocco semaforo|mi permette di operare"
QTA=0 : PZO=0 : TOTQTA=0 : TOTVAL=0 : S$="sprix"
PRINT " numero|quantita|prezzo "
PRINT "-----|-----|-----"

FOR I=1 TO 5
  QTA=I*12
  PZO=1+(I/3)
  S$=S$+STR$(I,0,0)
  PRINT " "+STR$(I,0,0)+" | "+STR$(QTA,0,0)+" | "+STR$(PZO,-1,0)
  TOTQTA=TOTQTA+QTA:TOTVAL=TOTVAL+(PZO*QTA)
NEXT

PRINT "-----"
PRINT " Tot.Qta "+STR$(TOTQTA,0,0)+" | Tot.Val. "+STR$(TOTVAL,-1,0)

SBLOCCASEM "**SEM1"
VIMSG 1,"sblocco e libero il semaforo" : CLS
VIMSG 1,"ora cancello un carattere alla volta|dalla stringa "+S$

CICLO3:
LUNG=LEN(S$) : IF LUNG<5 THEN GOTO FINE ELSE PRINT "": PRINT " "+S$
S$=LEFT$(S$,LUNG-1): GOTO CICLO3

FINE:
  VIMSG 1,"Esempio Terminato"
END ' per non proseguire con il codice di libreria

'LIBRERIA 'spx901.ss' elaborazioni numeriche'

ETIC1:
N=N1+N2:NEG=-N:NN=N*1.5498:NNN=ARRO(NN,3,1):NNNN=ARRO(NN*100,3,1)/100
PRINT" N1 e N2 valgono      = "+STR$(N1,0,1)+" e "+STR$(N2,-1,0)
PRINT" N1 + N2 valgono      N = "+STR$(N,-1,0)
PRINT" N negativo vale      NEG = "+STR$(NEG,-1,0)
PRINT" NEG in valore assoluto = "+STR$(ABS(NEG),0,0)
PRINT" NEG in lettere       = "+CIFLET$(NEG)
PRINT" N * 1.5498 vale      NN = "+STR$(NN,-1,0)
PRINT" NN fissato a 3 decimali = "+STR$(FIX(NN,3),-1,0)
PRINT" NN arrotondato vale  NNN = "+STR$(NNN,-1,0)
PRINT" NN arrotondato meglio NNNN = "+STR$(NNNN,-1,0)
PRINT" NNNN in lettere      = "+CIFLETDEC$(NNNN,3)
PRINT" NN+NNNN= "+STR$(NN+NNNN,-1,0)
SETDECPIX 2: PRINT" NN+NNNN fissato a 2 dec.li = "+STR$(NN+NNNN,-1,0)
PRINT" Carattere ASCII di 83 = "+CHR$(83)
PRINT" N Euro in $$ (cambio=0,5)="+STR$(CAMBIOVAL(2,5,N,"20210101"),5),-1,0)
PRINT" Impaccamento binario di N = " : IMP$(MKN$(N,5,5)
VIMSG 1,"Queste sono tutte|funzioni di trattamento VALORI NUMERICI"
T1$="Gli operatori ARITMETICI| (per espressioni/concatenazioni di valori)|"
T2$="formano con gli operatori LOGICI e DI RELAZIONE|"
T3$="questa gerarchia decrescente di priorit  d'esecuzione|"
T4$="NOT -negativo ^ * / MOD + -sottrazione <=> >=< > AND OR ( )"
VIMSG 1, T1$+T2$+T3$+"|"+T4$

FINE1:
  RETURN ' fine ETIC1

'LIBRERIA 'lib2.ss' elaborazioni alfanumeriche'

ETIC2A: 'spezza la stringa A$ e ritorna Array per quanti sono i ";"
I=1

CICLO1:
K=INSTR(A$,";")
IF K=0 THEN AR$(I)=A$ : GOTO SALTA
AR$(I)=LEFT$(A$,K-1) : A$=RIGHT$(A$,LEN(A$)-K) : I=I+1
IF INSTR(A$,";")=0 THEN AR$(I)=A$ : GOTO SALTA ELSE GOTO CICLO1
SALTA:
  RETURN ' fine ETIC2A

ETIC2B: 'elaborazioni alfanumeriche
A$=A1$+A2$ : B$=AR$(1)+AR$(2)+AR$(3)+AR$(4)+AR$(5)

PRINT"A1$ e A2$ valgono = "+A1$+" e "+A2$
PRINT"A1$ + A2$ vale A$ = "+A$
PRINT"A1$ + A2$ senza ; vale B$ = "+B$
PRINT"ASCII del primo carattere di B$ = "+STR$(ASC(B$),0,0)
PRINT"Lunghezza in caratteri di B$ = "+STR$(LEN(B$),0,0)
PRINT"Posiz. di MDS (o AR$(3)) in B$ = "+STR$(INSTR(B$,"MDS"),0,0)
PRINT"Pos. CaseInsensitive di Collage = "+STR$(INSTR_CI(B$,AR$(2)),0,0)
PRINT"Verifica C.I. se A$ = B$ = "":
IF STRCMP CI(A$,B$)=0 THEN PRINT "vero" ELSE PRINT "falso"
PRINT"Converte AR$(5) in un numero = "+STR$(VAL(AR$(5)),0,0)
PRINT"Disimpaccamento binario di N = "+STR$(CVN(IMP$,5,5),-1,0)
PRINT"Aggiunge 3 ASCII 0 a Sprix = "": MK_AR$=MKA$(AR$(1),8)
PRINT MK_AR$;: PRINT "(visibile solo in debug)"
PRINT"Toglie tutti ASCII 0 a Sprix = "": CV_AR$=CVA$(MK_AR$,8)
PRINT CV_AR$+(visibile solo in debug)"
PRINT"Converte B$ in maiuscolo = "+CVMAIU$(B$)
PRINT"Converte B$ in minuscolo = "+CVMINUS$(B$)
PRINT"Ripete 5 volte Sprix = "+STRING$(5,"Sprix")
PRINT"Estrae i primi 12 car. da B$ = "+LEFT$(B$,12)+"|"
PRINT"Estrae 7 car. da 6^ posiz.di B$ = "+|" +MID$(B$,6,7)+"|"
PRINT"Estrae gli ultimi 10 car. da B$ = "+|" +RIGHT$(B$,10)
PRINT"Estrae da sx e mantiene lung.B$ = "":
PRINT LEFT$(LEFT$(B$,12)+STRING$(LEN(B$), " ", LEN(B$))+"|"
PRINT"Estrae da dx e mantiene lung.B$ = "":
PRINT "|"+RIGHT$(STRING$(LEN(B$), " ") +RIGHT$(B$,10), LEN(B$))
VIMSG 1,"Queste sono tutte|funzioni di trattamento VALORI ALFANUMERICI"

```

```

FINE2:
    RETURN ' fine ETIC2B

LIBRERIA 'lib3.ss' elabora date e tempo

ETIC3:
PRINT"date DATA1$ e DATA2$      = "+DATA1$+" e "+DATA2$
PRINT"data DATA1$ formattata     = "+STRDATA$(DATA1$,3,"/")
PRINT"giorni da DATA1$ a DATA2$ = "+STR$(DIFDATA(DATA1$,DATA2$),0,0)
PRINT"giorno settimanale di DATA2$ = "+STR$(GIORSET(DATA2$),0,0);
    IF GIORSET(DATA2$)=7 THEN PRINT " (domenica)"
PRINT"DATA2$ diminuita di 3 giorni = "+PIUDATA$(DATA2$,-3)

FINE3:
    RETURN 'fine ETIC3

CRONOMETRO:
FOR I=1 TO 5
    AZZCRONO 1
    CICLO2:
    IF CRONO(1)<1 THEN GOTO CICLO2
    PRINT "": PRINT " Tempo in secondi = "+STR$(I,0,0)
NEXT
VIMSG 1,"Conto il tempo che impiegherai a confermare questo messaggio"
VIMSG 4,"secondi, millisecondi trascorsi = "+STR$(CRONO(1),-1,0)
AZZCRONO 1

FINECRONOMETRO:
    RETURN 'fine CRONOMETRO

SOSPENDO:
PRINT "": PRINT "      inizio PAUSA di 2,5 secondi"
SLEEP(2500)
PRINT "": PRINT "      termine PAUSA di 2,5 secondi"

FINESOSPENDO:
    RETURN ' fine SOSPENDO

```

Esempio2 comandi generali, numeri, stringhe, data, tempo, ordinamento, file I/O
(replicare -copia/incolla- delle librerie, del sorgente Sprix e del file di esito procedura)

Sprix che esegue un test di comandi su valori numerici, stringa, ora e data, ordinamento, file I/O
 Il programma lavora nella dir \temp, che deve esistere, e crea il file \temp\esito.txt con l'esito della procedura.
 Nella \temp va posto il file esito.ok (vedere in fondo a questo esempio) col quale viene effettuato il confronto.
 Se il confronto non e' OK la procedura e' congruente al file "esito.ok"

Sorgente Sprix "spxtest.ss"

```

#APPENDISPX <libtest>

' apre file esito
NOMEF_ESITO$="c:\temp\esito.txt"
GOSUB OPEN_ESITO

LOCATE 1,1: PRINT "TEST ISTRUZIONI SPRIX  :"
PRINT "- numeriche": PRINT "- stringa": PRINT "- ora e data"
PRINT "- ordinamento": PRINT "- file io (sequenziale e a indici) e SHELL"
PRINT "": PRINT "Viene generato il file "+NOMEF_ESITO$:
PRINT "": PRINT "Attendere prego"

'dichiaro Variabili
NUM_01=-45.123456 : ALFA_$="questa e' la stringa 123"

'dimensiono, valorizzo e poi azzero degli Array
DIM ARR(3) 'numerico a una dimensione di 3 elementi
ARR(1)=7 : ARR(2)=28.5 : ARR(3)=1492.333
DIM ARR$(,10,20) 'alfanumerico di 3 dimensioni e la 1^ automatica
ARR$(1,1,1)="Sprix" : ARR$(2,1,1)="Collage": ARR$(3,1,1)="MDS" etc.
AZZARRAY ARR : AZZARRAY ARR$ ' azzero gli array

T=0 'numeratore controllo esecuzione routine

INIZIO:
T=T+1 ' incremento il contatore
IF T>0 AND T<6 THEN GOSUB CONTROL ELSE IF T=0 OR T>=6 THEN GOSUB FINE

FINE:

GOSUB CLOSE_ESITO

' confronto il file esito generato col file esito.ok
ERRDIFF=0: MSG$=""
DIFF_FILE1$=NOMEF_ESITO$
DIFF_FILE2$="c:\temp\esito.ok"
DIFF_VCOORD=1
IF ESISFILE(DIFF_FILE2$) THEN PRINT "Confronto esito con "+DIFF_FILE2$
IF ESISFILE(DIFF_FILE2$) THEN GOSUB DIFF_FILE ELSE ERRDIFF=1

IF ERRDIFF=0 AND DIFF_MSG$<>"" THEN ERRDIFF=2

IF ERRDIFF=0 THEN MSG$="Confronto OK" ' Tutto OK
IF ERRDIFF<>0 THEN BEEP
IF ERRDIFF=1 THEN MSG$="Confronto non effettuato: manca "+DIFF_FILE2$
IF ERRDIFF=2 THEN MSG$=DIFF_MSG$

VIMSG 1, MSG$
END

CONTROL:
IF T=1 THEN GOSUB ELABORANUMERO : GOTO INIZIO
IF T=2 THEN GOSUB ELABORASTRINGA : GOTO INIZIO
IF T=3 THEN GOSUB ELABORADATA : GOTO INIZIO
IF T=4 THEN GOSUB ORDINA : GOTO INIZIO
IF T=5 THEN GOSUB TEST_FILE_IO : GOTO INIZIO
IF T>=6 THEN GOTO INIZIO

RETURN

```

```

-----
LABORANUMERO:
FESITO_BUF$="*** TEST NUMERICHE ***": GOSUB WRITE_ESITO
'dichiaro variabili numeriche e effettuo delle elaborazioni in libreria
N1=252000 : N2=770.14453 : GOSUB ETIC1
RETURN

-----
LABORASTRINGA:
FESITO_BUF$="*** TEST STRINGHE ***": GOSUB WRITE_ESITO
'dichiaro variabili stringa e le concateno
A1$="Sprix;Collage;" : A2$="MDS;SDK;2009" : A$=A1$+A2$
'spezza la stringa concatenata A$ e dimensiono Array stringa automatico
FESITO_BUF$= " la lib2 spezza la stringa "+CHR$(34)+A$+CHR$(34): GOSUB WRITE_ESITO
DIM AR$( ):GOSUB ETIC2A:FOR Z=1 TO I:FESITO_BUF$= STRING$(30," ") +AR$(Z): GOSUB WRITE_ESITO
NEXT
' ho spezzato la stringa Sprix;Collage;MDS;SDK;2009"
GOSUB ETIC2B
RETURN

-----
LABORADATA:
FESITO_BUF$="*** TEST DATA E ORA ***": GOSUB WRITE_ESITO
'dichiaro due date
DATA1$="20090303" : DATA2$="20090308" : GOSUB ETIC3
' ho trattato la DATA, adesso inizio a contare il TEMPO
GOSUB CRONOMETRO ' conto il tempo
' ora sospendo per 2,5 secondi
GOSUB SOSPENDO
RETURN

-----
ORDINA:
FESITO_BUF$="*** TEST ORDINAMENTO ARRAY DI STRINGHE ***": GOSUB WRITE_ESITO
SIZENOMI=5: DIM NOMI$( )
NOMI$(1)="ROBERTO": NOMI$(2)="LUCA": NOMI$(3)="STEFANO"
NOMI$(4)="ANDREA": NOMI$(5)="STEFANO"

' eseguo ordinamento
_SORT_STR_CI=1 ' case sensitive
LRECSORT=20 ' max lunghezza della stringa da ordinare
FOR M=1 TO SIZENOMI
PUTSORT NOMI$(M)
NEXT

' leggo nomi ordinati e li scrivo nell'esito
CTR=0
LEGGISORT:
GETSORT NOME$
IF NOME$<>" " THEN CTR=CTR+1: FESITO_BUF$=NOME$: GOSUB WRITE_ESITO: GOTO LEGGISORT
FESITO_BUF$="ORDINATI "+STR$(CTR,0,0)+" ELEMENTI": GOSUB WRITE_ESITO

' ora la successiva PUTSORT azzera la lista di sort e inserisce "ZARA"
ERRORE=0 ' 0=OK 1=errore
PUTSORT "PINO"
GETSORT NOME$
IF CVA$(NOME$,LEN(NOME$)) <> "PINO" THEN ERRORE = 1
IF ERRORE = 0 THEN GETSORT NOME$
IF ERRORE = 0 AND NOME$ <> " " THEN ERRORE = 1
IF ERRORE = 1 THEN FESITO_BUF$="Errore : la successiva PUTSORT non cancella la lista di sort": GOSUB WRITE_ESITO
RETURN

' sobroutine
ETIC1:
SETDECPIX 6
N=N1+N2:NEG=-N:NN=N*1.5498:NNN=ARRO(NN,3,1):NNNN=ARRO(NN*100,3,1)/100
FESITO_BUF$=STR$(N1,0,1)+" e "+STR$(N2,-1,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(N,-1,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(NEG,-1,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(ABS(NEG),0,0): GOSUB WRITE_ESITO
FESITO_BUF$=CIFLET$(NEG): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(NN,-1,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(FIX(NN,3),-1,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(NNN,-1,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(NNNN,-1,0): GOSUB WRITE_ESITO
FESITO_BUF$=CIFLETDEC$(NNNN,3): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(NN+NNNN,-1,0): GOSUB WRITE_ESITO
SETDECPIX 2
FESITO_BUF$=STR$(NN+NNNN,-1,0): GOSUB WRITE_ESITO
FESITO_BUF$=CHR$(83): GOSUB WRITE_ESITO

-----
'FESITO_BUF$=STR$(CAMBIOVAL(2,5,N,"20210101",5),-1,0): GOSUB WRITE_ESITO
'la Cambioval e' commentata poiche' dipende dalla configurazione dati, pertanto e' non operativa
NUMERO=126385.07227 : FESITO_BUF$=STR$(NUMERO,5,0): GOSUB WRITE_ESITO

-----
FESITO_BUF$=MRN$(N,5,5): GOSUB WRITE_ESITO ' Impaccamento binario

'Gli operatori ARITMETICI (per espressioni/concatenazioni di valori)
'formano con gli operatori LOGICI e DI RELAZIONE
'questa gerarchia decrescente di priorit  d'esecuzione
'NOT -negativo ^ * / MOD + -sottrazione < <= > = <> AND OR ( )

FINE1:
RETURN ' fine ETIC1

' sobroutine : spezza la stringa A$ e ritorna Array per quanti sono i ";"
ETIC2A:
I=1
CICLO1:
K=INSTR(A$,";")
IF K=0 THEN AR$(I)=A$ : GOTO SALTA
AR$(I)=LEFT$(A$,K-1) : A$=RIGHT$(A$,LEN(A$)-K) : I=I+1
IF INSTR(A$,";")=0 THEN AR$(I)=A$ : GOTO SALTA ELSE GOTO CICLO1
SALTA:
RETURN' fine ETIC2A

' sobroutine : elaborazioni alfanumeriche
ETIC2B:
A$=A1$+A2$ : B$=AR$(1)+AR$(2)+AR$(3)+AR$(4)+AR$(5)

```

```

FESITO_BUF$=A1$+" e "+A2$: GOSUB WRITE_ESITO
FESITO_BUF$=A$: GOSUB WRITE_ESITO
FESITO_BUF$=B$: GOSUB WRITE_ESITO
FESITO_BUF$=STR$(ASC(B$),0,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(LEN(B$),0,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(INSTR(B$,"MDS"),0,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(INSTR_CI(B$,AR$(2)),0,0): GOSUB WRITE_ESITO

' Verifica C.I. se A$ = B$
IF STRCMP_CI(A$,B$)=0 THEN FESITO_BUF$= "vero" ELSE FESITO_BUF$= "falso"
GOSUB WRITE_ESITO

FESITO_BUF$=STR$(VAL(AR$(5)),0,0): GOSUB WRITE_ESITO
FESITO_BUF$=STR$(CVN(IMP$,5,5),-1,0): GOSUB WRITE_ESITO

' Aggiunge 3 ASCII 0 a Sprix
MK_AR$=MKA$(AR$(1),8)
FESITO_BUF$= MK_AR$: GOSUB WRITE_ESITO

' Toglie tutti ASCII 0 a Sprix
CV_AR$=CVA$(MK_AR$,8)
FESITO_BUF$= CV_AR$: GOSUB WRITE_ESITO

FESITO_BUF$=CVMAIU$(B$): GOSUB WRITE_ESITO
FESITO_BUF$=CVMINUS$(B$): GOSUB WRITE_ESITO
FESITO_BUF$=STRING$(5,"Sprix"): GOSUB WRITE_ESITO
FESITO_BUF$=LEFT$(B$,12): GOSUB WRITE_ESITO
FESITO_BUF$=MID$(B$,6,7): GOSUB WRITE_ESITO
FESITO_BUF$=RIGHT$(B$,10): GOSUB WRITE_ESITO
FESITO_BUF$=LEFT$(LEFT$(B$,12)+STRING$(LEN(B$)," "),LEN(B$)): GOSUB WRITE_ESITO
'Estrae da dx e mantiene lung.B$
FESITO_BUF$=RIGHT$(STRING$(LEN(B$)," ")+RIGHT$(B$,10),LEN(B$)): GOSUB WRITE_ESITO

FINE2:
RETURN' fine ETIC2B

' subroutine
ETIC3:
FESITO_BUF$="date DATA1$ e DATA2$           = "+DATA1$+" e "+DATA2$: GOSUB WRITE_ESITO
FESITO_BUF$="data DATA1$ formattata         = "+STRDATA$(DATA1$,3,"/"): GOSUB WRITE_ESITO
FESITO_BUF$="giorni da DATA1$ a DATA2$     = "+STR$(DIFDATA(DATA1$,DATA2$),0,0): GOSUB WRITE_ESITO
FESITO_BUF$="giorno settimanale di DATA2$ = "+STR$(GIORSET(DATA2$),0,0): GOSUB WRITE_ESITO
                                           IF GIORSET(DATA2$)=7 THEN FESITO_BUF$= " (domenica)": GOSUB WRITE_ESITO
FESITO_BUF$="DATA2$ diminuita di 3 giorni = "+PIUDATA$(DATA2$,-3): GOSUB WRITE_ESITO
FINE3:
RETURN 'fine ETIC3

' subroutine
CRONOMETRO:
FOR I=1 TO 5: AZZCRONO 1
CICLO2: IF CRONO(1)<1 THEN GOTO CICLO2
        FESITO_BUF$=" Tempo in secondi = "+STR$(I,0,0): GOSUB WRITE_ESITO
NEXT

' controllo crono su 3 sec con 0,5 sec di tolleranza
AZZCRONO 1
SLEEP 3000
NMILLI=CRONO(1)*1000
IF NMILLI<2500 OR NMILLI>3500 THEN FESITO_BUF$="FALLITO CONTROLLO CRONO (3 SEC)" ELSE FESITO_BUF$="CONTROLLO CRONO (3 SEC) OK"
AZZCRONO 1
RETURN 'fine CRONOMETRO

' subroutine
SOSPENDO:
FESITO_BUF$= "        inizio PAUSA di 2,5 secondi": GOSUB WRITE_ESITO
SLEEP(2500)
FESITO_BUF$= "        termine PAUSA di 2,5 secondi": GOSUB WRITE_ESITO
FINESOSPENDO:
RETURN ' fine SOSPENDO

' subroutine
TEST FILE IO:
FESITO_BUF$="*** TEST FILE IO ***": GOSUB WRITE_ESITO

' TEST SU DIRECTORY E LEGGO FILE
ACCFILEUTE=1
DIRPROVA$="\temp\dirprova\"
SHELL "rmdir "+DIRPROVA$
SHELL "mkdir "+DIRPROVA$
NF=AUTOFILE()
FOR I=1 TO 5
    OPEN "R",NF,DIRPROVA$+STR$(I,0,0)+".txt",1:
    FIELD NF,1 AS BUF$: CLOSE NF
NEXT
LEGGIDIR DIRPROVA$, ""
FOR I=1 TO _NUMDIRFILE
    FESITO_BUF$=_DIRFILE$(I): GOSUB WRITE_ESITO
NEXT
FOR I=1 TO 5
    KILL DIRPROVA$+STR$(I,0,0)+".txt"
NEXT
SHELL "rmdir "+DIRPROVA$

' SCRIVO UN FILE SEQUENZIALE FORMATTATO
F$="\temp\file.txt"
KILL F$
CRLF$=CHR$(13)+CHR$(10)
NF=AUTOFILE() : OPEN "R",NF,F$,12:FIELD NF,10 AS REC$,2 AS CRLF$
FOR R=1 TO 9
    REC$=STRING$(R,STR$(R,0,0))
    PUT NF,R
NEXT
CLOSE NF

' LEGGO IL FILE SEQUENZIALE FORMATTATO
OPEN "R",NF,F$,12 : FIELD NF,10 AS REC$,2 AS CRLF$
NREC=LOF(NF)/12
FOR R=1 TO NREC

```

```

GET NF,R
FESITO_BUF$=CVAS(REC$,10): GOSUB WRITE_ESITO
NEXT

' AGGIUNGO DEI RECORD FORMATTATI
REC$="record 10" : PUT NF,10 : REC$="record 11" : PUT NF,11 : CLOSE NF
' AGGIUNGO RECORD (SOVRASCRIVO E PERDO FORMATTAZIONE -CRLF$-)
OPEN "R",NF,F$,12 : FIELD NF,10 AS REC$
REC$="SPRIX" : PUT NF,4 : REC$="COLLAGE" : PUT NF,7
REC$=" MDS" : PUT NF,12 : REC$=" SDK" : PUT NF,13
CLOSE NF

' LEGGO IL FILE NON FORMATTATO|(-record dimensione variabile-)
OPEN "R",NF,F$,1:FIELD NF,1 AS CAR$:DIMFILE=LOF(NF):RECORD$="":R=0
FOR C=1 TO DIMFILE
  GET NF,C
  IF CAR$<>CHR$(13) AND CAR$<>CHR$(10) THEN GOTO CONCATENA
  IF CAR$=CHR$(13) THEN GOTO PROSSIMOREC
  IF CAR$=CHR$(10) THEN GOTO VERIFICA
CONCATENA:
  RECORD$=RECORD$+CAR$ : GOTO PROSSIMOREC
VERIFICA:
  R=R+1
  IF INSTR(RECORD$,"333")<>0 THEN FESITO_BUF$="333|e' al record "+STR$(R,0): GOSUB WRITE_ESITO
  RECORD$=""
PROSSIMOREC:
NEXT
CLOSE NF

' CREO UNA COPIA DI BACKUP DEL FILE SEQUENZIALE E LO CANCELLO
IF ESISFILE(F$) THEN COPYFILE F$,"c:\temp\file_R_back.txt" : KILL F$

' CREO FILE A 2 CHIAVI CON: | Articolo, Descrizione, Categoria
F$="\temp\file.txt":FK1$="\temp\file1.k":F2K$="\temp\file2.k"
OPEN "K",NF,F$,FK1$;1;16,F2K$;41;10,52
FIELD NF,20 AS ART$ ALFA,20 AS DES$ ALFA
FIELD NF,10 AS CAT NUMO,2 AS CRLF$ ALFA
FOR I=1 TO 9
  I$=STR$(I,0)
  ART$="Articolo n."+I$: DES$="Descrizione "+I$: CAT=VAL(STRING$(I,I$))
  INSKEY NF : IF _ERRIND=1 THEN FESITO_BUF$="record "+STR$(I,0)+" gia' presente": GOSUB WRITE_ESITO
NEXT
CLOSE NF

' CERCO Articoli n.10 e n.5 PER CHIAVE PRIMARIA E LI MODIFICO
OPEN "K",NF,F$,FK1$;1;16,F2K$;41;10,52
FIELD NF,20 AS ART$ ALFA,20 AS DES$ ALFA,10 AS CAT NUMO,2 AS CRLF$ ALFA
KEY$="Articolo n.10": GETKEY NF,1,KEY$
IF _ERRIND=1 THEN FESITO_BUF$="articolo 10 cercato ma non trovato"
NEWKEY$="Articolo n.5": GETKEY NF,1,NEWKEY$
IF _ERRIND=0 THEN FESITO_BUF$="articolo 5 trovato, gli cambio descrizione": GOSUB WRITE_ESITO
NEWDES$=RIGHT$(STRING$(5,"NUOVO"),20):DES$=NEWDES$
VARKEY NF, NEWKEY$: IF _ERRIND=0 THEN FESITO_BUF$="Descrizione 5 variata": GOSUB WRITE_ESITO

' ORA LEGGO PROSSIMI RECORD E VARIO DESCRIZIONI
CICLO:
NEXTKEY NF,1: IF _ERRIND=1 THEN GOTO FINEFILE
KEYART$=LEFT$(ART$+STRING$(20,CHR$(0)),20)
DES$=NEWDES$:VARKEY NF, KEYART$
IF _ERRIND=1 THEN FESITO_BUF$="record (key "+KEYART$+" ) non variato": GOSUB WRITE_ESITO
GOTO CICLO
FINEFILE:
CLOSE NF:

' LEGGO A RITROSO PER CHIAVE NUMERICA SECONDARIA
CERCAT=4444: CERCAT$=RIGHT$(" " +STR$(CERCAT,0,0),10)
OPEN "K",NF,F$,FK1$;1;16,F2K$;41;10,52
FIELD NF,20 AS ART$ ALFA,20 AS DES$ ALFA,10 AS CAT NUMO,2 AS CRLF$ ALFA

FESITO_BUF$="chiavi totali: "+ STR$(LOFKEY(NF),0,0)
KEY$=RIGHT$(CERCAT$,10):GETKEY NF,2,KEY$
IF _ERRIND=0 THEN FESITO_BUF$="Numero Record Dati: "+STR$(_NRDATI,0,0): GOSUB WRITE_ESITO
CICLOPREV: PREVKEY NF,2: IF _ERRIND=1 THEN GOTO INIZFILE

FESITO_BUF$="Key Precedente "+STR$(CAT,0,0)+": con "+DES$: GOTO CICLOPREV
INIZFILE:

' ADESSO CERCA PER CATEGORIA (KEY2)
' SE TROVA VARIA, ALTRIMENTI INSERISCE NUOVO RECORD
' E CANCELLA IL RECORD CON KEY1 'Articolo n.5

VARIAZIONE:
CERCAT=4444: CERCAT$=RIGHT$(" " +STR$(CERCAT,0,0),10)
NEWCAT=0000

GETKEY NF,2,CERCAT$: OLDART$=ART$ 'memorizzo la key principale
IF _ERRIND=1 THEN TROV$="N" ELSE TROV$="S"

GOSUB CTRL_TEST_FILE: IF CTRL$<>"OK" THEN GOTO VARIAZIONE
IF TROV$="S" THEN VARKEY NF,OLDART$:CLOSE NF:GOTO CHECK
IF TROV$="N" THEN INSKEY NF: DELKEY NF, "Articolo n.5":CLOSE NF
'(VARKEY mantiene il format su editor rispetto a DELKEY+INKEY)

CHECK: IF _ERRIND<>0 THEN FESITO_BUF$="problemi in VAR/INS/DEL KEY": GOSUB WRITE_ESITO
RETURN

CTRL_TEST_FILE: ' effettuo controlli/modifiche...
CAT=VAL(" " +NEWCAT)
ART$="NEW ARTICOLO ????? ":DES$="NEW DESCRIZIONE ??? "
CTRL$="OK"
RETURN

```

Libreria libtest.ss

```

' Esegue la diff di due file :
' DIFF_FILE1$ = valorizzare con il pathname del 1o file
' DIFF_FILE2$ = valorizzare con il pathname del 2o file
' DIFF_VCOORD 0 esegue il controllo sulla dimensione dei 2 file
' prima di ciclare byte X byte e se diversi esce

```



```

'          subito; in questo modo non riesce a dare le coordinate
'          della differenza ma e' piu' veloce perche non ciclera'.
'          1 esegue sempre il controllo byte X byte riuscendo sempre
'          ad indicare le coordinate della la differenza trovata
' Ritorna DIFF_MSG$ se "" i due file sono uguali
'          altrimenti sono diversi e indica un msg di info
DIFF_FILE:
UGUALI = -1 ' 1=uguali 0=diversi -1=ancora da decidere
DIFF_MSG$ = ""
NF1 = AUTOFILE()
OPEN "R", NF1, DIFF_FILE1$, 1: FIELD NF1,1 AS BUF1$
NF2 = AUTOFILE()
OPEN "R", NF2, DIFF_FILE2$, 1: FIELD NF2,1 AS BUF2$

' Prima confronto la dimensione dei 2 file: se diversa -> files diversi
DIMF1 = LOF( NF1 ): DIMF2 = LOF( NF2 )
' se DIFF_VCOORD=0 e i file sono != allora salto il ciclo byte x byte
IF DIFF_VCOORD=0 AND DIMF1<>DIMF2 THEN UGUALI = 0: GOTO DIFF_FILE_FINE

CTREC=0 ' conta i caratteri letti
DIFF_CTRIGHE = 1 ' conta le righe (LF) lette

DIFF_FILE_CICLAFI: ' inizio ciclo lettura files
CTREC = CTREC+1
IF CTREC>DIMF1 OR CTREC>DIMF2 THEN GOTO DIFF_FILE_FINE ' sono a fine file
GET NF1,CTREC: GET NF2,CTREC
IF BUF1$ = CHR$(10) THEN DIFF_CTRIGHE=DIFF_CTRIGHE+1
IF BUF1$ <> BUF2$ THEN UGUALI = 0: GOTO DIFF_FILE_FINE ELSE GOTO DIFF_FILE_CICLAFI
' fine ciclo lettura files
DIFF_FILE_FINE:

' se UGUALI=-1 :
' il confronto byte x byte fino a qui non ha dato differenze
' pero' se uno dei 2 file e' piu grande dell'altro sono diversi
IF UGUALI=-1 AND DIMF1=DIMF2 THEN UGUALI = 1 ELSE UGUALI=0

IF UGUALI=0 AND DIFF_VCOORD=0 THEN DIFF_MSG$="FILE DIVERSI"
IF UGUALI=0 AND DIFF_VCOORD=1 THEN DIFF_MSG$="FILE DIVERSI (RIGA="+STR$(DIFF_CTRIGHE,0,0)+" BYTE="+STR$(CTREC,0,0)+")"

CLOSE NF1
CLOSE NF2

RETURN

```

```

' apre il creazione il file indicato da NOMEF_ESITO$ che va
' precedentemente preparata. (vedi WRITE_ESITO e CLOSE_ESITO)
OPEN_ESITO:
IF NOMEF_ESITO$="" THEN NOMEF_ESITO$="c:\temp\esito.txt"
KILL NOMEF_ESITO$

```

```

NFESITO=AUTOFILE(): CTRESITO=1
OPEN "R",NFESITO,NOMEF_ESITO$,82
FIELD NFESITO, 80 AS FESITO_BUF$, 2 AS FESITO_CRLF$
FESITO_CRLF$=CHR$(13)+CHR$(10)
RETURN

```

```

CLOSE_ESITO:
CLOSE NFESITO
RETURN

```

```

' Scrive la var FESITO_BUF$ portandola ad 80 byte sul file di esito
WRITE_ESITO:
LFESITO_BUF = LEN(FESITO_BUF$)
IF LFESITO_BUF > 80 THEN FESITO_BUF$=LEFT$(FESITO_BUF$,80)
IF LFESITO_BUF < 80 THEN FESITO_BUF$=FESITO_BUF$ + STRING$(80-LFESITO_BUF," ")
PUT NFESITO,CTRESITO: CTRESITO=CTRESITO+1
RETURN

```

Inserire nel percorso "C:\temp" il file dal nome "esito.txt" e con il seguente contenuto a lunghezza fissa di 80 caratteri

```

*** TEST NUMERICHE ***
252.000 e 770,14453
252770,14453
-252770,14453
252770
DUECENTOCINQUANTADUEMILASETTECENTOSETTANTA
391743,169993
391743,169
391743
391743,17
TRECENTONOVANTUNOMILASETTECENTOQUARANTATRE/170
783486,339993
783486,33
S
126385,07227
...ã ¡µ
*** TEST STRINGHE ***
la lib2 spezza la stringa "Sprix;Collage;MDS;SDK;2009"
Sprix
Collage
MDS
SDK
2009
Sprix;Collage; e MDS;SDK;2009
Sprix;Collage;MDS;SDK;2009
SprixCollageMDS SDK2009
83
22
13
6
falso
2009
0
Sprix
Sprix
SPRIXCOLLAGEMDSSDK2009
sprixcollagemdssdk2009
SprixSprixSprixSprixSprix

```

```
SprixCollage
Collage
MDSSDK2009
SprixCollage
MDSSDK2009
*** TEST DATA E ORA ***
date DATA1$ e DATA2$      = 20090303 e 20090308
data DATA1$ formattata     = 03/03/2009
giorni da DATA1$ a DATA2$ = -5
giorno settimanale di DATA2$ = 7
(domenica)
DATA2$ diminuita di 3 giorni = 20090305
Tempo in secondi = 1
Tempo in secondi = 2
Tempo in secondi = 3
Tempo in secondi = 4
Tempo in secondi = 5
    inizio PAUSA di 2,5 secondi
    termine PAUSA di 2,5 secondi
*** TEST ORDINAMENTO ARRAY DI STRINGHE ***
ANDREA
LUCA
ROBERTO
STEFANO
STEFANO
ORDINATI 5 ELEMENTI
*** TEST FILE IO ***
1.txt
2.txt
3.txt
4.txt
5.txt
1
22
333
4444
55555
666666
7777777
88888888
999999999
333|e' al record 3
articolo 5 trovato, gli cambio descrizione
articolo 5 trovato, gli cambio descrizione
Numero Record Dati: 4
```

Esempio: utilizzo del file di Libreria in Sprix con #APPENDISPX

Quando si utilizza frequentemente una porzione di codice è conveniente inserirlo in un file di Libreria (non compilabile/eseguibile) che può essere richiamato in altri programmi. Dal momento che tale codice di Libreria viene richiamato deve necessariamente iniziare con una etichetta e terminare con l'istruzione RETURN per restituire il controllo al programma chiamante. Analogamente, il programma chiamante deve utilizzare l'istruzione GOSUB (associata alla RETURN) per chiamare la Libreria. La libreria deve essere dichiarata in testa al programma chiamante con l'istruzione #APPENDISPX, cioè, deve essere collocata nella prima riga in assoluto del programma e che il nome del file del codice richiamato sia tra i simboli <...> e senza estensione. La compilazione accoda il codice di Libreria nell'eseguibile del programma chiamante, per questa ragione è consigliato terminare il programma chiamante con istruzione END. Esempio:

```
(File chiamante da installare in un menù diverso da Libreria)
#APPENDISPX 'Riferimento alla libreria "spx906.ss"
'Sprix che lancia la libreria "spx906.ss"
MESS$="IL QUADRATO DI UN NUMERO E' SEMPRE NON NEGATIVO!!!"
INTER=0
N_FIN=AUTOFIN()
AFRIFIN N_FIN,1,0,3,15,4,50
ATTIVAFIN N_FIN
LOCATE 1,1
COLOR 3
PRINT "<<<<<<<<>>>>>>>>"
'INPUT DEL DEL QUADRATO DEL NUMERO DA RICERCARE...
INPUT_NUM:
SETTF "501,505"
LOCATE 3,1
COLOR 0
PRINT "QUADRATO DEL NUMERO(>0): ";
LOCATE 3,25
INPUT "N6",NUM,15
IF _TF=505 THEN GOTO FINE
IF _TF=501 THEN GOSUB LIB_RAD_Q
IF RAD_Q=-1 THEN VIMSG 1,MESS$ ELSE VIMSG 1,STR$(RAD_Q,6,0)
GOTO INPUT_NUM
FINE:
CHIUDIFIN N_FIN
END
```

(File libreria da installare nel menù Libreria)

Questa libreria serve a calcolare la radice quadrata di un numero. 'Questa funzione si rende necessaria in quanto l'operatore "^" accetta esponenti a partire da 1, mentre la radice quadrata e' 1/2. Per stimare la radice quadrata ho usato il metodo di bisezione che permette di stimare una funzione continua in un'intervallo [a,b] con un margine di errore TOLL (assunto pari a 0.00001). Tale libreria restituisce RAD_Q. Se l'utente inserisce un numero negativo la libreria non esegue la radice e restituisce RAD_Q=-1.

```
LIB_RAD_Q:
I=0
TOLL=0.000001
INTERVALLO:
'RICERCA DELL'INTERVALLO [LIM_INF,LIM_SUP]
IF NUM<0 THEN RAD_Q=-1:RETURN
IF I^2=NUM THEN RAD_Q=I: RETURN
IF I^2>NUM THEN LIM_INF=I-1:LIM_SUP=I+1:GOTO RADICE
I=I+1
GOTO INTERVALLO
RADICE:
'RICERCA DELLA RADICE QUADRATA DI NUM
RAD_Q=(LIM_INF+LIM_SUP)/2
IF (RAD_Q^2=NUM) OR (ABS(LIM_SUP-LIM_INF)/2 IF RAD_Q^2 I=I+1)
GOTO RADICE
```

Caso: CHR\$() in abbinamento con altre istruzioni

Prestare attenzione all'uso di CHR\$ abbinato all'istruzione PRINT in quanto il risultato a video è condizionato da vari fattori esterni (terminale, font, ecc...). Per questa ragione sono previste apposite funzioni per sostituirlo in determinate casistiche, per esempio la PRINTF per disegnare fincature i finestre e l'istruzione BEEP per emettere un segnale acustico.

In caso di stampa su file testo, può essere necessario dovere andare a capo. L'utilizzo dei caratteri speciali Carriage Return + Line Feed [CRLF\$=CHR\$(13)+CHR\$(10)] consente di andare a capo, ma in caso di stampa in pdf possono generarsi dei disallineamenti. Nella stampa pdf, infatti, il carattere speciale CHR\$(10) è interpretato come impostazione del carattere espanso e causa l'aumento della dimensione dei caratteri successivi; sono permessi i codici speciali dal CHR\$(1) a CHR\$(11) per la formattazione del carattere di stampa, per cui, per quanto riguarda il carattere CHR\$(13) non accade alcunché poiché filtrato.

Caso: caratteri estesi validi

I presenti codici sono accettati nei soli campi del gestionale che accettano tali valori.
Elenco dei caratteri estesi accettati dalla funzione CHR\$():

_Euro	128	AMaiusAccentoAcuto	193	aMinusAccentoAcuto	225
_ApiceChiuso	146	AMaiusCirconflesso	194	aMinusCirconflesso	226
_PuntoEsclamativoInverso	161	AMaiusTilde	195	aMinusTilde	227
_Cent	162	AMaiusDieresi	196	aMinusDieresi	228
_Lira	163	AMaiusGrado	197	aMinusGrado	229
_Yen	165	_AeMauscolo	198	_AeMinuscolo	230
_BarraVerticaleInterrotta	166	CMaiusCediglia	199	cMinusCediglia	231
_Sezione	167	EMaiusAccentoGrave	200	eMinusAccentoGrave	232
_Dieresi	168	EMaiusAccentoAcuto	201	eMinusAccentoAcuto	233
_Copyright	169	EMaiusCirconflesso	202	eMinusCirconflesso	234
_OrdinaleFemminile	170	EMaiusDieresi	203	eMinusDieresi	235
_VirgoletteDoppieAperte	171	IMaiusAccentoGrave	204	iMinusAccentoGrave	236
_NegazioneLogica	172	IMaiusAccentoAcuto	205	iMinusAccentoAcuto	237
_TrattinoNascosto	173	IMaiusCirconflesso	206	iMinusCirconflesso	238
_Registrato	174	IMaiusDieresi	207	iMinusDieresi	239
_Macron	175	_Eth1	208	_Eth2	240
_Grado	176	NMaiusTilde	209	nMinusTilde	241
_PiuMeno	177	OMaiusAccentoGrave	210	oMinusAccentoGrave	242
_Esponente2	178	OMaiusAccentoAcuto	211	oMinusAccentoAcuto	243
_Esponente3	179	OMaiusCirconflesso	212	oMinusCirconflesso	244
_Mu	181	OMaiusTilde	213	oMinusTilde	245
_Paragrafo	182	OMaiusDieresi	214	oMinusDieresi	246
_PuntoCentrale	183	_Moltiplicazione	215	_Divisione	247
_Cediglia	184	_OBarratoMaiuscolo	216	_OBarratoMinuscolo	248
_Esponente1	185	UMaiusAccentoGrave	217	uMinusAccentoGrave	249
_OrdinaleMaschile	186	UMaiusAccentoAcuto	218	uMinusAccentoAcuto	250
_VirgoletteDoppieChiuse	187	UMaiusCirconflesso	219	uMinusCirconflesso	251
_UnQuarto	188	UMaiusDieresi	220	uMinusDieresi	252
_UnMezzo	189	YMaiusAccentoAcuto	221	yMinusAccentoAcuto	253
_TreQuarti	190	_Thorn2	222	_Thorn1	254
_PuntoInterrogativoInverso	191	_Beta	223		
AMaiusAccentoGrave	192	aMinusAccentoGrave	224		

Esempio: CALL per eseguire un collage che richiama un altro collage

L'istruzione CALL permette di richiamare una subroutine di un collage da un altro collage. Il parametro definisce il nome del file collage da chiamare. Tale nome deve essere dichiarato senza estensione e senza direttorio, inoltre il file deve essere posizionato nel direttorio degli Sprix attualmente in uso. Il parametro definisce l'etichetta, senza il carattere finale due punti, che si vuole chiamare. Per maggiori informazioni sull'istruzione CALL e sulle altre istruzioni associate ad essa (STARTCOLLAGE, STOPCOLLAGE, ecc...).

Esempio:

(Collage chiamante)

```
ON 612_MDTDOC:
N_FILE$="spxma"
STARTCOLLAGE N_FILE$
ETICK$="ON_IN_MADSART"
CALL N_FILE$, ETICK$
STOPCOLLAGE N_FILE$
END
```

(Collage chiamato: spxma.es)

```
ON IN MADSART:
VIMSG 1,"COLLAGE ATTIVATO CON SUCCESSO!!!"
END
```

Esempio DIM: sintassi di 3 dimensionamenti di array numerico 'N' e stringa 'A\$'

	una dimensione	due dimensioni	tre dimensioni
Automatici:	DIM N()	DIM N(,10)	DIM N(,10,20)
Non automatici:	DIM A\$(5)	DIM A\$(5,10)	DIM A\$(5,10,20)

Esempio DIM:

asigna 5 valori (1+1, 2+1, 3+1, 4+1,5+1) che assume l'indice ai 5 elementi che automaticamente ne conseguono ed a cui accedere e gestire con indice diretto (N(3)=4), indice variabile (X=2: N(X)=3) o ciclico (con indice X da 1 a 5 per ottenere N(1)=2, N(2)=3, N(3)=5, N(4)=5, N(5)=6):

```
DIM A$(3): A$(1)="pass" 'assegno al 1° di 3 elementi la stringa "pass"
DIM N(): AZZARRAY N: FOR I=1 TO 5: N(I)=I+1: NEXT
```

Esempio DIM con due librerie**Sprix:**

```
#APPENDISPX <libdic>
#APPENDISPX <libprog>
```

```
GOSUB RT_MAIN
VIMSG 1, "Fine programma"
END
```

Libreria libdic:

```
' dimensiona array automatico a 2 dimensioni
DIM A$( , 2)
```

Libreria libprog:

```

RT_MAIN:
  I = 1
  ' esegue ciclotabellare con selezione range di articoli ordinati per descrizione
+-----+
|Cod.tab.:ar MAGAZZINO - ANAGRAFICA ARTICOLO |
|Ordine 3:ardes() |
|Scelte... |
| arcod() >= "COD001" AND arcod() <= "COD006" |
+-----+
ar  A$(I, 1) = arcod()
ar  A$(I, 2) = ardes()
ar  I = I + 1
ar  I = I - 1
    R = 3
    C = 20
    FOR X = 1 TO I
      LOCATE R + X, C : PRINT A$(X, 1) + " - " + A$(X, 2)
      'INPUT "A", B$, 1
      INPUT "C", B$, 1
    NEXT
RETURN
  
```

Esempio IF:

```

IF VA$="+" THEN TOT=IMPOR1+IMPOR2 ELSE TOT=IMPOR1-IMPOR2
IF IF VA$="X" THEN IF B=0 THEN N=2 ELSE N=3
IF VA$="X" THEN N=1 ELSE IF VA$="Y" THEN N=2 ELSE N=3
IF N<1 THEN GOSUB SUB0 ELSE IF N<2 THEN GOSUB SUB1
  
```

Esempio IFF:

```

IFF A = 1
  VIMSG 4, "Sono la prima IFF"
  B = B + 1
  VAR$ = "Ciao"
ELSEF
  IFF A = 2
    VIMSG 4, "Sono il primo ELSEF"
  ELSEF
    VIMSG 4, "Sono il secondo ELSEF"
  C = C + 1
  ENDF
ENDF
  
```

Esempio GOTO:

```

X=0 : INIZIO: X=X+1: VISUALIZZA_X: IF X=3 THEN GOTO FINE ELSE GOTO INIZIO : FINE: END
  
```

Esempio GOSUB:

```

VA$="var alfa": FOR I=1 TO 3 .... NEXT
IF A=1 THEN N=10: GOSUB ETIC1 ELSE ...
GOTO FINE
ETIC1: RETURN : FINE: END
  
```

Esempio FOR-NEXT:

```

FOR <varnum1>=<x> TO <y> [STEP <z>]
  ISTRUZIONE1...
FOR <varnum2>=<x> TO <k> [STEP <z>]
  ISTRUZIONE2...
NEXT
NEXT
DIM C3(10,5)
FOR C1=1 TO 10
GOSUB UNO
  FOR C2=5 TO 1 STEP -1
GOSUB DUE : GOSUB TRE...
C3(C1,C2)=C1*C2: stampa C3(C1,C2)
NEXT
NEXT
END
  
```

Chiusura di Ciclo FOR in caso di STEP di avanzamento a frazioni di unità (es. 0,1)

```

DA_A=10 : A B=11
FOR I=DA_A TO A B STEP 0.1
  VIMSG 5, STR$(I,-1,0)
NEXT
'....

DA_A=19 : A B=20
TDA A=DA_A*10 : TA B=A B*10
FOR I=TDA_A TO TA B STEP 1
  VIMSG 5, STR$(I/10,-1,0)
NEXT
  
```

Esempio SORT:

```

_LRECSORT=8
PUTSORT"SPRIX 2": PUTSORT"SPRIX 6": PUTSORT"SPRIX 5": PUTSORT"SPRIX 1"
PUTSORT" ddd ": PUTSORT" bbb ": PUTSORT" aaa ": PUTSORT" ccc "
LEGGI:
GETSORT NOMEPR$: PRINT NOMEPR$: VIMSG 1, NOMEPR$
IF NOMEPR$="" THEN GOTO FINE ELSE GOTO LEGGI
FINE: VIMSG 4, " fine ": END
  
```

Esempi di ordinamento con le istruzioni PUT/GETSORT

Esempio con file esterno:

```

|'INIZIALIZZAZIONE DELLE VARIABILI
|N_FILE=AUTOFILE()
|NOMEFILE$="C:\MIOFILE.TXT"
|
|'CREAZIONE E SCRITTURA DEL FILE ESTERNO
|IF ESISFILE(NOMEFILE$)=1 THEN KILL NOMEFILE$
|OPEN "R",N_FILE,NOMEFILE$,42
|FIELD N_FILE,12 AS COD$
|FIELD N_FILE,28 AS DESCR$
|FIELD N_FILE,2 AS EOL$
|EOL$=CHR$(13)+CHR$(10)
|+-----+
||Cod.Tab.:ar ARTICOLO - ANAGRAFICA |
||Scelte..: |
|| artip()="A" |
|+-----+
ar |COD$=arcod()
ar |DESCR$=ardes()
ar |N_REC=N_REC+1
ar |PUT N_FILE,N_REC
|
|'ORDINAMENTO (CON GET/PUTSORT) E STAMPA
|IF N_REC<=0 THEN END
|_LRECSORT=40 'SOMMA DELLE DIMENSIONI DI CODICE E DESCRIZIONE
|FOR I=1 TO N_REC
|GET N_FILE,I
|PUTSORT DESCR$+COD$
|NEXT
N|CODICE DESCRIZIONE
N|
|FOR I=1 TO N_REC
|GETSORT STRI$
N|~RIGHT$(STRI---~LEFT$(STRI$,28)\\\\\\\\\\\\\\\\\\\\
|NEXT
|CLOSE N_FILE
|END

```

Esempio con Ciclo Tabellare:

```

|'LETTURA ORDINATA CON GETSORT DEGLI ARTICOLI DI MAGAZZINO
|L'ORDINAMENTO AVVIENE RISPETTO A ardes()+arcod()
|
|DIM_ART=12 '<---[Dimensione codice articolo]
|DIM_DESCR=28 '<---[Dimensione descrizione articolo]
|
|FOR I=1 TO DIM_DESCR
|SPAZI$=SPAZI$+" "
|NEXT
|_LRECSORT=DIM_ART+DIM_DESCR
|+-----+
||Cod.Tab.:ar ARTICOLO - ANAGRAFICA |
||Scelte..: |
|| artip()="A" |
|+-----+
ar |COD$=LEFT$(ardes()+SPAZI$,DIM_DESCR)
ar |PUTSORT COD$+arcod()
ar |N_REC=N_REC+1
|
N|Codice --- Descrizione
N|
|
|FOR I=1 TO N_REC
|GETSORT STRI$
N|~arcod(RIGHT --- ~ardes(RIGHT$(STRI$,12)\\\\\\
|NEXT
|END

```

Esempio di applicazione PassBuilder con elementi di Libreria e [sequenza incrementale di uno Sprix in 10 step](#) che prevede una maschera di input, letture ciclate di anagrafiche articoli e gestione di file sequenziale, ordinamenti e stampa.

Codice	Nome	Versione	Tipo	Stato
904702COSPXB1801	Es. Applicazione Sprix+Libreria	1.0.0	<input type="radio"/> App Utente	● Ok
904702GIOVA	Applicazione in Modifica	1.0.0	<input type="radio"/> App Utente	● <input type="radio"/> Modificata - Disabilita
912709LABORDTK	Applicazione Datatrek	1.0.0	<input type="radio"/> App Utente	● Modificata
924148CORSO1017	Applicazione Sprix Mobile 2017	1.0.2	<input type="radio"/> App Utente	● Ok
957660APPALE	Applicazione maschera video	1.0.0	<input type="radio"/> App Utente	● Modificata

App "Es. Applicazione Sprix+Libreria"

Nome	Tipo	Nome file
Funzioni di utilita'	Libreria Sprix	lb_utils.ss
Step 01. Input mask	Sprix	● Menu' Azienda spx01.es - s
Step 02. Lettura Articoli	Sprix	● Menu' Azienda spx02.es - s
Step 03. Lettura Articoli Migl.	Sprix	● Menu' Azienda spx03.es - s
Step 04. Output mask	Sprix	● Menu' Azienda spx04.es - s
Step 05. Creaz. file seq vuoto	Sprix	● Menu' Azienda spx05.es - s
Step 06. Popolo file seq	Sprix	● Menu' Azienda spx06.es - s
Step 07. Leggo file seq	Sprix	● Menu' Azienda spx07.es - s
Step 08. File seq a video	Sprix	● Menu' Azienda spx08.es - s
Step 09. File seq a video riord.	Sprix	● Menu' Azienda spx09.es - s
Step 10. Stampa	Sprix	● Menu' Azienda spx10.es - s

Libreria

```

* *****
* Nome File .....: lb_utils.ss
* Descrizione ....: RaCcolta di procedure di Utilita' generale.
*
* Questa libreria contiene tutta una serie di funzioni di natura generica che possono essere
* utilizzate nei propri programmi Sprix. Prima di ciascuna procedura, si trova una breve descri-
* zione per l'utilizzo.
* Tutte le variabili utilizzate dalla libreria iniziano con suffisso MUT,seguite poi da I per le
* variabili di input, da O per le variabili di output (risultato) o da W per le variabili di
* lavoro interne.
* Le procedure iniziano tutte con suffisso PRCUT_
* Prima di poter utilizzare le procedure della libreria, e' necessario chiamare la procedura di
* inizializzazione : PRCUT_INITIALIZE
*
* Elenco procedure disponibili:
  
```

```

' * -----*
' * PRCUT_GETFILENAMEANDEXT *
' * -----*
' *****
PRCUT_INITIALIZE:
' Variabili di lavoro interne della libreria
MUTW_I1 = 0
MUTW_LVL = 0
MUTW_SLEN1 = 0
MUTW_FNSEP$ = "" 'Separatore livelli path
MUTW_STR2$ = ""
MUTW_SLEN2 = 0
MUTW_DPOINTS$ = ""
DIM MUTW_STR1$()

IF sxsys = "W" THEN MUTW_FNSEP$ = "\" ELSE MUTW_FNSEP$ = "/"
RETURN
' -----
' PRCUT_GETFILENAMEANDEXT - Estrazione del nomefile e dell'estensione da un nomefile qualificato
' con un fullpath.
' Variabili da valorizzare:
' MUTI_FPNAME$ - Nome del file completo di path
' MUTO_FNAME$ - Nome del file
' MUTO_FEXT$ - Estensione del file
PRCUT_GETFILENAMEANDEXT:
MUTO_FNAME$ = ""
MUTO_FEXT$ = ""
MUTW_I1 = 1
MUTW_LVL = 1
MUTW_SLEN1 = 0

AZZARRAY MUTW_STR1$

PRCUT_LEVEL_LOOP:
MUTW_I1 = INSTR(MUTI_FPNAME$, MUTW_FNSEP$)
IF MUTW_I1 = 0 THEN MUTW_STR1$(MUTW_LVL) = MUTI_FPNAME$ : GOTO PRCUT_END_LEVEL_LOOP
MUTW_STR1$(MUTW_LVL) = LEFT$(MUTI_FPNAME$, MUTW_I1 - 1)
MUTI_FPNAME$ = RIGHT$(MUTI_FPNAME$, LEN(MUTI_FPNAME$) - MUTW_I1)
MUTW_LVL = MUTW_LVL + 1
GOTO PRCUT_LEVEL_LOOP
PRCUT_END_LEVEL_LOOP:
MUTW_SLEN1 = LEN(MUTW_STR1$(MUTW_LVL))
MUTW_I1 = INSTR(MUTW_STR1$(MUTW_LVL), ".")
IF MUTW_I1 <> 0
MUTO_FNAME$ = LEFT$(MUTW_STR1$(MUTW_LVL), MUTW_I1 - 1)
MUTO_FEXT$ = RIGHT$(MUTW_STR1$(MUTW_LVL), MUTW_SLEN1 - MUTW_I1)
ELSEIF
MUTO_FNAME$ = MUTW_STR1$(MUTW_LVL)
MUTO_FEXT$ = ""
ENDIF
RETURN
' -----
' PRCUT_NUM2STR - Convertire il numero fornito in una stringa di lunghezza preimpostata. Se il numero
' e' piu' piccolo, il numero in formato stringa risultante viene fillato con degli
' zero per i digit privi di significato.
' Variabili da valorizzare:
' MUTI_NUM - Numero da convertire
' MUTI_LEN - Lunghezza in digit del numero convertito
' MUTO_NUM$ - Numero convertito in stringa
' MUTO_ERR$ - Esito dell'operazione.
PRCUT_NUM2STR:
AZZARRAY MUTW_STR1$
MUTO_NUM$ = ""
MUTW_STR2$ = STR$(MUTI_NUM, 0, 0)
MUTW_SLEN2 = LEN(MUTW_STR2$)
MUTW_SLEN1 = MUTI_LEN

IF MUTI_LEN < MUTW_SLEN2 THEN MUTO_ERR$ = "La lunghezza fornita per la conversione in stringa e' inferiore alla lunghezza del numero da
convertire. Impossibile fare la conversione." : RETURN

FOR MUTW_I1 = 1 TO MUTW_SLEN1
MUTW_STR1$(MUTW_I1) = "0"
NEXT

FOR MUTW_I1 = MUTW_SLEN2 TO 1 STEP -1
MUTW_STR1$(MUTW_SLEN1) = MID$(MUTW_STR2$, MUTW_I1, 1)
MUTW_SLEN1 = MUTW_SLEN1 - 1
NEXT

FOR MUTW_I1 = 1 TO MUTI_LEN
MUTO_NUM$ = MUTO_NUM$ + MUTW_STR1$(MUTW_I1)
NEXT

MUTO_ERR$ = ""
RETURN
' -----
' PRCUT_VALIDATEFILENAME - Verifica il tipo di sistema operativo e aggiusta la forma del filepath.
' Variabili da valorizzare:
' MUTI_FPNAME$ - Nome del file completo di path
' MUTO_FPNAME$ - Nome del file completo di path validato
PRCUT_VALIDATEFILENAME:
AZZARRAY MUTW_STR1$
MUTO_FPNAME$ = ""

MUTW_SLEN1 = LEN(MUTI_FPNAME$)
FOR MUTW_I1 = 1 TO MUTW_SLEN1
MUTW_STR1$(MUTW_I1) = MID$(MUTI_FPNAME$, MUTW_I1, 1)
NEXT

IF sxsys = "W"
FOR MUTW_I1 = 1 TO MUTW_SLEN1
IF MUTW_STR1$(MUTW_I1) = "/" THEN MUTW_STR1$(MUTW_I1) = "\"
NEXT
ELSEIF
FOR MUTW_I1 = 1 TO MUTW_SLEN1
IF MUTW_STR1$(MUTW_I1) = "\" THEN MUTW_STR1$(MUTW_I1) = "/"
NEXT
ENDIF
ENDEF

```

```

FOR MUTW_I1 = 1 TO MUTW_SLEN1
MUTO_FFNAME$ = MUTO_FFNAME$ + MUTW_STR1$(MUTW_I1)
NEXT
RETURN
-----
' PRCUT_DEC2STR - Convertire il numero fornito in una stringa di lunghezza preimpostata. Se il numero
' e' piu' piccolo, il numero in formato stringa risultante viene fillato con degli
' zero per i digit privi di significato.
' Variabili da valorizzare:
' MUTI_NUM - Numero da convertire
' MUTI_LEN - Lunghezza in digit della parte intera del numero convertito
' MUTI_DLEN - Lunghezza in digit della parte decimale del numero convertito
' MUTI_DPOINT$ - Carattere di separazione decimale (. o ,)
' MUTI_NOSIGN$ - Se impostato a 'S' sopprime l'espressione del segno + / -
' MUTO_NUM$ - Numero convertito in stringa
' MUTO_ERR$ - Esito dell'operazione.
PRCUT_DEC2STR:
AZZARRAY MUTW_STR1$
MUTO_NUM$ = ""
MUTW_STR2$ = STR$(MUTI_NUM, MUTI_DLEN, 0)
MUTW_SLEN2 = LEN (MUTW_STR2$)
MUTW_DPOINT$= ""

IFF MUTI_DPOINT$ <> ""
IF MUTI_DPOINT$ <> "." AND MUTI_DPOINT$ <> "," THEN MUTO_ERR$ = "Carattere di separazione decimale non valido. I soli simboli
ammessi sono '.' o ',' : RETURN
ENDIF
IF MUTI_NOSIGN$ <> "S" AND MUTI_NOSIGN$ <> "N" THEN MUTO_ERR$ = "Flag di soppressione segno non impostato correttamente. I soli valori
ammessi sono 'S' o 'N'. " : RETURN

IFF MUTI_NOSIGN$ = "N" AND MUTI_DPOINT$ <> ""
MUTW_SLEN1 = MUTI_LEN + MUTI_DLEN + 2
ENDIF
IFF MUTI_NOSIGN$ = "N" AND MUTI_DPOINT$ = ""
MUTW_SLEN1 = MUTI_LEN + MUTI_DLEN + 1
ENDIF
IFF MUTI_NOSIGN$ = "S" AND MUTI_DPOINT$ <> ""
MUTW_SLEN1 = MUTI_LEN + MUTI_DLEN + 1
ENDIF
IFF MUTI_NOSIGN$ = "S" AND MUTI_DPOINT$ = ""
MUTW_SLEN1 = MUTI_LEN + MUTI_DLEN
ENDIF

IF (MUTI_LEN+MUTI_DLEN) < MUTW_SLEN2 THEN MUTO_ERR$ = "La lunghezza fornita per la conversione in stringa e' inferiore alla lunghezza
del numero da convertire. Impossibile fare la conversione." : RETURN

FOR MUTW_I1 = 1 TO MUTW_SLEN1
MUTW_STR1$(MUTW_I1) = "0"
NEXT

IFF MUTI_NOSIGN$ = "N"
'IF MUTI_NUM >= 0 THEN MUTW_STR1$(1) = "+" : MUTW_I2 = 1 ELSE MUTW_STR1$(1) = "-" : MUTW_I2 = 2
IF MUTI_NUM >= 0 THEN MUTW_STR1$(1) = "+" ELSE MUTW_STR1$(1) = "-"
ENDIF

'MUTW_STR1$(MUTI_LEN + 2) = MUTI_DPOINT$
IFF MUTI_DPOINT$ <> ""
MUTW_STR1$(MUTW_SLEN1 - MUTI_DLEN) = MUTI_DPOINT$
ENDIF

MUTW_STR2$ = STR$(MUTI_NUM, 0, 0)
MUTW_SLEN2 = LEN (MUTW_STR2$)
MUTW_I2 = MUTW_SLEN1

'FOR MUTW_I1 = MUTW_SLEN2 TO MUTW_I2 STEP -1
' MUTW_STR1$(MUTW_SLEN1 - MUTI_DLEN - 1) = MID$(MUTW_STR2$, MUTW_I1, 1)
' MUTW_SLEN1 = MUTW_SLEN1 - 1
'NEXT
FOR MUTW_I1 = MUTW_SLEN2 TO 1 STEP - 1
IFF MUTI_DPOINT$ <> ""
MUTW_STR1$(MUTW_I2 - MUTI_DLEN - 1) = MID$(MUTW_STR2$, MUTW_I1, 1)
MUTW_I2 = MUTW_I2 - 1
ELSEIF
MUTW_STR1$(MUTW_I2 - MUTI_DLEN) = MID$(MUTW_STR2$, MUTW_I1, 1)
MUTW_I2 = MUTW_I2 - 1
ENDIF
NEXT

MUTW_STR2$ = STR$(MUTI_NUM, MUTI_DLEN, 0)
MUTW_SLEN2 = LEN (MUTW_STR2$)

'MUTW_SLEN1 = MUTI_LEN + 3
IF INSTR(MUTW_STR2$, ",") <> 0 THEN MUTW_DPOINT$ = ","
IF INSTR(MUTW_STR2$, ".") <> 0 THEN MUTW_DPOINT$ = "."
IF MUTW_DPOINT$ = "" THEN MUTO_ERR$ = "Separatore decimali non trovato. Il numero passato sembra un intero. Utilizzare la procedura
PRCUT_NUM2STR." : RETURN

MUTW_I2 = MUTW_SLEN1 - MUTI_DLEN + 1
FOR MUTW_I1 = INSTR(MUTW_STR2$, MUTW_DPOINT$) + 1 TO MUTW_SLEN2
MUTW_STR1$(MUTW_I2) = MID$(MUTW_STR2$, MUTW_I1, 1)
MUTW_I2 = MUTW_I2 + 1
NEXT

'MUTW_SLEN1 = MUTI_LEN + MUTI_DLEN + 2
FOR MUTW_I1 = 1 TO MUTW_SLEN1
MUTO_NUM$ = MUTO_NUM$ + MUTW_STR1$(MUTW_I1)
NEXT

MUTO_ERR$ = ""
RETURN

```

Step 01.

```

' *****
' * Nome File .....: spx01.ss *
' * Descrizione ....: Gestione richiesta dati da video *
' * * * * *
' * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica. *

```



```

' *          Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0.          *
' *          La quantita' la vogliamo tenere a 6 cifre.          *
' *          *****          *

RT_MAIN:
  GOSUB  RT_INIT
  GOSUB  RT_GETFILENAME
END
-----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  KEY$     = ""
  NRITEMS  = 0
RETURN
-----
RT_GETFILENAME:
  WDP      = AUTOFIN()
  APRIFIN_TIT WDP,1,0,8,20,5,58,TITLE01$,""
  SETPRFIN WDP, 2: SETTF "505Esci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + STRING$( 6, STR$(NRITEMS,0,0) )

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEY$, 0
    GOTO WAIT_LOOP
RETURN
-----

```

Step 02.

```

' *          *****          *
' * Nome File .....: spx02.ss          *
' * Descrizione ....: Gestione richiesta dati da video          *
' *          *****          *
' * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica.          *
' *          Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0.          *
' *          La quantita' la vogliamo tenere a 6 cifre.          *
' * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array.          *
' *          Aggiorniamo la nostra maschera video con il nr reale di articoli.          *
' *          *****          *

RT_MAIN:
  GOSUB  RT_INIT
  GOSUB  RT_CONTA_ANAG_ARTICOLI
  GOSUB  RT_GETFILENAME
END
-----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  KEY$     = ""
  NRITEMS  = 0
  IDX      = 0
  DIM CODART$ ( )
RETURN
-----
RT_GETFILENAME:
  WDP      = AUTOFIN()
  APRIFIN_TIT WDP,1,0,8,20,5,58,TITLE01$,""
  SETPRFIN WDP, 2: SETTF "505Esci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + STR$(NRITEMS,0,0)

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEY$, 0
    GOTO WAIT_LOOP
RETURN
-----
RT_CONTA_ANAG_ARTICOLI:
  AZZARRAY CODART$

  IDX = 1
+-----+
|Cod.tab.:ar  MAGAZZINO - ANAGRAFICA ARTICOLO
+-----+
ar          CODART$(IDX) = arcod()
ar          IDX = IDX + 1
ar          NRITEMS = IDX - 1
RETURN
-----

```

Step 03.

```

#APPENDISPX <lb_utils>
' *          *****          *
' * Nome File .....: spx03.ss          *
' * Descrizione ....: Gestione richiesta dati da video          *
' *          *****          *
' * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica.          *
' *          Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0.          *
' *          La quantita' la vogliamo tenere a 6 cifre.          *
' * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array.          *
' *          Aggiorniamo la nostra maschera video con il nr reale di articoli.          *
' * Step 03.: Aggiungiamo una piccola complicanza. Il numero che mostriamo a video, ora non e' piu'          *
' *          di 6 digit e vogliamo ripristinarlo.          *
' *          *****          *

RT_MAIN:
  GOSUB  RT_INIT
  GOSUB  RT_CONTA_ANAG_ARTICOLI
  GOSUB  RT_GETFILENAME
END
-----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  KEY$     = ""
  NRITEMS  = 0
  NRITEMS$ = "000000"
  IDX      = 0
  DIM CODART$ ( )
  GOSUB  PRCUT_INITIALIZE

```

```

RETURN
'
-----
RT_GETFILENAME:
  WDP      = AUTOFIN()
  APRIFIN TIT WDP,1,0,8,20,5,58,TITLE01$,""
  SETPRFIN WDP, 2: SETTF "505Esci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + NRITEM$

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEY$, 0
    GOTO WAIT_LOOP
RETURN
'
-----
RT_CONTA_ANAG_ARTICOLI:
  AZZARRAY CODART$

  IDX = 1
+-----+
|Cod.tab.:ar  MAGAZZINO - ANAGRAFICA ARTICOLO
+-----+
ar          CODART$(IDX) = arcod()
ar          IDX = IDX + 1
           NRITEMS = IDX - 1

           MUTI_NUM = NRITEMS
           MUTI_LEN = 6
           GOSUB  PRCUT_NUM2STR
           IF  MUTO_ERR$ <> "" THEN VIMSG 1, MUTO_ERR$
           NRITEM$ = MUTO_NUM$
RETURN
'
-----

```

Step 04.

```

#APPENDISPX <lb_utils>
' *****
' * Nome File .....: spx04.ss
' * Descrizione ....: Gestione richiesta dati da video
' *
' * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica.
' *          Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0.
' *          La quantita' la vogliamo tenere a 6 cifre.
' * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array.
' *          Aggiorniamo la nostra maschera video con il nr reale di articoli.
' * Step 03.: Aggiungiamo una piccola complicanza. Il numero che mostriamo a video, ora non e' piu'
' *          di 6 digit e vogliamo ripristinarlo.
' * Step 04.: Prepariamo un layout tabellare per mostrare i dati letti.
' *****

RT_MAIN:
  GOSUB  RT_INIT
  GOSUB  RT_CONTA_ANAG_ARTICOLI
  GOSUB  RT_GETFILENAME
END
'
-----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  TITLE02$ = "Contenuto Anagrafica Articoli"
  KEY$     = ""
  NRITEMS  = 0
  NRITEM$  = "000000"
  IDX      = 0
  DIM CODART$ ( )
  GOSUB  PRCUT_INITIALIZE
RETURN
'
-----
RT_GETFILENAME:
  WDP      = AUTOFIN()
  APRIFIN TIT WDP,1,0,8,20,5,58,TITLE01$,""
  SETPRFIN WDP, 2: SETTF "501Elabora,505Esci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + NRITEM$

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEY$, 0
    IF _TF = 501 THEN CHIUDIFIN WDP : GOSUB  RT_OUTPUT_MASK : RETURN
    IF _TF = 505 THEN CHIUDIFIN WDP : RETURN
    GOTO WAIT_LOOP
RETURN
'
-----
RT_CONTA_ANAG_ARTICOLI:
  AZZARRAY CODART$

  IDX = 1
+-----+
|Cod.tab.:ar  MAGAZZINO - ANAGRAFICA ARTICOLO
+-----+
ar          CODART$(IDX) = arcod()
ar          IDX = IDX + 1
           NRITEMS = IDX - 1

           MUTI_NUM = NRITEMS
           MUTI_LEN = 6
           GOSUB  PRCUT_NUM2STR
           IF  MUTO_ERR$ <> "" THEN VIMSG 1, MUTO_ERR$
           NRITEM$ = MUTO_NUM$
RETURN
'
-----
RT_OUTPUT_MASK:
  WDP      = AUTOFIN()
  APRIFIN TIT WDP,1,0,2,20,18,80,TITLE02$,""
  SETPRFIN WDP, 0: SETTF "505Esci"
  LOCATE 1, 2 : PRINT "===== "
  LOCATE 2, 2 : PRINT " CODICE           | DESCRIZIONE "
  LOCATE 2,62 : PRINT "| COSTO ULT."
  LOCATE 3, 2 : PRINT "===== "
  LOCATE 4,25 : PRINT "***** FINE ELENCO *****"
  LOCATE 5, 2 : PRINT "===== "
  WAIT_LOOP_OUT:
    LOCATE 20, 80 : INPUT "A", KEY$, 0

```

```
GOTO WAIT_LOOP_OUT
RETURN
```

Step 05.

```
#APPENDISPX <lb_utils>
' *****
' * Nome File .....: spx05.ss
' * Descrizione ....: Gestione richiesta dati da video
' *
' * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica.
' * Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0.
' * La quantita' la vogliamo tenere a 6 cifre.
' * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array.
' * Aggiorniamo la nostra maschera video con il nr reale di articoli.
' * Step 03.: Aggiungiamo una piccola complicita. Il numero che mostriamo a video, ora non e' piu'
' * di 6 digit e vogliamo ripristinarlo.
' * Step 04.: Prepariamo un layout tabellare per mostrare i dati letti.
' * Step 05.: Creazione di un file sequenziale vuoto.
' *****

RT_MAIN:
  GOSUB RT_INIT
  GOSUB RT_CONTA_ANAG_ARTICOLI
  GOSUB RT_GETFILENAME
END
' -----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  TITLE02$ = "Contenuto Anagrafica Articoli"
  KEYS$ = ""
  NRITEMS = 0
  NRITEMS$ = "000000"
  IDX = 0
  DIM CODART$( )
  GOSUB PRCUT_INITIALIZE
RETURN
' -----
RT_GETFILENAME:
  WDP = AUTOFIN()
  APRIFIN TIT WDP,1,0,8,20,5,58,TITLE01$,""
  SETPRFIN WDP, 2: SETTF "50IElabora,50SEsci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + NRITEMS$

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEYS$, 0
    IF _TF = 501 THEN CHIUDIFIN WDP : GOSUB RT_ELABORA : RETURN
    IF _TF = 505 THEN CHIUDIFIN WDP : RETURN
    GOTO WAIT_LOOP
RETURN
' -----
RT_CONTA_ANAG_ARTICOLI:
  AZZARRAY CODART$

  IDX = 1
+-----+
|Cod.tab.:ar MAGAZZINO - ANAGRAFICA ARTICOLO
+-----+
ar CODART$(IDX) = arcod()
ar IDX = IDX + 1
ar NRITEMS = IDX - 1

  MUTI_NUM = NRITEMS
  MUTI_LEN = 6
  GOSUB PRCUT_NUM2STR
  IF MUTO_ERR$ <> "" THEN VIMSG 1, MUTO_ERR$
  NRITEMS$ = MUTO_NUMS$
RETURN
' -----
RT_ELABORA:
  GOSUB RT_CREA_FILE_SEQ_VUOTO
  GOSUB RT_CHIUDI_FILE
  GOSUB RT_OUTPUT_MASK
RETURN
' -----
RT_OUTPUT_MASK:
  WDP = AUTOFIN()
  APRIFIN TIT WDP,1,0,2,20,18,80,TITLE02$,""
  SETPRFIN WDP, 0: SETTF "50SEsci"
  LOCATE 1, 2 : PRINT "=====|"
  LOCATE 2, 2 : PRINT " CODICE | DESCRIZIONE"
  LOCATE 2,62 : PRINT "| COSTO ULT."
  LOCATE 3, 2 : PRINT "=====|"
  LOCATE 4,25 : PRINT "***** FINE ELENCO *****"
  LOCATE 5, 2 : PRINT "=====|"
  WAIT_LOOP_OUT:
    LOCATE 20, 80 : INPUT "A", KEYS$, 0
    GOTO WAIT_LOOP_OUT
RETURN
' -----
RT_CREA_FILE_SEQ_VUOTO:
  NF1 = AUTOFIN()
  OPEN "R", NF1, "\\file_stp_sprix.txt", 64 '64 lunghezza record
  IFF_ERRFILE$ <> ""
    VIMSG 1, "Si e' verificato un errore nel tentativo di creare il file. " + _ERRFILE$
  ELSEF
    VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato creato con successo."
  ENDF
RETURN
' -----
RT_CHIUDI_FILE:
  CLOSE NF1
RETURN
' -----
```

Step 06.

```
#APPENDISPX <lb_utils>
' *****
' * Nome File .....: spx06.ss
' * Descrizione ....: Gestione richiesta dati da video
' *
' * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica.
' *          Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0.
' *          La quantita' la vogliamo tenere a 6 cifre.
' * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array.
' *          Aggiorniamo la nostra maschera video con il nr reale di articoli.
' * Step 03.: Aggiungiamo una piccola complicanza. Il numero che mostriamo a video, ora non e' piu'
' *          di 6 digit e vogliamo ripristinarlo.
' * Step 04.: Prepariamo un layout tabellare per mostrare i dati letti.
' * Step 05.: Creazione di un file sequenziale vuoto.
' * Step 06.: Popolo il file sequenziale con struttura a 3 campi (codice+descr+costo ult)
' *****

RT_MAIN:
  GOSUB RT_INIT
  GOSUB RT_CONTA_ANAG_ARTICOLI
  GOSUB RT_GETFILENAME
END
' -----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  TITLE02$ = "Contenuto Anagrafica Articoli"
  KEYS$ = ""
  NRITEMS = 0
  NRITEM$ = "000000"
  IDX = 0
  DIM CODART$ ( )
  GOSUB PRCUT_INITIALIZE
RETURN
' -----
RT_GETFILENAME:
  WDP = AUTOFIN()
  APRIFIN TIT WDP,1,0,8,20,5,58,TITLE01$,"
  SETPRFIN WDP, 2: SETTF "501Elabora,505Esci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + NRITEM$

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEYS, 0
    IF _TF = 501 THEN CHIUDIFIN WDP : GOSUB RT_ELABORA : RETURN
    IF _TF = 505 THEN CHIUDIFIN WDP : RETURN
    GOTO WAIT_LOOP
RETURN
' -----
RT_CONTA_ANAG_ARTICOLI:
  AZZARRAY CODART$

  IDX = 1
+-----+
|Cod.tab.:ar MAGAZZINO - ANAGRAFICA ARTICOLA
+-----+
ar CODART$(IDX) = arcod()
ar IDX = IDX + 1
ar NRITEMS = IDX - 1

  MUTI_NUM = NRITEMS
  MUTI_LEN = 6
  GOSUB PRCUT_NUM2STR
  IF MUTO_ERR$ <> "" THEN VIMSG 1, MUTO_ERR$
  NRITEM$ = MUTO_NUM$
RETURN
' -----
RT_ELABORA:
  GOSUB RT_CREA_FILE_SEQ_VUOTO
  GOSUB RT_POPOLA_FILE_SEQ
  GOSUB RT_CHIUDI_FILE
  GOSUB RT_OUTPUT_MASK
RETURN
' -----
RT_OUTPUT_MASK:
  WDP = AUTOFIN()
  APRIFIN TIT WDP,1,0,2,20,18,80,TITLE02$,"
  SETPRFIN WDP, 0: SETTF "505Esci"
  LOCATE 1, 2 : PRINT "=====
  LOCATE 2, 2 : PRINT " CODICE | DESCRIZIONE"
  LOCATE 2,62 : PRINT "| COSTO ULT."
  LOCATE 3, 2 : PRINT "=====
  LOCATE 4,25 : PRINT "***** FINE ELENCO *****"
  LOCATE 5, 2 : PRINT "=====
  WAIT_LOOP_OUT:
    LOCATE 20, 80 : INPUT "A", KEYS, 0
    GOTO WAIT_LOOP_OUT
RETURN
' -----
RT_CREA_FILE_SEQ_VUOTO:
  NF1 = AUTOFIN()
  OPEN "R", NF1, "]"file_stp_sprix.txt", 64 '64 lunghezza record
  IFF_ERRFILES <> ""
  VIMSG 1, "Si e' verificato un errore nel tentativo di creare il file. " + _ERRFILES$
  ELSEIF
  VIMSG 1, "Il file 'file_stp_sprix.txt' e' stato creato con successo."
  ENDF
RETURN
' -----
RT_CHIUDI_FILE:
  CLOSE NF1
RETURN
' -----
RT_POPOLA_FILE_SEQ:
  CRLF$ = CHR$(13) + CHR$(10)
  FIELD NF1, 10 AS F_CODART$, 39 AS F_DESART$, 13 AS F_COSULT$, 2 AS CRLF$
  FOR IDX = 1 TO NRITEMS
    F_CODART$ = CODART$(IDX)
    F_DESART$ = ardes(CODART$(IDX))
    F_COSULT$ = STR$(arcul(CODART$(IDX)),2,0)
    PUT NF1, IDX
  
```

```

NEXT
VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato popolato con successo."
RETURN

```

Step 07.

```

#APPENDISPX <lb_utils>
! *****
! * Nome File .....: spx07.ss
! * Descrizione ....: Gestione richiesta dati da video
! *
! * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica.
! *          Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0.
! *          La quantita' la vogliamo tenere a 6 cifre.
! * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array.
! *          Aggiorniamo la nostra maschera video con il nr reale di articoli.
! * Step 03.: Aggiungiamo una piccola complicita. Il numero che mostriamo a video, ora non e' piu'
! *          di 6 digit e vogliamo ripristinarlo.
! * Step 04.: Prepariamo un layout tabellare per mostrare i dati letti.
! * Step 05.: Creazione di un file sequenziale vuoto.
! * Step 06.: Popolo il file sequenziale con struttura a 3 campi (codice+descr+costo ult)
! * Step 07.: Leggo il contenuto del file sequenziale precedentemente scritto.
! *****

RT_MAIN:
  GOSUB RT_INIT
  GOSUB RT_CONTA_ANAG_ARTICOLI
  GOSUB RT_GETFILENAME
END
-----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  TITLE02$ = "Contenuto Anagrafica Articoli"
  KEYS$ = ""
  NRITEMS = 0
  NRITEMS$ = "000000"
  IDX = 0
  DIM CODART$ ( )
  DIM F_RECS (,3)

  GOSUB PRCUT_INITIALIZE
RETURN
-----
RT_GETFILENAME:
  WDP = AUTOFIN()
  APRIFIN TIT WDP,1,0,8,20,5,58,TITLE01$,""
  SETPRFIN WDP, 2: SETTF "501Elabora,505Esci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + NRITEMS$

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEYS$, 0
    IF _TF = 501 THEN CHIUDIFIN WDP : GOSUB RT_ELABORA : RETURN
    IF _TF = 505 THEN CHIUDIFIN WDP : RETURN
    GOTO WAIT_LOOP

RETURN
-----
RT_CONTA_ANAG_ARTICOLI:
  AZZARRAY CODART$

  IDX = 1
-----
|Cod.tab.:ar MAGAZZINO - ANAGRAFICA ARTICOLO
-----
ar CODART$(IDX) = arcod()
ar IDX = IDX + 1
  NRITEMS = IDX - 1

  MUTI_NUM = NRITEMS
  MUTI_LEN = 6
  GOSUB PRCUT_NUM2STR
  IF MUTO_ERR$ <> "" THEN VIMSG 1, MUTO_ERR$
  NRITEMS$ = MUTO_NUM$
RETURN
-----
RT_ELABORA:
  GOSUB RT_CREA_FILE_SEQ_VUOTO
  GOSUB RT_POPOLA_FILE_SEQ
  GOSUB RT_LEGGI_FILE_SEQ
  GOSUB RT_CHIUDI_FILE
  GOSUB RT_OUTPUT_MASK
RETURN
-----
RT_OUTPUT_MASK:
  WDP = AUTOFIN()
  APRIFIN TIT WDP,1,0,2,20,18,80,TITLE02$,""
  SETPRFIN WDP, 0: SETTF "505Esci"
  LOCATE 1, 2 : PRINT "=====|"
  LOCATE 2, 2 : PRINT " CODICE | DESCRIZIONE"
  LOCATE 2,62 : PRINT "| COSTO ULT."
  LOCATE 3, 2 : PRINT "=====|"
  LOCATE 4,25 : PRINT "***** FINE ELENCO *****"
  LOCATE 5, 2 : PRINT "=====|"
  WAIT_LOOP_OUT:
    LOCATE 20, 80 : INPUT "A", KEYS$, 0
    GOTO WAIT_LOOP_OUT
RETURN
-----
RT_CREA_FILE_SEQ_VUOTO:
  NF1 = AUTOFILE()
  OPEN "R", NF1, "\file_stp_sprix.txt", 64 '64 lunghezza record
  IFF_ERRFILE$ <> ""
    VIMSG 1, "Si e' verificato un errore nel tentativo di creare il file. " + _ERRFILE$
  ELSEF
    VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato creato con successo."
  ENDF
RETURN
-----

```

```

RT_CHIUDI_FILE:
  CLOSE NF1
RETURN
-----
RT_POPOLA_FILE_SEQ:
  CRLF$ = CHR$(13) + CHR$(10)
  FIELD NF1, 10 AS F_CODART$, 39 AS F_DESART$, 13 AS F_COSULT$, 2 AS CRLF$
  FOR IDX = 1 TO NRITEMS
    F_CODART$ = CODART$(IDX)
    F_DESART$ = ardes(CODART$(IDX))
    F_COSULT$ = STR$(arcul(CODART$(IDX)),2,0)
    PÙT NF1, IDX
  NEXT
  VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato popolato con successo."
RETURN
-----
RT_LEGGI_FILE_SEQ:
  AZZARRAY F_REC$
  CRLF$ = CHR$(13) + CHR$(10)

  NREC = LOF (NF1) / 64
  FOR IDX = 1 TO NRITEMS
    GET NF1, IDX
    F_REC$(IDX,1) = F_CODART$
    F_REC$(IDX,2) = F_DESART$
    F_REC$(IDX,3) = F_COSULT$
  NEXT
  VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato letto con successo."
RETURN
-----

```

Step 08.

```

#APPENDISPX <lb_utils>
! *****
! * Nome File .....: spx08.ss *
! * Descrizione ....: Gestione richiesta dati da video *
! * * * * *
! * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica. *
! * Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0. *
! * La quantita' la vogliamo tenere a 6 cifre. *
! * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array. *
! * Aggiorniamo la nostra maschera video con il nr reale di articoli. *
! * Step 03.: Aggiungiamo una piccola complicità. Il numero che mostriamo a video, ora non e' piu' *
! * di 6 digit e vogliamo ripristinarlo. *
! * Step 04.: Prepariamo un layout tabellare per mostrare i dati letti. *
! * Step 05.: Creazione di un file sequenziale vuoto. *
! * Step 06.: Popolo il file sequenziale con struttura a 3 campi (codice+descr+costo ult) *
! * Step 07.: Leggo il contenuto del file sequenziale precedentemente scritto. *
! * Step 08.: Metto sul video di output gli ultimi 10 record del file. *
! *****

RT_MAIN:
  GOSUB RT_INIT
  GOSUB RT_CONTA_ANAG_ARTICOLI
  GOSUB RT_GETFILENAME
END
-----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  TITLE02$ = "Contenuto Anagrafica Articoli"
  KEYS$ = ""
  NRITEMS = 0
  NRITEM$ = "000000"
  IDX = 0
  ROW = 0
  DIM CODART$ ( )
  DIM F_REC$ ( ,3)

  GOSUB PRCUT_INITIALIZE
RETURN
-----
RT_GETFILENAME:
  WDP = AUTOFIN()
  APRIFIN_TIT WDP,1,0,8,20,5,58,TITLE01$,""
  SETPRFIN WDP, 2: SETTF "501Elabora,505Esci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + NRITEM$

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEYS, 0
    IF _TF = 501 THEN CHIUDIFIN WDP : GOSUB RT_ELABORA : RETURN
    IF _TF = 505 THEN CHIUDIFIN WDP : RETURN
  GOTO WAIT_LOOP
RETURN
-----
RT_CONTA_ANAG_ARTICOLI:
  AZZARRAY CODART$

  IDX = 1
+-----+
|Cod.tab.:ar MAGAZZINO - ANAGRAFICA ARTICOLO
+-----+
ar CODART$(IDX) = arcod()
ar IDX = IDX + 1
NRITEMS = IDX - 1

  MUTI_NUM = NRITEMS
  MUTI_LEN = 6
  GOSUB PRCUT_NUM2STR
  IF MUTO_ERR$ <> "" THEN VIMSG 1, MUTO_ERR$
  NRITEM$ = MUTO_NUM$
RETURN
-----
RT_ELABORA:
  GOSUB RT_CREA_FILE_SEQ_VUOTO
  GOSUB RT_POPOLA_FILE_SEQ
  GOSUB RT_LEGGI_FILE_SEQ
  GOSUB RT_CHIUDI_FILE

```

```

GOSUB RT_OUTPUT_MASK
RETURN
' -----
RT_OUTPUT_MASK:
WDP = AUTOFIN()
APRIFIN TIT WDP,1,0,2,20,18,80,TITLE02$,"
SETPRFIN WDP, 0: SETTF "505Esci"
LOCATE 1, 2 : PRINT "-----"
LOCATE 2, 2 : PRINT " CODICE | DESCRIZIONE"
LOCATE 2,62 : PRINT "| COSTO ULT."
LOCATE 3, 2 : PRINT "-----"
LOCATE 4,25 : PRINT "***** FINE ELENCO *****"
ROW = 4
FOR IDX = NRITEMS TO NRITEMS - 9 STEP -1
LOCATE ROW, 3 : PRINT F_REC$(IDX, 1)
LOCATE ROW,24 : PRINT F_REC$(IDX, 2)
LOCATE ROW,64 : PRINT F_REC$(IDX, 3)
ROW = ROW + 1
NEXT
LOCATE ROW, 2 : PRINT "-----"
WAIT_LOOP_OUT:
LOCATE 20, 80 : INPUT "A", KEYS, 0
GOTO WAIT_LOOP_OUT
RETURN
' -----
RT_CREA_FILE_SEQ_VUOTO:
NF1 = AUTOFIN()
OPEN "R", NF1, "\file_stp_sprix.txt", 64 '64 lunghezza record
IF _ERRFILE$ <> ""
VIMSG 1, "Si e' verificato un errore nel tentativo di creare il file." + _ERRFILE$
ELSEF
VIMSG 1, "Il file 'file_stp_sprix.txt' e' stato creato con successo."
ENDIF
RETURN
' -----
RT_CHIUDI_FILE:
CLOSE NF1
RETURN
' -----
RT_POPOLA_FILE_SEQ:
CRLF$ = CHR$(13) + CHR$(10)
FIELD NF1, 10 AS F_CODART$, 39 AS F_DESART$, 13 AS F_COSULT$, 2 AS CRLF$
FOR IDX = 1 TO NRITEMS
F_CODART$ = CODART$(IDX)
F_DESART$ = ardes(CODART$(IDX))
F_COSULT$ = STR$(arcul(CODART$(IDX)),2,0)
PUT NF1, IDX
NEXT
VIMSG 1, "Il file 'file_stp_sprix.txt' e' stato popolato con successo."
RETURN
' -----
RT_LEGGI_FILE_SEQ:
AZZARRAY F_REC$
CRLF$ = CHR$(13) + CHR$(10)

NREC = LOF (NF1) / 64
FOR IDX = 1 TO NRITEMS
GET NF1, IDX
F_REC$(IDX,1) = F_CODART$
F_REC$(IDX,2) = F_DESART$
F_REC$(IDX,3) = F_COSULT$
NEXT
VIMSG 1, "Il file 'file_stp_sprix.txt' e' stato letto con successo."
RETURN
' -----

```

Step 09.

```

#APPENDISPX <lb utils>
' *****
' * Nome File .....: spx09.ss *
' * Descrizione ....: Gestione richiesta dati da video *
' * * *
' * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica. *
' * Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0. *
' * La quantita' la vogliamo tenere a 6 cifre. *
' * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array. *
' * Aggiorniamo la nostra maschera video con il nr reale di articoli. *
' * Step 03.: Aggiungiamo una piccola complicita. Il numero che mostriamo a video, ora non e' piu' *
' * di 6 digit e vogliamo ripristinarlo. *
' * Step 04.: Prepariamo un layout tabellare per mostrare i dati letti. *
' * Step 05.: Creazione di un file sequenziale vuoto. *
' * Step 06.: Popolo il file sequenziale con struttura a 3 campi (codice+descr+costo ult) *
' * Step 07.: Leggo il contenuto del file sequenziale precedentemente scritto. *
' * Step 08.: Metto sul video di output gli ultimi 10 record del file. *
' * Step 09.: Riordino i record da visualizzare e pongo attenzione sul costruito IFF *
' *****

RT_MAIN:
GOSUB RT_INIT
GOSUB RT_CONTA_ANAG_ARTICOLI
GOSUB RT_GETFILENAME
END
' -----
RT_INIT:
TITLE01$ = "Lettura anagrafica articoli"
TITLE02$ = "Contenuto Anagrafica Articoli"
KEY$ = ""
NRITEMS = 0
NRITEM$ = "000000"
IDX = 0
ROW = 0
ERR = 0
SORTITEM$ = ""
DIM CODART$ ()
DIM F_REC$ (,3)
DIM R_SORT$ (,3)

```

```

GOSUB PRCUT_INITIALIZE
RETURN
'-----
RT_GETFILENAME:
WDP = AUTOFIN()
'APRIFIN TIT WDP,1,0,8,20,5,58,TITLE01$,"A501Elabora"
APRIFIN TIT WDP,1,0,8,20,5,58,TITLE01$,""
SETPRFIN WDP, 2: SETTF "501Elabora,505Esci"
LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + NRITEM$

WAIT_LOOP:
LOCATE 6, 59 : INPUT "A", KEYS$, 0
LOCATE 3, 19 : INPUT EXT 2, "F", FNAME$, 40, 128, ""
IF _TF = 501 THEN CHIUDIFIN WDP : GOSUB RT_ELABORA : RETURN
IF _TF = 505 THEN CHIUDIFIN WDP : RETURN
GOTO WAIT_LOOP

RETURN
'-----
RT_CONTA_ANAG_ARTICOLI:
_AZZARRAY CODART$

IDX = 1
+-----
|Cod.tab.:ar MAGAZZINO - ANAGRAFICA ARTICOLO
+-----
ar CODART$(IDX) = arcod()
ar IDX = IDX + 1
NRITEMS = IDX - 1

MUTI_NUM = NRITEMS
MUTI_LEN = 6
GOSUB PRCUT_NUM2STR
IF MUTO_ERR$ <> "" THEN VIMSG 1, MUTO_ERR$
NRITEM$ = MUTO_NUM$
RETURN
'-----
RT_ELABORA:
GOSUB RT_CREA_FILE_SEQ_VUOTO
GOSUB RT_POPOLA_FILE_SEQ
GOSUB RT_LEGGI_FILE_SEQ
GOSUB RT_CHIUDI_FILE
GOSUB RT_ORDINA
GOSUB RT_OUTPUT_MASK
RETURN
'-----
RT_OUTPUT_MASK:
WDP = AUTOFIN()
APRIFIN TIT WDP,1,0,2,20,18,80,TITLE02$,""
SETPRFIN WDP, 0: SETTF "505Esci"
LOCATE 1, 2 : PRINT "=====
LOCATE 2, 2 : PRINT " CODICE | DESCRIZIONE"
LOCATE 2,62 : PRINT "| COSTO ULT."
LOCATE 3, 2 : PRINT "=====
LOCATE 4,25 : PRINT "***** FINE ELENCO *****"
ROW = 4
'FOR IDX = NRITEMS TO NRITEMS - 9 STEP -1
' LOCATE ROW, 3 : PRINT F_REC$(IDX, 1)
' LOCATE ROW,24 : PRINT F_REC$(IDX, 2)
' LOCATE ROW,64 : PRINT F_REC$(IDX, 3)
' ROW = ROW + 1
'NEXT
FOR IDX = 1 TO 10
LOCATE ROW, 3 : PRINT R_SORT$(IDX, 1)
LOCATE ROW,24 : PRINT R_SORT$(IDX, 2)
LOCATE ROW,64 : PRINT R_SORT$(IDX, 3)
ROW = ROW + 1
NEXT

LOCATE ROW, 2 : PRINT "=====
WAIT_LOOP_OUT:
LOCATE 20, 80 : INPUT "A", KEYS$, 0
GOTO WAIT_LOOP_OUT
RETURN
'-----
RT_CREA_FILE_SEQ_VUOTO:
NF1 = AUTOFIN()
OPEN "R", NF1, "\file_stp_sprix.txt", 64 '64 lunghezza record
IFF_ERRFILE$ <> ""
VIMSG 1, "Si e' verificato un errore nel tentativo di creare il file. " + _ERRFILE$
ELSEF
VIMSG 1, "Il file 'file_stp_sprix.txt' e' stato creato con successo."
ENDIF
RETURN
'-----
RT_CHIUDI_FILE:
CLOSE NF1
RETURN
'-----
RT_POPOLA_FILE_SEQ:
CRLF$ = CHR$(13) + CHR$(10)
FIELD NF1, 10 AS F_CODART$, 39 AS F_DESART$, 13 AS F_COSULT$, 2 AS CRLF$
FOR IDX = 1 TO NRITEMS
' F_ROW$ = STRING$(IDX, MKA$(F_COSULT$, 62))
' F_ROW$ = MKA$(F_ROW$, 62)
' VIMSG 1, F_ROW$
F_CODART$ = CODART$(IDX)
F_DESART$ = ardes(CODART$(IDX))
F_COSULT$ = STR$(arcul(CODART$(IDX)),2,0)
PUT NF1, IDX
NEXT
VIMSG 1, "Il file 'file_stp_sprix.txt' e' stato popolato con successo."
RETURN
'-----
RT_LEGGI_FILE_SEQ:
_AZZARRAY F_REC$
CRLF$ = CHR$(13) + CHR$(10)
FIELD NF1, 10 AS F_CODART$, 39 AS F_DESART$, 13 AS F_COSULT$, 2 AS CRLF$

```



```

NREC = LOF (NFI) / 64
FOR IDX = 1 TO NRITEMS
  GET NFI, IDX
  F_RECS (IDX,1) = F_CODART$
  F_RECS (IDX,2) = F_DESART$
  F_RECS (IDX,3) = F_COSULT$
NEXT
VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato letto con successo."
RETURN
'-----
RT_ORDINA:
  _ERR = 0
  _LRECSORT = 6      'Lunghezza in caratteri del dato sul quale effettuo l'ordinamento

  FOR IDX = NRITEMS TO NRITEMS - 9 STEP -1
    MUTI_NUM = IDX
    MUTI_LEN = 6
    GOSUB  PRCUT_NUM2STR
    IFF MUTO_ERR$ <> ""
      VIMSG 1, MUTO_ERR$ + "|Impossibile completare l'ordinamento"
      ERR = 1
    ELSEF
      NRITEM$ = MUTO_NUM$
      PUTSORT MUTO_NUM$ + F_RECS (IDX, 2)
    ENDF
    IF ERR = 1 THEN IDX = NRITEMS - 9
  NEXT

  AZZARRAY R_SORT$
  FOR  IDX = 1 TO 10
    GETSORT  SORTITEM$
    R_SORT$ (IDX, 1) = F_RECS ( VAL( LEFT$(SORTITEM$,6) ),1 )
    R_SORT$ (IDX, 2) = F_RECS ( VAL( LEFT$(SORTITEM$,6) ),2 )
    R_SORT$ (IDX, 3) = F_RECS ( VAL( LEFT$(SORTITEM$,6) ),3 )
  NEXT
RETURN
'-----

```

Step 10.

```

#APPENDISPX <lb_utils>
' *****
' * Nome File .....: spx10.ss
' * Descrizione ....: Gestione richiesta dati da video
' *
' * Step 01.: Creazione maschera video che ci mostra il nr di articoli presenti in anagrafica.
' *          Ci occupiamo di realizzare il solo video e di mostrare come quantita' lo 0.
' *          La quantita' la vogliamo tenere a 6 cifre.
' * Step 02.: Contiamo tutti i codici articoli presenti in anagrafica e li memorizziamo in un array.
' *          Aggiorniamo la nostra maschera video con il nr reale di articoli.
' * Step 03.: Aggiungiamo una piccola complicanza. Il numero che mostriamo a video, ora non e' piu'
' *          di 6 digit e vogliamo ripristinarlo.
' * Step 04.: Prepariamo un layout tabellare per mostrare i dati letti.
' * Step 05.: Creazione di un file sequenziale vuoto.
' * Step 06.: Popolo il file sequenziale con struttura a 3 campi (codice+descr+costo ult)
' * Step 07.: Leggo il contenuto del file sequenziale precedentemente scritto.
' * Step 08.: Metto sul video di output gli ultimi 10 record del file.
' * Step 09.: Riordino i record da visualizzare e pongo attenzione sul costruito IFF
' *****

RT_MAIN:
  GOSUB  RT_INIT
  GOSUB  RT_CONTA_ANAG_ARTICOLI
  GOSUB  RT_GETFILENAME
END
'-----
RT_INIT:
  TITLE01$ = "Lettura anagrafica articoli"
  TITLE02$ = "Contenuto Anagrafica Articoli"
  KEY$ = ""
  NRITEMS = 0
  NRITEM$ = "000000"
  IDX = 0
  ROW = 0
  ERR = 0
  SORTITEM$ = ""
  DIM CODART$ ()
  DIM F_RECS (,3)
  DIM R_SORT$ (,3)

  GOSUB  PRCUT_INITIALIZE
RETURN
'-----
RT_GETFILENAME:
  WDP = AUTOFIN()
  'APRIFIN_TIT WDP,1,0,8,20,5,58,TITLE01$,"A501Elabora"
  APRIFIN_TIT WDP,1,0,8,20,5,58,TITLE01$,""
  SETPRFIN WDP, 2: SETTF "501Elabora,505Esci"
  LOCATE 3, 1 : PRINTD "Numero Articoli presenti : " + NRITEM$

  WAIT_LOOP:
    LOCATE 6, 59 : INPUT "A", KEY$, 0
    'LOCATE 3, 19 : INPUT_EXT 2, "F", FNAME$, 40, 128, ""
    IF _TF = 501 THEN CHIUDIFIN WDP : GOSUB RT_ELABORA : RETURN
    IF _TF = 505 THEN CHIUDIFIN WDP : RETURN
  GOT0 WAIT_LOOP
RETURN
'-----
RT_CONTA_ANAG_ARTICOLI:
  AZZARRAY CODART$

  IDX = 1
+-----+
|Cod.tab.:ar  MAGAZZINO - ANAGRAFICA ARTICOLO
+-----+
ar          CODART$(IDX) = arcod()
ar          IDX = IDX + 1
ar          NRITEMS = IDX - 1

```

```

MUTI_NUM = NRITEMS
MUTI_LEN = 6
GOSUB PRCUT_NUM2STR
IF MUTO_ERR$ <> "" THEN VIMSG 1, MUTO_ERR$
NRITEMS = MUTO_NUM$
RETURN
'-----
RT_ELABORA:
  GOSUB RT_CREA_FILE_SEQ_VUOTO
  GOSUB RT_POPOLA_FILE_SEQ
  GOSUB RT_LEGGI_FILE_SEQ
  GOSUB RT_CHIUDI_FILE
  GOSUB RT_ORDINA
  GOSUB RT_OUTPUT_MASK
RETURN
'-----
RT_OUTPUT_MASK:
  WDP = AUTOFIN()
  APRIFIN TIT WDP,1,0,2,20,18,80,TITLE02$, "605Stampa"
  SETPRFIN WDP, 0: SETTF "505Esci,605Stampa"
  LOCATE 1, 2 : PRINT "=====
  LOCATE 2, 2 : PRINT " CODICE | DESCRIZIONE"
  LOCATE 2,62 : PRINT "| COSTO ULT."
  LOCATE 3, 2 : PRINT "=====
  LOCATE 4,25 : PRINT "***** FINE ELENCO *****"
  ROW = 4
  FOR IDX = NRITEMS TO NRITEMS - 9 STEP -1
  ' LOCATE ROW, 3 : PRINT F_REC$(IDX, 1)
  ' LOCATE ROW,24 : PRINT F_REC$(IDX, 2)
  ' LOCATE ROW,64 : PRINT F_REC$(IDX, 3)
  ' ROW = ROW + 1
  'NEXT
  FOR IDX = 1 TO 10
  LOCATE ROW, 3 : PRINT R_SORT$(IDX, 1)
  LOCATE ROW,24 : PRINT R_SORT$(IDX, 2)
  LOCATE ROW,64 : PRINT R_SORT$(IDX, 3)
  ROW = ROW + 1
  NEXT
  LOCATE ROW, 2 : PRINT "=====
  WAIT_LOOP_OUT:
  LOCATE 20, 80 : INPUT "A", KEYS, 0
  IF TF = 605 THEN GOSUB RT_STAMPA_ARTICOLI : RETURN
  GOTO WAIT_LOOP_OUT
RETURN
'-----
RT_CREA_FILE_SEQ_VUOTO:
  NF1 = AUTOFIN()
  OPEN "R", NF1, "\file_stp_sprix.txt", 64 '64 lunghezza record
  IFF_ERRFILE$ <> ""
  VIMSG 1, "Si e' verificato un errore nel tentativo di creare il file. " + _ERRFILE$
  ELSEF
  VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato creato con successo."
  ENDF
RETURN
'-----
RT_CHIUDI_FILE:
  CLOSE NF1
RETURN
'-----
RT_POPOLA_FILE_SEQ:
  CRLF$ = CHR$(13) + CHR$(10)
  FIELD NF1, 10 AS F_CODART$, 39 AS F_DESART$, 13 AS F_COSULT$, 2 AS CRLF$
  FOR IDX = 1 TO NRITEMS
  F_CODART$ = CODART$(IDX)
  F_DESART$ = ardes(CODART$(IDX))
  F_COSULT$ = STR$(arcul(CODART$(IDX)),2,0)
  PUT NF1, IDX
  NEXT
  VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato popolato con successo."
RETURN
'-----
RT_LEGGI_FILE_SEQ:
  AZZARRAY F_REC$
  CRLF$ = CHR$(13) + CHR$(10)
  NREC = LOF(NF1) / 64
  FOR IDX = 1 TO NRITEMS
  GET NF1, IDX
  F_REC$(IDX,1) = F_CODART$
  F_REC$(IDX,2) = F_DESART$
  F_REC$(IDX,3) = F_COSULT$
  NEXT
  VIMSG 1, "Il File 'file_stp_sprix.txt' e' stato letto con successo."
RETURN
'-----
RT_ORDINA:
  ERR = 0
  _LRECSORT = 6 'Lunghezza in caratteri del dato sul quale effettuo l'ordinamento
  FOR IDX = NRITEMS TO NRITEMS - 9 STEP -1
  MUTI_NUM = IDX
  MUTI_LEN = 6
  GOSUB PRCUT_NUM2STR
  IFF MUTO_ERR$ <> ""
  VIMSG 1, MUTO_ERR$ + "|Impossibile completare l'ordinamento"
  ERR = 1
  ELSEF
  NRITEMS = MUTO_NUM$
  PUTSORT MUTO_NUM$ + F_REC$(IDX, 2)
  ENDF
  IF ERR = 1 THEN IDX = NRITEMS - 9
  NEXT
  AZZARRAY R_SORT$
  FOR IDX = 1 TO 10
  GETSORT SORTITEM$
  R_SORT$(IDX, 1) = F_REC$( VAL( LEFT$(SORTITEM$,6) ),1 )

```

```

R_SORT$(IDX, 2) = F_REC$( VAL( LEFT$(SORTITEM$,6) ),2 )
R_SORT$(IDX, 3) = F_REC$( VAL( LEFT$(SORTITEM$,6) ),3 )
NEXT
RETURN
'-----
RT_STAMPA ARTICOLI:
  APRISTAMP 127, ""
  Y = 32

LPRINTLINE 0, 10, 10, 180, 0, "C255000000"
LPRINTTEXT "ANAGRAFICA ARTICOLO", "Courier New", 12, "G", 0, 95, 11, "AC C000000255"
LPRINTLINE 0, 10, 17, 180, 0, "C255000000"

LPRINTTEXT "-----", "Courier New", 10, "N", 0, 10, 20, "C000000000"
LPRINTTEXT "CODICE | DESCRIZIONE | COSTO", "Courier New", 10, "N", 0, 10, 24, "C000000000"
LPRINTTEXT "-----", "Courier New", 10, "N", 0, 10, 28, "C000000000"
FOR IDX = 1 TO NRITEMS
  PRT_ROW$ = ""
  PRT_ROW$ = F_REC$(IDX,1) + "|" + F_REC$(IDX,2) + "|" + F_REC$(IDX,3)
  LPRINTTEXT PRT_ROW$, "Courier New", 10, "N", 0, 10, Y, "C000000000"
  Y = Y + 5
NEXT
LPRINTTEXT "-----", "Courier New", 10, "N", 0, 10, Y, "C000000000"

CHIUDISTAMP
RETURN

```

Regola generale nella gestione della SEMAFORICA:

Riportiamo una regola generale per la gestione della semaforica, in modo da evitare la sovrapposizione fra utenti nella gestione di un medesimo supporto dati (file) ed eventuale conseguente perdita di dati.

Quando il programma scrive e registra i dati, si utilizza un semaforo di tipo SCRIVO_SOLO_IO_E_BLOCCA_IL_*DISCO e si utilizza un semaforo CONTROLLO_SU_CHIAVE_EVENTUALE_DEL_RECORD.

In questo modo l'aggiornamento del file dati e soprattutto del file ad indice è effettuato da un utente alla volta, in modo da poter operare con attesa e senza interferenze.

Generalmente si tende a impostare il lavoro in questo modo:

```

SCRIVO_SOLO_IO_E_NON_ATTENDO:
BLOCCASEM " *DISCO "
RETURN

```

Ciò può comportare problemi accennati.

La forma migliore è invece:

```

SCRIVO_SOLO_IO_ED_ATTENDO:
BLOCCASEM " *DISCO "
IF_ERRSEM THEN GOTO SCRIVO_SOLO_IO_ED_ATTENDO
END

```

```

SBLOCCA:
SBLOCCASEM " *DISCO " (lo sbloccasem deve sentire l'evento oppure anche all'uscita del programma)
RETURN

```

```

SCRITTURA:
GOSUB SCRIVO_SOLO_IO_ED_ATTENDO:
PUT NFILE, NREC
oppure
INSKEY / VARKEY NFILE, NKEY
GOSUB SBLOCCA
e eventuale sblocco semaforo chiave (***)
RETURN

```

```

(***)
BLOCCA_MATRICOLA:
KMAT$ = STR$(MATRICOLA,0,0)
BLOCCASEM KMAT$

```

```

SBLOCCA_MATRICOLA:
IF KMAT$ = "" THEN RETURN (evita lo sblocco involontario di tutto, in quanto lo sblocco
di una stringa nulla comporta lo SBLOCCASEM di tutto
senza alcuna protezione degli accessi)
SBLOCCASEM KMAT$
RETURN

```

Esempio SEMAFORICA

```

Sprix1
BLOCCA:BLOCCASEM "*SEM1"
IF_ERRSEM THEN VIMSG 4, "Sem. bloccato da Sprix2/Collage": GOTO BLOCCA
PRINT "posso scrivere...":INPUT "A",A$,1:SBLOCCASEM "*SEM1": END

```

```

Sprix2
BLOCCO:BLOCCASEM "*SEM1"
IF_ERRSEM THEN VIMSG 4, "Sem. bloccato da Sprix1/Collage":GOTO BLOCCO
VIMSG 4, "via libera...": SBLOCCASEM "*SEM1":END

```

```

Collage
ON 612_ALL:
BLOC:BLOCCASEM "*SEM1"
IF_ERRSEM THEN VIMSG 4, "SEM. bloccato da Sprix1/Sprix2":GOTO BLOC
VIMSG 4, "posso procedere...": SBLOCCASEM "*SEM1":END

```

Esempio Collage sul controllo dei semafori (da compilare nel menu AUTO)

```

ON_612_ALL:
IF FL_SEMA=0 THEN BLOCCASEM "*SEMAH":FL_SEMA=1:VIMSG 4,"Bloccsem":GOTO LAB_VIS
IF FL_SEMA=1 THEN SBLOCCASEM "*SEMAH":FL_SEMA=0:VIMSG 5,"Sbloccsem":GOTO LAB_VIS
LAB VIS:

```

```
IF _ERRSEM THEN VIMSG 3,"ERRORE BLOCCO/SBLOCCO "+ STR$_( _ERRSEM,0,0)
END
```

Messaggio di errore: Input da tastiera con semafori in uso

E' un warning non bloccante e senza conseguenze. Comporta il blocco dei terminali fino a che l'utente Collage non conferma l'input da tastiera che lo stesso Collage potrebbe richiedere (ad esempio un messaggio di conferma con Invio) durante l'esecuzione in un modulo di un'operazione di scrittura. Se viene rilevato deve essere comunicato in modo dettagliato all'assistenza oppure modificare la sequenza di eventi.

Esempio LIVLICAPP:

```
A = LIVLICAPP()
IFF A <> 10 AND _MYDB$(1) <> "1000000"
  VIMSG 4,"Versione DEMO|Accettato solo codice INIZIALE = '1000000'"
  _TF=0
ELSEF
```

Messaggio: errore piugd() superato il limite

Significa che si sta elaborando la formazione di una data fino a superare i limiti di gestione (inferiore/superiore) consentiti per essa, esempio, verificare che non si stia oltrepassando l'anno 2048 (es. sommando tanti giorni ad una data).

Messaggi relativi a controlli sulle date e interruzioni di routine collage

L'esecuzione Sprix/Collage controlla la data di accesso e le date che vengono inserite/utilizzate nel gestionale. L'utilizzo di campi data (del gestionale o della personalizzazione) con valori incompatibili (scadenza) a una determinata data, potrebbero generare questi tipi di messaggi. Possono prevedersi soluzioni alternative per non incorrere nell'interruzione della routine tramite STOPCOLLAGE sulla routine in uscita dal campo, precedente a quello del messaggio; utilizzo di variabili globali, _GLOBAL e _GLOBAL\$ (poichè viste da tutti i Collage; il secondo collage verifica lo stop e, appurate le condizioni, lo fa ripartire da altro diverso punto.

Valorizzazione nulla dei campi data: "" oppure "00000000"

La valorizzazione nulla di un campo data puo' avvenire in due formati differenti: i campi di input data quando sono nulli ritornano "" (es: _SINP\$ sulla data di scadenza di un ordine cliente); mentre la variabile archiviata o letta successivamente come data nulla contiene "00000000" (es. _MMSCA\$(1))

Funzione ARRO: ome arrotondare un numero con decimali

Per effettuare l'arrotondamento di un numero con decimali tramite l'istruzione ARRO è possibile il seguente espediente N = numero da arrotondare (esempio 12,536) B = numero arrotondato per difetto (arrotondamento di tipo 1) al centesimo -> B = ARRO (N*100,1,1)/100 . In questo caso si è moltiplicato il valore per 100 in modo da portarlo ad intero, poi lo si è arrotondato, ed infine si è diviso il risultato dell'espressione per lo stesso valore (100) per il quale era stato moltiplicato all'inizio (B rilascia in questo caso 12,53 che rappresenta un arrotondamento per difetto).

Esempi di conversione separatore decimali da "," in "." :

```
N$="1,15": N1$="": FOR I=1 TO LEN(N$)
IF MID$(N$,I,1) = "," THEN N1$=N1$+"." ELSE N1$=N1$+MID$(N$,I,1): NEXT
N$="1,15": POS=INSTR(N$,"")
IF POS>0 THEN N1$=LEFT$(N$,POS-1) + "."+RIGHT$(N$,LEN(N$)-POS) ELSE N1$=N$
```

Caso: CVA\$ nel ordinamento del ciclotabellare

Non essendo prevista la CVA\$ nel ordinamento del ciclotabellare (ad esempio per migliorare l'effetto di rottura su padri strutturati che, invece, la consentita MID\$(arcod(),radice) non garantisce, in quanto la rottura sente il primo record successivo appartenente al primo figlio; questo, specialmente in presenza di blank successivi alla radice padre e come effetto è quello della stampa di due righe per ogni padre), cioè, è possibile che la CVA\$ non sia realizzabile in ciclo (poichè interviene nella costruzione della stringa in momento differente rispetto alla MID\$), si propone il possibile work around, ovvero, surrogare la rottura con condizione IF in base alla radice della struttura.

```
ar |IF LEN(arcod()) <= amdra THEN GOTO STAMPA ELSE GOTO SALTA
ar |STAMPA:
ar N|~arcod()
ar |SALTA:
```

AUDIO - VIDEO - TASTIERA - MOUSE

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AUDIO		
BEEP	Istruzione che emette un suono da sistema. Emette un suono acustico dall'altoparlante. E' utilizzabile per segnalare un errore o per richiamare l'attenzione dell'operatore per un evento particolare. es.	S/C/Cd
ON_BEEP:	Evento all'emissione di un suono	S/C/Cd
VIDEO		
_NCAM\$	Nome codice campo di input del Gestionale. es.	C/Cd
_DES612\$	Descrizione del tasto [F12] es.	C/Cd
_SINP\$	Valore del campo in immissione es.	C/Cd
GETPOSCURSX()	Par: (<par>). Funzione che legge la posizione di colonna del cursore. es.	S/C/Cd
GETPOSCURSY()	Par: (<par>). Funzione che legge posizione di riga del cursore. es.	S/C/Cd
SCREEN\$()	Par: (<nfin>, <rig>, <col>, <larg>). Funzione che legge i caratteri a video. es. <i>Nota bene: funzione vincolante, si raccomanda l'uso solo in caso di necessità, in quanto soggetta a variazioni in relazione a modifiche dell'interfaccia Gestionale</i>	S/C/Cd
SETMAN	Par: <codvoce\$>. Istruzione che richiama l'help in linea creato con [Ctrl+F9]. es.	S/C/Cd
VIMSG	Par: <tipo>, <msg\$>. Istruzione che visualizza finestra con messaggio. <i>Esempio: VIMSG 1, "messaggio".</i>	S/C/Cd/Sm/Cm
VIMSG_EXT	Par: <m\$>, <t\$>, [s\$] Istruzione che visualizza messaggio e imposta tasti funzione	S/C/Cd
RICCONF	Par: <m\$>, <t\$>, [s\$] Istruzione che visualizza messaggio di conferma impostabile.	S/C/Cd
Parametri Esecuzione (tasto funzione [F6])	Finestra dei parametri di esecuzione. E' una macro che consente di definire fino a 10 variabili (no di tipo array) la cui valorizzazione viene chiesta in una finestra di INPUT automatica all'esecuzione dello sprix che gestirà queste variabili. Nel campo 'Righe del modulo' si può stabilire la quantità di righe di stampa del modulo (compatibilmente a eventuali righe di stampa di tipo 'P'), altrimenti sono assunte quelle di configurazione del dispositivo di stampa scelto (campo obsoleto). I campi 'Nome' e 'Descrizione' della variabile sono di 12 e 20 caratteri; in ambienti a carattere la visualizzazione della descrizione è completata con '.....' fino al raggiungimento della descrizione più lunga presente. Il campo 'Ln' stabilisce la quantità di caratteri e il campo 'Dec' stabilisce il tipo di variabile: Numerica (inserendo il numero decimali da 0 a 6), Stringa (A o vuoto), Conto di Contabilità (C), Articolo di magazzino (M), Data in formato stringa (D) che richiede 'Ln' = 5, 8 o 10. Attiva le macro funzioni "monitor" del gestionale (la ricerca, la modifica, l'inserimento...). Le variabili sono utilizzabili ciclo tabellare (tasto funzione [F4]). Vedere anche istruzione INPUT. Nota: non è importabile nell'Editor attraverso la funzione di import del sorgente. es.	S
LOCATE	Par: <rig>, <col>. Istruzione che setta le coordinate del cursore. es.1 es.2	S/C/Cd
AUTOFIN()	Par: (). Funzione che ritorna il primo numero finestra disponibile. es.1 es.2 L'apertura di finestre in Collage fa assumere a quelle del gestionale una numerazione progressiva negativa.	S/C/Cd
APRIFIN	Par: <nfin>, <colore>, <rifnfin>, <rig>, <col>, <alt>, <larg>. Istruzione apre finestra. es. Possono essere aperte contemporaneamente fino a 9 finestre video più la finestra base del gestionale (corrisponde alla numero 0). Attenzione: la SETTF posiziona i tasti funzione nella finestra base. Per renderli visibili nell'interfaccia grafica successiva alla v2020J (v801) occorre avvalersi di APRIFIN_TIT o APRIFIN_EXT; in alternativa si rendono visibili spostando e fissando (con l'icona "spillo/puntina" presente nell'angolo in alto/destra della finestra del modulo che copre la finestra base in cui detti tasti funzione sono visualizzati).	S/C/Cd
APRIFIN_TIT	Par: <nfin>, <colore>, <rifnfin>, <rig>, <col>, <alt>, <larg>, <tit>, <tasti> Istruzione che apre finestra con titolo e tasti funzione	S/C/Cd
APRIFIN_EXT	Par: <nfin>, <rifnfin>, <rig>, <col>, <alt>, <larg>, <tit\$>, <pulsanti\$>, <aspetto\$> Estensione istruzione APRIFIN_TIT con le Icone e Tooltip	S/C/Cd
NUMPAGFIN	Par: <n1>, <n2>, <n3>, [s\$] Istruzione per impostare il numero pagina e pagine totali nella barra di APRIFIN_EXT	S/C/Cd
SETPRFIN	Par: <numfin>, <modo>. Istruzione che setta modalità proporzionale della finestra. es.	S/C/Cd
GRIGLIAFIN	Par: <n1>, <n2>, <n3>, <n4>, <n5>, <n6>, [s\$] Istruzione per gestire le caratteristiche della finestra con griglia righe "chiaro/scuro"	S/C/Cd
SEZIONEFIN	Par: <rig>, <col>, <alt>, <larg>, [tit\$] Istruzione per disegnare sezioni nella finestra attiva.	S/C/Cd
CLS_EXT	Par: <param\$>, <titolo\$> Istruzione per cancellare le sezioni impostate con SEZIONEFIN; per altre cancellazioni consultare CLS.	S/C/Cd
ATTIVAFIN	Par: <nfin>. Istruzione che attiva una finestra. es. Istruzione che attiva una finestra la finestra se precedentemente aperta. L'apertura di una finestra disattiva la precedente finestra attiva, poiché può esistere sempre e solo una finestra attiva.	S/C/Cd
_NFIN	Par: = N. Variabile numero di finestra attiva. In Collage, quelle del gestionale sono negative. es.	S/C/Cd
CERCACODFIN()	Par: (<IDfin\$>). Funzione che restituisce il numero di finestra	S/C/Cd
GETCODFIN\$()	Par: (<numfin>). Funzione che restituisce l'identificativo della finestra	S/C/Cd
GETFINTIT\$()	Par: (<numfin>) funzione lettura titolo finestra <numfin> (se = _NFIN è quella attiva)	S/C/Cr/Cd/Sm
CHIUDIFIN	Par: <nfin>. Istruzione che chiude una finestra. es.	S/C/Cd
CLS	Istruzione che cancella il contenuto video	S/C/Cd
DISVID	Par: <dis>. Istruzione che abilita/disabilita il video	S/C/Cd

COLOR	Par: <colore>. Istruzione che cambia colore del cursore. es.1 es.2	S/C/Cd
PRINT	Par: <espr\$>[.]. Istruzione che visualizza l'espressione a video. es.	S/C/Cd/Sm/Cm
PRINTD	Par: <str\$...>. Istruzione che visualizza in modo 'descrizione campo'. es.	S/C/Cd
PRINTF	Par: <n>,<nrip>. Istruzione che crea fincature a video. es.	S/C/Cd
PRINTICONA	Par: <icona\$>. Istruzione che visualizza icona a video. es.	S/C/Cd
TASTIERA		
INPUT	Par: <tipo\$>,<str\$>,<larg>. Istruzione immissione da tastiera. es.	S/C/Cd
_SCORRI	Condiziona l'istruzione INPUT	S/C/Cd
INPUT_TXT	Input multi-linee	S/C/Cd
NRIGDISPFIN()	Par: (<pulsanti_azione>). Numero righe disponibili (utili)	S/C/Cd
NCOLDISPFIN(,)	Par: (<barra_navigazione, pulsanti_scheda>). Numero colonne disponibili (utili)	S/C/Cd
SCROLL	Par: <rig>,<col>,<alt>,<larg>,<dir>. Istruzione che esegue lo scrolling di un'area video	S/C/Cd
_INTER	Attiva/disattiva l'interruzione del programma. es.	S/C/Cd
SETTF	Par: <str\$>. Sintassi = "<cod>[@idico][desc], , <cod>[@idico][desc]" Istruzione che setta i tasti funzione che sono attivi. Esempi SETTF "512Riga prec.,513,615" : SETTF "505@19Uscita Abbandonare,508,501Proseguì". es.	S/C/Cd
_TF	Controlla (o forza) la pressione di tasto funzione in INPUT.	S/C/Cd
INKEY()	Par: (.). Funzione che controlla la pressione di un tasto. es.	S/C/Cd
PUTTAST	Par: <str\$>. Istruzione che inserisce stringa in buffer di tastiera. es.	S/C/Cd
AZZTAST	Istruzione che cancella caratteri del buffer di tastiera.	S/C/Cd
DISTAST	Par: <dis>. Istruzione che disattiva l'uso della tastiera. es.	S/C/Cd
SETTATAST	Par: <tast\$>,[ag\$]. Istruzione che attiva tastiera personalizzata. es. Non consentita nel Livello di sviluppo (F2)	S/C/Cd
MOUSE		
INCAR()	Par:(opz\$). Funzione di controllo pressione del tasto rilevando le zone mouse. es.	S/C/Cd
SETFSM	Par:<nfin>. Istruzione che abilita finestra alla gestione mouse. Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/C/Cd
SETZM	Par:<riga>,<colonna>,<lungzonamouse>,<numzonamouse> Istruzione che imposta la zona del mouse Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/C/Cd
INPUT_EXT	Par:<numzonamouse>,<tipo\$>,<str\$>,<larg>,<maxlunstr>,<param\$> Istruzione di input a scorrimento in zona mouse (e gestione anche di file e directory) Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/C/Cd
INPUT_EXT_EXT	Par:<numzonamouse>,<tipo\$>,<str\$>,<larg>,<maxlunstr>,<param\$>,[elencoscelte\$] Estensione istruzione INPUT_EXT es.	S/C/Cd
INPUT3	Par:<numzonamouse>,<tipo\$>,<str\$>,<larg>,<maxlunstr>,<param\$>,<err>,[elencoscelte\$] Estensione istruzione INPUT_EXT_EXT es.	S/C/Cd
_TMZ	Variabile zona mouse (v. SETFSM, SETZM o INPUT_EXT)	S/C/Cd
_TMR	Numero riga finestra mouse (v. SETFSM, INPUT_EXT)	S/C/Cd
_TMC	Numero colonna finestra mouse (v. SETFSM, INPUT_EXT)	S/C/Cd
CREATELIST		
CREATELIST	Par: <nomelista\$>,<modolista>,<riga>,<colonna> Istruzione che esegue una finestra elenco i cui valori sono stabiliti dall'utente. es.	S/C/Cd
RIPTAFCT()	Par: ("par"); Funz. che ripristina globalmente tasti funzione creati da _CTTASTFUN\$().	S/C/Cr/Cd/Sm/Cm
_ERRCT e _ERRCT\$	Variabili di errore di CREATELIST	S/C/Cd
_CT	Variabili di struttura di CREATELIST	S/C/Cd

DIMENSIONI VIDEO

**Indipendentemente dalla risoluzione la gestione delle finestre certificata è
19 righe e 94 colonne per interfacce dal 2014 al 2019
25 righe e 110 colonne per interfacce dal 2020**

Le prime 4 righe e le ultime 2 righe sono riservate

Le dimensioni ottimali (19 x 94 per interfacce < 2020 e di 25 x 110 per interfacce dal 2020) sono quelle ottimali, per aprire finestre in tutte le risoluzioni video. L'utilizzo dell'interfaccia < 2014 consente 76 colonne (in alcuni casi anche 77 - 78) per 17 righe. Dal 2014 i limiti sono di 94 colonne (96 in certe condizioni con le barre di navigazione) per 19 righe; dal 2020 tali limiti sono aumentati di 6 righe e 16 colonne.

Questi suddetti sono i limiti per aprire finestre con tutte le risoluzioni video.

Le funzioni NRIGDISPFIN() e NCOLDISPFIN(,) ritornano la quantità di righe e colonne **disponibili**, in relazione alle variabili di dizionario "sxrig" e "sxcol" che, invece, ritornano il numero di righe e colonne **a video**; la variabile "sxclg" ritorna se la compatibilità collage è attiva.

Nell'interfaccia dal 2020, la differenza di "due righe" e di "una colonna", cioè riga disponibile 31 e riga video 29, colonna disponibile 136 e colonna video 135 (in finestra base sprix risultano 148 disponibile e 147 a video), erano già presenti nella precedente interfaccia (2014 < v < 2020) cioè, "riga disponibile 25" e "riga video 23", "colonna disponibile 120" e "colonna video 119".

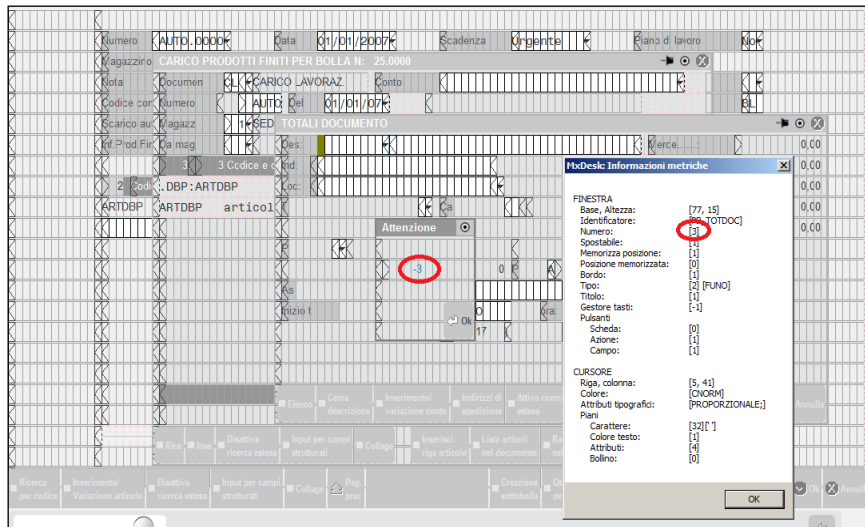
Tale suddetta differenza di due righe e una colonna non è mutata, bensì è aumentata le quantità di 6 righe e 16 colonne dell'interfaccia.

Esempio:

sxrig = NRIGDISPFIN(1)	interfaccia v < 2014
sxcol = NCOLDISPFIN(1,1)	interfaccia v < 2014
sxrig = NRIGDISPFIN(1) - 2	interfaccia 2014 < v < 2020 - certificate 19R x 94C/96C
sxcol = NCOLDISPFIN(1,1) - 1	interfaccia 2014 < v < 2020 - certificate 19R x 94C/96C

sxrig = NRIGDISPFIN(1) - 2 interfaccia v > 2020 - certificate 25R x 110C/112C
 sxc0l = NCOLDISPFIN(1,1) - 1 interfaccia v > 2020 - certificate 25R x 110C/112C

**Informazioni Video utili per comandi _NFIN, SCREEN\$, LOCATE, ecc..
 - premere [Ctrl+Spazio] e cliccare sulla finestra di interesse -**



Rispetto alle interfacce obsolete (precedenti al 2014), la SCREEN\$ può avere esito diverso poiché la posizione della medesima informazione può essere ubicata in una diversa posizione a video. Alcuni dati, come ad esempio il numero delle righe in emissione revisione documenti, si è mantenuta nella stessa posizione per non invalidare i Collage preesistenti, ma altre informazioni possono essere state posizionate in righe colonne diverse. Se la variabile dizionario "sxc0l" è > 80, significa che si sta utilizzando la nuova interfaccia, quindi occorre lavorare con le nuove dimensioni. La SCREEN\$ non è interessata dalla dimensione del carattere o dalla risoluzione video. Fa eccezione la SCREEN\$ per i dati dell'ultima riga di emissione revisione documenti in quanto tale finestra è soggetta al tipo di interfaccia che cambia le righe rappresentate a video.

Interfaccia carattere e grafica: alcune differenze

Negli anni l'interfaccia ha ricevuto notevoli modifiche e migliorie grafiche, che hanno comportato delle differenze di comportamento su alcune funzioni a video (es. SCREEN\$, GETPOSCURS,...) fino a non consentirne più l'utilizzo, come la _SINP\$ per campi di input su finestre di dialogo (es. messaggistica di conferma S/N nel messaggio "confermi operazione? N "). Se l'interfaccia in uso non consente il mantenimento di alcune funzionalità applicate su vecchie grafiche, ripristinare la vecchia grafica (es. pulsante chiave inglese + pulsante mano con indice + cambiare menu) oppure attivare il client con interfaccia in modalità LITE.

Definizioni generali dei pulsanti nelle finestre del Gestionale

Anagrafica articoli

Codice 00KBREND04 0404 Anag. prec. e succ. V 92841

Tipologia Mercì Struttura PROGRAMMATO

Descrizione GIUBBOTTO BRENDA DONNA Tasti tipo scheda

Unita' di Misura PZ Barra di navigazione

Altra U.M. T25 TAGLIA

Aliquota Iva 22

Costo ultimo

Prezzo

Note EX 11K1004 NUOVA DISTINTA BASE Destinata Scadenza

Carico

Magazzino TOT MAGAZZINI Pagine

Data progressivi

Esistenza

Disponibile Netto

Disponibile Lordo Tasti tipo azione

Descrizioni in lingue Tasti tipo campo Tasti magic

Valuta Seconda unita' di misura Annulla/Ripristina Elimina Ok Annulla

Nuovo Duplica Apri codice Filtro avanzato Modifica

Acquisisci docuvision Altri dati anagrafici

Condizioni contabili Magazzini/Progressivi

Condizioni commerciali Immagini articolo

Schede tecniche Fornitori

Visualizza righe documenti Dati Aggiuntivi

esempi e casi

Esempio: Albero di Natale (PRINTICONA)

```
GOSUB INIT
FOR I=1 TO 5
  GOSUB CICLO1
  GOSUB CICLO2
  GOSUB CICLO3
NEXT
END

CICLO1:
LOCATE 4,40 : PRINTICONA A6$ : SLEEP (200)
LOCATE 5,38 : PRINTICONA A2$:PRINTICONA A3$:PRINTICONA A4$ : SLEEP (200)
LOCATE 6,36 : PRINTICONA A3$:PRINTICONA A6$:PRINTICONA A8$:PRINTICONA A3$:PRINTICONA A2$ : SLEEP (200)
LOCATE 7,34 : PRINTICONA A3$:PRINTICONA A2$:PRINTICONA A1$:PRINTICONA A3$:PRINTICONA A8$
PRINTICONA A3$:PRINTICONA A6$ : SLEEP (200)
LOCATE 8,32 : PRINTICONA A3$:PRINTICONA A8$:PRINTICONA A5$:PRINTICONA A3$:PRINTICONA A2$
PRINTICONA A1$:PRINTICONA A3$:PRINTICONA A8$:PRINTICONA A3$: SLEEP (200)
LOCATE 9,30 : PRINTICONA A3$:PRINTICONA A2$:PRINTICONA A3$:PRINTICONA A8$:PRINTICONA A5$
PRINTICONA A3$:PRINTICONA A2$:PRINTICONA A1$:PRINTICONA A3$:PRINTICONA A8$:
PRINTICONA A3$: SLEEP (200)
LOCATE 10,28: PRINTICONA A3$:PRINTICONA A8$:PRINTICONA A3$:PRINTICONA A8$:PRINTICONA A5$
PRINTICONA A3$:PRINTICONA A2$:PRINTICONA A1$:PRINTICONA A3$:PRINTICONA A8$
PRINTICONA A3$:PRINTICONA A5$:PRINTICONA A6$: SLEEP (200)
LOCATE 11,38 :PRINTICONA A7$:PRINTICONA A7$:PRINTICONA A7$
LOCATE 12,38 :PRINTICONA A7$:PRINTICONA A7$:PRINTICONA A7$
RETURN

CICLO2:
IF C=1 THEN RETURN
C=1
LOCATE 12,1: PRINT "          B" : SLEEP (200)
LOCATE 12,1: PRINT "          BU" : SLEEP (200)
LOCATE 12,1: PRINT "          BUO" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON " : SLEEP (200)
LOCATE 12,1: PRINT "          BUON N" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON NA" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON NAT" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON NATA" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON NATALE" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON NATALE" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON NATALE !" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON NATALE !!" : SLEEP (200)
LOCATE 12,1: PRINT "          BUON NATALE !!!"
RETURN

CICLO3:
LOCATE 4,40 : PRINTICONA A6$
LOCATE 5,38 : PRINTICONA A4$:PRINTICONA A3$:PRINTICONA A5$
LOCATE 6,36 : PRINTICONA A3$:PRINTICONA A6$:PRINTICONA A8$:PRINTICONA A3$
PRINTICONA A2$
LOCATE 7,34 : PRINTICONA A6$:PRINTICONA A2$:PRINTICONA A7$:PRINTICONA A3$:PRINTICONA A6$
PRINTICONA A3$:PRINTICONA A5$
LOCATE 8,32 : PRINTICONA A1$:PRINTICONA A2$:PRINTICONA A4$:PRINTICONA A3$:PRINTICONA A3$
PRINTICONA A1$:PRINTICONA A3$:PRINTICONA A7$:PRINTICONA A3$
LOCATE 9,30 : PRINTICONA A3$:PRINTICONA A5$:PRINTICONA A3$:PRINTICONA A7$:PRINTICONA A6$
PRINTICONA A3$:PRINTICONA A2$:PRINTICONA A5$:PRINTICONA A3$:PRINTICONA A8$
PRINTICONA A3$
LOCATE 10,28: PRINTICONA A3$:PRINTICONA A2$:PRINTICONA A3$:PRINTICONA A1$:PRINTICONA A2$
```



```
PRINTICONA A3$:PRINTICONA A4$:PRINTICONA A2$:PRINTICONA A3$:PRINTICONA A8$
PRINTICONA A3$:PRINTICONA A6$:PRINTICONA A7$:
```

```
RETURN
```

```
INIT:
```

```
A1$="PALLINO_GIALLO"
A2$="PALLINO_ROSSO"
A3$="PALLINO_VERDE"
A4$="PALLINO_BLU"
A5$="PALLINO_CIANO"
A6$="PALLINO_MAGENTA"
A7$="PALLINO_GRIGIO"
A8$="PALLINO_BIANCO"
RETURN
```

Esempio: gestione di diverse finestre

Nota: nell'editor sorgente premere [F6] per completare la seguente impostazione:

```
+-----+
|Righe modulo di stampa:      |
|      Parametri              |
|Nome      Descrizione      Ln Dec|
|VAR$     Inserisci un Input  10  A |
+-----+
```

Il sorgente può proseguire nel seguente modo. Aprirà varie finestre (APRIFIN di varia natura) da consultare/imputare.

```
.....
'----- loop
_INTER=0
LOCATE 4,3 : PRINT "Con l'Input della finestra hai valorizzato una variabile con: "+VAR$
LOCATE 5,3 : PRINT "(la finestra di input che hai visto e' quella piu semplice, impostabile nei Parametri di Esecuzione [F6])."
LOCATE 7,3 : PRINT "Premi [F5] altrimenti non esco dal Loop e non proseguo."
LOOP:
  SLEEP 100: N=INKEY()
  IF N<>605 THEN GOTO LOOP ELSE LOCATE 9,3 : PRINT "Hai premuto il tasto..."; : IF N=605 THEN PRINT "[F5]"
  LOCATE 12,3 : PRINT "Apriro' una sequenza di APRIFIN 'diverse'"
  LOCATE 13,3 : PRINT "Per proseguire premi Invio, Esc o clicca il mouse nelle etichette e campi della finestra"
  '----- 1^ finestra
FINESTRAL:
  _INTER=0
  NF1=AUTOFIN()
  RFIN=0
  APRIFIN_TIT NF1,1,RFIN,0,0,20,50,"Titolo: Finestra 1","50IInvio,6100k,505Esc"
  SETTF "505Esc"
  ATTIVAFIN NF1
  SETFSM 1
  VIS:
  VIS01:
    SETZM 1,10,25,011
    COLOR 3: LOCATE 1,11: PRINT "RIGA-TMZ=11": COLOR 0
    COLOR 3: LOCATE 1,36: PRINT "RIGA-TMZ=0": COLOR 0
  VIS02:
    LOCATE 2,1: INPUT "W",A$,20: GOSUB INPUTMOUSE
  VIS03:
    SETZM 3,10,25,013
    COLOR 3: LOCATE 3,11: PRINT "RIGA-TMZ=13": COLOR 0
    COLOR 3: LOCATE 3,36: PRINT "RIGA-TMZ=0": COLOR 0
  VIS04:
    LOCATE 4,1: INPUT "WP",B$,20: GOSUB INPUTMOUSE
  VIS05:
    SETZM 5,10,25,015
    COLOR 3: LOCATE 5,11: PRINT "RIGA-TMZ=15": COLOR 0
    COLOR 3: LOCATE 5,36: PRINT "RIGA-TMZ=0": COLOR 0
  VIS06:
    LOCATE 6,1: INPUT_EXT 016,"W",C$,20,100,"CONFERMA": GOSUB INPUTMOUSE
  VIS07:
    LOCATE 7,1: INPUT_EXT 017,"WP",D$,20,100,"": GOSUB INPUTMOUSE
  VIS08:
    LOCATE 8,1: INPUT_EXT_EXT 018,"W",E$,20,100,"": GOSUB INPUTMOUSE
  VIS09:
    LOCATE 9,1: INPUT_EXT_EXT 019,"WP",F$,20,100,"CONFERMA": GOSUB INPUTMOUSE
  VIS10:
    LOCATE 10,1: INPUT3 020,"W",G$,20,100,"",ERR$: GOSUB INPUTMOUSE
  VIS11:
    LOCATE 11,1: INPUT3 021,"WP",H$,20,100,"CONFERMA",ERR$: GOSUB INPUTMOUSE
  VIS12:
    DIM ST1$(1): ST1$(1)=" adesso ": ST1$(2)=" utilizza ": ST1$(3)=" solo il click": ST1$(4)=" del mouse ":
    LOCATE 13,2: INPUT_TXT 023,ST1$(1),4,15,4,15,"": GOSUB INPUTMOUSE
    DIM ST2$(1): ST2$(1)=" stringa 1": ST2$(2)=" stringa 2":
    LOCATE 14,22: INPUT_TXT 024,ST2$(1),2,20,2,20,"": GOSUB INPUTMOUSE

  SETFSM 0
  VIS17:
    COLOR 1: R=18: C=2: LOCATE R,C: PRINT "COLORE 1";
    COLOR 0: R=18: C=11: LOCATE R,C: PRINT "COLORE NORMALE";
    AZZCRONO 1:
      CICLO: LOCATE 18,30: PRINT "Cronometro: ";: PRINT CRONO(1): IF CRONO(1) < 0.199 THEN GOTO CICLO
      VIMSG 1, "Aprò una 2^ finestra riferita a questa 1^ finestra che hai aperto, progressivamente, con
AUTOFIN ()"

  TASTO:
    _INTER=0
    T=INKEY()
    IF T=505 THEN GOTO CHIUDI1 ELSE GOTO FINESTRA2
  GOTO TASTO
  GOTO VIS

INPUTMOUSE:
  VIMSG 1, "hai cliccato su ZONAMOUSE: " + STR$( _TMZ,0,0)
  IF _TMZ=011 THEN GOTO VIS01 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=012 THEN GOTO VIS02 '(resta sul campo)
  IF _TMZ=013 THEN GOTO VIS03 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=014 THEN GOTO VIS04 '(resta sul campo)
  IF _TMZ=015 THEN GOTO VIS05 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=016 THEN GOTO VIS06 '(resta sul campo)
  IF _TMZ=017 THEN GOTO VIS07 '(resta sul campo)
```

```

IF _TMZ=018 THEN GOTO VIS08 '(resta sul campo)
IF _TMZ=019 THEN GOTO VIS09 '(resta sul campo)
IF _TMZ=020 THEN GOTO VIS10 '(resta sul campo)
IF _TMZ=021 THEN GOTO VIS11 '(resta sul campo)
IF _TMZ=022 THEN GOTO VIS12 '(resta sul campo)
RETURN

'----- 2^ finestra
FINESTRA2:
NF2=AUTOFIN()
APRIFIN_EXT NF2,NF1,0,0,20,101,"Titolo: Finestra 2 di sola lettura","A505,A610,503Finestra3","NOVISEZIONE"
SETTF "503Finestra3"
GRIGLIAFIN 1,1,1,1,10,20,""
DISVID 1
ATTIVAFIN NF2
'SETPRFIN NF2,1 : SETPRFIN NF2,3
SEZIONEFIN 00,00,12,52,"Sezione 1"
'SETPRFIN NF2,1
SEZIONEFIN 00,52,12,52,"Sezione 2"
SEZIONEFIN 12,00,5,103,"Sezione 3"
SEZIONEFIN 16,00,6,103,"Sezione 4"

FOR R=1 TO 9 : LOCATE R+1,01: PRINT "Riga "+STR$(R,0,0); :NEXT
'SETPRFIN NF2,0
FOR C=1 TO 101
LOCATE 14,C: PRINT STRING$(C MOD 10)=0,"|")+STRING$(C MOD 10)>0,STR$(C MOD 10),0,0);
NEXT
DISVID 0
T=INCAR(""):

'----- 3^ finestra
FINESTRA3:
NF3=AUTOFIN()
SETTF "6100k,505Esc,606Finestra4,512#1,513#2,510#3,511#4,602@11Ricerca|Tooltip,604Nuovo"

APRIFIN_EXT NF3,NF2,5,5,10,50,"Titolo: Finestra 3","6100k,505Esc,606Finestra4","IDFIN=ASSFTOTA" 'IDFIN funziona SOLO se NON
indico CSFONDO

COLOR 3: LOCATE 2,10:PRINT "Clicca i pulsanti e nella finestra": COLOR 0
SETFSM 1 : SETZM 2,10,10,111:

VIS2:
LOCATE 4,1: INPUT_EXT 99,"A",A$,20,100,""
LOCATE 6,2: PRINT "TF : " : LOCATE 6,6: PRINT _TF
LOCATE 7,2: PRINT "TMZ: " : LOCATE 7,6: PRINT _TMZ
LOCATE 8,2: PRINT "TMR: " : LOCATE 8,6: PRINT _TMR
LOCATE 9,2: PRINT "TMC: " : LOCATE 9,6: PRINT _TMC
IF _TF=606 THEN GOTO FINESTRA4
IF _TF=505 THEN GOTO CHIUDI2 ELSE GOTO VIS2
GOTO VIS2:

'----- 4^ finestra
FINESTRA4:
_INTER=0
NF4=AUTOFIN()
APRIFIN_EXT NF4,NF3,0,0,20,101,"Titolo:", "6100k,501Invio,505Esc",""
SETTF "6100k,501Invio,505Esc"
ATTIVAFIN NF4 : SETFSM NF4 'visualizza l'icona [F2] per la COMBOBOX (incompatibile con attivazione in APRIFIN_EXT di
"602@11Ricerca|Tooltip")

FOR RIGA = 1 TO 4
COLOR 0: LOCATE RIGA,08: IF RIGA MOD 2 <> 0 THEN PRINTICONA ("PALLINO_ROSSO")
COLOR 0: LOCATE RIGA,08: IF RIGA MOD 2 = 0 THEN PRINTICONA ("PALLINO_GIALLO")
NEXT

LOCATE 5,1 : PRINT "ALFANUM: " : PRINT "CONTO: "
COD2$="GMM.CCCCC" 'imposto la variabile 'COD2$' per ricercare una stringa...

STR:
LOCATE 5,10: INPUT "A",COD1$,10
IF _TF=505 THEN GOTO CHIUDI4 : GOTO FINE
IF _TF=501 THEN GOTO CTO 'IF _TF=-1 THEN GOTO CTO
GOTO STR

CTO:
LOCATE 6,10: PRINT "(cerca il codice conto [F2]) ";: PUTTAST COD2$: INPUT "C",COD2$,10
IF _TF=505 THEN GOTO CHIUDI4 : GOTO FINE
IF _TF=501 THEN GOTO EXT 'IF _TF=-1 THEN GOTO EXT
GOTO CTO

EXT:
DIM ARRAY$() : _INTER = 0 : GOSUB CMB
LOCATE 9,1 : PRINT "input_ext_ext combo (premi Elenco [F2]): ";: PUTTAST " " : INPUT_EXT_EXT
NF4,"COMBO",VAR1$,1,1,"",ARRAY$();:
IF _TF=505 THEN GOTO CHIUDI4 : GOTO FINE
IF _TF=501 THEN LOCATE 10,1 : PRINT "input_ext_ext check con mouse/spazio : ";: INPUT_EXT_EXT
NF4,"CHECK",VAR2,1,1,"",ARRAY$
LOCATE 11,1 : PRINT "wait" : SLEEP 1000
LOCATE 12,1 : PRINT "input_ext_ext medesima_combo (premi Elenco [F2]): ";: INPUT_EXT_EXT
NF4,"COMBO",VAR3$,10,10,"",ARRAY$
VIMSG 1, "Fine esercizio, per uscire premi Esc"
GOTO EXT

CMB:
VAR1$="pippo"
AZZARRAY ARRAY$() : ARRAY$(1)="1=opzione A": ARRAY$(2)="2=opzione B": ARRAY$(3)="3=opzione C": ARRAY$(4)="4=Includi":
ARRAY$(5)="5=Escludi"
RETURN

CHIUDI4:
VIMSG 1, "chiudo 4^ finestra": CHIUDIFIN NF4
CHIUDI3:
VIMSG 1, "chiudo 3^ finestra": CHIUDIFIN NF3
CHIUDI2:
VIMSG 1, "chiudo 2^ finestra": CHIUDIFIN NF2
CHIUDI1:
VIMSG 1, "chiudo 1^ finestra": CHIUDIFIN NF1

FINE:
END

```

Esempio Video Tastiera

```

Parametri di esecuzione, tasto funzione [F6]
+-----+
|Righe modulo:                               |
|      PARAMETRI DI ESECUZIONE              |
|Nome      Descrizione      Ln Dec|
|DAT$      Dalla Data      10 D |
|ADAT$     Alla Data      10 D |
|CTO$      Conto Contabile  9 C |
|ART$      Articolo      20 M |
|NUM       Numero a 2 decimali 10 2 |
|ALF$     Stringa      12 A |
+-----+
M1$="i seguenti valori sono utilizzabili con variabili nel programma"
M2$=DAT$+ " - " +ADAT$+ " - " +CTO$+ " - " +ART$+ " - " +STR$(NUM,0,0)+ " - " +ALF$
VIMSG 1,M1$+"|" +M2$
LOCATE 10,10 : PRINT"Questo INPUT di Magazzino ha un SETMAN (premi F1)":
LOCATE 11,35 : SETMAN "cod1" : INPUT "M",ART$,20 : SETMAN "" : CLS
_INTER=1
NF=AUTOFIN() : APRIFIN NF,1,0,2,6,15,45
DIM N$( ) : FOR N=1 TO 10 : N$(N)=STR$(N,0,0) : NEXT : DIM S$(10)
LOCATE 1,4 : PRINT "TITOLO: finestra di PRINT e INPUT video "
LOCATE 2,1 : PRINT STRING$(45,"-")
LOCATE 3,1:COLOR 4:PRINT"Input per avanzamento..."
LOCATE 4,1 : COLOR 3:PRINT"n.0"+"1" : LOCATE 4,6 : INPUT"AE",S$(1),10
LOCATE 4,16:COLOR 2:PRINT"n.0"+N$(2) : LOCATE 4,21:INPUT"AE",S$(2),10
LOCATE 4,31:COLOR 1:PRINT"n.0"+N$(3) : LOCATE 4,36:INPUT"AE",S$(3),10
LOCATE 5,1:COLOR 0:PRINT" "

LOCATE 6,1:PRINT"Input tutto visibile... "
LOCATE 7,1:PRINT"n.0"+"4":LOCATE 7,16:PRINT"n.0"+N$(5)
LOCATE 7,31:PRINT"n.0"+N$(6)
LOCATE 8,1:PRINT"n.0"+"7":LOCATE 8,16:PRINT"n.0"+N$(8)
LOCATE 8,31:PRINT"n.0"+N$(9)
LOCATE 9,1:PRINTF 3,1:LOCATE 9,2 :PRINTF 1,34:LOCATE 9,34 :PRINTF 9,1
LOCATE 9,35 :PRINTF 1,10:LOCATE 9,45 :PRINTF 4,1
LOCATE 10,1:PRINTF 2,1:LOCATE 10,34:PRINTF 2,1:LOCATE 10,45:PRINTF 2,1
LOCATE 11,1:PRINTF 2,1:LOCATE 11,34:PRINTF 2,1:LOCATE 11,45:PRINTF 2,1
LOCATE 12,1:PRINTF 5,1:LOCATE 12,2:PRINTF 1,34:LOCATE 12,34:PRINTF 8,1
LOCATE 12,35:PRINTF 1,10:LOCATE 12,45:PRINTF 6,1
LOCATE 7,6 : INPUT"AE",S$(4),10:LOCATE 7,21:INPUT"AE",S$(5),10
LOCATE 7,36:INPUT"AE",S$(6),10
LOCATE 8,6 : INPUT"AE",S$(7),10:LOCATE 8,21:INPUT"AE",S$(8),10
LOCATE 8,36:INPUT"AE",S$(9),10

'-----
S$(10)=" proposta " : NUM=1234
LOCATE 10,2:PRINTD " n."+N$(10)+" (INPUT proposta).....:"
LOCATE 11,2:PRINTD " n.11 (SCORRE INPUT come PRINT):"
LOCATE 10,35:INPUT "AE",S$(10),10
_SCORRI=1 : LOCATE 11,38 : INPUT "N",NUM,4 : _SCORRI=0 'Input = Print

'-----
GIRO: R=0 '(input C/M/ecc..) R è il flag per il giro finestra2
LOCATE 13,2 : PRINT"Tipo cod. M=Mag./C=Conti/F=Finestra: "; : INPUT"AM",TIPO$,1
IF TIPO$="F" THEN GOSUB NUOVAFIN2 ELSE IF TIPO$<>"M" AND TIPO$<>"C" THEN GOTO GIRO
IF R=1 THEN GOTO GIRO
LOCATE 14,5 : PRINT"CODICE: ";: INPUT TIPO$,COD$,60
M1$="<Messaggio allineato a sinistra:|<...hai inserito il codice "
M2$="...>|"+"|"+"ORA SETTO IL TASTO FUNZIONE|F8: Nuova Finestra 2"
VIMSG 1, M1$ + COD$ + M2$
SETTF "505Interruzione,501Proseguì,608Nuova Finestra2"
IF _TF=501 THEN GOTO LEGGIVIDEO
IF _TF=608 THEN GOSUB NUOVAFIN2 ELSE GOTO LEGGIVIDEO
LEGGIVIDEO:
LOCATE 15,5:PRINT"DATA : ";:INPUT "D",D$,10
Y=GETPOS CURSY(1) : X=GETPOS CURSX(1) : DATA$=SCREEN$(NF,Y,X,10)
IF D$="" OR DATA$="" THEN DATA$="blank"
VIMSG 4, ">Messaggio allineato a destra|Data inserita = ' +DATA$+ " '"

NUOVAFIN2:
BEEP: M1$="Emette un ' beep ' di avvertimento..."
M2$="...e apro un altro tipo di finestra|con altri controlli"
VIMSG 2, M1$+"|" +M2$
NUOVAFIN2A:
_INTER=0
NF=AUTOFIN()
APRIFIN NF,4,0,6,23,7,45
SETPRFN NF,3: SETTF "501Invio,512Su,610Conferma,606NuovaFinestra3,505Esci"
LOCATE 1,1:PRINTD "Dalla Data :":LOCATE 2,1:PRINTD "Alla Data :":
LOCATE 3,1:PRINTD "Conto Contabile:":LOCATE 4,1:PRINTD "Numero 6 dec.li:":
COLOR 4
LOCATE 1,18:PRINT STRING$(10," "):LOCATE 2,18:PRINT STRING$(10," ")
LOCATE 3,18:PRINT STRING$(25," "):LOCATE 4,18:PRINT STRING$(15," ")
CAMPO1: LOCATE 1,18:INPUT "D",DDATA_$,10
IF _TF=512 THEN GOTO CAMPO1
IF _TF=501 THEN GOTO CAMPO2
IF _TF=610 THEN GOTO CONTROLLO
IF _TF=606 THEN GOSUB NUOVAFIN3
IF _TF=505 THEN GOTO CHIUDIF2
CAMPO2: LOCATE 2,18:INPUT "D",ADATA_$,10
IF _TF=512 THEN GOTO CAMPO1
IF _TF=501 THEN GOTO CAMPO3
IF _TF=610 THEN GOTO CONTROLLO
IF _TF=606 THEN GOSUB NUOVAFIN3
IF _TF=505 THEN GOTO CHIUDIF2
CAMPO3: LOCATE 3,18:INPUT "C",CONTO_$,25
IF _TF=512 THEN GOTO CAMPO2
IF _TF=501 THEN GOTO CAMPO4
IF _TF=610 THEN GOTO CONTROLLO
IF _TF=606 THEN GOSUB NUOVAFIN3
IF _TF=505 THEN GOTO CHIUDIF2
CAMPO4: LOCATE 4,18:INPUT "N",NDEC,15
IF _TF=512 THEN GOTO CAMPO3
IF _TF=501 THEN GOTO CAMPO1
IF _TF=610 THEN GOTO CONTROLLO

```

```

IF _TF=606 THEN GOSUB NUOVAFIN3
IF _TF=505 THEN GOTO CHIUDIF2
GOTO CAMPO1
CONTROLLO:
IF DDATA $="" THEN VIMSG 5, "Campo Obbligatorio":GOTO CAMPO1
IF ADATA $="" THEN VIMSG 5, "Campo Obbligatorio":GOTO CAMPO2
IF CONTO $="" THEN VIMSG 5, "Campo Obbligatorio":GOTO CAMPO3
IF NDEC=0 THEN VIMSG 5, "Campo Obbligatorio":GOTO CAMPO4
IF DDATA $>ADATA $ AND ADATA $<>" THEN VIMSG 4, "date errate":GOTO CAMPO1
VIMSG 1, "apertura di una terza finestra"
NUOVAFIN3:
NF=AUTOFIN()
_INTER=0
APRIFIN NF,1,0,14,52,4,24
SETTF"501Proseguì,505Fine"
AZZCRONO 1
CICLO:
DISTAST 1
LOCATE 1,1: PRINT "Finestra Numero = "+STR$(NF,0,0)
LOCATE 2,1: PRINT "Blocca Tasti x 5 Sec. "+STR$(S,0,0)
S=CRONO(1): IF CRONO(1)<6 THEN GOTO CICLO: AZZTAST: AZZCRONO 1
DISTAST 0
PAUSA:
LOCATE 3,1: PRINT "Premi Fine o Proseguì "
T=INKEY()
IF T<>501 AND T<>505 THEN SLEEP 100 : GOTO PAUSA ELSE GOTO PROSEGUI
PROSEGUI:
IF T=505 THEN GOTO FINE ELSE IF T=501 THEN GOTO SCROLLA
SCROLLA: CHIUDIFIN NF:ATTIVAFIN NF-1: INTER=1:APRIFIN NF,1,0,3,54,9,10
RIP:X=X+1:PRINT "Sprix "+STR$(X,0,0):SLEEP 500:IF X<8 THEN GOTO RIP
PRINT "Sprix 9";: SCROLL 1,1,19,10,1
LOCATE 9,1: PRINT "COLLAGE"
SETTATAST "tastsp": INPUT "AE",A$,7
VIMSG 1, "Imposta INPUT con quanto | stabilito dal pulsante scelto"
CHIUDIFIN 1 : ATTIVAFIN 0 : _INTER=1 : R=0
FOR I=1 TO 15 : DISVID 1
PUTTAST "Sprix#501"
LOCATE 3+I,20+I : PRINT "Campo n."+STR$(I,0,0)+" ";: INPUT "AM",A$,5
NEXT : DISVID 0
FINE: VIMSG 2, "FINE ESERCIZIO": END
CHIUDIF2: R=1: CHIUDIFIN 2
DDATA $="" : ADATA $="" : CONTO $="" : NDEC=0
_INTER=1: ATTIVAFIN 1 : GOTO GIRO

```

Esempio: inizializzazione di campo data in INPUT

Per inizializzare un'istruzione INPUT di tipo data ("D") occorre considerare che il gestionale memorizza le date nel formato AAAAMMGG. Anche se si utilizza un campo data con 10 caratteri (ad esempio "01/05/2002") viene memorizzato il formato sopra scritto ("20020501").

```

_INTER=0
N_FIN=AUTOFIN()
COLORE=2
APRIFIN N_FIN,COLORE,0,5,10,1,30
SETTF "505Esc"
DATA$="20020501"
LOCATE 1,1
PRINT "INSERISCI DATA...";
DATA:
INPUT "D",DATA$,10
IF _TF=505 THEN GOTO FINE
GOTO DATA
FINE:
CHIUDIFIN N_FIN
END

```

Caso: alternativa all'azzeramento del campo di input settato di tipo "C", "A", "M", ecc..

```

_INTER=0
N_FIN=AUTOFIN()
COLORE=3
APRIFIN N_FIN,COLORE,0,1,1,10,50
SETTF "501,505"
LOCATE 1,1
PRINT "CAMPO 1: "
PRINT "CAMPO 2: "
'IMPOSTO LA VARIABILE 'COD2$' PER RICERCARE UNA STRINGA...
COD2$=","F"
CAMPO1:
LOCATE 1,10
INPUT "A",COD1$,10
IF _TF=501 THEN GOTO CAMPO2
IF _TF=505 THEN GOTO FINE
GOTO CAMPO1
CAMPO2:
LOCATE 2,10
INPUT "C",COD2$,10 'questo campo di input viene settato e di conseguenza a video risulta vuoto anzichè = ",F";
'per generare una proposizione in input a video occorre anticipare (e gestire) con PUTTAST COD2$;
IF _TF=501 THEN GOTO CAMPO1
IF _TF=505 THEN GOTO FINE
GOTO CAMPO2
FINE:
CHIUDIFIN N_FIN
END

```

INPUT campo di tipo password per digitare caratteri non visualizzati

Non è disponibile un simile campo, ma è previsto un work around su finestra di partenza (in cui digitare la password) di tipo non spostabile oppure la finestra base. Per finestre non spostabili si intendono le finestre di colore 11, 12, 13, 14, 15 e 19 finestre senza bordo.

```

LOCATE 1,1: PRINT "Password ":
APRIFIN 1,11,0,1,11,1,20
ATTIVAFIN 0
LOCATE 1,11: INPUT "A",PWD$,10:
CHIUDIFIN 1

```

INPUT: concatenazione di stringa

Di seguito le due soluzioni più utilizzate:

- 1) più LOCATE ognuna delle quali riporta la porzione di stringa che andrà a costituire quella "intera" che provvederemo a comporre in ambito separato;
- 2) attraverso una subroutine che si avvalga dell'istruzione INKEY, in modo da poter leggere da tastiera la battitura sequenziale di ogni singolo carattere, per comporre a video l'intera stringa, anche se spezzata in più parti.

Caratteri Ammessi in visualizzazione ed INPUT "A"/"AM"

I caratteri ammessi in un campo di visualizzazione ed input sono: per le tipologie AM dal numero (tabella ASCII) 75 al 90; per le tipologie A dal numero 32 al 126. Occorre tuttavia testare quelli di input considerazione dei font utilizzati ed interfaccia.

Esempi SETTATAS**Esempio Sprix: data la tastiera "tastspk"**

```
D$="": N$="Str$": LOCATE 5,1: INPUT "A",N$,30: SETTATAS "tastspk"
N$="Str$+Tast": LOCATE 5,1: INPUT "A",N$,30: LOCATE 6,1: PRINT "DATA";:
INPUT "D",D$,10: SETTATAS "tastspk": LOCATE 6,1: PRINT "Data+Tast";:
INPUT "D",D$,10 ....
```

Esempio Collage

SETTATAS "tastcol","a" (l'utente può così scegliere una fra le due tastiere (la "standard" e la "tastcol") sullo stesso campo.

Esempio Collage

"ESEGUCOLLAGE:ETICHETTA".

Tastiera 'tastcol' con tre pulsanti in cui uno ha 'ESEGUCOLLAGE:ETICHETTA2' e Tastiera 'ivacod' con pulsante aggiuntivo che si attiva all'ingresso del campo iva in anagrafica articolo (E SENZA L'UTILIZZO DI SETTATAS); dati due Collage "Colla" e "CollB" entrambi compilati nel menu 'MA'

CollA	CollB
-----	-----
ON IN MADSART:	
SETTATAS "tastcol"	ETICHETTA3:
ETICHETTA1:	VIMSG 4, "ETIC.3 da ESEGUCOLLAGE"
VIMSG 4, "ETIC.1 su ON_IN_MADSART"	END
END	
ETICHETTA2:	
VIMSG 4, "ETIC.2 da ESEGUCOLLAGE"	
END	

Esempio INKEY:

```
_INTER=0
PRINT "Premi F5 altrimenti non termina il Loop"
LOOP:
SLEEP 100
N=INKEY()
IF N<>605 THEN GOTO LOOP ELSE PRINT "Hai premuto il tasto...";
IF N=605 THEN PRINT "F5"
INPUT "A",A$,10
_INTER=1
END
```

Esempio _NCAM\$:

Collage Apre una finestra che riporta la descrizione del campo di INPUT in cui si trova il cursore (compilazione in menu AUTO)

```
ON_START:
FLAG=1
END

ON_IN_ALL:
IF FLAG=1 THEN GOTO SALTO1
GOSUB FINESTRA
SALTO1:
END

ON_OUT_ALL:
IF FLAG=1 THEN GOTO SALTO2
CHIUDIFIN NUMFIN
SALTO2:
FLAG=0
END

FINESTRA:
NUMFIN=AUTOFIN() : X=GETPOS CURSX(2) : Y=GETPOS CURSY(2)
L=LEN(_NCAM$)
IF Y=20 THEN Y=18
APRIFIN NUMFIN,11,_NFIN,Y+1,X,1,10+L
LOCATE 1,1:COLOR 3: PRINT "Cod.Camp.:"+_NCAM$;
RETURN
```

Esempio: APRIFIN con visualizzazione dei tasti funzione attivi della finestra personalizzata (nota: gestibile invece con APRIFIN_TIT)

```
_INTER=0
N_FIN=AUTOFIN()
COLORE=4
LARGH=20
N_RIGHE=4
APRIFIN N_FIN,COLORE,0,3,20,N_RIGHE,LARGH
ATTIVAFIN N_FIN
SETPRFIN N_FIN,1 '(potrebbe doversi commentare a seconda del tipo di interfaccia)
COLOR 5
PRINT "FUNZIONI ATTIVE"
PRINTF 1,LARGH
COLOR 0
PRINT "Esc Uscita"
```

```
PRINT "F9 Tasti funzione";
DIGITA:
TASTO=INKEY()
IF TASTO=505 THEN GOTO FINE
GOTO DIGITA
FINE:
CHIUDIFIN N_FIN
END
```

Esempi _SINP\$ (come automatizzare l'input sottoposto a condizioni)

Qualora si rendesse necessaria la creazione di un processo automatico che sia in grado di modificare (tramite inserimento / aggiornamento / eliminazione) le informazioni contenute in un archivio (negli esempi che seguono si tratta dell'archivio dei documenti di magazzino) si potrebbe utilizzare la variabile _SINP\$ all'interno di un Collage.

ESEMPIO 1: L'esempio che segue permette, in base all'importo totale del documento, di decidere se inserire o meno un articolo all'interno del documento stesso.

```
ON_IN_MDCODART:
RIGA_RIF=VAL(SCREEN$(0,7,1,4))
IF APPENDIS<>"SI" OR _SINP$<>" " OR RIGA_RIF<>1 THEN END
_SINP$="911RS"
PUTTAST "#501"
END
```

```
ON_IN_MDQTA:
IF APPENDIS<>"SI" OR _SINP$<>" " OR RIGA_RIF<>1 THEN END
_SINP$="3"
PUTTAST "#501"
END
```

```
ON_IN_MDPZZO:
IF APPENDIS<>"SI" OR RIGA_RIF<>1 THEN END
PUTTAST "#501"
END
```

```
ON_IN_MDSCON:
IF APPENDIS<>"SI" OR _SINP$<>" " OR RIGA_RIF<>1 THEN END
_SINP$="20+10+5"
APPENDIS="NO"
PUTTAST "#501"
END
```

```
ON_IN_MDCODPG:
IF APPENDIS="NO" THEN END
TOTDOC$=SCREEN$(-1,14,64,12)
TRASFORMA:
POS=INSTR(TOTDOC$,".")
IF POS=0 THEN GOTO FINE_TRASF
TOTDOC$=LEFT$(TOTDOC$,POS-1)+RIGHT$(TOTDOC$,LEN(TOTDOC$)-POS)
GOTO TRASFORMA
FINE_TRASF:
TOTDOC=VAL(TOTDOC$)
IF TOTDOC>1000000 THEN APPENDIS="NO": END
APPENDIS="SI"
PUTTAST "#505#607"
END
```

ESEMPIO 2: L'esempio che segue permette di inserire la quantità di un articolo a taglie in modo automatico e, in un secondo momento, permette all'utente di verificare se la quantità inserita corrisponde con la quantità che si desidera inserire.

```
ON_IN_MDCODART:
RIGA_RIF=VAL(SCREEN$(0,7,1,4))
END
```

```
ON_IN_MDQTATGL:
AZZVARSYS 4
GETATTMM
IF _MMQTA(RIGA_RIF)>0 THEN END
I=I+1
IF I<3 THEN PUTTAST "#513":END
_SINP$=STR$(10,0,0)
IF I<4 THEN PUTTAST "#610"
END
```

```
ON_IN_MDQTA:
I=0
END
```

ESEMPIO 3: Qualora si conosca la descrizione della taglia (ad esempio la taglia XL), ma non l'indice a cui è associata la taglia stessa l'esempio 2 potrebbe essere riscritto come segue:

```
ON_IN_MDCODART:
RIGA_RIF=VAL(SCREEN$(0,7,1,4))
END
```

```
ON_IN_MDQTATGL:
TAGLIA$="XL"
AZZVARSYS 4
GETATTMM
IF _MMQTA(RIGA_RIF)>0 THEN END
I=I+1
IF mtdes(VAL(RIGHT$(arum2(_MMART$(RIGA_RIF)),1)),I)<>TAGLIA$ THEN PUTTAST "#513":END
_SINP$=STR$(10,0,0)
PUTTAST "#610"
END
```

```
ON_IN_MDQTA:
I=0
END
```

Caso: _SINP\$ perché non funziona con mouse o F2

_SINP\$ si valorizza in entrata o in uscita da un campo a seconda dell'evento stabilito. Se l'evento è comandato da tastiera, significa che il campo può essere già stato valorizzato, ma se l'evento (ad es. in uscita dal campo) è generato da mouse o F2, il campo non è ancora stato valorizzato, per cui _SINP\$ risulta vuota.

Per esempio, `_SINP$` in uscita dal campo `ON_OUT_MDTDOC` è valorizzato se l'input è da tastiera (si può visualizzare il valore con messaggio dopo l'utilizzo della tastiera), ma se è utilizzato il mouse o F2, l'uscita è da un campo non ancora valorizzato.

Es. `ON_OUT_MDTDOC: SIG$=LEFT$(_SINP$,2)`

In questo caso è utile una funzione di lettura a video con `SCREEN$` oppure predisporre il codice del programma in modo tale che torni sul campo (prevedendo e anticipando eventuali situazioni di loop) e valorizzi `_SINP$`. Ricordiamo infine di valutare le caratteristiche di `ON_STARTMM`.

Es. `ON_IN_MDCAU: SIG$=SCREEN$(0,2,8,5)`

Esempio TASTI:

```
'dato il codice articolo 'ART' e il Collage con la pressione di F12 sul campo note del menu 'MD':
ON_612_MDNOTE: CODART$="A" : PUTTAST "#501" : DISTAST 1 : DISVID 1
FOR I=1 TO 10 : PUTTAST CODART$+"#501#501#501#501": NEXT: END
```

Esempio:

```
'Collage in menu 'CA'
ON_612_CARICCOD: PUTTAST"#602@TEMPO:#501@TEMPO:#501@TEMPO:#505#505#505" : END
TEMPO: AZZCRONO 1
CICLO: IF CRONO(1)<3 THEN GOTO CICLO: END
```

CASO: l'evento ON_OUT può rilevare solo _TF=501

La variabile `_TF` controlla in `INPUT` e in evento `ON_OUT`, specialmente attraverso `SETTF`. `_TF` nelle finestre dialogo di conferma (per esempio `ON_OUT_MAAUOTC` del Gestionale, ovvero la finestra di conferma dell'autocodifica per articoli strutturati), specialmente quelle di conferma dell'operazione, è disponibile solo l'Invio (e/o tasti ferccia) in quanto `Esc/F10` sono una mappatura logica della risultante del valore `N/S + Invio`. A seconda del default della finestra (S o N), il tasto funzione (`Esc` o `F10`) cambia in automatico il valore della finestra di dialogo (da S a N o viceversa). Le descrizioni e i valori possono essere letti con `SCREEN$` e `_SINP$`.

Esempio SETTATASTI:

Esempio Sprix

```
'data la tastiera "tastspk":
D$="": N$="Str$": LOCATE 5,1: INPUT "A",N$,30: SETTATAST "tastspk"
N$="Str$+Tast": LOCATE 5,1: INPUT "A",N$,30: LOCATE 6,1: PRINT "DATA":;
INPUT "D",D$,10: SETTATAST "tastspk": LOCATE 6,1: PRINT "Data+Tast";;
INPUT "D",D$,10 ....
```

Esempio Collage

Data "ESEGUCOLLAGE:ETICHETTA". Tastiera 'tastcol' con tre pulsanti in cui uno ha 'ESEGUCOLLAGE:ETICHETTA2' e Tastiera 'ivacod' con pulsante aggiuntivo che si attiva all'ingresso del campo iva in anagrafica articolo (e senza uso di `SETTATAST`); dati due Collage "CollA" e "CollB" entrambi compilati nel menu 'MA'

Collage A	Collage B
<pre>ON_IN_MADSART: SETTATAST "tastcol" ETICHETTA1: VIMSG 4, "ETIC.1 su ON_IN_MADSART" END ETICHETTA2: VIMSG 4, "ETIC.2 da ESEGUCOLLAGE" END</pre>	<pre>ETICHETTA3: VIMSG 4, "ETIC.3 da ESEGUCOLLAGE" END</pre>

Esempio INCAR: aprire i primi tre campi di input premendo più volte Invio, poi cliccare su un campo di Input. Il cursore è visualizzato posizionato all'altezza del campo selezionato e, dopo due secondi, viene aperto un quarto campo di Input.

```
APRIFIN 1,1,0,0,0,10,50 : COLOR 3 : SETFSM 1
SETZM 3,10,10,013 : SETZM 4,10,10,014 : SETZM 5,10,10,015
```

```
VIS:
LOCATE 3,1: INPUT_EXT 013,"A",A$,20,100,"" : GOSUB VISCUR
LOCATE 4,1: INPUT_EXT 014,"A",A$,20,100,"" : GOSUB VISCUR
LOCATE 5,1: INPUT_EXT 015,"A",A$,20,100,"" : GOSUB VISCUR
GOTO VIS:
```

```
VISCUR:
IF _TMZ=013 OR _TMZ=014 OR _TMZ=015 THEN VISCU=INCAR("VISCURS") : SLEEP 2000 ELSE RETURN
LOCATE 6,1: INPUT_EXT 016,"A",A$,20,100,""
```

Esempio, cliccando una qualsiasi area della finestra sono riportate le coordinate di riga/colonna (TMR/TMC) in cui è avvenuto il "click". Cliccando sulla zonamouse relativa alla PRINT "XXXXXXXXXX" viene rilasciato il codice TMZ=111, mentre su zonamouse della INPUT_EXT il codice TMZ è 99.

```
APRIFIN 1,1,0,0,0,10,50
COLOR 3: LOCATE 2,10:PRINT "XXXXXXXXXX": COLOR 0
SETFSM 1
SETZM 2,10,10,111:
```

```
VIS:
LOCATE 3,1: INPUT_EXT 99,"A",A$,20,100,""
LOCATE 4,1: PRINT "TF : " : LOCATE 4,5: PRINT _TF
LOCATE 5,1: PRINT "TMZ: " : LOCATE 5,5: PRINT _TMZ
LOCATE 6,1: PRINT "TMR: " : LOCATE 6,5: PRINT _TMR
LOCATE 7,1: PRINT "TMC: " : LOCATE 7,5: PRINT _TMC
GOTO VIS
```

'attenzione: nell'esempio sopra la presenza di `INPUT` è funzionale per evitare che il programma entri in `LOOP` (e le variabili non si valorizzerebbero, non farebbero in tempo a sentire il click del mouse che già sono riprocessate/inizializzate)

Esempio, aprire i tre campi di input con Invio (ad ogni Invio visualizza _TMZ=0), poi cliccare con il mouse su ciascun campo di Input per visualizzare quale _TMZ è stata cliccata (12, 14, 16), poi cliccare sulla riga 1 o riga 3 o riga 5, sia i campi Print (con visualizzazione della relativa _TMZ) che le zone vuote (in particolare dalla colonna 1 alla 9, dalla 22 alla 34, dalla 46 alla 50) che corrisponderanno al valore _TMZ=0.

```
_INTER=1
APRIFIN 1,1,0,0,0,10,50 : SETFSM 1
```

```

VIS:

VIS01:
  SETZM 1,10,25,011
  COLOR 3: LOCATE 1,11 : PRINT "RIGA-TMZ=11" : COLOR 0
  COLOR 3: LOCATE 1,36 : PRINT "RIGA-TMZ=0" : COLOR 0

VIS02:
  LOCATE 2,1: INPUT_EXT 012,"A",A$,20,100,"" : GOSUB INPUTMOUSE

VIS03:
  SETZM 3,10,25,013
  COLOR 3: LOCATE 3,11 : PRINT "RIGA-TMZ=13" : COLOR 0
  COLOR 3: LOCATE 3,36 : PRINT "RIGA-TMZ=0" : COLOR 0

VIS04:
  LOCATE 4,1: INPUT_EXT 014,"A",B$,20,100,"" : GOSUB INPUTMOUSE

VIS05:
  SETZM 5,10,25,015
  COLOR 3: LOCATE 5,11 : PRINT "RIGA-TMZ=15" : COLOR 0
  COLOR 3: LOCATE 5,36 : PRINT "RIGA-TMZ=0" : COLOR 0

VIS06:
  LOCATE 6,1: INPUT_EXT 016,"A",C$,20,100,"" : GOSUB INPUTMOUSE

GOTO VIS

INPUTMOUSE:
  VMSG 1, "hai cliccato su ZONAMOUSE: " + STR$( _TMZ,0,0)
  IF _TMZ=011 THEN GOTO VIS01 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=012 THEN GOTO VIS02 '(resta sul campo)
  IF _TMZ=013 THEN GOTO VIS03 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=014 THEN GOTO VIS04 '(resta sul campo)
  IF _TMZ=015 THEN GOTO VIS05 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=016 THEN GOTO VIS06 '(resta sul campo)

RETURN

```

Notare: la variabile `_TMZ` è globale per tutte le finestre aperte in Sprix

Esempio: INPUT_EXT di tipo File

```

INPUT_EXT 0,"F",MIAVAR$,30,128,"" input di tipo file con tutte le opzioni
INPUT_EXT 0,"F",MIAVAR$,30,128,"NOCANC" input di tipo file spegne la cancellazione dei file
INPUT_EXT 0,"F",MIAVAR$,30,128,"NOESEC" input di tipo file spegne l'esecuzione dei file
INPUT_EXT 0,"F",MIAVAR$,30,128,"NOESEC;NOCANC" input di tipo file spegne l'esecuzione e la cancellazione dei file
Apertura automatica di un file:
NOME_FILE$ = "<C:\Appoggio\Notule_farmacie.pdf"
PUTTAST "#605#501#501"
INPUT_EXT 0,"F",NOME_FILE$,1,128,""

```

Esempio: INPUT_EXT di tipo Directory

```

INPUT_EXT 0,"P",MIAVAR$,30,128,"" input di tipo directory con tutte le opzioni
INPUT_EXT 0,"P",MIAVAR$,30,128,"CREADIR" input di tipo directory con abilitazione di creazione directory

```

Esempio di verifica esistenza e creazione automatica directory

```

CREADIR:
  DIR$ = "!\\dirapp\" '(directory da ricercare in ..\dati\datigen\datiutente\)
  LEGGDIR DIR$, ""
  IF_ERRDIR$ = "" THEN RETURN 'Se non torna errore la directory esiste
  DISVID 1 'Speno il video
  PUTTAST "#501#610" 'Imposto i tasti da passare
  LOCATE 5,40: INPUT_EXT 0,"P",DIR$,5,138,"CREADIR" 'Creo la directory
  DISVID 0 'Riaccendo il video
RETURN

```

Esempio: utilizzo del campo di tipo password in INPUT, INPUT_EXT, INPUT_EXT_EXT, INPUT3

```

_INTER=1
APRIFIN 1,1,0,0,0,15,50 : SETFSM 1
VIS:
VIS01:
  SETZM 1,10,25,011
  COLOR 3: LOCATE 1,11 : PRINT "RIGA-TMZ=11" : COLOR 0
  COLOR 3: LOCATE 1,36 : PRINT "RIGA-TMZ=0" : COLOR 0
VIS02:
  LOCATE 2,1: INPUT "W",A$,20 : GOSUB INPUTMOUSE
VIS03:
  SETZM 3,10,25,013
  COLOR 3: LOCATE 3,11 : PRINT "RIGA-TMZ=13" : COLOR 0
  COLOR 3: LOCATE 3,36 : PRINT "RIGA-TMZ=0" : COLOR 0
VIS04:
  LOCATE 4,1: INPUT "WP",B$,20 : GOSUB INPUTMOUSE
VIS05:
  SETZM 5,10,25,015
  COLOR 3: LOCATE 5,11 : PRINT "RIGA-TMZ=15" : COLOR 0
  COLOR 3: LOCATE 5,36 : PRINT "RIGA-TMZ=0" : COLOR 0
VIS06:
  LOCATE 6,1: INPUT_EXT 016,"W",C$,20,100,"CONFERMA" : GOSUB INPUTMOUSE
VIS07:
  LOCATE 7,1: INPUT_EXT 017,"WP",D$,20,100,"" : GOSUB INPUTMOUSE
VIS08:
  LOCATE 8,1: INPUT_EXT_EXT 018,"W",E$,20,100,"" : GOSUB INPUTMOUSE
VIS09:
  LOCATE 9,1: INPUT_EXT_EXT 019,"WP",F$,20,100,"CONFERMA" : GOSUB INPUTMOUSE
VIS10:
  LOCATE 10,1: INPUT3 020,"W",G$,20,100,"",ERR$ : GOSUB INPUTMOUSE
VIS11:
  LOCATE 11,1: INPUT3 021,"WP",H$,20,100,"CONFERMA",ERR$ : GOSUB INPUTMOUSE
VIS12:
  DIM ST$(2) : ST$(1)="primaTxt" : ST$(2)="secondaTxt"
  LOCATE 12,1: INPUT_TXT 022,ST$(2),1,10,1,10,"" : GOSUB INPUTMOUSE

```



```
GOTO VIS

INPUTMOUSE:
  VIMSG 1, "hai cliccato su ZONAMOUSE: " + STR$( _TMZ,0,0)
  IF _TMZ=011 THEN GOTO VIS01 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=012 THEN GOTO VIS02 '(resta sul campo)
  IF _TMZ=013 THEN GOTO VIS03 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=014 THEN GOTO VIS04 '(resta sul campo)
  IF _TMZ=015 THEN GOTO VIS05 '(resta sul campo, ovvero la successiva Input)
  IF _TMZ=016 THEN GOTO VIS06 '(resta sul campo)
  IF _TMZ=017 THEN GOTO VIS07 '(resta sul campo)
  IF _TMZ=018 THEN GOTO VIS08 '(resta sul campo)
  IF _TMZ=019 THEN GOTO VIS09 '(resta sul campo)
  IF _TMZ=020 THEN GOTO VIS10 '(resta sul campo)
  IF _TMZ=021 THEN GOTO VIS11 '(resta sul campo)
  IF _TMZ=022 THEN GOTO VIS12 '(resta sul campo)

RETURN
```

Esempio _DES612\$: due Collage, entrambi sul modulo (MA) di anagrafica articoli di magazzino:

<pre>ON_IN_MADSART: _DES612\$="primo F12" : END ON_612_MACODAL: VIMSG 4, "routine 1": END</pre>	<pre>ON_IN_MACODAL: _DES612\$="secondo F12" : END ON_612_MACODAL: VIMSG 4, "routine 2": END</pre>
---	---

Esempio: numero di finestra attiva _NFIN (per Collage)

```
ON_612_ALL: VIMSG 1, STR$( _NFIN,0,0) : END
```

Esempio Collage: come personalizzare le etichette dei campi nelle videate

L'istruzione PRINT consente di scrivere un testo su una videata personalizzata (combinata con le istruzioni APRIFIN, ecc...), ma anche per personalizzare le videate della procedura standard. Il seguente Collage personalizza l'etichetta "Nome ricerca" solo se le prime tre cifre del codice conto sono "501".

```
ON_IN ALL:
IF _NCAMS="CARICCOD" THEN END
AZZVARSYS 1
GETATTPC
IF LEFT$( _PCCOD$,3)="501" THEN GOSUB SCRIVI
IF LEFT$( _PCCOD$,3)>"501" THEN GOSUB CANCELLA
END

CANCELLA:
COLOR 0
CLS
RETURN

SCRIVI:
ATTIVAFIN -1      '(se la videata su cui scrivere non fosse attiva, occorre attivarla con istruzione ATTIVAFIN)

LOCATE 5,24      '(riferimenti di riga e colonna)
COLOR 4          '(colore nuova etichetta)
PRINT "MIA LABEL" '(consente di scrivere un testo sulla videata attiva del gestionale; è utilizzabile anche nella
                  '(costruzione di una nuova videata personalizzata creata con APRIFIN)

RETURN
```

Caso: PRINT genera Errore Interno Vis(): la stringa eccede i limiti della finestra

Il messaggio è dovuto al fatto che l'istruzione PRINT ha superato le dimensioni della videata in cui scrivere la stringa. L'esempio che segue provoca tale errore essendo la larghezza della videata pari a 10 caratteri contro i 21 della stringa da scriversi con l'istruzione PRINT.

```
_INTER=0
COLORE=2
N_FIN=AUTOFIN()
APRIFIN N_FIN,COLORE,0,3,10,1,10
ATTIVAFIN N_FIN
PRINT "VERIFICA ERRORE VIS()"

DIGITA:
TASTO=INKEY()
IF TASTO=505 THEN GOTO FINE
GOTO DIGITA

FINE:
CHIUDIFIN N_FIN
END
```

Caso: PRINT provoca rallentamento

In caso di rallentamenti di refresh video di buffer corposi di PRINT occorre far precedere la funzione PRINT da una DISVID 1 e successivamente da una DISVID 0

```
DISVID 1
PRINT "CICLO_A"
FOR I=1 TO 200
PRINT "AAA";
NEXT
PRINT ""
DISVID 0
INPUT "A",A$,1

DISVID 1
PRINT "CICLO_B"
FOR I=1 TO 300
PRINT "BBB";
NEXT
PRINT ""
```

```
DISVID 0
INPUT "A",B$,1

PRINT "CICLO_C"
PRINT STRING$(125,"C");
PRINT STRING$(125,"C");
PRINT STRING$(125,"C");
PRINT STRING$(125,"C")
INPUT "A",C$,1
END
```

PRINT/INPUT: scelta del font

Non sono disponibili istruzioni per personalizzare il font a video; la scelta del font è consentito in stampa con l'istruzione LPRINTTXT". Una alternativa per differenziare il font a video può consistere, nel caso dell'istruzione INPUT, nell'utilizzare il font dell'istruzione PRINT (le due istruzioni utilizzano due font differenti). Di seguito viene mostrato un semplice esempio:

```
_INTER=0
N_FIN=AUTOFIN()
COLORE=2
APRIFIN N_FIN,COLORE,0,5,10,4,40
SETTF "505Esc"
NOME:
LOCATE 1,1
COLOR 0
PRINT "NOME.....";
INPUT "AM",NOME$,30
IF _TF=505 THEN GOTO FINE
COGNOME:
LOCATE 2,1
PRINT "COGNOME..";
INPUT "AM",COGNOME$,30
IF _TF=505 THEN GOTO FINE
LOCATE 2,11
COLOR 4
COGNOME$=LEFT$(COGNOME$+STRING$(30," "),30)
PRINT COGNOME$
GOTO NOME
FINE:
CHIUDIFIN 1
END
```

Casi:

Copertura parziale monitor: all'avvio del programma creare finestre per coprire alcune zone dello schermo e gestire il posizionamento del cursore all'uscita dal campo (ON_OUT_?) quando si preme INVIO o F10.

Pulizia campi sporchi: non c'è istruzione specifica, si puliscono chiudendo e riaprendo la finestra.

Informazioni a video: i dati ed i valori non ancora registrati sono disponibili nelle variabili di struttura archivi attraverso l'istruzione GETATT?? eseguibile su un evento attivo ed intercettabile.

Chiusura finestra resta in sospensione: verificare la corretta posizione e impostazione di una eventuale _INTER

Evento per ATTIVAFIN: In presenza di ATTIVAFIN e di successive istruzioni di modifica a video (LOCATE, PRINTD, ecc...), può capitare di utilizzare eventi generali non idonei (es ON_START?), poiché troppo a ridosso del momento in cui il programma disegna la maschera da modificare ed attivata con ATTIVAFIN, ottenendo per contro, modifiche solo parziali o addirittura sottostanti la maschera stessa. Occorre pertanto utilizzare eventi successivi al disegno della maschera da modificare (una ON_IN_ oppure una ON_OUT_ di un campo).

Errore "impossibile chiudere la finestra": la CHIUDIFIN non rispetta la gerarchia delle finestre; provare a stabilire una gestione delle finestre all'uscita del campo e non all'entrata; eventualmente fra un APRIFIN ed un'eventuale ATTIVAFIN interporre una PRINT.

Selezione a video della stampante non intercettabile: vedere esempi e casi nella sezione delle STAMPE

INCAR() per libreria gestione messaggi:

```
MESSAGGIO:
FIN_MESS=AUTOFIN() : TF=_TF : _TF=0
IF DES_MESS$="" THEN DES_MESS$="Confermi Operazione"
IF LEFT$(CVMAIU$(RISP_MESS$),1)<>"S" THEN CONF_MESS$="No" ELSE CONF_MESS$="Si"
COL_MESS=LEN(DES_MESS$)+2 : IF COL_MESS<21 THEN COL_MESS=21
APRIFIN_EXT FIN_MESS, NFIN,0,0,3,COL_MESS,"Attenzione","A610Si,A505No",""
LOCATE 2,2 : PRINT DES_MESS$;
MESSAGGIO1:
SETTF "501"+CONF_MESS$+",610,505"
TINCAR=INCAR("")
IF CVMAIU$(CHR$(TINCAR))="N" THEN CONF_MESS$="No" : GOTO MESSAGGIO1
IF CVMAIU$(CHR$(TINCAR))="S" THEN CONF_MESS$="Si" : GOTO MESSAGGIO1
IF INSTR("501,610,505",RIGHT$("000"+STR$(TINCAR,0,0),3))=0 THEN GOTO MESSAGGIO1
CHIUDIFIN FIN_MESS
_TF=TF
IF INSTR("505",RIGHT$("000"+STR$(TINCAR,0,0),3))>0 THEN RISP_MESS$="N" : RETURN
IF INSTR("610",RIGHT$("000"+STR$(TINCAR,0,0),3))>0 THEN RISP_MESS$="S" : RETURN
RISP_MESS$=LEFT$(CONF_MESS$,1)
RETURN
```

Utilizzo della funzione:
RISP_MESS\$="N" ' parametro di impostazione iniziale
DES_MESS\$="Confermi Creazione" 'messaggio da visualizzare
GOSUB MESSAGGIO
IF RISP_MESS\$="N" THEN ... ELSE ...

Esempio Sprix CREATELIST:

```
'Preparazione Lista
AZZVARSYS 51
FOR A = 1 TO 10
  _CTVALCOL1$(A) = "ValoreFisso" ' Valori della colonna 1
  _CTVALCOL2$(A) = "Colonna 2 Riga: " + STR$(A,0,0) ' Valori della colonna 2
  _CTVALCOL3$(A) = STRDATA$(sxdat,3,"/") ' Valori della colonna 3
  _CTVALCOL4$(A) = STR$(A,0,0) ' Valori della colonna 4
NEXT
_CTVTOTRIG = 10 ' Numero di righe totali da mettere a video
_CTVTOTCOL = 4 ' Numero di colonne
' _CTTYPECOL$(1) = "I" ' valore di tipo icona (vedere SETTF)
_CTYPECOL$(1) = "S" ' Tipo di colonna accetta i valori I=icona, S=Stringa, N=numero, D=Data
```

```

_CTYPECOL$ (2) = "S" ' Tipo di colonna accetta i valori I=icona, S=Stringa, N=numero, D=Data
_CTYPECOL$ (3) = "D" ' Tipo di colonna accetta i valori I=icona, S=Stringa, N=numero, D=Data
_CTYPECOL$ (4) = "N" ' Tipo di colonna accetta i valori I=icona, S=Stringa, N=numero, D=Data
_CTTITCOL$ (1) = "Colonna 1" ' Intestazione colonna 1
_CTTITCOL$ (2) = "Colonna 2" ' Intestazione colonna 2
_CTTITCOL$ (3) = "Colonna 3" ' Intestazione colonna 3
_CTTITCOL$ (4) = "Colonna 4" ' Intestazione colonna 4
'_CTVALCOLI$ (1) = "PALLINO_VERDE" (descrizione icona, vedere SETTF)
_CTLENCOL (1) = 14 ' Lunghezza della colonna da considerare per la ricerca del dato
_CTVLENCOL (1) = 14 ' Lunghezza della colonna da vedere a video.
_CTLENCOL (2) = 20 ' Lunghezza della colonna da considerare per la ricerca del dato
_CTVLENCOL (2) = 20 ' Lunghezza della colonna da vedere a video.
_CTLENCOL (3) = 10 ' Lunghezza della colonna da considerare per la ricerca del dato
_CTVLENCOL (3) = 10 ' Lunghezza della colonna da vedere a video.
_CTLENCOL (4) = 10 ' Lunghezza della colonna da considerare per la ricerca del dato
_CTVLENCOL (4) = 10 ' Lunghezza della colonna da vedere a video.

```

```

CREATELIST "Lista Esempio", 2, 8, 0
IF _ERRCT <> 0 THEN VIMSG 4, STR$( _ERRCT,0,0) + " " + _ERRCT$

'Valori di ritorno:
'_CTTOTRIGSEL ' Quantita totali degli indici selezionati
' FOR I = 1 TO _CTTOTRIGSEL
'     ' _CTINDRIGSEL(I) ' Numero dell'indice selezionato
' NEXT

```

CREATELIST esegue la funzione nativa di lista standard a scelta multipla del Gestionale, pertanto non mantiene la selezione effettuata nel caso filtri successivamente alla stessa.

CREATELIST non consente l'inserimento di tasti funzione e/o filtri, tali proprietà sono riservate al Gestionale in quanto non si tratta di una finestra di elaborazione, ma solo di visualizzazione dei dati

L'istruzione **CREATELIST** è una macro alquanto atomica, non consente ulteriori interventi (perchè apre una finestra fuori dal controllo diretto di sprix/collage ai quali però restituisce valori selezionati). Solo alcune possibilità di intervento sono consentite, come ad esempio la riapertura in seguito ad invii da tastiera, che, nonostante siano eseguiti lato Server, qualora fossero molto numerosi, richiederebbero tempi maggiori per la ulteriore visualizzazione della **CREATELIST**.

Nella seguente routine, in seguito alla selezione di un record e la relativa conferma, il programma riapre la **CREATELIST** e riposiziona il cursore nel record precedentemente selezionato. Più il numero di record è elevato, maggiori sono i tempi di invio da tastiera e visualizzazione finale.

```

CREATELIST "Lista Esempio", 2, 8, 0
IF _ERRCT <> 0 THEN VIMSG 4, STR$( _ERRCT,0,0) + " " + _ERRCT$
IND_RIGA=CTINDRIGSEL( _CTTOTRIGSEL)
VIMSG 1, STR$(IND_RIGA,0,0) ' Numero dell'indice selezionato
FOR I= 1 TO IND_RIGA-1
    PUTTAST "#513"
NEXT
'_CTFILTEXT$=_CTVALCOLI$(IND_RIGA) ' per mantenere il filtro impostato nel campo di ricerca
CREATELIST "Lista Esempio", 2, 8, 0 ' Riapertura con posizionamento del cursore nel punto di selezione

```

Essendo una macro che trae origine dalla funzione finestra elenco (come Emissione revisione documenti) non è consentito il riposizionamento a video nel medesimo record selezionato. Al limite, piuttosto che avere il record selezionato in basso nella finestra della **CREATELIST**, matematicamente lo si può posizionare nella prima riga.

Esempio

```

CREATELIST "Lista Esempio", 2, 8, 0
IF _ERRCT <> 0 THEN VIMSG 4, STR$( _ERRCT,0,0) + " " + _ERRCT$
IND_RIGA=CTINDRIGSEL( _CTTOTRIGSEL)
VIMSG 1, STR$(IND_RIGA,0,0) ' Numero dell'indice selezionato
FOR I= 1 TO IND_RIGA-1
    PUTTAST "#513"
NEXT
'_CTPOSCURFI=1
FOR I= 1 TO QUANTITA_RIGHE-IND_RIGA
    PUTTAST "#512"
NEXT
CREATELIST "Lista Esempio", 2, 8, 0

```

In caso di presenza di filtri, la variabile non **_CTPOSCURFI** (posizionamento a fine elenco) potrebbe non funzionare.

La possibilità di sapere a quale colonna appartiene il filtro inserito dall'utente (poichè la **_CTFILTEXT\$** è solo un valore passato al campo di ricerca in cui non c'è controllo) è consentito comunque dall'indice di riga con il quale comporre la stringa da analizzare per capire a quale colonna appartiene l'elemento della stringa trovato dall'utente.

```

FILTRO$=_CTVALCOLI$(IND_RIGA)+_CTVALCOL2$(IND_RIGA)
IF INSTR(FILTRO$,"Colonna")<>0 THEN VIMSG 1, "nel filtro hai messo anche la colonna 2"

```

'Gestione di tasti funzione personalizzati nella CRATELIST

```

'INTER=1
'AZZARRAY VALN() : AZZARRAY VALO$()

'//////////
' CREATELIST SPRIX //
'//////////

DIM VALN() : DIM VALO$()

VALN( 1)=29 : VALO$( 1) = "ZZZ" : VALN( 2)=28 : VALO$( 2) = "YYY" : VALN( 3)=27 : VALO$( 3) = "XXX" : VALN( 4)=26 : VALO$( 4) = "WWW"
VALN( 5)=25 : VALO$( 5) = "VVV" : VALN( 6)=24 : VALO$( 6) = "UUU" : VALN( 7)=23 : VALO$( 7) = "TTT" : VALN( 8)=22 : VALO$( 8) = "SSS"
VALN( 9)=21 : VALO$( 9) = "RRR" : VALN(10)=20 : VALO$(10) = "QQQ" : VALN(11)=36 : VALO$(11) = "PPP" : VALN(12)=35 : VALO$(12) = "OOO"
VALN(13)=34 : VALO$(13) = "NNN" : VALN(14)=33 : VALO$(14) = "MMM" : VALN(15)=32 : VALO$(15) = "LLL" : VALN(16)=31 : VALO$(16) = "KKK"
VALN(17)=30 : VALO$(17) = "JJJ" : VALN(18)=09 : VALO$(18) = "III" : VALN(19)=08 : VALO$(19) = "HHH" : VALN(20)=07 : VALO$(20) = "GGG"
VALN(21)=06 : VALO$(21) = "FFF" : VALN(22)=05 : VALO$(22) = "EEE" : VALN(23)=04 : VALO$(23) = "DDD" : VALN(24)=03 : VALO$(24) = "CCC"
VALN(25)=02 : VALO$(25) = "BBB" : VALN(26)=01 : VALO$(26) = "AAA" : VALN(27)=10 : VALO$(27) = "999" : VALN(28)=19 : VALO$(28) = "888"
VALN(29)=18 : VALO$(29) = "777" : VALN(30)=17 : VALO$(30) = "666" : VALN(31)=16 : VALO$(31) = "555" : VALN(32)=15 : VALO$(32) = "444"
VALN(33)=14 : VALO$(33) = "333" : VALN(34)=13 : VALO$(34) = "222" : VALN(35)=12 : VALO$(35) = "111" : VALN(36)=11 : VALO$(36) = "000"

```

NRVAL=36 ' numero righe gestite 36 (su totale di 26 righe a video)

AZZCATVARSYS "CT": ' AZZVARSYS 51

_CTVTOTCOL=20 ' numero di colonne a video

'_CTYPECOL\$(1-20) = Tipo di colonna (I=icona, S=Stringa, N=Numero, D=Data)
' CTTITCOL\$(1-20) = Intestazione del numero di colonna

```

                '_CTLENCOL(1-20) = Lunghezza della colonna da considerare per la ricerca del dato
                '_CTVWLENCOL(1-20) = Lunghezza della colonna da vedere a video

_CTYPECOL$(1)="I": _CTTITCOL$(1)="Ico"      : _CTLENCOL(1)=3 : _CTVWLENCOL(1)=3
_CTYPECOL$(2)="N": _CTTITCOL$(2)="Numero"  : _CTLENCOL(2)=6 : _CTVWLENCOL(2)=6
_CTYPECOL$(3)="S": _CTTITCOL$(3)="String"   : _CTLENCOL(3)=6 : _CTVWLENCOL(3)=6
_CTYPECOL$(4)="D": _CTTITCOL$(4)="Data"     : _CTLENCOL(4)=11: _CTVWLENCOL(4)=11

FOR C=5 TO 20
_CTYPECOL$(C)="S": _CTTITCOL$(C)="CoL. " + STR$(C,0,0) : _CTLENCOL(C)=10 : _CTVWLENCOL(C)=10
NEXT

_CVTOTRIG=0 ' Impostazione a Tutto Schermo

NU=0

FOR I=1 TO NRVAL ' Numero di righe totali da mettere a video (36)
  NU=NU+1
  S$="Str$ "+STR$(I,0,0)
  _CTVTOTRIG=I
  _CTVALCOL1$(I) = "PALLINO_VERDE"
  _CTVALCOL2$(I) = STR$(VALN(I),0,0)
  _CTVALCOL3$(I) = VALO$(I)
  _CTVALCOL4$(I) = STRDATA$(sxdat,3,"/")
  _CTVALCOL5$(I) = S$
  _CTVALCOL6$(I) = "C.6 R."+STR$(I,0,0)
  _CTVALCOL7$(I) = "C.7 R."+STR$(I,0,0)
  _CTVALCOL8$(I) = "C.8 R."+STR$(I,0,0)
  _CTVALCOL9$(I) = "C.9 R."+STR$(I,0,0)
  _CTVALCOL10$(I) = "C.10 R."+STR$(I,0,0)
  _CTVALCOL11$(I) = "C.11 R."+STR$(I,0,0)
  _CTVALCOL12$(I) = "C.12 R."+STR$(I,0,0)
  _CTVALCOL13$(I) = "C.13 R."+STR$(I,0,0)
  _CTVALCOL14$(I) = "C.14 R."+STR$(I,0,0)
  _CTVALCOL15$(I) = "C.15 R."+STR$(I,0,0)
  _CTVALCOL16$(I) = "C.16 R."+STR$(I,0,0)
  _CTVALCOL17$(I) = "C.17 R."+STR$(I,0,0)
  _CTVALCOL18$(I) = "C.18 R."+STR$(I,0,0)
  _CTVALCOL19$(I) = "C.19 R."+STR$(I,0,0)
  _CTVALCOL20$(I) = "C.20 R."+STR$(I,0,0)
NEXT
NT=0

'//////////
'// Identificativo con cui verranno cercate le etichette associate //
'// ai tasti relative a QUESTA CREATELIST //
'// Es:1'etichetta per la gestione del tasto F7 in QUESTA CREATELIST //
'// 'dovra' obbligatoriamente chiamarsi ON 607 ETIC: //
'//////////
_CTEVETIC$ = "ETIC"

'//////////
'// Abilita i Tasti funzione che si vogliono utilizzare //
'//////////

_CTTASTFUNES$ = "501,505" ' attivazione tasti 501 e 505 per CREATELIST di tipo zero (0)

NT=NT+1: _CTTASTFUN$(NT) = "A505Esc-|Esc|Uscita"
NT=NT+1: _CTTASTFUN$(NT) = "A501Invio-|Invio|Seleziona"
NT=NT+1: _CTTASTFUN$(NT) = "A608Prima-|Prima Posizione|Sposta in testa all'elenco"
NT=NT+1: _CTTASTFUN$(NT) = "A606Visualizza-|Visualizza|Mostra"
NT=NT+1: _CTTASTFUN$(NT) = "A611Modifica-|Modifica Record|Modifica riga corrente"
NT=NT+1: _CTTASTFUN$(NT) = "A620Cancella-|Cancella Record|Cancella riga corrente"
NT=NT+1: _CTTASTFUN$(NT) = "A604Nuovo-|Nuovo Record|Inserisce nuove righe"
NT=NT+1: _CTTASTFUN$(NT) = "A607Duplica-|Duplica Selezionati|Duplica righe scelte" 'valido per multiselezioni
NT=NT+1: _CTTASTFUN$(NT) = "A614Cancella-|Elimina Selezionati|Cancella righe scelte" 'valido per multiselezioni
NT=NT+1: _CTTASTFUN$(NT) = ""

'//////////
'// Creazione della finestra elenco //
'//////////

'NOTA BENE: la scelta del tipo di CREATELIST condiziona il funzionamento delle azioni disponibili
'-----

' CREATELIST "Lista di Esempio",0,0,0 'selezione semplice (per attivare i tasti 501 e 505 utilizzare _CTTASTFUNES)
CREATELIST "Lista di Esempio",1,0,0 'multi selezione con tasto F9 e accettazione dei valori con tasto INVIO
' CREATELIST "Lista di Esempio",2,0,0 'multi selezione con tasto INVIO accettazione dei valori con tasto F10

'//////////
'/// Se errore Stampa/Gestisci errore //
'//////////

IF _ERRCT THEN VIMSG 5, STR$( _ERRCT,0,0)+"+_ERRCT$
'GOSUB CHECK_VAL
END

'//////////
'// Tasto 505 Esc //
'//////////
ON 505_ETIC:
  VIMSG 1, "ON 505_ETIC: hai selezionato Esc"
  GOSUB CHECK_VAL
  _CTAZIONE$ = "SCF_RIDISEGNA_CHIUDI"
  _CTAZIONE$ = "SCF_OPERAZIONE_ANNULLATA"

END

'//////////
'// Tasto 501 Invio //
'//////////
ON 501_ETIC:
  VIMSG 1, "ON 501_ETIC: hai selezionato Invio"
  GOSUB CHECK_VAL
  _CTAZIONE$ = "SCF_RIDISEGNA_CHIUDI"

END

'//////////
'// Tasto 608 Sposta in Prima posizione //
'//////////

```

```

'////////////////////////////////////
ON 608_ETIC:
VIMSG 1, "ON 608_ETIC: hai selezionato Sposta la riga selezionata in prima posizione"
  GOSUB CHECK_VAL
  _CTAZIONE$ = "SCF_RIDISEGNA_PPRIMA_POS"
END

'////////////////////////////////////
'// Tasto 606 Visualizza //
'////////////////////////////////////
ON 606_ETIC:
VIMSG 1, "ON 606_ETIC: hai selezionato visualizza"
  GOSUB CHECK_VAL
  _CTAZIONE$ = "SCF_RIDISEGNA_CHIUDI"
END

'////////////////////////////////////
'// Tasto 611 Modifica i dati della riga corrente //
'////////////////////////////////////
ON 611_ETIC:
VIMSG 1, "ON 611_ETIC: hai selezionato Modifica riga corrente"
  GOSUB CHECK_VAL
  _CTAZIONE$ = "SCF_RIDISEGNA_VER_MODIFICA"
  RIGAATT = _CTCURRRIGA ' Indice riga corrente
  GOSUB EDIT_ROW ' chiamo la funzione di editing di una riga
  ' se la funzione di editing mi segnala che stato chiesto il rollback annullo la modifica
  IF DOROLLBACK=1 THEN _CTAZIONE$ = "SCF_OPERAZIONE_ANNULLATA" ' modifica annullata
  ERR=RIPTAFCT(" ")
END

'////////////////////////////////////
'//// Tasto 620 Cancella la riga corrente //
'////////////////////////////////////
ON 620_ETIC:
VIMSG 1, "ON 620_ETIC: hai selezionato Cancella riga corrente"
  GOSUB CHECK_VAL
  _CTAZIONE$ = "SCF_RIDISEGNA_CANCEL_POS"
  ' Se a fronte di Controlli Volessi annullare la cancellazione dela riga devo cambiare il valore di _CTAZIONE$
  IFF 1 < 0
  _CTAZIONE$ = "SCF_OPERAZIONE_ANNULLATA"
  ENDF
END

'////////////////////////////////////
'// Tasto 604 Inserisce nuove righe //
'////////////////////////////////////
ON 604_ETIC:
VIMSG 1, "ON 604_ETIC: hai selezionato Inserimento di nuove righe"
  GOSUB CHECK_VAL
  VIMSG 1, "Inserisco delle nuove righe in fondo alla lista"

  _CTAZIONE$ = "SCF_RIDISEGNA_INSER_POS"
  NU=_CTNUMRIGHE

  NU=NU+1: _CTVALCOL1$(NU)="PALLINO_VERDE": _CTVALCOL2$(NU)=STR$(NU,0,0): _CTVALCOL3$(NU)="New "+STR$(NU,0,0): _CTVALCOL4$(NU)=STRDATA$(sxdat,3,"/")
  NU=NU+1: _CTVALCOL1$(NU)="PALLINO_VERDE": _CTVALCOL2$(NU)=STR$(NU,0,0): _CTVALCOL3$(NU)="New "+STR$(NU,0,0): _CTVALCOL4$(NU)=STRDATA$(sxdat,3,"/")
  NU=NU+1: _CTVALCOL1$(NU)="PALLINO_VERDE": _CTVALCOL2$(NU)=STR$(NU,0,0): _CTVALCOL3$(NU)="New "+STR$(NU,0,0): _CTVALCOL4$(NU)=STRDATA$(sxdat,3,"/")
  NU=NU+1: _CTVALCOL1$(NU)="PALLINO_VERDE": _CTVALCOL2$(NU)=STR$(NU,0,0): _CTVALCOL3$(NU)="New "+STR$(NU,0,0): _CTVALCOL4$(NU)=STRDATA$(sxdat,3,"/")
  NU=NU+1: _CTVALCOL1$(NU)="PALLINO_VERDE": _CTVALCOL2$(NU)=STR$(NU,0,0): _CTVALCOL3$(NU)="New "+STR$(NU,0,0): _CTVALCOL4$(NU)=STRDATA$(sxdat,3,"/")

  _CTINSRIGHE = NU- _CTNUMRIGHE
  VIMSG 1, "Hai inserito "+STR$( _CTINSRIGHE,0,0)+" in fondo alla lista, con i primi 4 campi valorizzati"

END

'////////////////////////////////////
'// Tasto 607 Duplica le righe selezionate //
'////////////////////////////////////
ON 607_ETIC:
VIMSG 1, "ON 607_ETIC: hai selezionato Duplica le righe selezionate"
  GOSUB CHECK_VAL
  IFF _CTTOTRIGSEL > 0
  NRRIGHEINS=0
  _CTAZIONE$ = "SCF_RIDISEGNA_INSER_POS"
  NU=_CTNUMRIGHE ' Numero Totale delle righe elencate
  FOR I=1 TO _CTTOTRIGSEL ' Per ogni riga selezionata
  IFF 1 > 0 ' Se i dati soddisfano la condizione

  'Aggiungo i dati della riga "duplicata" da quella selezionata dopo l'ultima riga gia' presente
  NU=NU+1:
  _CTVALCOL1$(NU) = _CTVALCOL1$( _CTINDRIGSEL(I)): _CTVALCOL2$(NU) = _CTVALCOL2$( _CTINDRIGSEL(I)) 'DUP'
  _CTVALCOL3$(NU) = _CTVALCOL3$( _CTINDRIGSEL(I)) ' DUP

  NRRIGHEINS=NRRIGHEINS+1
  ENDF
  NEXT
  _CTINSRIGHE = NRRIGHEINS ' Comunica il numero delle righe inserite
  ELSEF ' Altrimenti nessuna riga selezionata per cui operazione annullata
  _CTAZIONE$ = "SCF_OPERAZIONE_ANNULLATA"
  VIMSG 4, "ETIC_607 (NESSUNA RIGA SELEZIONATA)"
  ENDF
END

'////////////////////////////////////
'//// tasto 614 Cancella le righe selezionate //
'////////////////////////////////////
ON 614_ETIC:
VIMSG 1, "ON 614_ETIC: hai selezionato Cancella le righe selezionate"
  GOSUB CHECK_VAL
  _CTAZIONE$ = "SCF_RIDISEGNA_CANCEL_SELEZ"
  IFF 1 > 0 ' Se <condizione> vera proseguo con la cancellazione
  FOR I=1 TO _CTTOTRIGSEL ' per ogni riga selezionata
  'IFF VAL( _CTVALCOL1$( _CTINDRIGSEL(I) )) = 10 'verifico se riga da cancellare,
  'qui non si vuole cancellare la riga in cui il campo "10"
  ' _CTINDRIGSEL(I) = 0 ' se non da cancellare ne annullo la cancellazione
  ' ENDF
  NEXT
  ELSEF ' Se <condizione> non e' vera annullo la cancellazione delle righe selezionate
  _CTAZIONE$ = "SCF_OPERAZIONE_ANNULLATA"

```

```

ENDF
END
'#####
'////////////////////////////////////
'//// CONTROLLO VALORI DI RITORNO ///
'////////////////////////////////////
CHECK_VAL:
NUMRIGHE$ = "Quante righe in elenco = " + STR$(CTNUMRIGHE,0,0) + "|"
INSRIGHE$ = "Quante righe aggiunte dopo quelle in elenco = " + STR$(CTINSRIGHE,0,0) + "|"
RIGA_COR$ = "Numero indice riga corrente = " + STR$(CTCURRRIGA,0,0) + "|"

'Per selezioni e operazioni multiriga
DIM IND_RIGA$( )
QTA_RIGA$ = "Quante righe selezionate e confermabili = " + STR$(CTTOTRIGSEL,0,0) + "|"
FOR I=1 TO CTTOTRIGSEL
IND_RIGA$(I) = STR$(CTINDRIGSEL(I),0,0)+" " : IND_RIGATOT$ = IND_RIGATOT$ + IND_RIGA$(I)
NEXT

VIMSG 1, NUMRIGHE$ + INSRIGHE$ + RIGA_COR$ + QTA_RIGA$ + "Numeri indice multiriga selezionata = |" + IND_RIGATOT$
NUMRIGHE = 0 : NUMRIGHE$ = "" : INSRIGHE = 0 : INSRIGHE$ = "" : RIGA_COR = 0 : RIGA_COR$ = "" : IND_RIGATOT$ = "" : AZZARRAY IND_RIGA$( )

'..... FILTRO$=_CTVALCOL1$(IND_RIGA)+CTVALCOL2$(IND_RIGA)
'..... IF INSTR(FILTRO$,"abcd")<>0 THEN VIMSG 1, "hai messo anche ...."

'..... FOR I=1 TO IND_RIGA-1
'..... PUTTAST "#513"
'..... NEXT

'..... _CTFILTEXT$=_CTVALCOL1$(IND_RIGA) ' per mantenere il filtro impostato nel campo ricerca
'..... CREATELIST "Lista di Esempio",2,0,0 ' riapertura con posizionamento cursore nel punto di selezione

'..... _CTPOSCURFI=1
'..... FOR I=1 TO 36-IND_RIGA 'quantita' in base IND_RIGA
'..... PUTTAST "#512"
'..... NEXT
RETURN

'////////////////////////////////////
'// FUNZIONE CHE CONSENTE L'EDITING DEI DATI DI UNA RIGA ( +/- ) //
'////////////////////////////////////

'////////////////////////////////////

EDIT ROW:
MAXLENTIT=0
MAXLENROW=0
FILLER$= ".....: "
FOR I=1 TO _CTVTOTCOL
IF LEN( _CTTITCOL$(I)) > MAXLENTIT THEN MAXLENTIT = LEN( _CTTITCOL$(I)) + 3
IF _CTLENCOL(I) > MAXLENROW THEN MAXLENROW = _CTLENCOL(I)
NEXT

APRIFIN_TIT 1,1,0, 3, 5, 4+_CTVTOTCOL, 4+MAXLENTIT + MAXLENROW, "Modifica Riga", ""

DOROLLBACK=0
_SCORRI=1
_INTER=0
DONE = 0
NXTFLD = 0

' OLDVAL1$="PALLINO VERDE" ' forzo la non modifica dell'icona
OLDVAL1$=_CTVALCOL1$(RIGAATT)
OLDVAL2$=_CTVALCOL2$(RIGAATT)
OLDVAL3$=_CTVALCOL3$(RIGAATT)
OLDVAL4$=_CTVALCOL4$(RIGAATT)
OLDVAL5$=_CTVALCOL5$(RIGAATT)
OLDVAL6$=_CTVALCOL6$(RIGAATT)
OLDVAL7$=_CTVALCOL7$(RIGAATT)
OLDVAL8$=_CTVALCOL8$(RIGAATT)
OLDVAL9$=_CTVALCOL9$(RIGAATT)
OLDVAL10$=_CTVALCOL10$(RIGAATT)
OLDVAL11$=_CTVALCOL11$(RIGAATT)
OLDVAL12$=_CTVALCOL12$(RIGAATT)
OLDVAL13$=_CTVALCOL13$(RIGAATT)
OLDVAL14$=_CTVALCOL14$(RIGAATT)
OLDVAL15$=_CTVALCOL15$(RIGAATT)
OLDVAL16$=_CTVALCOL16$(RIGAATT)
OLDVAL17$=_CTVALCOL17$(RIGAATT)
OLDVAL18$=_CTVALCOL18$(RIGAATT)
OLDVAL19$=_CTVALCOL19$(RIGAATT)
OLDVAL20$=_CTVALCOL20$(RIGAATT)

CURFLD=1
DONE = 0

START_INPUT_CYCLE:
IF DONE <> 0 THEN GOTO END_INPUT_CYCLE

SETTF "610-Commit,505-Rollback"
LOCATE 2+CURFLD, 2
PRINT _CTTITCOL$(CURFLD)+RIGHT$(FILLER$,MAXLENTIT-LEN( _CTTITCOL$(CURFLD)));
LOCATE 2+CURFLD, 2+MAXLENTIT

IFF _CTTYPECOL$(CURFLD) = "N"
IF CURFLD = 1 THEN VARN=VAL( _CTVALCOL1$(RIGAATT))
IF CURFLD = 2 THEN VARN=VAL( _CTVALCOL2$(RIGAATT))
IF CURFLD = 3 THEN VARN=VAL( _CTVALCOL3$(RIGAATT))
IF CURFLD = 4 THEN VARN=VAL( _CTVALCOL4$(RIGAATT))
IF CURFLD = 5 THEN VARN=VAL( _CTVALCOL5$(RIGAATT))
IF CURFLD = 6 THEN VARN=VAL( _CTVALCOL6$(RIGAATT))
IF CURFLD = 7 THEN VARN=VAL( _CTVALCOL7$(RIGAATT))
IF CURFLD = 8 THEN VARN=VAL( _CTVALCOL8$(RIGAATT))
IF CURFLD = 9 THEN VARN=VAL( _CTVALCOL9$(RIGAATT))

```

```

IF CURFLD = 10 THEN VARN=VAL (_CTVALCOL10$ (RIGAATT) )
IF CURFLD = 11 THEN VARN=VAL (_CTVALCOL11$ (RIGAATT) )
IF CURFLD = 12 THEN VARN=VAL (_CTVALCOL12$ (RIGAATT) )
IF CURFLD = 13 THEN VARN=VAL (_CTVALCOL13$ (RIGAATT) )
IF CURFLD = 14 THEN VARN=VAL (_CTVALCOL14$ (RIGAATT) )
IF CURFLD = 15 THEN VARN=VAL (_CTVALCOL15$ (RIGAATT) )
IF CURFLD = 16 THEN VARN=VAL (_CTVALCOL16$ (RIGAATT) )
IF CURFLD = 17 THEN VARN=VAL (_CTVALCOL17$ (RIGAATT) )
IF CURFLD = 18 THEN VARN=VAL (_CTVALCOL18$ (RIGAATT) )
IF CURFLD = 19 THEN VARN=VAL (_CTVALCOL19$ (RIGAATT) )
IF CURFLD = 20 THEN VARN=VAL (_CTVALCOL20$ (RIGAATT) )

' INPUT "N", VARN, _CTLENCOL (CURFLD)
INPUT "N", VARN, 13

' IF CURFLD = 1 THEN _CTVALCOL1$ (RIGAATT) ="PALLINO_VERDE"
IF CURFLD = 1 THEN _CTVALCOL1$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 2 THEN _CTVALCOL2$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 3 THEN _CTVALCOL3$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 4 THEN _CTVALCOL4$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 5 THEN _CTVALCOL5$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 6 THEN _CTVALCOL6$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 7 THEN _CTVALCOL7$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 8 THEN _CTVALCOL8$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 9 THEN _CTVALCOL9$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 10 THEN _CTVALCOL10$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 11 THEN _CTVALCOL11$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 12 THEN _CTVALCOL12$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 13 THEN _CTVALCOL13$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 14 THEN _CTVALCOL14$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 15 THEN _CTVALCOL15$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 16 THEN _CTVALCOL16$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 17 THEN _CTVALCOL17$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 18 THEN _CTVALCOL18$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 19 THEN _CTVALCOL19$ (RIGAATT) =STR$ (VARN, 0, 0)
IF CURFLD = 20 THEN _CTVALCOL20$ (RIGAATT) =STR$ (VARN, 0, 0)

ELSEF

IF CURFLD = 1 THEN VARA$=_CTVALCOL1$ (RIGAATT)
IF CURFLD = 2 THEN VARA$=_CTVALCOL2$ (RIGAATT)
IF CURFLD = 3 THEN VARA$=_CTVALCOL3$ (RIGAATT)
IF CURFLD = 4 THEN VARA$=_CTVALCOL4$ (RIGAATT)
IF CURFLD = 5 THEN VARA$=_CTVALCOL5$ (RIGAATT)
IF CURFLD = 6 THEN VARA$=_CTVALCOL6$ (RIGAATT)
IF CURFLD = 7 THEN VARA$=_CTVALCOL7$ (RIGAATT)
IF CURFLD = 8 THEN VARA$=_CTVALCOL8$ (RIGAATT)
IF CURFLD = 9 THEN VARA$=_CTVALCOL9$ (RIGAATT)
IF CURFLD = 10 THEN VARA$=_CTVALCOL10$ (RIGAATT)
IF CURFLD = 11 THEN VARA$=_CTVALCOL11$ (RIGAATT)
IF CURFLD = 12 THEN VARA$=_CTVALCOL12$ (RIGAATT)
IF CURFLD = 13 THEN VARA$=_CTVALCOL13$ (RIGAATT)
IF CURFLD = 14 THEN VARA$=_CTVALCOL14$ (RIGAATT)
IF CURFLD = 15 THEN VARA$=_CTVALCOL15$ (RIGAATT)
IF CURFLD = 16 THEN VARA$=_CTVALCOL16$ (RIGAATT)
IF CURFLD = 17 THEN VARA$=_CTVALCOL17$ (RIGAATT)
IF CURFLD = 18 THEN VARA$=_CTVALCOL18$ (RIGAATT)
IF CURFLD = 19 THEN VARA$=_CTVALCOL19$ (RIGAATT)
IF CURFLD = 20 THEN VARA$=_CTVALCOL20$ (RIGAATT)

' INPUT "A", VARA$, _CTLENCOL (CURFLD)
INPUT "A", VARA$, 13

IF CURFLD = 1 THEN _CTVALCOL1$ (RIGAATT) =VARA$
IF CURFLD = 2 THEN _CTVALCOL2$ (RIGAATT) =VARA$
IF CURFLD = 3 THEN _CTVALCOL3$ (RIGAATT) =VARA$
IF CURFLD = 4 THEN _CTVALCOL4$ (RIGAATT) =VARA$
IF CURFLD = 5 THEN _CTVALCOL5$ (RIGAATT) =VARA$
IF CURFLD = 6 THEN _CTVALCOL6$ (RIGAATT) =VARA$
IF CURFLD = 7 THEN _CTVALCOL7$ (RIGAATT) =VARA$
IF CURFLD = 8 THEN _CTVALCOL8$ (RIGAATT) =VARA$
IF CURFLD = 9 THEN _CTVALCOL9$ (RIGAATT) =VARA$
IF CURFLD = 10 THEN _CTVALCOL10$ (RIGAATT) =VARA$
IF CURFLD = 11 THEN _CTVALCOL11$ (RIGAATT) =VARA$
IF CURFLD = 12 THEN _CTVALCOL12$ (RIGAATT) =VARA$
IF CURFLD = 13 THEN _CTVALCOL13$ (RIGAATT) =VARA$
IF CURFLD = 14 THEN _CTVALCOL14$ (RIGAATT) =VARA$
IF CURFLD = 15 THEN _CTVALCOL15$ (RIGAATT) =VARA$
IF CURFLD = 16 THEN _CTVALCOL16$ (RIGAATT) =VARA$
IF CURFLD = 17 THEN _CTVALCOL17$ (RIGAATT) =VARA$
IF CURFLD = 18 THEN _CTVALCOL18$ (RIGAATT) =VARA$
IF CURFLD = 19 THEN _CTVALCOL19$ (RIGAATT) =VARA$
IF CURFLD = 20 THEN _CTVALCOL20$ (RIGAATT) =VARA$

ENDF
IF _TF = 610 THEN DONE = 1 : CURFLD = 0
IF _TF = 505 THEN DONE = -1 : CURFLD = 0

CURFLD = CURFLD + 1
IF _SCORRI=1 AND CURFLD >= _CTVTOTCOL THEN _SCORRI=0
IF CURFLD > _CTVTOTCOL THEN CURFLD = 1
GOTO START_INPUT_CYCLE

END_INPUT_CYCLE:
IFF DONE <> 1
DOROLLBACK=1

_CTVALCOL1$ (RIGAATT) = "PALLINO_VERDE"
_CTVALCOL2$ (RIGAATT) = OLDVAL2$
_CTVALCOL3$ (RIGAATT) = OLDVAL3$
_CTVALCOL4$ (RIGAATT) = OLDVAL4$
_CTVALCOL5$ (RIGAATT) = OLDVAL5$
_CTVALCOL6$ (RIGAATT) = OLDVAL6$
_CTVALCOL7$ (RIGAATT) = OLDVAL7$
_CTVALCOL8$ (RIGAATT) = OLDVAL8$
_CTVALCOL9$ (RIGAATT) = OLDVAL9$

```

```

_CTVALCOL10$(RIGAATT) = OLDVAL10$
_CTVALCOL11$(RIGAATT) = OLDVAL11$
_CTVALCOL12$(RIGAATT) = OLDVAL12$
_CTVALCOL13$(RIGAATT) = OLDVAL13$
_CTVALCOL14$(RIGAATT) = OLDVAL14$
_CTVALCOL15$(RIGAATT) = OLDVAL15$
_CTVALCOL16$(RIGAATT) = OLDVAL16$
_CTVALCOL17$(RIGAATT) = OLDVAL17$
_CTVALCOL18$(RIGAATT) = OLDVAL18$
_CTVALCOL19$(RIGAATT) = OLDVAL19$
_CTVALCOL20$(RIGAATT) = OLDVAL20$

ENDF
CHIUDIFIN 1

RETURN

'//////////
'//////////
'//////////
'//////////

'EDIT_ROW:
'MAXLENTIT=0
'MAXLENROW=0
'FILLER$= ".....: "
'FOR I=1 TO _CTVTOTCOL
'IF LEN( _CTTITCOL$(I) ) > MAXLENTIT THEN MAXLENTIT = LEN( _CTTITCOL$(I) ) + 3
'IF _CTLENCOL(I) > MAXLENROW THEN MAXLENROW = _CTLENCOL(I)
'NEXT
'
'APRIFIN_TIT 1,1,0, 3, 5, 4+_CTVTOTCOL, 4+MAXLENTIT+MAXLENROW, "Utente", ""
'
'DOROLLBACK=0
'_SCORRI=1
'_INTER=0
'DONE = 0
'NXTFLD = 0
'OLDVAL1$=_CTVALCOL1$(RIGAATT)
'OLDVAL2$=_CTVALCOL2$(RIGAATT)
'OLDVAL3$=_CTVALCOL3$(RIGAATT)
'CURFLD=1
'DONE = 0
'START_INPUT_CYCLE:
'IF DONE <> 0 THEN GOTO END_INPUT_CYCLE
'
'SETTF "610-Commit,505-Rollback"
'LOCATE 2+CURFLD, 2
'PRINT _CTTITCOL$(CURFLD)+RIGHT$(FILLER$,MAXLENTIT-LEN( _CTTITCOL$(CURFLD)));
'LOCATE 2+CURFLD, 2+MAXLENTIT
'IFF _CTTYPECOL$(CURFLD) = "N"
'IF CURFLD = 1 THEN VARN=VAL( _CTVALCOL1$(RIGAATT) )
'IF CURFLD = 2 THEN VARN=VAL( _CTVALCOL2$(RIGAATT) )
'IF CURFLD = 3 THEN VARN=VAL( _CTVALCOL3$(RIGAATT) )
'INPUT "N", VARN, _CTLENCOL(CURFLD)
'IF CURFLD = 1 THEN _CTVALCOL1$(RIGAATT)=STR$(VARN,0,0)
'IF CURFLD = 2 THEN _CTVALCOL2$(RIGAATT)=STR$(VARN,0,0)
'IF CURFLD = 3 THEN _CTVALCOL3$(RIGAATT)=STR$(VARN,0,0)
'ELSEF
'IF CURFLD = 1 THEN VARA$=_CTVALCOL1$(RIGAATT)
'IF CURFLD = 2 THEN VARA$=_CTVALCOL2$(RIGAATT)
'IF CURFLD = 3 THEN VARA$=_CTVALCOL3$(RIGAATT)
'INPUT "A", VARA$, _CTLENCOL(CURFLD)
'IF CURFLD = 1 THEN _CTVALCOL1$(RIGAATT)=VARA$
'IF CURFLD = 2 THEN _CTVALCOL2$(RIGAATT)=VARA$
'IF CURFLD = 3 THEN _CTVALCOL3$(RIGAATT)=VARA$
'ENDF
'IF _TF = 610 THEN DONE = 1 : CURFLD = 0
'IF _TF = 505 THEN DONE = -1 : CURFLD = 0
'CURFLD = CURFLD + 1
'IF _SCORRI=1 AND CURFLD >= _CTVTOTCOL THEN _SCORRI=0
'IF CURFLD > _CTVTOTCOL THEN CURFLD = 1
'GOTO START_INPUT_CYCLE
'
'END_INPUT_CYCLE:
'IFF DONE <> 1
'DOROLLBACK=1
'_CTVALCOL1$(RIGAATT) = OLDVAL1$
'_CTVALCOL2$(RIGAATT) = OLDVAL2$
'_CTVALCOL3$(RIGAATT) = OLDVAL3$
'ENDF
'CHIUDIFIN 1
'
'RETURN
'
```

Caso: dimensioni per APRIFIN nella gestione di finestra per client palmare

Le dimensioni massime relative alla APRIFIN lanciabile da client palmare sono COLONNE 32 e RIGHE 17, carattere non proporzionale. Alcuni dispositivi e versioni di sistema operativo, posizionano la finestra in colonna 1 e riga 1 del video, indipendentemente dai valori delle coordinate di posizionamento di APRIFIN.

Esempio INPUT_EXT_EXT

```

DIM ARRAY$()
_INTER = 0

GOSUB PROTOCOL

SETTF "501Invio,505Esci"
LOCATE 4,25 : INPUT_EXT_EXT 0,"COMBO",PROT1$,10,10,"",ARRAY$
IF _TF=505 THEN END
IF _TF=501 THEN LOCATE 5,25 : INPUT_EXT_EXT 0,"COMBO",PROT2$,10,10,"",ARRAY$
LOCATE 6,25 : PRINT "wait" : SLEEP 1000
LOCATE 7,25 : INPUT_EXT_EXT 0,"COMBO",PROT3$,10,10,"",ARRAY$
END
```



```

PROTOCOL:
  AZZARRAY ARRAY$( )
  ARRAY$(1) = "1=http"
  ARRAY$(2) = "2=https"
RETURN

```

Esempio INPUT3

```

_INTER = 0
TEST:
SETTF "501Invio,505Esci"
  LOCATE 4,25 : INPUT3 0,"N",PROT1,3,3,"",ERR$
  IF _TF=505 THEN END
  IF PROT1 <1 OR PROT1 >100 THEN ERR$ = "Valore minimo 1 massimo 100" GOTO TEST:
END

```

Caso: logica delle coordinate di apertura della finestra taglie nel documento di magazzino, per gestire la posizione di una propria aprifin

In caso di apertura di finestra con posizione della riga in negativo (quando coperta dalla finestra taglie), l'apertura della finestra cerca di non coprire la riga interessata; se c'è spazio viene aperta sotto la riga interessata, altrimenti sopra. Si riporta uno schema per ciascuna tabella taglie su tutte le righe (la colonna non cambia mai) il calcolo che risulta in base all'altezza della finestra taglie (cioè in base alla quantità di taglie) ed il posizionamento di riga del cursore:

```

Taglie 1: si apre dopo la riga in corso; dalla 2 'ultima passa sopra la riga;
Taglie 2: si apre dopo la riga in corso; dalla 2 'ultima passa sopra la riga;
Taglie 3: si apre dopo la riga in corso; dalla 3 'ultima passa sopra la riga;
Taglie 4: si apre dopo la riga in corso; dalla 4 'ultima passa sopra la riga;
Taglie 5: si apre dopo la riga in corso; dalla 5 'ultima passa sopra la riga;
Taglie 6: si apre dopo la riga in corso; dalla 6 'ultima passa sopra la riga;
Taglie 7: si apre dopo la riga in corso; dalla 7 'ultima passa sopra la riga;
Taglie 8: si apre dopo la riga in corso; dalla 8 'ultima passa sopra la riga;
Taglie 9: si apre dopo la riga in corso; dalla 9 'ultima passa sopra la riga;
Taglie 10: si apre dopo la riga in corso; dalla 10 'ultima passa sopra la riga;
cambio di logica
Taglie 11: si apre dopo la riga in corso; dalla 11^ si sovrappone e resta fissa fino alla 9 'ultima (dalla quale passa sopra);
Taglie 12: si apre dopo la riga in corso; dalla 5^ si sovrappone e resta fissa fino alla 8 'ultima (dalla quale passa sopra);
Taglie 13: si apre dopo la riga in corso; dalla 4^ si sovrappone e resta fissa fino alla 7 'ultima (dalla quale passa sopra);
Taglie 14: si apre dopo la riga in corso; dalla 3^ si sovrappone e resta fissa fino alla 6 'ultima (dalla quale passa sopra);
Taglie 15: si apre dopo la riga in corso; dalla 2^ si sovrappone e resta fissa fino alla 5 'ultima (dalla quale passa sopra);
cambio di logica
Taglie 16: si apre sulla 1^ riga fino alla 5 'ultima e dalla 4 'ultima passa sopra;
Taglie 32: come taglia 16 ed è a scorrimento;

```

STAMPE

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
Tasti funzione [F5]	<p>Riga di stampa. es</p> <p>Il tasto funzione posiziona il cursore in un campo a sinistra della linea verticale dell'Editor, su cui con [F2] aprire l'Elenco di tipologie di stampa della riga corrente:</p> <p>'A' stampa in carattere automatico, cioè normale o compressa in base alla lunghezza di riga assegnata nella configurazione di stampante secondo i seguenti rapporti fra la Lunghezza riga stampante e Numero di caratteri della riga in stampa (la riga 'A' è una riga 'N' che diviene 'C' se riga in stampa supera la larghezza del modulo):</p> <ul style="list-style-type: none"> • Normale (80 contro <=79 e 132 contro <=131) • Compresso (80 contro >79 e 132 contro >131). <p>'N' stampa in carattere normale. 'E' stampa in carattere espanso (non gestibile in Rtf). 'G' stampa in carattere grassetto. 'C' stampa in carattere compresso (non gestibile in Rtf). 'I' stampa in carattere corsivo. 'F' scrittura della riga come record su File. Il nome file è obbligatorio (tutto il percorso compreso fra doppi apici "c:\tmp\esempio.txt"). Il file è occupato fino al termine del programma e solo dopo sono consentite operazioni di gestione file (open, copia,...); un'alternativa consiste nell'uso di stampante configurata di tipo File. NB: è obbligatorio per ciascuna riga, anche in caso di duplicazione di riga, poichè l'assenza del percorso può provocare errore per violazione di memoria (specialmente nella Libreria in quanto sprovvista di controllo da parte del compilatore). 'P' invia alla stampante un salto pagina forzato; la riga non è scrivibile e la quantità di esse deve essere compatibile con lunghezza del modulo; 'S' stampa in carattere sottolineato (non gestibile in 'Pdf').</p> <p>Per le combinazioni di carattere sulla stessa riga utilizzare LPRINTTXT</p>	S/C/Cd
Tasto funzione [F2]	<p>Campo di stampa. es</p> <p>La stampa dei valori di Variabili e di Dizionario sulla singola riga avviene tramite l'attivazione del campo di stampa in cui definire nel 'Nome campo' la variabile o il Dizionario (ricercabile in elenco tramite [F2]) ed i relativi attributi stampa, proposti, modificabili: la lunghezza di stampa (se viene riconosciuta ne viene proposta la relativa misura, altrimenti è 15 caratteri); l'allineamento (default = automatico); il segno di valori negativi (default = sinistra); il numero di decimali (default = solo se eventualmente presenti); il punto separatore di migliaia (default = assente); il valore a zero (default = non stampare); il carattere di riempimento per valori di lunghezza < alla dimensione del campo (default = blank); l'azzeramento delle variabili dopo la stampa, tranne per i valori di quelle utilizzate come indici di array (default = no). A seconda del tipo di valore che va stampato, questo ha già una forma predefinita: es. data, ora, codice conto, codice tributo, protocolli Iva, codice articolo, anno gestito, anno in linea, ecc. Il nome della Variabile da inserire nel Nome campo, può essere valore numerico, alfanumerico e array, anche un'espressione o risultato di essa. Alla conferma del campo il nome è visualizzato in reverse nell'editor in lunghezza prestabilita e prefissato con il carattere tilde (~).</p> <p>Esempio: VARNUM, VARALF\$, VN(3), VA\$(2), ESPR=VARNUM+VAL(VARALF\$) Esempio: ESPR\$=VARALF\$+ STR\$(VARNUM,0.0)</p>	S/C/Cd
Tasto funzione [SHF5]	<p>Riga di intestazione. es</p> <p>Contrassegna la riga corrente con la lettera 'H' e le attribuisce proprietà di Intestazione di Stampa ad ogni salto pagina; non può consistere in una riga di ciclotabellare, nè può contenere comandi ETICHETTE, GOTO e GOSUB. Consentite invece le istruzioni, le assegnazioni e la stampa del valore delle variabili in quel momento. Consentiti gruppi di righe 'H' solo se contigue, non vuote e non precedute da altre righe di stampa. Consentite più testate con memorizzazione dell'ultima incontrata per essere stampata automaticamente ad ogni salto pagina prima di stampare la prima riga.</p>	S/C/Cd
PRINTH	Istruzione che forza la stampa della riga di intestazione	S/C/Cd
_INTEST	Abilita/disabilita intestazione moduli. es	S/C/Cd
_NUMSTAMP	Numero di stampante aperta	S/C/Cd
APRISTAMP	Par: <numstamp>, <nomemod\$>. Istruzione che apre stampante. es	S/C/Cr/Cd/Sm
APRISTAMP_EXT	Par: <numstamp>, <nomemod\$>, <lun>, [modstsp\$] Istruzione che apre stampante con lunghezza riga	S/C/Cr/Cd/Sm
	<p>Nota: il parametro [modstsp\$] per la stampa modulo grafico per Sprix creato nella App PassBuilder è un dato opzionale relazionato alle istruzioni grafiche LPRINTGRAF_???, pertanto, se tale parametro [modstsp\$] viene omesso l'istruzione APRISTAMP_EXT conserva l'originaria funzionalità relazionata alle tradizionali istruzioni LPRINT???. es.</p> <p>L'istruzione è disponibile in Collage Server Remoto se la stampante è correttamente configurata per non essere interattiva (no I/O video)</p>	
_STPLUNGRIG	Variabile lunghezza della pagina in caratteri	S/C/Cd
_STPLARGCOL	Variabile larghezza della pagina in caratteri	S/C/Cd

_STPLUNGRMM	Variabile lunghezza della pagina in millimetri	S/C/Cd
_STPLARGCMM	Variabile larghezza della pagina in millimetri	S/C/Cd
APRISTAMP_FILE	Par: <tipofile\$>,<nomefile\$>,<lar>,<modstsp\$> Stampa su file PDF / CSV. Vedere anche istruzioni relative alla gestione file (_ACCFILEUTE e _ERRFILE\$)	S/C/Cr/Cd
APRISTAMP_EMAIL	Par: <numstamp\$>,<mittente\$>,<destinatario\$>,<cc\$>,<ccn\$>,<oggetto\$>,<testo\$>,<firma\$>,<nomeallegato\$>,<allegato\$>,<modstsp\$> Istruzione che apre una stampante email Nota: il parametro [modstsp\$] per la stampa modulo grafico per Sprix creato nella App PassBuilder è un dato opzionale relazionato alle istruzioni grafiche LPRINTGRAF_???, pertanto, se tale parametro [modstsp\$] viene omesso l'istruzione APRISTAMP_EMAIL conserva l'originaria funzionalità relazionata alle tradizionali istruzioni LPRINT???. es. A semplice titolo informativo storico: prima dell'introduzione di APRISTAMP_EMAIL, il work around per l'invio automatico di email era consentito attraverso le tre seguenti impostazioni: l'indirizzo e-mail nei conti del piano dei conti, una stampante e-mail configurata in automatico su emissione documenti, un Collage oppure uno Sprix con cicli FOR di lettura (con modifica del cliente eristampa documento) di un documento fittizio ma memorizzato su record (ad esempio una matrice), in cui inserire l'articolo testo con il contenuto desiderato. L'istruzione è disponibile in Collage Server Remoto se la stampante è correttamente configurata per non essere interattiva (no I/O video) Non consentita nei Livelli di sviluppo obsoleti (F2, L3, F3)	S/C/Cr/Cd
APRISTAMP_EMAIL_EXT	Par:<numstamp\$>,<mittente\$>,<nomemittente\$>,<rispondiun\$>,<nomerispondiun\$>,<destinatario\$>,<cc\$>,<ccn\$>,<oggetto\$>,<testo\$>,<firma\$>,<filetestohtml\$>,<nomeallegato\$>,<allegati\$>,<mittenteinCCN\$>,<confermalettura\$>,<prioritasmt\$>,<modstsp\$> Istruzione per stampare su stampante email e tra i vari parametri gli si può passare il percorso di un file HTML che diventerà (quindi sostituirà) il corpo dell'email.	S/C/Cr/Cd
ON_STARTSTPEMAIL:	Evento all'apertura della stampante Email Nota bene: l'evento ha effetti diversi sulle Variabili Collage di Modulistica. Se il modulo è "grafico" questo viene composto dopo che scatta l'evento in questione, pertanto le variabili sono valorizzate. Se il modulo è "tradizionale" questo viene composto prima che scatti l'evento in questione, pertanto le variabili restano vuote; in alternativa usare i consueti eventi ON_STARTMM: oppure ON_PUTMM:	C/Cr/Cd
EMAILGETATT	Istruz. che legge nella struttura _EMAIL??? le impostazioni correnti di invio Email;	C
EMAILPUTATT	Istruz. che tramite la struttura _EMAIL??? modifica le impostazioni d'invio Email;	C
_EMAIL	Variabili di struttura Email SMTP per le istruzioni EMAILGETATT ed EMAILPUTATT	C
_ERRSTPEMAIL	Variabile errore se stampante non di tipo email	S/C/Cd
CHIUDISTAMP_EXT	Istruzione che chiude la stampante email aperta con APRISTAMP_EMAIL; è in grado di fungere come CHIUDISTAMP, ma non valorizza le variabili _ERRCHSTAMP e _ERRCHSTAMP\$	S/C/Cr/Cd/Sm
_ERRCHSTAMP	Variabile con il codice risultato di CHIUDISTAMP_EXT	S/C/Cd
_ERRCHSTAMP\$	Variabile con il messaggio operazione di CHIUDISTAMP_EXT	S/C/Cd
_ERRIMAP	Variabile errore memorizzazione posta inviata dopo CHIUDISTAMP_EXT	S/C/Cd
_ERRIMAP\$	Variabile errore memorizzazione posta inviata dopo CHIUDISTAMP_EXT	S/C/Cd
CHIUDISTAMP	Istruzione che chiude la stampante aperta. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. es	S/C/Cd
LUNGMOD\$()	Par: (<numrighe>). Funzione che setta la dimensione del modulo stampa	S/C/Cr/Cd
ACAPO\$()	Par: (). Funzione che torna caratteri per fare salto riga	S/C/Cr/Cd
SALTOPAG\$()	Funzione che torna una stringa per fare il salto pagina. es	S/C/Cr/Cd/Sm
LPRINT	Par: <espr\$>[;]. Istruzione che invia caratteri alla stampante. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. es es	S/C/Cr/Cd/Sm
INIZCO\$()	Par: (). Funzione che torna una stringa per inizio compresso	S/C/Cr/Cd
FINECO\$()	Par: (). Funzione che torna una stringa per fine compresso	S/C/Cr/Cd
INIZES\$()	Par: (). Funzione che torna una stringa per inizio espanso	S/C/Cr/Cd
FINEES\$()	Par: (). Funzione che torna una stringa per fine espanso	S/C/Cr/Cd
INIZGR\$()	Par: (). Funzione che torna una stringa per inizio grassetto	S/C/Cr/Cd
FINEGR\$()	Par: (). Funzione che torna una stringa per fine grassetto	S/C/Cr/Cd
INIZIT\$()	Par: (). Funzione che torna una stringa per inizio corsivo	S/C/Cr/Cd
FINEIT\$()	Par: (). Funzione che torna una stringa per fine corsivo	S/C/Cr/Cd
INIZSO\$()	Par: (). Funzione che torna una stringa per iniz.sottolineato	S/C/Cr/Cd
FINESO\$()	Par: (). Funzione che torna una stringa per fine sottolineato	S/C/Cr/Cd
LPRINTTXT	Par: <str\$>,<font\$>,<dimfont\$>,<attrfont\$>,<um>,<x>,<y>,<opz\$> Istruzione che stampa testo grafico. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. es	S/C/Cr/Cd/Sm
LPRINTBARCODE	Par: <codice\$>,<tipo>,<prncod>,<um>,<x>,<y>,<cx>,<cy> Istruzione che stampa codice a barre CODE 39 EAN 8/13. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. es	S/C/Cr/Cd/Sm
LPRINTQRCODE	Par: <codice\$>,<tipo>,<lunghezza_codice>,<um>,<x>,<y>,<lunghezza_lato>,<livello_errore> Istruzione che stampa qrcode. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. es	S/C/Cr/Cd/Sm
LPRINTIMG	Par: <nomefile\$>,<um>,<x>,<y>,<cx>,<cy> Istruzione che stampa immagini. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder	S/C/Cr/Cd/Sm
LPRINTLINE	Par: <um>,<x>,<y>,<lung>,<spessore>,<opz\$> Istruzione che stampa una linea. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder	S/C/Cr/Cd/Sm
LPRINTGRAF_NUM	Par: <nomediz\$>,<indicediz>,<num>	S/C/Cr/Cd

	Istruzione per valorizzare le variabili di dizionario numeriche nelle stampe modulo grafico Sprix in App PassBuilder. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. es.	
LPRINTGRAF_STR	Par: <nomediz\$>,<indicediz>,<str\$> Istruzione per valorizzare le variabili di dizionario alfanumeriche nelle stampe modulo grafico Sprix in App PassBuilder. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. es.	S/C/Cr/Cd
LPRINTGRAF_SEZ	Par: <tiposez>,<dovedettaglio> Istruzione per scegliere la sezione da stampare nel modulo grafico Sprix in App PassBuilder. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. es.	S/C/Cr/Cd
GETMODSTPGRAF	Par: <tipo>. Istruzione per recuperare l'elenco dei moduli di stampa. es.	S/C/Cr/Cd

**STAMPE
CSV, XML, EXCEL, FATTURA PA**

Il formato CSV è disponibile **ESCLUSIVAMENTE** nelle stampe della modulistica personalizzata. In differenti contesti, tutti i dispositivi con formato CSV non risultano disponibili all'atto di stampa. Relativamente all'utilizzo della modulistica personalizzata vedere sezione "Servizi", modulo "Personalizzazioni", capitoli "Modulistica documenti grafica" e "Stampe personalizzate"

Il formato XML EXCEL è disponibile **ESCLUSIVAMENTE** nelle stampe della modulistica personalizzata. In differenti contesti, tutti i dispositivi con formato XML EXCEL non risultano disponibili all'atto di stampa. Relativamente all'utilizzo della modulistica personalizzata vedere sezione "Servizi", modulo "Personalizzazioni", capitoli "Modulistica documenti grafica" e "Stampe".

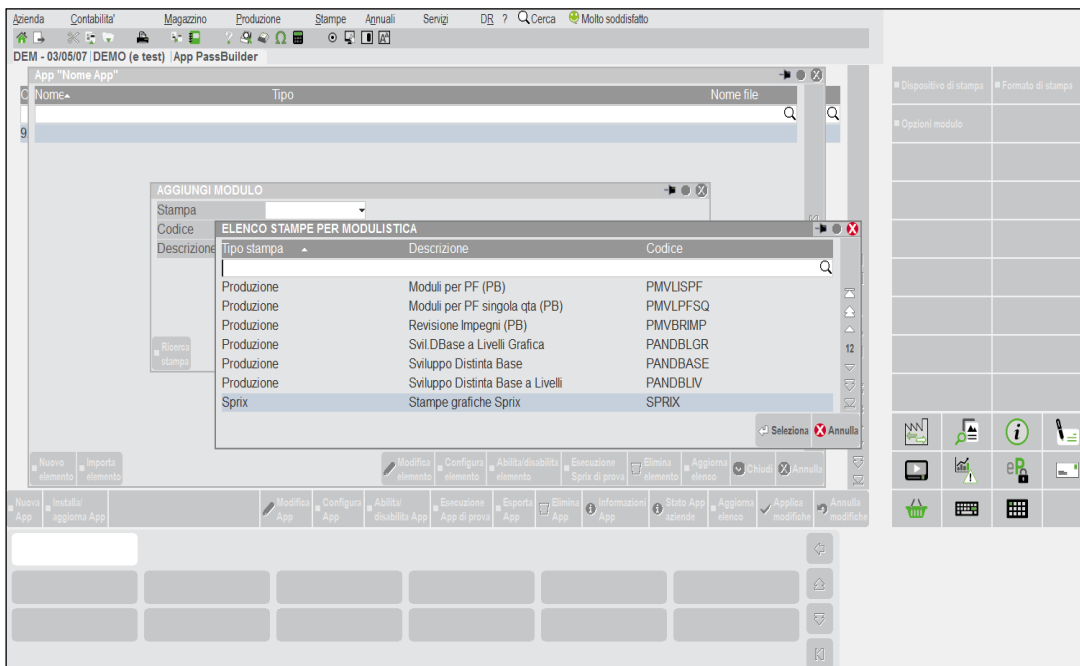
Il formato FATTURA PA è disponibile **ESCLUSIVAMENTE** nelle stampe della modulistica personalizzata distribuita fra gli esempi Passepartout e denominata "Fattura XML PA". In differenti contesti, tutti i dispositivi con formato FATTURA PA non risultano disponibili all'atto di stampa. Relativamente all'utilizzo della modulistica personalizzata vedere sezione "Servizi", modulo "Personalizzazioni", capitoli "Modulistica documenti grafica".

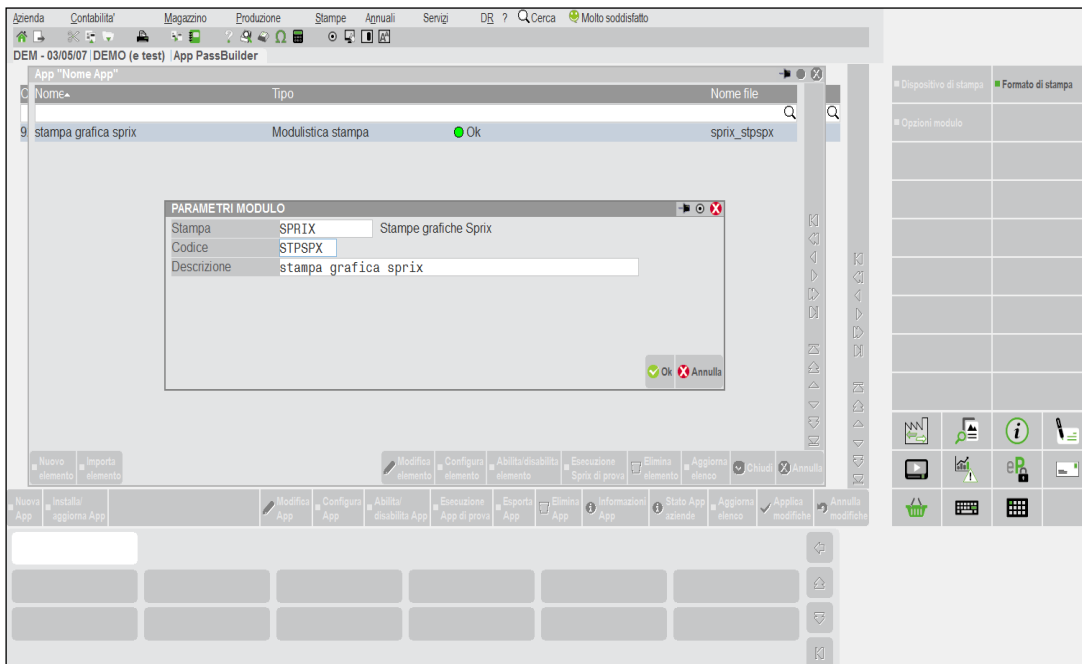
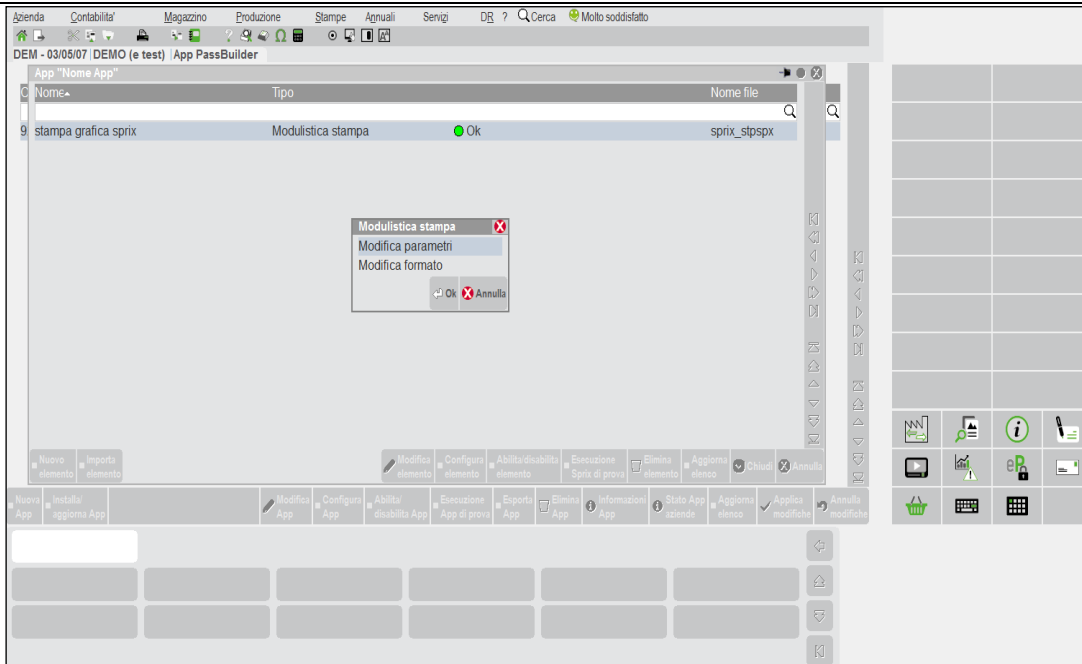
STAMPA MODULO GRAFICO IN SPRIX (per App in PassBuilder)

S/C/Cd

Procedura per realizzare la Stampa Grafica in Sprix per App in PassBuilder

Creazione della modulistica di stampa: creare nella App un nuovo elemento "Modulistica stampa" selezionando il tipo stampa "SPRIX". La creazione del modulo avviene, come per tutti gli altri tipi di stampa, utilizzando le variabili di dizionario generale, le variabili di dizionario stampa e la combinazione delle due.





Il dizionario di stampa prevede per ogni sezione un vettore di variabili di tipo numerico e un vettore di variabili di tipo stringa entrambe di 99 elementi. Quando si esegue una stampa grafica, prima dell'esecuzione del formato grafico, viene generato un array di elementi che compongono il record fisico di stampa. Gli elementi che compongono il record di stampa sono i seguenti:

_inteststr → vettore variabili stringa INTESAZIONE (disponibili in tutte le sezioni)

_intestnum → vettore variabili numeriche INTESAZIONE (disponibili in tutte le sezioni)

nota: le suddette variabili sono di intestazione di stampa e non possono mai cambiare fino all'esecuzione del comando CHIUDISTAMP, se sono presenti più valori viene sempre letto l'ultimo.

_corpostr → vettore variabili stringa CORPO (disponibili nella sezione di corpo e sezioni dettaglio)

_corponum → vettore variabili numeriche CORPO (disponibili nella sezione di corpo e sezioni dettaglio)

nota: le suddette variabili compongono i veri dati di stampa e servono anche per eseguire ordinamenti o rotture dei dati che vengono stampati.

_det1str → vettore variabili stringa DETTAGLIO 1 (disponibili nella sezione dettaglio 1)

_det1num → vettore variabili numeriche DETTAGLIO 1 (disponibili nella sezione dettaglio 1)

_det2str → vettore variabili stringa DETTAGLIO 2 (disponibili nella sezione dettaglio 2)

_det2num → vettore variabili numeriche DETTAGLIO 2 (disponibili nella sezione dettaglio 2)

_det3str → vettore variabili stringa DETTAGLIO 3 (disponibili nella sezione dettaglio 3)

_det3num → vettore variabili numeriche DETTAGLIO 3 (disponibili nella sezione dettaglio 3)

nota: le suddette variabili compongono i sottocorpi di stampa e possono variare al variare di un corpo: esempio nel corpo si stampa il codice articolo nei tre elementi di dettaglio si inseriscono le taglie i lotti, gli alias, ecc...

Tali variabili non possono essere utilizzate per rotture o ordinamenti.

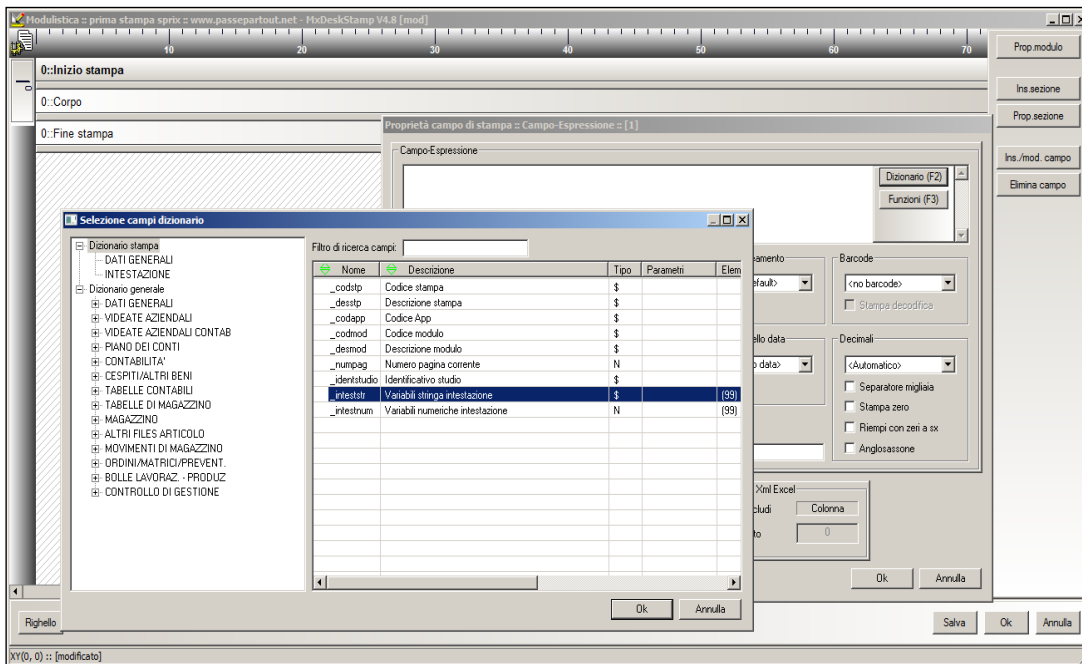
_pedestr → vettore variabili stringa PIEDE (disponibili nella sezione fine stampa)
_pedenum → vettore variabili numeriche PIEDE (disponibili nella sezione fine stampa)
nota: le suddette variabili di piede non possono mai cambiare fino all'esecuzione del comando CHIUDISTAMP, se sono presenti più valori viene sempre letto l'ultimo.

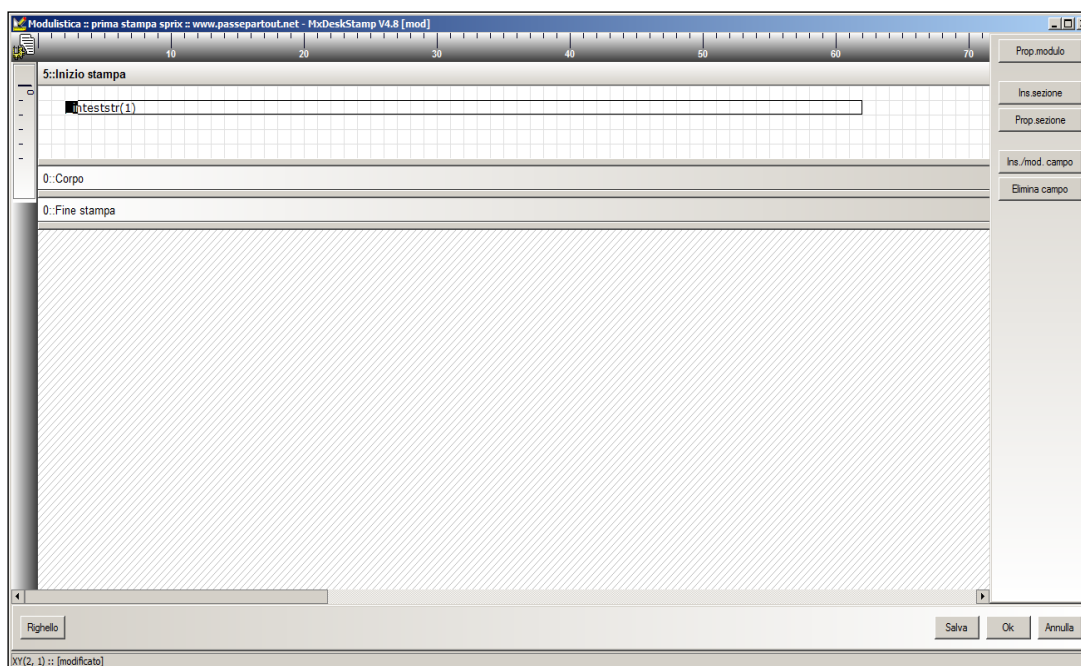
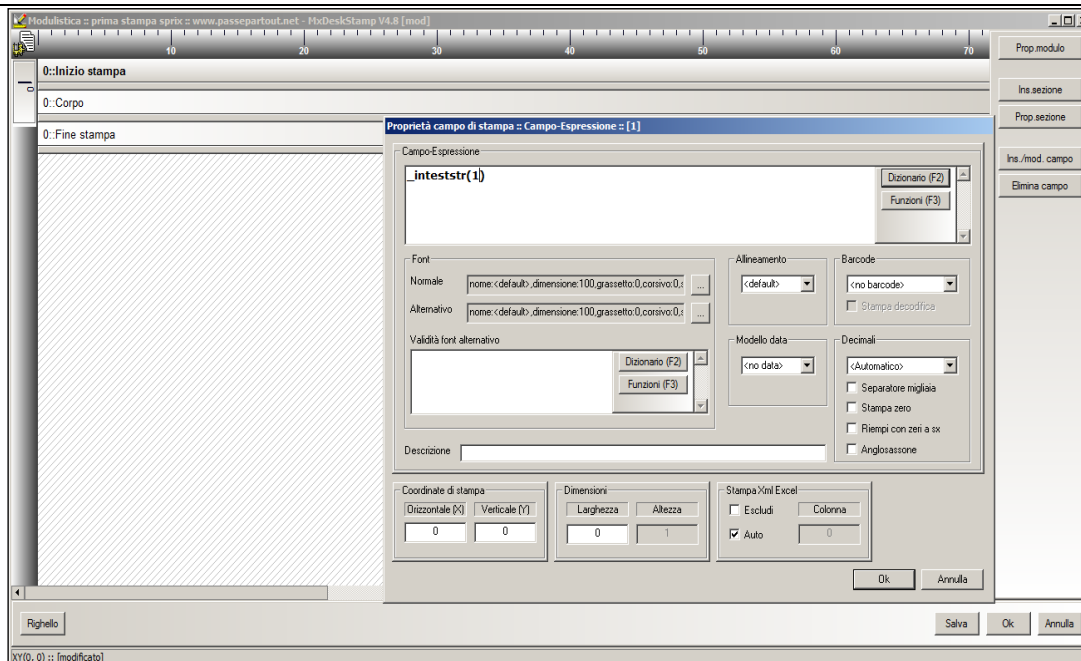
Esempio di un recordset di stampa corretto:

inteststr	corpostr	det1str	det2str	det3str	Piedestr
Proprietario	Articolo	Taglia	Lotto	Alias	Totale
Giovanni	Monitor	37			10
Giovanni	Monitor	37	1		10
Giovanni	Monitor	37	1	Lcd	10
Giovanni	Monitor	38			10
Giovanni	Monitor	38	2		10
Giovanni	Stampante				10
Giovanni					

Esempio di un recordset di stampa errato:

inteststr	corpostr	det1str	det2str	det3str	Piedestr
Proprietario	Articolo	Taglia	Lotto	Alias	Totale
Giovanni	Monitor	37			10
Giovanni	Monitor	37	1		10
Giovanni	Monitor	37	1	Lcd	10
Francesco	Monitor	38			10
Francesco	Monitor	38	2		10
Francesco	Stampante				10
Francesco					





La realizzazione del formato rende disponibile un tasto di tipo scheda che consente la scelta del modulo di stampa



Il programmatore Sprix deve inizializzare le variabili di dizionario di stampa inserite nel modulo con i valori che vuole stampare o utilizzare come parametri per accesso alle variabili di dizionario generale prima di eseguire la stampa delle sezioni.

Generare la stampa da Sprix

- 1 Indicare un modulo di stampa in apertura stampante
Esempi: dato il seguente codice modulo Sprix "STPSPX" l'apertura è tramite APRISTAMP_EXT -1, "MOD", 220, "STPSPX" oppure APRISTAMP_EMAIL 10,"aa@bb.it","cc@dd.it","","Oggetto","Testo","","STPSPX"
- 2 Inizializzare le variabili di dizionario stampa utilizzate nel modulo e comandare la stampa delle sezioni corpo o dettaglio

Per generare la stampa grafica il programmatore Sprix deve valorizzare le variabili di dizionario utilizzate nel modulo indicato in apertura stampante e poi comandare la stampa della sezione.

Le sezioni disponibili sono:

- Intestazione
- Corpo
- Dettaglio 1
- Dettaglio 2
- Dettaglio 3
- Piede

Per le sezioni Intestazione e piede non è necessario comandarne la stampa, in quanto vengono stampate in automatico rispettivamente prima del primo corpo o dettaglio stampato e dopo l'ultimo corpo o dettaglio stampato.

Per valorizzare le variabili di dizionario utilizzare le seguenti istruzioni Sprix per inizializzare variabili di stampa di tipo numerico e stringa

- LPRINTGRAF_NUM
- LPRINTGRAF_STR

Entrambe le istruzioni hanno 3 parametri:

1. Nome vettore variabile di dizionario di stampa
2. Indice vettore variabile di dizionario di stampa
3. Valore da assegnare

```
LPRINTGRAF_STR "_corpostr", 1, "Stringa corpo 1"
LPRINTGRAF_STR "_corpostr", 2, "Stringa corpo 2"
LPRINTGRAF_NUM "_corponum", 1, 1
LPRINTGRAF_NUM "_corponum", 1, 2
```

Dopo l'inizializzazione delle variabili dizionario di stampa occorre stampare la sezione tramite la seguente istruzione: LPRINTGRAF_SEZ 0, 0

Il primo parametro indica il tipo di sezione:

- 0 = Corpo
- 1 = Dettaglio 1
- 2 = Dettaglio 2
- 3 = Dettaglio 3

Il secondo parametro, utilizzato solo per sezioni dettaglio (il primo parametro vale 1, 2 o 3) indica dove collocare il dettaglio:

- 1 = Stampa il dettaglio dopo l'ultimo corpo stampato
- 2 = Ordina il dettaglio con i corpi in caso di modulo con ordinamento
- 3 = Stampa il dettaglio prima di tutti i corpi
- 4 = Stampa il dettaglio dopo tutti i corpi

3 Infine, chiudere le stampa con CHIUDISTAMP

Nota Bene: l'esecuzione di una stampa avviene tramite due processi:

1. Caricamento buffer di stampa

Il caricamento del buffer di stampa avviene tramite l'utilizzo delle istruzioni sprix LPRINTGRAF_NUM LPRINTGRAF_STR con la chiusura del record del buffer LPRINTGRAF_SEZ. Durante il caricamento del buffer i dati di intestazione e piede devono essere uguali. Se diversi possono essere interpretati per eseguire rotture ma è sempre letto l'ultimo valore presente nel buffer.

2. Esecuzione del modulo di stampa.

L'esecuzione del modulo elabora i dati presenti nel buffer è in base ai campi che compongono il modulo, intestazioni rotture ordinamenti, esegue la stampa. Gli unici dati utili a livello di rotture e ordinamento sono i dati presenti nelle variabili di corpo. I sottocorpi non possono essere utilizzati a livello di ordinamento o rotture.

Esempio

```
APRISTAMP_EXT -1, "PIPP0", 220, "STPSPX"
```

```
LPRINTGRAF_STR "_inteststr", 1, "Stringa Intestazione 1"
LPRINTGRAF_NUM "_intestnum", 1, 1234
LPRINTGRAF_STR "_inteststr", 2, "Stringa Intestazione 2"
LPRINTGRAF_NUM "_intestnum", 2, 5678
```

```
FOR CORPO = 1 TO 5
  FOR RIGA = 1 TO 2
    LPRINTGRAF_STR "_corpostr", RIGA, "Corpo " + STR$(CORPO,0,0) + " Riga " + STR$(RIGA,0,0)
    LPRINTGRAF_NUM "_corponum", RIGA, (CORPO * 100) + RIGA
  NEXT
  LPRINTGRAF_SEZ 0, 0
```

```
FOR RIGA = 1 TO 2
  LPRINTGRAF_STR "_det1str", RIGA, "Corpo " + STR$(CORPO,0,0) + " Dettaglio 1" + " Riga " + STR$(RIGA,0,0)
  LPRINTGRAF_NUM "_det1num", RIGA, (CORPO * 100) + 10 + RIGA
NEXT
```


LPRINTGRAF_SEZ 1, 3 ' Stampa Dettaglio prima di tutti i corpi

FOR RIGA = 1 TO 2

LPRINTGRAF_STR "_det2str", RIGA, "Corpo " + STR\$(CORPO,0,0) + " Dettaglio 2" + " Riga " + STR\$(RIGA,0,0)

LPRINTGRAF_NUM "_det2num", RIGA, (CORPO * 100) + 20 + RIGA

NEXT

LPRINTGRAF_SEZ 2, 1 ' Stampa Dettaglio dopo il corpo

FOR RIGA = 1 TO 2

LPRINTGRAF_STR "_det3str", RIGA, "Corpo " + STR\$(CORPO,0,0) + " Dettaglio 3" + " Riga " + STR\$(RIGA,0,0)

LPRINTGRAF_NUM "_det3num", RIGA, (CORPO * 100) + 30 + RIGA

NEXT

LPRINTGRAF_SEZ 3, 4 ' Stampa dettaglio dopo tutti i corpi

NEXT

LPRINTGRAF_STR "_pedestr", 1, "Stringa Piede 1"

LPRINTGRAF_NUM "_pedenum", 1, 1111

LPRINTGRAF_STR "_pedestr", 2, "Stringa Piede 2"

LPRINTGRAF_NUM "_pedenum", 2, 2222

CHIUDISTAMP

Risultato di stampa:

STAMPA SPRIX DA APP	
Stringa Intestazione 1	1234
Stringa Intestazione 2	5678
Corpo 1 Dettaglio 1 Riga 1	111
Corpo 1 Dettaglio 1 Riga 2	112
Corpo 2 Dettaglio 1 Riga 1	211
Corpo 2 Dettaglio 1 Riga 2	212
Corpo 3 Dettaglio 1 Riga 1	311
Corpo 3 Dettaglio 1 Riga 2	312
Corpo 4 Dettaglio 1 Riga 1	411
Corpo 4 Dettaglio 1 Riga 2	412
Corpo 5 Dettaglio 1 Riga 1	511
Corpo 5 Dettaglio 1 Riga 2	512
Corpo 1 Riga 1	101
Corpo 1 Riga 2	102
Corpo 1 Dettaglio 2 Riga 1	121
Corpo 1 Dettaglio 2 Riga 2	122
Corpo 2 Riga 1	201
Corpo 2 Riga 2	202
Corpo 2 Dettaglio 2 Riga 1	221
Corpo 2 Dettaglio 2 Riga 2	222
Corpo 3 Riga 1	301
Corpo 3 Riga 2	302
Corpo 3 Dettaglio 2 Riga 1	321
Corpo 3 Dettaglio 2 Riga 2	322
Corpo 4 Riga 1	401
Corpo 4 Riga 2	402
Corpo 4 Dettaglio 2 Riga 1	421
Corpo 4 Dettaglio 2 Riga 2	422
Corpo 5 Riga 1	501
Corpo 5 Riga 2	502
Corpo 5 Dettaglio 2 Riga 1	521
Corpo 5 Dettaglio 2 Riga 2	522
Corpo 1 Dettaglio 3 Riga 1	131
Corpo 1 Dettaglio 3 Riga 2	132
Corpo 2 Dettaglio 3 Riga 1	231
Corpo 2 Dettaglio 3 Riga 2	232
Corpo 3 Dettaglio 3 Riga 1	331
Corpo 3 Dettaglio 3 Riga 2	332
Corpo 4 Dettaglio 3 Riga 1	431
Corpo 4 Dettaglio 3 Riga 2	432
Corpo 5 Dettaglio 3 Riga 1	531
Corpo 5 Dettaglio 3 Riga 2	532
Stringa Piede 1	1111
Stringa Piede 2	2222

Esempio: APRISTAMP_FILE di tipo CSV

Come nel precedente esempio, predisponete un modulo grafico [modstspx\$] (come richiesto dal parametro tipofile\$ = "C") e nel corpo del formato inserite un campo di stampa _corpostr(1). Chiamate il modulo con codice STPSPX1 ed eseguite il seguente Sprix.

APRISTAMP_FILE "C", "<C:\Temp\file.csv",80,"STPSPX1"

LPRINTGRAF_STR "_corpostr", 1, "Stringa corpo 1;"

LPRINTGRAF_SEZ 0,0

CHIUDISTAMP

Esempio: Unione di file PDF

DIM DIR\$(2)

DIR\$(1)="<C:\Temp\fileA.pdf"

DIR\$(2)="<C:\Temp\fileB.pdf"

DIR_UNIONE\$="<C:\Temp\fileUNIONE.pdf"

APRISTAMP_FILE "P",DIR\$(1),220,"STPSPXA"

LPRINTGRAF_STR "_corpostr",1, "Stringa corpo modulo A"

LPRINTGRAF_SEZ 0,0

CHIUDISTAMP

APRISTAMP_FILE "P",DIR\$(2),220,"STPSPXB"

```
LPRINTGRAF_STR "_corpostr",1, "Stringa corpo modulo B"
LPRINTGRAF_SEZ 0,0
CHIUDISTAMP

UNISCIPDF DIR$( ),DIR_UNIONES$,"", "", "",0,0
VIMSG 1, _ERRFILE$
```

Esempio: stampa su modulo grafico della tabella dizionario APPALTI PUBBLICI

I dati di una tabella dizionario si possono leggere in modo tabellare (ciclotabellare) oppure con lettura diretta, quando previsto, passando le cosiddette "chiavi" della tabella. E' un concetto che viene spiegato in questo manuale nella sezione [COMANDI PER ARGOMENTO - MODULI E ARCHIVI DEL GESTIONALE - Struttura & Dizionario].

Per stampare la lista della tabella si esegue il ciclotabellare e andare direttamente in stampa (ad esempio su dispositivo PDF) sfruttando l'editor ed il ciclotabellare. Per farlo, invece, su un modulo grafico, si può realizzarne uno personalizzato (anche all'interno della stessa App, come nell'esempio che segue), ma con la conseguenza è che **il ciclotabellare non può eseguire ed inviare direttamente dati alla stampa del modulo grafico.**

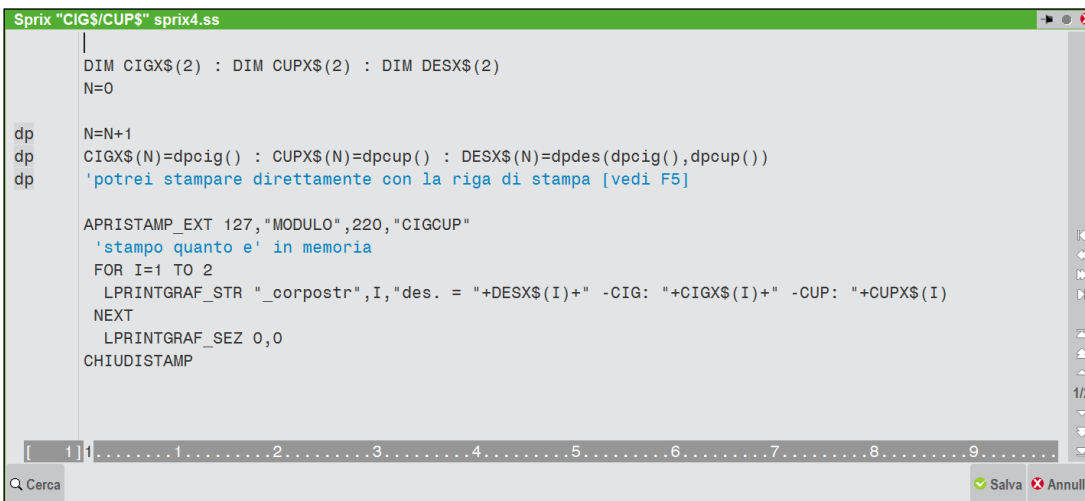
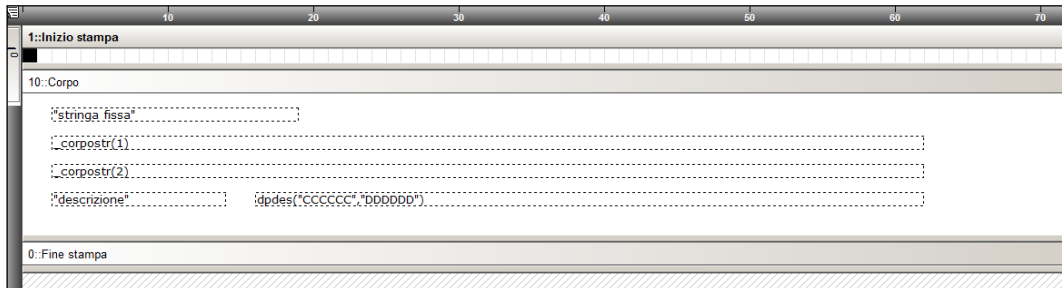
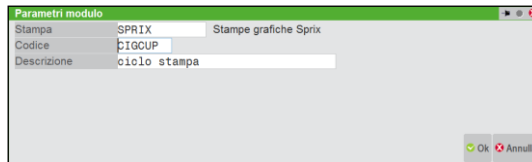
Per avviare a questo vincolo, basta caricare e memorizzare i vettori tramite il ciclotabellare e poi andare in grafica con un semplice ciclo di stampa forn next, oppure, passare direttamente le chiavi al dizionario, in questo caso dpdes(dpcig(),dpcup()).

Esempio: dati due record nella tabella appalti pubblici, stampo le descrizioni in base a CIG e CUP presenti:

1. descrizione "Appalto 1" con CIG = "AAAAA", CUP = "BBBBB"
2. descrizione "Appalto 2" con CIG = "CCCCC", CUP = "DDDDDD"

Data un App con un modulo di stampa ed uno Sprix che lo utilizza

ciclo stampa	Modulistica stampa	● Ok
CIG\$/CUP\$	Sprix	● Menu' Azienda



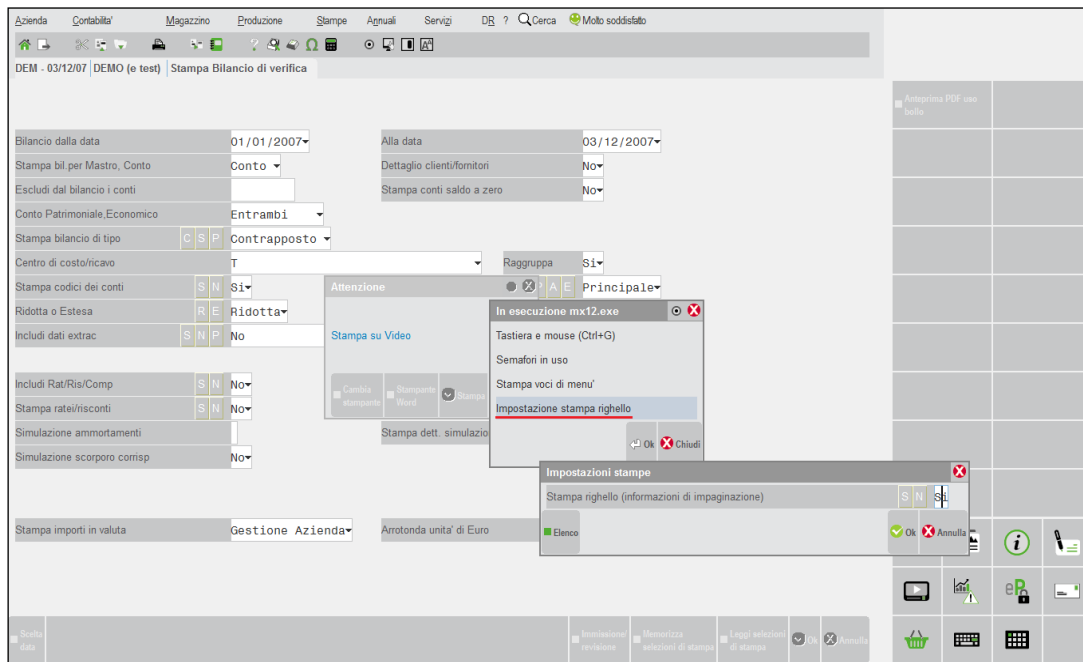
Risultato

	<p>stringa fissa</p> <p>des. = appalto 1 -CIG: AAAAA -CUP: BBBB</p> <p>des. = appalto 2 -CIG: CCCCC -CUP: DDDDD</p> <p>descrizione appalto 2</p>			
	<p>Le variabili che seguono sono per STAMPARE il MODULO del DOCUMENTO. Richiedono predisposizione di campi nella modulistica (tradizionale e/o grafica) con degli appositi corrispondenti codici di stampa (scelti dalla voce COLLAGE in elenco dei codici stampa) di massimo 14 caratteri numerici oppure 80 caratteri stringa (*), relativi a:</p> <ul style="list-style-type: none"> • Testata/Piede del modulo tradizionale (numerici XN01,..., XN20 e stringa XA01,..., XA20) e grafico (numerici _mmsxn(1),..., _mmsxn(20) e stringa _mmsxa(1),..., _mmsxa(20)); • Corpo del modulo tradizionale (numerici WN01,..., WN10 oppure WF01,..., WF20 e stringa WA01,..., WA10 oppure WS01,..., WS20) e grafico (numerici _mmswn(1),..., _mmswn(10) oppure _mmsws(1),..., _mmsws(20) e stringa _mmswa(1),..., _mmswa(10) oppure _mmswf(1),..., _mmswf(20)). Le relative variabili sono <u>bidimensionali</u> ed _NMM è il numero di riga). <p>(*) Per ottenere campi di stampa superiori ai suddetti limiti, occorre ripetere il campo di stampa (una concatenazione del medesimo campo) in cui in ciascuno di essi riportare la porzione (LEFT, MID, RIGHT) della variabile in questione.</p> <p>Notare: le Variabili Collage della <u>Modulistica tradizionale</u> non sono attive sulle righe di <u>dettaglio lotti</u>, ma solo sulla prima riga (relativa all'articolo). Sono invece attive nella sezione lotti della <u>Modulistica grafica</u>.</p> <p style="text-align: center;">Attenzione!</p> <p>Nel menu ME (emissione differita documenti) ricordarsi che l'evento ON_PUTMM viene processato solamente con l'effettiva scrittura, per esempio una trasformazione del documento, pertanto, l'evento ON_PUTMM non viene processato in caso di una ristampa! ovvero con "Emissione di prova = S" (è però attivo ON_STARTMM).</p> <p>Gli array possono essere azzerati da operazioni di lettura (GET) oppure se inseriti dopo articoli descrittivi. Il modulo documento (tradizionale non grafico) può essere anche modificato tramite altri tipi di variabili (_SST\$, _TSST\$), le quali hanno degli appositi eventi (ON_ST_?, ON_TST_?) e sono utili, soprattutto, per altri tipi di stampe (es. report, elenchi, bilanci...).</p> <p>Nota bene: il momento della valorizzazione delle Variabili Collage in Modulistica può risultare differente in base all'utilizzo del modulo "grafico" oppure "tradizionale". Se il modulo è "grafico" questo viene composto dopo che scatta l'evento in questione (per esempio ON_STARTSTPEMAIL), pertanto le variabili sono già valorizzate. Se, invece, il modulo è "tradizionale" questo viene composto prima che scatti l'evento in questione (per esempio ON_STARTSTPEMAIL), pertanto le variabili restano vuote (in alternativa occorre usare i consueti eventi ON_STARTMM: oppure ON_PUTMM:)</p>	S/C/Cd		
<p>_MMSXA\$()</p>	<p>Par: (1-20). Variabile stringa monodimensionale di testa/piede del modulo documento collegata al campo di testa/piede della modulistica tradizionale (XA01-20) oppure della modulistica grafica (_mmsxa). In Collage non utilizzare eventi ON_ST_ oppure ON_TST_, ma bensì altri eventi come ON_STARTMM, ON_PUTMM, ON_612_? Ecc.</p> <p style="text-align: center;"><small>Esempi</small></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"> Sprix (variabile modulo XA01) GETMM "OC3": VMSG 1, _ERRMM\$ _MMSXA\$(1)="modifica" : PUTMM 4 : END </td> <td style="padding: 2px;"> Collage menu MD (variabile modulo XA01) ON_612_ALL: ON_PUTMM: _MMSXA\$(1)="modifica" : END </td> </tr> </table> <p>Nota bene: per l'esecuzione di emissione differita, leggere le suddette avvertenze, indicate anche nei comandi GETATTMM e ON_PUTMM.</p>	Sprix (variabile modulo XA01) GETMM "OC3": VMSG 1, _ERRMM\$ _MMSXA\$(1)="modifica" : PUTMM 4 : END	Collage menu MD (variabile modulo XA01) ON_612_ALL: ON_PUTMM: _MMSXA\$(1)="modifica" : END	S/C/Cd
Sprix (variabile modulo XA01) GETMM "OC3": VMSG 1, _ERRMM\$ _MMSXA\$(1)="modifica" : PUTMM 4 : END	Collage menu MD (variabile modulo XA01) ON_612_ALL: ON_PUTMM: _MMSXA\$(1)="modifica" : END			
<p>_MMSXN()</p>	<p>Par: (1-20). Variabile numerica monodimensionale di testa/piede del modulo documento collegata al campo di testa/piede della modulistica tradizionale (XN01-20) oppure della modulistica grafica (_mmsxn). In Collage non utilizzare eventi ON_ST_ oppure ON_TST_, ma bensì altri eventi come ON_STARTMM, ON_PUTMM, ON_612_? Ecc.</p> <p>Nota bene: per l'esecuzione di emissione differita, leggere le suddette avvertenze, indicate anche nei comandi GETATTMM e ON_PUTMM.</p>	S/C/Cd		
<p>_MMSWA\$()</p>	<p>Par: (R,1-10). Variabile stringa bidimensionale di riga del corpo del modulo documento collegata al campo del corpo di modulistica tradizionale (WA01-10) oppure di modulistica grafica (_mmswa).</p> <p style="text-align: center;"><small>Esempio: stabiliti nel modulo: XN01 di testata e WA01 di corpo</small></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"> Sprix _MMMODSTP\$="2" : _MMSXN(1)=33 _MMSWA\$(1,1)="new" : PUTMM 4 : END </td> <td style="padding: 2px;"> Collage ON_PUTMM: _MMSXN(1)=33: _MMSWA\$(1,1)="new" : END </td> </tr> </table> <p>Nota bene: per l'esecuzione di emissione differita, leggere le suddette avvertenze, indicate anche nei comandi GETATTMM e ON_PUTMM.</p>	Sprix _MMMODSTP\$="2" : _MMSXN(1)=33 _MMSWA\$(1,1)="new" : PUTMM 4 : END	Collage ON_PUTMM: _MMSXN(1)=33: _MMSWA\$(1,1)="new" : END	S/C/Cd
Sprix _MMMODSTP\$="2" : _MMSXN(1)=33 _MMSWA\$(1,1)="new" : PUTMM 4 : END	Collage ON_PUTMM: _MMSXN(1)=33: _MMSWA\$(1,1)="new" : END			
<p>_MMSWN()</p>	<p>Par: (R,1-10). Variabile numerica bidimensionale di riga del corpo del modulo</p>	S/C/Cd		

	<p>documento collegata al campo del corpo di modulistica tradizionale (WN01-10) oppure della modulistica grafica (_mmswn).</p> <p>Nota bene: per l'esecuzione di emissione differita, leggere le suddette avvertenze, indicate anche nei comandi GETATTMM e ON_PUTMM.</p>	
_MMSWS\$()	<p>Par: (R,1-20). Variabile stringa bidimensionale di riga del corpo del modulo documento collegata al campo del corpo di modulistica tradizionale (WS01-20) oppure della modulistica grafica (_mmsws).</p> <p>Nota bene: per l'esecuzione di emissione differita, leggere le suddette avvertenze, indicate anche nei comandi GETATTMM e ON_PUTMM.</p>	S/C/Cd
_MMSWF()	<p>Par: (R,1-20). Variabile numerica bidimensionale di riga del corpo del modulo documento collegata al campo del corpo di modulistica tradizionale (WF01-20) oppure della modulistica grafica (_mmswf).</p> <p>Nota bene: per l'esecuzione di emissione differita, leggere le suddette avvertenze, indicate anche nei comandi GETATTMM e ON_PUTMM.</p>	S/C/Cd
_MMRICSTP	Variabile di ristampa	

Etichette delle Stampe in Collage

Per costruire l'etichetta di una stampa premere [Ctrl+G] dopo la selezione del dispositivo di stampa (preferibilmente PDF), poi selezionare la "Impostazione stampa righella" per ottenere le informazioni di impaginazione. Es: menu SCBB (Stampa Contabile Bilancio Verifica).



```

00001 Codice stampa          : BILANCIO
00002 Altezza modulo         (righe): 135
00003 Larghezza modulo    (colonne): 157
00004 Larghezza riga      (colonne): 131
00005
00006 1      10      20      30      40      50      60      70      80      90      100     110     120     13
00007 -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
00008 INT1 03/12/2007          SITUAZIONE PATRIMONIALE dal 01/01/07 al 03/12/07          Pag. 1
00009 INT2 [DEM] DEMO (e test)
00010 INT3
00011
00012 A T T I V I T A'          EUR P A S S I V I T A'          EUR
00013 RIGA 104      IMMOBILIZZAZIONI MATERIALI          1.902.226,44      101      CREDITI V/SOCI P/VERSAMENTI ANCORA          600.000,00
00014 RIGA 104.00001 TERRENI          166,67      101.00001 SOCI C/SOTTOSCRIZIONE          600.000,00
00015 RIGA 104.00002 FABBRICATI CIVILI          1.869.193,10
00016 RIGA 104.00004 COSTRUZIONI LEGGERE          1.000,00      114      FONDO AMM.TO IMMOBILIZ. MATERIALI          119.916,65
00017 RIGA 104.03001 MOBILI E MACCHINE ORDIN.D'UFFICIO          1.866,67      114.00002 F/AMM. FABBRICATI CIVILI          76.816,65
00018 RIGA 104.03003 AUTOCARRI/AUTOVETTURE          30.000,00      114.01003 F/AMM. MACCHINARI          4.600,00
00019 RIGA 201      DENARO - VALORI IN CASSA - ASSEGNI          5.947.494,49      114.03001 F/AMM. MOBILI E MACCH.ORD.D'UFFICIO          1.200,00
00020 RIGA 201.00001 CASSA          5.947.494,49      114.03003 F/AMM. AUTOCARRI/AUTOVETTURE          37.300,00
00021 RIGA
00022 RIGA 202      BANCHE          200.000,00      208      ALTRI CREDITI VS CLIENTI          2.000,00
00023 RIGA 202.00001 CONTO CORRENTE X/Y PRESSO BNL          200.000,00      208.00005 EFFETTI INSOLUTI E PROTESTATI          2.000,00
00024 RIGA
00025 RIGA 208      ALTRI CREDITI VS CLIENTI          11.693.376,26      309      UTILE (PERDITA)DELL'ESERCIZIO          674.739,41
00026 RIGA 208.00001 EFFETTI IN PORTAFOGLIO          4.000,00      309.00001 RISULTATO D'ESERCIZIO          674.739,41
00027 RIGA 208.00002 EFFETTI ALLO SCONTO          11.689.376,26      403      DEBITI V/BANCHE          3.945,00
00028 RIGA
00029 RIGA 212      CREDITI TRIBUTARI E IMPOSTE ANTICIP.          707.171,45      403.00001 BANCA C/MUTUI IPOTECARI          3.000,00
00030 RIGA 212.00001 IVA SU ACQUISTI          707.171,45      403.00004 BANCA C/RICEVUTE S.B.F.          945,00
00031 RIGA
00032 RIGA 222      CREDITI VS ALTRI-CREDITI VARI          31.060,03      415      DEB.TRIB.E V/ISTIT.DI PREVID.SOCIALE          4.195.549,91
00033 RIGA 222.00004 CREDITI PER EMISSIONE NOTULE          31.060,03      415.00002 ERARIO C/IVA          2.598,45
00034 RIGA
00035 RIGA 301      CAPITALE          300.000,00      415.00003 IVA SU VENDITE          4.188.691,86
00036 RIGA 301.00001 CAPITALE SOCIALE          300.000,00      415.00006 ERARIO C/RITENUTE EFFETTUATE          3.796,08
00037 RIGA
00038 RIGA 403      DEBITI V/BANCHE          21.785,41      415.00008 IVA A ESIGIBILITA' DIFF. SU VENDITE          463,52
00039 RIGA 403.00005 BANCA C/ANTICIPI SU FATTURE          21.785,41      417      ALTRI DEBITI          2.000,00
00040 RIGA
00041 RIGA 409      ALTRI DEBITI VERSO FORNITORI          1.750,00      417.00004 DEBITI PER CAUZIONI          2.000,00
00042 RIGA 409.00001 FATTURE DA RICEVERE          1.750,00      601      DEBITI VERSO FORNITORI          4.818.122,06
00043 RIGA
00044 RIGA 501      CREDITI VERSO CLIENTI ITALIA          16.046.004,50      603      DEBITI VERSO FORNITORI CON R.D.A.          14.200,00
00045 RIGA
00046 RIGA 502      CREDITI VERSO CLIENTI PAESI U.E.          1.200,00      605      DEBITI VERSO AGENTI          1.000,00
00047 RIGA
00048 RIGA 503      CREDITI VERSO CLIENTI EXTRA U.E.          6.859,96
00049 RIGA
00050 RIGA
00051 TOTALE A T T I V I T A'          36.858.928,54      TOTALE P A S S I V I T A'          10.431.473,03
00052 TRAT
00053 RISESE
00054 TRAT          |Risul. d'esercizio provvisorio anno precedente          1.306.312,55
    
```

Gli elementi per costruire l'etichetta sono "Codice stampa" (CODSTAMPA) e "Codice riga" (CODRIGA); questo metodo vale anche per le stampe fiscali grafiche non definitive (se definitive le etichette non sono attive) e vale anche per la stampa del modulo del documento, dove il "Codice stampa" è (MODULO) e i "Codici riga" sono (TESnn...), (CORnn...), (PIEnn...).

Eccetto l'etichetta ON_STP_DOC: per le stampe magazzino personalizzate "Movimenti Magazzino" e "Documenti Magazzino", le stampe che possiedono etichette sono solamente quelle disponibili in formato standard (cioè quelle non personalizzate).

```

00001 Codice stampa          : MODULO
00002 Altezza modulo         (righe): 135
00003 Larghezza modulo    (colonne): 157
00004 Larghezza riga      (colonne): 131
00005
00006 1      10      20      30      40      50      60      70      80      90      100     110     120
00007 -----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
00008 TES01
00009 TES02
00010 TES03
00011 TES04          DEMO (e test)
00012 TES05
00013 TES06          P.Iva:
00014 TES07          Tel.
00015 TES08
00016 TES09
00017 TES10
00018 TES11
00019 TES12
00020 TES13          ORDINE CLIENTE          11473 /          19/01/2007          1
00021 TES14
00022 TES15
00023 TES16          501.00055          0
00024 TES17
00025 TES18
00026 TES19
00027 TES20          ESP          322174,70€
00028
00029
00030
00031
00032
00033
00034
00035
00036
00037
00038
00039
00040
00041
00042
00043
00044
00045
00046 COR01          ART1          art1          nr          1,00          4.852,92
00047 COR00
00048 COR00
00049 COR00
00050 COR00
00051 COR00
00052 COR00
00053 COR00
00054
00055
00056
00057
00058
00059
00060
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070
00071
00072
00073
00074
00075
00076
00077
00078
00079
00080
00081
00082
00083
00084
00085
00086
00087
00088
00089
00090
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
    
```

La creazione di un'etichetta con il CODSTAMPA esegue la routine solamente in quella stampa, e aggiungendo il CODRIGA esegue la routine solamente in quella riga; se uno dei due (oppure entrambi) sono sostituiti dal comando (ALL) l'esecuzione è generalizzata su tutte le stampe e/o tutte le righe. Il comando ALL è utile per stampe che non prevedono CODSTAMPA / CODRIGA, i quali sono ricercati per primi indifferentemente dalla loro posizione all'interno del collage; non tutte le righe (anche con codice) sono intercettabili, specialmente se sono di testata e/o piede, poichè sono riservate ad uso esclusivo ed in genere hanno il prefisso '_' (come dimostrato nell'immagine precedente relativa alla stampa del bilancio); generalmente, quindi, le righe utilizzabili appartengono al corpo della stampa.

Attenzione: per una medesima stampa le etichette potrebbero diversificarsi a seconda dei parametri di selezione. Verificare sempre con l'attivazione della funzione "righello" ([Ctrl+G] prima della conferma di stampa).

Vediamo alcuni esempi.

L'etichetta relativa al nome della Stampa di Inventario cambia in base al tipo di ordinamento selezionato:

- ordinamento per codice Primario, "P", il nome della stampa corrispondente è "INVART"
- ordinamento per codice Secondario, "S", il nome della stampa corrispondente è "INVARTS"
- ordinamento per Descrizione dell'articolo, "D", il nome della stampa corrispondente è "INVARTD"

Nella Stampa Inventario (INVART) le etichette RIGINV sono attive solo se è valorizzato il campo "alla data". Qualora non fosse desiderata tale valorizzazione, l'etichetta da intercettare risulterebbe RIINVPE.

Per la Stampa dei Listini Articoli (Stampe\Magazzino\LISTINI ARTICOLI) è possibile notare i diversi "Cod_Riga".

L'attivazione di alcuni codici riga potrebbero dipendere dalle condizioni in cui vengono eseguite le relative stampe. Verificare sempre con l'attivazione della funzione "righello" ([Ctrl+G] prima della conferma di stampa). Per esempio, nella stampa dell'estratto conto normale (STESCON), se nelle condizioni commerciali della anagrafica Cli/For esiste il conto dell'agente, saranno stampate entrambe le righe INT1 e INT2, altrimenti sarà stampata solo INT2 (si aggiunge anche la nota relativamente alla riga ON_ST_STESCON_TRAT, poiché non tutti i tratteggi sono intercettati, per esempio quello della prima separazione tra testata e righe, ma solo alcuni ad esempio i tratteggi separatori dei totali); in alcune stampe l'attivazione di una stampa estesa anziché ridotta, oppure l'attivazione di determinati flag, rende disponibili alcuni CODRIGA che precedentemente non lo erano. L'evento di stampa non anticipa quello di default del gestionale e può essere attivato anche da alcune condizioni (es. anno di ingresso nel programma < al 2002 per il modulo 'SFD' tipo di scadenario, tipo dell'azienda per moduli SFG, SRA, SRV, NEC,...).

I principali eventi di stampa e relative variabili sono elencati nella seguente tabella

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
ON_STARTST_CODSTAMPA:	Evento alla partenza della stampa dopo l'apertura stampante	C/Cd
ON_ENDST_CODSTAMPA:	Evento alla fine della stampa, prima della chiusura stampante	C/Cd
ON_ST_CODSTAMPA_CODRIGA:	Evento che prima di stampare la riga su stampa rende disponibile nella variabile _SST\$ la stringa da stampare e terminata la routine, la manda in stampa. Non utilizzabile con le variabili collage di modulistica	C/Cd
_SST\$	Variabile con la riga in stampa (evento ON_ST_). Funziona con carattere a spaziatura fissa; non è attiva nel uso di variabili modulistica VTXT e VTDE poiché sono campi riservati alle stampanti di tipo W o PDF, e quindi sono già stati processati del gestionale, pertanto sono inibiti gli eventi ON_ST_? poiché si è fuori dal controllo del gestionale (per questa ragione non è utilizzabile nei moduli grafici). La variabile è globale, prestare attenzione nelle routine contemporanee ad altri Collage. Esempio: ON_ST_MODULO_COR01: _SST\$="variiriga": END; Esempio: ON_ST_MODULO_CORRIF: VIMSG 4, _SST\$: END 'riferimenti documento'; Esempio: ON_ST_BILANCIO_ALL: VIMSG 4, _SST\$: END; Esempio: ON_ST_ALL: _SST\$ = "intercettata": END; questo esempio mostra quali righe si possono intercettare in una qualsiasi stampa generica che sia senza etichette ma che sia intercettabile, per esempio la stampa del LIFO).	C/Cd
ON_TST_CODSTAMPA_CODRIGA:	Evento che prima della stampa rende disponibile nella variabile array _TSST\$(1-99) la stringa da stampare; modificando _TSST\$(1) si modifica la riga e tutti gli elementi che sono aggiunti (dal 2 compreso in poi) e sono considerate nuove righe in stampa dalla posizione corrente; azzerando tutti gli elementi (= "") la riga viene soppressa fisicamente compattando la stampa. L'aggiunta di righe in modulistica oltre il numero stabilito nel MODULO disallinea la pagina (il conteggio è successivo alla stampa). Le stampanti di tipo 'S' consentono un dfs (file definizione stampante) che annulla il saltopagina (diversamente modificarlo impostando il "comando spg:no;"). Negli eventi ON_TST_ di testata "TESn" e di piede "PIEn", l'aggiunta di righe, annulla l'evento collegato al numero di riga immesso: es. se all'evento ON_TST_MODULO_TES01: si caricassero _TSST\$(1) e _TSST\$(2), l'evento successivo intercettabile non sarebbe intuitivamente ON_TST_MODULO_TES02.; bensì ON_TST_MODULO_TES03: Non utilizzabile con le variabili collage di modulistica	C/Cd
_TSST\$()	Par: (1-99). Array con la riga in stampa, modificabile nella quantità di riga (evento ON_TST). Funziona con carattere a spaziatura fissa; non è attiva nel uso di variabili modulistica VTXT e VTDE poiché sono campi riservati alle stampanti di tipo W o PDF, e quindi sono già stati processati del gestionale, pertanto sono inibiti gli eventi ON_TST_? poiché si è fuori dal controllo del gestionale (per questa ragione non è utilizzabile nei moduli grafici). Non utilizzare in modelli prefincati, non utilizzare in stampe fiscali contabili (anche se c'è il codice stampa CODSTAMPA) data l'incompatibilità con la numerazione automatica del documento bollato. La variabile è globale prestare attenzione nelle routine contemporanee ad altri Collage. es Esempio ON_TST_MODULO_COR01: FOR X=1 TO 3 : _TSST\$(X)="piuriga": NEXT : END;	C/Cd
ON_STP_DOC:	Evento solo per stampe magazzino personalizzate "Movimenti Magazzino" e "Documenti Magazzino". Non applicabile nella stampa del Modulo Documento di Magazzino. N.B.: le variabili _MM lette da istruzione GETATTMM in questo evento sono un set parziale che include esclusivamente: per le informazioni di testata, _MMSIG\$, _MMSAZ, _MMSER, _MMNUM, _MMDAT\$, _MMCLI\$, _MMSIGRE\$, _MMNUMRES, _MMDATRES, _MMTPRIFRES\$, _MMCTCVRES\$, _MMDOCOLD, _MMRBO, _MMRSB, _MMRFB, _MMCODCONT, _MMMAG, _MMAMA, _STPMMRIG; per le informazioni di riga, _MMTPR\$, _MMART\$, _MMDES\$, _MMDESAGG\$, _MMTSD\$, _MMDEE\$, _MMNCRERIGA, _MMTST\$, _MMARR\$, _MMNCRERIGA; per le informazioni generiche, _NMM, _NTMM.	C/Cd
_STPTNO	Esclusione righe di stampa (evento ON_STP_DOC)	C/Cd
_STPMMRIG	Numero di Riga elaborata (evento ON_STP_DOC)	C/Cd
_STPGXN_STPGXA\$	variabili Stampe Magazzino Personalizzate (evento ON_STP_DOC)	C/Cd
_STPTXN_STPTXA\$	variabili Stampe Magazzino Personalizzate (evento ON_STP_DOC)	C/Cd
_STPRWN_STPRWA\$	variabili Stampe Magazzino Personalizzate (evento ON_STP_DOC)	C/Cd

esempi e casi

Esempio INTEST

```

| P=0
| GIRO: VAR$="RIGA numero" : P=P+1
| IF P=4 THEN _INTEST=0 ELSE _INTEST=1
HA | stampa carattere AUTOMATICO pag. ~P\ (NB saltipagina + _INTEST)
HN | stampa carattere NORMALE
HE | stampa carattere ESPANSO
HG | stampa carattere GRASSETTO
HG |
C | stampa carattere COMPRESSO
P |
I | stampa carattere CORSIVO
S | stampa carattere SOTTOLINEATO
F | invia questa riga ~P\ su file esterno ("c:\temp\filesprix.txt" )
| FOR R=1 TO 10
N | ~VAR$\\\\\\\\\\\\\\\\ : ~R\
|NEXT
|IF P<=5 THEN GOTO GIRO ELSE CHIUDISTAMP
|APRISTAMP 24,"" : N= NUMSTAMP : LPRINT LUNGMOD$(66)
|LPRINTIMG "c:\temp\logo.bmp",0,50,20,0,0
|LPRINTBARCODE "40235028",1,1,0,15,80,30,20
|LPRINTTXT "RIGA LPRINTTXT","Courier",10,"GIS",0,15,60,"AC C255000000"
|LPRINT INIZCO$( ) : LPRINT "Riga in COMPRESSO" : LPRINT FINECO$( )
|LPRINT INIZES$( ) : LPRINT "Riga in ESPANSO" : LPRINT FINEES$( )
|LPRINT INIZGR$( ) : LPRINT "Riga in GRASSETTO" : LPRINT FINEGR$( )
|LPRINT INIZIT$( ) : LPRINT "Riga in CORSIVO" : LPRINT FINEIT$( )
|LPRINT INIZSO$( ) : LPRINT "Riga in SOTTOLINEATO" : LPRINT FINESO$( )
|FOR I=1 TO 10
|LPRINT "prima LPRINT "; LPRINT "seconda LPRINT":
|IF I=3 THEN LPRINT ACAPO$( ) : LPRINT "Alla n." + STR$(I,0,0) + " va a capo"
|IF I=5 THEN LPRINT SALTOPAG$( )
|IF I=5 THEN PRINTH
|NEXT
|CHIUDISTAMP
|GETMM "M1/271":IF _ERRMM<>0 THEN VIMSG 4, _ERRMM$
|_MMMODSTP$="X":_MMSXN(1)=1234:_MMSWA$(1,1)=" VAR_COLLAGE":PUTMM 4
|END

```

Esempio di uno Sprix che elabora il documento di una stampa modificata con Collage

Attenzione: l'esempio è datato, pertanto potrebbe non rispondere più alle circostanze e/o implementazioni (interventi sul documento in caso di immissione o revisione, eventuali ristampe, nuovi dispositivi, fattura elettronica, ecc... ecc...). Potrebbe quindi rendersi necessario spostare l'evento di shellsprix dal ON_ENDMM all'ingresso (ON_IN_) nel primo campo dopo la conferma del documento (campo che potrebbe essere il MDTDOC, o il MDRICCOD, ecc.).

Collage	Sprix spx102
<pre> ON_ST_MODULO_TES01: _SST\$="** riga modulo modificata **" END ON_ENDMM: GETATTMM IF _ERRMM\$<>" " THEN VIMSG 1, _ERRMM\$ ARG\$= _MMSIG\$+STR\$(_MMNUM,0,0) VIMSG 1, ARG\$ SHELLSPRIX "-pspx102 -v"+ARG\$ END </pre>	<pre> GETMM _ARGCMD\$ IF _ERRMM\$<>" " THEN VIMSG 1, _ERRMM\$ _MMNOT\$(1)="Nuova nota" \ (nota aggiunta dopo) PUTMM 0 IF _ERRMM\$<>" " THEN VIMSG 1, _ERRMM\$ END </pre>

Nota: per l'utilizzo di SHELLSPRIX fare attenzione anche agli esempi su SHELLSPRIX

CASO: Ristampa modificata di un documento

Si può eseguire in vari modi:

- variabili _SST\$/TSST\$ agli eventi ON_ST_MODULO_? o ON_TST_MODULO_?;
- variabili Collage di modulistica (es. _MMSXA\$) agli eventi utili (es. ON_PUTMM);
- variabili _FTE all'evento ON_STP_XML processato dopo la chiusura del documento;
- Collage che dopo la registrazione del documento su un determinato modulo esegue uno Sprix di servizio che ristampa lo stesso documento, ma su un altro modulo in cui non sono presenti determinate informazioni. Si considerino i seguenti presupposti:

- 1) si supponga di creare un modulo di ordine "OCP" che non presenta i campi di stampa relativi ai prezzi;
- 2) si consideri che la PUTMM 4 esegue solo una stampa;
- 3) si consideri che la variabile _MMMODSTP\$ consente di scegliere il modulo di stampa ("P") che sarà associato al _MMSIG\$="OC".

Pertanto, se all'ON_PUTMM: eseguo una GETATTMM mi carico le variabili _MM della chiave del documento che passo nell'argomento "-v" alla SHELLSPRIX la quale lancia uno Sprix (spxishe.ss) in background che riceve questa chiave nella variabile globale _ARGCMD\$ in modo da fare una successiva GETMM <chiave documento> per poi sostituire _MMMODSTP\$ (ovvero la "P") e quindi poter eseguire la PUTMM 4 con la stampa del documento con il modulo "P"

COLLAGE

```

ON_STARTMM:
VIMSG 1, "startmm"
END

```

```

ON_PUTMM:
VIMSG 1, "putmm"
GETATTMM: SIG$= _MMSIG$ : SER= _MMSER : NUM= _MMNUM
ARG$=SIG$+STR$(SER,0,0)+"/"+STR$(NUM,0,0)
SHELLSPRIX "-pspxishe -v"+ARG$
END

```

```

ON_ENDMM:
VIMSG 1, "endmm"
END

```

SPRIX (spxishe.ss)

```

GETMM _ARGCMD$ : VIMSG 1, _ERRMM$
_MMODSTP$="X"
PUTMM 4
VIMSG 1, _ERRMM$
END

```

Esempio: Collage esegue uno Sprix per eseguire la ristampa del documento su un altro modulo (lo standard stampa su un modulo e Sprix lo ristampa su un altro. Richiede

configurazione dei dispositivi di stampa e definizione di modulo diversi

COLLAGE su menu 'MD' (leggo il modulo e lancio servizio Sprix di ristampa)

```
ON_PUTMM:
GETATTMM
IF_ERRMM<>0 THEN VIMSG 4,"ERRORE"
VIMSG 1,_MMMODSTP$
ARG$=_MMMODSTP$ :
ARG$=_MMSIG$ + STR$(_MMSER,0,0) + "/" + STR$(_MMNUM,0,0) + "2"
SHELLSPRIX "-psprix4" + "-" + ARG$
END
```

SPRIX: (eseguo lettura ultimo documento e modifico modulo di stampa).

```
VIMSG 4, "PARTO CON LO SPRIX"
NRDOC$=LEFT$( _ARGCMD$, LEN(_ARGCMD$)-1)
GETMM NRDOC$
IF_ERRMM<>0 THEN VIMSG 4, _ERRMM$
VIMSG 4, _MMCLI$
_MMMODSTP$=RIGHT$( _ARGCMD$,1)
PUTMM 4
VIMSG 4, _ERRMM$
END
```

Ristampa del documento e richiesta stampante

La ristampa del documento si può gestire in vari modi a seconda dei casi, con apposita procedura. Per esempio, ci sono casi in cui non si vuole dovere scegliere la stampante ad ogni ristampa, ma bensì si vuole la proposizione di default del dispositivo di stampa scelto. Esistono delle istruzioni e variabili che gestiscono stampe e dispositivi di stampa, ma non sono universalmente risolutive e dipende da ciascun singolo caso, per esempio l'istruzione APRISTAMP oppure la variabile _MMRICSTP per ristampare in fase di PUTMM (la _MMRICSTP è nata proprio per eseguire la richiesta, non per proporre la stampa di default di un determinato dispositivo di stampa) e che comunque, richiede la conferma dell'operatore. Valutare anche l'ipotesi di non indicare il modulo nell'APRISTAMP.

Ogni caso può richiedere, quindi, una determinata soluzione con gli strumenti a disposizione, anche quelli presenti in modo standard. Basti pensare alle funzionalità standard presenti nella stessa modulistica e che insieme a SPRIX possono aiutare a risolvere il problema del cambio di dispositivo di stampa e ristampa con scelta di default. In sintesi: il modulo ha un dispositivo di stampa di default (es. stampante W) e un altro modulo (copia identica del primo) può avere un'altro dispositivo (es. stampa a video su formato pdf) e può prevedere nelle sue "opzioni" la possibilità di effettuare sia un numero di ristampe sia infinite ristampe a richiesta (parametro = "S"); con SPRIX è possibile pilotare tutto questo. Per fare un semplice esempio: si hanno 2 moduli distinti di cui uno è la copia dell'altro (BCX e BCY). BCX prevede la stampa sul dispositivo n.1 (stampante di tipo W) e BCY prevede la stampa sul dispositivo n.2 (stampante pdf) e che prevede opzioni di ristampa a "S". Con il seguente SPRIX è possibile, inizialmente stampare due documenti sul dispositivo 1 (stampante W) e successivamente n. documenti sul dispositivo 2 (stampante pdf) e con la sola necessità di conferma della stampa (oppure stabilire il default del numero preciso di ristampe). Il tutto è pilotato da _MMMODSTP\$:

```
AZZVARSYS 4
FOR I=1 TO 5
GETMM "BC7"
IF I>=3 THEN _MMMODSTP$="Y" ELSE _MMMODSTP$="X"
PUTMM 2
NEXT
END
```

Nelle soluzioni adottate, occorre comunque fare attenzione alla differenza di velocità di processo/esecuzione del programma e di esecuzione della stampa (fisica, su file, a video, ecc.). Le differenze di comportamento sono ben visibili in debug e quindi, queste velocità possono richiedere determinati accorgimenti a seconda del dispositivo di stampa.

Esempio: modifica del documento con Sprix lanciato con SHELLSPRIX durante la sessione Collage

La non corretta gestione delle routine Collage in base agli eventi, nella maggior parte dei casi provoca errori del tipo "documento in uso dall'utente X sul terminale 1". Questo dipende anche dal fatto che un documento sia gestito in immissione o in revisione, oppure, con la modalità di immissione "tradizionale" oppure a "lista". Le varianti, pertanto, sono molteplici. Indichiamo qualche esempio per riuscire a modificare il record grazie allo sblocco del semaforo:

Supponendo il seguente Sprix spx110.es che esegue la modifica della nota al documento:

```
GETMM _ARGCMD$
IF_ERRMM$<>" " THEN VIMSG 1, _ERRMM$
_MMNOT$(1)="Nota aggiunta"
PUTMM 0
IF_ERRMM$<>" " THEN VIMSG 1, _ERRMM$
END
```

Il seguente Collage opera in nuova immissione del documento

```
ON_ENDMM:
GETATTMM
IF_ERRMM$<>" " THEN VIMSG 1, _ERRMM$
ARG$=_MMSIG$+STR$(_MMNUM,0,0)
VIMSG 1, ARG$
SHELLSPRIX "-pspx110 -v"+ARG$ : END
```

Il seguente Collage opera in revisione del documento (modalità di immissione "tradizionale")

```
ON_IN_MDTDOC:
IF_ARG$<>" " THEN SHELLSPRIX "-pspx110 -v"+ARG$ : ARG$=""
END
```

```
ON_ENDMM:
GETATTMM
IF_ERRMM$<>" " THEN VIMSG 1, _ERRMM$
ARG$=_MMSIG$+STR$(_MMNUM,0,0)
VIMSG 1, ARG$
END
```

Il seguente Collage opera in revisione del documento (modalità di immissione a "lista")

```
ON_IN_MDRICDOC: IF_ARG$<>" " THEN SHELLSPRIX "-pspx110 -v"+ARG$ : ARG$="" : END
ON_IN_MDTDOC: IF_ARG$<>" " THEN SHELLSPRIX "-pspx110 -v"+ARG$ : ARG$="" : END
ON_ENDMM:
GETATTMM : IF_ERRMM$<>" " THEN VIMSG 1, _ERRMM$
ARG$=_MMSIG$+STR$(_MMNUM,0,0) : VIMSG 1, ARG$
END
```

Altra modalità a lista.

```
ON_IN_MDRICDOC:
IF_ARG$<>" " THEN SHELLSPRIX "-pspx123 -v"+ARG$
ARG$=""
END
```

```
ON_IN_MDTDOC:
IF_ARG$<>" " THEN SHELLSPRIX "-pspx123 -v"+ARG$
```



```

ARG$=""
END

ON_PUTMM:
GETATMM
ARG$=_MMCLI$+LEFT$(_MMSIG$,2)+STR$(_MMNUM,0,0)
END

```

Esempi LPRINTBARCODE

LPRINTBARCODE arcod(),1,0,0,15,80,30,20
LPRINTBARCODE aralt(),0,1,0,20,20+40*COD,30,20 (<y> variabile in base a...il parametro "y" indica il margine superiore da cui stampare il codice stesso. Tale margine deve variare da codice a codice in funzione dell'altezza dei codici precedenti, dei loro margini superiori e delle impostazioni di pagina della stampante.

LPRINTBARCODE CHR\$(134)+"104512XA"+CHR\$(134)+"17050101",4,1,0,0,0,50,40 (cod. EAN/UCC 128 con carattere FNC1 prima di ogni AI, nell'esempio il codice "4512XA" preceduto da AI"10" e data "050101" preceduta da AI "17")

Le funzioni di stampa posizionano per riga e in orizzontale, l'orientamento di stampa in verticale può avvenire con il processo di stampa o appositi strumenti grafici a cui sprix può inviare i dati da elaborare nel modo desiderato. Non sono quindi consentiti orientamenti misti: verticali in un intervallo di righe posizionate in orizzontale (es. barcode in verticale affiancato da stringhe descrittive stampate in orizzontale a fianco del barcode).

Esempio ON_STP_DOC: leggendo i documenti, nel caso in cui la sigla sia "FT" imposta un valore nelle variabili dichiarate, per le "BD" esclude la riga elaborata dalla stampa:

Attenzione, l'esempio risulta obsoleto e non più utilizzato in quanto la variabile _STPTXA\$() risulta obsoleta/non utilizzabile

```

ON_STP_DOC:
_STPTNO = 0
GETATMM
IF _MMSIG$ = "FT" THEN _STPTXA$(1) = "Valore personalizzato"
FOR I = 1 TO _NMM
IF _MMSIG$ = "FT" THEN _STPRXA$(I,1) = "Valore personalizzato"
NEXT
IF _MMSIG$ = "BD" THEN _STPTNO = 1
END

```

Casi: SALTOPAGINA

SALTOPAG\$() esegue saltopagina solo se si stampa almeno un carattere oppure si va "a capo" di una riga. Non funziona con LPRINTTXT (che stampa in grafica e non influenza i contatori interni di stampa), bensì funziona con LPRINT (che ad ogni uso aumenta il contatore dei caratteri nella riga e, in assenza del carattere ";", genera un "a capo" che aumenta il contatore delle righe nella pagina). Se LPRINT SALTOPAG\$() non possiede il ";" alla fine, lei stessa genera un a capo regolare, con avanzamento del contatore interno che provoca, al secondo passaggio, il salto pagina giusto.

NOTARE: se la prima pagina non è esegue il saltopagina (mentre le successive lo eseguono) è possibile che il primo GOSUB alla routine di stampa LPRINTTXT non esegui il primo avanzamento dei contatori, pertanto, inserire un LPRINT "" subito dopo la LPRINTTXT al solo scopo di "muovere" i contatori. Esempio:

```

STAMPA:
LPRINTTXT STRING$, FONT$, DIM, AT$, 0, X, Y, AL$
LPRINT ""
RETURN

```

Nella stampa PDF il saltopagina è influenzato da una dimensione fissa del carattere nella configurazione della stampante e che imposta l'area di stampa del formato con il calcolo delle righe per il saltopagina. Esempio.

```

RIGA = 1
FOR I=1 TO 300
A$=STR$(I,0,0)
LPRINTTXT A$,"Courier",11,"",1,1,RIGA,"A$"
RIGA = RIGA + 1
IF I MOD 83 = 0 THEN LPRINTTXT A$,"Courier",11,"",1,1,RIGA,"A$": LPRINT "":
LPRINT SALTOPAG$(): RIGA=1
NEXT

```

Nella stampa di etichette articoli tramite ciclotabellare, alla fine di ogni stampa il saltopagina è impostabile in vari modi: al termine dello sprix; con CHIUDISTAMP; con SALTOPAG\$(); con riga di stampa di tipo "P"; impostando la lunghezza del modulo in base alle dimensioni dell'etichetta. Considerare, inoltre, l'eventuale impostazione della lunghezza del modulo stabilita nella videata "Parametri di esecuzione" (tasto funzione [F6]) in relazione alle righe di stampa di tipo "P" stabilite nell'editor. Anche l'array _TSST\$() può influenzare il saltopagina in seguito ad eventuali disallineamenti in base ai conteggi di riga. Esempio:

Se si dovesse modificare una stampa, ad esempio agendo sul saltopagina, si potrebbe decidere di utilizzare anche l'array _TSST\$() se la stampa lo consente. Tale array infatti permette di personalizzare la riga che deve essere inviata alla stampante (_TSST\$(1)) e permette di inserire eventuali righe aggiuntive prima che venga stampata la prossima riga di stampa (per un totale di 99 righe al massimo). Date le proprietà di tale array, date le impostazioni nella configurazione della stampante e dato il numero di record che si desidera stampare all'interno di una pagina, è possibile creare un Collage come quello che segue:

```

ON_TST_LISTART_INT1:
DIM ART$( )
END
ON_TST_LISTART_ALL:
NUM=NUM+1
IF NUM<=20 THEN END
ART$(NUM-20)=LEFT$( _TSST$(1),12)
IF NUM=30 THEN GOTO NUOVA_PAG
_TSST$(1)=" "
END
NUOVA_PAG:
FOR I=21 TO 29
_TSST$(I-20)=ART$(I-20)
NEXT
NUM=1
END

```

Dove 20 sono le righe per pagina che si desiderano stampare, 29 sono le righe che la stampante stamperebbe di default, NUM è un contatore che indica quale riga della pagina si sta stampando.

Casi: LPRINTBARCODE

Di seguito vengono elencate alcune problematiche legate all'utilizzo dell'istruzione LPRINTBARCODE e le loro relative soluzioni.

Difficoltà nella lettura del codice a barre con il lettore ottico: se si incontrano difficoltà con LPRINTBARCODE nella lettura del codice a barre con il lettore ottico, occorre controllare le dimensioni dell'etichetta (proporzionalmente alle dimensioni stabilite, occorre un rapporto di 2 di altezza contro 3 di lunghezza) e la correttezza del check digit (ultimo carattere di controllo).

LPRINTBARCODE "1236",0,0,0,15,5,21,18
LPRINTBARCODE "1236",1,0,0,15,5,21,18

LPRINTBARCODE "12341238",0,0,0,15,5,21,18
LPRINTBARCODE "12341238",1,0,0,15,5,21,18

Se il codice a barre della LPRINTBARCODE risulta fuori standard, occorre aggiornare la versione.

Scarsa risoluzione del codice a barre: dopo l'eventuale aggiornamento di un sistema operativo, può verificarsi una scarsa risoluzione del codice a barre della LPRINTBARCODE. Verificare la corretta configurazione della stampante di tipo W (ad es. deve poter stampare una videata anche con Shift+F2) ed utilizzare il driver appropriato e certificato per il sistema operativo in uso. Se la stampante è di tipo termico, provare l'invio diretto di stringa di caratteri ASCII.

Nel caso in cui l'illeggibilità del barcode fosse imputabile ad un non corretto STRETCH da parte del driver (con il risultato di un deterioramento dell'immagine), per ovviare al problema si rende necessario, qualora non fosse possibile influire sul driver, creare in anticipo l'immagine in formato bmp, contenente solo il codice a barre: si effettua utilizzando la LPRINTBARCODE su stampante di tipo file in formato immagine, utilizzando all'occorrenza, qualora vi fosse un problema legato al rischio di sovrapposizione dei file se fossero generati automaticamente, il saltopagina -riga di tipo "P"- con ciclo FOR NEXT). NB: si rende comunque necessario impostare le dimensioni in millimetri della funzione LPRINTBARCODE uguali a quelli che vengono stabiliti nella configurazione della stampante di tipo file immagine (es. 150 mm x 100 mm). Tutto ciò, al fine di ottenere un'immagine bmp che contenga esattamente il codice a barre. Successivamente si rende necessario (per la generazione delle etichette barcode), eseguire la funzione sprix LPRINTIMG su tali file, avendo l'accortezza di impostare gli ultimi due parametri a zero, il che significa, che la stampa avviene senza alcuna manipolazione (leggasi stretch) rispettando le dimensioni originali. Se la stampa finale non risultasse delle dimensioni desiderate, occorrerebbe ripetere l'operazione procedendo per tentativi, variando le dimensioni in millimetri della LPRINTBARCODE.

Questioni relative alla tipologia di driver, sistema operativo, protocollo, problemi della stampante in genere, vanno affrontati con il fornitore.

Caso: Algoritmo di generazione del carattere di controllo del codice EAN

Qualora si rendesse necessario l'utilizzo dei codici a barre è indispensabile sapere che Passepartout ha già al suo interno la funzionalità che permette di generare il carattere di controllo (tasto F2), tuttavia, se per un motivo qualunque si rendesse necessario la determinazione (ad esempio tramite uno Sprix) di tale carattere, allora è bene conoscere l'algoritmo che lo genera.

Il carattere più a destra assume la funzione di check digit (cifra di controllo), ed è calcolato sulla base delle restanti cifre con l'algoritmo seguente (le posizioni dei caratteri sono numerate da destra verso sinistra):

- 1) Partendo dalla posizione 2 sommare i valori dei caratteri in posizione pari.
- 2) Moltiplicare per 3 il risultato dell'operazione 1.
- 3) Partendo dalla posizione 3 sommare i valori dei caratteri in posizione dispari.
- 4) Sommare i risultati delle operazioni 2 e 3.
- 5) Il check digit è il più piccolo numero che sommato al risultato dell'operazione 4 da un numero multiplo di 10.

Esempio di codice: 4 2 7 6 2 2 1 3 5 7 4 6 C
Posizione del carattere: 13 12 11 10 9 8 7 6 5 4 3 2 1

Operazione 1: 2 + 6 + 2 + 3 + 7 + 6 = 26
Operazione 2: 26 * 3 = 78
Operazione 3: 4 + 7 + 2 + 1 + 5 + 4 = 23
Operazione 4: 78 + 23 = 101
Operazione 5: 110 - 101 = 9 (cifra di controllo)

Esempio di codice: 1 2 3 4 5 6 7 C
Posizione del carattere: 13 12 11 10 9 8 7 6 5 4 3 2 1

Operazione 1: 1 + 3 + 5 + 7 = 16
Operazione 2: 16 * 3 = 48
Operazione 3: 2 + 4 + 6 = 12
Operazione 4: 48 + 12 = 60
Operazione 5: 60 - 60 = 0 (cifra di controllo)

Caso: LPRINTBARCODE - Specifiche codice UCC/EAN 128 - ANSI/AIM 128 - USS Code 128 - GS1-128

UCC/EAN 128
Il codice UCC/EAN-128 è, dal punto di vista tecnico, un normale CODE128. Quello che cambia è il significato che assumono i numeri e le lettere impiegate. Per esempio se si dovesse inserire la data di scadenza 01/01/2005 in un normale CODE128 si potrebbe creare un codice di questo tipo:



oppure:
La stessa cosa nel formato UCC/EAN-128 invece deve sottostare ad alcune regole. Le date infatti sono tutte nel formato AAMGGG e devono essere precedute da un cosiddetto AI (Application Identifier), cioè un numero che specifica il tipo di dato che segue che, nel caso di date di scadenza è il "17".

Ogni AI deve essere sempre preceduto a sua volta da un carattere di controllo chiamato FNCl. Tale carattere di controllo non sarà mai visibile nella stringa sottostante al codice, ma le sue barre sì. Questo serve per permettere ad eventuali software che devono eseguire degli automatismi in merito al codice letto, di capire dove finisce un dato e ne comincia un altro. Nel nostro esempio la stessa data in formato UCC/EAN-128 diventa:



In un EAN/UCC-128 è possibile inserire più di una informazione, quindi oltre alle date potremmo inserire per esempio il NUMERO DI LOTTO "4512XA" che ha come AI il numero "10". Il nostro esempio diventa allora:



e così via.

In Sprix per gestire questo FNC1 che deve precedere ogni AI, si deve utilizzare il CHR\$(134) inserendolo opportunamente nella stringa da stampare.

Nel nostro esempio l'istruzione che genera il codice a barre sovrastante è:
LPRINTBARCODE CHR\$(134)+"104512XA"+CHR\$(134)+"17050101",4,1,0,0,0,50,40

AGGIORNAMENTO CHIARIMENTO

Il barcode 128 che siamo in grado di stampare viene chiamato ufficialmente in diversi modi.

Un elenco esemplificativo:

ANSI/AIM 128,
ANSI/AIM Code 128,
USS Code 128,
Uniform Symbology Specification Code 128
GSI-128

e si riferiscono tutti alla stessa cosa.

GSI-128

è uno standard di applicazione implementata dalla GSI usando le specifiche del codice a barre Code 128.

L'eventuale argomento del Set A oppure B oppure C, si riferisce al set di caratteri che è in grado di supportare, vale a dire:

Set A = tutti i caratteri alfanumerici maiuscoli U.S. + i caratteri di punteggiatura + i caratteri di controllo (cioè i caratteri con valore ASCII da 0 a 95 incluso) + sette caratteri speciali usati come marcatori all'interno del barcode.

Set B = tutti i caratteri alfanumerici maiuscoli U.S. + i caratteri minuscoli + sette caratteri speciali usati come marcatori all'interno del barcode

Set C = solo numeri e precisamente il set di 100 coppie di cifre numeriche (cioè da 00 a 99) + tre caratteri speciali usati come marcatori all'interno del barcode. Permette di codificare lunghe sequenze di soli numeri in spazi più stretti.

la funzione LPRINTBARCODE (e quindi anche l'equivalente variabile dentro alla modulistica documenti VBAR) tramite il parametro 'Tipo' è in grado di stampare i suddetti codici a barre:

0 = CODE39 pieno;
1 = EAN8 o EAN13 pieno;
2 = CODE39 standard;
3 = EAN8 o EAN13 standard;
4 = CODE128;
5 = CODE128 compresso

Nello specifico il tipo 4 corrisponde al set A e B e il tipo 5 al set C.

Il codice 128 richiesto dalle poste **Code 128 AIM USS-128 tipo C** è quindi pienamente supportato

Caso: formattazione CODE39

Premesso che la capacità di leggere un codice a barre dipende molto della qualità del lettore e dell'etichetta, le seguenti sono indicazioni di massima da verificare poi sempre "su campo". Nell'impostazione della larghezza e lunghezza nella funzione LPRINTBARCODE si consiglia indicativamente di mantenere queste impostazioni:

Un'altezza di 10 mm

Una larghezza di 38 mm con un codice di 4 caratteri, con un incremento di 6 mm per ogni carattere aggiunto.

N.B. L'aumento dell'altezza deve ovviamente essere corrisposta dall'aumento della larghezza in modo proporzionale. Considerare ulteriori aspetti

L'eventuale asterisco non può essere usato come carattere utile di un CODE39 perchè è riservato dalla codifica come segnali di start e stop.

LPRINTBARCODE lo aggiunge automaticamente all'inizio e alla fine della stringa passata come primo parametro ed essendo questi asterischi riservati alla codifica essi non vengono riportati nella stampa del codice in chiaro nella parte inferiore. Per cui se nella stringa di partenza sono presenti degli asterischi il codice a barre finale risulta illeggibile perchè avrebbe un doppio start all'inizio e un doppio stop alla fine.

Un altro aspetto è che la dimensione del CODE39, rispetto ai classici codici a barre, necessita generalmente di misure maggiori per essere leggibile. Nel risultato finale bisogna però anche mettere in conto il rapporto tra numero di caratteri della stringa e la dimensione totale, nel senso che più caratteri ci sono e più la dimensione va allargata. Non è possibile a parità di dimensione mettere un numero infinito di caratteri. Ancora, altra cosa da considerare è la qualità della stampante. Le termiche in genere sono migliori delle laser.

Caso: LPRINTBARCODE Farmacode 32 (pharmacode)

Come trasformare un codice a barre farmaceutico Farmacode 32 (Es. A020096020) nel suo corrispondente a barcode39 ad utilizzo di LPRINTBARCODE.

La lettera A non è presente nelle barre quindi prendiamo solo le cifre.

Innanzitutto trasformare il numero da base 10 a base 32

prendiamo ad esempio il numero

020096020 (con i separatori di migliaia forse è più leggibile 20.096.020)

ora lo dividiamo per 32

$20.096.020 / 32 = 628.000,625$

calcoliamo il resto di questa divisione

$20.096.020 - (628.000 * 32) = 20$

quindi 20 è la prima cifra partendo da destra del nostro risultato. Si prosegue con lo stesso calcolo ma prendendo la parte intera della divisione precedente cioè 628.000.

$628.0000 / 32 = 19.625$

calcoliamo il resto di questa divisione

$628.0000 - (19.625 * 32) = 0$

quindi 0 è la seconda cifra partendo da destra del nostro risultato. Si prosegue con lo stesso calcolo ma prendendo la parte intera della divisione precedente cioè 19625.

$19.625 / 32 = 613,28125$

calcoliamo il resto di questa divisione

$$19.625 - (19.625 * 32) = 9$$

quindi 9 è la terza cifra partendo da destra del nostro risultato. Si prosegue con lo stesso calcolo ma prendendo la parte intera della divisione precedente cioè 613.

$$613 / 32 = 19,15625$$

calcoliamo il resto di questa divisione

$$613 - (613 * 32) = 5$$

quindi 5 è la quarta cifra partendo da destra del nostro risultato. Si prosegue con lo stesso calcolo ma prendendo la parte intera della divisione precedente cioè 19.

siccome 19 non è divisibile per 32 lo si prende tutto quanto

quindi 19 è la quinta cifra partendo da destra del nostro risultato.

abbiamo ottenuto quindi i seguenti numeri su base 32

19 5 9 0 20

che sono 5 numeri ma siccome ne vogliamo per forza 6 allora filleriamo la parte a sinistra con tanti zeri quanti sono le posizioni mancanti (cioè nel nostro caso una) e otteniamo

0 19 5 9 0 20

Ora in base alla seguente tabella trasformiamo queste 6 cifre nei loro equivalenti caratteri alfanumerici e avremo:

Carattere in base 32	Valore in base 10	Carattere in base 32	Valore in base 10
0	0	J	16
1	1	K	17
2	2	L	18
3	3	M	19
4	4	N	20
5	5	P	21
6	6	Q	22
7	7	R	23
8	8	S	24
9	9	T	25
B	10	U	26
C	11	V	27
D	12	W	28
F	13	X	29
G	14	Y	30
H	15	Z	31

0 M 5 9 0 N

Ora è sufficiente produrre un code 39 con i suddetti valori.

Es. LPRINTBARCODE "0M590N",0,1,0,0,0,50,40

Se leggiamo le barre stampate su carta con un lettore senza abilitazione al code 32 risulterà:

0M590N (cioè un classico code 39)

Ma se nel lettore abilitiamo il code32 risulterà:

020096020

Come si calcola il check digit:

Nel nostro esempio l'ultima cifra di 020.096.020 (cioè la cifra nell'estrema destra cioè 0) è già il check digit delle 8 cifre precedenti 02009602.

(questo significa quindi le cifre utili per designare il prodotto sono le prime 8 e tenendo poi conto del fatto che la prima a sinistra deve essere per forza zero vuol dire che abbiamo solo 7 cifre utili).

La cifra di controllo del codice italiano delle specialita' medicinali viene determinata attraverso le seguenti operazioni effettuate sulle altre otto cifre.

Siano :

a1 a2 a3 a4 a5 a6 a7 a8

le prime otto cifre a sinistra del codice della specialita' medicinale. Si determinano i valori dei seguenti prodotti :

$$x1 = 2*a2$$

$$x2 = 2*a4$$

$$x3 = 2*a6$$

$$x4 = 2*a8$$

Sia P la somma dei quozienti e dei resti ottenuti dividendo x1, x2, x3, x4 per 10. Sia D la somma delle cifre a1, a3, a5, a7 del codice. Si determini il valore :

$$S = P + D$$

La cifra di controllo e' il resto della divisione di S per 10.

Esempio con il nostro numero:

0 2 0 0 9 6 0 2

a1 a2 a3 a4 a5 a6 a7 a8

$$x1 = 2 * a2 = 2 * 2 = 4$$

$$x2 = 2 * a4 = 2 * 0 = 0$$

$$x3 = 2 * a6 = 2 * 6 = 12$$

$$x4 = 2 * a8 = 2 * 2 = 4$$

$$P = x1/10 + x2/10 + x3/10 + x4/10 + \text{resti}$$

$$P = 4/10 + 0/10 + 12/10 + 4/10 + rx1 + rx2 + rx3 + rx4$$

$$P = 0 + 0 + 1 + 0 + 4 + 0 + 2 + 4$$

$$P = 11$$

$$D = a1 + a3 + a5 + a7 = 0 + 0 + 9 + 0 = 9$$

$$S = P + D = 11 + 9 = 20$$

$$\text{Cifra di controllo} = S/10 = 20/10 = 2 \text{ resto } 0$$

Codice completo di check digit : 0 2 0 0 9 6 0 2 0

Come trasformare un numero base 32 in base 10

esempio:

0 19 5 9 0 20

tenendo conto della posizione dei numeri a partire dalla destra verso sinistra a ogni posizione c'è il suo moltiplicatore che sarebbe 32 elevato alla posizione occupata avremo:

Posizione 5	Posizione 4	Posizione 3	Posizione 2	Posizione 1	Posizione 0	Totale
$32^5=33554432$	$32^4=1048576$	$32^3=32768$	$32^2=1024$	$32^1=32$	1	
$(0*33554432)+$	$(19*1048576)+$	$(5*32768)+$	$(9*1024)+$	$(0*32)+$	$(20*1)+$	20096020

Esempio: LPRINTQRCODE che stampa 45 volte l'intero alfabeto maiuscolo di 26 lettere per un totale di 1170 caratteri.

```
FOR B=1 TO 5 STEP 1
FOR A=1 TO 9 STEP 1
FOR I=65 TO 90 STEP 1
_QRCODECOD$(B) = _QRCODECOD$(B) + CHR$(I)
NEXT
NEXT
NEXT
LPRINTQRCODE "",0,0,0,0,100,1
```

SEsempio: LPRINTQRCODE indirizzo web.

```
CODE$="https://forum.passepartout.net/"
LPRINTQRCODE CODE$,0,0,0,0,50,1
```

Selezione a video della stampante non intercettabile:

Il gestionale consente di utilizzare un dispositivo di stampa associato ad un modulo. Questo consente, dopo la prima emissione e stampa del documento, di ottenere un default di stampa in caso di emissione di un ulteriore documento con le stesse caratteristiche del documento precedentemente emesso. La presenza di Collage che opera, fra la ON_PUTMM e la richiesta del dispositivo di stampa (es. una PUTTAST "#505" sulla finestra di dialogo), può essere reso inoperante in caso di default di stampa, poiché non avverrebbe la richiesta della scelta del dispositivo di stampa ed il Collage non potrebbe intervenire fra i due eventi (ON_PUTMM e ON_STAMPA), poiché troppo ravvicinati e immediati fra loro. Questo dato di fatto non esclude la possibilità di soluzioni alternative, ma dipendono dalla modalità operativa dell'utente e dal codice scritto nel Collage, che potrebbe intervenire in modo diverso e su eventi diversi. Queste alternative devono tenere in considerazione gli aspetti generali relativi all'intercettabilità degli eventi (v. paragrafo Linguaggio Sprix e Collage) e sono descritte in seguito, nei prossimi casi (*).

In questo specifico caso, invece, un'ulteriore soluzione può essere quella di NON utilizzare un dispositivo di stampa di default, in modo da consentire sempre la richiesta a video della finestra di dialogo; un'altra soluzione ancora può essere l'uso di APRISTAMP dirottato su altro dispositivo; ulteriore soluzione può essere quella di spostare la condizione del Collage al di fuori dell'intervallo fra gli eventi ON_PUTMM e ON_STAMPA e al verificarsi della condizione forzare un APRISTAMP a stampa soppressa o che richieda la finestra di dialogo per la selezione del nome modulo; questo potrebbe richiedere default a "soppresso" del dispositivo di stampa del modulo in oggetto.

Scegliere una stampante con cui stampare sempre in APRISTAMP

Una soluzione consisteva nell'aprire una finestra, leggere i numeri delle stampanti nel file di configurazione pxconf.ini, riempire la finestra con questi numeri, impostare un input di selezione, memorizzare il numero selezionato in una variabile, impostare un APRISTAMP con questa variabile.

Messaggio "stampante occupata o già in uso"

La stampante ufficiale definita in configurazione non è aggirabile e APRISTAMP non può essere utilizzata per aprire un'altra stampante diversa da quella stabilita in configurazione, dopo un evento di fine stampa ON_ENDST. Questo genera il messaggio << stampante occupata o già in uso >> (se il formato documento non prevede alcun modulo/dispositivo associato, l'APRISTAMP è consentita al evento ON_PUTMM, probabilmente anche anticipando la stampa del eventuale modulo grafico). Prima della versione del 2006G esistevano alcune soluzioni che riportiamo a semplice scopo informativo, poiché nonostante non fossero sempre risolutive (specialmente in caso di evento ON_ENDST?), potevano comunque generare spunti di riflessione sui quali eseguire delle prove per trovare la soluzione al singolo caso:

1) Attendere il termine dalla stampa ufficiale sull'evento ON_ENDST? e successivamente richiedere l'APRISTAMP dirottando la stampa su un'altra stampante (diversa da quella selezionata nell'elenco definito nella configurazione) ed infine effettuare la relativa chiusura con CHIUDISTAMP (necessario, altrimenti scatta ancora l'errore). La sequenza APRISTAMP e CHIUDISTAMP è importante, altrimenti la stampa avverrebbe solo al termine dell'esecuzione con END e quindi, stampe (come ad esempio quella grafica su stampanti W) avverrebbero solo al termine dell'esecuzione dello Sprix; ciò richiede, quindi, l'inizio della sessione di stampa grafica con la funzione APRISTAMP ed il termine con CHIUDISTAMP. Ricordarsi che CHIUDISTAMP potrebbe provocare un messaggio di errore << immettere uno dei seguenti valori (S/N), ripetere l'immissione >>; esso si attiva quando si esce con ESC da una routine di Collage che, su un evento ON_612_?, esegue un APRISTAMP (es. a video con il parametro 127), genera una stampa con LPRINT ed infine esegue CHIUDISTAMP; se il campo dell'evento non è vuoto, in uscita da questo, viene visualizzato il messaggio di errore. Per normalizzare, occorre azzerare il buffer di tastiera (AZZTAST).

2) Attendere il termine dalla stampa ufficiale sull'evento ON_ENDST? e successivamente richiamare la funzione APRISTAMP con uno Sprix.

3) Intercettare l'evento successivo (es ON_START?) con l'accortezza di impostare un contatore, al fine di evitare che parta sempre e comunque al manifestarsi di questo evento (es. Contatore = 0 non partire, contatore = 1 parti).

Stampa immediata di documenti di magazzino senza premere il tasto "Invio" richiesto settando "PUTMM 2"

Occorre simulare la pressione di "Invio" tramite l'istruzione PUTTAST valorizzata subito dopo l'istruzione PUTMM. Inoltre, qualora si volesse ristampare in maniera diretta, occorre settare la variabile "_MMRICSTP" valorizzandola a "-1".

APRISTAMP dispositivo di stampa di tipo File

Dopo la scelta del dispositivo di stampa di tipo File (configurato con selezione del file "????????????") Collage non ha più eventi collegati e quindi non intercetta, pertanto, il modo per ottenere l'input automatico del nome file nella stampa su file da Collage è quello di fare precedere l'uscita in stampa (APRISTAMP) da una serie di PUTTAST con il nome file. Esempio: ON_???: PUTTAST "nomefile.txt" : APRISTAMP...

VTXT

In alcuni sistemi, ad esempio Linux, c'è sostituzione automatica del font in "Courier", qualora il font utilizzato risulti non gestito dal sistema). es. VTXT (#XA01,...)

Modulo stampato con immagini diverse in fronte retro

Il gestionale effettua la stampa di un'immagine per ciascun foglio e non ha il controllo del dispositivo (es. impostazione fronte retro della stampante); diversamente occorre ricostruire il modulo in sprix (LPRINTIMG, SALTOPAG, ecc..., per esempio, per stampare un'immagine .wmf o .bmp di sfondo si può utilizzare LPRINTIMG come prima istruzione e successivamente scrivere tutte le altre desiderate). Soluzioni alternative:

- prestampare l'immagine sul retro del modulo documento e utilizzare in solo fronte la risma (prestampata sul retro); in entrambi i casi non è necessaria la stampa fronte retro;
- agire sulla modulistica per generare un unico processo di stampa pdf, con due pagine ciascuna con una diversa immagine; si tratta di

eludere il processo di stampa realizzando due formati modulo, il primo per un determinato contenuto (una prima immagine, il contenuto del documento, ecc..), il secondo per un contenuto secondario (seconda immagine, testo fisso, ecc..); nelle opzioni dei formati il primo modulo richiama il secondo ed entrambi hanno dispositivo di stampa PDF e flag "Uso esclusivo stampante" = C. In questo modo si genera un'unica stampa pdf, da cui aprire il device della stampante (fisica) per scegliere la modalità fronte/retro. Nota: questo metodo vale solo per documenti di 1+1 pagina; sono da valutare possibilità col formato RTF, né di "Immagine pdf di sfondo" nelle dimensioni del formato.

Modalità per stampare la fincatura o una linea continua

Avvalersi di PRINTF oppure CHR\$ del carattere ASCII ripetuto con STRING\$, oppure PRINT "_"

LPRINT: lettura difficoltosa per la stampante ad aghi.

Eliminare dalle impostazioni della stampante la retinatura ed richiedere una risoluzione almeno di 360x360

LPRINT: stampa in grassetto con una determinata stampante

In base al manuale della stampante utilizzare i caratteri di ESCAPE nella funzione LPRINT.

LPRINT CHR\$: Stampa di caratteri ASCII su stampanti di tipo "W"

Utilizzando la stampante "W" (che non è una stampante ASCII) i codici passati tramite istruzione LPRINT e funzione CHR\$, sul foglio non corrispondono al carattere ASCII inviato ma al carattere associato al font utilizzato alla stampante "W". Il default è il Courier New, si può controllare la corrispondenza mediante il programma "mappa caratteri" distribuito sui sistemi operativi Microsoft Windows (con xp si trova: programmi\accessori\utilità di sistema). Qualora si voglia utilizzare questo metodo per disegnare le fincature dei documenti si può valutare se stampare la fincatura appoggiandosi alla LPRINTIMG.

Fare anche attenzione all'utilizzo di CHR\$ abbinato all'istruzione PRINT in quanto il risultato a video è condizionato da vari fattori esterni, come il terminale, il font ecc.. Per tale ragione esistono apposite funzioni per sostituirlo in determinate casistiche, come ad esempio la PRINT per disegnare correttamente la fincatura delle finestre e l'istruzione BEEP per effettuare un segnale acustico.

LPRINTTXT: Linux spool e configurazione stampanti personalizzate in Linux

Non è possibile avvalersi della LPRINTTXT in Linux spool, ma occorre avvalersi di un'alternativa con la LPRINT. Occorre ottenere i comandi della stampante (dovrebbero essere previsti nel manuale della stampante) e impostare le righe di codice Sprix avvalendosi della LPRINT e di tali comandi; la stampante eseguirà la stampa di caratteri o l'esecuzione dei comandi a seconda di ciò che si sarà stabilito. Per la configurazione di stampanti personalizzate in Linux, se non si desidera avvalersi dalla LPRINTTXT o dei trattamenti di stringa (es INZCO\$, FINECO\$, ...) è possibile effettuare impostare con LPRINT CHR\$(...)+CHR\$(...)+... ma se le stampanti fossero fra esse differenti, ognuna avrebbe le sue caratteristiche e relativo linguaggio di programmazione; ciò risulterebbe lungo e difficile, pertanto sarebbe consigliabile potere disporre di un file .dfs per ogni stampante da utilizzare in sostituzione di uno generico.

Messaggio di errore: tremote_bufferizza_msg() buffer messaggi overflow

Errore di overflow nella gestione della comunicazione tra mxdesk e il gestionale (passando per mxserver). Potrebbe accadere durante la stampa fatta da sprix, e può essere dovuto a quando nel canale di comunicazione passano tanti messaggi e il gestionale non ha il tempo per gestirli, pertanto concedere margini, con SLEEP oppure routine di counter "a vuoto".

Errore interno RegOpenKeyEx erdoc() pxstwin.c DisplayKeyData()

L'errore indica la mancanza della chiave di Windows che contiene il nome della stampante predefinita per l'utente utilizzato. Forse nella macchina, tra le stampanti installate, nessuna di esse è mai stata settata come stampante predefinita. Se il problema si verifica con una stampante client modificare la stampante predefinita del client. Se il problema si verifica con una stampante server verificare e/o modificare la stampante predefinita del server accedendo al server stesso con lo stesso utente utilizzato dal gestionale (quello inserito nel setup, ovvero MexalAdmin). Alternativamente inserire la stampante fissa anziché "Predefinita". Se si utilizza una stampante condivisa residente su un altro sistema non si può garantire la risoluzione dello stesso.

Esempio Sprix GETMODSTPGRAF:

```
N GETMODSTPGRAF 0 (Predefiniti dentro la App e Personalizzati)
GETMODSTPGRAF 0
IF _MODGRA_NUM=0 THEN END
FOR I=1 TO _MODGRA_NUM
N   _MODGRA_COD _MODGRA_COD$(I) _MODGRA_DES _MODGRA_DES$(I)
NEXT

N GETMODSTPGRAF 1 (Solo moduli Predefiniti dentro la App)
GETMODSTPGRAF 1
IF _MODGRA_NUM=0 THEN GOTO STEP 2
FOR I=1 TO _MODGRA_NUM
N   _MODGRA_COD _MODGRA_COD$(I) _MODGRA_DES _MODGRA_DES$(I)
NEXT

STEP 2:
N GETMODSTPGRAF 2 (Solo moduli Personalizzati fuori la App)
GETMODSTPGRAF 2
IF _MODGRA_NUM=0 THEN END
FOR I=1 TO _MODGRA_NUM
N   _MODGRA_COD _MODGRA_COD$(I) _MODGRA_DES _MODGRA_DES$(I)
NEXT

END
```

Caso: LUNGMOD\$

Il comando LPRINT LUNGMOD\$(numrighe) su stampante stampante di tipo "S", non imposta il numero righe determinato. Se si immette un valore nell'F6 questo prevale nei confronti di LUNGMOD\$

Motivazione....

Si può utilizzare il comando alternativo:
 Per stampanti laser pcl
 LPRINT CHR\$(27)+"&l"+"18"+"P"; (18 è il numero di righe che si desidera impostare compreso di margine)
 Per stampanti centronics standard
 LPRINT CHR\$(27)+"C"+"18"+CHR\$(13);

Quando la stampante è settata, nella configurazione delle stampanti, a NN righe, tale valore non può essere modificato con una chiamata della LUNGMOD\$, in caso sia settato il numero delle righe nell'F6 dello sprix, questo valore assume la precedenza sul valore immesso nella configurazione

Caso: combinazione modulo e stampante in Collage

Il cliente può dovere scegliere il modulo/dispositivo dopo la realizzazione del documento, ma ciò non è consentito in sede di trasformazione, poiché il campo formato è bloccato e non più accessibile. La soluzione data dalla conferma della trasformazione sul modulo "sbagliato" con richiamo del documento da ristampare sul modulo "giusto", può non bastare se l'individuazione del modulo successiva al contenuto della trasformazione è una consuetudine normale per il cliente, che quindi, non può stampare il documento trasformato su un modulo a scelta in base a varie condizioni (nazione, numero colli, vettore, ecc...).

Questo aspetto sorge perché il modulo (che comanda il dispositivo) è prestabilito in base al tipo documento e non rispetto ai controlli del cliente in generazione/trasformazione documento.

Collage può intervenire in vari modi (in trasformazione non con puttast, perché il campo mtdoc è bloccato): la shellsprx per eseguire una PUTMM 4 o APRISTAMP su modulo ricostruito (con documento solo con formato grafico o senza tradizionale formato di stampa per evitare le problematiche di cui al chiarimento n. 1660). Dato che Collage non consentirebbe una PUTMM 4 con scelta di _MMMODSTP\$, due buone soluzioni potrebbero essere:

- una APRISTAMP con scelta del modulo su cui trasferire i dati memorizzati in GETATTMM;

- opzioni del modulo di stampa che consentissero la scelta "al momento" del ulteriore n. formato documento (del tipo "?"), con num. copie ristampa/a richiesta;

Caso: ESECUZIONE TEMPORIZZATA di una stampa/email

Possono essere realizzate con comandi shell e batch. Nel gruppo di comandi DATABASE - FILE - PORTA COM - SHELL consultare l'esempio realizzato con mxrsbatch.exe nei casi "SHELLSPRIX oppure esecuzione BATCH"

Caso: più cicli di stampa in sequenza causano lo splittamento del posizionamento della stampa sulle etichette

L'esecuzione di più cicli di stampa (con APRISTAMP e CHIUDISTAMP) possono causare il disallineamento della stampa ad ogni ciclo, nonostante la configurazione del formato di stampa sia corretto (v. menu ZCS). Questo può essere dovuto al fatto che dopo lo chiudistamp Sprix non ha il potere di controllare o decidere ciò che può/deve avvenire (e come) lato dispositivo, per esempio, il riallineamento della carta in base al fatto che lo "strappo" sia avvenuto oppure no. Ovviamente, in più cicli di stampa in sequenza, lo strappo non avviene. Alcune stampanti termiche, ad esempio, hanno la possibilità di configurare setting alla chiusura della stampa (un ADJUST SET per le regolazioni fini delle posizioni iniziali di stampa, del taglio, ecc...).

Caso: stampa Sprix da aprire su applicativo Excel

Una soluzione può consistere nella generazione di un file testo con estensione ".csv" in quanto è compatibile con l'applicazione Excel, ma ci sono dei limiti in quanto Excel potrebbe interpretare le stringhe modificando il tipo dato nella cella; soluzione invece più idonea è quella di riattare le stampe personalizzate da Sprix, creando un modulo di stampa che può essere reindirizzato in qualunque tipo di stampa desiderato. Esempio: riassunto della gestione di una stampa tramite modulo grafico:

Apertura stampante

```
APRISTAMP_EXT 127, "Titolo della finestra", 220, "nome del modulo grafico"
```

tramite un ciclo for indichiamo cosa stampare

```
FOR IDX = 1 TO LOF(NFILE)
```

```
GET NFILE,IDX
```

```
LPRINTGRAF_STR "_corpostr", 1, CAMPOFILE1$
```

```
LPRINTGRAF_NUM "_corponum", 1, VAL(CAMPOFILE$)
```

tramite questi comandi impostiamo i relativi campi di stampa presente nel modulo

il numero indica l'elemento del vettore _corpostr(),_corponum() che vogliamo utilizzare ne abbiamo massimo 99

```
LPRINTGRAF_SEZ 0, 0 tramite questo comando stampiamo
```

NEXT

CHIUDISTAMP chiudiamo la stampante

ipotizzando di vole stampare un file di 1000 righe non dobbiamo usare "_corpostr", 1000, anche perché non esiste ma usiamo sempre "_corpostr", 1, in quanto ogni qualvolta si utilizza LPRINTGRAF_SEZ 0, 0 la variabile "_corpostr", 1 viene liberata per essere riutilizzata, in pratica "_corpostr", 1 rappresenta il cursore di stampa.

Questi sono gli elementi minimi per eseguire una stampa.

Logicamente deve esistere il modulo di stampa grafico.

DATABASE – FILE – PORTA COM – SHELL

Il menu dei Servizi del gestionale Passepartout già consente di eseguire trasmissioni (import/export) di archivi dati contenuti su file (formato CSV, ASCII, Import Moduli), ma per un'ulteriore personalizzazione di iterazioni con il "mondo esterno" sono disponibili comandi aggiuntivi.

- Import/Export dati da/verso supporti esterni (mirroring, file, database). Per esempio, l'import dati da un database esterno è possibile mediante collegamento ODBC (vedere driver e RDBMS supportati in <https://www.edupass.it/manuali/manualistica-mexal/manuale-prodotto?a=manuale-prodotto/nozioni-general/ambienti-operativi/rdbms-per-moduli-su-database>) e la realizzazione di Sprix con istruzioni SQL (SQL_CONNECT, SQL_BINDCOL, SQL_FETCH, ecc.). Ciò consente di accedere ai dati in lettura o scrittura del DBMS (es. Access) ed interrogarlo in modo da valorizzare le variabili di strutture archivi (_AR, _MM, ecc) consentendo, quindi, attraverso le istruzioni PUT?? di memorizzare i dati negli archivi del gestionale. Ovviamente, l'importazione è possibile solo per quei dati per cui è prevista una scrittura degli archivi.
- Gestione di seriali e porte di comunicazione (es. per registratori di cassa ECR, lettori barcode, bilance, ecc..).
- Comandi di gestione di file (es. apertura, copia, cancellazione).
- Shell locali/remote ed esecuzioni batch.
- Esecuzioni di ulteriori programmi Sprix/Collage, oppure una loro remotizzazione su altri Server.

Attenzione: in caso di applicazioni in ambienti Live di Server Farm, verificare sempre se sono consentite operazioni di connessione, mirroring, accessi a file system e tutto quanto richieda una configurazione di sistema. In caso di connessioni SQL via ODBC in ambiente Live (Server Farm) occorre considerare che in ambiente Live, lato Client sono consentiti reindirizzamenti "<c:\", per l'acquisizione di file personali (ascii, csv, jpg, ecc...), ma per il deposito lato Server sono consentiti i soli percorsi accessibili consentiti, ovvero:

PERCORSO <DATI GENERALI> "...dati\datigen\datiutente\" (abbreviabile con "!\"")
 PERCORSO <DATI AZIENDA> "...dati\datiaz\azi\datiutente\" (abbreviabile con "J\"")

Al di fuori di queste modalità, in ambiente Live l'utilizzo di software diversi dal Gestionale che richiedono configurazioni particolari (es: operazioni SQL su file Xls/Mdb via ODBC, connessioni MySQL, ecc..) non sono disponibili. Nelle installazioni Live non e' possibile accedere a database esterni tramite connessioni ODBC in quanto la parametrizzazione del DSN non e' disponibile. Nello scambio dati da/a database esterni è però utile considerare che la componente Shaker ".Net" è disponibile in ambiente Live ed essendo questi un Client esterno al Server in Live, è in grado di eseguire in modo indipendente le sue connessioni (per scrivere o interrogare database esterni) per poi eseguire istruzioni Sprix di interesse per:

- la scrittura di file e loro copia su Server in Live nei suddetti percorsi accessibili per renderli disponibili alle procedure gestionali di servizio trasferimento archivi (ascii/csv)
- la scrittura diretta negli archivi del Gestionale tramite le istruzioni PUT?

Si ricorda che la struttura dei dati del Gestionale è consultabile nei seguenti modi:

- Le Tabelle e Colonne sono definite nel documento "ElencoColonneMexalDB.Xls" allegato al chiarimento soluzione n. 30663 del servizio di assistenza EasyPass accessibile dall'Area Riservata di www.Passepartout.net. I contenuti delle colonne nel file vanno aperte ed estese. Per ulteriori approfondimenti consultare il capitolo MexalDB del Manuale Utente.
- I Tracciati CSV sono pubblicati nel chiarimento n. 54452 del supporto on-line. I Tracciati CSV sono verificabili nel menu del Gestionale [Servizi - Trasferimento Archivi - Export/Import CSV - ASCII]
- Specifiche dei campi Dizionario nel menu del Gestionale [Servizi - Personalizzazioni - Stampa Dizionario Sprix]

I dati obbligatori sono indicati a video dal Gestionale e le relazioni fra i dati sono descritte nel Manuale Utente

Attenzione! Valutare attentamente le configurazioni del Sistema. Esistono restrizioni in accesso dei percorsi Server (ambiente protetto, specie se collocato in Server Farm), che richiedono specifiche directory di lavoro o redirezioni di lavoro su Client remoto ("<C:\..") se non dotate di maggiore libertà come le installazioni locali, come accennato nell'introduzione di questo manuale. Anche l'utente, può essere sottoposto a restrizioni / autorizzazioni configurate da regole dell'Amministratore.

Consultare il MANUALE GENERALE DELL'UTENTE: nelle "NOZIONI GENERALI" i capitoli dedicati a "RDBMS", "COMPONENTE CLIENT / SERVER", "CONFIGURAZIONI UTENTI e TERMINALI"; nei "SERVIZI" le "CONFIGURAZIONI" della "AMMINISTRAZIONE UTENTI".

NOMENCLATURE DEI FILE

- "...dati\datigen\datiutente\" (abbreviabile con "!\"")
- "...dati\datiaz\azi\datiutente\" (abbreviabile con "J\"")
- "...dati\datiaz\azi\app\(\nomeapp)\nomecartella.bindati\" (abbreviabile con nomecartella\$nomefile)
- "...dati\datiaz\azi\datiapp\(\nomeapp)\datiutente\" (abbreviabile con %nomefile)

Per accedere alle cartelle dati, utilizzare la sintassi NomeCartella\$NomeFile.est (Il percorso viene risolto con "\" finale dopo il nome cartella) esempio fileapp\$immagine.jpg. Per ulteriori approfondimenti si rimanda al manuale PassBuilder.

Attenzione: l'utilizzo di caratteri speciali (esempio: ù, ü, à, ecc..) nei nomi file e/o nei percorsi Server accessibili, può impedire l'aggiornamento degli archivi dell'azienda che ne fa uso e ne rilascia il conseguente messaggio "punto di ripristino non creato". Rimuovere i caratteri e ripetere l'aggiornamento.

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
<u>_ACCFEUTE</u>	Modalità di accesso ai file utente. es.	S/C/Cd

_ERRFILE\$	Messaggio per la gestione file utente. es.	S/C/Cd
LEGGIDIR	Par: <directory\$> o <nomecartella\$>,[maschera\$]. Istruzione che legge il contenuto di una directory oppure di un nome <cartelladati\$> dell'Applicazione PassBuilder. es. Non consentita nel Livello di sviluppo (F2)	S/C/Cr/Cd/Sh
_ERRDIR\$	Messaggio di Errore su directory	S/C/Cd
_NUMDIRFILE	Numero di file nella directory	S/C/Cd
_DIRFILE\$()	Array con i nomi dei file nella directory	S/C/Cd
ESISFILE()	Par: (<nomefile\$>). Funzione che controlla l'esistenza di un file	S/C/Cr/Cd/Sh
KILL	Par: <nomefile\$>. Istruzione che cancella un file	S/C/Cr/Cd/Sh
COPYFILE	Par: <nomefile1\$>,<nomefile2\$>. Istruzione che copia un file in altro file Al termine della esecuzione del ciclotabellare l'istruzione COPYFILE dell'archivio interessato non è consentita, perchè il file risulta aperto. Questo perchè qualsiasi operazione di accesso agli archivi, sia diretto che ciclotato, apre i file (rudt.sig, arti.sig, ecc...) e non li chiude fino al termine dell'esecuzione dell'intero SPRIX, a tutela dei file stessi.	S/C/Cr/Cd/Sh
CONVPDFA	Par: <pdf\$>,<pdfa\$>,<type\$> [,<tit\$>,<aut\$>,<obj\$><app\$>,<datetime\$>]; Istruzione per convertire il file PDF nel formato PDF/A	S/C/Cr/Cd
NUMPAGPDF()	Par: (<pdf\$>). Funzione che conta il numero di pagine di file pdf. Es PG=NUMPAGPDF("file.pdf")	S/C/Cr/Cd
UNISCIPDF	Par: <elePDF\$>,<PDFgen\$>,[tit\$],[aut\$],[obj\$],[app],[DTcrea] Istruzione per unire file pdf	S/C/Cr/Cd
OPENCsv()	Par: (<NomeFile\$>,<Separatore\$>); Funzione che apre file CSV e stabilisce il separatore; es.	S/C/Cr/Cd Sm/Cm
CSVITERSTRT()	Par: (<Hnd>); Funzione di avvio dell'iteratore sul file CSV es.	S/C/Cr/Cd Sm/Cm
CSVITERNXT()	Par: (<Hnd>); Funzione di avanzamento dell'iteratore sul file CSV es.	S/C/Cr/Cd Sm/Cm
CSVROWVAL\$()	Par: (<Hnd>,<NrColonna>,<Trim\$>); Funzione di acquisizione dei valori di interesse es.	S/C/Cr/Cd Sm/Cm
CLOSECSV()	Par: (<Hnd>); Funzione che chiude il file CSV es.	S/C/Cr/Cd Sm/Cm
_ERRCSV	Codice risultato operazioni lettura file CSV es.	S/C/Cd
_ERRCSV\$	Messaggio operazione di lettura file CSV es.	S/C/Cd
FIRMAFILE()	Par: (<nomfile\$>,<nomefirma\$>,<utente\$>,<modo\$>) Funzione per eseguire la firma digitale di un file mediante SmartCard	S/C
_ERRFIRMAF\$	Variabile che ritorna l'errore della funzione FIRMAFILE	S/C/Cd
DECODP7M()	Par: (<FileInput_p7m\$>,<FileOutput_cont\$>); Funzione per decodificare i file firmati p7m Es: RET=DECODP7M ("fattura.p7m", "fattura.xml") : IF RET <> 0 THEN VIMSG 4, _ERRP7M\$	S/C/Cr/Cd Sm/Cm
_ERRP7M\$	Descrizione dell' eventuale errore di DECODP7M(), altrimenti è vuota	S/C/Cd
CODIF_BASE64	Par: (<TipoOp\$>,<File_Input\$>,<File_Output\$>); Istruzione di codifica/decodifica Base64. L'istruzione è utile in Collage Server Remoto per gestire la transizione di file (es. immagini) fra Server e WebApp tramite file Json. Per inviare immagini da Collage Server Remoto verso la webapp occorre passare tramite un file Json che, nonostante consenta di passare solo testo, non esclude la possibilità di convertire l'immagine in BASE64, cioè una sequenza di caratteri che vengono scritti come un qualsiasi altro testo. Successivamente, lato webapp, questi caratteri vengono ricodificati per riavere l'immagine e le pagine web elaborano anche il BASE64. In ambito Json per la codifica BASE64 è disponibile la JSONGETFILE.	S/C/Cr/Cd
_ERRBASE64 _ERRBASE64\$	Codice e messaggio esito operazione codifica/decodifica Base64.	S/C/Cd
ZIP_CREA	Par: <NomeArchivioZIP\$>, <Flag_Sovrascrivi_Aggiungi>, <PATH_da_eliminare>, <Array_File\$()>; Istruzione per creare/implementare archivi file .ZIP. Esempio DIM FILE\$() FILE\$(1)="c:\tmp\file\file1.txt" FILE\$(2)="c:\tmp\file\file2.png" FILE\$(3)="c:\tmp\file\file3.jpg" ZIP_CREA "c:\tmp\file.ZIP",0,"c:\tmp",FILE\$() IF _ERRZIP\$<>"" THEN VIMSG 4,_ERRZIP\$ END Può essere utile eseguire un preventivo ciclo per controllare gli accessi ai file, v. _ACCFILEUTE I file ZIP sono letti / scritti / elencati nei contenuti con funzioni "stand-alone" indipendenti (utilizzabili anche se non accompagnate ad altre funzioni); tali funzioni per Zippare / Unzippare sono utili per gestire XML grandi con dati grandi (non gestibili con le consuete variabili XML) e per il successivo invio di questi file in formato ZIP. Quanto alla funzione WEBSVC questa è già in grado di inviare file, anche zippato, ma solo se il file ZIP è già presente sul disco; dato che non è possibile costruire lo ZIP in runtime con il codice SPRIX, le suddette funzioni consentono di costruirlo direttamente prima di inviarlo. Nota bene: e' possibile includere un massimo di 1024 files per archivio.	S/C/Cr/Cd
ZIP_ESTRAI	Par: <NomeArchivioZIP\$>, <DirectoryEstrazione\$>, <FlagEliminaPath>	S/C/Cr/Cd
ZIP_LISTA	Par: <NomeArchivioZIP\$>, <NomeFile\$()>, <DirectoryFile\$()>	S/C/Cr/Cd
_ERRZIP\$	La variabile di sistema _ERRZIP\$ riporterà eventuali errori occorsi durante l'esecuzione dell'istruzione, un valore vuoto (" ") indica che l'esecuzione è terminata correttamente	S/C/Cd
VISUALIZZAFI	Par: <tipoeditor>,<file\$>. Istruzione che visualizza e/o apre file con editor	S/C/Cd/Sm
INPUT_EXT	Istruzione di gestione file e directory (e anche di input a scorrimento in zona mouse)	S/C/Cd
GETINTERNET	Par: <url\$>,<file\$>. Istruzione che scarica file html da Internet. es.	S/C/Cr/Cd
_ERRGETINT\$	Messaggio operazione di GETINTERNET	S/C/Cd
GEOMAPPA	Par: [LINK\$],[IND\$],[LOC\$],[CAP\$],[PROV\$],[ISONAZ\$],[NAZIONE\$]. Istruzione che apre la mappa	S/Cr/Cd/Sm/Cm
IMOD	Import Moduli Sprix (v. tabella Moduli e Archivi del Gestionale)	S
AUTOFILE()	Par: (). Funzione che torna il primo numero di file disponibile	S/C/Cr/Cd/Sh
OPEN	Par: <tipo\$>,<numfile>,<nomefile\$>[,<nomefile\$;pos1;lun1;pos2;lun2;...>,<lr> Istruzione che apre file sequenziale o indici. NB: lettura è su file formattato (record a dimensione fissa); la lettura a record variabile richiede il riempimento con "blank" i record con bytes inferiori alla massima dimensione del record fisso, oppure, la lettura del record un carattere per volta con controllo sul carattere di fine record (da conoscerlo per individuare quando la riga termina oppure no). Anche CSV è un formato testo con carattere di fine riga (esempio ";"). es.	S/C/Cr/Cd/Sh
FIELD	Par: <numfile>,<lunbuf\$> AS <varbuf\$> [tipoimp], Istruzione che dichiara i buffers di un file Si possono gestire fino a 128 campi per ogni file, ogni record può avere al massimo 2048 caratteri ed ogni singolo FIELD può avere al massimo 255 caratteri. Se più istruzioni FIELD si riferiscono allo stesso, esse costituiscono un'unica FIELD in cui l'istruzione successiva è la continuazione di quella precedente. Se la lunghezza record del file specificata nella OPEN risulta superiore rispetto alla sommatoria delle lunghezze dei campi specificati nella FIELD la differenza viene assunta come composta da tanti ascii 0.	S/C/Cr/Cd/Sh

	viceversa viene generato un errore in fase di esecuzione. Se la variabile dell'istruzione FIELD è una array il relativo indice deve essere una costante: FIELD 1, 2 AS XXX(1) (Istruzione Ammessa) ; FIELD 1, 2 AS XXX(!) (Istruzione Errata). Esempio: buffer composto da 4 campi, il primo alfanumerico lungo 2, il secondo alfanumerico lungo 3 di tipo stringa, il terzo numerico senza decimali lungo 6 di tipo ASCII, il quarto numerico con 1 decimale lungo 3 di tipo binario. es.	
LOF()	Par: (<numfile>). Funzione che torna l'ampiezza in caratteri di un file sequenziale	S/C/Cr/Cd/Sh
LOFKEY()	Par: (<numfile>). Funzione che torna il numero di record di un file indici	S/C/Cr/Cd/Sh
PUT	Par: (<numfile>,<numrec>). Istruzione che scrive record su file tipo sequenziale	S/C/Cr/Cd/Sh
GET	Par: (<numfile>,<numrec>). Istruzione che legge il record file tipo sequenziale (R)	S/C/Cr/Cd/Sh
GETKEY	Par: (<numfile>,<numkey>,<key\$>[,key1\$]). Istruzione che legge il record file tipo ad indici (K)	S/C/Cr/Cd/Sh
NEXTKEY	Par: (<numfile>,<numkey>). Istruzione che legge il record successivo	S/C/Cr/Cd/Sh
PREVKEY	Par: (<numfile>,<numkey>). Istruzione che legge il record precedente	S/C/Cr/Cd/Sh
DELKEY	Par: (<numfile>,<key\$>). Istruzione che cancella un record di un file ad indici	S/C/Cr/Cd/Sh
VARKEY	Par: (<numfile>,<key\$>). Istruzione che varia il record di un file ad indici. es.	S/C/Cr/Cd/Sh
INSKEY	Par: (<numfile>). Istruzione inserimento record in file ad indici	S/C/Cr/Cd/Sh
_NRDATI	Numero record (chiavi attive); variabile utile al controllo (come anche _ERRIND) per testare l'avvenuta operazione sul record, ma anche per sapere se in un certo periodo di tempo il file ad indici ha subito variazioni di sorta, fra cui ad esempio INSKEY, in cui _NRDATI riporta il numero di record di inserimento per quanti elementi sono stati inseriti con INKEY	S/C/Cd
_ERRIND	Codice esito operazione su file indici	S/C/Cd
CLOSE	Par: (<numfile>). Istruzione che chiude il file es.	S/C/Cr/Cd/Sh
SQL_DRIVERCONN	Par:<modulo>,<ncon>,<tipo\$>,<driver\$>,<server\$>,<porta>,<nomedb\$>,<ute\$>,<pwd\$>. Istruzione che apre connessione a database.	S/C/Cr/Cd/Sh
SQL_CONNECT	Par:<nc>,<DataSource\$>,<UserName\$>,<Password\$>. Istruzione che carica il driver e connette il DataBase. Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_AUTOCOMMIT	Par:<nc>,<tipo>. Istruzione per la modalita' di gestione delle transazioni Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_ALLOCSTMT	Par:<nc>,<ns>. Istruzione che alloca memoria per handle di statement Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_PREPARE	Par:<nc>,<ns>,<str1\$>[,str2\$]... Istruzione che prepara stringa SQL per l'esecuzione Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_BINDCOL	Par:<nc>,<ns>,<nbind>,<nomevar\$>[,D]. Istruzione che assegna var. all'invocazione SQL_FETCH Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_EXECDIRECT	Par:<nc>,<ns>,<str1\$>[,str2\$]... Istruzione che esegue una istruzione SQL Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_EXECUTE	Par:<nc>,<ns>. Istruzione che esegue uno statement "preparato" Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_FETCH	Par:<nc>,<ns>. Istruzione che legge una riga dal result set Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
DTK_GETCOLONNA	Istruzione DataTrek che ritorna il valore del campo nell'elemento della BINDCOL dichiarata in precedenza. Data Trek consente di interagire con le informazioni presenti sulla sua base dati, ed eventualmente del MexalDb SQLite. Ciò è possibile solo ed esclusivamente dall'app Datatrek. In programmazione Sprix/Collage fare attenzione alla chiusura degli Statment, poiché se lasciati aperti le successive operazioni creano errori di data base locked. La base dati impiegata, infatti, è di tipo DBMS e non RDBMS, pertanto il programmatore deve ricordarsi di rilasciare il database per evitare errori. SQLite esegue il lock a livello di filesystem (il database viene allocato nella sua interezza). Per effettuare una richiesta: <ul style="list-style-type: none"> • Stabilire la connessione con SQL_DRIVERCONN (modulo 4) e fare riferimento al database tramite il codice oggetto DataTrek. I programmi devono trovarsi rigorosamente in un'App DataTrek. <u>Non è consentito accedere a oggetti DataTrek presenti su altre PassApp.</u> • Allocare un nuovo statement : SQL_ALLOCSTMT • Effettuare il Bind con le variabili sprix (SQL_BINDCOL) • Eseguire la richiesta SQL : SQL_EXECDIRECT • Estrazione di un record per volta (SQL_FETCH) • Estrazione per singolo campo e bind del valore (DTK_GETCOLONNA) • Al termine della lettura rilasciare le risorse dello statement (SQL_FREESTMT) • Chiudere la connessione con SQL_DISCONNECT 	S/C/Cd
_ERRSQL_NDF	Variabile indicatore di fine dati nel cursore	S/C/Cd
SQL_FREESTMT	Par:<nc>,<ns>. Istruzione che libera risorse associate allo statement Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_TRANSACT	Par:<nc>,<tipo>. Istruzione che richiede operazione commit o rollback Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
SQL_DISCONNECT	Par:<nc>. Istruzione che chiude la connessione Non consentita in Passcom, non consentita in Server Farm	S/C/Cr/Cd/Sh
_ERRSQL_N-M-\$S\$	Variabili di errore operazioni SQL	S/C/Cd
	Nota: le indagini sugli errori SQL possono anche essere eseguite sul file di log pxerrsql (vedere cartella pxlog della cartella dati) e approfondite con altri log del server. Per altri tipi di anomalie odbc, verificare i log e le documentazioni odbc. Altre casi di errori, invece, potrebbero essere causate da modifiche (attivazioni, configurazioni, implementazioni) relative ad altri moduli che utilizzano connessioni e repliche a database (mexaldb, docuvision, dbmirror, adp, datatrek, ecc...), pertanto occorre individuare quale fra questi potrebbe essere responsabile; un metodo empirico (per approssimazioni successive) è quello di disattivare il modulo "sospetto" e cercare di replicare l'errore. Per ulteriori approfondimenti consultare il capitolo MexalDB del Manuale Utente	
DTK_GETCOLONNA	Par: (<nc>,<nstmt>,<nomecampo>,<elementobindcol> Istruzione che ritorna il valore del campo nell'elemento della bindcol. Vedi esempio sotto.	S/C/Cd

ADCOMMERCIALE	Par: <ModAgg>, <MM>, <AnnInizElab>, <AnnoFineElab>, <PrzCalcMarg>, <OC>, <OF>, <MA> Istruzione che esporta dati commerciali in Dbase di ADCOMMERCIALE Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/Cd
_ERRADCOM	Codice risultato istruzione ADCOMMERCIALE	S/C/Cd
_ERRACDOM\$	Messaggio operazione ADCOMMERCIALE	S/C/Cd
ADPCOMMERCIALE / ADPINDICI	Par: <annorif>, <annoprec>, <esplostrut>. Istruzioni (obsolete) per esportare dati su obsolete analisi dati Commerciale e di Indici di Bilancio Non consentite in Passcom, non consentite in Server Farm	S/Cd
ASCMIRROR	Par: <tabella>, <dir>, <file>. Istruzione che esporta dati su file ASCII (*a)	S/Cr/Cd/Sh
_ERRASC MIR	Codice risultato operazione ASCMIRROR (*a)	S
ASCMIRROR_EXT	Par: <tip>, <rif>, <dir>, <file>, <slot>, <att>, <var>, <azz>, [vis] Istruzione che esporta selezione di dati su file ASCII (*a) Non consentito nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
ASCMIRROR_EXT2	Par: <tip>, <rif>, <dir>, <file>, <slot>, <att>, <var>, <azz>, [impo\$] Istruzione che esporta selezione di dati su file ASCII parametrico (*a) Non consentita nel Livello di sviluppo (F2), non consentita in Server Farm	S/Cr/Cd/Sh
DBMIRROR	Par: <tabella>. Istruzione che esporta dati su DataBase (*a) Non consentita in Passcom, non consentita in Server Farm	S/Cr/Cd/Sh
DBMIRROR_EXT	Par: <tip>, <rif>, <slot>, <att>, <var>, <azz>, [vis]. Istruzione che esporta selezione di dati su DataBase (*a) Non consentita in Passcom, non consentita in Server Farm	S/Cr/Cd/Sh
_ERRDBMIRROR	Codice risultato esportazione su Dbase; vedere _ERRDBMIRR\$ (*a)	S
_ERRDBMIRR\$	Messaggio operazione esportazione su DB(*a)	S
MEXALDB	Par: <p1>, <p2>[,d\$]. Istruzione per esportazione dati su data base Mexaldb (*a). La versione dei dati di MexalDB la si ottiene eseguendo la query seguente: SELECT NGB_VERS_DB FROM [Sigla azienda]_RP_INFO Non consentita nel Livello di sviluppo (F2), non consentita in Passcom e/o in Server Farm Dalla versione 2020A (v750) è consentita la replica su SQLite in chiaro.	S/Cr/Cd/Sh
_ERRMEXALDB	Codice risultato operazione con MEXALDB (*a)	S
_ERRMEXALDB\$	Messaggio operazione con MEXALDB (*a)	S
	(*a) Nel ciclotabellare non utilizzare le operazioni di mirroring. Ulteriori registrazioni degli errori di mirroring avvengono nel file di log pxerrsql.<num term>. Si ricorda che la struttura dei dati del Gestionale è consultabile nei seguenti modi: <ul style="list-style-type: none"> Le Tabelle e Colonne sono definite nel documento "ElencoColonneMexalDB.xls" allegato al chiarimento soluzione n. 30663 del servizio di assistenza EasyPass accessibile dall'Area Riservata di www.Passepartout.net. I contenuti delle colonne nel file vanno aperte ed estese. Per approfondimenti consultare il capitolo MexalDB del Manuale Utente I Tracciati CSV sono pubblicati nel chiarimento n. 54452 del supporto on-line. I Tracciati CSV sono verificabili nel menu del Gestionale [Servizi - Trasferimento Archivi - Export/Import CSV - ASCII] Specifiche dei campi Dizionario nel menu del Gestionale [Servizi - Personalizzazioni - Stampa Dizionario Sprix] I dati obbligatori sono indicati a video dal Gestionale e le relazioni fra i dati sono descritte nel Manuale Utente. I casi di Errore connessione o di connessione chiusa potrebbero richiedere la modalità sincrona di MexalDB ed evitare la concorrenza di connessioni su MexalDB fra l'App e il Gestionale oppure altre sue connessioni	
OPENCOM	Par: <com>, <baudrate>, <parity>, <size>, <stopbit>, <timeout>, <flag> Istruzione che apre una porta di comunicazione seriale. es.	S/C/Cr/Cd/Sh
GETCOM	Par: <n>. Istruzione che legge i dati dalla porta seriale. es.	S/C/Cr/Cd/Sh
_BUFCOM\$	Buffer in cui GETCOM scrive i byte. es.	S/C/Cr/Cd/Sh
PUTCOM	Par: <a>, <n>. Istruzione che scrive dati sulla porta seriale. es.	S/C/Cr/Cd/Sh
_ERRCOM\$	Errori funzioni gestione porta seriale. es.	S/C/Cr/Cd/Sh
CLOSECOM	Istruzione che chiude la porta seriale aperta. es.	S/C/Cr/Cd/Sh
SHELL	Par: <comando>. Istruzione shell di comando di sistema operativo. es. Non consentita lato Server Passcom, non consentita in Server Farm	S/C/Cd
SHELLSPRIX	Par: [<NomeDft>][<t<NumTerm>][<o<NumOper>][<x<Data>][<a<SiglaAz>[:<codSottAz\$>]] [-k=<Ute:Pwd> oppure -k<Chiave>] -p[#]<Sprix> [-v<ArgCmd>]. Istruzione che lancia uno Sprix da un altro Sprix oppure da Collage es. Nota: istruzione non consentita nel Livello di sviluppo (L3); uno Sprix può essere lanciato anche da riga di comando esterna (consultare "mxrs.exe" e "mxrsbatch.exe" nel manuale utente);	S/C/Cr/Cd
_ARGCMD\$	Variabile con argomenti linea comando	S/C/Cd
SPXREMOTO\$()	Par: (<ind>, <par1>, <par2>, <filed>, <filee>, <sync>). Funzione che esegue uno Sprix Remoto	S/C/Cd/Sh
PUTREM_DATI	Par: <buff>. Istruzione di scrittura file dati da Sprix Remoto	S/C/Cd
PUTREM_ERR	Par: <buff>. Istruzione di scrittura file errori da Sprix Remoto	S/C/Cd
FTPESECCMD	Istruzione per invocare comandi FTP (Gest 64 bit) es.	S/C/Cr
_FTP	Variabili utilizzate da FTPESECCMD comunicare ed eseguire comandi su Server FTP/FTPS es.	S/C/Cd

esempi e casi

Esempi Sprix:

Gestione di file ad accesso diretto e ad indici

Attenzione! il copia e incolla del testo nell'Editor Sprix, potrebbe sostituire il carattere (") con (), pertanto, appoggiare il contenuto su file testo, quindi eseguire la sostituzione massiva del carattere con ".

```
VIMSG 1, "IMPORTO FILE HTML DA INTERNET"
  _ACCFILEUTE=1
GETINTERNET "www.passepartout.net", "c:\temp\Pass.html"
VIMSG 1, "TEST SU DIRECTORY E LEGGO FILE"
IF _ERRGETINT$<>" THEN VIMSG 4, _ERRGETINT$

LEGGIDIR "c:\temp", "" : FOR F=1 TO _NUMDIRFILE
  WA_FILE$=WA_FILE$+"|"+_DIRFILE$(F) : NEXT : VIMSG 1, "<FILE:|"+WA_FILE$

VIMSG 1, "SCRIVO UN FILE SEQUENZIALE FORMATTATO"
F$="c:\temp\file.txt": CRLF$=CHR$(13)+CHR$(10)
NF=AUTOFILE() : OPEN "R",NF,F$,12:FIELD NF,10 AS REC$,2 AS CRLF$
FOR R=1 TO 9 : REC$=STRING$(R,STR$(R,0,0)) : PUT NF,R : NEXT : CLOSE NF
VIMSG 1, "LEGGO IL FILE SEQUENZIALE FORMATTATO"
```

```

OPEN "R",NF,F$,12 : FIELD NF,10 AS REC$,2 AS CRLF$
NREC=LOF(NF)/12 : DIM REC$( ) : FOR R=1 TO NREC : GET NF,R : REC$(R)=REC$
TRC$=TRC$+"|"+CVA$(REC$(R),10):NEXT:VIMSG 1, "<Record Totali:|" +TRC$

VIMSG 1, "AGGIUNGO DEI RECORD FORMATTATI"
REC$="record 10" : PUT NF,10 : REC$="record 11" : PUT NF,11 : CLOSE NF

VIMSG 1, "AGGIUNGO RECORD (SOVRASCRIVO E PERDO FORMATTAZIONE -CRLF$-)"
OPEN "R",NF,F$,12 : FIELD NF,10 AS REC$
REC$="SPRIX" : PUT NF,4 : REC$="COLLAGE" : PUT NF,7
REC$=" MDS" : PUT NF,12 : REC$=" SDK" : PUT NF,13
CLOSE NF

VIMSG 1, "LEGGO IL FILE NON FORMATTATO|(-record dimensione variabile-)"
OPEN "R",NF,F$,1:FIELD NF,1 AS CAR$:DIMFILE=LOF(NF):RECORD$="" :R=0
FOR C=1 TO DIMFILE: GET NF,C
IF CAR$<>CHR$(13) AND CAR$<>CHR$(10) THEN GOTO CONCATENA
IF CAR$=CHR$(13) THEN GOTO PROSSIMOREC
IF CAR$=CHR$(10) THEN GOTO VERIFICA
CONCATENA: RECORD$=RECORD$+CAR$ : GOTO PROSSIMOREC
VERIFICA:R=R+1
IF INSTR(RECORD$,"333")<>0 THEN VIMSG 1,"333|è al record "+STR$(R,0)
RECORD$=""
PROSSIMOREC: NEXT : CLOSE NF

VIMSG 1, "CREO UNA COPIA DI BACKUP DEL FILE SEQUENZIALE E LO CANCELLO"
IF ESISFILE(F$) THEN COPYFILE F$,"c:\temp\file_r_back.txt" : KILL F$

VIMSG 1, "CREO FILE A 2 CHIAVI CON:| Articolo, Descrizione, Categoria"
F$="c:\temp\file.txt":FK$="c:\temp\file1.k":F2K$="c:\temp\file2.k"
OPEN "K",NF,F$,FK$:1;16,F2K$:4;1;10,52
FIELD NF,20 AS ART$ ALFA,20 AS DES$ ALFA
FIELD NF,10 AS CAT NUMO,2 AS CRLF$ ALFA
FOR I=1 TO 9: I$=STR$(I,0)
ART$="Articolo n."+I$: DES$="Descrizione "+I$: CAT=VAL(STRING$(I,I$))
INSKEY NF : IF _ERRIND=1 THEN VIMSG 4,"record già presente"
NEXT: CLOSE NF

VIMSG 4, "CERCO|Articoli n.10 e n.5|PER CHIAVE PRIMARIA E LI MODIFICO"
OPEN "K",NF,F$,FK$:1;16,F2K$:4;1;10,52
FIELD NF,20 AS ART$ ALFA,20 AS DES$ ALFA,10 AS CAT NUMO,2 AS CRLF$ ALFA
KEY$="Articolo n.10": GETKEY NF,1,KEY$
IF _ERRIND=1 THEN VIMSG 4,"articolo 10 cercato ma non trovato":
NEWKEY$="Articolo n.5": GETKEY NF,1,NEWKEY$
IF _ERRIND=0 THEN VIMSG 1,"articolo 5 trovato, gli cambio descrizione"
NEWDESS$=RIGHT$(STRING$(5,"NUOVO"),20):DESS$=NEWDESS$
VARKEY NF, NEWKEY$: IF _ERRIND=0 THEN VIMSG 1, "Descrizione 5 variata"

VIMSG 1, "ORA LEGGO PROSSIMI RECORD E VARIO DESCRIZIONI"
CICLO: NEXTKEY NF,1: IF _ERRIND=1 THEN GOTO FINEFILE
KEYART$=LEFT$(ART$+STRING$(20,CHR$(0)),20)
DESS$=NEWDESS$:VARKEY NF, KEYART$:IF _ERRIND=1 THEN VIMSG 4,"non variati"
IF _ERRIND=1 THEN VIMSG 4,"record non variato"
GOTO CICLO
FINEFILE: CLOSE NF: VIMSG 1, "file terminato"

VIMSG 4, "LEGGO A RITROSO PER CHIAVE NUMERICA SECONDARIA"
LOCATE 5,2:PRINT "Leggi dalla categoria 1,22,333 o 4444.": :
INPUT "N",CERCAT,5: CERCAT$=RIGHT$(" "+STR$(CERCAT,0,0),10)
OPEN "K",NF,F$,FK$:1;16,F2K$:4;1;10,52
FIELD NF,20 AS ART$ ALFA,20 AS DES$ ALFA,10 AS CAT NUMO,2 AS CRLF$ ALFA

VIMSG 1, "chiavi totali: "+ STR$(LOFKEY(NF),0,0)
KEY$=RIGHT$(CERCAT$,10):GETKEY NF,2,KEY$
IF _ERRIND=0 THEN VIMSG 1,"Numero Record Dati: "+STR$( _NRDATI,0,0)
CICLOPREV: PREVKEY NF,2: IF _ERRIND=1 THEN GOTO INIZFILE

VIMSG 4, "Key Precedente "+STR$(CAT,0,0)+": con "+DESS$: GOTO CICLOPREV
INIZFILE: VIMSG 1, "lettura termine file effettuata": CLS
T1$="ADESSO CERCA PER CATEGORIA (KEY2)"
T2$="SE TROVA VARIA, ALTRIMENTI INSERISCE NUOVO RECORD"
T3$="E CANCELLA IL RECORD CON KEY1 |'Articolo n.5"
VIMSG 1,T1$+"|"+T2$+"|"+T3$

VARIAZIONE:
LOCATE 5,2:PRINT "Cerca Categoria (5 numeri).: ":
INPUT "N",CERCAT,5: CERCAT$=RIGHT$(" "+STR$(CERCAT,0,0),10)
LOCATE 6,2:PRINT "Nuova Categoria (5 numeri).: ":
INPUT "N",NEWCAT,5

GETKEY NF,2,CERCAT$: OLDART$=ART$ 'memorizzo la key principale
IF _ERRIND=1 THEN TROV$="N" ELSE TROV$="S"

GOSUB CTRL: IF CTRL$<>"OK" THEN GOTO VARIAZIONE
IF TROV$="S" THEN VARKEY NF,OLDART$:CLOSE NF:GOTO CHECK
IF TROV$="N" THEN INSKEY NF: DELKEY NF, "Articolo n.5":CLOSE NF
'(VARKEY mantiene il format su editor rispetto a DELKEY+INSKEY)

CHECK: IF _ERRIND<>0 THEN VIMSG 4, "problemi in VAR/INS/DEL KEY"
END

GOTO VARIAZIONE:
CTRL: VIMSG 1, "effettuo controlli/modifiche..."
CAT=VAL(" "+NEWCAT
ART$="NEW ARTICOLO ????? ":DESS$="NEW DESCRIZIONE ??? "
CTRL$="OK": RETURN

```

Esempi GETINTERNET

```

' Host che consente sia protocollo non sicuro che protetto "www.rfc-editor.org/rfc/rfc5321.txt"

'PERCORSO SERVER (ACCEDERE A FILE SYSTEM NON E' CONSENTITO IN AMBIENTE LIVE)
'GETINTERNET "http://www.rfc-editor.org/rfc/rfc5321.txt", "C:\TMP\rfc5321.txt"
'GETINTERNET "https://www.rfc-editor.org/rfc/rfc5321.txt", "C:\TMP\rfc5321.txt"

'PERCORSO CLIENT
'GETINTERNET "http://www.rfc-editor.org/rfc/rfc5321.txt", "C:\TMP\rfc5321.txt"
'GETINTERNET "https://www.rfc-editor.org/rfc/rfc5321.txt", "C:\TMP\rfc5321.txt"

```

```
'PERCORSO <DATI AZIENDA>\ ("C:Pass\dati\datiaz\sig\datiutente\...")
'GETINTERNET "http://www.rfc-editor.org/rfc/rfc5321.txt","]\rfc5321.txt"
'GETINTERNET "https://www.rfc-editor.org/rfc/rfc5321.txt","]\rfc5321.txt"

'PERCORSO '<DATI GENERALI>\ ("C:Pass\dati\datigen\datiutente\...")
'GETINTERNET "http://www.rfc-editor.org/rfc/rfc5321.txt","]\rfc5321.txt"
'GETINTERNET "http://www.rfc-editor.org/rfc/rfc5321.txt","]\rfc5321.txt"

' Test Errore
' VIMSG 4,_ERRGETINT$

' Copia da Server a Client
' COPYFILE "C:\TMP\rfc5321.txt", "<C:\TMP\rfc1234.txt" '(LOCALHOST O AUTORIZZAZIONE ACCESSO UTENTE)
' COPYFILE "]\rfc5321.txt", "<C:\TMP\rfc1234.txt"
' COPYFILE "!\rfc5321.txt", "<C:\TMP\rfc1234.txt"
'END
```

File sequenziale con lunghezza record a dimensione variabile (ulteriore esempio)

Per evitare di "riempire" con blank i record con bytes inferiori alla massima dimensione del record (per ottenere la formattazione), si può leggere il record un byte alla volta con controllo sul byte di fine record. L'istruzione OPEN gestisce i file sequenziali a lunghezza record variabile, leggendo carattere per carattere e individuando i caratteri di fine riga del testo.

```
FILE$="C:\TESTO.TXT"
IF NOT ESISFILE(FILE$) THEN VIMSG 4,"FILE INESISTENTE":END
N_FILE=AUTOFILE()
OPEN "R",N_FILE,FILE$,1
FIELD N_FILE,1 AS CARATT$
DIMFILE=LOF(N_FILE)
FOR I=1 TO DIMFILE
GET N_FILE,I
IF CARATT$=CHR$(13) THEN GOTO PROSSIMO
IF CARATT$=CHR$(10) THEN GOTO STAMPA
IF I=DIMFILE THEN GOTO STAMPA
RIGA$=RIGA$+CARATT$
PROSSIMO:
NEXT
CLOSE N_FILE
END
STAMPA:
PRINT RIGA$
RIGA$=""
GOTO PROSSIMO
```

Apertura a ritroso di più file (una serie di 13 file in C:\):

```
FOR I=13 TO 1 STEP -1
F$="c:\file"+STR$(I,0,0)+".txt"
OPEN "R",I,F$,10
NEXT
INPUT "A", A$, 1
END
```

Confronto fra file

```
` Dati due file su disco esegue la comparazione (byte per byte)

UGUALI = 1 ` 1=uguali 0=diversi
NF1 = 1: FILE1$ = "c:\temp\file1.txt"
NF2 = 2: FILE2$ = "c:\temp\file2.txt"
OPEN "R", NF1, FILE1$, 1: FIELD NF1,1 AS BUF1$
OPEN "R", NF2, FILE2$, 1: FIELD NF2,1 AS BUF2$
` Prima confronto la dimensione dei 2 file: se diversa -> files diversi
DIMF1 = LOF( NF1 ) : DIMF2 = LOF( NF2 )
IF DIMF1<>DIMF2 THEN UGUALI = 0: GOTO FINE
` ciclo lettura byte per byte con confronto
`FOR CTREC=1 TO DIMF1
` GET NF1,CTREC: GET NF2,CTREC
` IF BUF1$ <> BUF2$ THEN UGUALI = 0
`NEXT
CTREC=0
CICLAFI: ` inizio ciclo lettura files
CTREC = CTREC+1
IF CTREC > DIMF1 THEN GOTO FINE ` sono a fine file
GET NF1,CTREC: GET NF2,CTREC
IF BUF1$ <> BUF2$ THEN UGUALI = 0: GOTO FINE ELSE GOTO CICLAFI
` fine ciclo lettura files
FINE:
IF UGUALI = 1 THEN VIMSG 4, "FILE UGUALI" ELSE VIMSG 4, "FILE DIVERSI"
CLOSE NF1
CLOSE NF2
```

Esempio VARKEY (consigli di utilizzo)

L'istruzione VARKEY permette di variare le informazioni contenute all'interno di un record (che per potere essere variato deve esistere) in un file a chiave. Si sottolinea il ruolo fondamentale che ha la dimensione del/i campo/i che compongono la chiave principale, infatti se ad esempio si avesse una chiave di 10 caratteri, ma il valore della chiave da modificare fosse "PIPP0" (di soli 5 caratteri dunque) si genererebbe un errore in quanto la stringa di ricerca dovrebbe essere comunque lunga 10 caratteri. Di conseguenza, per potere eseguire una ricerca in modo corretto è sempre consigliabile scrivere una stringa come quella che segue:

```
OPEN "K",NUM_FILE,NOME_FILE$,KEY_FILE$:1;10,20
FIELD NUM_FILE,10 AS KEY$
.....
.....
`STRINGA DA RICERCARE: "PIPP0"
CHIAVE$=LEFT$( "PIPP0"+STRING$(10,CHR$(0)),10)
VARKEY NUM_FILE,CHIAVE$
IF _ERRIND=1 THEN VIMSG 1,"ERRORE LEGATO A VARKEY"
.....
.....
CLOSE NUM_FILE
```

END

Esempio Sprix:

Per ordinare una tabella in modo crescente sul primo campo e decrescente sul secondo. Questo è uno fra i vari metodi che si possono adottare

```
VIMSG 1, "CREO FILE E CHIAVI CON: | Descrizione, Importo"
CREO_FILE:
F$="c:\temp\fileimp.txt"
F3K$="c:\temp\fileimp3.k" 'Tutto
NF=AUTOFILE()

OPEN "K",NF,F$,F3K$;1;30,32

FIELD NF,20 AS NOME$ ALFA 'primo campo di 20 caratteri
FIELD NF,10 AS IMP NUM2,2 AS CRLF$ ALFA 'secondo campo di 10 caratteri + CRLF$
'Questi sono i dati che andremo a caricare.
'Pippo 1000.00
'Pluto02 3000.00
'Paperino03 200.00
'Pippo 1200.00
'Paperino03 3200.00
'Pluto02 320.00

NOME$ = "Pippo"
IMP = 1000
CRLF$ = CHR$(13)+CHR$(10) ' Ritorno a capo
INSKEY NF
IF _ERRIND=1 THEN VIMSG 4,"record gia presente"

NOME$ = "Pluto02"
IMP = 3000
CRLF$ = CHR$(13)+CHR$(10) ' Ritorno a capo
INSKEY NF
IF _ERRIND=1 THEN VIMSG 4,"record gia presente"

NOME$ = "Paperino03"
IMP = 200
CRLF$ = CHR$(13)+CHR$(10) ' Ritorno a capo
INSKEY NF
IF _ERRIND=1 THEN VIMSG 4,"record gia presente"

NOME$ = "Pippo"
IMP = 1200
CRLF$ = CHR$(13)+CHR$(10) ' Ritorno a capo
INSKEY NF
IF _ERRIND=1 THEN VIMSG 4,"record gia presente"

NOME$ = "Paperino03"
IMP = 3200
CRLF$ = CHR$(13)+CHR$(10) ' Ritorno a capo
INSKEY NF
IF _ERRIND=1 THEN VIMSG 4,"record gia presente"

NOME$ = "Pluto02"
IMP = 320
CRLF$ = CHR$(13)+CHR$(10) ' Ritorno a capo
INSKEY NF
IF _ERRIND=1 THEN VIMSG 4,"record gia presente"

CLOSE NF
'-----
'Dato un file contenente questi dati, andremo a generare un file di output
'ordinato in modo crescente per il primo campo e decrescente per il secondo campo.

'Esempio di file esterno contenente questi dati in Input c:\fileimp.txt
'Pippo 1000.00
'Pluto02 3000.00
'Paperino03 200.00
'Pippo 1200.00
'Paperino03 3200.00
'Pluto02 320.00

'Il risultato che vogliamo ottenere sara' questo
'Paperino03 3200.00
'Paperino03 200.00
'Pippo 1200.00
'Pippo 1000.00
'Pluto02 3000.00
'Pluto02 320.00
'e verra' scritto su file fileout.txt in c:\qtemp
STAMPO_FILE:
CRLF$ = CHR$(13)+CHR$(10) ' Ritorno a capo
DIM ARRTMP$( )
OLD_DESCR$=""
F$="c:\temp\fileimp.txt"

FOUT$="c:\temp\fileout.txt"
FOUTK$="c:\temp\fileout.k" 'Chiave x tutta lunghezza primo e secondo campo

F1K$="c:\temp\fileimp1.k"
F2K$="c:\temp\fileimp2.k"
F3K$="c:\temp\fileimp3.k"
NF=AUTOFILE()
NFOUT=NF + 1

OPEN "K",NFOUT,FOUT$,FOUTK$;1;30,32
FIELD NFOUT,20 AS NOMEOUT$ ALFA
FIELD NFOUT,10 AS IMPOUT NUM2,2 AS CRLFOUT$ ALFA

OPEN "K",NFOUT,FOUT$,FOUTK$;1;30,32
FIELD NFOUT,20 AS NOMEOUT$ ALFA
FIELD NFOUT,10 AS IMPOUT NUM2,2 AS CRLFOUT$ ALFA

KEY$ = ""
GETKEY NF,1,KEY$ ' MI POSIZIONO AL PRIMO RECORD
```

```

LOOP:
IFF (OLD_DESCR$="")
  \ PRIMA LETTURA
  I = 0
  OLD_DESCR$=NOME$
ENDF
IFF OLD_DESCR$ = NOME$
  GOSUB ORDINA
ELSEF
  GOSUB SCRIVI_FILE
  AZZARRAY ARRTMP$
  I = 0
  GOSUB ORDINA
  OLD_DESCR$ = NOME$
ENDF
NEXTKEY NF,1

IF _NRDATI <> 0 THEN GOTO LOOP

GOSUB SCRIVI_FILE \ Ultimo elemento che altrimenti non veniva scritto.

CLOSE NF
CLOSE NFOUT
END

\-----
ORDINA:
  _LRECSORT=8          \Ordine sulla base di 8 caratteri
  ST$ = STR$(IMP,0,0)
  PUTSORT RIGHTS("00000000"+ST$,8)
  VIMSG 2,"Nome = "+NOME$ +"|"+"Importo "+ ST$
  I = I +1
RETURN
\-----
SCRIVI_FILE:
  AZZARRAY ARRTMP$
  FOR X = 1 TO I
    GETSORT VISUALIZZA$
    ARRTMP$(X) = VISUALIZZA$
  \ GETSORT VISUALIZZA$
  VIMSG 2, "VIS :"+ ARRTMP$(X)
  NEXT
  FOR X = I TO 1 STEP -1
    \SCRIVI FILE SU DISCO...
    NOMEOUT$ = OLD_DESCR$
    IMPOUT  = VAL(ARRTMP$(X))
    CRLFOUT$ = CRLF$
    INSKEY NFOUT
    IF _ERRIND=1 THEN VIMSG 4,"record gia presente"

    VIMSG 2, "VIS ORD:" + ARRTMP$(X)
  NEXT
RETURN

```

Esempio: apertura di file dinamici (con contenuto a record variabile) con scrittura carattere per carattere

```

MAIN:
GOSUB APRO
STRINGA$="testo12345" :GOSUB SCRIVO
STRINGA$="testo123" :GOSUB SCRIVO
GOSUB CHIUDO
END

APRO:
NF=AUTOFILE()
F$="<c:\tmp\file.txt"
OPEN "R",NF,F$,1
FIELD NF,1 AS DATO$
REC=LOF(NF)
RETURN

CHIUDO:
CLOSE NF
RETURN

SCRIVO:
FOR R=1 TO LEN(STRINGA$)
REC=REC+1
DATO$=MID$(STRINGA$,R,1)
PUT NF,REC
NEXT
'inserisco CR LF come terminatore record

REC=REC+1
DATO$=CHR$(13)
PUT NF,REC

REC=REC+1
DATO$=CHR$(10)
PUT NF,REC
RETURN

```

Per un ulteriore esempio consultare
<https://forum.passepartout.net/t/creazione-di-file-con-open-con-record-di-lunghezza-diversa/12782>

Esempio: apertura di file dinamici (con contenuto a riga record variabile) con istruzioni di stampa

```

PUTTAST "<c:\temp\file.csv#610"
APRISTAMP 120,""
IF _NUMSTAMP<>NUMSTAMP THEN AZZTAST:CHIUDISTAMP: VIMSG 5,"Errore Stampante" : END
LPRINT "aaaaa;abbbbbb;";
LPRINT "dddddd;" 'questa viene aggiunta alla precedente
LPRINT "CCCC;" ' questa su una nuova riga
CHIUDISTAMP

```

Configurazione stampante:
 dispositivo collegato al server
 tipo file
 da richiedere
 formato testo
 richiesta mai

nel formato azzerare i campi larghezza e altezza
 come file file mettere fine riga crlf
 inserendo il separatore ";" si può ottenere il CSV)

Caso: La OPEN di file K*, infatti, cancella i record dati (le chiavi duplicate vengono eliminate dalla riorganizzazione).

In caso di file ad indici manipolato (record cancellati) con relativa gestione delle chiavi compromessa, per cercare di recuperare il contenuto provare ad aprirlo con riorganizzazione delle chiavi (K*), oppure, provare ad aprirlo a lettura sequenziale e riscriverlo in modalità K*.

Messaggio: File già aperto

Più COLLAGE possono aprire in automatico più file DIVERSI e quindi solo se i "nomefile" sono diversi (si ricorda che AUTOFILE adotta la numerazione progressiva per ciascun COLLAGE, per cui non è complessiva per tutti i COLLAGE attivi; per gestire in modo globale la numerazione è disponibile la variabile _GLOBAL). Si ricorda, infine, che più COLLAGE che desiderano accedere allo stesso "nomefile" è richiesto che il primo COLLAGE che utilizza il "nomefile" deve preoccuparsi di chiuderlo prima che il successivo COLLAGE lo apra a sua volta con lo stesso "nomefile".

Messaggio: Errore Interno: Violazione protezione di memoria

L'Errore Interno un bug generico ("trap") che può verificarsi in qualsiasi punto del gestionale, chiudendolo improvvisamente. Tale errore non equivale all'Errore non correggibile (descritto più avanti), in quanto riguarda uno sporco di memoria, cioè la scrittura su una porzione di memoria relativa ad un'altra applicazione. La violazione di memoria non comporta gravi conseguenze, ma solamente il disturbo di dovere riavviare il Client e l'eventuale necessità di ripetere e/o correggere l'operazione in corso. Se l'errore è sistematico (replicabile) l'individuazione delle cause e la relativa correzione sono particolarmente difficili, in quanto questa tipologia di errore non è tracciabile, nemmeno aumentando il grado di tracciatura da parte dei Log a disposizione. In quest'ultimo caso è necessaria la massima collaborazione del Cliente e del Partner, con attività che possano aiutare a capire l'origine del problema, in modo da circoscriverlo in qualche modo, fino a trovare la soluzione più opportuna.

PRESUPPOSTI UTILI PER LA SEGNALAZIONE AL SUPPORTO ON LINE

- **Aggiornare all'ultima versione**
- **Riorganizzare tutti gli archivi, soprattutto degli anni in linea nel caso in cui l'errore si manifestasse nei menu contabilità e magazzino**, in quanto necessaria al ricalcolo di alcuni importi/quantità (es."Valori Bolle/ordini in corso")
- **Assicurarsi che l'errore sia provocato esclusivamente dall'Applicazione, cioè verificare se replicando la stessa funzione/operatività (a mano e/o con le funzioni del Gestionale) il problema si replica**
- **Verificare correttezza/coerenza dei dati trattati dalle funzioni coinvolte**, poiché sono fra le cause le più frequenti: client copiati senza setup, dati importati e quindi provenienti da sorgenti/strumenti non correttamente allineati alla base dati che vuole il Gestionale, codifiche sporche o incoerenti ai parametri di configurazione del Gestionale, CF/PIVA incoerenti alle codifiche dei conti, numerazioni ridondanti, chiavi di archiviazione duplicate, gestione di input a video che provocano loop, ecc...; controllare che non si stia effettuando operazioni non consentite (trasformazioni/modifiche a documenti, password, chiavi, parametri,..)
- **Debuggere e circoscrivere l'errore nel codice per identificare la funzione coinvolta (cosa e quando) e le condizioni d'uso**
- **Da un browser accedere al Server per aumentare il Livello di Log a "dettaglio buffer trasferiti"** (nella barra di ricerca del browser inserire <nome macchina>:<porta mxserver + 1>, per esempio miocpu:9001) **poi replicare l'errore ed annotarsi l'ora e la data in cui si è manifestato**
- **Esportare l'azienda e l'applicazione e consegnarli all'assistenza on-line insieme ai Log (eventualmente del registro degli eventi del Sistema Operativo del Server)**
- **Nella segnalazione è utile indicare:** se prodotto cloud, presenza di shell/remote, spxremote o simile, la voce di menu in cui si è manifestato l'errore, la sequenza dei tasti premuti ed il contenuto delle operazioni eseguite dalla funzione in uso; esempi: tipo di conto? tipo di articolo? tipo di documento? tipo di operazione eseguita? automatismi presenti? quali altre funzioni coinvolte (esempio: archiviazioni documentali, database, lotti e matricole, distinte basi)? Natura e provenienza di eventuali dati importati? come sono importati? ...)

Diversamente, si può solo sperare di indagare tramite la combinazione operativa fra la macchina e quei dati elaborati, ma che potrebbe anche non replicarsi più oppure replicarsi solo su una determinata macchina, con quel sistema operativo di quella versione, con quella configurazione dell'applicativo, con quei dati aziendali, con un certo tipo di condizioni riguardante il contesto che solo quell'utente ha come, per esempio, le condizioni della rete. Nell'attesa che il problema si manifesti definitivamente oppure non si manifesti più.

Messaggio: Errore non correggibile:

```
....Errore x_readfile(): Errore Handle
....Errore x_esisfile(): Errore Create File
....Errore x_tremote_esisfile(): Errore...
.....
.....
.....Il File è utilizzato da un altro processo
.....Impossibile accedere al file
.....Impossibile aprire il file Accesso negato (sistema operativo) Riferimenti: mx44.exe form1() pclib1.c Create File()
.....Impossibile leggere il file, Record fuori dai limiti, Riferimenti mxlib1.c getbuf_trov()
```

L'errore non correggibile ("syser") con prefisso "x_" è spesso di competenza SISTEMISTICA. Le elaborazioni SPRIX nelle varie configurazioni e situazioni sono l'evento provocante l'errore; in realtà si tratta della combinazione di più elementi: Rete, Sistema Operativo Client/Server, Device, accessi alle Risorse e loro stato, fino all'operatività dell'utente, alle configurazioni del Gestionale, tipo di operazioni e di informazioni elaborate dal Server.

Oltre ai suggerimenti previsti in **PRESUPPOSTI UTILI PER LA SEGNALAZIONE AL SUPPORTO ON LINE** si aggiungono altri suggerimenti su cause che possano provocare bug non collegati al "normale utilizzo" del file (*):

- verificare i permessi su percorsi, file, programmi, sorgenti/eseguibili Sprix/Collage; (es. attributi di "sola lettura")
- verificare la presenza di disconnessione in funzione di invio/ricezione in remoto (tremote_);
- verificare che non siano rimasti bloccati e indisponibili file e eseguibili; (es. impedimenti ad accedere a file ".tmp" in dati\temp, disabilitare antivirus sulle cartelle di interesse, dai dati Passepartout ai files *.tmp)
- verificare che non siano rimasti appesi processi bloccanti i percorsi; terminarli e/o riavviarli, da Server e/o dai Servizi del S.O.;
- rinnovare i programmi mx44.exe, mx45.exe e librerie pxlib.dll copiandoli dalla stessa versione; (nel caso fossero rovinati o non aggiornati correttamente)
- ricompilare i sorgenti rovinati; in Collage, può capitare che la sequenza del Collage sia ostacolata da uno solo di essi, il cui file eseguibile possa avere dei problemi: in tal caso, disattivarli in sequenza e procedere "per esclusione" in modo da scoprire quale fra essi sia da ricompilare; in vecchie installazioni, il file Sprix000 corrotto doveva essere recuperato dalle copie di sicurezza;
- verificare che non vi siano stati eventuali passaggi da Linux a Windows (in tal caso, se vi sono delle aperture di file esterni, provvedere anche alla chiusura di tali file -non solo l'apertura- poiché Linux consente di tenerli aperti, Windows no)
- verificare se è possibile ottimizzare l'output nelle elaborazioni ed evitare un'eventuale sospensione sessione
- Si ricorda che le istruzioni esisfile, copyfile, killfile, ecc... potrebbero consentire altre soluzioni/work around

(*) L'apertura di file già aperto ha effetti diversi in base al tipo di file; in caso di ".xls" o ".csv" non è rilasciato il messaggio "file già aperto", ma x_esisfile() create file ed eventuale chiusura forzata del Client, in quanto EXCEL esegue controlli differenti sui file .xls o .csv. Non è previsto un controllo non bloccante e nemmeno una apertura in modalità di "sola lettura" dei file già aperti.

Errore non correggibile. Riferimenti mx44.exe/form1()/form2()/form3()/form4()/form5()/form6()/form7()/form8()/form9()/ecc. importa(): Attesa(R)

Sono errori relativi al FORM (formato file del programma Sprix/Collage); il numero di form si riferisce al modulo in cui l'errore form si presenta (Sprix Aziende, Sprix Contabilità, Sprix Magazzino, Sprix Produzione, Sprix Stampe, Sprix Annuali, Sprix Servizi, Collage, Libreria). Per esempio, il gestionale cerca di aprire un record che non trova o di importare un file sorgente di sprix. Possibili cause sui file: il sorgente ".ss" è corrotto oppure alcuni eseguibili ".exe" sono rovinati. Suggerimenti:

- ripristinare una copia di backup del sorgente
- sostituzione dei file eseguibili ".exe" (rinnovarli sostituendo i programmi mx44.exe e mx45.exe della medesima versione);
- se consentito, esportare il contenuto sorgente su file di testo e reimportarlo (nota: si perdono i ciclitabellari, righe di stampa e il contenuto della funzione dei parametri di esecuzione [F6]); l'errore form non consente di utilizzare/importare il file come sorgente (al massimo è importabile come testo), perché corrotto o non utilizzabile (es. le righe di codice NON iniziano tutte col carattere 'R', identificativo di riga seguito da altre relative informazioni).
- si ricorda che il livello di sviluppo in SPRIX/MDS non influenza su questo tipo di errore.

Errore non correggibile. Riferimenti mx44.exe editor_app

Tipo form(), bini(), bina(), sbinl(), sbinf(), ecc...descrizione errore: numero troppo grande, stringa troppo lunga, ecc...

Questi errori hanno origine di varia natura, comunque riferibili a valori non consentiti oppure non corretti, che sporcano i campi del gestionale con dati eccedenti o non conformi rispetto a quanto richiesto o consentito dai medesimi campi.

Esempi: valori anomali nei progressivi delle anagrafiche; operazioni di import/export su file; corruzioni dell'indice delle chiavi dei file; import di sorgente esportato con un nome file eccedente il consentito; dimensionamenti errati dell'array.

Messaggio: errore non correggibile cckey() di chiave da cancellare non trovata

Significa che la chiave è presente sul file delle chiavi primarie (es. chiave 1) ma non sugli altri file delle chiavi (es. chiave 2). Questo può accadere se con DELKEY si eliminano record del file a chiavi e contemporaneamente si eliminano gli indici di tale record presenti sui file chiave. Rimedio: sprix di servizio per riorganizzare le chiavi (OPEN "K*"). Si ricorda che la riorganizzazione, inoltre, cancella le chiavi duplicate.

Editor: messaggio "Livello di formato non accessibile Ripetere l'immissione"

Il sorgente potrebbe avere una quantità di righe superiore al limite consentito, oppure essersi rovinato a seguito di una compilazione non terminata correttamente, oppure non essere allineato nel tipo di file (es. installazioni fra ambienti con diversi sistemi operativi, utilizzando comandi di copia di tipo non idoneo al tipo di file).

Messaggio: Numero di variabili di sistema non valido (numero variabili system). Ricompilare

La versione di compilazione ha più istruzioni di quella di esecuzione, ovvero, si sta cercando di eseguire Sprix/Collage su una versione del gestionale precedente (più vecchia) a quella che ha compilato il programma Sprix/Collage.

Messaggio: SS_BSPIXSOR

Indica che il programma è uscito per errore di sistema in presenza di un semaforo acceso (es. sorgente sprix aperto, stampa su file, ...); (SS_Semaforo stringa, SPIX Sprix, SOR Sorgente). Al ripristino del programma viene indicato qual'era il semaforo in questione e di riorganizzare gli archivi collegati.

Messaggio: LA LICENZA D'USO DI QUESTO PROGRAMMA NON CONSENTE L'ESECUZIONE DI SPRIX OLTRE LA DATA ...

Il controllo è effettuato sulla data di sistema e sulla data di ingresso del programma. La presenza di istruzioni con parametri data (es. APRIAZIE) o cicli tabellari con condizioni su date possono rilasciare questo messaggio. Se il messaggio è rilasciato per vecchie installazioni (ad es. effettuate a fini di copia sulla stessa macchina) il problema si risolve utilizzando il codice contratto dell'installazione ufficiale

Caso: compressione File alternativo al impaccamento di FIELD.

SHELL su file.bat con comando esecuzione del compressore + parametri di esecuzione (v. es. www.memecode.com/docs/winzip.html). Dimensione file (alternativa a LOF per non aprirlo): file.bat con comando (ad esempio tipo dir pippo.txt >"%1" 2>"%2" ... oppure uno specifico comando di lettura dimensioni) e uno START del file bat con contenuto relativo a file di output e errore (.\\ppp.bat .\\ppp.out .\\ppp.err), c'è comunque qualcosa da interrogare e leggere con OPEN.

Caso: import da/export su file Xls/Mdb

Il Gestionale dispone di funzioni standard di trasmissione archivi nei formati ASCII, CSV, ... In Sprix e Collage è possibile sfruttare istruzioni "SQL ..." per accedere ad un DBMS Access/Excel e per interrogare il database esterno accedendo ai suoi dati in lettura/scrittura. Scrivendo il risultato di una query sulle variabili strutture archivi (ad esempio "AR... ") si possono leggere/scrivere i dati del Gestionale con il risultato dell'interrogazione. Si ricorda che questo metodo richiede buona conoscenza di SQL e del Gestionale

Esempio SQL_CONNECT: Lettura di record da un database

Le istruzioni che utilizzano il prefisso "SQL" vengono utilizzate per accedere ai dati del database. In particolare se si volesse leggere il contenuto di una tabella i cui campi sono Codice (con valori numerici) e Descrizione (con stringhe alfanumeriche) si potrebbe eseguire uno Sprix come quello che segue:

```
DSN$="temp" '(nome del profilo DSN che deve essere stato preventivamente creato).
STRSQL$="SELECT CODICE, DESCRIZIONE FROM Tabella1"
SQL_CONNECT 1,DSN$,"", "" :GOSUB GEST_ERR_SQL
SQL_ALLOCSTMT 1,1 :GOSUB GEST_ERR_SQL
SQL_BINDCOL 1,1,1,C1 :GOSUB GEST_ERR_SQL
SQL_BINDCOL 1,1,2,C2$ :GOSUB GEST_ERR_SQL
SQL_EXECDIRECT 1,1,STRSQL$ :GOSUB GEST_ERR_SQL
```

```
ESTRAI_DATI:
SQL_FETCH 1,1
IF _ERRSQL_NDF=1 THEN GOTO FINE
PRINT STR$(C1,0,0)+" "+C2$
GOTO ESTRAI_DATI
```

```
FINE:
SQL_FREESTMT 1,1 :GOSUB GEST_ERR_SQL
SQL_DISCONNECT 1 :GOSUB GEST_ERR_SQL
END
```

```
GEST_ERR_SQL:
IF _ERRSQL_N=0 AND _ERRSQL_S$="" THEN RETURN
PRINT STR$( _ERRSQL_N,0,0)+" "+_ERRSQL_S$+" "+_ERRSQL_M$
END
```

Casi: gestione della stringa nella query

Per le query con stringa SQL superiore a 255 caratteri spezzare la variabile in più stringhe (es. STRSQL1\$ + STRSQL2\$) oppure eseguire SQL_PREPARE per inserire più variabili concatenate direttamente da lei (esempio SQL_PREPARE 1,1,STR_SQL1\$,STR_SQL2\$).

Le istruzioni SQL devono essere compatibili con la base dati di riferimento (per esempio Excel non consente le delete di record); le istruzioni

SQL vanno eseguite nel corretto ordine di sequenza, altrimenti sono rilasciati messaggi "function sequence error" o si può incorrere in errori della procedura (es. liberando lo statement senza alcuna allocazione di memoria).

Le stringhe SQL eseguite da Sprix via ODBC richiedono accorgimenti: la sintassi deve essere idonea al tipo di data base, in quanto non tutte le stringhe sono compatibili per tutti i database. Ci possono essere delle differenze.

Per query con stringa contenente dei caratteri speciali, se consentito è opportuno simulare il carattere speciale con CHR\$(); per esempio l'apice (') è simulato con doppio codice ASCII CHR\$(39), es. "L'Aquila" → STRSQL\$="INSERT INTO Tab (Cod,Desc) VALUE (1,'L'+CHR\$(39)+CHR\$(39)+'Aquila')". In caso di lettere accentate nei nomi tabella controllare l'interpretazione di CHR\$(), se è errata causa un errore di esecuzione dello statement.

Nel gestionale la virgola è separatore di decimali, mentre nelle query di scrittura numerica in SQL è rappresentata dal punto (la virgola separa i campi): la seguente query di inserimento in tabella del valore 2,5 nel Campo1 e del valore 1 nel Campo2 - INSERT INTO Tabella(Campo1,Campo2) VALUES (2,5,1) - rilascerebbe un errore "[Microsoft][Driver ODBC Microsoft Access] Il numero dei valori nella query non corrisponde a quello dei campi di destinazione". Occorre quindi trasformare il separatore da virgola a punto [Esempio1: NUMS="1,15" : NUM1\$="" : FOR I=1 TO LEN(NUMS) : IF MID\$(NUMS,I,1) = "," THEN NUM1\$=NUM1\$+"." ELSE NUM1\$=NUM1\$+MID\$(NUMS,I,1) : NEXT; Esempio2: NUMS="1,15" : POS=INSTR(NUMS,",") : IF POS>0 THEN NUM1\$=LEFT\$(NUMS,POS-1)+"."+RIGHT\$(NUMS,LEN(NUMS)-POS) ELSE NUM1\$=NUMS]. Il database Access riconosce il campo numerico e converte la stringa in numero, esempio :

```
DSN$="TEMP" '(nome del profilo DSN da creare preventivamente)
NUM$="2,5" 'stringa con virgola
STR_SQL$="INSERT INTO PrzArt(Articolo,Prezzo) VALUES('merce','"+NUM$+"")"
'STR_SQL$="INSERT INTO PrzArt(Articolo,Prezzo) VALUES('merce','2,5')"
SQL_CONNECT 1,DSN$,"", "" : GOSUB ERRORI_SQL
SQL_ALLOCTMT 1,1 : GOSUB ERRORI_SQL
SQL_PREPARE 1,1,STR_SQL$ : GOSUB ERRORI_SQL
SQL_EXECUTE 1,1 : GOSUB ERRORI_SQL
SQL_FREESTMT 1,1 : GOSUB ERRORI_SQL
SQL_DISCONNECT 1 : GOSUB ERRORI_SQL
END

ERRORI_SQL:
IF _ERRSQL_N=0 AND _ERRSQL_S$="" THEN RETURN
--- IF _ERRSQL_M$="" THEN RETURN
PRINT STR$(_ERRSQL_N,0,0)+" "+_ERRSQL_S$+" "+_ERRSQL_M$
END
```

Anche MySQL separa i decimali con il punto e quindi la presenza della virgola azzer/tronca i decimali e provoca errori nell'applicativo del tipo "errore non correggibile (nome_modulo/tipo_errore) binf(): numero troppo grande". MySQL consente la modalità "Don't use setlocale" da attivare nel profilo ODBC (odbc.ini) con la voce OPTION = 256, in modo che il driver cambi le impostazioni del separatore decimali e quindi non interferire con l'applicativo chiamante. Nota: questa impostazione del separatore decimali non riguarda l'impostazione del separatore in Windows ed impostabile tramite pannello di controllo. Si ricorda che il database MySQL non è certificato né ufficialmente supportato, si consiglia di non eseguire query direttamente (SQL_EXECDIRECT), ma piuttosto di prepararle (SQL_PREPARE) ed eseguirle (SQL_EXECUTE).

Per il parametro di condizione LIKE, diversamente da Access (che usa MSJET) in ambiente ODBC (standard ANSI) i caratteri jolly "*" e "?" vanno rappresentati da "%*" e da "%_". Query su Foglio di calcolo Excel: essendo la prima riga di Excel un'intestazione di colonna il cui contenuto è il nome della variabile inseribile all'interno della SELECT, per definire la tabella di riferimento (FROM), utilizzare la notazione CHR\$(96)+NomeFoglio\$+CHR\$(96).

In alcuni casi è importante verificare che il contenuto della variabile che si sta leggendo dal database al fine di registrarla negli archivi del gestionale sia compatibile fra i medesimi archivi. La presenza di caratteri "blank" (ASCII 0) nelle variabili, infatti, può essere la principale causa di incompatibilità fra gli stessi archivi all'interno del gestionale la cui riorganizzazione segnala incongruenze e cancella i dati che non riesce a trovare o che considera errati.

Esempio. In un archivio anagrafico è già presente un dato con valore CODICE_ANAGRAFICA_SPEDIZIONE\$="ABCD123" (codice di indirizzo di spedizione) e si vuole acquisire dal database un documento che riporta il medesimo codice di indirizzo spedizione (CODICE_DOCUMENTO_SPEDIZIONE\$="ABCD123") al fine di registrarlo nell'archivio documenti del gestionale. La registrazione va a buon fine e non segnala alcuna incongruenza, ma alla prima riorganizzazione degli archivi viene segnalato che il dato è non trovato o errato, pertanto viene cancellato.

Per quale motivo? Apparentemente i codici sembrano equivalenti, ma potrebbe non essere vero!

In realtà, la lettura dal database potrebbe essere avvenuta con dei caratteri blank, ovvero CODICE_DOCUMENTO_SPEDIZIONE\$="ABCD123 ", pertanto, il codice che Sprix è riuscito a registrare nel documento non corrisponde a quello della relativa anagrafica. Nel gestionale il carattere blank ha un valore, di conseguenza, si rende necessario un preventivo trattamento della stringa (ripulire) prima di eseguire la registrazione.

In caso di problemi come questo (o anche per caratteri sporchi, speciali, ecc...) si consiglia di verificare in **DEBUG** la corretta rappresentazione e/o corrispondenza dei caratteri delle variabili di struttura o di dizionario che si stanno considerando.

Le istruzioni Sprix **NON SEMPRE** controllano che il dato inserito in un archivio sia congruo/corrispondente ad un altro che sia già presente in un altro archivio; eccetto specifici e stringenti controlli voluti da Sprix/Gestionale, la verifica di correttezza del dato (specialmente se proviene dall'esterno) spetta alle considerazioni del programmatore, in quanto il dato potrebbe essere considerato libero/nuovo da Sprix per eventuali ulteriori trattamenti e/o processi. La PUT nell'archivio "B", pertanto, non parte dal presupposto di eseguire anche una verifica di congruità del dato attraverso una GET (in automatico e background) nell'archivio "A", poiché questo rientra nel compito/autonomia del programmatore.

COMPUTE di RecordSet.

La gestione è per singolo RecordSet e non è consentita la COMPUTE SUM di più RecordSet. La soluzione può essere quella di "spezzare" la query effettuando due execute: la prima per ottenere la lista dei record (il dettaglio dei singoli record); la seconda per effettuare le totalizzazioni desiderate mediante le funzioni di aggregazione sui domini (SUM, COUNT, AVG, MAX, ecc.) e la clausola GROUP BY

Casi:

In caso di duplicazione del record:
dopo SELECT e FETCH non inserire VIMSG e poi test _ERRSQL -> prima test _ERRSQL e poi VIMSG.

Messaggio di Errore :[Microsoft] [ODBC SQL SERVER] ... Il contesto del database è stato sostituito con 'nome database':
rappresenta un WARNING dato da ODBC che può avvenire in fase di connect a database SQL SERVER e che non influisce sulla CONNECT. Occorre quindi prenderne atto e proseguire nell'elaborazione.

Messaggio: "ODBC SQL Driver" "La connessione è occupata da un'altra statement" (e mancata elaborazione dati).
Causa del messaggio: contemporanea esecuzione di più select sulla stessa connessione tramite driver ODBC per RDBMS Microsoft SQLServer (non altri, esempio ACCESS, Mysql,...) per eseguire differenti statement SQL sulla stessa connessione attiva. Il Gestionale si limita a richiamare le funzioni native ODBC, pertanto occorre eseguire differenti SQL_CONNECT al database per ciascun statement contemporaneamente aperto.

Esecuzione contemporanea di più statement su SQL Server.
Eseguito più statement (query) contemporaneamente su SQL Server utilizzando come driver:
[SQL Server] per connettersi a SQL Server 2000 / \MSDE
[SQL Native Client] per connettersi a SQL Server 2005
[SQL Server Native Client 10.0] per connettersi a SQL Server 2005

L'esecuzione della seconda o successiva statement con la prima ancora in esecuzione il driver ODBC restituisce il seguente errore:
_ERRSQL_S\$ = "S1000"
_ERRSQL_M\$=" [Microsoft][SQL Native Client]La connessione è occupata dai risultati di un altro comando"

Se il sistema è in inglese: _ERRSQL_M\$="[Microsoft][ODBC SQL Server Driver]Connection is busy with results for another hstmt"

Esempio di codice:

```
SQL_CONNECT 1,...
SQL_ALLOCSTMT 1,1
QUERY_1$ = "Select ...."
SQL_PREPARE 1,1, QUERY_1$
  SQL_BINDCOL 1,1,....
....
SQL_EXECUTE 1,1

SQL_FETCH 1,1

SQL_ALLOCSTMT 1,2
QUERY_2$ = "Select ...."
SQL_PREPARE 1,2, QUERY_2$
  SQL_BINDCOL 1,2,....
....
SQL_EXECUTE 1,2
SQL_FETCH 1,2
.....
```

All'esecuzione della seconda SQL_FETCH si presenta l'errore.
Per aggirare questo limite del driver si consiglia di eseguire le successive query su un'altra connessione nel modo seguente:

```
SQL_CONNECT 1,...
SQL_ALLOCSTMT 1,1
QUERY_1$ = "Select ...."
SQL_PREPARE 1,1, QUERY_1$
  SQL_BINDCOL 1,1,....
....
SQL_EXECUTE 1,1

SQL_FETCH 1,1

SQL_CONNECT 2,...
SQL_ALLOCSTMT 2,1
QUERY_2$ = "Select ...."
SQL_PREPARE 2,1, QUERY_2$
  SQL_BINDCOL 2,1,....
....
SQL_EXECUTE 2,1
SQL_FETCH 2,1
.....
```

SQL_CONNECT e SQL_FREESTMT chiudono improvvisamente il client desk oppure generano un errore interno violazione protezione di memoria

A seguito di implementazioni sui nuovi strumenti di sviluppo, potrebbero introdursi anche maggiori controlli sulle istruzioni SQL fino a provocare blocchi della procedura. Questo potrebbe avvenire nei casi in cui si invochi una SQL_FREESTMT dopo la SQL_CONNECT, anziché fare anticipare la liberazione dello statement da un all'allocazione della memoria (SQL_ALLOCSTMT), operazione comunque consentita in quanto non vi sono espresse indicazioni contrarie per evitare equivoci interpretativi dello sviluppatore in base alle sue condizioni d'opera ed in riferimento allo standard delle API ODBC Microsoft (per esempio, in base alla presenza di procedure "one-shot" oppure "loop" nella trasmissione dati in e/o out).

ODBC: connessione a unità in rete

ODBC su profilo DSN di sistema per unità mappata connessa in rete, all'esecuzione della SQL_CONNECT rilascia il seguente errore: [Stato S1009][Nativo -1023] [Microsoft][Driver ODBC Excel] "(sconosciuto)" non è un percorso valido. Assicurarsi che il nome del percorso sia corretto e di essere collegati al server in cui si trova il file.

Nei programmi client-server è un limite frequente, poiché l'utente del servizio non ha i diritti sulla condivisione oppure non ha il profilo che possa vedere l'unità mappata. Nemmeno l'accesso con percorso assoluto "\\...\cartella di lavoro" (ovvero ciò che dovrebbe essere usato anziché la mappatura) può risolvere in quanto non gradito nella configurazione ODBC Microsoft.

Potrebbe risolvere solo l'implementazione di un reindirizzamento su client con propri driver (esempio implementando una "client-connect" o una "sql-client")

Caso: profilo ODBC su DSN di sistema e Driver Connect dinamica

Alcuni moduli (MexalDB, Docuvision e AD) non usano profili ODBC ma la DRIVER CONNECT, ovvero una connessione dinamica che evita di dover gestire il profilo ODBC in quanto basta stabilire (in menu Moduli - Configurazione) il Server del database, cioè il nome macchina o indirizzo IP del server dove risiede il database.

In alcuni casi obsoleti, può essere richiesta una connessione ODBC, effettuata con DSN (data source name), cioè il nome dato alla connessione al data base, stabilito in pannello di controllo \ strumenti amministrazione \ Origini dati (ODBC) in cui indicare il DSN di Sistema che punta al database con un determinato driver.

La configurazione ODBC è infatti stata mantenuta per consentire la compatibilità nei vecchi sistemi, in particolare per la DBMIRROR e le istruzioni SQL in SPRIX, Questa configurazione ODBC lavora su DSN di sistema, nelle macchine client che la gestiscono.

Tale profilo di connessione <DataSource\$> è il nome o codice della sorgente dati per la connessione al driver ODBC per puntare al database di riferimento. Il profilo ODBC deve essere sul Server in cui è installato Mexal (e non una stringa di connessione).

Il profilo deve essere funzionante, indipendentemente dalla sua configurazione. Mexal non gestisce l'ODBC bensì lo utilizza se configurato correttamente. Per le verifiche di funzionamento, basta provare ad usarlo con altri strumenti che non siano SPRIX o MEXALDB.

Se è impostato il profilo con il driver impostato in Mexaldb viene segnalato che il driver non è corretto, poiché in SQL Server occorre selezionare il driver appropriato a 32 / 64 bit.

La gestione di unità in rete (sconsigliata) è sempre condizionata e soggetta a gestione di permessi e utenti.

Installazione Driver a 64 bit per Microsoft Access

Nei sistemi operativi Windows a 64 bit, in cui non è installato il programma Office a 64 bit, non è presente il driver ODBC di Access a 64 bit. Di conseguenza, se si utilizza l'architettura di Passepartout a 64 bit è necessario installarlo manualmente nel sistema operativo della macchina dove risiede il server di Passepartout.

Per verificare la presenza del driver sul sistema posizionarsi sulla scheda Driver di questo pannello:

```
C:\Windows\system32\odbcad32.exe --> ODBC 64 bit se Mexal a 64 bit
C:\Windows\syswow64\odbcad32.exe --> ODBC a 32 bit se Mexal a 32 bit
```

E verificare la presenza della entry "Microsoft Access Driver ...".

Il pacchetto "Microsoft Access Database Engine 2010 Redistributable" si può scaricare dal sito della Microsoft al link <http://www.microsoft.com/it-it/download/details.aspx?id=13255> nella versione a 32 o 64 bit ed installare sul server.

Si ricorda che i file *.mdb creati con il driver "Microsoft Access Driver (*.mdb, accdb)" non sono retrocompatibili con il driver "Microsoft Access Driver (*.mdb)"; la compatibilità è invece garantita nel caso opposto.

NOTA: in presenza di Office 2007 o 2010 a 32 bit l'installazione del driver a 64 bit potrebbe fallire. Si può tentare di installarlo aprendo un prompt dei comandi come amministratore e lanciando il comando:

```
AccessDatabaseEngine X64.exe /passive
```

Messaggio: Errore Driver ODBC non trovato o Errata corrispondenza di architettura tra il driver e l'applicazione

Il Gestionale con architettura a 64 bit richiede DSN a 64 bit e non a 32 bit.

Nei sistemi windows a 64 bit, Vista, Server 2008 R2, Windows 7, Windows 8 e Server 2012, (XP e Server 2003 non son più supportati), i programmi compilati a 32 bit accedono ai profili odbc a 32bit.

Purtroppo l'ODBC Manager di Microsoft che si trova fra gli strumenti amministrativi crea profili solo per programmi compilati a 64 bit e quindi non utilizzabili dai moduli di Mexal quali ADP commerciale, DBMirror ed istruzione SQL_Connect di Sprix. Si provocano problemi di connessione non aperta.

Per creare DSN per programmi a 32 bit occorre utilizzare ODBC Manager che si trova nella cartella:

```
C:\Windows\SysWOW64\odbcad32.exe (installazione windows di default)
```

Si consiglia di creare un collegamento sul desktop denominandolo "Origine Dati (ODBC) 32 bit" per distinguerlo da quello che si trova fra gli strumenti amministrativi.

I DSN creati con questo programma saranno visti da tutti moduli di Mexal che utilizzano connessioni ODBC.

Se il DSN viene creato con Origine dati a 64 bit si riscontrano errori tipo:

```
[Microsoft][Driver Manager ODBC] Errata corrispondenza di architettura tra il driver e l'applicazione nel DSN specificato
```

E nei file di log xdb si troveranno righe simili alle seguenti:

```
28/03/13 09:44:16.966 360354fvWiAE [!] START (AVI)
Funz: DatiConnect,1, Variabile: pMsg
Nome: Sconosciuto 0;DSN: provaeti;UTE: sa;Server: ;Porta: 0;Driver: ;ServerGetI
nfo: ;NomeGetInfo: ;DriverNameGetInfo: ;DriverODBCVerGetInfo: ;DbmsVerGetInfo:
;InitString: ;Odbc ini: ;TipoDbAperto:
28/03/13 09:44:16.982 360354fvWiAE [!] START (AVI)
Funz: IsOk,1, Variabile: err
[Stato IM014][Nativo 0] [Microsoft][Driver Manager ODBC] Errata corrispondenza
di architettura tra il driver e l'applicazione nel DSN specificato
```

Messaggio: [unixODBC][Driver Manager]Data source name not found, and no default driver specified. Sistema Windows su linux via jdbc.

Verifica congruità impostazione DSN, cioè che nel pxclient.ini (es. sezione [XDB] DSN=) non sia impostato un DSN inesistente (verificabile dai file di configurazione presenti in /etc/odbc.ini e /etc/odbcinst.ini).

Messaggio: parametri di connessione al database non specificati in ambiente Linux a 64 bit

Il messaggio in oggetto si verifica in installazione Mexal in ambiente Linux a 64 se vi sono installti lo unixODBC e il driver ODBC per MySQL a 64 bit. In questo ambiente, per utilizzare i moduli di Mexal che si connettono ad un RDBMS è necessario installare i driver a 32 bit e la versione a 32 bit dello unixODBC. Sono stati testati i pacchetti:

- unixODBC-2.2.12-198.17.x86_64 fornito con SuSE 11.1
- unixODBC-2.2.12-198.17.i386 fornito con SuSE 11.1
- unixODBC-2.2.11-7.1.x86_64 fornito con Red Hat 5.3
- unixODBC-2.2.11-7.1.i386 fornito con Red Hat 5.3
- unixODBC-2.2.14 scaricato dal sito www.unixodbc.org

Utilizzando i pacchetti forniti con il sistema operativo è necessario installare sia la versione a 64 bit che quella a 32 bit, in caso contrario non funziona lo unixODBC; se si utilizza il pacchetto scaricato dal sito dello unixODBC è sufficiente compilarlo a 32 bit aggiungendo gli appositi flag nel comando ./configure (CFLAGS=-m32 LDFLAGS=-m32 CXXFLAGS=-m32). In quest'ultimo caso anche i comandi odbcinst e isql saranno a 32 bit come quelli effettuati da Mexal, mentre con i pacchetti forniti con il sistema operativo tali comandi risultano sono a 64 bit.

Anche il connector di MySQL da installare è quello a 32 bit, quindi il pacchetto i386 e non quello x86_64.

Caso: configurazione Moduli tramite MxDbSrv remotizzato

Per accedere dal Gestionale installato in ambiente Linux ad un database gestito da Microsoft SQL Server (2005/2008) oppure Access occorre installare il programma mxdbsrv.exe in configurazione "remotizzata", cioè su un PC con sistema operativo Windows, tipicamente lo stesso dove risulta installato anche l'RDBMS a cui si vuole accedere. In questa configurazione tutti i moduli, MexalDB, Docuvision, ADP B&B, ADP Commerciale, Dbmirror e/o Sprix accederanno al database tramite il programma mxdbsrv.exe e quindi tramite l'odbc di Windows e non più tramite lo unixODBC.

Questa soluzione sostituisce di fatto il pxjdbc che permetteva di accedere agli RDBMS di Microsoft dall'ambiente Linux.

Per attivare la remotizzazione del server mxdbsrv.exe occorre procedere in questo modo:

- 1) Sulla macchina Windows dov'è installato SQL server (o MS Access) creare una cartella, chiamata ad esempio Remote_DB_Server e copiare al suo interno il programma mxdbsrv.exe della stessa versione del Mexal installato in Linux (nel dvd si trova nella cartella mxi_asincdb\prog\win32).
- 2) Sempre in questa cartella con un editor di testo creare un file che si chiama mxdbsrv.ini con il seguente contenuto:

```
[MxSrvStartup]
MaxConn=100

AddrMxServer= <-- inserire l'indirizzo ip del server linux
PortMxServer= <-- inserire la porta d'ascolto del mxserver installato in ambiente linux ad esempio 9000

PortAsincDB= <-- impostare la porta d'ascolto tcp del mxdbsrv, ad esempio 9300

ServiceID=MxDbSrv

[Log]
WriteLog=0

[Installazione]
DirTemp=C:\Remote_DB_Server\Temp
```

- 3) Creare la cartella temp all'interno della cartella creata al punto 1.
- 4) Creare un utente nel sistema operativo con diritti di amministratore, ad esempio mexaladm con relativa password.
- 5) Concedere a questo utente il privilegio di agire come servizio.
- 6) Installare mxdbsrv come servizio del sistema operativo, da prompt di comando:

```
cd C:\Remote_DB_Server
mxdbsrv.exe --install --serviceid=MxDbSrv --user=mexaladm --password= PASSWORD
```

- 7a) Per il modulo MexalDB occorre creare in SQL Server un database, ad esempio MexalDB, ed un utente con i privilegi dbowner nel db creato.
- 7b) Per il modulo Docuvision occorre creare in SQL Server un database, ad esempio Docuvision, ed un utente con i privilegi dbowner nel db creato.
- 7c) Per il modulo ADP B&B occorre creare in SQL Server un utente con ruolo "dbcreator".

- 7d) Per ADP Commerciale, per ogni azienda che utilizzerà questo modulo occorre creare un database, ad esempio ADPCOM_SIGLA, un utente con i privilegi "dbowner" nel db ed eseguire l'apposito ddl.
- 7e) Per i moduli dbmirror e sprix occorre creare in SQL Server un database, ed un utente con i privilegi dbowner nel db creato.
- 8a) Per il modulo MexalDB creare il DSN per accedere ai database creati al punto 7a.
- 8b) Per il modulo Docuivision creare il DSN per accedere ai database creati al punto 7b.
- 8c) Per il modulo ADP B&B non serve creare alcun DSN.
- 8d) Per il modulo ADP Commerciale creare il DSN per accedere ai database creati al punto 7d.
- 8e) Per i moduli dbmirror e sprix creare il DSN per accedere ai database creati al punto 7e.
- 9) Avviare il servizio installato al punto 6.
- 10) Sul server linux arrestare il server mexal, # /etc/init.d/mxserver stop
- 11) Eseguendo il setup.sh attivare MexalDB e/o Docuivision inserendo il DSN creato al punto 8 e l'utente e password del punto 7. Attenzione: il setup permettere di impostare come RDBMS solo MySQL, quindi si accetta questa tipologia che poi si cambierà successivamente. Alla fine del setup non avviare mxserver.
- 12) Modificare il file pxconf.ini: individuare la sezione XDB e apportare i seguenti cambiamenti:

```
[XDB]
PortAsincDB=9003
AsincronoDB=1      <-- impostare 2
RaggruppaAnni=1
RaggruppaAziende=0
TipoRP=1008      <-- impostare 1005 (codice SQL server) oppure 1004 (codice Access)
Tipo=1008        <-- impostare 1005 (codice SQL server) oppure 1004 (codice Access)
GestDbSeparati=0
TipoODBC=1
AddrAsincDB=     <-- inserire questa chiave con l'indirizzo IP del server windows dove è installato l'RDBMS
DBSyncRemoto=1   <-- inserire questa chiave con valore 1
MaxElementiBO=32000 <-- inserire questa chiave con valore compreso fra 1024 e 65000, valore consigliato 32000
```

- 13) avviare il server Mexal, # /etc/init.d/mxserver start
- 14) Dall'apposita funzione configurare i dati di accesso all' RDBMS per i moduli ADP B&B, ADP Commerciale, Dbmirror e/o sprix.

Attenzione: ricordarsi sempre di aggiornare mxbsrv.exe installato nella macchina Windows quando si aggiorna il server Mexal nella macchina Linux.

Caso: Connessione database Errore protocollo - Hdr - Il programma verra' terminato.

Questo si verifica quando c'è un altro servizio che è in ascolto e risponde al posto dal server mxbsrv.exe.

Messaggio: Connessione database 10061 No connection could be made because the target machine actively refused it. Il programma verrà terminato

Quando si cerca di accedere al gestionale con MexalDB attivo, se appare il messaggio in oggetto significa che l'RDBMS a cui si collega MexalDB per la replica dei dati non è avviato o non è raggiungibile, oppure, che il programma della replica asincrona dei dati, mxbsrv.exe, non è avviato.

Il più delle volte la causa è il mancato avvio di mxbsrv.exe. Verificare il processo in esecuzione con il Task manager. Se non c'è occorre arrestare il servizio server MexalBPServer (mxserver.exe), verificare che effettivamente si sia arrestato e poi avviarlo e poi verificare che sia avviato anche mxbsrv.exe. Se non si avvia mxbsrv.exe consultare i file di log, mxserver [ANNO_MESE_CORRENTE].log e mxbsrv.log, ed eventualmente inviarli all'assistenza per una verifica. Verificare anche che mxbsrv.log non superi 1,9GB ed in tal caso rinominarlo.

Messaggio: [microsoft] [Driver ODBC Microsoft Access] "(sconosciuto)" non è un percorso valido

Con la funzione Sprix SQL_CONNECT, passando a parametro un profilo DSN ODBC che utilizza driver MSAccess, la variabile _ERRSQL_M\$ può tornare il seguente errore: [microsoft] [Driver ODBC Microsoft Access] "(sconosciuto)" non è un percorso valido. Assicurarsi che il nome del percorso sia corretto e di essere collegati al server in cui si trova il file.

Tale situazione si presenta quando il server MXI_SERVER viene avviato come servizio. Alcuni dispositivi di rete mappati come unità disco o stampante possono non essere raggiunti dal servizio MXI_SERVER. Stessa casistica è riscontrabile a connessioni di altre funzioni similari come DBMIRROR, ADP, PAGHE, MexalDB, Docuivision, ecc.

Le cause dipendono dal modo di operare di MSWindows e l'associazione di lettere-driver a percorsi di rete in convenzione UNC (*). Nel caso specifico, se il profilo DSN MSAccess riferenzia un file .mdb collocato su una unità di rete mappata da lettera, il server MXI_SERVER può non essere in grado di raggiungere il file indicato nel profilo. Per risolvere il problema si suggerisce di referenziare il file .mdb nel profilo DSN utilizzando la convenzione UNC indicando il percorso assoluto nella forma \\nomemacchina\percorso\file.mdb.

(*) In riferimento all'Articolo Informativo 180362 pubblicato da Microsoft, quando MSWindows avvia un servizio non realizza le associazioni lettera-driver = percorso UNC. Questo avviene anche se il servizio si avvia all'interno di una sessione di utente diverso da LocalSystem. Tali associazioni non sono relative al nome dell'utente che le ha create ma al SID (Logon Security Identifier) che è un identificatore univoco per sessione di logon. La sessione di logon effettuata dal servizio è diversa dal logon effettuata da console di utente. Pertanto, in virtù del fatto che Microsoft non garantisce la visibilità dei driver mappati ad applicazioni avviate come servizio, si sconsiglia vivamente di utilizzare dal Gestionale l'accesso a risorse unità disco o stampanti mappati da lettera-driver. Il funzionamento delle unità mappate è garantito dal Client del Gestionale. Pertanto, tali unità devono essere mappate sul client e, in tutti i punti del programma in cui si desidera accedere ad esse, impostare un percorso locale al client stesso. Ad esempio, <Z:\nomefile.

Caso: connessione a database MySQL in Linux

Premessa: MySQL non è certificato né supportato, le seguenti informazioni a scopo sperimentale non sono vincolate da responsabilità.

Installare MySql, si possono reperire sia i sorgenti che il compilato al sito www.mysql.com.

Se si sceglie di prendere i pacchetti in formato rpm scaricare i seguenti tre pacchetti:

- mysql-versione.i386.rpm
- mysql-shared-versione.i386.rpm
- mysql-client-versione.i386.rpm

dove al posto della parola versione ci sarà espresso la versione, ad esempio 3.23.49 a -1.

Installato i pacchetti verrà richiesto di cambiare la password di root con il comando:
\$ /usr/bin/mysqladmin -u root -p password 'new password'

Il suddetto comando è errato, la forma corretta è
\$ /usr/bin/mysqladmin -u root password 'new password'

Dopo l'impostazione della password occorre riavviare il server mysql, nel modo seguente:
\$ mysqladmin -p shutdown
\$ /usr/local/safe_mysql &

oppure sfruttando le shell con cui il sistema operativo avvia il server al boot, con la distribuzione di Red Hat, i comandi sono:
\$/etc/rc.d/init.d/mysql stop
\$/etc/rc.d/init.d/mysql start

È buona norma crearsi uno o più utenti che lavorano con il data base ma non lo amministrano, questo si può fare nel modo seguente:

```
$/mysql -u root -p
mysql>connect mysql;
mysql>INSERT into user (Host, User, Password, Select_priv, Insert_priv, Update_priv, Delete_priv, Create_priv, Drop_priv, References_priv,
Index_priv, Alter_priv) VALUES ('IP_ADDRESS', 'UTENTE', password('PASSWORD'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
mysql>exit
$ mysqladmin -p shutdown
$ /usr/local/safe_mysql &
```

dove le parole IP ADDRESS, UTENTE e PASSWORD vanno sostituite con i valori reali, per quanto riguarda l'IP ADDRESS utilizzando il carattere %,

assume il significato da qualunque ip, altrimenti sarà accettata solo se la richiesta di connessione proviene da un host con ip specificato.

Ricordarsi che modificato gli utenti, occorre riavviare il server mysql affinché si attivano.

Esistono dei software per ambiente windows con cui operare, alcuni anche basati su connessioni odbc, il driver odbc per windows si può scaricare dal sito www.mysql.com. Come software client utilizzarne uno a scelta, fra cui quello freeware chiamato DBTools versione 1.0.12 scaricabile dal sito www.dbtools.com che, fra le varie funzioni, ha anche l'importazione di un db da un file di access.

Per connettersi al database mysql dal Gestionale in Linux occorre utilizzare il pxjdbc, per cui occorre avere il Java runtime environment versione 1.2.2 disponibile al sito java.sun.com (jre-1_2_2_008-linux-i386[1].tar.gz), ma non solo, mysql non è corredato di driver jdbc, pertanto, occorre procurarlo da terze parti fra cui quello consigliato al sito di mysql chiamato il mm jdbc driver, che è un applicativo java, mm.mysql.2.0.4-bin.jar da scaricare e copiare nel direttorio dove si trova il pxjdbc.jar. Dopo la configurazione pxjdbc.ini, creando un sezione di nome mysql e definendo il driver e la connection come segue:

```
[mysql]
driver=org.gjt.mm.mysql.Driver
connection=jdbc:mysql://redhat/"nome data base"
dove redhat è il nome della macchina che ospita il database mysql; si poteva anche scrivere il suo indirizzo ip.
```

In seguito, eseguire pxjdbc server nel modo seguente:
`$/jre1.2.2/bin/java -cp .:pxjdbc.jar:mm.mysql-2.0.4-bin.jar pxjdbc`

Da sprix e collage l'istruzione SQL_CONNECT va configurata così:
`SQL_CONNECT 1, "mysql@redhat:5024", "UTENTE", "PASSWORD"`

Per completare la configurazione è comodo che il pxjdbc server venga avviato in automatico al boot del sistema, questo si può fare in modo abbastanza semplice creando una shell di nome pxjserv con le seguenti righe:

```
cd /pxjdbc
/jre1.2.2/bin/java -cp .:pxjdbc.jar:mm.mysql-2.0.4-bin.jar pxjdbc
dove /pxjdbc è il direttorio in cui è installato il pxjdbc
```

Successivamente, in fondo al file /etc/rc.d/rc.local inserire la seguente riga:
`/pxjdbc/pxjserv /dev/tty12 &`

Caso: messaggio di connessione a database "MySQL server has gone away"

Errore generico che si presenta in alcune circostanze: (1) per errore apparati di rete che interrompono il servizio (malfunzionamenti di scheda, cavi, switch, hub ecc...); (2) dopo lungo periodo di inattività (aumentare il parametro `wait_timeout` del server MySQL: inserire nel file di configurazione `/etc/my.cnf` se MySQL è installato in linux oppure `my.ini` se è installato in Windows (il file si trova il `c:\ProgramData\MySQL\MySQL Server 5.x`), nella sezione `[mysqld]` inserire le chiavi `wait_timeout` e `interactive_timeout` espressa in secondi, il default è 28800 = 8 ore entrambe con lo stesso valore. Impostare entrambe a 1200300. Riavviare il server MySQL per rendere attiva questa nuova impostazione); (3) l'errore avviene mentre è in esecuzione la query, cioè con query che durano oltre il tempo di `wait_timeout` (aumentare il `wait_timeout` oppure, meglio, aumentare la memoria assegnata a MySQL per migliorare il tempo necessario ad eseguire le query, pertanto impostare il parametro `innodb_buffer_pool_size` ad un valore compreso fra il 25% e il 75% della ram fisica del pc; prima di fissarlo, occorre considerare la memoria necessaria agli altri programmi da eseguire sulla stessa macchina. Il valore di default di questo parametro è 8 MB, un valore insufficiente per i moduli del gestionale.

Messaggio SERVER SHAKER INTERRUPTO - Rilevato tentativo di muovere il puntatore del file a prima dell'inizio del file (errore #131) (... \dati\pxlog\xdb001.log) Riferimenti: pxlibl.c x_lseek()

Operando con Shaker e moduli su database possono capitare simili segnalazioni. In alcuni casi si verifica quando il file in oggetto (`xdb001.log`) supera le dimensioni consentite, pertanto, rimuovere il file dalla cartella `dati\pxlog`. Il fatto che il file sia diventato grande può dipendere da alcuni fattori:

- 1) Nel `pxconf.ini` nella sezione XDB la chiave `TraceDB` ha valore diverso da 0 per cui nel file `xdbNNN.log` e `xrvsdbNNN.log` vengo logate molte informazioni aggiuntive utili per indagare su problematiche relative ai dati inseriti nel database; in tal caso occorre impostare `TraceDB=0` nel file `pxconf.ini` e rimuovere tutti i file `xdb*.log` e `xrvxdb*.log` dalla cartella `dati\pxlog`.
- 2) Nel file log sono riportati diversi errori, in tal caso occorre esaminare il file per decidere le correzioni più opportune a eliminare la causa dell'errore registrato continuamente nel file di log.

Esempio SQL Datatrek

Datatrek lavora con un database Sqlite. I comandi sono quelli standard di un qualunque Sql. Fa eccezione la connessione. Lavorando con un Database standard da Sprix la sequenza dei comandi da utilizzare è la seguente:

```
SQL_CONNECT ...
per connettersi al DB

SQL_ALLOCSTMT
per preparare lo statement da utilizzare

SQL_BINCOL ....
per preparare le variabili di ritorno di una eventuale select

SQL_EXECUTE... o SQL_EXEDIRECT...
per eseguire i comandi SQL

SQL_FETCH
per leggere il ritorno del comando SQL select.

Eventuali Comandi
SQL_FREESTMT per liberare lo statement
SQL_DISCONNECT per chiudere la connessione
```

In Datatrek la sequenza di operazione è quasi identica. Crea un APP in cui è presente l'oggetto Datatrek ed un collage datatrek o un collage standard o uno sprix, la sequenza delle operazioni da eseguire è la seguente:

- `SQL_DRIVERCONN` dove gli unici parametri richiesti sono il primo tipo di connessione e il secondo numero della connessione che deve essere univoco, se il comando è in collage datatrek, se si esegue invece il comando in un app con datatrek deve essere passato come nome del database quello del **codice dell'elemento DataTrek** (click sull'elemento DataTrek -> campo "Codice")

Nota: in DataTrek è prevista una specifica MexalDB che si attiva in voce di menu Servizi - Configurazioni - Configurazione moduli (Alt+ZCC) - MexalDB per DataTrek; occorre inoltre attivare il flag per la replica su MexalDB (abilitazione dell'app 123456APPLIC con MexalDB per DataTrek) che si trova nella Configurazione dell'elemento DataTrek; infine, aperto il programma DataTrek occorre popolare il database con Azienda - MexalDB - Trasferimento dati per DataTrek (Alt+ALD)

- `SQL_ALLOCSTMT` per preparare lo statement da utilizzare
- `SQL_BINCOL` per preparare le variabili di ritorno di una eventuale select
- `SQL_EXEDIRECT...` per eseguire i comandi SQL
- `SQL_FETCH` che a differenza del comando tradizionale non ritorna i valori nelle variabili create con la `SQL_BINDCOL`, ma subito dopo il comando `SQL_FETCH` bisogna eseguire il comando
 - `DTK_GETCOLONNA par1,par2,par3$,par4` (par1 = numero della connessione)
 - par2 = numero dello statement

par3\$ = nome della colonna della tabella del DB
 par4 = numero della bindcol a cui assegnare il valore

quindi i comandi per liberare lo stantment e chiudere la connessione.
 Vediamo un esempio.

Creato il nostro database ipotizziamo di avere al suo interno una table cosi composta:

```
Nome table = "fantasia"
campo1 type text
campo2 type text
campo3 type text
```

in Sprix/collage DTK scriveremo questo sorgente:

```
SQL DRIVERCONN 4,1,"","","",0,"","","" ` stabiliamo la connessione al DB
IF _ERRSQL_S$ <> "" THEN VIMSG 5,_ERRSQL_S$ + " Connect"
SQL_ALLOCSTMT 1,1 ` prepariamo lo stantment
IF _ERRSQL_S$ <> "" THEN VIMSG 5,_ERRSQL_S$ + " Alloc"
SQL_BINDCOL 1,1,1,MIAVAR1$ ` bindiamo le variabili
SQL_BINDCOL 1,1,2,MIAVAR2$
SQL_BINDCOL 1,1,3,MIAVAR3$
SQL_EXECDIRECT 1,1,"select campo1,campo2,campo3 from fantasia " ` eseguiamo il comando sql
IF _ERRSQL_S$ <> "" THEN VIMSG 5,_ERRSQL_S$ + " Execdirect"
A_LEGGI: ` ciclo per leggere il recordset
SQL_FETCH 1,1
IF _ERRSQL_S$ <> "" THEN VIMSG 5,_ERRSQL_S$
IF _ERRSQL_NDF=1 THEN GOTO A_USCITA
DTK_GETCOLONNA 1,1,"campo1",1 ` il comando ritorna nella variabile MIAVAR1$ il contenuto del campo1
DTK_GETCOLONNA 1,1,"campo2",2 ` il comando ritorna nella variabile MIAVAR2$ il contenuto del campo2
DTK_GETCOLONNA 1,1,"campo3",3 ` il comando ritorna nella variabile MIAVAR3$ il contenuto del campo3
GOTO A_LEGGI:
A_USCITA:
SQL_FREESTMT 1,1 ` libero lo stantment
SQL_DISCONNECT 1 ` chiudo la connessione.
END
```

```
SQL_EXECDIRECT 1,1,"Insert into fantasia (campo1,campo2,campo3) values ('uno','due','tre')"
```

con questo comando eseguiamo una insert

Notare: indipendentemente da una insert, update oppure delete dopo la EXECDIRECT in SQLite occorre fare comunque una SQL_FETCH per applicare le modifiche al DB

con questo sorgente abbiamo eseguito una select all'interno di datatrek

Esempio: SQL_DRIVERCONN e DTK_GETCOLONNA

```
SQL_DRIVERCONN 4,1,"","","",0,"","",""
IF _ERRSQL_S$ <> "" THEN VIMSG 5,_ERRSQL_S$ + " Connect"
SQL_ALLOCSTMT 1,1
SQL_BINDCOL 1,1,1,CAMPO1$
IF _ERRSQL_S$ <> "" THEN VIMSG 5,_ERRSQL_S$
SQL_EXECDIRECT 1,1,"select count(recordid) AS tipes from TBPrima"
IF _ERRSQL_S$ <> "" THEN VIMSG 5,_ERRSQL_S$
LEGGI_RECORD:
SQL_FETCH 1,1:
IF _ERRSQL_S$ <> "" THEN VIMSG 5,_ERRSQL_S$
IF _ERRSQL_NDF=1 THEN GOTO USCITA
DTK_GETCOLONNA 1,1,"tipes",1
LPRINT CAMPO1$
GOTO LEGGI_RECORD:
USCITA:
```

Esempio Sprix MIRRORING

```
_INTER=1: ASCMIRROR "AR_ANAGRAFICHE","",""
IF _ERRASC MIR THEN VIMSG 5, STR$( _ERRASC MIR,0,0): END
_INTER=1: LEGGIDESC=1: DBMIRROR "MM RIGHE"
IF _ERRDBMIRROR THEN VIMSG 5, _ERRDBMIRROR$: END
ASCMIRROR_EXT2 0,"AR_ANAGRAFICHE","C:\a","artic.txt",0,0,0,"SV=1,CI=,CT="
```

Caso MIRRORING

Nei mirroring su Access, inserire nei campi una descrizione univoca e non costituita da caratteri non validi nei nomi di tabella sul database (per creare una tabella per ciascuna videata, riportabile e non), altrimenti viene generato l'errore "impossibile definire il campo più di una volta"

Messaggio: Errore non correggibile: afile_ext(): ...superato limite file aperti

L'esportazione di una serie di tabelle che genera l'apertura superiore al numero massimo di file consentiti, provoca errore. Si rende necessario "spezzare" l'elenco procedendo a step con più procedure sprix.

Caso: possibilità di gestire Indici in tabelle create tramite DBMIRROR

La creazione di indici non viene gestita dalla funzionalità DBMIRROR. La costruzione di indici puo' essere eventualmente realizzata con una procedura esterna e separata dal gestionale sapendo che:

- 1) Passepartout non garantisce nella totalità delle tabelle esportate l'univocità degli indici.
- 2) A fronte di una ri-esportazione (o riorganizzazione archivi) tramite DBMIRROR le tabelle vengono rimosse, e di conseguenza quindi anche eventuali indici.

Si ricorda che la funzione MEXALDB (che contempla la gestione degli indici) è un'alternativa alla funzione DBMIRROR.

La DBMIRROR è un funzione sempre più in disuso. In passato, per risalire alla chiave logica utilizzata dal programma veniva rilasciata della documentazione tecnica a questo riguardo (tabelle, tracciati, diagrammi), ma ormai non più aggiornata. Nella colonna 47 se il campo è valorizzato a 1 significa che la colonna fa' parte della chiave logica, a zero si tratta di una colonna non chiave.

Si ricorda che la struttura dei dati del Gestionale è consultabile nei seguenti modi:

- Consultare con editor di testo il file presente nel percorso \programmi\binloc\dbmirror.dat" dell'installazione
- Le Tabelle e Colonne sono definite nel documento "ElencoColonneMexalDB.xls" allegato al chiarimento soluzione n. 30663 del servizio di assistenza EasyPass accessibile dall'Area Riservata di www.Passepartout.net. I contenuti delle colonne nel file vanno aperte ed estese. Per ulteriori approfondimenti consultare il capitolo MexalDB del Manuale Utente
- I Tracciati CSV sono pubblicati nel chiarimento n. 54452 del supporto on-line. I Tracciati CSV sono verificabili nel menu del Gestionale | Servizi - Trasferimento Archivi - Export/Invio CSV

- ASCII]

- Specifiche dei campi Dizionario nel menu del Gestionale [Servizi - Personalizzazioni - Stampa Dizionario Sprix]

I dati obbligatori sono indicati a video dal Gestionale e le relazioni fra i dati sono descritte nel Manuale Utente

Messaggio: _ERRDBMIRROR errore durante esecuzione script dbmirror.vba con Access 2000

Lo script dbmirror.vba (che serve a creare le tabelle del database con comandi tipo DAO in esecuzione su Access 2000) necessita di alcune impostazioni.

Si ricorda che la funzione MEXALDB (che contempla la gestione degli indici) è un'alternativa alla funzione DBMIRROR. La DBMIRROR è un funzione sempre più in disuso. In passato, per risalire alla chiave logica utilizzata dal programma veniva rilasciata della documentazione tecnica a questo riguardo (tabelle, tracciati, diagrammi), ma ormai non più aggiornata.

Si ricorda che la struttura dei dati del Gestionale è consultabile nei seguenti modi:

- Consultare con editor di testo il file presente nel percorso \programmi\binloc\dbmirror.dat" dell'installazione.
- Le Tabelle e Colonne sono definite nel documento "ElencoColonneMexalDB.xls" allegato al chiarimento soluzione n. 30663 del servizio di assistenza EasyPass accessibile dall'Area Riservata di www.Passepartout.net. I contenuti delle colonne nel file vanno aperte ed estese. **Per ulteriori approfondimenti consultare il capitolo MexalDB del Manuale Utente**
- I Tracciati CSV sono pubblicati nel chiarimento n. 54452 del supporto on-line. I Tracciati CSV sono verificabili nel menu del Gestionale [Servizi - Trasferimento Archivi - Export/Import CSV - ASCII]
- Specifiche dei campi Dizionario nel menu del Gestionale [Servizi - Personalizzazioni - Stampa Dizionario Sprix]

I dati obbligatori sono indicati a video dal Gestionale e le relazioni fra i dati sono descritte nel Manuale Utente

Messaggio: errore di collegamento dbmirror da Pass Linux a SQL Server Windows tramite pxjdbc

In caso di errori di connessione fra il gestionale in ambiente Linux e comunicazioni DB MS-SQL Server ("DSN not found, Driver not found..."), il pxjdbc.ini potrebbe contenere la seguente configurazione:

```
[SQLServer]
driver=com.ms.jdbc.odbc.jdbcOdbcDriver
connection=jdbc:odbc:sqlserver
domain4type=sqlserver
```

Occorre utilizzare il driver della SUN: e la seguente configurazione pxjdbc.ini:

```
[SQLServer]
driver=sun.jdbc.odbc.jdbcOdbcDriver
connection=jdbc:odbc:sqlserver
domain4type=sqlserver
```

Ricostruzione database mexaldb

Effettuando la ricostruzione durante l'elaborazione potrebbe comparire il seguente Errore Interno SP_MMAGVALRIGUM1():Chiave non trovata oppure Errore Interno SP_MMAGQTARIGUM1():Chiave non trovata. Per risolvere occorre eseguire le seguenti operazioni:

- Abilitare MexalDB nei file ini
- Non abilitare la replica in chiaro nei parametri di apertura azienda
- riorganizzare gli archivi di magazzino
- Effettuare l'operazione di servizio "Ricostruz.database mexaldb"

A questo punto è possibile abilitare anche la replica in chiaro.. Per eventuali chiarimenti consultare il manuale al capitolo MEXAL-DB paragrafo AVVERTENZE. In ogni caso ogni qualvolta si abilita mexaldb nei file ini, occorre effettuare la riorganizzazione degli archivi di magazzino, prima di procedere con la ricostruzione del database mexaldb

Caso: perchè i dati non vengono scritti e per aggiornare occorre eseguire la Ricostruzione DataBase ?

Il Client potrebbe segnalare che i dati non vengono scritti su DB e che per aggiornare occorre eseguire Ricostruzione Database. Nelle impostazioni la modalità di trasmissione è stata impostata con "A" asincrona e poi nelle varie tabelle da Configurazione Moduli - MexalDB - la "A" di trasmissione Automatica. Questo inconveniente non aggiorna più i dati su DB. Perché?

Spiegazione: potrebbe essere la riorganizzazione archivi. C'è un messaggio che, a seconda di cosa si risponde, è in grado di modificare la trasmissione da automatica (A) in manuale (M).

Esempio: esecuzione di un programma esterno con SHELL

Qualora si rendesse necessario eseguire un programma esterno al gestionale si può utilizzare l'istruzione SHELL che esegue a livello di sistema operativo il comando. Di seguito viene visualizzato un Collage di esempio che prevede l'apertura del programma MSPAINT qualora venga premuto il tasto F12. In particolare, se presente, verrà visualizzata l'immagine relativa l'articolo attivo in "Anagrafica articoli".

```
ON_612_MADSART:
AZZVARSYS 3
GETATTAR
NOME_F$=sxdir+azsig+"\FOTO\"+_ARCOD$+".JPG"
NOME_PR$="C:\WINDOWS\SYSTEM32\MSPAINT.EXE "
IF NOT ESISFILE(NOME_F$) THEN VIMSG 1,"IMMAGINE NON DISPONIBILE!" ELSE SHELL "START "+NOME_PR$+NOME_F$
END
```

E' importante l'utilizzo di START nella SHELL. Se si esegue SHELL "START ..." Sprix/Collage continuerà ad eseguire in maniera autonoma tutte le istruzioni che seguono la SHELL. Se il parametro START viene ommesso, oppure se si utilizza il parametro "START /WAIT ..." (oppure anche "START /W ..."), allora lo Sprix (o il Collage) resterà in attesa fino a quando la finestra (che viene attivata) non viene chiusa. In tal caso qualora il sistema operativo dovesse avere qualche tipo di problema, ne risentirebbe anche Sprix. Si consiglia sempre e comunque di utilizzare il parametro "START" davanti al comando da eseguire con l'istruzione SHELL. Utilizzando l'istruzione SHELL è anche possibile settare i parametri del programma che si intende lanciare., purchè questi possano essere settati anche al di fuori di Passepartout.

Esempio: esecuzione di un file ".bat" che scrive un file con data e ore minuti secondi (alternativa per i secondi mancanti nel dizionario 'xsor')

Predisporre un file ".bat" contenente il seguente comando dos che scrive un file in percorso "C:\filetime.txt" con l'informazione desiderata

```
cd C:\
echo %date% %time% > filetime.txt
```

Eseguire il seguente Sprix

```
-INTER=1 : _ACCFILEUTE=1
SHELL "<START C:\time.bat" '(esecuzione lato Client Mexal/Passcom)
'oppure
SHELL "START C:\mexalbp\dati\datiaz\sig\datiutente\time.bat" '(esecuzione lato Server Mexal e non funzionante lato Server Passcom o Server Farm)
VIMSG 1, "eseguito"
END
```

Caso: esecuzione SHELL START su Server genera mancata chiusura completa di una sessione: possibile blocco nuove sessioni e blocco riavvio servizio

In seguito all'esecuzione via sprix di comandi SHELL sul server, in particolare seguita da comando START, se il processo lanciato non termina correttamente, una sessione potrebbe non chiudersi totalmente. Lato utente, all'esecuzione fine lavoro si vede la maschera bloccata quando in realtà il processo applicativo ha regolarmente chiuso. A fronte di tale condizione, possono verificarsi possibili ulteriori scenari anomali:

- rimanendo tale (qualora non risulti terminabile dall'Amministrazione) la connessione appesa blocca operazioni che richiedono l'assenza assoluta di sessioni attive, tipicamente l'aggiornamento (Live o Liveupdate),
- il blocco di nuove connessioni, con possibile messaggio ritornato al client "Connessione terminata, risorsa al momento non disponibile";
- tentando di riavviare il servizio, il nuovo processo MxServer.exe non riesce ad accettare nuove connessioni poiché il vecchio mantiene ancora la porta occupata in virtù della connessione pendente. In tal caso, occorre riavviare tassativamente il sistema server.

Ad ogni modo, fino al riavvio servizio, le connessioni ancora in essere permangono regolarmente operative. Come indicato inizialmente la causa del problema è la mancata chiusura di processi cmd.exe invocati da sprix con indirizzamento server. Per sbloccare la situazione in caso si presenti il problema occorre terminare forzatamente cmd.exe. Per evitare di entrare in questa condizione bisognerebbe limitare le shell server ed accertarsi che in qualsiasi condizione possano portare a termine la procedura e chiudersi autonomamente. In ogni caso, se dovesse accidentalmente verificarsi questa situazione, per sbloccarla non occorre il riavvio del sistema bensì risulta sufficiente terminare il processo cmd rimasto appeso. Infatti, quanto sopra evidenziato a fronte di SHELL START sul server, non risulta un'anomalia Passepartout bensì un comportamento strutturale del sistema che mantiene appeso il processo cmd e blocca la procedura di uscita lato MxServer benché il vero e proprio processo applicativo sia terminato.

Caso: SHELL di esecuzione programmi lato Server

Non è possibile avviare sessioni di tipo Terminale, come ad es. editor di testi o applicazioni ancor più generiche, che utilizzano il video (per es. ls -la produce output video che non può essere visualizzato sul client, per maggiori informazioni consultare il capitolo relativo a SPRIXREMOTO). La possibilità di avviare shell su Server, va sfruttata solo per shell/programmi di natura batch. Tutto ciò è indipendente dal sistema operativo Server usato.

Caso: SHELL con comando START e parametri

Per influenzare il comando START, utilizzare i parametri che sono consentiti dal tipo di sistema operativo e che sono visualizzabili eseguendo uno START /? nell'esecuzione da prompt (cmd). Esempio /W /MIN /I /B etc...

L'esecuzione di una SHELL seguita dal comando start non può essere utilizzata con l'opzione start /b. Tale parametro infatti entra in conflitto con la normale esecuzione del programma.

Caso: SHELL in Linux

La SHELL in ambiente Linux non esegue il reindirizzamento (<,>,>>) dei comandi. Per poterli effettuare occorre costruire una shell (ad esempio sprix.sh sotto la directory prog dell'installazione), inserire qui dentro i reindirizzamenti, e lanciare quest'ultima sempre tramite l'istruzione SHELL. Per verificare la possibilità di esecuzione di un comando: controllare che il comando sia correttamente referenziato (percorso completo), che il comando esista e sia eseguibile dall'utente di sistema. Una semplice prova può consistere nel eseguire un comando che funzioni con certezza, in modo da essere sicuri di dover correggere il comando o il percorso o le necessarie autorizzazioni dell'utente sul comando. Per esempio: SHELL "/tmp/file.sh" dove file.sh esegue nella directory temp la lista dei file della directory corrente (ls -al /home/mexaladmin > /tmp/file.txt).

Caso: SHELL Linux mail controllo campo destinatario

Controllo su finestra di stampante email configurata (indirizzo=!). Verificare la possibilità consentita dal sendmail di linux e istruzione mutt (con accodamento dei messaggi senza attesa dell'effettivo invio delle mail da parte di sendmail). SHELL "echo TESTO | mutt -a/home/mexaladm/mexal/laser/allegato.pdf -s'OGGETTO' array_email".

Caso SHELL Linux mail oggetto allegato

```
B$="a.b@pass.sm"
C$="b.c@pass .sm"
CMD$="cat /home/mexaladm/mexal/dati/contenuto.txt "+CHR$(124)+" mutt -s "+oggetto mail"+" -a /home/mexaladm/mexal/dati/allegato.pdf -b "+B$+ "
"+C$
SHELL CMD$
```

Esempio: Esecuzione dello Sprix "spx101.es" (SHELLSPRIX e _ARGCMD\$)

Dato lo Sprix "spx101.es" con il seguente codice contenuto: VIMSG 1, _ARGCMD\$: END

Esecuzione di spx101.es tramite un altro Sprix che ha il seguente codice:

```
ARG$=CHR$(34)+"Pass 123"+CHR$(34) 'se senza blank ARG$="Pass123"
CMD$= sxdft + " -t0 -o0 -x01032009 -aAZI -pspx101 -v"+ARG$ ' oppure semplicemente CMD$="-pspx101 -v"+ARG$
SHELLSPRIX CMD$
END
```

Esecuzione di spx101.es tramite un Collage che ha il seguente codice:

```
ON_612_ALL:
ARG$="Pass123"
CMD$=" -t0 -o0 -x"+STRDATA$(sxdft,3,"")+" -a"+azsig+" -v"+ARG$+" -p"
SHELLSPRIX CMD$+"#101"
END
```

Caso: variabile per passare l'argomento da Sprix a Collage

Non è disponibile una variabile per rendere disponibile a Collage ciò che proviene da Sprix, occorre un supporto (file/database) in cui riversare risultati che Collage può utilizzare. Spesso, infatti, si cerca di lanciare uno Sprix da Collage per modificare il record in uso (e bloccato) da Collage, provocando l'errore sul record (articolo, documento, etc.) in uso dall'utente X sul terminale Y, per cercare di trasmettere la modifica al medesimo record. Occorre fare eseguire questa modifica al Collage che ha in carico quel record oppure farlo eseguire da Sprix solo dopo che è stato sbloccato il record. I metodi sono vari, sia attraverso un file di appoggio elaborato da Sprix e poi gestito da Collage sul record (attraverso _SINF\$), oppure gestione di eventi Collage fuori dal blocco del record, per esempio, ON_612_MADSART blocca il record, ON_ENDAR: o ON_PUTAR o ON_612_MASCEART ecc.. sono eventi che sbloccano il record.

Esempio, dato lo Sprix 302 che modifica l'aliquota iva al 22% : ARG\$=_ARGCMD\$: GETAR ARG\$,0 : _ARIVA\$="22" : PUTAR 0

Il seguente Collage non incontra il blocco

```
ON_612_MASCEART:
CODART$=CVA$(SCREEN$( _NFIN,GETPOSURS(2),3,4),16) : ARG$=CHR$(34)+CODART$+CHR$(34) : SHELLSPRIX "-pspx320 -v"+ARG$
END
```

Il seguente Collage non incontra il blocco sul record

```
ON_ENDAR: GETATTAR : CODART$=_ARCOD$ : SHELLSPRIX "-pspx320 -v"+CODART$ : END
```

Casi: SHELLSPRIX oppure esecuzione BATCH ?

Premessa importante: le seguenti dissertazioni riguardano una raccolta storica di casistiche relative ad argomenti SHELL e BATCH e dei quali non è garantita l'attuale validità in relazione alle vigenti condizioni tecnologiche del Gestionale e dei Sistemi Operativi, pertanto, si raccomanda di prendere visione di quanto riportato nelle Nozioni Generali contenute nel Manuale Generale dell'Utente in materia di gestione SHELL e BATCH nel sistema Client Server Passepartout.

L'istruzione SHELLSPRIX è una facile operazione SINCRONA utile per contestualizzare immediatamente i risultati dell'esecuzione dello Sprix. Se ciò non è indispensabile si consigliano le operazioni ASINCRONE, cioè differimenti batch dell'esecuzione di Sprix grazie all'attivazione di uno fra i

due eseguibili chiamati "mxrs.exe" e "mxrsbatch.exe": questa operatività è più solida rispetto alle implementazioni ed agli aumenti di complessità del Gestionale e conseguenti controlli su accessi, archivi, tempi di risposta fra processi collegati, ecc; è più opportuna anche perché consente di avvalersi di supporti esterni e/o temporanei (come file, database, ecc.) per eseguire all'occorrenza, quanto necessario fare su altri gruppi di archivi e senza interferire con l'operatività in corso sull'azienda in uso (per esempio: tempi di apertura e chiusura fra le sessioni dei processi padre/figlio).

Per esempio, l'elaborazione temporizzata automatica di un file proveniente da fonte esterna, Sprix è in grado di eseguirla in modalità differita tramite lo Scheduler del Sistema Operativo ("Operazioni Pianificate" Windows, "Cron" Linux) che preveda una riga di comando per l'esecuzione dei suddetti eseguibili al fine di lanciare lo Sprix in oggetto, analogamente a come avviene con SHELLSPRIX e la relativa sequenza di parametri. Grazie a questi eseguibili l'esecuzione Sprix può avvenire con riga di comando da Prompt.

Dalla versione 2020A (v750) è disponibile lo scheduler integrato di Sprix nel menu Servizi - Configurazioni - Operazioni Pianificate. L'operazione pianificata è un'operazione batch senza interazione I/O ed eseguibile solo lato server (non redirezionabile su client con "<"). Per gli ambienti Live, si ricorda di tenere in considerazione il periodo di fermo macchine per il backup.

Esempio di esecuzione di uno Sprix che esegue la duplicazione di un documento OC (senza stabilire una SETPARSOSTMATR in quanto già configurato nei parametri di magazzino la Sostituzione automatica ordini matrice = N). L'esito del file errori è nel percorso "c:\temp\err.txt"

Lo Sprix si chiama "Sprix4" e appartiene all'App "123400APPSPTXT" e contiene il seguente sorgente:

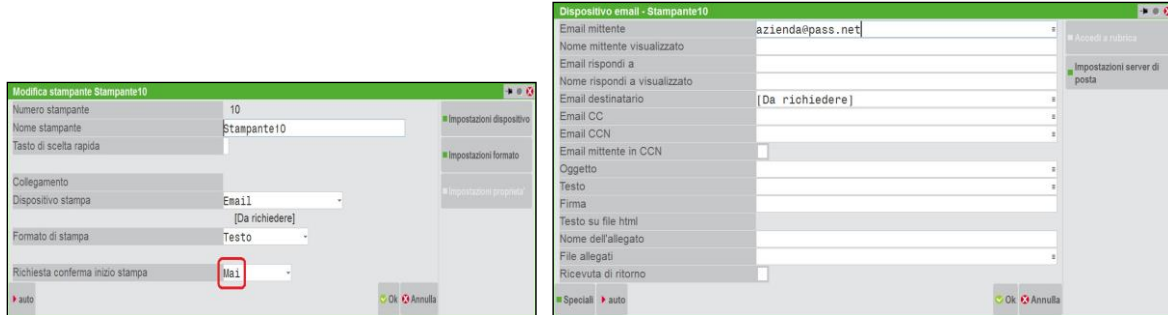
```
GETMM "OC105"
_MNNUM=0
_PUTMM 0
IF _ERRMM$<>" THEN PUTREM_ERR _ERRMM$
```

L'esecuzione di mxrsbatch.exe avviene attraverso un file ".bat" che contiene il seguente comando con una determinata sintassi (apposita per l'esecuzione da un file ".bat"):

```
C:
cd C:\mexalbp\programmi\prog
mxrsbatch.exe --porta=90000 -kADMIN:admin123 -p123400APPSPTXT@sprix4 -aDBL -x01012021 --file=c:\temp\err.txt --filed=c:\temp\data.txt --attendi
```

Esempio di esecuzione temporizzata di invio e-mail: data una stampante e-mail configurata in modo silente (perché l'utilizzo di mxrsbatch.exe non interazioni I/O) e uno sprix che esegue un invio email.

Stampante (10)



Sprix: APRISTAMP_EMAIL 10,"","mario.rossi@impresa.com","","Oggetto","Testo","","", " : LPRINT "prova di stampa" :CHIUDISTAMP_EXT
Operazione pianificata (oppure esecuzione di file.bat) con la seguente riga di comando:
C:\mexalbp\programmi\prog\mxrsbatch.exe --login=ADMIN --password=ADMIN --porta=90000 -p928900EMAIL@Sprix1 -kadmin:admin -aDEM -x3 --file=c:\temp

La SHELL (con eseguibile e relativi parametri) può essere inserita in Operazioni pianificate (Windows) o nello Scheduler (Linux), mxrsbatch.exe risiede nella cartella "prog" del Server ed esegue lo Sprix. In caso di file ".bat" o script shell che eseguono mxrsbatch.exe, questi possono risiedere in qualunque directory a patto che, poi, per eseguire il comando si spostino nella cartella "prog" (del Server o Client, a seconda di dove dev'essere eseguito). Il programma Mxrsbatch.exe deve essere lanciato dal percorso dove risiede, ovvero quella dei programmi del Server. Il programma Mxrs.exe, come punto d'ingresso, si lancia da due contesti: riga di comando esterna e Shellsprix. Nel primo caso è terminale utente, nel secondo caso è un terminale di servizio. Mxrs.exe può stabilire terminali e utenti.

L'utilizzo di mxrsbatch.exe (che peraltro diverrebbe sincrono se impostato col parametro -attendi) può avvenire solo quando sono rispettate le condizioni richieste per l'utilizzo dell'istruzione SPRIXREMOTO, ovvero l'assenza di istruzioni e funzioni con interazioni con il terminale dell'utente, in particolare quelle I/O a video. A differenza di mxrs.exe, l'eseguibile mxrsbatch.exe è un'esecuzione batch silent mode e che NON consente interazioni video. Da questo consegue che: la PUTMM con stampa (PUTMM 2, PUTMM 3, PUTMM 4) non è consentita e viene rilasciato l'errore 'Parametro non valido.|Valori ammessi 0 1 10 11'; tutte le operazioni aprono una finestra nel gestionale non sono consentite (es. conferma di stampa, scelta modulo, accoda o sostituisci file, ecc...); non sono utilizzabili i comandi APRISTAMP, APRISTAMP_EXT che richiedano interazione a video e questo vincolo non è forzabile configurando la stampante in base al parametro di "Richiesta conferma inizio stampa"; è consentita l'eccezione d'uso del comando APRISTAMP_EMAIL e l'utilizzo della stampante di tipo Email, poiché l'invio della mail avviene, ma solo se non c'è interazione a video; la configurazione della stampante deve essere fatta affinché nessuna finestra si apra, compresa quella di gestione di eventuali errori; la gestione dei file può avvenire tramite i parametri:

```
--filed=<file dat> (directory/nomefile dove verrà indirizzato eventuale output derivante dall'istruzione sprix PUTREM DATI)
--filee=<file err> (directory/nomefile dove verranno indirizzati eventuale errori derivante dall'istruzione sprix PUTREM_ERR)
```

Nelle operazioni mxrs.exe in cui viene eseguito anche un contatore a video, ad esempio con un ciclo FOR di PRINT, può capitare di rilevare che il processo termini in anticipo e non sia andato a buon fine in quanto il contatore non termina il suo ciclo. Questo è solo apparente, poiché la scrittura a video è più lenta dell'esecuzione del processo, e quando questo termina (NB: mxrs è simile al processo sprix ma non uguale, perché quando termina chiude la connessione, mentre il processo sprix torna a menu), il flusso di informazioni necessario per disegnare a video potrebbe essere ancora in corso e interrompersi in anticipo. Ciò che sembra essere terminato prima del dovuto, in realtà è già finito. Per sapere se il processo è terminato correttamente, non basta un contatore fittizio (e a volte inutile e/o pesante), ma è più opportuno sapere se il risultato richiesto è stato ottenuto oppure no, e questo, può essere richiesto con un'altra azione, un valore di ritorno, un messaggio VIMSG, una scrittura file, ecc.).

Dalla versione 2020A (v750) è disponibile lo scheduler integrato di Sprix nel menu Servizi - Configurazioni - Operazioni Pianificate. L'operazione pianificata è un'operazione batch senza interazione I/O ed eseguibile solo lato server (non redirezionabile su client con "<"). Per gli ambienti Live, si ricorda di tenere in considerazione il periodo di fermo macchine per il backup.

Le operazioni pianificate (come mxrsbatch) possono funzionare solo con file lato server, percorsi lato client non possono funzionare. Quindi anche se il percorso è di rete ma viene aperto col "<" viene considerato un percorso client e non è accettato (es.Sprix remoto interrotto interfaccia utente non disponibile il file non esiste). In casi di percorsi di rete questi accedono se la macchina in cui è presente un file da raggiungere ci sono i permessi di accesso per l'utente mexal (quindi un utente che deve esistere anche sulla macchina dove risiede il file) oppure se come permessi di accesso ha "everyone". In questo caso potrebbe essere raggiungibile anche senza "<" (simbolo che farebbe raggiungere il file dalla macchina client).

Caso: SHELLSPRIX errore su mxrs.exe e librerie mxlib.dll

Durante l'aggiornamento, il mxrs.exe poteva essere bloccato (da sola lettura, da utilizzo da altro terminale, ... o per altri motivi). E' possibile quindi che quello presente nei programmi non sia allineato alla versione. Oltre alla mxlib.dll, copiare nuovamente anche il mxrs.exe dal cd (e togliergli gli attributi di sola lettura).

Caso: SHELLSPRIX non si avvia per errore x_exesprix().

Non è un'anomalia, significa che la sessione avviata con SHELLSPRIX non è potuta nascere per cui la sessione che fa la richiesta, per non rimanere bloccata e sospesa, dopo un pò esce con l'errore perché sua figlia non la avvertirà mai della sua fine. Il motivo di ciò può essere valutabile nei log (mxserver.log e px.log). Se causato da errore di rete -10054 la cosa si complica: se saltuario può essere provocato da qualsiasi motivo; se invece è frequente occorre diagnosticare la rete (saturazioni? scheda? tempi di latenza eccessivi nelle risposte? ecc...).

Esempio SPXREMOTO\$ ()

Client (Communicator Locale) con indirizzo del suo Server "localhost:11000" vuole elaborare l'articolo "ART" sull'azienda DBL in un'altra installazione su Server Remoto (Communicator Remoto) con indirizzo "100.100.0.100:9000".

```

---> Sprix "spxloc.ss" su Communicator Locale <-----
P2$="ART":R$=SPXREMOTO$("100.100.0.100:9000","-aDBL -t0 -x01012009 -kUTE:PASS
-pspxrem",P2$,"datdpx.txt","errspx.txt",0)
IF R$<>" " THEN VIMSG 1, R$: END

```

```

---> Sprix "spxrem.ss" su Communicator Remoto <-----
B$= ARGCMD$: VAR$="L'argomento inviato per l'elaborazione è " + B$
IF B$="ART_DIVERSO" THEN ERR=1
PUTREM_DATI VAR$: IF ERR<>0 THEN PUTREM_ERR "ERRORE"

```

Esempio SPXREMOTO\$ ()

Per eseguire lo sprix spx101.es in modalità remota sulla medesima installazione: A\$ = SPXREMOTO\$ ("LOCALHOST:9000", "-adbp -x2 -pspx101", "", sxdatt+"dati.txt", sxdatt+"errori.txt") i file dati.txt e errori.txt vengono riempiti dalle istruzioni PUTREM_DATI e PUTREM_ERR contenute nello sprix spx101. Se la variabile A\$ è vuota allora l'istruzione SPXREMOTO\$ è stata eseguita correttamente. Nota: lo sprix spx101 non necessita di parametri per la sua esecuzione altrimenti sarebbero stati inseriti nel terzo parametro (lasciato vuoto).

Esempio di Sprix remoto che esegue quanto richiesto, gestendo l'output sui due file (dati ed errori) mediante le funzioni PUTREM_DATI e PUTREM_ERR.

```

PUTREM_DATI "L'argomento passato dal Communicator Remoto: " + _ARGCMD$
PUTREM_DATI CHR$(13) + CHR$(10)
FOR X=1 TO 10
PUTREM_DATI CIFLET$(X) + CHR$(13) + CHR$(10)
NEXT
' PUTREM_ERR "Output in caso di errori"
PUTREM_ERR "Output in caso di AAAAAAAAAAAAAAAAAAAAA"

```

Esempio SPXREMOTO\$ ()

SPXREMOTO\$ <NomeServer:Porta>,<parametri mxrs>,<parametri SPRIX>,<Nome file dati>,<Nome file errori>,<modalità esecuzione>

Ritorna una stringa contenente l'eventuale errore dell'elaborazione.

Permette di eseguire uno SPRIX in un COMMUNICATOR REMOTO. Il risultato ritorna in un file. Utilizza la comunicazione fra COMMUNICATOR locale e COMMUNICATOR di una installazione remota.

L'esecuzione della funzione SPXREMOTO\$ avviene sul COMMUNICATOR locale. In base al parametro <COMMUNICATOR:Porta> viene contattato il corrispondente COMMUNICATOR remoto.

Il COMMUNICATOR remoto esegue lo sprix in base a quanto definito in <parametri mxrs>; gli eventuali parametri dello sprix da eseguire sono indicati in <parametri SPRIX>.

Lo sprix, che deve necessariamente esistere nell'installazione remota, ritorna il risultato dell'elaborazione sotto forma di file il cui nome deve essere definito in <Nome file dati>.

L'eventuale errore di elaborazione ritorna nel file definito in <Nome file errori>.

Lo sprix, che costituisce un'elaborazione di tipo batch, non deve assolutamente avere interazioni video. In questo caso l'elaborazione dello sprix termina notificando l'errore nel file degli errori <Nome file errori>. Sia il file <Nome file dati> che il file <Nome file errori> vengono creati lato COMMUNICATOR LOCALE.

Il parametro <modalità esecuzione> indica la modalità di esecuzione della funzione SPXREMOTO\$.

L'esecuzione può essere sincrona (parametro = 0) o asincrona (parametro = 1).

Se l'esecuzione è sincrona la funzione SPXREMOTO\$ attende il risultato dell'elaborazione dello SPRIX eseguito dal COMMUNICATOR remoto. Nel caso in cui esista il file degli errori <Nome file errori> ritorna il relativo errore. Al termine dell'esecuzione della funzione SPXREMOTO\$ è disponibile il file risultato dell'elaborazione <Nome file dati> ed eventualmente il file degli errori.

Se l'esecuzione è asincrona la funzione SPXREMOTO\$ non attende il risultato dell'elaborazione dello SPRIX eseguito dal COMMUNICATOR remoto. Essa termina. I file risultato dell'elaborazione saranno presenti lato COMMUNICATOR locale al termine dell'elaborazione dello sprix lato COMMUNICATOR remoto.

Istruzione PUTREM_DATI <buffer>

Istruzione PUTREM_ERR <buffer>

Sono istruzioni associate alla funzione SPXREMOTO\$ e servono per la creazione del file contenente il risultato dell'elaborazione e per l'eventuale creazione del file degli errori.

Esse devono essere utilizzate nello SPRIX eseguito lato COMMUNICATOR remoto.

PUTREM_DATI <buffer>: inserisce nel "file risultato elaborazione" i buffer contenenti le informazioni risultato.

PUTREM_ERR <buffer>: inserisce nel "file errore elaborazione" i buffer contenenti l'errore.

Esempio di SPRIX eseguito lato COMMUNICATOR locale.

'Esempio di SPRIX che richiede la disponibilità di un articolo
'nel magazzino di una sede distaccata dell'azienda
'per mezzo dell'esecuzione di uno SPRIX remoto

```

FNAME$="c:\dispo.txt" 'Nome File in cui verrà scritto il risultato dell'elaborazione
FNAMEE$="c:\dispoerr.txt" 'Nome File in cui verrà scritto il risultato dell'elaborazione
KILL FNAME$
KILL FNAMEE$

```

```

'Codice dell'articolo di cui si vuole acquisire l'esistenza
CODART$="BULL 080"

```

```
'Avvio dell'esecuzione dello SPRIX remoto in modalita sincrona
A$=SPKREMOTOS("192.100.100.10:19000"," -aDBP -t0 -x01012020 -pspx301", CODART$, FNAME$, FNAMEE$, 0)
```

```
GOSUB STAMPA_RISULTATO
```

```
END
```

```
STAMPA_RISULTATO:
IF ESISFILE(FNAME$) =0 THEN GOSUB PRINTERR:GOTO STAMPA_RISULTATO_EXIT
```

```
OPEN "R", 1, FNAME$,1
FIELD 1, 1 AS C$
```

```
NBYTE=LOF(1)
IF NBYTE =0 THEN CLOSE 1:GOSUB PRINTERR:GOTO STAMPA_RISULTATO_EXIT
```

```
RISUL$=""
```

```
FOR I=1 TO NBYTE
GET 1,I
IF C$>=" " AND C$ <="z" THEN RISUL$=RISUL$+C$
NEXT
CLOSE 1
VIMSG 4, "Disponibilita' dell'articolo "+CODART$+": "+STR$(VAL(RISUL$),0,0)
STAMPA_RISULTATO_EXIT:
```

```
RETURN
```

```
PRINTERR:
IF ESISFILE(FNAMEE$) =0 THEN GOTO PRINTERR_EXIT
NF=AUTOFILE()
OPEN "R",NF, FNAMEE$,1
FIELD NF, 1 AS C$
NRBYTE=LOF(NF)
IF NRBYTE=0 THEN GOTO PRINTERR_EXIT
RIGA$=""
FOR X=1 TO NRBYTE
GET NF, X
RIGA$=RIGA$+C$
IF C$=CHR$(10) THEN VIMSG 5, RIGA$:RIGA$=""
NEXT
IF LEN(C$) > 0 THEN VIMSG 5, RIGA$
CLOSE NF
PRINTERR_EXIT:
RETURN
```

```
Esempio di SPRIX eseguito lato COMMUNICATOR remoto.
SPRIX spx301.ss
```

```
'SPRIX eseguito in seguito alla richiesta di un COMMUNICATOR remoto
'e che proietta il risultato dell'elaborazione sullo stesso COMMUNICATOR
'remoto
```

```
'Inizializzazione
CODART$=""
```

```
CODART$=_ARGCMD$ 'Codice dell'articolo richiesto
```

```
IF ardes(CODART$) = "" THEN GOTO ERRORE 'Se l'articolo non esiste
'Se l'articolo esiste
'Letture progressivi di magazzino e calcolo disponibilita'
INV=aqinv(CODART$,1,sxann)
CAR=aqcar(CODART$,1,sxann)
SCAR=aqsca(CODART$,1,sxann)
IMP=aqli(CODART$,1,sxann)
NDISP =INV+CAR-SCAR-IMP
DISP$=RIGHT$(STRING$(20," ") +STR$(NDISP,-1,0),20)
```

```
'PROIEZIONE DEL DATO SUL SERVER REMOTO
PUTREM_DATI DISP$
GOTO FINE
```

```
ERRORE:
'Comunicazione dell'errore
PUTREM_ERR "Articolo non trovato"
```

```
FINE:
END
```

Esempio Porta Seriale (COM)

```
APRI:
OPENCOM "COM2",9600,2,"7",0,60 : IF _ERRCOM$<>0 THEN GOTO ERRORE
DIM C$( ): DIM Q( ): EXIT=0: P=0: J=1
LEGGI_TESTA:
GETCOM 1 : IF _ERRCOM$<>0 THEN GOTO ERRORE
V=ASC(_BUFCOM$): IF V<>ASC("H") THEN GOTO LEGGI_TESTA
LEGGI:
FOR I=1 TO 80 : GETCOM 1: IF _ERRCOM$<>0 THEN GOTO ERRORE
IF _BUFCOM$="" THEN EXIT=1
IF _BUFCOM$="[ THEN GOTO COMP
IF EXIT=1 THEN GOTO STOP
NEXT
COMP:
FOR I=1 TO 80 : GETCOM 1 : IF _ERRCOM$<>0 THEN GOTO ERRORE
V=ASC(_BUFCOM$)
IF V=ASC("]") OR V=ASC("H") THEN C$(J)=LEFT$(C$(J),LEN(C$(J))-1):GOTO FINE
C$(J)=C$(J)+_BUFCOM$
NEXT
FINE:
GETCOM 1 : IF _ERRCOM$<>0 THEN GOTO ERRORE
IF _BUFCOM$="I" THEN GETCOM 1: IF _BUFCOM$ = "N" THEN GOTO STOP
IF _BUFCOM$=CHR$(13) THEN J=J+1:GOTO LEGGI:ELSE Q(J)=Q(J)+VAL(_BUFCOM$)
GOTO FINE
```

```

` FOR J=1 TO I
` IF C$(J)=-_BUFCOM$ AND P=0 THEN Q(J)=Q(J)+1 ELSE C$(I)=-_BUFCOM$:Q(I)=1:P=1
` NEXT
` NEXT
STOP:
FOR X=1 TO J: PRINT C$(X)+" ";: PRINT STR$(Q(X),0,0): NEXT
CLOSECOM

ERRORE:
IF _ERRCOM$<> "" THEN VIMSG 5, _ERRCOM$: END

OPEN...
... VAR$="Sprix"
PUTCOM VAR$,5 : IF _ERRCOM$<>0 THEN GOTO ERRORE
CLOSECOM

```

Casi: porta seriale

Sprix non gestisce librerie esterne (es. librerie Tapi). Con OPENCOM e GETCOM è possibile restare in attesa sulla porta seriale (es. dei comandi tipici del modem generati durante la chiamata). E' necessario comunque gestire il protocollo di comunicazione in Sprix.

OPENCOM non ha parametri di controllo di flusso poiché viene impostato il controllo Hardware.

La funzionalità OPENCOM consente di aprire una porta seriale per una comunicazione, ad esempio con un registratore di cassa. Si consiglia di aprire la porta seriale nel momento in cui s'intende utilizzarla, in quanto molti sistemi operativi una volta aperta, ma inutilizzata, per un certo periodo la impostano allo stato di "dormiente" e mostrano difficoltà a "risvegliarla" in tempi brevi utili all'utilizzo, ad esempio, un per Collage che deve effettuare la stampa di uno scontrino conviene effettuare la OPENCOM all'atto della PUTMM e poi chiuderla con CLOSECOM subito dopo la stampa. La frequenza della trasmissione va regolata attraverso i parametri e (non effettuando ad esempio degli SLEEP), ed ovviamente i protocolli di comunicazione vanno realizzati in funzione di quelli adottati dal registratore.

Non risultano impedimenti di OPENCOM di USB, ma non con apertura generica e diretta di device usb, ma attraverso adattatore usb-seriale il cui collegamento aggiunge una com utilizzabile da sprix (per verificare l'associazione al numero com verificare le proprietà hardware del sistema).

Installazioni in Server Farm consentono al Client l'applicazione del dispositivo Registratore di Cassa attraverso implementazione Sprix/Collage, ma non tramite Shaker. Non è quindi consentito l'utilizzo dell'applicativo java MEXCR. La possibilità Sprix è consentita attraverso il protocollo Xon-Xoff utilizzabile dai più comuni stampanti misuratori fiscali. In assenza di porte COM su cui redirezionare la OPENCOM "<COM" sul Client, è consentito utilizzare l'adattatore USB-COM con apposito driver.

Per verificare la possibilità di colloquio Client-Server è sufficiente fare dei test, ad esempio con un Collage simile, sviluppato per misuratore Custom Kube:

```

ON PUTMM:
A$=" "+CHR$(034)+"PROVA "+CHR$(034)+"1*1437H1R 1440H1T"
OPENCOM "<COM7",9600,0,"8",0,3,"0"
IF _ERRCOM$<>"" THEN VIMSG 1, _ERRCOM$
PUTCOM A$,30
IF _ERRCOM$<>"" THEN VIMSG 1, _ERRCOM$
CLOSECOM

```

Esempio Collage

Dati cinque Collage di cui quattro sul menu 'MD' e uno su AUTO

MD	COLLAGE 01	coll_01
MD	COLLAGE 02	coll_02
MD	COLLAGE 03	coll_03
AUTO	COLLAGE 04	coll_04
MD	COLLAGE 05	coll_05

```

coll_01 (MD) | coll_02 (MD)
-----|-----
ON_OUT_MDCAU: |ON_IN_MDQTA:
STARTCOLLAGE "coll_05" | _GLOBAL(1)= _GLOBAL(1)+1
END | VIMSG 4,"coll_02"
| STOPCOLLAGE "coll_02"
ON_IN_MDRICCOD: |END
CALL "coll_05","ETIC_05" |-----|
END | coll_03 (MD)
|-----|
ON_OUT_MDTDOC: |ON_IN_MDFEZZO:
TIPDOC$=CVMAIU$( _SINP$) | _GLOBAL(1)= _GLOBAL(1)+1
ABILCOLLAUTO "coll_04",1 | VIMSG 4,"coll_03"
END | STOPCOLLAGE "coll_03"
|END
ON_IN_MDCODART: |-----|
IF TIPDOC$="BC" THEN STARTCOLLAGE "coll_02" | coll_04 (AUTO)
IF TIPDOC$="FT" THEN STARTCOLLAGE "coll_03" |-----|
QUANTICOLL=MAXCOLLAGE() |ON_IN_MDSCON:
NOMCOLL$=NOMECOLLAGE$(1) | VIMSG 4,"Collage AUTO"
NUMCOLL=CARCOLLAGE("coll_01") | STOPCOLLAGE "coll_04"
_GLOBAL(1)= _GLOBAL(1)+1 |END
END |-----|
| coll_05 (MD)
|-----|
|ETIC_05:
| VIMSG 4,"routine ETIC_05"
|END
|-----|

```

COMMENTO: in emissione documenti di magazzino carica i collage 01, 02, 03 e 05, di cui il numero 01 è in memoria caricato dall'etichetta ON_OUT_MDTDOC: all'uscita della quale è memorizza il Tipo Documento (es. FT) in TIPDOC\$ e ABILITA il caricamento del collage AUTOMATICO 04 ogni volta che si scatena l'evento sullo sconto in esso contenuto (ON_IN_MDSCON:). Avanzando in uscita dalla causale (ON_OUT_MDCAU:) il coll_01 carica il coll_05 per potere richiamare con CALL la sua etichetta (ETIC_05:) all'ingresso nel campo di ricerca di Cli/For (ON_IN_MDRICCOD:) poiché non sarebbe consentita in quello stesso punto una STARTCOLLAGE "coll_05" dato che le istruzioni Collage sono eseguite all'END dell'evento. Lasciato il campo del Cli/For, all'ingresso del codice articolo (ON_IN_MDCODART:) il coll_01 controlla il valore di TIPDOC\$ per stabilire quale fra collage (coll_02 e coll_03) attivare con STARTCOLLAGE in modo che allo scatenarsi dei rispettivi eventi (ingresso in quantità o prezzo) eseguano le routine per poi scaricarsi dalla memoria con la STOPCOLLAGE; lo stesso coll_01 ricava con MAXCOLLAGE() la quantità di collage attivi che sono caricati in memoria, ricava con NOMECOLLAGE\$(1) il nome del collage caricato per primo e infine con CARCOLLAGE("coll_01") il numero dell'ordine di posizione del caricamento in Collage e incrementa il primo elemento dell'array numerico _GLOBAL(1) così come lo incrementano coll_02 e coll_03 quando sono caricati.

Esempio CALL FTP /FTPS / FTP+TLS

```

REM *** ELENCO FILE ***
_FTPCMD$="DIR" ' COMANDO
_FTPURL$="ftps://demo.wftpserver.com/upload/" ' Protocollo Host [porta] percorso
_FTPUSER$="user" ' Nome Utente
_FTPPASS$="user" ' Password (casualmente uguale a username)
_FTPLOGFILE$="FTP_TEST_SPRIX.log" ' Nome File di Log
_FTPLOGLEV = 1 ' Livello del log

_FTPDIRNOMI = 1 ' Riporta solo nomi, nessun attributo
_FTPNOCRTCHK=1 ' Ignora verifica certificato server

FTPSECCMD ' Esecuzione del comando
IF _FTPCODERR <> 0 THEN VMSG 5,_FTPDESERR$:END ' Se errore visualizza

FOR I =1 TO 1024 ' Per tutti i possibili file
IF _FTPDIRLST$(I) = "" THEN I=9999 ' se raggiunta fine lista, termina
PRINT LEFT$_FTPDIRLST$(I), 70 ' Stampa nome file
NEXT

REM *** GET DI UN FILE ***
_FTPCMD$="GET" ' COMANDO
_FTPURL$="ftps://demo.wftpserver.com/download" ' Protocollo Host [porta] percorso
_FTPUSER$="user" ' Nome Utente
_FTPPASS$="user" ' Password (casualmente uguale a username)
_FTPLOGFILE$="FTP_TEST_SPRIX.log" ' Nome File di Log
_FTPLOGLEV = 1 ' Livello del log
_FTPREMFIL$ = "manual_en.pdf" ' Nome remote del file
_FTPLOCFIL$ = "~manuale.pdf" ' Nome locale del file
_FTPNOCRTCHK=1 ' Ignora verifica certificato server

FTPSECCMD ' Esecuzione del comando

IF _FTPCODERR = 0 THEN END ' Se errore visualizza

IX=1
LAB1:
IF _FTPRESF$(IX) <> "" THEN PRINT _FTPRESF$(IX):IX=IX+1:GOTO LAB1

WKEY:
A=INKEY()
IF A<>501 THEN GOTO WKEY
END

REM *** PUT DI UN FILE ***
_FTPCMD$="PUT" ' COMANDO
_FTPURL$="ftps://demo.wftpserver.com/upload" ' Protocollo Host [porta] percorso
_FTPUSER$="user" ' Nome Utente
_FTPPASS$="user" ' Password (casualmente uguale a username)
_FTPLOGFILE$="FTP_TEST_SPRIX.log" ' Nome File di Log
_FTPLOGLEV = 1 ' Livello del log
_FTPREMFIL$ = "man_drfile.png" ' Nome remote del file
_FTPLOCFIL$ = "]manuale.pdf" ' Nome locale del file
_FTPNOCRTCHK=1 ' Ignora verifica certificato server
_FTPCLRCMDC$="PASSIVE"

FTPSECCMD ' Esecuzione del comando
IF _FTPCODERR = 0 THEN END ' Se errore visualizza

IX=1
LAB1:
IF _FTPRESF$(IX) <> "" THEN PRINT _FTPRESF$(IX):IX=IX+1:GOTO LAB1

WKEY:
A=INKEY()
IF A<>501 THEN GOTO WKEY
END

REM *** DEL DI UN FILE ***
_FTPCMD$="DEL" ' COMANDO
_FTPURL$="ftps://demo.wftpserver.com/" ' Protocollo Host [porta] percorso
_FTPUSER$="user" ' Nome Utente
_FTPPASS$="user" ' Password (casualmente uguale a username)
_FTPLOGFILE$="FTP_TEST_SPRIX.log" ' Nome File di Log
_FTPLOGLEV = 1 ' Livello del log
_FTPREMFIL$ = "upload/man_drfile.png" ' Nome remote del file
_FTPNOCRTCHK=1 ' Ignora verifica certificato server
_FTPCLRCMDC$="PASSIVE"

FTPSECCMD ' Esecuzione del comando
IF _FTPCODERR = 0 THEN END ' Se errore visualizza

IX=1
LAB1:
IF _FTPRESF$(IX) <> "" THEN PRINT _FTPRESF$(IX):IX=IX+1:GOTO LAB1

WKEY:
A=INKEY()
IF A<>501 THEN GOTO WKEY
END

```

SFTP

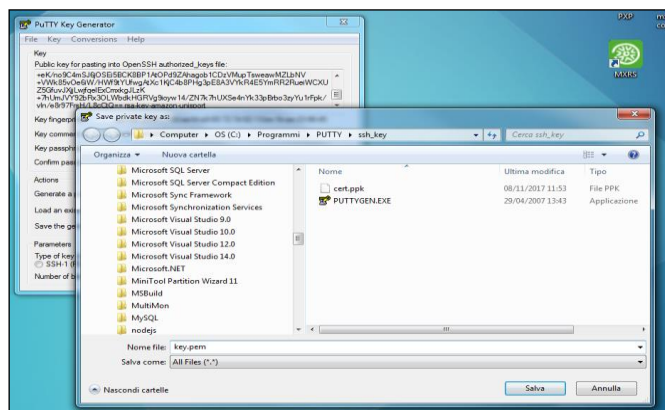
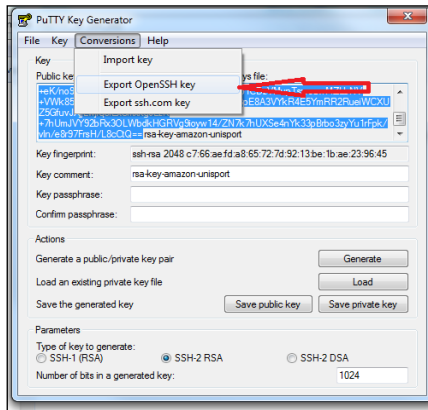
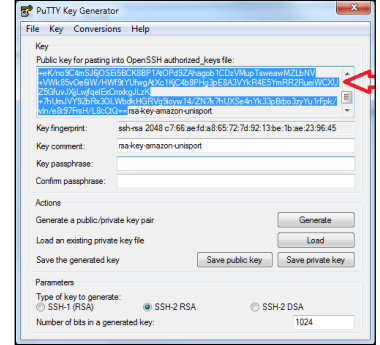
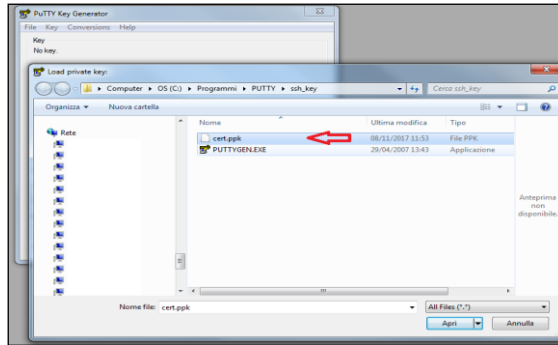
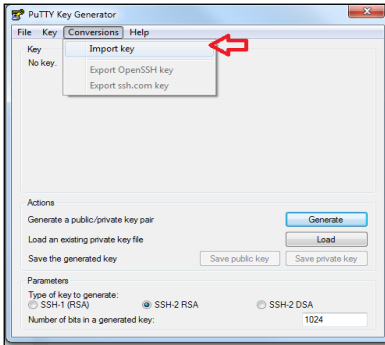
Premessa: per questo Protocollo è consigliato utilizzare installazioni Windows con compilazione del Gestionale in architettura a 64 bit, poiché la libreria libssh2.dll necessita di funzionalità che in talune versioni di Sistemi Operativi a 32 bit non sono presenti (supporto alle funzioni di crittografia). Su Installazioni a 32 bit non essendo previsto l'utilizzo di libssh, cercando di eseguire una richiesta SFTP, si ottiene l' Errore libcurl 1 - Protocollo non supportato dalla libreria usata.

Si raccomanda che esistano le seguenti dipendenze con relativa collocazione delle librerie dll nella cartella prog del Server dell'installazione: la libreria libmxserverssl.dll deve avere le tre dipendenze (verificabili con programma Windows "depends.exe") libcurl.dll, libeay32.dll e sslseay32.dll; la libreria libcurl.dll (versione 7.40.0) deve avere le quattro dipendenze: libeay32.dll, sslseay32.dll, zlib1.dll, libssh2.dll.

Il protocollo SFTP non richiede variabili che contengono informazioni sui certificati SSL/TLS, il file con il repository delle autorità di certificazione (CA) e neppure le informazioni relative al Control Channel in quanto in questo caso tutto il traffico transiterà cifrato su un unico canale logico attraverso SSH. I file con chiave privata (key.pem) e pubblica (pub.key) devono essere indicati in un formato utilizzato da PuTTY o WinSCP.

Le chiavi utilizzate per questi esempi sono state ricavate da un certificato consegnato dal gestore del servizio SFTP; nel caso di Amazon lo si può ottenere da una loro applicazione web (<https://console.aws.amazon.com/ec2/>), il cui accesso è consentito tramite credenziali; nonostante non sia chiaro il formato del certificato (es. cert.ppk), ci sono tool in grado di convertire nel formato usato da libcurl (OpenSSH, Putty, ecc).

Per questi esempi è stato utilizzato il tool PuTTYGEN.exe: avviarlo e nel menu Conversions selezionare "Import key" e scegliere il file cert.ppk fornito (parametri conversione: SSH-2 RSA e Number of bits in a generated key = 1024). Dopo l'apertura del file cert.ppk selezionare il testo nel field "Public key for pasting into OpenSSH..." come indicato e salvarlo nel file pub.key usato nell'esempio; successivamente nel menu "Conversions" selezionare la voce "Export OpenSSH key", confermare la domanda relativa all'assenza della passphrase, quindi salvare la chiave privata nel file key.pem usato nell'esempio:



```

REM *** ELENCO FILE ***
_INTER=0
_FTPCMD$ = "DIR"
_FTPURL$ = "sftp://sftp-eu.amazonaws.com:2222/download/"
_FTPUSER$ = "G6A524PJ6I40"
_FTPPASS$ = ""
_FTPLOGFILE$ = "ftplog.log"
_FTPLOGLEV = 1
_FTPDIRNOMI = 1
_FTPCRTKEY$ = "]key.pem"
_FTPPUBKEY$ = "]pub.key"
_FTPNOCRTCHK = 0
_FTPWRHOST = 1
_FTPKNWNHST$ = "]known_hosts"

PRINT "Avvio connessione"
AZZCRONO(1)

FTPSECCMD
PRINT "Connessione terminata - "+ "Tempo: "+ STR$(CRONO(1),3,0) + " sec."

IF _FTPCODERR <> 0 THEN VIMSG 1, _FTPDESERR$ + "|" + _FTPRESPP$(1) : END
FOR IDX = 1 TO 1024
  IF _FTPDIRLST$(IDX) = "" THEN IDX = 9999
  PRINT LEFT$(_FTPDIRLST$(IDX), 70)
NEXT

PRINT "Premi [INVIO] per terminare:"

A:
A=INKEY()
IF A = 0 THEN GOTO A
END

REM *** GET DI UN FILE ***
_INTER=0
_FTPCMD$ = "GET"
_FTPURL$ = "sftp://sftp-eu.amazonaws.com:2222/download/"
_FTPUSER$ = "G6H504NJ6I40"
_FTPPASS$ = ""
_FTPLOGFILE$ = "ftplog.log"
_FTPLOGLEV = 1
_FTPLOCFILE$ = "]YHS27850.T0000001.dwl"
_FTPREMPFILE$ = "YHS27850.T0000001"

_FTPCRTKEY$ = "]key.pem"

```

```

_FTPUBKEY$ = "]pub.key"

PRINT "Inizio Trasferimento del file: "+_FTPLOCFILE$
FTPSECCMD
IF _FTP CODERR <> 0 THEN VIMSG 1, _FTPDESERR$ + "|" + _FTPRESF$(1) : END
PRINT "Trasferimento Terminato"

A:
A=INKEY()
IF A = 0 THEN GOTO A
END

REM *** DEL DI UN FILE ***
_INTER=0
_FTPCMD$ = "DEL"
_FTPURL$ = "sftp://sftp-eu.amazonsedi.com:2222/"
_FTPUSER$ = "G6A524PJ6I40"
_FTPPASS$ = ""
_FTPLOGFILE$ = "ftplog.log"
_FTPLOGLEV = 1
_FTPREMFIL$ = "download/YHS27850.T0000003"
_FTPCRTKEY$ = "]key.pem"
_FTPUBKEY$ = "]pub.key"

PRINT "Inizio cancellazione del file: "+_FTPREMFIL$
FTPSECCMD

IF _FTP CODERR <> 0 THEN VIMSG 1, _FTPDESERR$ + "|" + _FTPRESF$(1) : END
PRINT "Cancellazione Eseguita"

A:
A=INKEY()
IF A = 0 THEN GOTO A
END

```

Considerazioni sull'utilizzo delle istruzioni FTP da installazioni live.

Nel caso di installazione live è bene ricordare che chi materialmente effettua la connessione è la parte server del gestionale. Ne consegue che, dato che il server live è dotato di IP pubblico, la connessione ad eventuali server FTP presenti all'interno della rete locale dei client iDesk è impossibile quando viene specificato come indirizzo di destinazione quello locale privato. Per poter collegarsi ad un server FTP all'interno della rete locale client è pertanto necessario:

- configurare il NAT sul router internet dell'installazione client;
- inserire come indirizzo IP di destinazione quello pubblico del router client o, in alternativa, un nome DNS valido.

Considerazioni su messaggi FTP.

I comandi inviabili al server FTP sono molti e Sprix ne considera una parte, ovvero: DIR, GET, PUT, DEL e relative variabili che servono a questi scopi.

I messaggi relativi al dialogo Client-Server FTP, alcuni sono creati dal Client, mentre quelli prettamente FTP sono creati dal Server. I messaggi del Server, come "testo" e come "stato", dipendono da come è stato concepito il protocollo in fatto di regole e assunzioni, ovvero, degli "accordi sul protocollo" stabiliti da chi ha realizzato quel Server FTP ed ai quali ci si deve adeguare.

Alcuni comandi, per esempio il cambio directory, passando una path/nome vengono inviati implicitamente analizzando la richiesta, altri invece, su richiesta dell'utilizzatore, gli "stati" di risposta (100->199, 200->299, ..., 500->599) sono elementi decisi dal Server FTP e che, comunque, sono ricevuti da remoto.

Gli accordi sul protocollo cambiano molto in base agli autori, pertanto, per utilizzare FTP deve essere esaminato dettagliatamente il protocollo al fine di riuscire a valutare lo "stato" che il Server invia.

A fronte di un determinato "stato", infatti, potrebbero essere attribuiti anche dei significati diversi rispetto a quelli che "generalmente" ci si aspetterebbe, poiché ciascun Server FTP ha particolari accordi sul protocollo, il Server FTP della Passepartout anche, e cioè, ciascun Server FTP ha le sue regole e le sue assunzioni.

Di conseguenza, chiunque desideri comunicare con un Server FTP (come anche qualsiasi altro Server che utilizzi un determinato protocollo) deve sapere con chi sta comunicando e deve conoscere come deve scrivere il codice in modo appropriato (cioè quanto richiesto) in modo da essere in grado di comunicare con il suo corrispondente.

Per comprendere ciò la documentazione disponibile è molta (es. wikipedia.org/wiki/List_of_FTP_commands) ed il Prompt DOS consente di verificare i comandi FTP e le risposte ottenute connettendosi ai vari server FTP.

Ma è indispensabile conoscerli e sapere come sono stati implementati e di che cosa hanno bisogno per eseguire qualcosa in remoto. Magari anche, di conoscere i "capricci" che sono in grado di fare, per esempio: oggi ti rispondono utilizzando una certa macchina Client e domani no (a causa di un determinato aggiornamento del S.O. della macchina? oppure per modifiche alle regole del protocollo? oppure perché il Server è in manutenzione?), oppure non ti consentono di accedere a determinati percorsi o di cancellare certi file perché la path non è scritta proprio come vorrebbe il Server oppure perché i file hanno proprietà particolari? ecc...).

Una cosa deve essere chiara: sebbene sia comunque implementabile, la libreria che realizza il Client (libcurl) non può conoscere e/o "parare" tutte le condizioni avverabili per consentire a Sprix di essere universale per ogni situazione possibile e immaginabile. Libcurl non può prevedere qualsiasi caso esistente, cioè, non può essere universale e omnicomprensiva, altrimenti l'alternativa sarebbe partire da "RFC" e riscrivere tutto un Client adattivo con i Server FTP ed i loro comportamenti. Anche quelli più strani e non "consuetudinari".

Esempio: apertura, lettura e chiusura di file CSV

Le seguenti sono Funzioni Sprix per leggere file CSV senza creare routine, parser, ecc. Le funzioni di libreria del Gestionale possono leggere i valori di una singola colonna di una riga di un file CSV, pertanto, creando un iteratore si possono scorrere le righe del CSV e, della riga letta, si può chiedere il valore di ogni singola colonna; lo stato dell'iteratore viene raggiunto dalle diverse funzioni comunicando a queste l'Handle associato al file ".CSV" ottenuto in fase di apertura.

ESEMPIO DI FILE CSV DA LEGGERE ("bilue_economici.csv")

```

NKY_NOME_RICL;CKY_RICL;CKY_SEZIONE;CDS_RICL;NGB_LIVELLO;CDS_FORMULA;CDS_FORMRIGA;CKY_NATURA;IFL_PERCENT;IFL_UTILE;
CKY_NATURA_BEP;IFL_DIFF_UNITA;CKY_TAG;IFL_STP_PARZIALE;CDS_CODRSM;IFL_NASCONDI_XBRL;
CDS_CODICE_XBRL;IFL_FOGLIA_XBRL;IFL_ANTE_XBRL;end
;1;E;Valore della produzione;1;;SUM(+F6+F5+F4+F3+F2);R;;;T0015;;;itcc-ci:ValoreProduzione;N;S;end
;2;E;ricavi delle vendite e "delle" prestazioni;2;;R;;;C0015X01;;;itcc-ci:ValoreProduzioneRicaviVenditePrestazioni;S;S;end
;3;E;"variazioni delle rimanenze di prodotti in corso di lavorazione semilavorati e
finiti";2;;R;;;C0015X02;;;itccci:ValoreProduzioneVariazioniRimanenzeProdottiCorsoLavorazioneSemilavoratiFiniti;S;S;end

```

ESEMPIO DI SPRIX: Sessione di apertura, lettura e chiusura di un file CSV posizionato nella cartella "SIG\datiutente"

```

HND=OPENCSV("]file.csv","") ' apertura file in datiutente
IF _ERRCSV<>0 THEN VIMSG 5, _ERRCSV$ : END 'test errore apertura

NUMFLD=CSVITERSTR(T(HND) ' Avvio Iteratore

```



```

IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV$ : HERR=CLOSECSV(HND) : END 'test errore

VAL_FLD1$=CSVROWVAL$(HND,1,"T") 'Lettura prima riga della prima colonna
VAL_FLD2$=CSVROWVAL$(HND,2,"T") 'Lettura prima riga della seconda colonna
VAL_FLD3$=CSVROWVAL$(HND,3,"T") 'Lettura prima riga della terza colonna
VAL_FLD4$=CSVROWVAL$(HND,4,"T") 'Lettura prima riga della quarta colonna

PRINT VAL_FLD1$+" - "+VAL_FLD2$+" - "+VAL_FLD3$+" - "+VAL_FLD4$ 'Stampa valori

NRREC=CSVITERNXT(HND) 'La funzione esegue lo Skip Riga intestazioni (parte con NRREC=2)
' Se il file è dotato di intestazioni vengono restituite come prima riga, si possono usare o saltare
' e proseguire la lettura. Le intestazioni sono una convenzione fatta da chi utilizza quel file,
' ma sono comunque una riga di dati con un certo significato
IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV$: HERR=CLOSECSV(HND):END

'----->>>>
READ_NXT:
VAL_FLD1$=CSVROWVAL$(HND,1,"T") 'Lettura prima colonna
VAL_FLD2$=CSVROWVAL$(HND,2,"T") 'Lettura seconda colonna
VAL_FLD3$=CSVROWVAL$(HND,3,"T") 'Lettura terza colonna
VAL_FLD4$=CSVROWVAL$(HND,4,"T") 'Lettura quarta colonna

PRINT VAL_FLD1$+" - "+VAL_FLD2$+" - "+VAL_FLD3$+" - "+VAL_FLD4$ 'Stampa valori

NRREC=CSVITERNXT(HND) 'Avanza alla prossima riga
IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV$ ELSE GOTO READ_NXT
'<<<-----

HERR=CLOSECSV(HND)

NRREC= CSVITERNXT( HND ) ' Skip Riga intestazioni
IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV$: HERR=CLOSECSV( HND ): END 'TEST
'----->
READ_NXT:
VAL_FLD2$= CSVROWVAL$( HND, 2 ) ' Lettura seconda colonna
VAL_FLD3$= CSVROWVAL$( HND, 3 ) ' Lettura terza colonna
VAL_FLD4$= CSVROWVAL$( HND, 4 ) ' Lettura quarta colonna

PRINT VAL_FLD2$+" - "+VAL_FLD3$+" - "+VAL_FLD4$ ' Stampa valori

NRREC= CSVITERNXT( HND ) ' Avanza alla prossima riga
IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV$ ELSE GOTO READ_NXT 'TEST errore

HERR=CLOSECSV( HND ) 'Chiusura File

```

DESCRIZIONE DELLE FASI DELL'ESEMPIO

1. Apertura file CSV e definizione del carattere separatore

<HND> = OPENCNV(<NomeFile>,<Separatore>)
 <HND> è l'Handle che verrà passato alle altre funzioni per la lettura del file CSV

Parametri: <NomeFile> - [Percorso/]NomeFile CSV da aprire
 <Separatore> - Carattere utilizzato nel file CSV per separare i valori delle colonne

Descrizione: la Funzione apre il file CSV e comunica il tipo di separatore dei campi del file CSV. Quando la funzione termina con successo viene ritornato un valore numerico che è l'handle con il quale si farà riferimento nelle operazioni successive al file CSV. HND è sempre 0 oppure un valore intero qualunque. Il numero di file contemporaneamente apribili valore di 8. L'apertura dei file con la funzione OPENCNV, utilizzano le stesse strutture dati utilizzate dall'istruzione OPEN, per cui occorre che vi sia la disponibilità di HANDLE di file SPRIX per il successo dell'operazione.

In base alla codifica di generazione del file CSV, potrebbe non essere sufficiente il solo Line Feed (\n) bensì potrebbe essere necessario anche il Carriage Return (\r), rappresentabile in Sprix con CRLF\$=CHR\$(13)+CHR\$(10).

Esempio: HND = OPENCNV("bilue_economici.csv", ",") 'apertura file

2. Verifica eventuali Errori

_ERRCSV - codice numerico dell'esito operazione
 _ERRCSV\$ - descrizione dell'esito operazione

La variabile di sistema _ERRCSV sarà settata con il valore 0 (zero) se l'operazione è terminata con successo, altrimenti avrà un valore diverso da zero, quando assume un valore diverso da zero la variabile di sistema _ERRCSV\$ conterrà una descrizione dell'errore. Gli errori possibili _ERRCSV e _ERRCSV\$ sono:

- (1) - OPENCNV() - I parametri non sono stati settati correttamente. I parametri sono mancanti o con valore non valido (Es: separatore lungo più di un carattere)
- (2) - OPENCNV() - Errori vari di accesso al file CSV (Esempio: errori riportati dal Sistema Operativo o dalla Rete, errori dovuti al controllo accesso utente (v. ACCFILEUTE), ecc...)
- (3) - OPENCNV() - Non è possibile aprire un ulteriore file CSV, occorre chiudere uno di quelli aperti. Tentativo avanzare lo stato di un iteratore non avviato
- (4) - OPENCNV() - L'apertura del file CSV non è riuscita
- (8) - OPENCNV() - Non è possibile aprire un ulteriori file, occorre chiudere uno di quelli aperti. Troppi file SPRIX sono stati aperti, occorre chiuderne almeno uno

Esempio: IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV\$: END 'test di errore apertura

3. Avvio dell'iteratore sul file CSV già aperto per la lettura (se termina correttamente acquisisce già la prima riga e ritorna il numero di campi presenti nella riga)

<num_colonne> = CSVITERSTR(<HND>)
 <num_colonne> - è il valore di ritorno in numero delle colonne di dati nelle righe del file CSV, dovrebbe ovviamente essere lo stesso per tutte le righe del file

Parametri: <HND> - Handle ottenuto dalla funzione OPENCNV()

Esempio: NUMFLD = CSVITERSTR(HND) 'avvio iteratore

4. Verifica eventuali Errori

_ERRCSV - codice numerico dell'esito operazione
 _ERRCSV\$ - descrizione dell'esito operazione

La variabile di sistema _ERRCSV sarà settata con il valore 0 (zero) se l'operazione è terminata con successo, altrimenti avrà un valore

diverso da zero, quando assume un valore diverso da zero la variabile di sistema `_ERRCSV$` conterrà una descrizione dell'errore. Quando la funzione termina con successo viene ritornato un valore numerico che è il numero di colonne presenti in una riga del file CSV separate dal carattere separatore comunicato alla funzione `OPENCSV()`. L'esecuzione della funzione `CSVITERSTRT()` acquisisce automaticamente la prima riga di dati del file, se il file prevede una riga con l'intestazione delle colonne, sarà questa ad essere letta. . Gli errori possibili `_ERRCSV` e `_ERRCSV$` sono:

- (1) - `CSVITERSTRT()` - I parametri non sono stati settati correttamente. Il parametro `<HND>` non è valido.
- (6) - `CSVITERSTRT()` - Occorre aprire il file CSV prima di poter avviare un iteratore. Tentativo avanzare lo stato di un iteratore non avviato

Esempio: `IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV$: HERR=CLOSECSV(HND):END ` test`

5. Avanzamento dell'iteratore su file CSV alla prossima/successiva riga

`<nrtest> = CSVITERNXT(<HND>)`
`<nrtest>` - è il valore di ritorno in numero della riga letta

Parametro `<HND>` - Handle ottenuto dalla funzione `OPENCSV()`

`NRREC` ottenuto da `CSVITERNEXT` è semplicemente il numero di riga testè letta (non ha grande utilità, si possono contare anche in autonomia; la prima riga la si ottiene nel momento in cui si invoca la `CSVITERSTRT` che inizializza l'iteratore e si trova posizionato sulla prima riga.

Esempio: `NRREC= CSVITERNXT(HND) `Skip Riga intestazioni`

6. Verifica eventuali errori

`_ERRCSV` - codice numerico dell'esito operazione
`_ERRCSV$` - descrizione dell'esito operazione

La variabile di sistema `_ERRCSV` sarà settata con il valore 0 (zero) se l'operazione è terminata con successo, altrimenti avrà un valore diverso da zero, quando assume un valore diverso da zero la variabile di sistema `_ERRCSV$` conterrà una descrizione dell'errore. Quando la funzione termina con successo viene letta la riga successiva del file e ritornato un valore numerico che è il numero della riga letta. Gli errori possibili `_ERRCSV` e `_ERRCSV$` sono:

- (1) - `CSVITERNXT()` - i parametri non sono stati settati correttamente. Il parametro `<HND>` non è valido.
- (6) - `CSVITERNXT()` - Occorre avviare l'iteratore prima di poterlo avanzare sull'elemento successivo. Tentativo avanzare lo stato di un iteratore non avviato
- (7) - `CSVITERNXT()` - E' stata raggiunta la fine del file. L'ultima riga del file CSV è già stata letta

Esempio: `IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV$: HERR=CLOSECSV(HND): END `test`

7. Acquisizione dei valori delle colonne che interessano (la funzione che legge i valori si occupa anche della gestione dei delimitatori "" eventualmente presenti). La funzione acquisisce il valore dell'ennesima colonna dal record correntemente in memoria

`<valore$> = CSVROWVAL$(<HND>,<NrColonna>,<Trim$>)`

`<valore$>` - Valore di ritorno della colonna richiesta nella riga attualmente in memoria, letta dal file dalle funzioni `CSVITERSTRT()` e `CSVITERNXT()`

Parametri: `<HND>` - Handle ottenuto dalla funzione `OPENCSV()`

`<NrColonna>` - Numero della colonna di cui si desidera acquisire il valore

`<Trim$>` - Carattere che indica come effettuare il "trim" (elimina spazi non significativi) attorno al valore letto.

Assume valori "D", "S", "T" (qualunque altro valore vale come vuoto) ed ha il significato di eliminazione degli spazi (blank, TAB, CR, LF) D=destra, S= SINISTRA, T=TUTTI (sia destra che sinistra del valore):

"D" significa eliminare gli spazi a destra, per cui per es: " AA001 " viene restituito come " AA001"

"S" significa eliminare gli spazi a sinistra, per cui per es: " AA001 " viene restituito come "AA001 "

"T" significa eliminare tutti gli spazi non significativi, per cui per es: " AA001 " viene restituito come "AA001"

Sono considerati spazi i caratteri ; blank -0x20, TAB-0x08, CR-0x0d e LF-0x0a

Se passato vuoto ("") oppure con valori diversi da quelli sopra elencati, il valore viene restituito così come letto dal file.

Esempio:

```
READ NXT:
VAL_FLD2$= CSVROWVAL$( HND, 2 ) ` Lettura seconda colonna
VAL_FLD3$= CSVROWVAL$( HND, 3 ) ` Lettura terza colonna
VAL_FLD4$= CSVROWVAL$( HND, 4 ) ` Lettura quarta colonna
PRINT VAL_FLD2$+ " - "+VAL_FLD3$+ " - "+VAL_FLD4$ ` Stampa valori
```

8. Verifica eventuali Errori o raggiungimento della fine file

`_ERRCSV` - codice numerico dell'esito operazione
`_ERRCSV$` - descrizione dell'esito operazione

La variabile di sistema `_ERRCSV` sarà settata con il valore 0 (zero) se l'operazione è terminata con successo, altrimenti avrà un valore diverso da zero, quando assume un valore diverso da zero la variabile di sistema `_ERRCSV$` conterrà una descrizione dell'errore. Il valore letto sarà il valore al netto della presenza di delimitatori di stringa "" (virgolette) utilizzate per includere il carattere separatore nel valore ed anche le sequenze di escape per inserire anche le virgolette nel valore della colonna. Gli errori possibili `_ERRCSV` e `_ERRCSV$` sono:

- (1) - `CSVROWVAL()` - i parametri non sono stati settati correttamente. Il parametro `<HND>` non è valido
- (6) - `CSVROWVAL()` - Occorre avviare l'iteratore prima di poter acquisire il valore di una colonna. Tentativo di leggere il valore di una colonna ma il file CSV non è aperto

Esempio: `IF _ERRCSV <> 0 THEN VIMSG 5, _ERRCSV$ ELSE GOTO READ NXT `TEST errore`

9. Chiusura del file CSV

`<coderr> = CLOSECSV(<HND>)`

`<coderr>` - valore di ritorno con eventuale codice di errore, lo stesso assegnato alla variabile di sistema `_ERRCSV`

Parametri: `<HND>` - Handle ottenuto dalla funzione `OPENCSV()`

Descrizione: la funzione chiude il file CSV e rilascia le risorse usate dall'iteratore

10. Verifica eventuali Errori

`_ERRCSV` - codice numerico dell'esito operazione
`_ERRCSV$` - descrizione dell'esito operazione

La variabile di sistema `_ERRCSV` sarà settata con il valore 0 (zero) se l'operazione è terminata con successo, altrimenti avrà un valore diverso da zero, quando assume un valore diverso da zero la variabile di sistema `_ERRCSV$` conterrà una descrizione dell'errore. Gli errori possibili sono:

- (1) - `CLOSECSV()` - i parametri non sono stati settati correttamente. Il parametro `<HND>` non è valido
- (6) - `CLOSECSV()` - Occorre almeno aprire il file CSV prima di poterlo chiudere. Tentativo di chiudere un file CSV non aperto

Esempio: HERR=CLOSECSV(HND) 'Chiusura File

JSON

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
JSONCREATE()	Funzione che crea un elemento JSON_OBJECT vuoto. es	S/C/Cr/Cd
JSONADDOBJ()	Par: (<RifObj>, <Key\$>, <RifObjDalns>) Funz. aggiunge un valore JSON_OBJECT all'elemento es	S/C/Cr/Cd
JSONADDSTR()	Par: (<RifObj>, <Key\$>, <ValStr\$>) Funzione aggiunge un valore JSON_STRING all'elemento es ValStr\$ è da passare come codifica Mexal e non Utf-8 convertito, in quanto viene già convertito	S/C/Cr/Cd
_JSONINPA\$(i)	Array monodimensionale per splittare stringhe > 255 caratteri. es	S/C/Cd
_JSONOUTA\$(i)	Array monodimensionale in cui vengono splittate stringhe > 255 caratteri. es	S/C/Cd
JSONADDNUM()	Par: (<RifObj>, <Key\$>, <ValNum>) Funz. che aggiunge un valore JSON_NUMBER all'elemento es	S/C/Cr/Cd
JSONADDREAL()	Par: (<RifObj>, <Key\$>, <ValNumReal>) Funz. aggiunge un valore JSON_REAL all'elemento es	S/C/Cr/Cd
JSONADDBOOL()	Par: (<RifObj>, <Key\$>, <Bool>) Funzione che aggiunge un valore JSON_TRUE/FALSE es	S/C/Cr/Cd
JSONADDNULL()	Par: (<RifObj>, <Key\$>) Funzione per settare un valore JSON_NULL all'elemento es	S/C/Cr/Cd
JSONADDARR()	Par: (<RifObj>, <Key\$>, <RifArrayDalns>) Funzione che aggiunge JSON_ARRAY all'elemento es	S/C/Cr/Cd
JSONARRSIZE()	Par: (<RifArray>) Funzione che ritorna il numero degli elementi inseriti nell'array es	S/C/Cr/Cd
JSONADDFILE()	Par: (<RifObj>, <Key\$>, <NomeFileIn\$>) Funz. aggiunge il contenuto di file come JSON_STRING es	S/C/Cr/Cd
JSONDUMP	Par: <RifObj>, <CharIdent>, <NomeFileOut\$> Istruzione che scrive il dato json es	S/C/Cr/Cd
JSONGETFILE	Par: <RifObj>, <Key\$>, <IxElem>, <NomeFileOut\$> Istruzione che legge il dato json (e Base64) es	S/C/Cr/Cd
JSONLOAD()	Par: (<NomeFile\$>) Funzione che carica un dato json da file es	S/C/Cr/Cd
JSONGETOBJ()	Par: (<RifObj>, <Key\$>, <IxElem>) Funz. che legge il riferim. a un elemento del dato JSON figlio es	S/C/Cr/Cd
JSONDELOBJ	Par: <RifObj>, <Key\$> Istruzione che elimina l'elemento dell'oggetto con chiave es	S/C/Cr/Cd
JSONGETSTR()	Par: (<RifStrObj>) Funz. che legge il valore dell'elemento del dato JSON es	S/C/Cr/Cd
JSONGETSTRF()	Par: (<RifStrObj>, <nomefile>) Funz. legge il valore dell'elemento del dato JSON e scrive in file es	S/C/Cr/Cd/Sm/Cm
JSONGETNUM()	Par: (<RifNumObj>) Funzione che legge il valore dell'elemento del dato JSON es	S/C/Cr/Cd
JSONGETREAL()	Par: (<RifRealObj>) Funzione che legge il valore dell'elemento del dato JSON es	S/C/Cr/Cd
JSONGETBOOL()	Par: (<RifBoolObj>) Funzione che legge il valore dell'elemento del dato JSON es	S/C/Cr/Cd
JSONGETARR()	Par: (<RifArray>, <IxElem>) Funzione che legge il riferimento all'elemento dell'array. es	S/C/Cr/Cd
JSONDELARR	Par: <RifObj>, <IdElem> Istruzione che elimina uno o tutti gli elementi di un array es	S/C/Cr/Cd
JSONSTARTITR	Par: <RifObj>, <Key\$> Avvia <u>Iteratore</u> per elemento tipo JSON_OBJECT es	S/C/Cr/Cd
JSONITERKEY	Par: <HndIter> Ritorna il valore della chiave della posizione attuale dell' <u>Iteratore</u> es	S/C/Cr/Cd
JSONITERVAL	Par: <HndIter> Ritorna il riferimento a JSON_OBJECT della posizione attuale dell' <u>Iteratore</u> es	S/C/Cr/Cd
JSONTYPE()	Par: (<RifObj>) Funzione che ritorna il tipo di oggetto es	S/C/Cr/Cd
JSONITERNXT	Par: <HndIter> Sposta <u>Iteratore</u> sull'elemento successivo es	S/C/Cr/Cd
JSONSTOPITR	Par: <HndIter> Termina <u>Iteratore</u> con Handle es	S/C/Cr/Cd
JSONITERSET	Par: <HndIter>, <RifObj> Cambia il valore dell'elemento puntato attualmente dall' <u>Iteratore</u> es	S/C/Cr/Cd
JSONDESTROY	Par: <RifObj> Istruzione che rilascia tutte le risorse occupate es	S/C/Cr/Cd
_ERRJSON\$	Variabile errori ritornati dalle operazioni relative a JSON. es	S/C/Cd

Comunicazioni WEB

Per comunicazioni WEB SERVICE consultare la tabella WEB SERVICE e la documentazione Man Web Shaker (per le istruzioni _JSONDATIINP e _JSONDATIOUT)

esempi e casi

'SCRITTURA

'-----

'PUT JSON: creazione di un nuovo oggetto JSON e memorizzazione del suo riferimento nella variabile RADICE_JSON.

RADICE_JSON=JSONCREATE()

IF RADICE_JSON=0 THEN VIMSG 5, "Impossibile creare un nuovo oggetto JSON"+"|"+_ERRJSON\$: GOTO JSON_S_END

'Output:

' {}

'Aggiunta di un dato di tipo alfanumerico alla radice dell'elemento JSON. Il dato viene aggiunto con chiave "chiave1" e valore "valore1".

'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.

RIF=JSONADDSTR(RADICE_JSON,"chiave1","valore1")

IF RIF=0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo alfanumerico"+"|"+_ERRJSON\$: GOTO JSON_S_EXIT

'Output:

' {

' "chiave1" : "valore1"

' }

'Aggiunta di un dato di tipo numerico (senza parte decimale) alla radice dell'elemento JSON. Il dato viene aggiunto con chiave "chiave2" e

valore 123.

'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.

RIF=JSONADDNUM(RADICE_JSON,"chiave2",123)

IF RIF = 0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo numerico"+"|"+_ERRJSON\$: GOTO JSON_S_EXIT

'Output:

' {

' "chiave1" : "valore1"

' "chiave2" : 123

' }

'Aggiunta di valore JSON_BOOL (TRUE/FALSE) alla radice dell'elemento JSON. Il terzo parametro indica quale valore ammesso inserire: FALSE

(0), TRUE (<>0)

'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.

RIF=JSONADDBOOL(RADICE_JSON,"chiave3",1)

IF RIF=0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo JSON_BOOL"+"|"+_ERRJSON\$: GOTO JSON_S_EXIT

'Output:

' {

' "chiave1" : "valore1"

' "chiave2" : 123

' "chiave3" : true

' }

'Aggiunta di un array JSON alla radice dell'elemento JSON. L'array viene aggiunto con chiave "array1" e

```
'il suo riferimento e' restituito nella variabile di ritorno. Se l'inserimento fallisce il valore di ritorno e' = zero.
RIF_ARRAY=JSONADDARR(RADICE_JSON, "array1",0)
IF RIF_ARRAY=0 THEN VIMSG 5, "Impossibile aggiungere un array JSON"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
'Output:
' {
'   "chiave1" : "valore1"
'   "chiave2" : 123
'   "chiave3" : true
'   "array1"  : []
' }

'Aggiunta di un elemento di tipo alfanumerico all'array. Non occorre specificare la chiave, in quanto non e' utilizzata per gli elementi di
un array JSON.
'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
RIF = JSONADDSTR(RIF_ARRAY,"", "elemento1")
IF RIF=0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo alfanumerico all'array JSON"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
'Output:
' {
'   "chiave1" : "valore1"
'   "chiave2" : 123
'   "chiave3" : true
'   "array1"  : [
'     "elemento1"
'   ]
' }

'Letture del numero di elementi presenti nell'array. Se si verificano errori nella lettura il valore di ritorno e' = -1
NUM_ELEMENTI=JSONARRSIZE(RIF_ARRAY)
IF NUM_ELEMENTI=-1 THEN VIMSG 5, "Impossibile determinare il numero di elementi dell'array JSON"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
VIMSG 4, "Numero di elementi nell'array JSON: "+STR$(NUM_ELEMENTI,0)

'Creazione di un nuovo oggetto JSON.
'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
OGGETTO_2=JSONCREATE()
IF OGGETTO_2=0 THEN VIMSG 5, "Impossibile creare un nuovo oggetto JSON"+"|" + _ERRJSON$: GOTO JSON_S_EXIT

'Aggiunta di un elemento di tipo numerico REAL (con parte decimale).
'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
RIF=JSONADDRREAL(OGGETTO_2,"num",1.23)
IF RIF=0 THEN VIMSG 5, "Impossibile aggiungere un elemento di tipo numerico"+"|" + _ERRJSON$: GOTO JSON_S_EXIT

'Aggiunta dell'oggetto appena creato all'oggetto creato in precedenza.
'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
RIF=JSONADDOBJ(RADICE_JSON, "oggetto2", OGGETTO_2)
IF RIF=0 THEN VIMSG 5, "Impossibile aggiungere l'oggetto JSON al precedente"+"|" + _ERRJSON$: GOTO JSON_S_EXIT

'Output:
' {
'   "chiave1" : "valore1"
'   "chiave2" : 123
'   "chiave3" : true
'   "array1"  : [
'     "elemento1"
'   ],
'   "oggetto2" : {
'     "num" : 1.23
'   }
' }

'Dump su disco della struttura JSON nel percorso specificato (es. cartella datiutente).
'Il secondo parametro (3), indica il numero di spazi utilizzati per l'eventuale indentazione dei dati.
JSONDUMP RADICE_JSON, 3, "\json2\esempiol.json"

LETTURA
'-----

FILE$="\json2\esempiol.json"

'Controllo esistenza file
IF NOT ESISFILE(FILE$) THEN VIMSG 5, "File JSON mancante, crearlo con put json e riprovare " : GOTO JSON_L_END

'Letture un oggetto JSON da file
RADICE_JSON=JSONLOAD(FILE$)
IF RADICE_JSON=0 THEN VIMSG 5, _ERRJSON$+"|" + "Eseguire lo sprix put json e riprovare" : GOTO JSON_L_END

'Letture di una chiave dell'oggetto JSON.
CHIAVE$="chiave1"
RIF=JSONGETOBJ(RADICE_JSON,CHIAVE$,0)
IF RIF=0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|" + _ERRJSON$: GOTO JSON_L_EXIT

'Letture di un dato di tipo alfanumerico. Se lettura fallisce, _ERRJSON$ contiene l'errore.
DATO$=JSONGETSTR(RIF)
IF _ERRJSON$<>" " THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo alfanumerico"+"|" + _ERRJSON$: GOTO JSON_L_EXIT
VIMSG 4, "Chiave: "+CHIAVE$+"|" + "Dato: "+DATO$

'Letture di una chiave dell'oggetto JSON.
CHIAVE$ = "chiave2"
RIF=JSONGETOBJ(RADICE_JSON,CHIAVE$,0)
IF RIF=0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|" + _ERRJSON$: GOTO JSON_L_EXIT

'Letture di un dato di tipo numerico. Se lettura fallisce _ERRJSON$ contiene l'errore.
DATO=JSONGETNUM(RIF)
IF _ERRJSON$<>" " THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo numerico"+"|" + _ERRJSON$: GOTO JSON_L_EXIT
VIMSG 4, "Chiave: "+CHIAVE$+"|" + "Dato: "+STR$(DATO,0)

'Letture di una chiave dell'oggetto JSON.
CHIAVE$="chiave3"
RIF=JSONGETOBJ(RADICE_JSON,CHIAVE$,0)
IF RIF=0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|" + _ERRJSON$: GOTO JSON_L_EXIT

'Letture di un dato di tipo JSON_BOOL. Se lettura fallisce _ERRJSON$ contiene l'errore.
DATO=JSONGETBOOL(RIF)
IF DATO=-1 THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo JSON_BOOL"+"|" + _ERRJSON$: GOTO JSON_L_EXIT
MSG$="Chiave: "+CHIAVE$+"|" + "Dato: "
IF DATO=0 THEN MSG$=MSG$+"false" ELSE MSG$=MSG$+"true"
VIMSG 4, MSG$
```

```

'Lettura del riferimento all'array creato in precedenza.
RIF_ARRAY=JSONGETOBJ(RADICE_JSON,"array1",0)
IF RIF_ARRAY=0 THEN VIMSG 5, "Impossibile ottenere il riferimento all'array JSON"+"|"+ _ERRJSON$ : GOTO JSON_L_EXIT

'Lettura del numero di elementi dell'array.
NUM_ELEMENTI=JSONARRSIZE(RIF_ARRAY)
IF NUM_ELEMENTI=-1 THEN VIMSG 5, "Impossibile ottenere il numero di elementi dell'array JSON"+"|"+ _ERRJSON$ : GOTO JSON_L_EXIT

'Iterazione dei vari elementi, a partire dal primo elemento (0), fino all'ultimo (NUM_ELEMENTI-1)
ERR$=""
FOR INDICE_ELEM=0 TO NUM_ELEMENTI -1
RIF=JSONGETARR(RIF_ARRAY,INDICE_ELEM)
IF RIF=0 THEN ERR$="Impossibile leggere l'elemento in posizione "+STR$(INDICE_ELEM,0)+"|"+ _ERRJSON$ : INDICE_ELEM=NUM_ELEMENTI : GOTO
ARRAY_NEXT

DATO$=JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN ERR$="Impossibile leggere l'elemento come stringa in posizione "+STR$(INDICE_ELEM,0)+"|"+ _ERRJSON$ :
INDICE_ELEM=NUM_ELEMENTI : GOTO ARRAY_NEXT

VIMSG 4, "Posizione: "+STR$(INDICE_ELEM,0)+"|"+ "Elemento: "+DATO$

ARRAY_NEXT:
NEXT

IF ERR$<>" THEN VIMSG 5, ERR$ : GOTO JSON_L_EXIT

'ITERAZIONI CHIAVI
'-----
FILE$="]\json2\esempiol.json"

'Controllo esistenza file
IF NOT ESISFILE(FILE$) THEN VIMSG 5, "File JSON mancante, crearlo con put json e riprovare" : GOTO JSON_I_END

'Lettura un oggetto JSON da file.
RADICE_JSON=JSONLOAD(FILE$)
IF RADICE_JSON=0 THEN VIMSG 5, _ERRJSON$ +"|"+ "Eseguire lo sprix put json e riprovare" : GOTO JSON_I_END

'Inizializzazione dell'iterazione sulla radice dell'oggetto JSON
ITERAT=JSONSTARTITR(RADICE_JSON,"")
IF ITERAT=0 THEN VIMSG 5, "Impossibile inizializzare l'iterazione sull'oggetto specificato" +"|"+ _ERRJSON$ : GOTO JSON_I_EXIT

ITER_LOOP:
'Lettura della chiave alla posizione corrente dell'iteratore
CHIAVE$=JSONITERKEY(ITERAT)
IF CHIAVE$="" THEN VIMSG 5, "Errore nella lettura di una chiave" +"|"+ _ERRJSON$ : GOTO ITER_EXIT

'Lettura del riferimento al dato (al quale la chiave fa riferimento)
RIF=JSONINTERVAL(ITERAT)
IF RIF=0 THEN VIMSG 5, "Errore nella lettura del riferimento al dato della chiave "+CHIAVE$+"|"+ _ERRJSON$ : GOTO ITER_EXIT

'Lettura del tipo del dato appartenente alla chiave
TIPO=JSONTYPE(RIF)
MSG$="Chiave: "+CHIAVE$+"|"+ "Tipo: "+STR$(TIPO,0,0) + " ("

IF TIPO=0 THEN MSG$=MSG$+"JSON_OBJECT" : GOTO PRINT_MSG
IF TIPO=1 THEN MSG$=MSG$+"JSON_ARRAY" : GOTO PRINT_MSG
IF TIPO=2 THEN MSG$=MSG$+"JSON_STRING" : GOTO PRINT_MSG
IF TIPO=3 THEN MSG$=MSG$+"JSON_INTEGER" : GOTO PRINT_MSG
IF TIPO=4 THEN MSG$=MSG$+"JSON_REAL" : GOTO PRINT_MSG
IF TIPO=5 THEN MSG$=MSG$+"JSON_TRUE" : GOTO PRINT_MSG
IF TIPO=6 THEN MSG$=MSG$+"JSON_FALSE" : GOTO PRINT_MSG
IF TIPO=7 THEN MSG$=MSG$+"JSON_NULL" : GOTO PRINT_MSG

PRINT MSG:
VIMSG 4, MSG$+")"

ITER_NEXT:
'Avanzamento dell'iteratore sulla prossima chiave da leggere
ESITO_NXT=JSONITERNXT(ITERAT)

IF ESITO_NXT=0 THEN GOTO ITER_LOOP 'L'avanzamento e' riuscito correttamente
IF ESITO_NXT<0 THEN VIMSG 5, "Impossibile eseguire l'avanzamento"+"|"+ _ERRJSON$ : GOTO ITER_EXIT
IF ESITO_NXT>0 THEN GOTO ITER_EXIT 'Non esistono altre chiavi su cui avanzare (tutte le chiavi sono gia' state iterate)

'-----
ITER_EXIT:
'Rilascio delle risorse utilizzate per l'iterazione
ESITO=JSONSTOPITR(ITERAT) : ITERAT=0
IF ESITO<>0 THEN VIMSG 5, "Impossibile rilasciare le risorse di iterazione"+"|"+ _ERRJSON$

JSON_I_EXIT:
'Rilascio delle risorse impiegate in precedenza
JSONDESTROY RADICE_JSON : RADICE_JSON=0

JSON_I_END:
END

'-----
JSON_L_EXIT:
'Rilascio delle risorse impiegate in precedenza
JSONDESTROY RADICE_JSON : RADICE_JSON=0

JSON_L_END:
END

'-----
JSON_S_EXIT:
'Rilascio delle risorse impiegate in precedenza.
JSONDESTROY RADICE_JSON : RADICE_JSON=0
IF OGGETTO_2>0 THEN JSONDESTROY OGGETTO_2 : OGGETTO_2=0

JSON_S_END:
END

```

```

#####
' SCELTA OPERAZIONE #
#####

SCELTA:
  NF=AUTOFIN() : APRIFIN NF,1,0,5,10,10,60 : LOCATE 3,3 : PRINT "Scrivi (S) - Leggi (L) - Itera (I) "; : INPUT "A",P$,1 : CHIUDIFIN NF
  IF P$="S" THEN GOTO PUT_JSON 'Scrivi Json
  IF P$="L" THEN GOTO GET_JSON 'Leggi Json
  IF P$="I" THEN GOTO ITER_JSON 'Itera Json
END

#####
' SCRITTURA JSON #
#####

PUT_JSON:
  ROOTOBJ = JSONCREATE() 'creazione di un nuovo OBJECT JSON e memorizzazione del suo riferimento nella variabile ROOTOBJ
  IF ROOTOBJ = 0 THEN VIMSG 5, "Impossibile creare un nuovo OGGETTO JSON"+"|" + _ERRJSON$: GOTO JSON_S_END

  'NOTARE BENE
  -----
  'La JSONCREATE() crea dei NODI di tipo OBJECT.
  'Le specifiche JSON (RFC4627) consentono anche documenti con NODO RADICE di tipo ARRAY: appendendo un NODO di tipo ARRAY con una chiave
  'ARBITRARIA E NON VUOTA, si puo' eseguire la JSONDUMP passandole il RIFERIMENTO DEL NODO ARRAY per ottenere su disco un file ".json"
  'avente come ROOT una array. Il primo parametro della JSONDUMP, infatti, si basa sul RIFERIMENTO all'OGGETTO JSON, che puo' anche essere
  'di tipo JSON_ARRAY. Quindi e' sufficiente recuperare tale riferimento e indicarlo nella JSONDUMP (che opera in autonomia);
  'la prima chiave creata all'inizio programma (ovvero "ADDARRAY_1") funge solo da "appoggio con nome arbitrario".
  'Successivamente, l'eventuale lettura su disco tramite la JSONLOAD del documento trattando il nodo radice come array,
  'non risultano problemi.
  -----

  'Aggiunta di un array JSON alla radice dell'elemento JSON. L'array viene aggiunto con chiave "ADDARRAY_1" e
  'il suo riferimento e' restituito nella variabile di ritorno. Se l'inserimento fallisce il valore di ritorno e' = zero.
  RIF_ARRAY_1 = JSONADDARR(ROOTOBJ,"ADDARRAY_1",0) 'Riferimento Array 1
  IF RIF_ARRAY_1 = 0 THEN VIMSG 5, "Impossibile aggiungere un ARRAY JSON"+"|" + _ERRJSON$: GOTO JSON_S_EXIT

'-----OBJ1
  RIFN1 = JSONCREATE() 'creazione di OBJ1 e memorizzazione del suo riferimento nella variabile RIFN1 (RIFERIMENTO NODO 1)
  IF RIFN1 = 0 THEN VIMSG 5, "Impossibile creare un nuovo OGGETTO JSON"+"|" + _ERRJSON$: GOTO JSON_S_END
  RIOBJ1 = JSONADDOBJ(RIF_ARRAY_1,"ADDOBJ1",RIFN1) 'AddOBJ1 (Riferimento Oggetto 1)

  'Aggiunta di un primo dato ALFANUMERICO al riferimento OBJ1: il dato e' aggiunto con chiave "obj1-ADDSTR1" e valore "1".
  'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
  RF = JSONADDSTR(RIOBJ1,"obj1-ADDSTR1","A") 'Riferimento
  IF RF = 0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo ALFANUMERICO"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
  'Aggiunta di un secondo dato ALFANUMERICO al riferimento OBJ1: il dato e' aggiunto con chiave "obj1-ADDSTR2" e valore "2".
  'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
  RF = JSONADDSTR(RIOBJ1,"obj1-ADDSTR2","B")
  IF RF = 0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo ALFANUMERICO"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
  'Aggiunta di un dato di NUMERICO SENZA DECIMALE al riferimento OBJ1. Il dato e' aggiunto con Chiave "obj1-ADDDNUM1" e valore 33
  'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
  RF = JSONADDDNUM(RIOBJ1,"obj1-ADDDNUM",3)
  IF RF = 0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo NUMERICO"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
  'Aggiunta di valore BOOL (true/false) al riferimento OBJ1. Il terzo parametro indica il valore: FALSE (0), TRUE (<0)
  'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
  RF = JSONADDBOOL(RIOBJ1,"obj1-ADDBOOL",1) 'true
  IF RF = 0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo BOOL"+"|" + _ERRJSON$: GOTO JSON_S_EXIT

  'Aggiunta di un ARRAY JSON al riferimento OBJ1. L'array viene aggiunto con chiave "obj1-ADDARRAY_2"
  'Il suo riferimento e' restituito nella variabile di ritorno. Se l'inserimento fallisce il valore di ritorno e' = zero.
  RIF_ARRAY_2 = JSONADDARR(RIOBJ1,"obj1-ADDARRAY_2",0) 'Riferimento Array 2
  IF RIF_ARRAY_2 = 0 THEN VIMSG 5, "Impossibile aggiungere un ARRAY"+"|" + _ERRJSON$: GOTO JSON_S_EXIT

'-----OBJ2
  'Creazione di un nuovo OBJ2 e memorizzazione del suo riferimento nella variabile RIFN2
  'Il valore di ritorno indica il rif. all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
  RIFN2 = JSONCREATE() ' (RIFERIMENTO NODO 2)
  IF RIFN2 = 0 THEN VIMSG 5, "Impossibile creare un nuovo OGGETTO"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
  RIOBJ2 = JSONADDOBJ(RIF_ARRAY_2,"ADDOBJ2",RIFN2) 'AddOBJ2

  'Aggiunta di dati ALFANUMERICI al riferimento OBJ2 e con chiave "obj2-ADDSTR1/2/3/4/5" e valori "1/2/3/.../5"
  'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = 0
  RF = JSONADDSTR(RIOBJ2,"obj2-ADDSTR1","C")
  IF RF=0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo ALFANUMERICO"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
  RF = JSONADDSTR(RIOBJ2,"obj2-ADDSTR2","D")
  IF RF=0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo ALFANUMERICO"+"|" + _ERRJSON$: GOTO JSON_S_EXIT
  RF = JSONADDSTR(RIOBJ2,"obj2-ADDSTR3","E")
  IF RF=0 THEN VIMSG 5, "Impossibile aggiungere un dato di tipo ALFANUMERICO"+"|" + _ERRJSON$: GOTO JSON_S_EXIT

  'Split stringhe > 255 car. (vedi array monodimensionali _JSONINPA$(I) e _JSONOUTA$(I))
  FOR I=1 TO 3
    _JSONINPA$(I) = "(" + STR$(I,0,0)+" " 0123456789"
    _JSONINPA$(I) = _JSONINPA$(I) + "ABCDEFGHILMNOPQRSTUVZ"
  NEXT

  RF = JSONADDSTR(RIOBJ2,"obj2-ADDSTR4","F") : _JSONINPA$(1) = ""
  RF = JSONADDSTR(RIOBJ2,"obj2-ADDSTR5","G")

'-----OBJ3
  RIFN3 = JSONCREATE()
  RIOBJ3 = JSONADDOBJ(RIF_ARRAY_2,"ADDOBJ3",RIFN3) 'AddOBJ3
  RF = JSONADDSTR(RIOBJ3,"obj3-ADDSTR1","H")
  RF = JSONADDSTR(RIOBJ3,"obj3-ADDSTR2","I")

'-----OBJ1
  'Aggiunta di un elemento di tipo numerico REAL (con decimale) al riferimento OBJ1
  'Il valore di ritorno indica il riferimento all'elemento inserito. Se l'inserimento fallisce il valore di ritorno e' = zero.
  ERR=22.44
  RF = JSONADDREAL(RIOBJ1,"obj1-ADDREAL",ERR) ' ADD VALORE JSON_REAL
  RIF_ARRAY_3 = JSONADDARR(RIOBJ1,"obj1-ADDARRAY_3",0) 'Riferimento Array (3)

'-----OBJ4
  RIFN4 = JSONCREATE()

```

```

RIOBJ4 = JSONADDOBJ(RIF_ARRAY_3,"ADDOBJ4",RIFN4) 'AddOBJ4
RF      = JSONADDSTR(RIOBJ4,"obj4-ADDSTR1","L")
RF      = JSONADDSTR(RIOBJ4,"obj4-ADDSTR2","M")

'-----OBJ5
RIFN5 = JSONCREATE()
RIOBJ5 = JSONADDOBJ(RIF_ARRAY_3,"ADDOBJ5",RIFN5) 'AddOBJ5
RF      = JSONADDSTR(RIOBJ5,"obj5-ADDSTR1","N")
RF      = JSONADDSTR(RIOBJ5,"obj5-ADDSTR2","O")

'-----OBJ6
RIFN6 = JSONCREATE()
RIOBJ6 = JSONADDOBJ(RIF_ARRAY_3,"ADDOBJ6",RIFN6) 'AddOBJ6
RF      = JSONADDSTR(RIOBJ6,"obj6-ADDSTR1","P")
RF      = JSONADDSTR(RIOBJ6,"obj6-ADDSTR2","Q")

'-----OBJ1
RF      = JSONADDSTR(RIOBJ1,"obj1-ADDSTR4","R")
RF      = JSONADDSTR(RIOBJ1,"obj1-ADDSTR5","S")
RF      = JSONADDSTR(RIOBJ1,"obj1-ADDSTR6","T")

'-----
'LETTURA DEL NUMERO DI ELEMENTI PRESENTI NEGLI ARRAY.
'-----
'Se si verificano errori nella lettura il valore di ritorno e' = -1

NUM_ELEM_1=JSONARRSIZE(RIF_ARRAY_1)
NUM_ELEM_2=JSONARRSIZE(RIF_ARRAY_2)
NUM_ELEM_3=JSONARRSIZE(RIF_ARRAY_3)

IF NUM_ELEM_1=-1 THEN VIMSG 5, "Impossibile determinare il numero elementi ARRAY_1"+"|"+ _ERRJJSON$: GOTO JSON_S_EXIT
IF NUM_ELEM_2=-1 THEN VIMSG 5, "Impossibile determinare il numero elementi ARRAY_2"+"|"+ _ERRJJSON$: GOTO JSON_S_EXIT
IF NUM_ELEM_3=-1 THEN VIMSG 5, "Impossibile determinare il numero elementi ARRAY_3"+"|"+ _ERRJJSON$: GOTO JSON_S_EXIT

VIMSG 4, "Numero elementi nell'ARRAY_1: "+STR$(NUM_ELEM_1,0,0)+" elementi" '1'array_1 ha 1 elemento
VIMSG 4, "Numero elementi nell'ARRAY_2: "+STR$(NUM_ELEM_2,0,0)+" elementi" '1'array_2 ha 2 elementi
VIMSG 4, "Numero elementi nell'ARRAY_3: "+STR$(NUM_ELEM_3,0,0)+" elementi" '1'array_3 ha 3 elementi

'Dump su disco della struttura JSON nel percorso specificato (es. cartella datiutente).
'Il secondo parametro (3), indica il numero di spazi utilizzati per l'eventuale indentazione dei dati

JSONDUMP ROOTOBJ,      3, "\json2\root_dump.json"      'inserimento dell'OGGETTO nella ROOT
JSONDUMP RIF_ARRAY_1 , 3, "\json2\root_ARRAY_dump.json" 'inserimento dell'ARRAY nella ROOT
JSONDUMP RIOBJ2,      3, "\json2\OBJ2_dump.json"      'inserimento dell'OBJ2 nella ROOT

'NOTARE: il primo parametro della JSONDUMP si basa sul riferimento all'oggetto JSON, che puo' anche essere di tipo JSON_ARRAY.
'Quindi basta recuperare tale riferimento e indicarlo in JSONDUMP (che fa tutto da sola); la prima chiave creata in precedenza
'ovvero "ADDARRAY_1" fungeva solo da appoggio con nome arbitrario.

#####
' LETTURA JSON #
#####

GET_JSON:

'Dichiarazione file

FILE$="\json2\root_ARRAY_dump.json" 'Lettura a partire da ARRAY
'oppure FILE$="\json2\OBJ2_dump.json"
'oppure FILE$="\json2\root_dump.json"

'Controllo esistenza file
IF NOT ESISFILE(FILE$) THEN VIMSG 5, "File JSON mancante, occorre crearlo" : GOTO JSON_L_END

'Letture RADICE json da file
'-----
ROOTOBJ = JSONLOAD(FILE$) : TYPE=JSONTYPE(ROOTOBJ)
IF ROOTOBJ = 0 THEN VIMSG 5, _ERRJJSON$+"|"+"Esequire lo SPRIX PUT_JSON e riprovare" : GOTO JSON_L_END

'Letture del RIFERIMENTO all'ARRAY_1 creato in precedenza (la chiave d'appoggio ADDARRAY_1)
RIF_ARRAY_1 = JSONGETOBJ(ROOTOBJ,"ADDARRAY_1",0)
IF RIF_ARRAY_1 = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento all'array JSON"+"|"+ _ERRJJSON$: GOTO JSON_L_EXIT
'Letture del NUMERO di ELEMENTI dell'ARRAY.
NUM_ELEMENTI=JSONARRSIZE(ROOTOBJ)
IF NUM_ELEMENTI = -1 THEN VIMSG 5, "Impossibile ottenere il numero di elementi dell'array JSON"+"|"+ _ERRJJSON$: GOTO JSON_L_EXIT

'-----
'Letture di una CHIAVE dell'OGGETTO JSON
CHIAVE$ = "obj1-ADDSTR1"
RIF = JSONGETOBJ(RIF_ARRAY_1,CHIAVE$,0) : TYPE=JSONTYPE(RIF)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|"+ _ERRJJSON$: GOTO JSON_L_EXIT
'Letture di un DATO di tipo ALFANUMERICICO. Se lettura fallisce, _ERRJJSON$ contiene l'errore.
DATO$ = JSONGETSTR(RIF)
IF _ERRJJSON$ <> "" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo ALFANUMERICICO"+"|"+ _ERRJJSON$: GOTO JSON_L_EXIT
VIMSG 4, "Chiave1: " + CHIAVE$ + "|" + "Dato: " + DATO$

'-----
'Letture di una CHIAVE dell'OGGETTO JSON
CHIAVE$ = "obj1-ADDSTR2"
RIF = JSONGETOBJ(RIF_ARRAY_1,CHIAVE$,0) : TYPE=JSONTYPE(RIF)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|"+ _ERRJJSON$: GOTO JSON_L_EXIT
'Letture di un DATO di tipo ALFANUMERICICO. Se lettura fallisce, _ERRJJSON$ contiene l'errore.
DATO$ = JSONGETSTR(RIF)
IF _ERRJJSON$ <> "" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo ALFANUMERICICO"+"|"+ _ERRJJSON$: GOTO JSON_L_EXIT
VIMSG 4, "Chiave2: " + CHIAVE$ + "|" + "Dato: " + DATO$

'-----
'Letture di una CHIAVE dell'OGGETTO JSON
CHIAVE$ = "obj1-ADDNUM"
RIF=JSONGETOBJ(RIF_ARRAY_1,CHIAVE$,0) : TYPE=JSONTYPE(RIF)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+ _ERRJJSON$: GOTO JSON_L_EXIT
'Letture di un DATO di tipo NUMERICICO. Se lettura fallisce _ERRJJSON$ contiene l'errore.
DATO = JSONGETNUM(RIF)
IF _ERRJJSON$ <> "" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo NUMERICICO"+"|"+ _ERRJJSON$: GOTO JSON_L_EXIT
VIMSG 4, "Chiave3: "+CHIAVE$+"|"+"Dato: "+STR$(DATO,0,0)

'-----
'Letture di una CHIAVE dell'OGGETTO JSON

```



```

CHIAVE$ = "obj1-ADDBOOL"
RIF = JSONGETOBJ(RIF_ARRAY_1,CHIAVE$,0) : TYPE=JSONTYPE(RIF)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|" + _ERRJON$ : GOTO JSON_L_EXIT
'Letture di un dato di tipo JSON_BOOL. Se lettura fallisce _ERRJON$ contiene l'errore.
DATO = JSONGETBOOL(RIF)
IF DATO = -1 THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo JSON_BOOL"+"|" + _ERRJON$ : GOTO JSON_L_EXIT
MSG$ = "Chiave4: "+CHIAVE$+"|" + "Dato: "
'IF DATO = 0 THEN MSG$ = MSG$ + "false" ELSE MSG$ = MSG$ + "true"
IF TYPE = 6 THEN MSG$ = MSG$ + "false" ELSE IF TYPE = 5 THEN MSG$ = MSG$ + "true"
VIMSG 4, MSG$

'-----
'Letture del RIFERIMENTO all'ARRAY creato in precedenza
CHIAVE$ = "obj1-ADDARRAY_2"
RIF_ARRAY_2 = JSONGETOBJ(RIF_ARRAY_1,CHIAVE$,0) : TYPE=JSONTYPE(RIF_ARRAY_2)
IF RIF_ARRAY_2 = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento all'array JSON"+"|" + _ERRJON$ : GOTO JSON_L_EXIT
'Letture del NUMERO di ELEMENTI dell'ARRAY.
NUM_ELEMENTI=JSONARRSIZE(RIF_ARRAY_2)
IF NUM_ELEMENTI = -1 THEN VIMSG 5, "Impossibile ottenere il numero di elementi dell'array JSON"+"|" + _ERRJON$ : GOTO JSON_L_EXIT

'-----
'ITERAZIONE degli ELEMENTI a partire dal PRIMO elemento (0) fino all'ULTIMO (NUM_ELEMENTI-1)

ERR$=""
FOR INDICE_ELEM=0 TO NUM_ELEMENTI-1 'NB: lo ZERO e' il primo elemento dell'ARRAY, mentre l'ultimo elemento e' JSONARRSIZE-1
RIF = JSONGETARR(RIF_ARRAY_2,INDICE_ELEM)
IF RIF=0 THEN ERR$="Impossibile leggere elemento in posizione "+STR$(INDICE_ELEM,0)+"|" + _ERRJON$ : INDICE_ELEM=NUM_ELEMENTI : GOTO ARRAY_NEXT1

FOR I=1 TO 5
'Letture di una CHIAVE dell'ELEMENTO dell'ARRAY
IF INDICE_ELEM=0 THEN CHIAVE$ = "obj2-ADDSTR"+STR$(I,0) ELSE CHIAVE$ = "obj3-ADDSTR"+STR$(I,0,0)
IF INDICE_ELEM=NUM_ELEMENTI-1 AND I>2 THEN GOTO NEXT1_OBJ
RIFOBJAR = JSONGETOBJ(RIF,CHIAVE$,0) : TYPE=JSONTYPE(RIFOBJAR)
IF RIFOBJAR = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|" + _ERRJON$ : GOTO JSON_L_EXIT
'Letture di un DATO di tipo ALFANUMERICO. Se lettura fallisce, _ERRJON$ contiene l'errore.
DATO$ = JSONGETSTR(RIFOBJAR)
IF _ERRJON$<>" " THEN VIMSG 5, "Impossibile leggere il dato di tipo ALFANUMERICO"+"|" + _ERRJON$ : GOTO JSON_L_EXIT
VIMSG 4, "ChiaveX: " + CHIAVE$ + "|" + "Dato: " + DATO$
ARRAY_NEXT1:
NEXT
NEXT

NEXT1_OBJ:

'-----
'Letture di una CHIAVE dell'OGGETTO JSON
CHIAVE$ = "obj1-ADDREAL"
RIF = JSONGETOBJ(RIF_ARRAY_1,CHIAVE$,0) : TYPE=JSONTYPE(RIF)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|" + _ERRJON$ : GOTO JSON_L_EXIT
'Letture di un DATO di tipo REAL. Se lettura fallisce, _ERRJON$ contiene l'errore.
DATO = JSONGETREAL(RIF)
IF _ERRJON$ <> " " THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo REAL"+"|" + _ERRJON$ : GOTO JSON_L_EXIT
VIMSG 4, "Chiave1: " + CHIAVE$ + "|" + "Dato: " + STR$(DATO,2,0)

'-----
'Letture del RIFERIMENTO all'ARRAY creato in precedenza
CHIAVE$ = "obj1-ADDARRAY_3"
RIF_ARRAY_3 = JSONGETOBJ(RIF_ARRAY_1,CHIAVE$,0) : TYPE=JSONTYPE(RIF_ARRAY_3)
IF RIF_ARRAY_3 = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento all'array JSON"+"|" + _ERRJON$ : GOTO JSON_L_EXIT
'Letture del NUMERO di ELEMENTI dell'ARRAY.
NUM_ELEMENTI=JSONARRSIZE(RIF_ARRAY_3)
IF NUM_ELEMENTI = -1 THEN VIMSG 5, "Impossibile ottenere il numero di elementi dell'array JSON"+"|" + _ERRJON$ : GOTO JSON_L_EXIT

'-----
'Iterazione degli elementi a partire dal primo elemento (0) fino all'ultimo (NUM_ELEMENTI-1)
ERR$=""
FOR INDICE_ELEM=0 TO NUM_ELEMENTI-1 'NOTARE: ZERO e' il primo elemento dell'ARRAY, mentre l'ultimo e' JSONARRSIZE-1
RIF = JSONGETARR(RIF_ARRAY_3,INDICE_ELEM)
IF RIF=0 THEN ERR$="Impossibile leggere elemento in posizione "+STR$(INDICE_ELEM,0)+"|" + _ERRJON$ : INDICE_ELEM=NUM_ELEMENTI : GOTO ARRAY_NEXT2

FOR I=1 TO 2
'Letture di una CHIAVE dell'ELEMENTO dell'ARRAY
IF INDICE_ELEM=0 THEN CHIAVE$ = "obj4-ADDSTR"+STR$(I,0,0)
IF INDICE_ELEM=1 THEN CHIAVE$ = "obj5-ADDSTR"+STR$(I,0,0)
IF INDICE_ELEM=2 THEN CHIAVE$ = "obj6-ADDSTR"+STR$(I,0,0)

IF INDICE_ELEM=NUM_ELEMENTI-1 AND I>3 THEN GOTO NEXT2_OBJ
RIFOBJAR = JSONGETOBJ(RIF,CHIAVE$,0) : TYPE=JSONTYPE(RIFOBJAR)
IF RIFOBJAR = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|" + _ERRJON$ : GOTO JSON_L_EXIT
'Letture di un DATO di tipo ALFANUMERICO. Se lettura fallisce, _ERRJON$ contiene l'errore.
DATO$ = JSONGETSTR(RIFOBJAR)
IF _ERRJON$<>" " THEN VIMSG 5, "Impossibile leggere il dato di tipo ALFANUMERICO"+"|" + _ERRJON$ : GOTO JSON_L_EXIT
VIMSG 4, "ChiaveX: " + CHIAVE$ + "|" + "Dato: " + DATO$
ARRAY_NEXT2:
NEXT
NEXT

NEXT2_OBJ:

'-----
'Letture di una CHIAVE dell'OGGETTO JSON
FOR I=8 TO 10
CHIAVE$ = "obj1-ADDSTR"+STR$(I-4,0,0)
RIF = JSONGETOBJ(RIF_ARRAY_1,CHIAVE$,0) : TYPE=JSONTYPE(RIF)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave: "+CHIAVE$+"|" + _ERRJON$ : GOTO JSON_L_EXIT
'Letture di un DATO di tipo ALFANUMERICO. Se lettura fallisce, _ERRJON$ contiene l'errore.
DATO$ = JSONGETSTR(RIF)
IF _ERRJON$ <> " " THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo ALFANUMERICO"+"|" + _ERRJON$ : GOTO JSON_L_EXIT
VIMSG 4, "Chiave"+STR$(I,0,0)+" : " + CHIAVE$ + "|" + "Dato: " + DATO$
NEXT
END

#####
'ITERAZIONI CHIAVI #
#####

ITER_JSON:

```

```

'Dichiarazione file
FILE$=]\json2\root_ARRAY_dump.json"
'oppure FILE$=]\json2\OBJ2_dump.json"
'oppure FILE$=]\json2\root_dump.json"

'Controllo esistenza file
IF NOT ESISFILE(FILE$) THEN VIMSG 5, "File JSON mancante, crearlo con PUT JSON e riprovare" : GOTO JSON_I_END

'Letture un oggetto JSON da file.
ROOTOBJ = JSONLOAD(FILE$)
IF ROOTOBJ = 0 THEN VIMSG 5, _ERRJSON$+"|"+"Eseguire lo sprix put json e riprovare" : GOTO JSON_I_END

'Letture del RIFERIMENTO all'ARRAY 1 creato in precedenza (la chiave d'appoggio ADDARRAY_1)
RIF_ARRAY_1 = JSONGETOBJ(ROOTOBJ,"ADDARRAY 1",0)
IF RIF_ARRAY_1 = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento all'array JSON"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'Letture del NUMERO di ELEMENTI dell'ARRAY.
NUM_ELEMENTI=JSONARRSIZE(ROOTOBJ)
IF NUM_ELEMENTI = -1 THEN VIMSG 5, "Impossibile ottenere il numero di elementi dell'array JSON"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT

'Inizializzazione dell'iterazione sulla radice dell'oggetto JSON
ITERAT = JSONSTARTITR(ROOTOBJ,"")
ITERAT = JSONSTARTITR(RIF_ARRAY_1,"")
IF ITERAT = 0 THEN VIMSG 5, "Impossibile inizializzare l'iterazione sull'oggetto specificato" +"|"+_ERRJSON$ : GOTO JSON_I_EXIT

ITER_LOOP:
'Letture della chiave alla posizione corrente dell'iteratore
CHIAVE$ = JSONITERKEY(ITERAT)
IF CHIAVE$ = "" THEN VIMSG 5, "Errore nella lettura di una chiave" +"|"+_ERRJSON$ : GOTO ITER_EXIT
'Letture del riferimento al dato (al quale la chiave fa riferimento)
RIF = JSONINTERVAL(ITERAT)
IF RIF = 0 THEN VIMSG 5, "Errore nella lettura del riferimento al dato della chiave "+CHIAVE$+"|"+_ERRJSON$ : GOTO ITER_EXIT
'Letture del tipo del dato appartenente alla chiave
TIPO = JSONTYPE(RIF)
MSG$ = "Chiave: "+CHIAVE$+"|"+"Tipo: "+STR$(TIPO,0,0) + " ("

IF TIPO=0 THEN MSG$=MSG$+"JSON_OBJECT" : GOTO PRINT1_MSG
IF TIPO=1 THEN MSG$=MSG$+"JSON_ARRAY" : GOTO PRINT1_MSG
IF TIPO=2 THEN MSG$=MSG$+"JSON_STRING" : GOTO PRINT1_MSG
IF TIPO=3 THEN MSG$=MSG$+"JSON_INTEGER" : GOTO PRINT1_MSG
IF TIPO=4 THEN MSG$=MSG$+"JSON_REAL" : GOTO PRINT1_MSG
IF TIPO=5 THEN MSG$=MSG$+"JSON_TRUE" : GOTO PRINT1_MSG
IF TIPO=6 THEN MSG$=MSG$+"JSON_FALSE" : GOTO PRINT1_MSG
IF TIPO=7 THEN MSG$=MSG$+"JSON_NULL" : GOTO PRINT1_MSG

PRINT1_MSG:
VIMSG 4, MSG$+"")

ITER_NEXT:
'Avanzamento dell'iteratore sulla prossima chiave da leggere
ESITO_NXT = JSONITERNXT(ITERAT)

IF ESITO_NXT = 0 THEN GOTO ITER_LOOP 'L'avanzamento e' riuscito correttamente
IF ESITO_NXT < 0 THEN VIMSG 5, "Impossibile eseguire l'avanzamento"+"|"+_ERRJSON$ : GOTO ITER_EXIT
IF ESITO_NXT > 0 THEN GOTO ITER_EXIT 'Non esistono altre chiavi su cui avanzare (tutte le chiavi sono gia' state iterate)

ITER_EXIT:
ESITO = JSONSTOPITR(ITERAT) : ITERAT = 0
IF ESITO <> 0 THEN VIMSG 5, "Impossibile rilasciare le risorse di iterazione"+"|"+_ERRJSON$

'-----
'RILASCIO RISORSE UTILIZZATE IN PRECEDENZA
'-----
JSON_I_EXIT: 'Rilascio delle risorse utilizzate in ITERAZIONE
JSON_L_EXIT: 'Rilascio delle risorse utilizzate per la LETTURA
JSON_S_EXIT: 'Rilascio delle risorse utilizzate per la SCRITTURA

JSONDESTROY ROOTOBJ : ROOTOBJ = 0
IF RIFN1 > 0 THEN JSONDESTROY RIFN1 : RIFN1 = 0
IF RIFN2 > 0 THEN JSONDESTROY RIFN2 : RIFN2 = 0
IF RIFN3 > 0 THEN JSONDESTROY RIFN3 : RIFN3 = 0
IF RIFN4 > 0 THEN JSONDESTROY RIFN4 : RIFN4 = 0
IF RIFN5 > 0 THEN JSONDESTROY RIFN5 : RIFN5 = 0
IF RIFN6 > 0 THEN JSONDESTROY RIFN6 : RIFN6 = 0

'-----
'CHIUSURA
'-----
JSON_I_END: 'Chiusura Iterazione
JSON_L_END: 'Chiusura Lettura
JSON_S_END: 'Chiusura Scrittura

END

```

Esempio JSON

In seguito sono presenti esempi con istruzioni, funzioni e variabili per leggere e scrivere dei dati JSON (acronimo di JavaScript Object Notation), ovvero, un leggero formato di scambio di dati adatto all'interscambio di dati fra applicazioni client/server nell'ambito di programmazione web; è basato su linguaggio JavaScript, Standard ECMA-262 3^a Edition, ma ne è indipendente.

Creazione JSON

```

ROOTOBJ=JSONCREATE()

RIFARA=JSONADDARR(ROOTOBJ, "DOCUMENTS", 0)

RIFN1=JSONCREATE()
RIOBJD1=JSONADDOBJ(RIFARA, "OBJECT1", RIFN1 )
RF=JSONADDSTR(RIOBJD1, "ID", "1234562-2" )
RIFARB=JSONADDARR( RIOBJD1, "R:LOC", 0 )

RIFN2=JSONCREATE()
RIOBJD2=JSONADDOBJ(RIFARB, "OBJECT2", RIFN2 )
RF=JSONADDSTR(RIOBJD2, "COGNOME", "Pari" )
RF=JSONADDSTR(RIOBJD2, "NOME", "Christian " )
RF=JSONADDSTR(RIOBJD2, "CITTA", "SIRACUSA" )
RF=JSONADDSTR(RIOBJD2, "DDN", "04/04/1971" )
'RF=JSONADDSTR(RIOBJD2, "CF", "NNNCLD71T30H306D" )

```



```
RF=JSONADDSTR(RIOBJD1, "GARANTE"           , "Notarantonio Ramona")
RF=JSONADDSTR(RIOBJD1, "CF_GARANTE"        , "GARTZX66T30H144D")
RF=JSONADDSTR(RIOBJD1, "TIPO_EROG_INTER"   , "Fine contratto")
RF=JSONADDSTR(RIOBJD1, "NR_MENS_DEPOSITO"   , "3")
RF=JSONADDSTR(RIOBJD1, "LUOGO_STIPULA"     , "Rimini")
RF=JSONADDSTR(RIOBJD1, "DT_STIPULA"        , "22/01/2018" )
```

```
JSONDUMP ROOTOBJ, 3, "<c:\temp\root_dump.json"
```

```
JSONDUMP RIOBJD2, 3, "<c:\temp\OBJD2_dump.json"
```

```
JSONDESTROY(ROOTOBJ)
END
```

Esempio Sprix JSON

```
DIM STACK_I TR()
DIM STACK_HND()
DIM STACK_KEY$( )
STKI=0

JSON_OBJECT = 0
JSON_ARRAY = 1
JSON_STRING = 2
JSON_INTEGER = 3
JSON_REAL = 4
JSON_TRUE = 5
JSON_FALSE = 6
JSON_NULL = 7

JSON_ROOT = JSONLOAD("<c:\temp\input1.json")

IF JSON_ROOT <= 0 THEN VIMSG 4, _ERRJSON$:END

JSONDUMP JSON_ROOT, 3, "<c:\temp\paolo_dump.json"

JHND = JSON_ROOT
GOSUB PRINT_OBJECT

JSONDESTROY(JSON_ROOT)
I = 1
END

PRINT_OBJECT:
  STKI=STKI+1
  STACK_HND(STKI) = JHND
  STACK_I TR(STKI) = JSONSTARTI TR( STACK_HND(STKI), "" )

  ITER_LOOP:
    JSONOBJVAL = JSONINTERVAL( STACK_I TR(STKI) )
    K$=JSONITERKEY( STACK_I TR(STKI) ):
    STACK_KEY$(STKI) = K$
    JS_TYPE = JSONTYPE( JSONOBJVAL )

    IF JS_TYPE = 0 THEN OLDJHND=JHND:JHND=JSONOBJVAL:LPRINT STACK_KEY$(STKI)+": {"":GOSUB PRINT_OBJECT:LPRINT "":JHND=OLDJHND:GOTO READ_NEXT
    IF JS_TYPE = 1 THEN OLDJHND=JHND:JHND=JSONOBJVAL:LPRINT STACK_KEY$(STKI)+": [{"":GOSUB PRINT_ARRAY:LPRINT "":JHND=OLDJHND:GOTO READ_NEXT
    IF JS_TYPE = 2 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_STRING:LPRINT "":GOTO READ_NEXT
    IF JS_TYPE = 3 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_INTEGER:LPRINT "":GOTO READ_NEXT
    IF JS_TYPE = 4 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_REAL:LPRINT "":GOTO READ_NEXT
    IF JS_TYPE = 5 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_TRUE:LPRINT "":GOTO READ_NEXT
    IF JS_TYPE = 6 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_FALSE:LPRINT "":GOTO READ_NEXT
    IF JS_TYPE = 7 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_NULL

  '*** NEXT ***
  READ_NEXT:
    RET = JSONITERNXT( STACK_I TR(STKI) )
    IF RET < 0 THEN VIMSG 4, _ERRJSON$: GOTO PRINT_OBJECT_RET
    IF RET > 0 THEN GOTO PRINT_OBJECT_RET
    GOTO ITER_LOOP
  PRINT_OBJECT_RET:
    IF STKI > 1 THEN STKI = STKI - 1
  RETURN

PRINT_ARRAY:
  STKI=STKI+1
  STACK_HND(STKI) = JHND
  ARRAYSIZE = JSONARRSIZE( STACK_HND(STKI) )
  FOR I=0 TO ARRAYSIZE -1
    JSONOBJVAL = JSONGETARR( STACK_HND(STKI), I )
    JS_TYPE = JSONTYPE( JSONOBJVAL )
    IF JS_TYPE = 0 THEN OLDJHND=JHND:JHND=JSONOBJVAL:LPRINT STACK_KEY$(STKI)+": {"":GOSUB PRINT_OBJECT:LPRINT "":JHND=OLDJHND:GOTO
  ARRAY_NXT
  IF JS_TYPE = 1 THEN OLDJHND=JHND:JHND=JSONOBJVAL:LPRINT STACK_KEY$(STKI)+": [{"":GOSUB PRINT_ARRAY:LPRINT "":JHND=OLDJHND:GOTO
  ARRAY_NXT
  IF JS_TYPE = 2 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_STRING:LPRINT "":GOTO ARRAY_NXT
  IF JS_TYPE = 3 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_INTEGER:LPRINT "":GOTO ARRAY_NXT
  IF JS_TYPE = 4 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_REAL:LPRINT "":GOTO ARRAY_NXT
  IF JS_TYPE = 5 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_TRUE:LPRINT "":GOTO ARRAY_NXT
  IF JS_TYPE = 6 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_FALSE:LPRINT "":GOTO ARRAY_NXT
  IF JS_TYPE = 7 THEN LPRINT STACK_KEY$(STKI)+":":GOSUB PRINT_NULL

  ARRAY_NXT:
  NEXT
  IF STKI > 1 THEN STKI = STKI - 1
  RETURN

PRINT_STRING:
  LN=0
  IN = 0
  A$=JSONGETSTR( JSONOBJVAL )
  IF A$ = "" THEN GOTO PRINT_STRING_ARRAY
  LPRINT A$
  RETURN
  PRINT_STRING_ARRAY:
```

```

IN=0
LN=0
FOR EL=1 TO 15000
  IF _JSONOUTA$(EL) <> "" THEN IN=EL:LN=LN+LEN(_JSONOUTA$(EL)):LPRINT _JSONOUTA$(EL); ELSE EL=99999
NEXT
IF (LN > 512 ) THEN VIMSG 4, STR$(IN,0,0)+ " " +STR$(LN,0,0)
RETURN

PRINT_INTEGER:
  LPRINT STACK_KEY$(STKI)+" = "+STR$(JSONGETNUM( JSONOBJVAL ),0,1);
RETURN

PRINT_REAL:
  LPRINT STACK_KEY$(STKI)+" = "+STR$(JSONGETREAL( JSONOBJVAL ),3,1);
RETURN

PRINT_TRUE:
  LPRINT STACK_KEY$(STKI)+" = TRUE";
RETURN

PRINT_FALSE:
  LPRINT STACK_KEY$(STKI)+" = FALSE";
RETURN

PRINT_NULL:
  LPRINT STACK_KEY$(STKI)+" = NULL";
RETURN

PRINT_UNK:
  LPRINT "TIPO SCONOSCIUTO : " + STR$( JS_TYPE, 0, 0 );
RETURN

```

File "input.json"

```

{
  "web-app":
  {
    "servlet":
    [
      {
        "servlet-name": "cofaxCDS",
        "servlet-class": "org.cofax.cds.CDSServlet",
        "init-param":
        {
          "configGlossary:installationAt": "Philadelphia, PA",
          "configGlossary:adminEmail": "ksm@pobox.com",
          "configGlossary:poweredBy": "Cofax",
          "configGlossary:poweredByIcon": "/images/cofax.gif",
          "configGlossary:staticPath": "/content/static",
          "templateProcessorClass": "org.cofax.WysiwygTemplate",
          "templateLoaderClass": "org.cofax.FilesTemplateLoader",
          "templatePath": "templates",
          "templateOverridePath": "",
          "defaultListTemplate": "listTemplate.htm",
          "defaultFileTemplate": "articleTemplate.htm",
          "useJSP": false,
          "jspListTemplate": "listTemplate.jsp",
          "jspFileTemplate": "articleTemplate.jsp",
          "cachePackageTagsTrack": 200,
          "cachePackageTagsStore": 200,
          "cachePackageTagsRefresh": 60,
          "cacheTemplatesTrack": 100,
          "cacheTemplatesStore": 50,
          "cacheTemplatesRefresh": 15,
          "cachePagesTrack": 200,
          "cachePagesStore": 100,
          "cachePagesRefresh": 10,
          "cachePagesDirtyRead": 10,
          "searchEngineListTemplate": "forSearchEnginesList.htm",
          "searchEngineFileTemplate": "forSearchEngines.htm",
          "searchEngineRobotsDb": "WEB-INF/robots.db",
          "useDataStore": true,
          "dataStoreClass": "org.cofax.SqlDataStore",
          "redirectionClass": "org.cofax.SqlRedirection",
          "dataStoreName": "cofax",
          "dataStoreDriver": "com.microsoft.jdbc.sqlserver.SQLServerDriver",
          "dataStoreUrl": "jdbc:microsoft:sqlserver://LOCALHOST:1433;DatabaseName=goon",
          "dataStoreUser": "admin",
          "dataStorePassword": "DtStTtQy",
          "dataStoreTestQuery": "SET NOCOUNT ON;select test='test';",
          "dataStoreLogFile": "/usr/local/tomcat/logs/datastore.log",
          "dataStoreInitConns": 10,
          "dataStoreMaxConns": 100,
          "dataStoreConnUsageLimit": 100,
          "dataStoreLogLevel": "debug",
          "maxUrlLength": 500
        }
      },
      {
        "servlet-name": "cofaxEmail",
        "servlet-class": "org.cofax.cds.EmailServlet",
        "init-param":
        {
          "mailHost": "mail1",
          "mailHostOverride": "mail2"
        }
      },
      {
        "servlet-name": "cofaxAdmin",
        "servlet-class": "org.cofax.cds.AdminServlet"
      }
    ],
    {
      "servlet-name": "fileServlet",
      "servlet-class": "org.cofax.cds.FileServlet"
    }
  }
}

```

```

{
  "servlet-name": "cofaxTools",
  "servlet-class": "org.cofax.cms.CofaxToolsServlet",
  "init-param":
  {
    "templatePath": "toolstemplates/",
    "log": 1,
    "logLocation": "/usr/local/tomcat/logs/CofaxTools.log",
    "logMaxSize": "",
    "dataLog": 1,
    "dataLogLocation": "/usr/local/tomcat/logs/dataLog.log",
    "dataLogMaxSize": "",
    "removePageCache": "/content/admin/remove?cache=pages&id=",
    "removeTemplateCache": "/content/admin/remove?cache=templates&id=",
    "fileTransferFolder": "/usr/local/tomcat/webapps/content/fileTransferFolder",
    "lookInContext": 1,
    "adminGroupID": 4,
    "betaServer": true
  }
},
"servlet-mapping":
{
  "cofaxCDS": "/",
  "cofaxEmail": "/cofaxutil/aemail/*",
  "cofaxAdmin": "/admin/*",
  "fileServlet": "/static/*",
  "cofaxTools": "/tools/*"
},
>taglib":
{
  "taglib-uri": "cofax.tld",
  "taglib-location": "/WEB-INF/tlds/cofax.tld"
}
}

```

XML

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
XMLINIT	Istruzione che inizializza la libreria Xml per la creazione Xml	S/C/Cr/Cd
XMLCLEANUP	Istruzione creazione Xml	S/C/Cr/Cd
XMLNEWDOC()	Par: (<NomeNodoRadice\$>,<ValoreNodoRadice\$>). Funzione creazione Xml	S/C/Cr/Cd
XMLDOCADDCOM	Par: <RifDocumento>,<TestoCommento\$>. Istruzione creazione Xml	S/C/Cr/Cd
XMLADDELE()	Par: (<RifNodo>,<NomeElemento\$>,<ValoreElemento\$>). Funzione creazione Xml	S/C/Cr/Cd
XMLSETNODVAL	Par: <RifNodo>,<Valore\$>. Istruzione creazione Xml applicabile solo a nodi di tipo TEXT_NODE e CDATA_NODE	S/C/Cr/Cd
XMLADDNODVAL	Par: <RifNodo>,<Valore\$>. Istruzione creazione Xml	S/C/Cr/Cd
XMLINSDATI	Par: <FileXML\$>, <Tag\$()>, <FileOutput\$()> Istruzione di variazione nodi di Xml	S/C/Cr/Cd
XMLUNLINKNOD()	Par: (<RifNodo>). Istruzione creazione Xml	S/C/Cr/Cd
XMLLEADDCOMM()	Par: (<RifNodo>,<TestoCommento\$>). Funzione creazione Xml	S/C/Cr/Cd
XMLADDATTR()	Par: (<RifNodo>,<Nome\$>,<Valore\$>). Funzione creazione Xml	S/C/Cr/Cd
XMLADDNS()	Par: (<RifNodo>,<NSPrefix\$>,<NSHref\$>). Funzione creazione Xml	S/C/Cr/Cd
XMLADDCDATA()	Par: (<RifDoc>,<RifNodo>,<Nome\$>,<Valore\$>). Funzione creazione Xml	S/C/Cr/Cd
XMLDUMPFIL()	Par: (<RifDoc>,<NomeFile\$>,<Encoding\$>). Funzione creazione Xml	S/C/Cr/Cd
XMLFREEDOC()	Par: (<RifDocumento>). Istruzione creazione Xml	S/C/Cr/Cd
XMLLOAD()	Par: (<NomeFile\$>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETROOT()	Par: (<RifDocumento\$>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETSIBL()	Par: (<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETCHILD()	Par: (<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETPARENT()	Par: (<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETNCONT\$()	Par: (<RifDoc>,<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETELNAM\$()	Par: (<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETELVAL\$()	Par: (<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETELTYPE()	Par: (<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETFSTATT()	Par: (<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETNXTATT()	Par: (<RifAttr>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETATNAM\$()	Par: (<RifAttr>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETATVAL\$()	Par: (<RifAttr>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETATTNS()	Par: (<RifAttr>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETNODENS()	Par: (<RifNodo>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETNSPRE\$()	Par: (<RifNameSpace>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLGETNSHRE\$()	Par: (<RifNameSpace>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLFINDELEM()	Par: (<RifNodo>,<Nome\$>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLFINDATTR()	Par: (<RifNodo>,<Nome\$>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLFINDELNS()	Par: (<RifNodo>,<NomeElemento\$>,<NSPrefix\$>,<NSHref\$>,<Depth>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLTRIMMA\$()	Par: (<ValoreRaw\$>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLISBLANK()	Par: (<ValoreRaw\$>). Funzione lettura e navigazione Xml	S/C/Cr/Cd
XMLSRLOADFIL()	Par: (<NomeFile\$>). Funzione lettura Xml in modalit� STREAM	S/C/Cr/Cd
XMLSRFREE()	Par: (<RifXmlSTREAM>). Istruzione lettura Xml in modalit� STREAM	S/C/Cr/Cd
XMLSRWALK()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalit� STREAM	S/C/Cr/Cd
XMLSRTYPE()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalit� STREAM	S/C/Cr/Cd

XMLSRNAME\$()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalità STREAM	S/C/Cr/Cd
XMLSRVALUE\$()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalità STREAM	S/C/Cr/Cd
XMLSRDEPTH()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalità STREAM	S/C/Cr/Cd
XMLSRFRSTATT()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalità STREAM	S/C/Cr/Cd
XMLSNXTATTR()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalità STREAM	S/C/Cr/Cd
XMLSRNSPREF\$()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalità STREAM	S/C/Cd
XMLSRNSURI\$()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalità STREAM	S/C/Cd
XMLSRNSATTR()	Par: (<RifXmlSTREAM>). Funzione lettura Xml in modalità STREAM	S/C/Cd
ENCINIT()	Par: (<EncodingIN\$>,<EncodingOUT\$>). Funzione encoding Xml	S/C/Cr/Cd
ENCODING\$()	Par: (<EncodingREF>,<TextIn\$>). Funzione encoding Xml	S/C/Cr/Cd
ENCFREE()	Par: (<EncodingREF>). Istruzione encoding Xml	S/C/Cr/Cd
_ERRXML\$	Descrizione errore operazione Xml	S/C/Cd
_XML	variabili Xml	S/C/Cd

esempi e casi

ATTENZIONE

alcuni esempi Collage relativi alla modifica dei documenti di magazzino in formato elettronico XML (sia tramite variabili `_FTE` all'evento `ON_STP_XML`, che tramite l'handle `_FTE_DOCXML` per l'accesso all'uso di questi comandi) sono riportati negli esempi del capitolo DOCUMENTI DI MAGAZZINO (MM) relativo ai MODULI E ARCHIVI DEL GESTIONALE nella sezione COMANDI PER ARGOMENTO

Si riporta un accenno sui principi base di gestione del file XML

XML è un sottoinsieme di SGML (Standard Generalized Markup Language). Si tratta di formato universale per lo scambio di dati sul web. Questo è caratterizzato da una struttura ad albero dove gli elementi (tag) sono identificati da nomi racchiusi tra "<>". In pratica è una struttura che si basa sul concetto di padri/figli/fratelli dove il padre è il nodo che contiene i nodi "figli", i nodi figli a loro volta possono essere padri di altri figli, quindi contenere a loro volta dei nodi figli. I nodi che sono allo stesso livello si possono equiparare a "fratelli" (in pratica figli dello stesso padre).

Esempio :

```
<?xml version="1.0" encoding="utf-8"?>
<documento>
  <sezione id="sezione-a">
    <titolo livello="1">Titolo</titolo>
    <para>...</para>
  </sezione>
</documento>
```

Padre di tutti
Figlio di "documento"
figlio di "sezione" e fratello di "para"
figlio di "sezione" e fratello di "titolo"

Ogni "tag" deve necessariamente essere aperto e chiuso e non si può intersecare. In pratica non si può chiudere il tag del padre se prima non si è chiuso il/i tag tag dei figli.

Esempio errato

```
<?xml version="1.0" encoding="utf-8"?>
<documento></documento>
<elemento></elemento>
```

Gli elementi devono essere correttamente annidati. Quindi:

Esempio errato

```
<elemento>
  <para>
</elemento>
</para>
```

Esempio Corretto

```
<elemento>
  <para></para>
</elemento>
```

Per una corretta interpretazione del file xml bisogna inserire il prologo che serve a definire:

- versione di XML in uso (attributo version)
- codifica del documento (attributo encoding)

Esempio:

```
<?xml version="1.0" encoding="utf-8" >
<documento></documento>
```

Gli attributi devono essere sempre racchiusi tra virgolette. Quindi:

Esempio Errato

```
<elemento nome=valore></elemento>
```

Esempio Corretto

```
<elemento nome="valore"></elemento>
```

Caratteri speciali

I caratteri speciali ovvero quei caratteri speciali che l'XML utilizza per identificare elementi ed attributi, devono essere rappresentati con una particolare sequenza di escape

Esempio

```
&amp; ; (&)
&lt; ; (<)
&gt; ; (>)
&apos; ; (')
&quot; ; (")
```

Approfondimenti sono disponibili nella documentazione specialistica disponibile on-line. Esempi
<http://xmlsoft.org> - <http://lia.disi.unibo.it/Courses/ModApplRetiCalc0910/materiale/SGML,HTML,XML.pdf>

Elenco di CHR\$ e relative stringhe di escape per alcuni caratteri

```
CHR$(38) = "&";
CHR$(224) = "&agrave;";
CHR$(225) = "&aacute;";
CHR$(232) = "&egrave;";
CHR$(233) = "&eacute;";
CHR$(236) = "&igrave;";
CHR$(237) = "&iacute;";
CHR$(242) = "&ograve;";
CHR$(243) = "&oacute;";
CHR$(249) = "&ugrave;";
```

```
CHR$(250) = "&uacute;"
"'"      = "&apos;"
CHR$(34) = "&quot;"
"<"     = "&lt;"
">"     = "&gt;"
```

Si devono sostituire tutti i caratteri speciali con i relativi escape e viceversa. Esempio: si possono usare array operando carattere per carattere e se nelle stringhe si incontra il CHR\$() dell'elemento di sinistra lo si sostituisce col relativo "escape" presente nell'elemento di destra. RPLC\$(01,1)=CHR\$(38): RPLC\$(01,2)="&". Quindi se si trova CHR\$(38), ovvero '&', lo si sostituisce con "&"

Esempio1

Creazione

MAIN:

```
'Inizializzo l'ambiente XML
XMLINIT

'Crea il nuovo documento XML
NEWDOC = XMLNEWDOC("ObjectTable","Tabella")
IF NEWDOC=0 OR _ERRXML$<>"" THEN VIMSG 5,"Impossibile creare il documento: " + _ERRXML$: END

'Aggiunge un commento al documento
XMLDOCADDCOM NEWDOC, "Questo e un commento al documento"
IF _ERRXML$<>"" THEN VIMSG 5,"Impossibile aggiungere il commento: "+_ERRXML$: GOTO FINE:

'Cerco il riferimento al nodo principale
ROOTNODE = XMLGETROOT(NEWDOC)
IF ROOTNODE=0 OR _ERRXML$<>"" THEN VIMSG 5," Impossibile trovare il nodo: " + _ERRXML$: GOTO FINE:

'Aggiunge il namespace 'soapenv_ con href 'http://schemas.xmlsoap.org/soap/envelope/_ al nodo radice
NAMSPC=XMLADDS (ROOTNODE,"soapenv","http://schemas.xmlsoap.org/soap/envelope/")
IF NAMSPC=0 OR _ERRXML$<>"" THEN VIMSG 5,"Impossibile aggiungere il namespace: "+_ERRXML$: GOTO FINE:

'Aggiunge un elemento con nome "FieldList" al nodo ROOTNODE.
NEWNODE = XMLADDELE(ROOTNODE, "FieldList", "" )
IF NEWNODE=0 OR _ERRXML$<>"" THEN VIMSG 5, "Impossibile creare il nodo 'FieldList_": "+_ERRXML$: GOTO FINE:

'Aggiunge un elemento con nome "FieldName" e valore "Descrizione Cliente" al nodo ROOTNODE.
NEWNODE = XMLADDELE(NEWNODE, "FieldName", "Descrizione Cliente" )
IF NEWNODE=0 OR _ERRXML$<>"" THEN VIMSG 5,"Impossibile creare il nodo 'FieldName_": "+_ERRXML$:GOTO FINE:

'Modifica il valore dell elemento "FieldName" con il testo "Codice Cliente".
XMLSETNODVAL NEWNODE, "CodiceCliente"
IF _ERRXML$<>"" THEN VIMSG 5,"Impossibile impostare il valore del nodo:"+_ERRXML$: GOTO FINE:

'Aggiunge all elemento "FieldName" il con il testo " e Descrizione Cliente".
XMLSETNODVAL NEWNODE, "e Descrizione Cliente"
IF _ERRXML$<>"" THEN VIMSG 5,"Impossibile impostare il valore del nodo: "+_ERRXML$: GOTO FINE:

'Aggiunge un commento al nodo
COMMENTO = XMLLADDCOMM(NEWNODE, "Questo e un commento.")
IF COMMENTO=0 OR _ERRXML$<>"" THEN VIMSG 5,"Impossibile aggiungere commento: "+_ERRXML$: GOTO FINE:

'Aggiunge l'attributo required con valore "1" al nodo
ATTRIB = XMLADDATTR(NEWNODE,"required", "1")
IF ATTRIB=0 OR _ERRXML$<>"" THEN VIMSG 5,"Impossibile aggiungere l'attributo"+_ERRXML$: GOTO FINE:

'Elimina il nodo COMMENTO
XMLUNLINKNOD(COMMENTO)
IF _ERRXML$<>"" THEN VIMSG 5,"Impossibile eliminare il commento: "+_ERRXML$: GOTO FINE:

'Inserimento di una nodo di testo semplice riga XCDATA
NODOTESTO=XMLADDCDATA(NEWDOC, NEWNODE, "Testo semplice", "Riga di testo semplice, non interpretato.")
IF NODOTESTO=0 OR _ERRXML$<>"" THEN VIMSG 5,"Impossibile aggiungere il testo CDATA: "+_ERRXML$: GOTO FINE:

'Scrittura del documento XML sul file
ESITO = XMLDUMPPFILE (NEWDOC, "<C:\Prova.xml", "UTF-8")
IF ESITO=0 OR _ERRXML$<>"" THEN VIMSG 5,"Impossibile scrivere il file: "+_ERRXML$: GOTO FINE:

'Rilascio del documento XML
XMLFREEDOC (NEWDOC)

FINE:
'Rilascio dell'ambiente XML
XMLCLEANUP
```

Encoding

```
'Inizializzo l'ambiente XML
XMLINIT

'Inizializzo l'ambiente per l_encoding da ISO/IEC ("ISO-8859-1") ad Unicode ("UTF-8")
ENC_REF = ENCCINIT("ISO-8859-1", " UTF-8")
IF ENC_REF=0 OR _ERRXML$<>"" THEN GOTO FINE_ENC:
TEXT_OUT$= ENCODINGS(ENC_REF, "Prova encoding caratteri speciali "+CHR$(232)+CHR$(233))
IF _ERRXML$<>"" THEN GOTO FINE_ENC:

FINE_ENC:
ENCFREE(ENC_REF)
'Rilascio dell'ambiente XML
XMLCLEANUP

END
```

Esempio2

Dato il seguente file Xml

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <checkVatResponse xmlns="servizio.prova.it:metodoprova">
      <countryCode>IT</countryCode>
      <vatNumber>SM03473</vatNumber>
      <requestDate>2017-02-07+01:00</requestDate>
```



```

<valid>true</valid>
<name>PASSEPARTOUT s.p.a.</name>
<address>Via Consiglio dei Sessanta, 99 - 47891 Dogana Repubblica di San Marino</address>
</checkVatResponse>
</soapenv:Body>
</soapenv:envelope>

```

Esempio di scrittura del suddetto file XML simile alla risposta del WS di verifica partita IVA dell'Unione Europea

XMLINIT

```

`Creazione nuovo XML
XMLDOC=XMLNEWDOC("soapenv:envelope",""):
IF XMLDOC=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:

NODE1=XMLGETROOT(XMLDOC):
IF NODE1=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:

`aggiungo namespace
NODE2=XMLADDNS(NODE1,"soapenv","http://schemas.xmlsoap.org/soap/envelope/")

`aggiungo nodo 'body'
NODE3=XMLADDELE(NODE1,"soapenv:Body",""): IF NODE3=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:

`aggiungo nodo figlio al nodo body...
NODE4=XMLADDELE(NODE3,"checkVatResponse",""): IF NODE4=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:

`aggiungo attributo al nodo
NODE5=XMLADDATTR(NODE4,"xmlns","servizio.prova.it:metodoprova")

`aggiungo altri nodi (figli del nodo 'checkVatResponse')
NODE6=XMLADDELE(NODE4,"countryCode","IT"): IF NODE4=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:
NODE6=XMLADDELE(NODE4,"vatNumber","SM03473"): IF NODE4=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:
NODE6=XMLADDELE(NODE4,"requestDate","2017-02-07+01:00"): IF NODE4=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:
NODE6=XMLADDELE(NODE4,"valid","true"): IF NODE4=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:
NODE6=XMLADDELE(NODE4,"name","PASSEPARTOUT s.p.a."): IF NODE4=0 THEN ERR=1: GOTO ERROR_CREATE_XMLDOC:
NODE6=XMLADDELE(NODE4,"address","Via Consiglio dei Sessanta, 99 - 47891 Dogana Repubblica di San Marino"): IF NODE4=0 THEN ERR=1: GOTO
ERROR_CREATE_XMLDOC:

ERROR_CREATE_XMLDOC:

IFF ERR=0
ESITO = XMLDUMPFIL (XMLDOC,"<c:\test_xml.xml","UTF-8")
IF ESITO=0 OR _ERRXML$<>" THEN VIMSG 5,"Impossibile scrivere il file: "+_ERRXML$: ELSE VIMSG 5,"File XML scritto correttamente !!!"
ENDF
IF XMLDOC<>0 THEN XMLFREEDOC(XMLDOC)
IF ERR<>0 THEN VIMSG 5,"ERRORE|Errore durante la creazione del file XML !":

```

Esempio di lettura del suddetto file XML simile alla risposta del WS di verifica partita IVA dell'Unione Europea

XMLINIT

```

`Apertura del file
XMLDOC = XMLLOAD("<c:\test_xml.xml")
IF XMLDOC=0 OR _ERRXML$<>" THEN VIMSG 5, "Impossibile aprire il file " + _ERRXML$: GOTO ERR:

`Rintraccio il nodo principale...
R=XMLGETROOT(XMLDOC)
IF R=0 THEN VIMSG 5, "Impossibile trovare il nodo radice" + _ERRXML$: GOTO ERR:

`Cerco il nodo con nome 'countryCode', a partire dal nodo principale
N=XMLFINDELEM(R,"countryCode")
IF N=0 THEN VIMSG 5, "Impossibile trovare il nodo 'countryCode' "+ _ERRXML$: GOTO ERR:
`Leggo il contenuto del nodo...
COUNTRY$ = XMLGETNCONT$(XMLDOC,N)
IF _ERRXML$<>" THEN VIMSG 5,_ERRXML$: GOTO ERR:

`Cerco il nodo con nome 'vatNumber', a partire dal nodo principale
N=XMLFINDELEM(R,"vatNumber")
IF N=0 THEN VIMSG 5, "Impossibile trovare il nodo 'vatNumber' "+ _ERRXML$: GOTO ERR:
`Leggo il contenuto del nodo...
VAT$ = XMLGETNCONT$(XMLDOC,N)
IF _ERRXML$<>" THEN VIMSG 5,_ERRXML$: GOTO ERR:

`Cerco il nodo con nome 'requestDate', a partire dal nodo principale
N=XMLFINDELEM(R,"requestDate")
IF N=0 THEN VIMSG 5, "Impossibile trovare il nodo 'requestDate' "+ _ERRXML$: GOTO ERR:
`Leggo il contenuto del nodo...
DATE$ = XMLGETNCONT$(XMLDOC,N)
IF _ERRXML$<>" THEN VIMSG 5,_ERRXML$: GOTO ERR:

`Cerco il nodo con nome 'valid', a partire dal nodo principale
N=XMLFINDELEM(R,"valid")
IF N=0 THEN VIMSG 5, "Impossibile trovare il nodo 'valid' "+ _ERRXML$: GOTO ERR:
`Leggo il contenuto del nodo...
VALID$ = XMLGETNCONT$(XMLDOC,N)
IF _ERRXML$<>" THEN VIMSG 5,_ERRXML$: GOTO ERR:

`Cerco il nodo con nome 'name', a partire dal nodo principale
N=XMLFINDELEM(R,"name")
IF N=0 THEN VIMSG 5, "Impossibile trovare il nodo 'name' "+ _ERRXML$: GOTO ERR:
`Leggo il contenuto del nodo...
NAME$ = XMLGETNCONT$(XMLDOC,N)
IF _ERRXML$<>" THEN VIMSG 5,_ERRXML$: GOTO ERR:

`Cerco il nodo con nome 'address', a partire dal nodo principale
N=XMLFINDELEM(R,"address")
IF N=0 THEN VIMSG 5, "Impossibile trovare il nodo 'address' "+ _ERRXML$: GOTO ERR:
`Leggo il contenuto del nodo...
ADDRESS$ = XMLGETNCONT$(XMLDOC,N)
IF _ERRXML$<>" THEN VIMSG 5,_ERRXML$: GOTO ERR:

VIMSG 2,"nazione: "+COUNTRY$+"|partita iva: "+VAT$+"|data: "+DATE$+"|valida: "+VALID$+"|nome: "+NAME$+"|indirizzo: "+ADDRESS$

ERR:
XMLCLEANUP

END

```

Caso: inserimento come contenuto di un nodo un file .zip convertito codificato in Base64.

Funzioni SPRIX che consentono tale operazione

Istruzioni di gestione file compressi .ZIP (ZIP_CREA, ZIP_ESTRAI, ZIP_LISTA)
 Istruzione di codifica/decodifica Base64 di un file con un contenuto qualsiasi (CODIF_BASE64)
 E' possibile variare il contenuto di un nodo in un file XML (XMLINSDATI)

Esempio XMLINSDATI

```
...
.....
.....
```

Se il contenuto di <FileXML\$> fosse il seguente...

```
<?xml version="1.0" encoding="utf-8"?>
<bookstore xmlns:ele="http://www.soapui.org/book/" xmlns:eledat="http://www.soapui.org/dati/">
  <ele:book category="Digital documents">
    <eledat:dati></eledat:dati>
    <eledat:title lang="en">PDF Explained</eledat:title>
    <eledat:author>John Whittington</eledat:author>
    <eledat:year>2011</eledat:year>
    <eledat:price>30.00</eledat:price>
  </ele:book>
  <ele:book category="Programming">
    <eledat:title lang="en">RESTful Web Services</eledat:title>
    <eledat:author>Leonard Richardson</eledat:author>
    <eledat:author>Sam Ruby</eledat:author>
    <eledat:year>2007</eledat:year>
    <eledat:price>29.99</eledat:price>
  </ele:book>
  <ele:book category="Programming">
    <eledat:dati></eledat:dati>
    <eledat:title lang="en">RESTful Web Services Cookbook</eledat:title>
    <eledat:author>Subbu Allamaraju</eledat:author>
    <eledat:year>2010</eledat:year>
    <eledat:price>49.99</eledat:price>
  </ele:book>
  <ele:book category="Signals" cover="paperback">
    <eledat:dati></eledat:dati>
    <eledat:title lang="en">Tools for Signal Compression</eledat:title>
    <eledat:author>Nicolas Moreau</eledat:author>
    <eledat:year>2011</eledat:year>
    <eledat:price>39.95</eledat:price>
  </ele:book>
</bookstore>
```

...ed i valori evidenziati sono quelli da cambiare con dei valori (contenuti in file), lo SPRIX andrebbe scritto così

```
DIM STAG$()
DIM SFILE$()

STAG$(1) = "bookstore->ele:book[1]->eledat:price"
SFILE$(1) = "Price_PDF_Explained.txt"

STAG$(2) = "bookstore->ele:book[2]->eledat:author[2]"
SFILE$(2) = "Author_RestfulWs.txt"

STAG$(3) = "bookstore->ele:book[3]->eledat:year"
SFILE$(3) = "Year_Restful_cookbook.txt"

STAG$(4) = "bookstore->ele:book[4]->eledat:price"
SFILE$(4) = "Price_Toolforsignals.txt"

FILEXML$ = "jsp.xml"

XMLINSDATI FILEXML$, STAG$, SFILE$

IF _ERRXML <> "" THEN VIMSG 5, "Errore : "+_ERRXML
END
```

ESEMPIO DI LETTURA E STAMPA IN CASO DI TAG OMONIMI E PRESENTI A DIVERSI LIVELLI DI PROFONDITA' E CON CONTENUTI DIVERSI

Dato il seguente documento XML si vuole eseguire una lettura con una determinata sequenza di informazioni, come sono riportate dal documento stesso. Si noti che il documento contiene tag <item> con lo stesso nome e a diversi livelli di profondità. Ciascun <item> riporta informazioni relative al intestatario <utente> del documento, ma ciascun <utente> potrebbe (o meno) riportare delle <righe> aggiuntive.

In questo tipo di semantica del XML, l'eventuale utilizzo di XMLFINDELEM(ITEM,"righe") per accedere alle suddette <righe> provoca l'accesso immediato dopo il primo <utente> anziché dopo i primi tre <utente>, poiché l'<item> si trova a diversi livelli di profondità e riporta il medesimo nome tag <item>; di conseguenza la semantica del documento XML richiede l'utilizzo del parametro di profondità disponibile in XMLFINDELNS(ITEMCHILD,"righe", "", "", 2).

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <documenti>
    <item>
      <id>1</id>
      <anno>2017</anno>
      <numero>1</numero>
      <tipo>0</tipo>
      <data_emissione>2017-10-03</data_emissione>
      <iva>0.22</iva>
      <stato>1</stato>
      <esportato>0</esportato>
      <totale>0.00</totale>
      <totale_imponibile>0.00</totale_imponibile>
      <totale_iva>0.00</totale_iva>
      <utente>
        <id>5</id>
        <nome>Pippo</nome>
        <cognome>Calopresti</cognome>
        <codice_fiscale>PPOCLP90B12Q227A</codice_fiscale>
        <email>pippo.calopresti7@gmail.com</email>
        <luogo_nascita>Firenze (le Vigne di Querceto)</luogo_nascita>
```

```

<indirizzo>Via delle rose 72</indirizzo>
<comune>Rimini</comune>
<provincia>Rn</provincia>
<cap>47023</cap>
<sdi>000</sdi>
</utente>
</item>
<item>
<id>2</id>
<anno>2017</anno>
<numero>2</numero>
<tipo>0</tipo>
<data_emissione>2017-10-03</data_emissione>
<iva>0.22</iva>
<stato>1</stato>
<esportato>0</esportato>
<totale>0.00</totale>
<totale_imponibile>0.00</totale_imponibile>
<totale_iva>0.00</totale_iva>
<utente>
<id>27</id>
<nome>Giuseppe</nome>
<cognome>Giacobazzi</cognome>
<codice_fiscale>GPEGBZ55U04Z944Q</codice_fiscale>
<email>fomagnolo.gnurent@gmail.com</email>
<luogo_nascita>SOLAROLO (RA)</luogo_nascita>
<indirizzo>via lungo fiume 153</indirizzo>
<comune>Cesenatico</comune>
<provincia>Forlì Cesena</provincia>
<cap>47867</cap>
<sdi>000</sdi>
</utente>
</item>
<item>
<id>3</id>
<anno>2017</anno>
<numero>3</numero>
<tipo>0</tipo>
<data_emissione>2017-10-03</data_emissione>
<iva>0.22</iva>
<stato>1</stato>
<esportato>0</esportato>
<totale>2.00</totale>
<totale_imponibile>1.64</totale_imponibile>
<totale_iva>0.36</totale_iva>
<utente>
<id>35</id>
<nome>Signorina</nome>
<cognome>Silvani</cognome>
<codice_fiscale>SGASVI45K33U276Z</codice_fiscale>
<email>madama.dorè45@yahoo.it</email>
<luogo_nascita>ANCONA (AN)</luogo_nascita>
<indirizzo>Via di Orzinuovi 376</indirizzo>
<comune>Bologna</comune>
<provincia>Bo</provincia>
<cap>40128</cap>
<sdi>000</sdi>
</utente>
<riga>
<item>
<id>1</id>
<operazione>1</operazione>
<tipo>4</tipo>
<descrizione>costi vari di servizio aggiuntivo</descrizione>
<ordine>1</ordine>
<perc_commissione></perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>78</oggetto_id>
<data>2017-09-19 09:28:13</data>
<iva>0.22</iva>
<importo>1.00</importo>
<totale_imponibile>0.82</totale_imponibile>
<totale_iva>0.18</totale_iva>
</item>
<item>
<id>6</id>
<operazione>1</operazione>
<tipo>4</tipo>
<descrizione>costi vari di servizio aggiuntivo</descrizione>
<ordine>3</ordine>
<perc_commissione></perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>48</oggetto_id>
<data>2017-09-21 09:02:48</data>
<iva>0.22</iva>
<importo>1.00</importo>
<totale_imponibile>0.82</totale_imponibile>
<totale_iva>0.18</totale_iva>
</item>
</riga>
</item>
<item>
<id>4</id>
<anno>2017</anno>
<numero>4</numero>
<tipo>0</tipo>
<data_emissione>2017-10-03</data_emissione>
<iva>0.22</iva>
<stato>1</stato>
<esportato>0</esportato>
<totale>4.51</totale>
<totale_imponibile>3.70</totale_imponibile>
<totale_iva>0.81</totale_iva>
<utente>
<id>6</id>
<nome>Giulio Cesare</nome>
<cognome>Imperatore</cognome>
<codice_fiscale></codice_fiscale>
<email>avecesare@morituritesalutant.it</email>

```

```

<luogo_nascita>Roma</luogo_nascita>
<indirizzo>via Appia, 344</indirizzo>
<comune>Roma</comune>
<provincia>Roma</provincia>
<cap>00100</cap>
<sdi>000</sdi>
</utente>
<righe>
<item>
<id>2</id>
<operazione>0</operazione>
<tipo>1</tipo>
<descrizione>costi trasporto speciale</descrizione>
<ordine>1</ordine>
<perc_commissione>0.06000</perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>78</oggetto_id>
<data>2017-09-20 13:48:37</data>
<iva>0.22</iva>
<importo>2.66</importo>
<totale_imponibile>2.18</totale_imponibile>
<totale_iva>0.48</totale_iva>
</item>
<item>
<id>3</id>
<operazione>0</operazione>
<tipo>2</tipo>
<descrizione>maggiorazione spese servizi segreti</descrizione>
<ordine>2</ordine>
<perc_commissione>0.03000</perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>78</oggetto_id>
<data>2017-09-20 13:48:37</data>
<iva>0.22</iva>
<importo>1.50</importo>
<totale_imponibile>1.23</totale_imponibile>
<totale_iva>0.27</totale_iva>
</item>
<item>
<id>4</id>
<operazione>0</operazione>
<tipo>3</tipo>
<descrizione>quota videosorveglianza e guardianaggio</descrizione>
<ordine>3</ordine>
<perc_commissione></perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>78</oggetto_id>
<data>2017-09-20 13:48:37</data>
<iva>0.22</iva>
<importo>0.35</importo>
<totale_imponibile>0.29</totale_imponibile>
<totale_iva>0.06</totale_iva>
</item>
</righe>
</item>
<item>
<id>5</id>
<anno>2017</anno>
<numero>5</numero>
<tipo>0</tipo>
<data_emissione>2017-10-03</data_emissione>
<iva>0.22</iva>
<stato>1</stato>
<esportato>0</esportato>
<totale>6.65</totale>
<totale_imponibile>5.46</totale_imponibile>
<totale_iva>1.19</totale_iva>
<utente>
<id>12</id>
<ragione_sociale>Il Giardino delle Fate srl</ragione_sociale>
<indirizzo>via Carlo Magno, 101/c</indirizzo>
<cap>41400</cap>
<comune>Napoli</comune>
<provincia>Napoli</provincia>
<partita_iva>IT44714427709</partita_iva>
<codice_fiscale>GRDFTE45M77A944G</codice_fiscale>
<email>magamago.merlino@gmail.com</email>
</utente>
<righe>
<item>
<id>7</id>
<operazione>0</operazione>
<tipo>1</tipo>
<descrizione>costi per magie e incantesimi</descrizione>
<ordine>1</ordine>
<perc_commissione>0.06000</perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>48</oggetto_id>
<data>2017-09-25 11:46:39</data>
<iva>0.22</iva>
<importo>4.02</importo>
<totale_imponibile>3.30</totale_imponibile>
<totale_iva>0.72</totale_iva>
</item>
<item>
<id>8</id>
<operazione>0</operazione>
<tipo>2</tipo>
<descrizione>maggiorazione per riparazione danni</descrizione>
<ordine>2</ordine>
<perc_commissione>0.03000</perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>48</oggetto_id>
<data>2017-09-25 11:46:39</data>
<iva>0.22</iva>
<importo>2.28</importo>
<totale_imponibile>1.87</totale_imponibile>
<totale_iva>0.41</totale_iva>
</item>

```

```

<item>
  <id>9</id>
  <operazione>0</operazione>
  <tipo>3</tipo>
  <descrizione>quota fissa aggiornamento formule e pozioni magiche</descrizione>
  <ordine>3</ordine>
  <perc_commissione></perc_commissione>
  <fattura_coupon_id></fattura_coupon_id>
  <oggetto_id>48</oggetto_id>
  <data>2017-09-25 11:46:39</data>
  <iva>0.22</iva>
  <importo>0.35</importo>
  <totale_imponibile>0.29</totale_imponibile>
  <totale_iva>0.06</totale_iva>
</item>
</fighe>
</item>
<item>
  <id>6</id>
  <anno>2017</anno>
  <numero>6</numero>
  <tipo>0</tipo>
  <data_emissione>2017-11-30</data_emissione>
  <iva>0.22</iva>
  <stato>1</stato>
  <esportato>0</esportato>
  <totale>69.10</totale>
  <totale_imponibile>56.66</totale_imponibile>
  <totale_iva>12.44</totale_iva>
  <utente>
    <id>12</id>
    <ragione_sociale>Lupo Alberto srl</ragione_sociale>
    <indirizzo>via Mascarella, 144</indirizzo>
    <cap>40122</cap>
    <comune>Bologna</comune>
    <provincia>Bologna</provincia>
    <partita_iva>IT33414425509</partita_iva>
    <codice_fiscale>LP0ABT76M45U944Z</codice_fiscale>
    <email>silver.design@hotmail.com</email>
  </utente>
</fighe>
<item>
  <id>11</id>
  <operazione>0</operazione>
  <tipo>1</tipo>
  <descrizione>costo per pubblicazione</descrizione>
  <ordine>1</ordine>
  <perc_commissione>0.04750</perc_commissione>
  <fattura_coupon_id></fattura_coupon_id>
  <oggetto_id>314</oggetto_id>
  <data>2017-11-12 20:03:00</data>
  <iva>0.22</iva>
  <importo>10.45</importo>
  <totale_imponibile>8.57</totale_imponibile>
  <totale_iva>1.88</totale_iva>
</item>
<item>
  <id>12</id>
  <operazione>0</operazione>
  <tipo>2</tipo>
  <descrizione>supplemento edizioni speciali</descrizione>
  <ordine>2</ordine>
  <perc_commissione>0.03400</perc_commissione>
  <fattura_coupon_id></fattura_coupon_id>
  <oggetto_id>314</oggetto_id>
  <data>2017-11-12 20:03:00</data>
  <iva>0.22</iva>
  <importo>7.48</importo>
  <totale_imponibile>6.13</totale_imponibile>
  <totale_iva>1.35</totale_iva>
</item>
<item>
  <id>13</id>
  <operazione>0</operazione>
  <tipo>3</tipo>
  <descrizione>quota fissa inchiostro colorato</descrizione>
  <ordine>3</ordine>
  <perc_commissione></perc_commissione>
  <fattura_coupon_id></fattura_coupon_id>
  <oggetto_id>314</oggetto_id>
  <data>2017-11-12 20:03:00</data>
  <iva>0.22</iva>
  <importo>0.35</importo>
  <totale_imponibile>0.29</totale_imponibile>
  <totale_iva>0.06</totale_iva>
</item>
<item>
  <id>19</id>
  <operazione>0</operazione>
  <tipo>1</tipo>
  <descrizione>spese di trasporto e distribuzione</descrizione>
  <ordine>4</ordine>
  <perc_commissione>0.04750</perc_commissione>
  <fattura_coupon_id></fattura_coupon_id>
  <oggetto_id>259</oggetto_id>
  <data>2017-11-12 20:30:12</data>
  <iva>0.22</iva>
  <importo>7.84</importo>
  <totale_imponibile>6.43</totale_imponibile>
  <totale_iva>1.41</totale_iva>
</item>
<item>
  <id>20</id>
  <operazione>0</operazione>
  <tipo>2</tipo>
  <descrizione>maggiorazioni carta riciclata</descrizione>
  <ordine>5</ordine>
  <perc_commissione>0.03400</perc_commissione>
  <fattura_coupon_id></fattura_coupon_id>

```

```

<oggetto_id>259</oggetto_id>
<data>2017-11-12 20:30:12</data>
<iva>0.22</iva>
<importo>5.61</importo>
<totale_imponibile>4.60</totale_imponibile>
<totale_iva>1.01</totale_iva>
</item>
<item>
<id>21</id>
<operazione>0</operazione>
<tipo>3</tipo>
<descrizione>smaltimento colle di scarto</descrizione>
<ordine>6</ordine>
<perc_commissione></perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>259</oggetto_id>
<data>2017-11-12 20:30:12</data>
<iva>0.22</iva>
<importo>0.35</importo>
<totale_imponibile>0.29</totale_imponibile>
<totale_iva>0.06</totale_iva>
</item>
<item>
<id>31</id>
<operazione>0</operazione>
<tipo>1</tipo>
<descrizione>costo servizio per vendita</descrizione>
<ordine>7</ordine>
<perc_commissione>0.02500</perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>313</oggetto_id>
<data>2017-11-13 13:55:01</data>
<iva>0.22</iva>
<importo>6.75</importo>
<totale_imponibile>5.53</totale_imponibile>
<totale_iva>1.22</totale_iva>
</item>
<item>
<id>32</id>
<operazione>0</operazione>
<tipo>2</tipo>
<descrizione>costi servizio macchine rotative</descrizione>
<ordine>8</ordine>
<perc_commissione>0.03400</perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>313</oggetto_id>
<data>2017-11-13 13:55:01</data>
<iva>0.22</iva>
<importo>9.18</importo>
<totale_imponibile>7.52</totale_imponibile>
<totale_iva>1.66</totale_iva>
</item>
<item>
<id>33</id>
<operazione>0</operazione>
<tipo>3</tipo>
<descrizione>quota abbonamenti</descrizione>
<ordine>9</ordine>
<perc_commissione></perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>313</oggetto_id>
<data>2017-11-13 13:55:01</data>
<iva>0.22</iva>
<importo>0.35</importo>
<totale_imponibile>0.29</totale_imponibile>
<totale_iva>0.06</totale_iva>
</item>
<item>
<id>35</id>
<operazione>0</operazione>
<tipo>1</tipo>
<descrizione>costo servizio per vendita</descrizione>
<ordine>10</ordine>
<perc_commissione>0.06000</perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>267</oggetto_id>
<data>2017-11-13 13:55:24</data>
<iva>0.22</iva>
<importo>2.10</importo>
<totale_imponibile>1.72</totale_imponibile>
<totale_iva>0.38</totale_iva>
</item>
<item>
<id>36</id>
<operazione>0</operazione>
<tipo>2</tipo>
<descrizione>maggiorazioni varie</descrizione>
<ordine>11</ordine>
<perc_commissione>0.03400</perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>267</oggetto_id>
<data>2017-11-13 13:55:24</data>
<iva>0.22</iva>
<importo>1.19</importo>
<totale_imponibile>0.98</totale_imponibile>
<totale_iva>0.21</totale_iva>
</item>
<item>
<id>37</id>
<operazione>0</operazione>
<tipo>3</tipo>
<descrizione>carta carburante</descrizione>
<ordine>12</ordine>
<perc_commissione></perc_commissione>
<fattura_coupon_id></fattura_coupon_id>
<oggetto_id>267</oggetto_id>
<data>2017-11-13 13:55:24</data>
<iva>0.22</iva>
<importo>0.35</importo>

```

```

    <totale_imponibile>0.29</totale_imponibile>
    <totale_iva>0.06</totale_iva>
  </item>
  <item>
    <id>39</id>
    <operazione>0</operazione>
    <tipo>1</tipo>
    <descrizione>noleggio trasporto</descrizione>
    <ordine>13</ordine>
    <perc_commissione>0.06000</perc_commissione>
    <fattura_coupon_id></fattura_coupon_id>
    <oggetto_id>36</oggetto_id>
    <data>2017-11-13 20:11:42</data>
    <iva>0.22</iva>
    <importo>7.50</importo>
    <totale_imponibile>6.15</totale_imponibile>
    <totale_iva>1.35</totale_iva>
  </item>
  <item>
    <id>40</id>
    <operazione>0</operazione>
    <tipo>2</tipo>
    <descrizione>extra buffet</descrizione>
    <ordine>14</ordine>
    <perc_commissione>0.03400</perc_commissione>
    <fattura_coupon_id></fattura_coupon_id>
    <oggetto_id>36</oggetto_id>
    <data>2017-11-13 20:11:42</data>
    <iva>0.22</iva>
    <importo>4.25</importo>
    <totale_imponibile>3.48</totale_imponibile>
    <totale_iva>0.77</totale_iva>
  </item>
  <item>
    <id>41</id>
    <operazione>0</operazione>
    <tipo>3</tipo>
    <descrizione>pubblicazioni di servizio</descrizione>
    <ordine>15</ordine>
    <perc_commissione></perc_commissione>
    <fattura_coupon_id></fattura_coupon_id>
    <oggetto_id>36</oggetto_id>
    <data>2017-11-13 20:11:42</data>
    <iva>0.22</iva>
    <importo>0.35</importo>
    <totale_imponibile>0.29</totale_imponibile>
    <totale_iva>0.06</totale_iva>
  </item>
  <item>
    <id>51</id>
    <operazione>0</operazione>
    <tipo>1</tipo>
    <descrizione>costi per rappresentanza</descrizione>
    <ordine>16</ordine>
    <perc_commissione>0.06000</perc_commissione>
    <fattura_coupon_id></fattura_coupon_id>
    <oggetto_id>28</oggetto_id>
    <data>2017-11-20 18:39:15</data>
    <iva>0.22</iva>
    <importo>2.97</importo>
    <totale_imponibile>2.43</totale_imponibile>
    <totale_iva>0.54</totale_iva>
  </item>
  <item>
    <id>52</id>
    <operazione>0</operazione>
    <tipo>2</tipo>
    <descrizione>costi per servizi interbancari</descrizione>
    <ordine>17</ordine>
    <perc_commissione>0.03400</perc_commissione>
    <fattura_coupon_id></fattura_coupon_id>
    <oggetto_id>28</oggetto_id>
    <data>2017-11-20 18:39:15</data>
    <iva>0.22</iva>
    <importo>1.68</importo>
    <totale_imponibile>1.38</totale_imponibile>
    <totale_iva>0.30</totale_iva>
  </item>
</response>

```

DATO IL SEGUENTE SPRIX DI LETTURA E STAMPA

```
PERCORSO$ = "jes_export.xml" ' documento Xml presente nella cartella datitutente dell'azienda in uso
```

```
XMLINIT
```

```
XMLDOC = XMLLOAD(PERCORSO$) 'apertura del file
```

```
IF XMLDOC = 0 OR _ERRXML$<" THEN VIMSG 5, "Impossibile aprire il file " + _ERRXML$: GOSUB ERR
```

```
ROOTNODE = XMLGETROOT(XMLDOC) 'tracciamento del nodo principale
```

```
IF ROOTNODE = 0 THEN VIMSG 5, "Impossibile trovare il nodo radice" + _ERRXML$: GOSUB ERR
```

```
DOCUM = XMLFINDELNS( ROOTNODE, "documenti", "", "", 2 )
```

```
IF DOCUM = 0 THEN VIMSG 5, "Impossibile trovare il nodo <documenti>" + _ERRXML$: GOSUB ERR
```

```
ITEM = XMLFINDELNS( DOCUM, "item", "", "", 2 )
```

```
LEGGI_TESTA:
```

```
IF ITEM = 0 THEN GOTO END_LEGGI_TESTA
```

```
'----- Lettura dati documento
```

```
VALORE1$=""
```

```
NODE = XMLFINDELNS( ITEM, "numero", "", "", 2 )
```

```

IFF NODE <> 0
  VALORE1$ = XMLGETNCONT$( XMLDOC, NODE )
ENDF

VALORE2$=""
NODE = XMLFINDELNS( ITEM, "data_emissione", "", "", 2 )
IFF NODE <> 0
  VALORE2$ = XMLGETNCONT$( XMLDOC, NODE )
ENDF

'----- Lettura cliente

CLIENTE = XMLFINDELNS(ITEM,"utente", "", "", 2 )
IFF CLIENTE <> 0
  VALORE3$=""
  NODE = XMLFINDELNS( CLIENTE, "cognome", "", "", 2 )
  IFF NODE <> 0
    VALORE3$ = XMLGETNCONT$( XMLDOC, NODE )
  ENDF

  VALORE4$=""
  NODE = XMLFINDELNS( CLIENTE, "codice_fiscale", "", "", 2 )
  IFF NODE <> 0
    VALORE4$ = XMLGETNCONT$( XMLDOC, NODE )
  ENDF
ENDF

' stampa testata documento

N TESTA|~VALORE1$\\|~VALORE2$\\|~VALORE3$\\|~VALORE4$\\

'----- Lettura articolo

ITEMCHILD= XMLGETCHILD(ITEM)
IF ITEMCHILD = 0 THEN GOTO NEXTSIBL

RIGHE = XMLFINDELNS(ITEMCHILD,"righe", "", "", 2 )
IF RIGHE = 0 OR _ERRXML$ <> "" THEN GOTO NEXTSIBL
CODART = XMLFINDELNS(RIGHE,"item", "", "", 2 )

NEXT_ART:
IF CODART = 0 OR _ERRXML$ <> "" THEN GOTO NEXTSIBL

VALORE5$=""
NODE = XMLFINDELNS( CODART, "id", "", "", 2 )
IFF NODE <> 0
  VALORE5$ = XMLGETNCONT$( XMLDOC, NODE )
ENDF

VALORE6$=""
NODE = XMLFINDELNS( CODART, "descrizione", "", "", 2 )
IFF NODE <> 0
  VALORE6$ = XMLGETNCONT$( XMLDOC, NODE )
ENDF

' stampa righe documento

N      RIGHE|~VALORE5$\\|~VALORE6$\\

SALTA:
CODART = XMLGETSIBL(CODART)
IF CODART <> 0 THEN TIPO = XMLGETELTYPE(CODART): IF TIPO <> 1 THEN GOTO SALTA 'GET SOLO TIPO=1 (nodi ELEMENTO), SALTA GLI ALTRI
GOTO NEXT_ART

NEXTSIBL:
ITEM = XMLGETSIBL(ITEM)
IF ITEM <> 0 THEN TIPO = XMLGETELTYPE(ITEM) : IF TIPO <> 1 THEN GOTO NEXTSIBL 'GET SOLO TIPO=1 (nodi ELEMENTO), SALTA GLI ALTRI
GOTO LEGGI_TESTA
END_LEGGI_TESTA:
XMLCLEANUP

END

ERR:
XMLCLEANUP
END
RETURN

VERIFICA:
VALORE$ = ""
NODE = XMLFINDELEM(RIFNODO,NOMENODO$)
IFF XMLGETCHILD(NODE)<> 0
  VALORE$ = XMLGETATVAL$(NODE)
ENDF
RETURN

RISULTATO

TESTA|1      |2017-10-03      |Calopresti      |PPOCLP90B12Q227A
TESTA|2      |2017-10-03      |Giacobazzi      |GPEGBZ55U04Z944Q
TESTA|3      |2017-10-03      |Silvani         |SGASVI45K33U276Z
RIGHE|1      |costi vari di servizio aggiuntivo
RIGHE|6      |costi vari di servizio aggiuntivo
TESTA|4      |2017-10-03      |Imperatore      |
RIGHE|2      |costi trasporto speciale
RIGHE|3      |maggiorazione spese servizi segreti
RIGHE|4      |quota videosorveglianza e guardianaggio
TESTA|5      |2017-10-03      |                |GRDFTE45M77A944G
RIGHE|7      |costi per magie e incantesimi
RIGHE|8      |maggiorazione per riparazione danni
RIGHE|9      |quota fissa aggiornamento formule e pozi
TESTA|6      |2017-11-30      |                |LPOABT76M45U944Z
RIGHE|11     |costo per pubblicazione
RIGHE|12     |supplemento edizioni speciali
RIGHE|13     |quota fissa inchiostro colorato

```


WEB SERVICE

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 47	Azzerare variabili struttura _WS	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "WEBSVC"	Azzerare variabili struttura _WS	S/C/Cr/Cd Sm/Cm
CALLWEBSVC	Istruzione per chiamate Web Service	S/C/Cr
HTTPHDRTAG\$()	Funzione per gestire header HTTP maggiori di 255 caratteri.	S/C/Cr/Cd
_WS	Variabili struttura chiamate Web Service	S/C
esempi e casi		
<p>Esempio: collegamento ai servizi WEB SERVICE del Ministero delle Politiche Agricole Alimentari e Forestali (MIPAAF), relativamente al registro delle Sostanze Zuccherine. Tale esempio verifica l'esito di trasmissione al WS in base ad un ID che viene passato come parametro.</p> <pre> ` Se scelgo di usare i file crea un file XML con le righe del servizio da richiedere al WS: GOSUB SCRIVI_FILE ` altrimenti riempie l'array _WSIBODY\$() con le medesime righe... GOSUB PREPARA_ARRAY `Prepara la comunicazione con il WS... _WSURL\$="cooptest.sian.it/wsTOAST/services/wsRegVino" `URL ` _WSQRYSTR\$ ` query string (PARAMETRI del servizio, cioè la stringa separata dall'URL ` dal carattere '?' ` costituita da serie di nome=valore separati da '&' ` Es: www.google.it/?gfe_rd=cr&ei=-CE7V_WpCMYK0zeWk-AJ ` _WSPROTOCOL\$="https" ` `PROTOCOLLO di comunicazione (valori possibili http, https) ` Nel caso di PROXI ` _WSPRXHST\$ `nome Host (o indirizzo) del Proxy ` _WSPRXPRT `porta del proxy ` _WSPRXUSRFW\$ `dati per l'autenticazione sul proxy (user:password) ` _WSPRXAUTH `tipo autenticazione (0 = non spec., 1 = Any, 2 = Basic, 3 = NTLM, ` `4 = Negoziazione, 5 = Digest) ` _WSPRXTUNN `tunneling (0 = Non Abilitato, <> 0 = Abilitato) ` Per AUTENTICAZIONE verso http che lo richiede ` _WSAUTH\$ `tipo Autenticazione. Valori ammessi : "basic", "bearer", "utente" ` _WSUSER\$ `utente ` _WSPASS\$ `Password ` _WSTKN\$() `token ` Per CERTIFICATO SSL comunicazione criptata ` _WSCERTFILE\$ `eventuale percorso/nome_file del certificato OpenSSL del client ` _WSCERTPASS\$ `eventuale Passphrase richiesta per l'uso del certificato ` _WSKEYFILE\$ `eventuale percorso/nome_file del file contenente la chiave privata ` _WSCAFILE\$ `file contenente le autorità di certificazione ` _WSVERIFICA `se <> 0 richiede la verifica del peer e cessa la comunicazione ` _WSUSRAGENT\$ `testo da comporre liberamente il cui contenuto viene passato come UserAgent ` _WSMETHOD\$="POST" ` Metodo di comunicazione HTTP, GET PUT POST FORMMULTIPART ` FORMMULTIPARTTOM ` _WSSOAPACT\$="" ` SOAPAction, la definisce il WSDL del servizio, nel caso del ` MIPAAF va passata così' "" che e' diverso dal non assegnarle ` nessun valore e nel qual caso questo servizio restituisce un ` errore ` _WSHEADERS(1) = "Accept-Encoding: gzip,deflate" ` Header protocollo HTTP. Sspecifica le ` tipologie di compressioni accettate ` dal browser ` _WSICNTTYP\$ = "text/xml; charset=utf-8" ` Valore dell'HEADER. Specifica il ` Content-Type (tipo) del documento di ` invio ` _WSCOOKIES `eventuali cookie da passare al server ` _WSIBDYFNAM\$ = REQ_ON_FILE\$ ` File con il corpo XML per il servizio. Se non si ` utilizza il file valorizzare l'array _WSIBODY\$() ` la struttura del file xml deriva sempre dalla ` definizione wsdl del servizio, i valori dei dati invece ` vanno inseriti (convertiti nel charset indicato in ` _WSICNTTYP\$ in questo caso UTF-8) ` _WSOFNAME\$ = "]ws_GetOperSiRPV_response.xml" ` File in cui si chiede di depositare la ` RISPOSTA del servizio. Se non si ` utilizza il file la risposta verra' ` depositata nell'array _WSOHEADERS(), e ` _WSOBODY\$() `Se si e' scelto di usare l'ARRAY di righe, vengono resettati i riferimenti ai files IF USE_IO_FILES = 0 THEN _WSIBDYFNAM\$ = "" : _WSOFNAME\$ = "" ` _WSLOGNAME\$ = "SPRIX_HTTPWS.log" ` Eventuale file di log della comunicazione ` (creato nella cartella mexal prevista per i log) ` _WSLOGVERB = 1 ` "Verbosita' del Log: ` 0 = Nessun Log ` 1 = Log ASCII ` 2 = Log ASCII + Dump esadecimale dei byte ` inviati ricevuti APRIFIN TIT 1,1,0, 4, 4, 5, 40, "Invocazione WebService", "" SETTF "" LOCATE 2,2 : PRINT "Connessione al servizio Web..." CALLWEBSVC `Comunicazione con il WS. </pre>		

```

LOCATE 2,2 : PRINT "Richiesta terminata."
SLEEP 1000
CHIUDIFIN 1

IF _WSRAWERR$ <> "" THEN VIMSG 5,_WSRAWERR$: END
` _WSRAWERR$ contiene eventuali errori
` non originati dalla comunicazione
` Es: parametri errati, errori
` caricamento libreria,...

` Altre Variabili per la restituzione dei dati:

` Risposta http in caso di chiamata NON andata a buon fine
` _WSORESP `Codice del Response http ricevuto dal server
` _WSORESPD$ `Descrizione testuale del Response http ricevuto dal server
` _WSOCNTYPE$ `Restituisce il Content-Type del body del response inviato dal server
` `Specifica il tipo del documento di risposta

PROCEDURE DI SCRITTURA ARRAY/FILE:

` Scrive, nell'array _WSIBODY$() la richiesta al servizio da utilizzare,
` completandolo con i dati richiesti come parametri. (Alternativo alla creazione del file)
PREPARA ARRAY:
` _WSIBODY$(1)="<soapenv:Envelope xmlns:soapenv="+CHR$(34)+"http://schemas.xmlsoap.org/soap/envelope/"+CHR$(34)
` _WSIBODY$(2)=" xmlns:soap="+CHR$(34)+"http://cooperazione.sian.it/schema/SoapAutenticazione"+CHR$(34)+"
xmlns:wsm="+CHR$(34)+"http://cooperazione.sian.it/schema/wsmrga/"+CHR$(34)+">" +CHR$(13)+CHR$(10)
` _WSIBODY$(3)=" <soapenv:Header>" +CHR$(13)+CHR$(10)
` _WSIBODY$(4)=" <soap:SOAPAutenticazione>" +CHR$(13)+CHR$(10)
` _WSIBODY$(5)=" <username>" +USER$ + "</username>" +CHR$(13)+CHR$(10) 'Parametro
` _WSIBODY$(6)=" <password>" +PASS$ + "</password>" +CHR$(13)+CHR$(10) 'Parametro
` _WSIBODY$(7)=" <nomeServizio>" +SERV$ + "</nomeServizio>" +CHR$(13)+CHR$(10) 'Parametro
` _WSIBODY$(8)=" </soap:SOAPAutenticazione>" +CHR$(13)+CHR$(10)
` _WSIBODY$(9)=" </soapenv:Header>" +CHR$(13)+CHR$(10)
` _WSIBODY$(10)=" <soapenv:Body>" +CHR$(13)+CHR$(10)
` _WSIBODY$(11)=" <wsm:GetOperSiRPVInput>" +CHR$(13)+CHR$(10)
` _WSIBODY$(12)=" <wsm:IdTrasmissione>" +IDTRASMISS$ + "</wsm:IdTrasmissione>" +CHR$(13)+CHR$(10)
` _WSIBODY$(13)=" </wsm:GetOperSiRPVInput>" +CHR$(13)+CHR$(10)
` _WSIBODY$(14)=" </soapenv:Body>" +CHR$(13)+CHR$(10)
` _WSIBODY$(15)=" </soapenv:Envelope>" +CHR$(13)+CHR$(10)
RETURN

` Scrive il file con la richiesta al servizio, completandolo con i dati richiesti come parametri. (Alternativo alla creazione dell'array)
SCRIVI FILE:
IF ESISFILE( REQ_ON_FILE$ ) THEN KILL REQ_ON_FILE$
NUMFILE=AUTOFILE()
OPEN "R", NUMFILE, REQ_ON_FILE$, 1
FIELD NUMFILE, 1 AS CH$
CURRREC=1
WRITEBUFF$="<soapenv:Envelope xmlns:soapenv="+CHR$(34)+"http://schemas.xmlsoap.org/soap/envelope/"+CHR$(34)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" xmlns:soap="+CHR$(34)+"http://cooperazione.sian.it/schema/SoapAutenticazione"+CHR$(34)+"
xmlns:wsm="+CHR$(34)+"http://cooperazione.sian.it/schema/wsmrga/"+CHR$(34)+">" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" <soapenv:Header>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" <soap:SOAPAutenticazione>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" <username>"
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=USER$
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="</username>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" <password>"
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=PASS$
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="</password>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" <nomeServizio>"
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=SERV$
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="</nomeServizio>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" </soap:SOAPAutenticazione>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" </soapenv:Header>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" <soapenv:Body>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" <wsm:GetOperSiRPVInput>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" <wsm:IdTrasmissione>"
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=IDTRASMISS$
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="</wsm:IdTrasmissione>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" </wsm:GetOperSiRPVInput>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" </soapenv:Body>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$=" </soapenv:Envelope>" +CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
CLOSE NUMFILE
RETURN

` _WSURL$="ws.arcadia3000.net/listaXYZ.xml"
` _WSQRYSTR$=""
` _WSPROTOCOL$="http"
` _WSLOGNAME$ = "SPRIX_HTTPWS.log"
` _WSLOGVERB = 1

` autenticazione "digest"
` WSAUTENT$="digest"
` WSUSER$="nome utente"

```

Nota: teoricamente _WSUSER\$ deve essere = "nome_utente", ma se si riscontrasse un errore di libreria (libcurl v. 7.40 utilizzata per i contatti http/HTTPS ed FTP) occorre indicare anche l'informazione relativa "realm" (che è un valore ritornato dal Server e che il Client potrebbe non replicare in un eventuale nuovo contatto al Server); occorre pertanto fare in modo di poter conoscere questo dato "realm" dai log di dialogo Client-Server, pertanto, la forma eventualmente da utilizzare è quella che lo indica direttamente:

```
_WSUSER$="valore_realm/nome_utente"

_WSPASS$="password"
  \ comunicazione
  \ WSMETHOD$="GET"
  \ chiamata
  \ CALLWEBSVC
IF _WSRAWERR$ <> "" THEN VIMSG 5,_WSRAWERR$: END
```

```
CHIUDIFIN 1
\PROTOCOLLO
  \WSPROTOCOL$="https" \ valori possibili http, https

\URL - Test digest authentication - gli ultimi 2 componenti di questo URL
\ devono corrispondere rispettivamente ad utente ( _WSUSER$ ) e password ( _WSPASS$ )
  \_WSURL$="httpbin.org/digest-auth/auth/adamo/passwd"

\Metodo HTTP
  \WSMETHOD$ ="GET" \ metodi HTTP, GET PUT POST FORMMULTIPART FORMMULTIPARTMTOM
  \WSAUTENT$ = "digest" \ Tipo Autenticazione: Basic, Digest, Bearer, User
  \_WSUSER$="adamo"
  \_WSPASS$="passwd"

_WSOFNAMES$ = "<C:\Temp\wsspx\testDigest.json" \ file in cui si chiede di depositare la risposta del servizio
KILL _WSOFNAMES$

_WSLOGNAME$ = "SPRIX_HTTPWS.log" \ Eventuale file di log ( creato nella cartella prevista per i log )
_WSLOGVERB = 1 \ "Verbosita' del Log:
  \ 0 = Nessun Log
  \ 1 = Log ASCII
  \ 2 = Log ASCII + Dump esadecimale dei byte inviati ricevuti

\ ***** VISUALIZZAZIONE AVANZAMENTO/RISULTATO *****

APRIFIN TIT 1,1,0, 4, 4, 5, 40, "Invocazione Webservice", ""
SETTF ""
LOCATE 2,2 : PRINT "Connessione al servizio Web..."
CALLWEBSVC
LOCATE 2,2 : PRINT "Richiesta terminata. "

SLEEP 1000
CHIUDIFIN 1
PRINT "_WSOCNTYPE$:"+_WSOCNTYPE$

IF _WSRAWERR$ <> "" THEN VIMSG 5,_WSRAWERR$:END
  \ _WSRAWERR$ contiene eventuali errori non originati dalla comunicazione
  \ Es: parametri errati, errori caricamento libreria,...

GOSUB PRINT_HEADERS
IF _WSOFNAMES$ = "" THEN GOSUB PRINT_BODY_ARRAY ELSE GOSUB PRINT_BODY_FILE
CHIUDIFIN 2
CHIUDIFIN 1
END

\=====
\ STAMPA DEI RISULTATI
\=====
PRINT_HEADERS:
\=====
\ STAMPA Degli HEADERS ( alcuni SI devono analizzare poiche' istruiscono su come trattare i dati ricevuti,
\ Es : Content-type che informa circa il mediaType e la codifica dei dati ricevuti, quindi che "forma"
\ hanno i dati xml piuttosto che Json piuttosto che testo,...
\=====

APRIFIN TIT 1,1,0, 2, 2, 15, 70, "Header HTTP", ""
SETTF "505-Chiudi"
I=1
LASTFIN=1
PLNCHR=68
PLNRW=2
PLNCOL=2
GOSUB PRINTLINE
NXT_HEADER:
IF _WSOHEADER$(I) = "" THEN GOTO PRINT_HEADERS_END
PLIN$= _WSOHEADER$(I)
GOSUB PRINTLINE
I=I+1
GOTO NXT_HEADER
PRINT_HEADERS_END:

RETURN

PRINT_BODY_ARRAY:
\=====
\ STAMPA del BODY ( risposta vera e propria del servizio )
\ la sua struttura sempre e' imposta dalla definizione WSDL del servizio
\=====

APRIFIN TIT 2,1,0, 1, 4, 18, 78, "Body HTTP", ""
SETTF "505-Chiudi"
LASTFIN=2
I=1
PLNRW=2
PLNCOL=2
NXT_BODYLINE:
IF _WSOBODY$(I) = "" THEN GOTO PRINT_BODY_ARRAY_END
PLIN$= _WSOBODY$(I)
PLNCHR=68
GOSUB PRINTLINE
I=I+1
GOTO NXT_BODYLINE
PRINT_BODY_ARRAY_END:
GOSUB WAITESC
RETURN
```

```

PRINT_BODY_FILE:
=====
`
`   STAMPA del BODY ( risposta vera e propria del servizio )
`   la sua struttura sempre e' imposta dalla definizione WSDL del servizio
=====
APRIFIN TIT 2,1,0, 1, 4, 18, 78, "Body HTTP", ""
SETTF "505-Chiudi"
LASTFIN=2
OPEN "R", 1, WSOFNAME$, 1
FIELD 1, 1 AS CH$
LF=LOF(1)
NPR=1
PLNCHR=78
FL=0
LOCATE 2,2
FOR B=1 TO LF
  GET 1,B
  IF NPR >= PLNCHR THEN PRINT: NPR=1
  IF CH$ = "/" THEN IF FL=0 THEN FL=3
  IF FL=1 THEN IF CH$="/" THEN FL=2 ELSE FL=0
  IF CH$ = "<" THEN IF FL=0 THEN FL=1 ELSE FL = 0
  IF CH$ = ">" AND ( FL=2 OR FL = 3 ) THEN PRINT CH$:NPR=1:FL=0 ELSE NPR=NPR+1 : PRINT CH$;
NEXT
CLOSE 1
PRINT_BODY_FILE_END:
GOSUB WAITESC
RETURN

=====
`
`   STAMPA Una riga di testo
=====
PRINTLINE:
LN=LEN(PLIN$)
PLTMP$=PLIN$
IF RIGHT$(PLTMP$, 1 ) = CHR$(10) THEN LX=LEN(PLTMP$):PLTMP$=LEFT$(PLTMP$,LX-1)
IF RIGHT$(PLTMP$, 1 ) = CHR$(13) THEN LX=LEN(PLTMP$):PLTMP$=LEFT$(PLTMP$,LX-1)
PRINT_CHUNK:
PSL=LEN(PLTMP$)
IF PSL <= PLNCHR THEN LOCATE PLNROW, PLNCOL : PRINT PLTMP$ : PLNROW = PLNROW + 1 : RETURN
PSS=LEFT$(PLTMP$, PLNCHR)
LOCATE PLNROW, PLNCOL
PRINT PSS;
PLNROW = PLNROW + 1
PLTMP$=RIGHT$(PLTMP$, PSL-PLNCHR)
GOTO PRINT_CHUNK
RETURN

=====
`
`   Attende la pressione di un tasto
=====
WAITESC:
_INTER=0
STARTWAIT:
A=INKEY()
IF A <> 505 THEN GOTO STARTWAIT
_INTER=1
RETURN

=====
`
`   Scrive il file con la richiesta al servizio, completandolo con i dati
`   digitati dall'utente
=====
SCRIVI_INPUT:
IF ESI$FILE( REQ_ON_FILE$ ) THEN KILL REQ_ON_FILE$
NUMFILE=AUTOFILE()
OPEN "R", NUMFILE, REQ_ON_FILE$, 1
FIELD NUMFILE, 1 AS CH$
CURRREC=1
WRITEBUFF$="<?xml version="+CHR$(34)+"1.0"+CHR$(34)+" encoding="+CHR$(34)+"utf-8"+CHR$(34)+"?>"+CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="<soap12:Envelope xmlns:xsi="+CHR$(34)+"http://www.w3.org/2001/XMLSchema-instance"+CHR$(34)+"
xmlns:xsd="+CHR$(34)+"http://www.w3.org/2001/XMLSchema"+CHR$(34)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="  xmlns:soap12="+CHR$(34)+"http://www.w3.org/2003/05/soap-envelope"+CHR$(34)+">"+CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="    <soap12:Body>"+CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="      <GetCountryByCountryCode xmlns="+CHR$(34)+"http://www.webserviceX.NET"+CHR$(34)+">"+CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="        <CountryCode>it</CountryCode>"+CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="      </GetCountryByCountryCode>"+CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="    </soap12:Body>"+CHR$(13)+CHR$(10)
GOSUB SUB_WRITE_BUFF
WRITEBUFF$="  </soap12:Envelope>"+CHR$(13)+CHR$(10)

CLOSE NUMFILE
RETURN

=====
`
`   Scrive ogni carattere del buffer di testo <WRITEBUFF$> nel file <NUMFILE>
`   nel record <CURRREC>
=====
SUB_WRITE_BUFF:
FOR ISL = 1 TO LEN( WRITEBUFF$ )
  CH$=MID$( WRITEBUFF$, ISL, 1 )
  PUT NUMFILE, CURRREC
  CURRREC = CURRREC + 1
NEXT
RETURN

PREPARA_RIGHE:
_WSIBODY$(1)="<?xml version="+CHR$(34)+"1.0"+CHR$(34)+" encoding="+CHR$(34)+"utf-8"+CHR$(34)+"?>"+CHR$(13)+CHR$(10)
_WSIBODY$(2)="  <soap12:Envelope xmlns:xsi="+CHR$(34)+"http://www.w3.org/2001/XMLSchema-instance"+CHR$(34)+"
xmlns:xsd="+CHR$(34)+"http://www.w3.org/2001/XMLSchema"+CHR$(34)

```



```

_WSICNTTYP$ = "multipart/form-data"      ' Valore dell'HEADER Content-Type
                                           ' del file xml deriva sempre dalla definizione wsdl del servizio,
                                           ' i valori dei dati invece vanno inseriti
_WSCERTFILES= "AUTO"      ' Utilizzo dei certificati SSL dell'installazione ( quelli residenti in ../dati/conf/ )

'*****
' Inizio Definizione delle diverse parti del corpo multipart
'*****
' Parte Nr:1
_WSMELTYPE$(1) = "TESTO"
_WSMCTYPES(1) = "application/xml"      ' DEVO INDICARLO POICHE' LA LIBRERIA NON SA CAPIRE IL TIPO DI UN TESTO
_WSMFILNAMES(1) = "precompilata.xml"  ' SI PUO' NON INDICARLO SE IN PARTICOLARE QUESTO DATO NON VA SCRITTO SU FILE
_WSMCPNAMES(1) = "precomp"           ' NOME DA ASSEGNARE ALLA PARTE INVECE DI QUELLO CREATO AUTOMATICAMENTE

_WSMPBUFF$(1, 1) = "<?xml version="+CHR$(34)+"1.0"+CHR$(34)+" encoding="+CHR$(34)+"UTF-8"+CHR$(34)+"?">"
_WSMPBUFF$(1, 2) = "<precompilata xsi:noNamespaceSchemaLocation="+CHR$(34)+"730_precompilata.xsd"+CHR$(34)+"
xmlns:xsi="+CHR$(34)+"http://www.w3.org/2001/XMLSchema-instance"+CHR$(34)+">"
_WSMPBUFF$(1, 3) = " <opzionale1>text</opzionale1>"
_WSMPBUFF$(1, 4) = " <opzionale2>text</opzionale2>"
_WSMPBUFF$(1, 5) = " <opzionale3>text</opzionale3>"
_WSMPBUFF$(1, 6) = " <proprietario>"
_WSMPBUFF$(1, 7) = " <cfProprietario>XysMHuS2AKv/csaOTn9AUetario>"
_WSMPBUFF$(1, 8) = " </proprietario>"
_WSMPBUFF$(1, 9) = " <documentoSpesa>"
_WSMPBUFF$(1,10) = " <idSpesa>"
_WSMPBUFF$(1,11) = " <pIva>0000000000</pIva>"
_WSMPBUFF$(1,12) = " <dataEmissione>2015-01-01</dataEmissione>"
_WSMPBUFF$(1,13) = " <numDocumentoFiscale>"
_WSMPBUFF$(1,14) = " <dispositivo>1</dispositivo>"
_WSMPBUFF$(1,15) = " <numDocumento>1</numDocumento>"
_WSMPBUFF$(1,16) = " </numDocumentoFiscale>"
_WSMPBUFF$(1,17) = " </idSpesa>"
_WSMPBUFF$(1,18) = " </documentoSpesa>"
_WSMPBUFF$(1,19) = "</precompilata>"
_WSMPBUFF$(1,20) = ""

'''''' _WSMHEADER$(1,1)="Content-Disposition: form-data; name="+CHR$(34)+"medico"+CHR$(34)+";
filename="+CHR$(34)+"ComunicazioneSpeseMediche.xml"+CHR$(34)      ' CREATO AUTOMATICAMENTE CON IL VALORE DI DEFAULT DI COPYNAME

' Parte Nr:2
_WSMELTYPE$(2)="FILE"
_WSMFNAMES(2)="]medico.zip"
_WSMCTYPES(2)="application/zip"      ' VIENE INDICATO DAL PROGRAMMATORE PER SOSTITUIRE QUELLO CHE LA LIBRERIA CREA AUTOMATICAMENTE
_WSMHEADER$(2,1)="Content-Transfer-Encoding: binary"
'''''' _WSMHEADER$(2,2)="Content-Disposition: form-data; name="+CHR$(34)+"archivio"+CHR$(34)+";
filename="+CHR$(34)+"Allegato.zip"+CHR$(34)      ' VERRA' CREATO AUTOMATICAMENTE

' Parte Nr:3
_WSMELTYPE$(3)="FILE"
_WSMFNAMES(3)="]logo_pass.gif"
'''''''' _WSMCTYPES(3)="image/gif"      ' ASSEGNATO AUTOMATICAMENTE
_WSMHEADER$(3,1)="Content-Transfer-Encoding: binary"
_WSMHEADER$(3,2)="Content-Disposition: form-data; name="+CHR$(34)+"PassLogo"+CHR$(34)+"; filename="+CHR$(34)+"PasseLogo.gif"+CHR$(34)
' OVERRIDE DI QUELLO CHE VERREBBE CREATO AUTOMATICAMENTE

'*****
' Fine Definizione delle diverse parti del corpo multipart
'*****

_WSLOGNAME$ = "SPRIX_HTTPWS.log"      ' Eventuale file di log ( creato nella cartella prevista per i log )
_WSLOGVERB = 1      ' 'Verbosita' del Log:

APRIFIN TIT 1,1,0, 4, 4, 5, 40, "Invocazione WebService", ""
SETTF ""
LOCATE 2,2 : PRINT "Connessione al servizio Web"
LOCATE 3,2 : PRINT "invio di "+STR$(J,0,0)+" RIGHE..."
CALLWEBSVC
LOCATE 2,2 : PRINT "Richiesta terminata.      "
SLEEP 1000
CHIUDIFIN 1

IF _WSRAWERR$ <> "" THEN VMMSG 5, _WSRAWERR$:END      ' _WSRAWERR$ contiene eventuali errori non originati dalla comunicazione
                                           ' Es: parametri errati, errori caricamento libreria,...

END

```

Esempio chiamata Web Service tramite JSON

```

'Il seguente singolo SPRIX interfaccia il Gestionale ad un Marketplace di abbigliamento tramite uno scambio di informazioni su file JSON
'al Server del Markeplace, via protocollo HTTPS WEB SERVICES.

'
'Lo SPRIX puo essere adattato anche a forme di SERVIZIO oppure automatismi COLLAGE. Questo SPRIX richiede ad un WEB SERVICES di allineare
'le giacenze articoli pubblicate nel Marketplace con quelle risultanti dai progressivi anagrafici del Gestionale e di ricevere gli ordini
'registrati nel Marketplace. Analogamente a quanto possibile tramite file XML, anche JSON si basa su file dal contenuto "strutturato" in
'elementi con determinate proprieta'. Questo scambio di informazioni e' tramite file JSON: nel primo caso con il metodo POST (simile a una
'PUT" nel WS Marketplace dei dati contenuti nel file JSON), nel secondo con il metodo GET dei dati contenuti nel file JSON rilasciato dal
'Marketplace. In ogni caso, queste CALL richiedono autenticazione con accesso tramite "token" (ripetibile, ma non in questo esercizio) ed
'anche in questo caso lo scambio di informazioni sul token di accesso avviene tramite file JSON.

'#####
' SCELTA OPERAZIONE #      (apro una finestra per selezionare il tipo di operazione di scrittura al Market o lettura dal Market)
'#####

SCELTA:
NF=AUTOFIN() : APRIFIN NF,1,0,5,10,10,60 : LOCATE 3,3
PRINT "Scrivi giacenze (S) - Leggi ordini (L) " : INPUT "A",P$,1 : CHIUDIFIN NF
IF P$="S" THEN GOSUB POST_ART      'con metodo POST scrivo JSON con la struttura dei dati degli articoli per come sono richieste del Market
IF P$="L" THEN GOSUB GET_ORD      'con metodo GET leggo il JSON con la struttura degli ordini per come viene fornita dal Web Service
END

'#####
' SCRITTURA ARTICOLI #
'#####

POST ART:

```



```
'prima richiedere e ottenere il Token per eseguire la scrittura, carico la struttura Anagrafica Articoli da scrivere con metodo WS POST
'e poi eseguo le call al servizio

DIM NUM_ID() : DIM ART$() : DIM CAT$() : DIM BARCODE$() : DIM GIAC_TEO() : DIM GIAC_EFF() : DIM PREZ() : DIM MAR$( ) : DIM COMP$( )

'del ciclotabellare seleziono solo articoli che ho preventivamente stabilito tramite condizioni, per esempio
'LEFT$(arnds(1)(),6)="MARKET" AND armra(1)()="S", una nota e/o flag che riattivo se ho carico/scarico merce e occorre aggiornare giacenze
'Dopo che e avvenuto il corretto post, il flag viene abbassato per non processare nuovamente gli articoli.

I=0
+-----+
|Cod.tab.:ar  MAGAZZINO - ANAGRAFICA ARTICOLO
|Scelte...:
|          LEFT$(arnds(1)(),6)="MARKET" AND armra(1)()="S"
+-----+
ar  I=I+1
ar  INV=0 : CAR=0 : SCA=0 : CLSX=0 : IMP=0 : ORF=0 : NDISP_TEO=0 : NDISP_EFF=0
ar  INV=aginv(arcod(),1,sxann):CAR=aqcar(arcod(),1,sxann):SCA=aqzca(arcod(),1,sxann)
ar  CLSX=aqcls(arcod(),1,sxann):IMP=aqcli(arcod(),1,sxann):ORF=aqorf(arcod(),1,sxann)
ar  NDISP_TEO=INV+CAR+SCA+CLSX+IMP : NDISP_EFF=INV+CAR+SCA+IMP : ART$(I)=arcod()
ar  BARCODE$(I)=aralt() : GIAC_TEO(I)=NDISP_TEO : GIAC_EFF(I)=NDISP_EFF
ar  PREZ(I)=arprz(1)(arcod())
ar  N ~arcod()\ - ~ardta()\ - ~armet()\ - attivo: ~ - DISP_TEO: ~NDISP_TEO \ - DISP_EFFETTIVA ~NDISP_EFF \ - PREZ ~arprz(1)(arcod())

'Potrebbe essere necessario stabilire un limite di articoli es (I<=50) : 'IF I>49 THEN VIMSG 1, "Troppi Articoli, massimo 50" : END

'INIZIALIZZAZIONE FORZATA DI VARIABILI (per l'assenza di dati, quindi supponendo che la struttura da inviare sia stata caricata
'con i seguenti dati arbitrari sul primo elemento di array della struttura articoli da scrivere sul file JSON:

K=I : NUM_ID(1)=99 : CAT$(1)="Categoria" : MAR$(1)="Marca" : COMP$(1)="Composizione" : CODART$="Cod.Articolo"

'poi creo la struttura di oggetti del JSON

'-----+
CREAZIONE_FJSON: 'creazione del file per il dump giacenze articoli
'-----+
ROOTOBJ = JSONCREATE() : IF ROOTOBJ = 0 THEN VIMSG 5, "Impossibile creare un nuovo OGGETTO JSON"+"|"+_ERRJJSON$: GOTO JSON_S_END
RIF_ARRAY_1 = JSONADDARR(ROOTOBJ,"ADDARRAY_1",0) 'Riferimento Array_1
IF RIF_ARRAY_1 = 0 THEN VIMSG 5, "Impossibile aggiungere una ARRAY JSON"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

'--OBJ1
RIFN1 = JSONCREATE() 'creazione di OBJ1 e memorizzazione del suo riferimento nella variabile RIFN1 (RIFERIMENTO NODO 1)
IF RIFN1 = 0 THEN VIMSG 5, "Impossibile creare un nuovo OGGETTO JSON"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT
RIOBJ1 = JSONADDOBJ(RIF_ARRAY_1,"ADDOBJ1",RIFN1) 'AddOBJ1 (Riferimento Oggetto 1)

RF = JSONADDNUM(RIOBJ1,"item_id",NUM_ID(1))
IF RF = 0 THEN VIMSG 5, "Impossibile aggiungere dato tipo NUMERO"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ1,"manufacturer code",CODART$(1))
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ1,"gender","")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT ' non obbligatorio

RF = JSONADDSTR(RIOBJ1,"category",CAT$(1))
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDNULL(RIOBJ1,"category_id")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo NULL"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT ' non obbligatorio

RF = JSONADDSTR(RIOBJ1,"brand",MAR$(1))
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDNULL(RIOBJ1,"brand_id")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo NULL"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT 'non obbligatorio

RF = JSONADDSTR(RIOBJ1,"composition",COMP$(1))
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ1,"composition_details","")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT 'non obbligatorio

RF = JSONADDNULL(RIOBJ1,"season_id")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo NULL"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ1,"made_in","")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT 'non obbligatorio

RIF_ARRAY_2 = JSONADDARR(RIOBJ1,"stocks",0) 'Riferimento Array_2
IF RIF_ARRAY_2 = 0 THEN VIMSG 5, "Impossibile aggiungere un ARRAY JSON"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

'---OBJ2
FOR J=1 TO K
RIFN2 = JSONCREATE() ' (RIFERIMENTO NODO 2)
IF RIFN2 = 0 THEN VIMSG 5, "Impossibile creare un nuovo OGGETTO JSON"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT
RIOBJ2 = JSONADDOBJ(RIF_ARRAY_2,"ADDOBJ2",RIFN2) 'AddOBJ2

RF = JSONADDSTR(RIOBJ2,"color_description","nome colore")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDNULL(RIOBJ2,"color_id")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo NULL"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ2,"manufacturer_color","")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ2,"size_description","")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ2,"barcode",BARCODE$(J))
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ2,"manufacturer barcode","")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT

RF = JSONADDNUM(RIOBJ2,"stock",GIAC_EFF(J))
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo NUMERO"+"|"+_ERRJJSON$: GOTO JSON_S_EXIT
```

```
RF = JSONADDREAL(RIOBJ2,"price",PREZ(J))
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo REAL"+"|"+_ERRJSON$: GOTO JSON_S_EXIT

RF = JSONADDSTR(RIOBJ2,"img_url","http..")
IF RF=0 THEN VIMSG 5, "Impossibile aggiungere dato tipo STRINGA"+"|"+_ERRJSON$: GOTO JSON_S_EXIT
```

NEXT

'creo il file JSON

JSONDUMP RIF_ARRAY_1,3,"]\json_post\dump_articoli.json" 'inserimento dell'ARRAY nella ROOT

'Il file JSON che viene generato nel percorso sopra indicato ha questa struttura (per come richiesta dalla POST del WEB SERVICES, ovvero per come indicato dallo sviluppatore di quel WS Marketplace). In questo caso l'oggetto JSON inizia con una ARRAY "[...]" e non come un elemento oggetto "{...}":

```
[
  {
    "item_id": 99,
    "manufacturer_code": "Cod.Articolo",
    "gender": "",
    "category": "Categoria",
    "category_id": null,
    "brand": "Marca",
    "brand_id": null,
    "composition": "Composizione",
    "composition_details": "",
    "season_id": null,
    "made_in": "",
    "stocks": [
      {
        "color_description": "nome colore",
        "color_id": null,
        "manufacturer_color": "",
        "size_description": "",
        "barcode": "BARCODE-001",
        "manufacturer_barcode": "",
        "stock": 999,
        "price": 1000.11,
        "img_url": "http.."
      },
      {
        "color_description": "nome colore",
        "color_id": null,
        "manufacturer_color": "",
        "size_description": "",
        "barcode": "BARCODE-002",
        "manufacturer_barcode": "",
        "stock": 3000,
        "price": 20.22,
        "img_url": "http.."
      },
      {
        "color_description": "nome colore",
        "color_id": null,
        "manufacturer_color": "",
        "size_description": "",
        "barcode": "BARCODE-003",
        "manufacturer_barcode": "",
        "stock": 2000,
        "price": 333.33,
        "img_url": "http.."
      }
    ]
  }
]
```


' AZZERO FLAG STATO ANAGRAFICHE (perche' processati e inviati)
' Il flag deve essere "alzato" per gli articoli che vengono caricati o importati
' Dopo che e avvenuto il corretto post (qui non si controllano errori),
' il flag viene abbassato per NON processare nuovamente gli articoli.

```
FOR X=1 TO K
  GETAR ART$(X),1
  IF _ERRAR$<>" THEN VIMSG 1, _ERRAR$
  _ARMRA$(1)="N"
  PUTAR 1
  IF _ERRAR$<>" THEN VIMSG 1, _ERRAR$
NEXT
```


'dopo questo carico di dati nella struttura nel JSON, richiedo al Servizio l'apposito Token
GOSUB CALL_TKN_WS_HTTP 'consultare l'etichetta CALL_TKN_WS_HTTP:
'dopo la RETURN, se ricevo il Token in _WSOFNAME\$<>" allora proseguo con la prossima etichetta OK_TKN_POST:,
'altrimenti esco da questa CALL con errore (ma non in questo esercizio)

OK_TKN_POST:

```
AZZVARSYS 47
_WSURL$ = "https"
_WSURL$ = "//cloud.marketplace.com/api/supplier-product"
_WSAUTENT$ = "bearer" '(commentato perchè già presente in _WSHEADER$(3), può essere richiesta una forma oppure l'altra)
_WSUSER$ = REALM$+"/"+login"
_WSPASS$ = "password"
_WSTKN$(1-N) = "token"
_WSMETHOD$ = "POST"
_WSHEADER$(1) = "cache-control: no-cache"
_WSHEADER$(2) = "Content-Type: application/json"
_WSHEADER$(3) = "Authorization: Bearer " + TOKEN$ ' rilasciato da CALL_TKN_WS_HTTP
_WSICNTTYP$ = "application/json"
_WSIBDYFNAM$ = "]json\dump_articoli.json" 'file giacenze
_WSOFNAME$ = "]json\resp_dump.json" ' file risposta
_KILL _WSOFNAME$
_WSLOGNAME$ = "sprix_https.log"
_WSLOGVERB = 1

CALLWEBSVC : GOSUB TEST_HTTP
GOTO JSON_S_EXIT
```

```

#####
' LETTURA ORDINI #
#####
' Oltre al metodo GET su JSON, su puo' accedere al file JSON anche con ITERATORE (ma non e' previsto in questo esercizio)

GET_ORD: 'lettura degli ordini strutturati nel JSON del servizio WS Marketplace

GOSUB CALL_TKN_WS_HTTP 'consultare l'etichetta CALL_TKN_WS_HTTP:
' dopo la RETURN, se ricevo il Token in _WSOFNAME$<>" allora proseguo all'etichetta OK_TKN_POST,
' altrimenti esco da questa CALL con errore (ma non in questo esercizio)

OK_TKN_GET:
AZZVARSYS 47
  _WSPROTOCOL$ = "https"
  _WSURL$ = "//cloud.marketplace.com/api/get-orders"
  _WSQRYSTR$ = "from=${from}&to=${to}"
  ' _WSAUTENT$ = "bearer" '(commentato perchè già presente in _WSHEADER$(3), può essere richiesta una forma oppure l'altra)
  _WSUSER$ = "login" '(o con eventuale valore REALM)
  _WSPASS$ = "password"
  ' _WSTKN$(1-N) = "token....."
  _WSMETHOD$ = "GET"
  _WSHEADER$(1) = "cache-control: no-cache"
  _WSHEADER$(2) = "Content-Type: application/json"
  _WSHEADER$(3) = "Authorization: Bearer " + TOKEN$ ' rilasciato da CALL_TKN_WS_HTTP
  _WSICNTTYP$ = "application/json"
  _WSIBDYFNAM$ = "%\temp\...orders.json"
  _WSOFNAME$ = "]\json_get\test_get_orders.json"
' KILL _WSOFNAME$
  _WSLOGNAME$ = "sprix_https.log"
  _WSLOGVERB = 1

CALLWEBSVC : GOSUB TEST_HTTP

FILE$="]\json_get\get_orders.json" : IF NOT ESISFILE(FILE$) THEN VIMSG 5, "File JSON mancante, crearlo con put json e riprovare " : GOTO
JSON_L_END
RADICE_JSON=JSONLOAD(FILE$) : IF RADICE_JSON = 0 THEN VIMSG 5, _ERRJSON$+"|"+"Eseguire lo sprix put json e riprovare" : GOTO JSON_L_END
'alcuni controlli possibili
'IF _WSOFNAME$<>" ... IF (ESISFILE(_WSOFNAME$)) ... RADICE_JSON=JSONLOAD (_WSOFNAME$) ... IF (RADICE_JSON=0 OR _ERRJSON$<>"") THEN ...

TYPE=JSONTYPE(RADICE_JSON) ' la radice e' un object array
RIFOBJAR=JSONGETOBJ(RADICE_JSON,"orders",0)
IF RIFOBJAR=0 THEN VIMSG 5, "Impossibile ottenere il rif. a chiave orders"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
NUM_ELEM=JSONARRSIZE(RIFOBJAR) 'quanti elementi nell'object array

'DIMENSIONAMENTO ARRAY STRUTTURA_MM per (NUM_ELEM -1) per eseguire le PUTMM ORDINI
-----
DIM ID() : DIM DAT$() : DIM SKU$() : DIM TAG$() : DIM COL$() : DIM EAN$() : DIM QTA$() : DIM PZR$() : DIM STA$()

ERR$=""
FOR IND_ELEM=0 TO NUM_ELEM-1 'iterazione elementi (da ZERO a Numero Elementi -1)

RIF_EL=JSONGETARR(RIFOBJAR,IND_ELEM)
IF RIF_EL=0 THEN ERR$="Impossib. leggere elemento posiz. "+STR$(INDICE_ELEM,0,0)+"|"+_ERRJSON$:INDICE_ELEM=NUM_ELEMENTI : GOTO ARRAY_NEXT

' Lettura chiavi oggetto JSON

CHIAVE$ = "id" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF=0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
ID = JSONGETNUM(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo numerico"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'valorizzare array_MM???( ) di interesse ...

CHIAVE$ = "data" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF=0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
DAT$ = JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'valorizzare array_MM???( ) di interesse ...

CHIAVE$ = "sku" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF=0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
SKU$ = JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'valorizzare array_MM???( ) di interesse ...

CHIAVE$ = "taglia" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
TAG$ = JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'valorizzare array_MM???( ) di interesse ...

CHIAVE$ = "colore" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
COL$ = JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'valorizzare array_MM???( ) di interesse ...

CHIAVE$ = "barcode" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
EAN$ = JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'valorizzare array_MM???( ) di interesse ...

CHIAVE$ = "qty" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
QTA$ = JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'valorizzare array_MM???( ) di interesse ...

CHIAVE$ = "priceRetail" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
PZR$ = JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
'valorizzare array_MM???( ) di interesse ...

CHIAVE$ = "status" : RIF = JSONGETOBJ(RIF_EL,CHIAVE$,0)
IF RIF = 0 THEN VIMSG 5, "Impossibile ottenere il riferimento alla chiave:"+CHIAVE$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
STA$ = JSONGETSTR(RIF)
IF _ERRJSON$<>" THEN VIMSG 5, "Impossibile effettuare la lettura del dato di tipo stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT

```

```
'valorizzare array_MM???(?) di interesse ...

ARRAY_NEXT:
NEXT

IF ERR$<>" THEN VIMSG 5, ERR$ : GOTO JSON_L_EXIT

'-----
' FORMATTAZIONE GET FILE JSON ORDERS
'-----
' {
'   "orders": [
'     {
'       "id": 793351,
'       "data": "2020-08-14 06:45:25",
'       "sku": "3HYMAA YJ96Z",
'       "taglia": "0XS",
'       "colore": "002",
'       "barcode": "340001380808",
'       "qty": "1",
'       "priceRetail": "65.00",
'       "status": "confirmed"
'     },
'     {
'       "id": 793354,
'       "data": "2020-08-14 06:45:25",
'       "sku": "3H2T7K 2J53Z",
'       "taglia": "046",
'       "colore": "019",
'       "barcode": "340001389870",
'       "qty": "1",
'       "priceRetail": "145.00",
'       "status": "confirmed"
'     }
'   ]
' }

'nota: a seconda della struttura del file, la presenza del carattere virgola ","
'      all'ultimo elemento "}" dell'array "]", può provocare errore (togliere la virgola)

'-----
' CICLO SCRITTURA ORDERS PRESENTI SU FILE JSON
'-----
PUT_ORD:

' FOR .....
'   _MMSIG$="OC" : _MMSER=1 : _MMNUM=0 : _MMDAT$=sxdat : _MMMAG=1 : _MMCLI$="201.00049" : _MMIEV$="E"
'   FOR I=1 TO 3 : _MMTPR$(I)="R" : _MMART$(I)="000" : _MMQTA(I)=I : _MMORD$(I)="E" : _MMPRZ(I)=I+I : _MMALI$(I)="22" : NEXT
'   PUTMM 0 : IF _ERRMM$<>" THEN VIMSG 1, _ERRMM$
'   ..... NEXT

GOTO JSON_S_EXIT

'#####
' TOKEN BEARER #
'#####

'Il Servizio di autenticazione implementa il framework OAuth2.
CALL_TKN_WS_HTTP: 'richiesta JSON del Token al Servizio

'dopo la RETURN, se ricevo il Token in _WSOFNAME$<>" allora proseguo all'etichetta OK_TKN_POST,
'altrimenti esco da questa CALL con errore (ma non in questo esercizio)

AZZVARSYS 47
_WSPROTOCOL$ = "https"
_WSURL$ = "/cloud.marketplace.com/oauth/token"
_WSAUTENTS$ = "bearer"
_WSUSER$ = "username" ' (o con eventuale valore REALM)
_WSPASS$ = "password"
_WSMETHOD$ = "POST"
' _WSCONNTMOUT = 60 'Timeout connessione 60 sec (expires_in)
_WSHEADER$(1) = "cache-control: no-cache"
_WSHEADER$(2) = "content-type: application/json"
_WSICNTTYP$ = "application/json"
_WSIBDYFNAM$ = "]\\Json_token\\get_token.json" ' file con username
' per la creazione del token il WS richiede il metodo POST per la richiesta JSON e secondo
' la seguente struttura JSON della chiamata

' {
'   "grant_type": "password",
'   "client_id": "CLIENT-ID",
'   "client_secret": "CLIENT-SECRET",
'   "username": "USERNAME",
'   "password": "PASSWORD",
'   "scope": ""
' }

_WSOFNAME$ = "]\\Json_token\\ret_token.json" ' file risposta del servizio
_WSLOGNAME$ = "sprix_https.log" ' log comunicazione in cartella log del gestionale
_WSLOGVERB = 1
'----- visualizza avanzamento -----
APRIFIN TIT 1,1,0,4,4,5,40,"Invocazione Token WebService",""
SETTF ""
LOCATE 2,2 : PRINT "Connect per ricevere il Token ..."
CALLWEBSVC : GOSUB TEST_HTTP

'lettura del elemento relativo al Token ricevuto nel file JSON con la seguente struttura richiesta dal WS del Marketplace

' {
'   "token_type": "Bearer",
'   "expires_in": 31536000,
'   "access_token": "codicetoken",
'   "refresh_token": "..."
' }

LEGGI_TOKEN:

FILES=_WSOFNAME$
IF NOT ESISFILE(FILES) THEN VIMSG 5, "File ret_token.json mancante"
ROOTJSON=JSONLOAD(FILES) : IF ROOTJSON=0 THEN VIMSG 5, _ERRJSON$ : GOTO JSON_L_EXIT
```

```

TKN1$="token_type" : RIF=JSONGETOBJ(ROOTJSON,TKN1$,0) : IF RIF=0 THEN VIMSG 5, "Errore chiave:"+TKN1$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
TYPE$=JSONGETSTR(RIF) : IF _ERRJSON$<>" THEN VIMSG 5, "Errore lettura stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT

TKN2$="expires_in" : RIF=JSONGETOBJ(ROOTJSON,TKN2$,0) : IF RIF=0 THEN VIMSG 5, "Errore chiave:"+TKN2$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
EXP=JSONGETNUM(RIF) : IF _ERRJSON$<>" THEN VIMSG 5, "Errore lettura numero"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT

TKN3$="access_token" : RIF=JSONGETOBJ(ROOTJSON,TKN3$,0) : IF RIF=0 THEN VIMSG 5, "Errore chiave:"+TKN3$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
ACC$=JSONGETSTR(RIF) : IF _ERRJSON$<>" THEN VIMSG 5, "Errore lettura stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT

TKN4$="refresh_token" : RIF=JSONGETOBJ(ROOTJSON,TKN4$,0) : IF RIF=0 THEN VIMSG 5, "Errore chiave:"+TKN4$+"|"+_ERRJSON$ : GOTO JSON_L_EXIT
REF$=JSONGETSTR(RIF) : IF _ERRJSON$<>" THEN VIMSG 5, "Errore lettura stringa"+"|"+_ERRJSON$ : GOTO JSON_L_EXIT

TOKEN$=ACC$ 'token da passare nella chiamata
TIMEOUT=EXP

LOCATE 2,2 : PRINT "Richiesta terminata"
SLEEP 1000
CHIUDIFIN 1
IF _WSRAWERR$<>" THEN VIMSG 5, _WSRAWERR$ : END 'errori NON DI comunicazione (parametri errati, caricamento libreria,...)

'RESPONSE DEL SERVER HTTP (chiamata fallita)
VIMSG 1, "Codice Ric. : "+STR$( _WSORESP,0,0)+"|Desc.: "+_WSORESPD$+"|Tipo doc. : "+_WSOCNTYPE$ '(Content-Type, body risposta Server)

'Nota: non e' inserita la routine per eventuale refresh token dopo il timeout dell'autenticazione bearer. Si suppone Token OK.

RETURN ' ritorno alle routine che hanno lanciato la richiesta del Token tramite la GOSUB CALL_TKN_WS_HTTP

'*****
' TEST STATO HTTP
'*****
'-----
'Formattazione JSON REPONSE status code http
'-----
'
' {
'
' "message":"Messaggio generico dell'errore" ,
' "data": [],
' }
'-----

TEST_HTTP: ' codici stato chiamata http per come stabiliti dal Servizio

STATUS_CODE= _WSORESP
IF STATUS_CODE=200 THEN VIMSG 1, "Operazione riuscita correttamente"
IF STATUS_CODE=400 THEN VIMSG 1, "Validazione non conforme o mancante"
IF STATUS_CODE=404 THEN VIMSG 1, "Campi non trovati"
IF STATUS_CODE=500 THEN VIMSG 1, "Errore di sistema"

DIM ERR$( ) : DIM MSG$( )
IFF ( _WSORESP=0)
IF ( _WSRAWERR$<>"") THEN ERR=1 : N_ERR=N_ERR+1 : ERR$(N_ERR)="CALLWEBSVC: Errore di comunicazione con WS"
N_ERR=N_ERR+1 : ERR$(N_ERR)="_WSRAWERR: "+_WSRAWERR$ + "|" + _WSORESPD$ + "|" + _WSOCNTYPE$
ELSEF
IFF ( _WSORESP>=100 AND _WSORESP<=299)
'HTTP status codes: lxx - Information / 2xx - Success
MSG=1: N_MSG=N_MSG+1: MSG$(N_MSG)="CALLWEBSVC: Comunicazione WS OK"
N_MSG=N_MSG+1: MSG$(N_MSG)="HTTP status codes: "+STR$( _WSORESP,0,0)+" - "+_WSORESPD$
ELSEF
'HTTP status codes: 3xx - Redirection / 4xx - Client Error / 5xx - Server Error
ERR=2: N_ERR=N_ERR+1: ERR$(N_ERR)="CALLWEBSVC: Comunicazione WS OK":
N_ERR=N_ERR+1: ERR$(N_ERR)="HTTP status codes: "+STR$( _WSORESP,0,0)+" - "+_WSORESPD$:
ENDF

'FOR CH=1 TO 200
' IFF ( _WSOHEADER$(CH)<>"")
'
' IF (CH=1) THEN IF (ERR<>0) THEN N_ERR=N_ERR+1: ERR$(N_ERR)="|header|": ELSE N_MSG=N_MSG+1: MSG$(N_MSG)="|header|":
' IFF ( _WSOHEADER$(CH)<>CHR$(13)+CHR$(10) AND _WSOHEADER$(CH)<>CHR$(13) AND _WSOHEADER$(CH)<>CHR$(10)
' M$= _WSOHEADER$(CH): IF RIGHT$(M$,2)=CHR$(13)+CHR$(10) THEN M$=LEFT$(M$,LEN(M$)-2): ELSE IF RIGHT$(M$,1)=CHR$(13)
OR RIGHT$(M$,1)=CHR$(13) THEN M$=LEFT$(M$,LEN(M$)-1):
' IF (ERR<>0) THEN N_ERR=N_ERR+1: ERR$(N_ERR)=M$: ELSE N_MSG=N_MSG+1: MSG$(N_MSG)=M$:
'
' ENDF
' ELSEF
' CH=200+1:
' ENDF
'NEXT
'FOR CB=1 TO 200
' IF (CB=1) THEN IF (ERR<>0) THEN N_ERR=N_ERR+1: ERR$(N_ERR)="|body|": ELSE N_MSG=N_MSG+1: MSG$(N_MSG)="|body|":
' IFF ( _WSOBODY$(CB)<>"")
'
' IF (CB=1) THEN IF (ERR<>0) THEN N_ERR=N_ERR+1: ERR$(N_ERR)="|body|": ELSE N_MSG=N_MSG+1: MSG$(N_MSG)="|body|":
' IFF ( _WSOBODY$(CB)<>CHR$(13)+CHR$(10) AND _WSOBODY$(CB)<>CHR$(13) AND _WSOBODY$(CB)<>CHR$(10)
' M$= _WSOBODY$(CB): IF RIGHT$(M$,2)=CHR$(13)+CHR$(10) THEN M$=LEFT$(M$,LEN(M$)-2): ELSE IF RIGHT$(M$,1)=CHR$(13) OR
RIGHT$(M$,1)=CHR$(13) THEN M$=LEFT$(M$,LEN(M$)-1):
' IF (ERR<>0) THEN N_ERR=N_ERR+1: ERR$(N_ERR)=M$: ELSE N_MSG=N_MSG+1: MSG$(N_MSG)=M$:
'
' ENDF
' ELSEF
' CB=200+1:
' ENDF
'NEXT
ENDF
'IF (ERR<>0) THEN ....

'-----
' 1 modo) STAMPA HEADER DA ANALIZZARE (alcuni possono essere stati splittati su elementi di _WSOHEADER$( )
'-----
' IX_HDR=1
' ANALISI HEADER OUTPUT:
' IF _WSOHEADER$(IX_HDR) = "" THEN GOTO PROSEGUI
' IF _WSOHEADER$(IX_HDR) <> HTTPHDRTAG$( "BEGIN" ) THEN PRINT _WSOHEADER$(IX_HDR): GOTO NXT_HDR
' PRINT LONG HEADER:
' IX_HDR = IX_HDR + 1
' IF _WSOHEADER$(IX_HDR) <> HTTPHDRTAG$( "END" ) THEN PRINT _WSOHEADER$(IX_HDR):GOTO PRINT_LONG_HEADER ELSE GOTO NEXT_HDR
' NEXT_HDR:
' IX_HDR = IX_HDR+1
' GOTO ANALISI_HEADER_OUTPUT
'-----
' 2 modo) STAMPA HEADER DA ANALIZZARE (alcuni possono essere stati splittati su elementi di _WSOHEADER$( )
'-----
' PRINT_HEADERS:
' APRIFIN TIT 1,1,0,2,2,15,70,"Header HTTP"," : SETTF "505-Chiudi" : I=1 : LASTFIN=1 : PLNCHR=68 : PLNROW=2 : PLNCOL=2

```

```

'      GOSUB PRINTLINE
'      NXT_HEADER:
'      IF _WSOHEADERS(I) = "" THEN GOTO PRINT_HEADERS_END
'      PLIN$= WSOHEADERS(I)
'      GOSUB PRINTLINE
'      I=I+1
'      GOTO NXT_HEADER
'      PRINT_HEADERS_END:
'      RETURN
-----
RETURN

#####
' RILASCIO RISORSE #
#####

JSON_L_EXIT: 'Rilascio delle risorse utilizzate per la LETTURA
JSON_S_EXIT: 'Rilascio delle risorse utilizzate per la SCRITTURA

JSONDESTROY ROOTOBJ : ROOTOBJ = 0
IF RIFN1 > 0 THEN JSONDESTROY RIFN1 : RIFN1 = 0
IF RIFN2 > 0 THEN JSONDESTROY RIFN2 : RIFN2 = 0

'-----
'CHIUSURA
'-----
JSON_L_END: 'Chiusura Lettura JSON
JSON_S_END: 'Chiusura Scrittura JSON
END

'-----
' STAMPE DI CONTROLLO
'-----
'IF _WSOFNAME$ = "" THEN GOSUB PRINT_BODY_ARRAY ELSE GOSUB PRINT_BODY_FILE

CHIUDIFIN 2
CHIUDIFIN 1

'PRINT_BODY_ARRAY:
'=====
''      STAMPA DEL BODY ( risposta vera e propria del servizio )
''      la sua struttura sempre e' imposta dalla definizione WSDL del servizio
'=====
'APRIFIN TIT 2,1,0,1,4,18,78,"Body HTTP","" : SETTF "505-Chiudi" : LASTFIN=2 : I=1 : PLNROW=2 : PLNCOL=2
'NXT_BODYLINE:
' IF _WSOBODY$(I) = "" THEN GOTO PRINT_BODY_ARRAY_END
' PLIN$= _WSOBODY$(I) : PLNCHR=68
' GOSUB PRINTLINE
' I=I+1
' GOTO NXT_BODYLINE
' PRINT_BODY_ARRAY_END:
' GOSUB WAITESC
' RETURN
'
'PRINT_BODY_FILE:
'=====
''      STAMPA DEL BODY FILE ( risposta vera e propria del servizio )
''      la sua struttura sempre e' imposta dalla definizione WSDL del servizio
'=====
'APRIFIN TIT 2,1,0,1,4,18,78,"Body HTTP","" : SETTF "505-Chiudi" : LASTFIN=2
'OPEN "R", 1, _WSOFNAME$, 1
'FIELD 1, 1 AS CH$
'LF=LOF(1) : NPR=1 : PLNCHR=78 : FL=0
'LOCATE 2,2
'FOR B=1 TO LF
'  GET 1,B
'  IF NPR >= PLNCHR THEN PRINT: NPR=1
'  IF CH$ = "/" THEN IF FL=0 THEN FL=3
'  IF FL=1 THEN IF CH$="/" THEN FL=2 ELSE FL=0
'  IF CH$ = "<" THEN IF FL=0 THEN FL=1 ELSE FL = 0
'  IF CH$ = ">" AND ( FL=2 OR FL = 3 ) THEN PRINT CH$ : NPR=1 : FL=0 ELSE NPR=NPR+1 : PRINT CH$;
'NEXT
'CLOSE 1
'PRINT_BODY_FILE_END:
'GOSUB WAITESC
'RETURN
'
'=====
''      STAMPA UNA RIGA DI TESTO
'=====
'PRINTLINE:
'LN=LEN(PLIN$)
'PLTMP$=PLIN$
'IF RIGHT$(PLTMP$, 1 ) = CHR$(10) THEN LX=LEN(PLTMP$):PLTMP$=LEFT$(PLTMP$,LX-1)
'IF RIGHT$(PLTMP$, 1 ) = CHR$(13) THEN LX=LEN(PLTMP$):PLTMP$=LEFT$(PLTMP$,LX-1)
'PRINT_CHUNK:
'PSL=LEN(PLTMP$)
'IF PSL <= PLNCHR THEN LOCATE PLNROW, PLNCOL : PRINT PLTMP$ : PLNROW = PLNROW + 1 : RETURN
'PSS$=LEFT$(PLTMP$, PLNCHR)
'LOCATE PLNROW, PLNCOL
'PRINT PSS$;
'PLNROW = PLNROW + 1
'PLTMP$=RIGHT$(PLTMP$, PSL-PLNCHR)
'GOTO PRINT_CHUNK
'RETURN
'
'=====
''      ATTENDE LA PRESSIONE DI UN TASTO
'=====
'WAITESC:
'_INTER=0
'STARTWAIT:
'A=INKEY()
'IF A <> 505 THEN GOTO STARTWAIT
'_INTER=1
'RETURN
'
'=====
''      SCRIVE OGNI CARATTERE DEL BUFFER DI TESTO < WRITEBUFF$> nel file <NUMFILE> nel record <CURRREC>

```

```
'=====
'SUB WRITE_BUFF:
' FOR ISL = 1 TO LEN( WRITEBUFF$ )
'   CH$=MID$( WRITEBUFF$, ISL, 1 )
'   PUT NUMFILE, CURRREC
'   CURRREC = CURRREC + 1
' NEXT
'RETURN

#####

RETURN 'return del POST/GET

FINE:
END
```

COLLAGE DATATREK

DataTrek è un **framework** della suite di sviluppo PassBuilder che consente di costruire facilmente delle libere strutture di dati aziendali organizzati, da navigare su DataBase SQLite (di sola scrittura e lettura, non cancellazione,) strutturato in libere forme di concatenazioni di dati disponibili. Dalla versione 2020A (v750) la licenza MDS consente la replica del data base in SQLite chiaro. Dtk consente al progettista della struttura Dtk di realizzare un proprio e personale DataBase, in cui le informazioni inserite vengono aggregate ed analizzate secondo regole di Business realizzate liberamente dal progettista stesso. Il funzionamento di Dtk si basa su file (documenti) "Xml", nei quali è possibile gestire una parte di tali regole utilizzabili durante la navigazione dei dati nella base dati SQLite. I dati del Gestionale, infatti, possono essere gestiti sia tramite Sprix che tramite accesso al data base in replica MexalDB con SQLite. Data Trek consente di interagire con le informazioni presenti sulla sua base dati, ed eventualmente del MexalDb SQLite. Ciò è possibile solo ed esclusivamente dall'app DataTrek. In programmazione Sprix/Collage fare attenzione alla chiusura degli Statement, poichè se lasciati aperti le successive operazioni creano errori di data base lock. La base dati impiegata, infatti, è di tipo DBMS e non RDBMS, pertanto il programmatore deve ricordarsi di rilasciare il database per evitare errori. SQLite esegue il lock a livello di file system (il database viene allocato nella sua interezza).

DataTrek è uno strumento per navigare i dati del Gestionale a 360°, memorizzati su DataBase personale; l'interfaccia consente di percorrere le proprie personali concatenazioni di informazioni, anche invertendone il percorso. La funzionalità di base relativa a questa navigazione è implementabile ulteriormente con delle funzioni che relazionano fra loro i dati, ma è possibile spingersi oltre, implementando delle speciali procedure di lettura e scrittura dei dati tramite strumenti nativi, come Sprix. In questa sede saranno elencati i comandi disponibili per eseguire le personalizzazioni (oltre alle istruzioni SQL disponibili, vedere tabella comandi di accesso a DataBase), per altri approfondimenti sullo strumento si rimanda all'apposito Manuale DataTrek presente nel capitolo PassBuilder nel Manuale Utente.

La personalizzazione dell'**oggetto** DataTrek avviene tramite la creazione di un Collage DataTrek. La struttura di Collage Datatrek è identica a quella del Collage tradizionale, ma con le seguenti caratteristiche:

- collage DataTrek agisce esclusivamente sul framework DataTrek;
- collage DataTrek consente di utilizzare le Funzioni e le Istruzioni Sprix, compreso le GET e PUT degli archivi; quindi è in grado di leggere e scrivere direttamente nel Gestionale (funzioni che il Collage tradizionale non può fare), verosimilmente a quanto riesce a fare Collage Server Remoto a disposizione per lo Sprix Mobile;
- collage DataTrek non dispone di Ciclotabellare;
- collage DataTrek non utilizza comandi GETATT del Collage tradizionale, bensì si avvale della DTK_GETVALUE;
- collage DataTrek dispone di etichette proprietarie di DataTrek che gli consentono di eseguire delle routine;
- collage DataTrek dispone di istruzioni proprietarie di DataTrek e di variabili di ambiente;

Durante l'esecuzione dell'applicazione DataTrek, si verificano diversi eventi e al verificarsi di questi il programmatore può inserirsi creando dei comportamenti personalizzati. Gli eventi fanno riferimento a vari precisi momenti:

- avvio applicazione DataTrek;
- invocazione di un elemento del ribbon (voce di menù) (= attivazione tablepage "home");
- attivazione delle tablepages: "detail", "edit" ("nuovo" inserimento e/o "modifica");
- interazione nelle tablepages "home", "detail", "edit";
- alla conferma di inserimenti e/o variazioni (pressione del tasto funzione [F10]);

Traduzione degli attributi in seguito riportati. **Nota:** il valore assegnato all'attributo "LC" può essere al massimo 4 caratteri e la composizione di NomeTabella con il NomeCampo non può superare gli 8 caratteri.

NomeMenu = nome menu

NomeTabella = nome stabilito nell'attributo "LC" (label collage) della Tag <objectTable name="Xyz" labelcollage="DBTX">

NomeCampo = nome stabilito nell'attributo "LC" (labelcollage) della Tag <field ... labelcollage="URL">

Eventi sulla procedura	S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
<p>ON_STARTDATATREK: Variabili d'ambiente disponibili: nessuna</p> <p>Evento di procedura relativo all'avvio dell'applicazione DataTrek. Quando la procedura è richiamata dalla relativa voce di menù, al termine del processo di inizializzazione del framework (carico dei documenti Xml, esecuzione di eventuali processi di ETL ed eventuali import da CSV e prima di mostrare la vista predefinita di avvio), DataTrek esegue il codice personalizzato inserito sotto questa etichetta evento.</p>	Cd
<p>ON_MENUOBJECT_<attributo "LC" dell'objectTable>: Variabili d'ambiente disponibili: nessuna.</p> <p>Evento su video attivo sul click della voce di menu delle tabelle. Questo evento scatta all'invocazione di un elemento del ribbon e che corrisponde all'attivazione della tablePage "home". Scatta quando l'utente richiede a DataTrek di attivare la vista corrispondente all'oggetto tabella per la quale è definita l'etichetta collage specificata.</p>	Cd


```
<?xml version="1.0" encoding="iso-8859-1" ?>
<xml>
  <objectTable name="TBL_TIPO_CAMERA" labelwidth="20" labelCollage="TC">
    <fieldList>
      <field name="TIPO_CAMERA" type="text" lenght="03" required="1" trackHistory="1" labelCollage="TCAM" ></field>
      <field name="DESCRIZIONE" type="text" lenght="64" required="1" trackHistory="1" labelCollage="DESC" ></field>
      <field name="POSTI_LETTO" type="long" required="0" trackHistory="1" labelCollage="PL" ></field>
      <field name="FL_LETTI_SEPARATI" type="bool" required="0" trackHistory="1" labelCollage="FLLS" ></field>
      <field name="FL_LETTO_MATRIMONIALE" type="bool" required="0" trackHistory="1" labelCollage="FLMA" ></field>
    </fieldList>
  </objectTable>
  ON_MENUOBJECT_TC:
  VIMSG 1, "Sto per aprire la tabella relativa alle Tipologie Camere."
  _DKSTOPSTATE = 1
END
```



ON_DETAILOBJECT_<attributo "LC" dell'objectTable>:
 Variabili d'ambiente disponibili: _DKRECORDID\$

Cd

Evento su video attivo al click "detail" sul record di una tabella. L'evento scatta all'attivazione della tablepage "detail", cioè quando l'utente clicca sul tasto dettaglio in corrispondenza delle righe dell'elenco. DataTrek rilascia una variabile contenente il recordID della riga selezionata.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<xml>
  <objectTable name="TBL_TIPO_CAMERA" labelwidth="20" labelCollage="TC">
    <fieldList>
      <field name="TIPO_CAMERA" type="text" lenght="03"
      <field name="DESCRIZIONE" type="text" lenght="64"
      <field name="POSTI_LETTO" type="long"
    </fieldList>
  </objectTable>
  ON_DETAILOBJECT_TC:
  VIMSG 1, "Ho cliccato"
END
```



ON_NEWOBJECT_<attributo "LC" dell'objectTable>:
ON_EDITOBJECT_<attributo "LC" dell'objectTable>:
 Variabili d'ambiente disponibili: _DKRECORDID\$ (disponibile solo su on_editobject)

Cd

Eventi su video attivi quando nella maschera si preme il pulsante "Nuovo" (new) o "Modifica" (edit). Gli eventi scattano all'attivazione della tablepage "edit" (nuovo inserimento e/o modifica), cioè quando l'utente clicca sul tasto "Nuovo" o sul tasto "Modifica" della vista. La variabile d'ambiente che contiene il recordID è disponibile solo al verificarsi dell'evento ON_EDITOBJECT in quanto al momento di creare un nuovo record, il recordID non è ancora disponibile. Infatti il recordID viene creato in automatico da DataTrek al momento di scrivere fisicamente le informazioni sulla base dati.



ON_STOPNEW_<attributo "LC" dell'objectTable>:

Variabili d'ambiente disponibili: _DKSTOPSTATE

Evento di procedura attivo nel form che elimina il pulsante nuovo dalla lista. L'evento è relativo all'**interazione nella tablepage "home"**, cioè quando l'utente attiva la vista principale sull'oggetto tabella a cui si riferisce e dopo l'evento ON_MENUBJECT. L'evento serve per controllare l'inibizione della funzione di inserimento nuovo record. All'interno di questo evento si fanno le dovute considerazioni del caso e nel momento in cui si vuole bloccare l'accesso alla funzione di modifica basta settare la variabile d'ambiente `_DKSTOPSTATE = 0`. Questo aspetto vale solo per gli utenti non amministratori.

```
ON_STOPNEW_TC:
  VIMSG 1, "attivazione evento ON_STOPNEW"
  _DKSTOPSTATE = 0
END
```

Cd

ON_STOPEDIT_<attributo "LC" dell'objectTable>:

Variabili d'ambiente disponibili: _DKSTOPSTATE

Evento di procedura attivo nel form che elimina il pulsante modifica dal dettaglio. L'evento è relativo all'**interazione nella tablepage "detail"**, cioè quando l'utente attiva la tablepage di dettaglio e dopo l'evento ON_DETAILOBJECT. L'evento serve per controllare l'inibizione della funzione di modifica. All'interno di questo evento si fanno le dovute considerazioni del caso e nel momento in cui si vuole bloccare l'accesso alla funzione di modifica basta settare la variabile d'ambiente `_DKSTOPSTATE = 0`. Questo aspetto vale solo per gli utenti non amministratori. Esempio:

```
ON_STOPEDIT_TC:
  VIMSG 1, "attivazione evento ON_STOPEDIT"
  _DKSTOPSTATE = 0
END
```

Cd

ON_IN_<id menu mx + "LC" objectTable + "LC" field>:

ON_OUT_<id menu mx + "LC" objectTable + "LC" field>:

Variabili d'ambiente disponibili:

- `_DKRECORDID$` (disponibile solo se si entra in tablePage "edit" dal tasto "Modifica")
- `_DKSTOPSTATE` (se impostata a 1, interrompe l'esecuzione della procedura e torna sul primo campo di input)
- `_DKSETTF$` (analogo alla SETTF, abilita i tasti funzioni specificati)

Eventi di procedura attivo nel form all'ingresso (IN) o al uscita (OUT) di un campo di input. Gli eventi sono relativi all'**interazione nella tablepage "edit"**, cioè quando l'utente entra e abbandona il campo specificato. Per identificare univocamente un campo DataTrek vuole l'id del menù del gestionale dove si trova l'App concatenato con le etichette collage specificate sull'objectTable e sul campo della tabella. Esempio, per un'App collocata nel menu "AX" (Azienda) dove "TC" è l'etichetta collage impostata sull'oggetto tabella e "DESC" è l'etichetta impostata sul campo "descrizione" della tabella.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<xml>
  <objectTable name="TBL_TIPO_CAMERA" labelwidth="20" labelCollage="TC">
    <fieldlist>
      <field name="TIPO_CAMERA" type="text" lenght="03" required="1" trackHistory="1" labelCollage="TCAM" ></field>
      <field name="DESCRIZIONE" type="text" lenght="64" required="1" trackHistory="1" labelCollage="DESC" ></field>
      <field name="POSTI_LETTO" type="long" required="0" trackHistory="1" labelCollage="PL" ></field>
      <field name="FL_LETTI_SEPARATI" type="bool" required="0" trackHistory="1" labelCollage="FLLS" ></field>
      <field name="FL_LETTO_MATRIMONIALE" type="bool" required="0" trackHistory="1" labelCollage="FLMA" ></field>
    </fieldlist>
  </objectTable>
```

ON_IN_AXTCDESC:

App/MyDB (X)

ID menu/ID finestra/ID campo

AX/NULL/NULL

Cd

<p>ON_TastoFunzione: Variabile d'ambiente disponibile: _DKRECORDID\$ (disponibile solo all'ingresso in tablePage "edit" dal tasto "Modifica")</p> <p>Evento di procedura attivo nel form alla pressione di un tasto funzione personalizzato fra quelli specificati e abilitati tramite l'utilizzo della variabile _DKSETTF\$. L'evento è relativo all'interazione nella tablepage "edit" per intercettare quale tasto viene premuto dall'utente. Esempio:</p> <p>ON_CLICKMEN_DBT1: _DKSETTF\$ = "607Mia Procedura": END ON_607: (è la procedura che parte alla pressione del tasto F7) : END</p>	Cd
<p>DTK_GETVALUE <"LC" objectTable>,<"LC" field> Variabili di struttura disponibili:</p> <ul style="list-style-type: none"> • _DKNOMCAMPO\$(1) – Nome del campo • _DKVALCAMPO\$(1) – Valore del campo • _DKVLCMPOLD\$(1) – Valore precedente del campo • _DKERRNMCMP\$(1) – Nome del campo in cui visualizzare l'errore • _DKERRCAMPO\$(1) – Testo di errore da visualizzare per il campo <p>Istruzione relativa all'interazione in tablepage "edit" che legge i dati contenuti a video nel momento in cui viene eseguita e ritorna i valori delle variabili di struttura _DK? (l'istruzione è analoga a GETATT). Le istruzioni per inizializzare (azzerare) le variabili di struttura _DK sono AZZVARSYS 39 oppure AZZCATVARSYS "DTK".</p>	Cd
<p>ON_CHECKSTATE_<attributo "LC" dell'objectTable>: Variabili d'ambiente disponibili:</p> <ul style="list-style-type: none"> • _DKSTOPSTATE • _DKRECORDID\$ (disponibile solo all'ingresso in tablePage "edit" dal tasto "Modifica") <p>Evento di procedura attivo alla conferma con il tasto [F10] di inserimenti e/o variazioni dalla tablePage "edit". Questo evento ci consente di verificare lo stato di editing (variabile _DKSTOPSTATE) e decidere quale azione intraprendere. E' un evento utile per effettuare controlli prima di eseguire un'operazione di inserimento/modifica. In questo evento è attiva la variabile _DKSTOPSTATE che, se impostata ad 1, blocca l'esecuzione della procedura rimandando a video un eventuale messaggio di errore. La presenza di questo evento inibisce la scrittura automatica da parte del framework delle informazioni sulla base dati, ma ne demanda l'onere al programmatore. La variabile d'ambiente _DKRECORDID\$ è valorizzata <u>solo in fase di modifica</u>. Esempio.</p> <p>ON_CHECKSTATE_TC: VIMSG 1, "attivazione evento ON_CHECKSTATE" VIMSG 1, "Record ID : " + _DKRECORDID\$ _DKSTOPSTATE = 0 I=0 AZZVARSYS 39 IF A\$="" THEN _DKSTOPSTATE=1: I=I+1: _DKNOMCAMPO\$(I)="campo01" : _DKERRCAMPO\$(I)="Err campo 1 vuoto" END</p>	Cd
<p>ON_BEFOREINSERT_<attributo "LC" dell'objectTable>: Variabili d'ambiente disponibili: nessuna</p> <p>Evento attivo su tabelle alla conferma con il tasto [F10] di inserimenti dalla tablePage "edit" <u>prima</u> della Insert, quindi se e solo se la "edit" è stata aperta dalla funzione "Nuovo". Questo evento consente di intervenire e svolgere considerazioni <u>prima</u> che il framework scriva le informazioni sulla base dati (ad esempio controlli di validità). Esempio:</p> <p>ON_BEFOREINSERT_TC: VIMSG 1, "attivazione evento ON_BEFOREINSERT" END</p>	Cd
<p>ON_BEFOREUPDATE_<attributo "LC" dell'objectTable>: Variabili d'ambiente disponibili: nessuna.</p> <p>Evento attivo su tabelle alla conferma con il tasto [F10] di variazioni dalla tablePage "edit" <u>prima</u> del Update, quindi se e solo se la "edit" è stata aperta dalla funzione "Modifica". Questo evento consente di intervenire e svolgere considerazioni <u>prima</u> che il framework aggiorni le informazioni sulla base dati. Esempio:</p> <p>ON_BEFOREUPDATE_TC: VIMSG 1, "attivazione evento ON_BEFOREUPDATE" END</p>	Cd
<p>ON_BEFORESAVE_<attributo "LC" dell'objectTable>: Variabili d'ambiente disponibili: nessuna.</p> <p>Evento attivo su tabelle alla conferma con il tasto [F10] di inserimento e/o variazioni dalla tablePage "edit" <u>prima</u> del Salvataggio ed indipendentemente dal punto da cui la "edit" è stata aperta. Questo evento consente di intervenire e svolgere considerazioni <u>prima</u> che il framework scriva e/o aggiorni le informazioni sulla base dati. Questo evento è utile quando si vuole gestire un comportamento comune alla scrittura e/o all'aggiornamento. Esempio:</p>	Cd

<p>ON_BEFORESAVE_TC: VIMSG 1, "attivazione evento ON_BEFORESAVE" END</p>	
<p>ON_AFTERINSERT_<attributo "LC" dell'objectTable>: Variabile d'ambiente disponibile: _DKRECORDID\$</p> <p>Evento attivo su tabelle alla conferma con il tasto [F10] di inserimenti dalla tablePage "edit" <u>dopo</u> della Insert, quindi se e solo se la "edit" è stata aperta dalla funzione "Nuovo". Questo evento consente di intervenire e svolgere considerazioni <u>dopo</u> che il framework abbia salvato le informazioni sulla base dati. La variabile d'ambiente _DKRECORDID\$ contiene il riferimento univoco al "nodo" appena memorizzato. Esempio:</p> <p>ON_AFTERINSERT_TC: VIMSG 1, "attivazione evento ON_AFTERINSERT" VIMSG 1, "Record ID : " + _DKRECORDID\$ END</p>	Cd
<p>ON_AFTERUPDATE_<attributo "LC" dell'objectTable>: Variabile d'ambiente disponibile: _DKRECORDID\$</p> <p>Evento attivo su tabelle alla conferma con il tasto [F10] di variazioni dalla tablePage "edit" dopo del Update, quindi se e solo se la "edit" è stata aperta dalla funzione "Modifica". Questo evento consente di intervenire e svolgere considerazioni <u>dopo</u> che il framework abbia aggiornato le informazioni sulla base dati. La variabile d'ambiente _DKRECORDID\$ contiene il riferimento univoco al "nodo" appena aggiornato. Esempio:</p> <p>ON_AFTERUPDATE_TC: VIMSG 1, "attivazione evento ON_AFTERUPDATE" VIMSG 1, "Record ID : " + _DKRECORDID\$ END</p>	Cd
<p>ON_AFTERSAVE_<attributo "LC" dell'objectTable>: Variabile d'ambiente disponibile: _DKRECORDID\$</p> <p>Evento attivo su tabelle alla conferma con il tasto [F10] di inserimento e/o variazioni dalla tablePage "edit" <u>dopo</u> il Salvataggio ed indipendentemente dal punto da cui la "edit" è stata aperta. Questo evento consente di intervenire e svolgere considerazioni <u>dopo</u> che il framework abbia scritto e/o aggiornato le informazioni sulla base dati. Questo evento è utile quando si vuole gestire un comportamento comune dopo la scrittura e/o l'aggiornamento. La variabile d'ambiente _DKRECORDID\$ contiene il riferimento univoco al "nodo" appena aggiornato. Esempio:</p> <p>ON_AFTERSAVE_TC: VIMSG 1, "attivazione evento ON_AFTERSAVE" VIMSG 1, "Record ID : " + _DKRECORDID\$ END</p>	Cd
<p>DTK_GETCOLONNA istruzione che ritorna il valore del campo nell'elemento della BINDCOL dichiarata in precedenza.</p> <p>E' possibile interagire con le informazioni presenti sulla base dati Data Trek da programmi Sprix tramite apposite istruzioni. Per effettuare una richiesta:</p> <ul style="list-style-type: none"> • Stabilire la connessione con SQL_DRIVERCONN (modulo 4) e fare riferimento al database tramite il codice oggetto DataTrek. I programmi devono trovarsi rigorosamente in un'App DataTrek. <u>Non è consentito accedere a oggetti DataTrek presenti su altre PassApp.</u> • Allocare un nuovo statement : SQL_ALLOCSTMT • Effettuare il Bind con le variabili sprix (SQL_BINDCOL) • Eseguire la richiesta SQL : SQL_EXECDIRECT • Estrazione di un record per volta (SQL_FETCH) • Estrazione per singolo campo e bind del valore (DTK_GETCOLONNA) • Al termine della lettura rilasciare le risorse dello statement (SQL_FREESTMT) • Chiudere la connessione con SQL_DISCONNECT <p>Per ulteriori approfondimenti consultare le istruzioni SQL elencate nella tabella di raggruppamento comandi per le interzioni con database.</p>	S/C/Cr/Cd
<p>_DK Variabili di struttura di Collage Datatrek</p>	
<p>esempi e casi</p>	
<p>Esempio: SQL_DRIVERCONN e DTK_GETCOLONNA</p> <pre> SQL_DRIVERCONN 4,1,"", "", "", 0, "", "", "" IF _ERRSQL_S\$ <> "" THEN VIMSG 5, _ERRSQL_S\$ + " Connect" SQL_ALLOCSTMT 1,1 SQL_BINDCOL 1,1,1,CAMP0\$ IF _ERRSQL_S\$ <> "" THEN VIMSG 5, _ERRSQL_S\$ SQL_EXECDIRECT 1,1,"select count(recordid) AS tipes from TBPrima" IF _ERRSQL_S\$ <> "" THEN VIMSG 5, _ERRSQL_S\$ LEGGI_RECORD: SQL_FETCH 1,1: IF _ERRSQL_S\$ <> "" THEN VIMSG 5, _ERRSQL_S\$ IF _ERRSQL_NDF=1 THEN GOTO USCITA DTK_GETCOLONNA 1,1,"tipes",1 LPRINT CAMP0\$ </pre>	

```
GOTO LEGGI_RECORD:
USCITA:
```

Esempio SQL Datatrek

(nella sezione di comandi raggruppati per DATABASE - FILE ci sono altri esempi relativi ad istruzioni SQL generali)

Datatrek lavora con un database Sqlite. I comandi sono quelli standard di un qualunque Sql. Fa eccezione la connessione. Lavorando con un Database standard da Sprix la sequenza dei comandi da utilizzare è la seguente:

```
SQL_CONNECT ...
per connettersi al DB

SQL_ALLOCSTMT
per preparare lo statement da utilizzare

SQL_BINCOL ....
per preparare le variabili di ritorno di una eventuale select

SQL_EXECUTE... o SQL_EXEDIRECT...
per eseguire i comandi SQL

SQL_FETCH
per leggere il ritorno del comando SQL select.
```

Eventuali Comandi

```
SQL_FREESTMT per liberare lo statement
SQL_DISCONNECT per chiudere la connessione
```

In Datatrek la sequenza di operazione è quasi identica.

Creata un APP in cui è presente l'oggetto Datatrek ed un collage datatrek o un collage standard o uno sprix, la sequenza delle operazioni da eseguire è la seguente:

- SQL DRIVERCONN dove gli unici parametri richiesti sono il primo tipo di connessione e il secondo numero della connessione che deve essere univoco, se il comando è in collage datatrek, se si esegue invece il comando in un app con datatrek deve essere passato come nome del database quello del **codice dell'elemento DataTrek** (click sull'elemento DataTrek -> campo "Codice")

Nota: in DataTrek è prevista una specifica MexalDB che si attiva in voce di menu Servizi - Configurazioni - Configurazione moduli (Alt+ZCC) - MexalDB per DataTrek; occorre inoltre attivare il flag per la replica su MexalDB (abilitazione dell'app 123456APPLIC con MexalDB per DataTrek) che si trova nella Configurazione dell'elemento DataTrek; infine, aperto il programma DataTrek occorre popolare il database con Azienda - MexalDB - Trasferimento dati per DataTrek (Alt+ALD)

- SQL_ALLOCSTMT per preparare lo statement da utilizzare
- SQL_BINCOL per preparare le variabili di ritorno di una eventuale select
- SQL_EXEDIRECT... per eseguire i comandi SQL
- SQL_FETCH che a differenza del comando tradizionale non ritorna i valori nelle variabili create con la SQL_BINDCOL, ma subito dopo il comando SQL_FETCH bisogna eseguire il comando
- DTK_GETCOLONNA par1,par2,par3\$,par4 (par1 = numero della connessione)
par2 = numero dello statement
par3\$ = nome della colonna della tabella del DB
par4 = numero della bindcol a cui assegnare il valore

quindi i comandi per liberare lo statement e chiudere la connessione.
Vediamo un esempio.

Creato il nostro database ipotizziamo di avere al suo interno una table così composta:

```
Nome table = "fantasia"
campo1 type text
campo2 type text
campo3 type text
```

in Sprix/Collage DTK scriveremo questo sorgente:

```
SQL_DRIVERCONN 4,1,"","","",0,"","","" ` stabiliamo la connessione al DB
IF _ERRSQL_S$ <> "" THEN VIMSG 5, _ERRSQL_S$ + " Connect"
SQL_ALLOCSTMT 1,1 ` prepariamo lo statement
IF _ERRSQL_S$ <> "" THEN VIMSG 5, _ERRSQL_S$ + " Alloc"
SQL_BINDCOL 1,1,1,MIAVAR1$ ` bindiamo le variabili
SQL_BINDCOL 1,1,2,MIAVAR2$
SQL_BINDCOL 1,1,3,MIAVAR3$
SQL_EXECDIRECT 1,1,"select campo1,campo2,campo3 from fantasia " ` eseguiamo il comando sql
IF _ERRSQL_S$ <> "" THEN VIMSG 5, _ERRSQL_S$ + " Execdirect"
A LEGGI: ` ciclo per leggere il recordset
SQL_FETCH 1,1
IF _ERRSQL_S$ <> "" THEN VIMSG 5, _ERRSQL_S$
IF _ERRSQL_NDF=1 THEN GOTO A USCITA
DTK_GETCOLONNA 1,1,"campo1",1 ` il comando ritorna nella variabile MIAVAR1$ il contenuto del campo1
DTK_GETCOLONNA 1,1,"campo2",2 ` il comando ritorna nella variabile MIAVAR2$ il contenuto del campo2
DTK_GETCOLONNA 1,1,"campo3",3 ` il comando ritorna nella variabile MIAVAR3$ il contenuto del campo3
GOTO A LEGGI:
A USCITA:
SQL_FREESTMT 1,1 `libero lo statement
SQL_DISCONNECT 1 ` chiudo la connessione.
END
```

```
SQL_EXECDIRECT 1,1,"Insert into fantasia (campo1,campo2,campo3) values ('uno','due','tre')"
con questo comando eseguiamo una insert
```

Notare: indipendentemente da una insert, update oppure delete dopo la EXECDIRECT in Sqlite occorre fare comunque una SQL_FETCH per applicare le modifiche al DB

con questo sorgente abbiamo eseguito una select all'interno di datatrek

SPRIX MOBILE

<p style="text-align: center;">Attenzione</p> <p>Nella seguente tabella è riportata una sintesi dei comandi specifici per Sprix Mobile, ma occorre considerare che in questo Manuale Sprix & Collage sono riportati anche tanti altri comandi utilizzabili in Sprix Mobile (ad esempio #APPENDISPX) e la cui fruibilità viene indicata dalla sigla 'Sm' nella colonna a destra delle tabelle.</p> <p style="text-align: center;">PER SPRIX MOBILE E' DISPONIBILE L'ELEMENTO PASSBUILDER "COLLAGE CONTROLLO APP MOBILE" CHE CONSENTE DI UTILIZZARE ETICHETTE DI GESTIONE DEI SEGUENTI EVENTI: ON_NUOVA_APP: ON_MODIFICA_APP: ON_POST_SYNC_APP:</p> <p style="text-align: center;">PER APPROFONDIMENTI CONSULTARE IL MANUALE SPRIX MOBILE</p>		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc																																																				
AZZVARSYS 38	Azzerare variabili struttura _WF (form grafico)	S/C/Cr/Cd Sm/Cm/Sh																																																				
AZZCATVARSYS "WF"	Azzerare variabili struttura _WF (form grafico)	S/C/Cr/Cd Sm/Cm																																																				
AZZVARSYS 41	Azzerare variabili struttura _WI (input grafico)	S/C/Cr/Cd Sm/Cm/Sh																																																				
AZZCATVARSYS "WI"	Azzerare variabili struttura _WI (input grafico)	S/C/Cr/Cd Sm/Cm																																																				
AZZVARSYS 42	Azzerare variabili struttura _WL (lista grafica)	S/C/Cr/Cd Sm/Cm/Sh																																																				
AZZCATVARSYS "WL"	Azzerare variabili struttura _WL (lista grafica)	S/C/Cr/Cd Sm/Cm																																																				
AZZVARSYS 43	Azzerare variabili struttura _WOP (oggetto grafico)	S/C/Cr/Cd Sm/Cm/Sh																																																				
AZZCATVARSYS "WO"	Azzerare variabili struttura _WOP (oggetto grafico)	S/C/Cr/Cd Sm/Cm																																																				
AZZVARSYS 44	Azzerare variabili struttura _AF (verifica esistenza archivio)	S/C/Cr/Cd Sm/Cm/Sh																																																				
AZZCATVARSYS "AF"	Azzerare variabili struttura _AF (verifica esistenza archivio)	S/C/Cr/Cd Sm/Cm																																																				
WCREATEFORM	Par: <form\$>. Istruzione per la form di preparazione finestra	Sm/Cm																																																				
WCREATEINPUT	Par: <input\$>. Istruzione che definisce input campi da inserire nel form	Sm/Cm																																																				
WCREATELIST	Par: <lista\$>. Istruzione che esegue il disegno della lista	Sm/Cm																																																				
WSHOWFORM	Par: <idform\$>. Istruzione che mostra il form preparato	Sm/Cm																																																				
WPRTFORMOPT	Par: <idform\$>. Istruzione che a stampa offline di un form (sostituisce WCALL "PRINTFORM")	Sm																																																				
WCALL	Par: <azione\$>,<id\$>. Istruzione che esegue funzione per oggetto grafico	Sm																																																				
WSET	Istruzione che imposta il valore della proprietà dichiarata <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Wset</th> <th>Descrizione</th> <th>Collage</th> </tr> </thead> <tbody> <tr> <td>Input</td> <td>WIICOS\$</td> <td>Icona Pulsante</td> <td>NO</td> </tr> <tr> <td></td> <td>WIVALUES\$</td> <td></td> <td>SI, tranne nei pulsanti</td> </tr> <tr> <td></td> <td>WIDESS\$</td> <td>Label descrizione</td> <td>NO</td> </tr> <tr> <td></td> <td>WIHINT\$</td> <td>Placeholder</td> <td>NO</td> </tr> <tr> <td></td> <td>WIOUTONLY\$</td> <td></td> <td>SI</td> </tr> <tr> <td></td> <td>WILENCAR</td> <td></td> <td>NO</td> </tr> <tr> <td></td> <td>WILINES\$</td> <td></td> <td>NO</td> </tr> <tr> <td></td> <td>WILISTID</td> <td> <ul style="list-style-type: none"> Eliminare la lista Aggiungere una lista a un input Sostituire una lista esistente </td> <td>SI, solo nell'evento on_show</td> </tr> <tr> <td></td> <td>WILISTNFLD</td> <td></td> <td>SI, solo nell'evento on_show</td> </tr> <tr> <td>Form</td> <td>WFTITLE\$</td> <td>Titolo</td> <td>NO</td> </tr> <tr> <td></td> <td>WFHDRTXT\$</td> <td>Header</td> <td>NO</td> </tr> <tr> <td></td> <td>WFFTRTXT\$</td> <td>Footer</td> <td>NO</td> </tr> </tbody> </table>		Wset	Descrizione	Collage	Input	WIICOS\$	Icona Pulsante	NO		WIVALUES\$		SI, tranne nei pulsanti		WIDESS\$	Label descrizione	NO		WIHINT\$	Placeholder	NO		WIOUTONLY\$		SI		WILENCAR		NO		WILINES\$		NO		WILISTID	<ul style="list-style-type: none"> Eliminare la lista Aggiungere una lista a un input Sostituire una lista esistente 	SI, solo nell'evento on_show		WILISTNFLD		SI, solo nell'evento on_show	Form	WFTITLE\$	Titolo	NO		WFHDRTXT\$	Header	NO		WFFTRTXT\$	Footer	NO	Sm/Cm
	Wset	Descrizione	Collage																																																			
Input	WIICOS\$	Icona Pulsante	NO																																																			
	WIVALUES\$		SI, tranne nei pulsanti																																																			
	WIDESS\$	Label descrizione	NO																																																			
	WIHINT\$	Placeholder	NO																																																			
	WIOUTONLY\$		SI																																																			
	WILENCAR		NO																																																			
	WILINES\$		NO																																																			
	WILISTID	<ul style="list-style-type: none"> Eliminare la lista Aggiungere una lista a un input Sostituire una lista esistente 	SI, solo nell'evento on_show																																																			
	WILISTNFLD		SI, solo nell'evento on_show																																																			
Form	WFTITLE\$	Titolo	NO																																																			
	WFHDRTXT\$	Header	NO																																																			
	WFFTRTXT\$	Footer	NO																																																			
WGET	Par. <idobj\$>. Istruzione che legge il valore della proprietà dichiarata	Sm/Cm																																																				
WGETOID	Par: <obj\$>. Istruzione che restituisce l'oggetto (analoga WGET)	Sm/Cm																																																				
WCREATEDIALOG	Par: <idnmediolog\$>. Istruzione per gestire finestre di dialogo	Sm																																																				
WSHOWDIALOG	Par: <iddialog\$>. Istruzione per gestire finestre di dialogo	Sm																																																				
_ERRWF	Codice esito operazione WCREATEFORM.	Sm																																																				
_ERRWF\$	Descrizione esito operazione WCREATEFORM.	Sm																																																				
_ERRWL	Codice esito operazione WCREATELIST	Sm																																																				
_ERRWL\$	Descrizione esito operazione WCREATELIST	Sm																																																				
_ERRWI	Codice esito operazione WCREATEINPUT	Sm																																																				
_ERRWI\$	Descrizione esito operazione WCREATEINPUT	Sm																																																				
_ERRWP	Codice esito operazione WPRTFORMOPT	Sm																																																				
_ERRWP\$	Descrizione esito operazione WPRTFORMOPT	Sm																																																				
_WF	Variabili struttura WCREATEFORM	Sm																																																				
_WL	Variabili struttura WCREATELIST	Sm																																																				
_WOP	Variabili struttura per gli oggetti	Sm																																																				
_WI	Variabili struttura WCREATEINPUT	Sm																																																				
_WD	Variabili struttura WCREATEDIALOG	Sm																																																				
_WP	Variabili struttura WPRTFORMOPT	Sm																																																				
GETROWVALFLD	Par: <"vararch\$">,<varloc\$> Istruzione che valorizza variabile locale con	Sm																																																				

	il valore variabile di struttura	
GETARCHFLD	Istruzione che valorizza variabili _AF	Sm
AF	Variabili struttura esistenza archivio (GETARCHFLD)	Sm
SRVCALL	Par: <nome_coll_remoto\$>, <etichetta_server\$>, <etichetta_client\$> Istruzione che esegue elaborazioni bidirezionali Mobile-Server es.1 es.2	Sm
PUTREM_NUM	Par: <tag\$>, <numero> Istruzione per inserire numeri dal Mobile al Server e/o viceversa. es.	Cr/Sm
PUTREM_STR	Par: <tag\$>, <stringa\$> Istruzione per inserire stringhe dal Mobile al Server e/o viceversa. es.	Cr/Sm
PUTREM_ARRAY	Par: <tag\$>, <array_di_stringhe\$/array_di_numeri> Istruzione per inserire array dal Mobile al Server e/o viceversa. es.	Cr/Sm
PUTREM_FILE	Par: <tag\$>, <percorso_file\$> Istruzione per inserire file dal Mobile al Server e/o viceversa. es.	Cr/Sm
GETREM_NUM	Par: <tag\$>, <numero> Istruzione per prendere numeri da inviare al Mobile o al Server. es.	Cr/Sm
GETREM_STR	Par: <tag\$>, <stringa\$> Istruzione per prendere stringhe da inviare al Mobile o al Server. es.	Cr/Sm
GETREM_ARRAY	Par: <tag\$>, <array_di_stringhe\$/array_di_numeri> Istruzione per prendere array da inviare al Mobile o al Server. es.	Cr/Sm
GETREM_FILE	Par: <tag\$>, <percorso_file\$> Istruzione per prendere file da inviare al Mobile o al Server. es.	Cr/Sm
SRVSYNCDOC	Par: <doc\$>, <conto\$>, <etichetta\$> Istruzione sincronizzazione documento magazzino Mobile/Server. es.	Sm/Cm
ACQUISISCIFILE	Par: <etichetta\$>. Istruzione per acquisizione file dal dispositivo mobile.	Sm
ACQFILE\$	Variabile di struttura con riferimento all'oggetto acquisito.	Sm
ERRSSD	Codice esito operazione SRVSYNCDOC. es.	Sm
ERRSSD	Descrizione esito operazione SRVSYNCDOC. es.	Sm
SSD	Variabili di gestione sincronizzazione; es.	Sm
GETGPSDATA	Par: <AnnoMese>,[giorno],[azienda\$],[idutente\$],[idapp\$] Istruzione Collage Server Remoto per recupero dati struttura _GPS	Sm/Cr
ON_GETGPSDATA	Evento di call back	Sm
GPS	Variabili dati GPS	Sm
ON_START_SPRIX:	Evento in ingresso nella voce di menu	Sm
ON_IN_CAMPO:	Evento in ingresso di CAMPO (vedi WCREATEINPUT)	Sm
ON_OUT_CAMPO:	Evento in uscita di CAMPO (vedi WCREATEINPUT)	Sm
ON_PRESS_PULSANTE:	Evento in pressione di PULSANTE (vedi WCREATEINPUT)	Sm
ON_ROW_LISTA:	Evento in ogni riga disegnata da LISTA (vedi WCREATELIST)	Sm
ON_ROWFILTER_LISTA:	Evento in ogni riga disegnata da LISTA (vedi WCREATELIST)	Sm
ON_PRESS_LISTA:	Evento in conferma su riga di LISTA (da WCREATELIST)	Sm
ON_PRESS_Nome_tasto:	Evento tasti intercettabili in WCREATEDIALOG	Sm
ON_CART_LISTA:	Evento Carrello (attivato da _WLTYPE\$="CART"). In uscita disponibili le variabili: _WLARPRZNFLD(), _WLARPRZFMF\$, _WLARIMPDES\$, _WLARQTATAB\$, _WLARARTNUM, _WLARART\$, _WLARCOL(), _WLARQTA(), _WLARTAGTAB(), _WLARTAG(), _WLARTAGXL(), _WLARPRZ(), _WLARSCO\$(), _WLARIMPVAL(), _WLARPRO(), _WLARARTIND()	Sm/Cm
ON_CARTQTA_LISTA:	Evento Carrello richiamato al cambio della quantità / colli sull'articolo. In ingresso disponibili le variabili _WLARARTNUM, _WLARART\$, _WLARCOL(), _WLARQTA(), _WLARTAGTAB(), _WLARTAG(), _WLARTAGXL(); In uscita sono disponibili le variabili _WLARPRZ(), _WLARSCO\$(), _WLARIMPVAL(), _WLARPRO()	Sm/Cm
ON_CARTOK_LISTA:	Evento Carrello . In ingresso (tutti gli articoli nel carrello) sono disponibili le variabili WLARARTNUM, WLARART\$, WLARQTA(), WLARTAGTAB(), WLARTAG(), WLARTAGXL(), WLARCOL(), WLARPRZ(), WLARSCO\$(), WLARPRO(), WLARIMPVAL(), WLARARTIND()	Sm/Cm
ON_PRECLOSE_NOMEFORM:	Evento che scatta prima della chiusura del FORM (l'oggetto form può gestire solo due eventi che scattano alla chiusura del form stesso: tap sul tasto di back oppure cambio di menu, ovvero un altro sprix-mobile). Se è valorizzata la variabile _WFPCFMSG\$, quando scatta l'evento viene mostrata una finestra di avvertimento con il testo specificato. Viene valorizzata la variabile _WFPCFEVT: _WFPCFEVT=100 si sta facendo un back al form precedente; _WFPCFEVT=1 si sta uscendo dallo sprix mobile premendo Esci, Home oppure si sta cambiando Sprix mobile;	Sm/Cm
ON_SHOW_NOMEFORM:	Evento Collage ON_SHOW in cui è possibile aggiungere pulsanti all'App che si intende <u>estendere</u> . In questo evento è possibile estendere nuove proprietà <ol style="list-style-type: none"> 1. WCREATEINPUT con nome univoco e parentid form corrente altrimenti errore 2. posizione ("AFTER", "BEFORE", "BEFOREFIRST", "AFTERLAST"): _WIINSPOSS\$. Es: _WIINSPOSS\$="AFTER" 3. riferimento del campo : WIINSRIFID 4. Stessa condizione vale per i pulsanti ACTION Con AFTER e BEFORE il pulsante viene inserito dopo o prima dell'input specificato con _WIINSRIFID; con BEFOREFIRST e AFTERLAST, invece, il pulsante viene inserito prima del primo input che trova o dopo	Sm/Cm

	<p>l'ultimo.</p> <p>In caso di esecuzione ON_PRESS_PULSANTE: (pulsante definito in Collage)</p> <ol style="list-style-type: none"> Tutto consentito tranne lanciare eventi collage su oggetti collage Permessa la show form su form sprix solo su quello di partenza [WSHOWFORM: permessa solo su form creati dallo stesso collage o sul primo form sprix a cui sono associati gli input creati da collage.] <p>In caso di SHOW permessi solo i comandi:</p> <ol style="list-style-type: none"> WSET WGET WGETOID WCREATEINPUT : solo pulsanti o pulsanti action e solo sul form corrente. <p>Nota bene: tutti i nomi degli oggetti devono essere univoci. Si consiglia di inserire un prefisso in creazione di nuovi form.</p>	
ON_CLOSE_NOMEFORM:	Evento che scatta alla chiusura del FORM. In entrambi i casi, nelle rispettive routine di gestione dell'evento, non è possibile utilizzare istruzioni grafiche ma solo effettuare operazioni di salvataggio o pulizia di variabili.	Sm/Cm
ON_ARCODSTRU_OGGETTO:	Evento ch scatta all'uscita della maschera d'inserimento del articolo strutturato.	Sm/Cm
ON_BARCODE_NOMEINPUT:	Evento lettura Barcode/QRcode. Restituisce una array _WIBARCODE\$ (con quantità di elementi _WIBARNUM) di stringhe contenente il tipo di codice letto (_WIBARTYPE\$) e suddivisa in stringhe di 255 caratteri. Se nell'evento viene effettuata una settex del campo di input, il valore sovrascrive quello letto dal lettore barcode.	Sm/Cm
ON_ARQTA_OGGETTO	Evento che scatta quando la casella delle taglie prende il fuoco e si apre la maschera per la selezione facilitata delle taglie	Sm/Cm
ON_ARQTAOK_OGGETTO	Evento che scatta quando si sono scelte le taglie alla chiusura dell'interfaccia per l'inserimento facilitato.	Sm/Cm
ON_CHECKBOX_OGGETTO	Evento che scatta tutte le volte che l'input di tipo CHECKBOX cambia stato	Sm/Cm
GETSIGN	par: <titolo_finestra_firma\$> Istruzione di gestione della firma di presa visione. Per approfondimenti consultare il Manuale Sprix Mobile.	Sm/Cm
_SN	Variabili per la gestione della firma di presa visione. Per approfondimenti consultare il Manuale Sprix Mobile.	Sm/Cm
_ERRSN\$	Variabile di errore per la gestione della firma di presa visione. Per approfondimenti consultare il Manuale Sprix Mobile.	Sm/Cm
Gestione parametri in aree di memoria (LOC)		
PUTLOC_STR / NUM / ARRAY	Par: <tipo>,<chiave>,<valore>. Istruzione di scrittura	Sm
GETLOC_STR / NUM / ARRAY	Par: <tipo>,<chiave>,<valore>. Istruzione di lettura	Sm
DELLOC	Par: <tipo>,<chiave>; Istruzione di cancellazione selettiva	Sm
LSTLOC	Par: <tipo>; Lista parametri memorizzati. Valorizza array _LSTLOC(): array con tutte le chiavi _LSTLOCTIPO\$(): array con il tipo di dato ("S", "N", "A")	Sm
STORELOC	Par: <tipo>. Istruzione di salvataggio dati in memoria.	Sm
CLEARLOC	Par: <tipo>. Istruzione di cancellazione globale	Sm
_ERRLOC\$	Descrizione esito operazione LOC	Sm
	<p>Esempio:</p> <pre> PUTLOC_STR "T", "Cliente", CLI\$ (Temporaneo) GETLOC_STR "T", "Cliente", OLDCLI\$ PUTLOC_NUM "P", "Operazioni", NUMOP (Persistente) STORELOC "P", salva in modo persistente i dati in memoria. </pre>	
Iteratore archivi		
ITERARC	<p>Par: <archivio\$>,<nomeiteratore\$>; Funzione di richiamo dell'iteratore</p> <p>Dopo l'esecuzione dell'istruzione apparirà una progressbar non annullabile che indicherà il progresso dell'iteratore</p> <p>E' possibile interromperne l'esecuzione lato utente, in questo caso arriveranno gli errori al programmatore nelle variabili _ERRIA , _ERRIA\$</p> <p>Non è possibile eseguire contemporaneamente due o più iteratori.</p>	Sm
_IA	<p>Variabili struttura dati dell'iteratore</p> <p>_IAORDNM\$(1) = " ARCOD\$". - opzionale , indica i campi su cui ordinare _IAORDDR\$(1) = "D". - opzionale, indica le direzioni di ordinamento dei campi su cui ordinare</p>	Sm/Cm
ON_ITER_NOMEITER:	Etichetta evento di inizio Iteratore	Sm/Cm
ON_ITERFILTER_NOMEITER:	<p>Etichetta evento Iteratore</p> <p>Variabili:</p> <ol style="list-style-type: none"> _IAOK\$ _IASTOP\$ 	Sm/Sm
ON_ITEREND_NOMEITER:	Etichetta evento di termine Iteratore	Sm/Cm
_ERRIA	Codice Errore	Sm/Cm

ERRIAS\$	Descrizione Errore	Sm/Cm
ON_NUOVA_APP	<p>Evento per agire con Collage di controllo App sugli stati dell'App Quando si verifica questo evento, se l'etichetta è stata opportunamente gestita dentro l'elemento Collage Controllo App Mobile, vengono valorizzate le seguenti variabili di struttura: _WCCOD\$: Codice dell'app che è stata installata _WCVER\$: Versione dell'app</p>	Ca
ON_MODIFICA_APP	<p>Evento per agire con Collage di controllo App sugli stati dell'App Questo evento, se opportunamente gestito, valorizza le seguenti variabili di struttura: _WCCOD\$: Codice dell'app che è stata modificata _WCVER\$: Versione aggiornamento (Nota: l'evento scatta anche se la versione rimane invariata) _WCVERSPRE\$: Versione precedente all'aggiornamento</p>	Ca
ON_POST_SYNC_APP	<p>Evento ad ogni volta che, dopo una sincronizzazione, vengono aggiornati dei record</p> <p>Questo evento, se opportunamente gestito, scatta ogni volta che si esegue una sincronizzazione e il processo di sincronizzazione aggiorna dei record. E' possibile definire anche un pulsante, che comparirà nella finestra di riepilogo della sincronizzazione, alla pressione del quale far aprire un determinato punto di menu (ad esempio, se ci sono state modifiche sull'anagrafica dei clienti posso portare l'utente direttamente sullo sprix mobile che gestisce i clienti).</p> <p>Dato che la sincronizzazione può coinvolgere molti record, le variabili rilasciate (a parte codice e versione) saranno tutte di tipo Array Le variabili di struttura gestite da questo evento sono le seguenti: _WCCOD\$: Codice dell'app _WCVER\$: Versione dell'app _WCNUM - numero record aggiornati (identifica la dimensione degli array) _WCTYPEOP\$: Array con il tipo di sincronizzazione effettuata.</p> <p>Può assumere i seguenti valori: "VAR": modifica di un record "DEL": cancellazione di un record "NEW": inserimento di un nuovo record da mobile "INS": inserimento di un nuovo record dal server _WCTYPEKEY\$: Array con il tipo archivio coinvolto nell'aggiornamento (es. CLFO,ARTI,etc) _WCKEY\$: Array con le chiavi dei record coinvolti nell'aggiornamento _WCKEYNEW\$() – Array con l'eventuale nuova chiave del record _ERRWC\$() – Array con eventuali errori di sincronizzazione del particolare record</p> <p>Oltre a queste variabili, che sono disponibili solo in lettura, è possibile definire dentro l'evento ON_POST_SYNC_APP anche due variabili di struttura per la definizione di un pulsante: _WCBTNVALUE\$ - etichetta pulsante che verrà visualizzato nella finestra di esito della sincronizzazione _WCOPENMENU\$ - nome sprix-mobile che verrà eseguito quando si preme sul pulsante (il formato è il medesimo dell'istruzione WCALL OPENMENU)</p>	Ca

Per altre informazioni non ancora riportate in questa tabella, si rimanda al capitolo finale "ChangeLog" del Manuale Sprix Mobile

esempi e casi

Esempio

```

ON_START_APP: `inizio App
  _ERRWF = 0 : _ERRWF$ = "" : AZZVARSYS 38
  _WFTITLE$ = "Form e lista Articoli" ` Titolo Form
  WCREATEFORM "FORML1" : IDF_FORML1 = _WFOID ` Creazione Form
  IFF _ERRWF <> 0
    VIMSG 4, "Errore creazione lista Form1 " + _ERRWF$
  ELSEF
    `campi maschera
    _ERRWI$ = "" : _ERRWI = 0 : AZZVARSYS 41
    GOSUB FZ_LISTAFORM1 ` Creazione lista
    _WLISTID = ID LISTFORM1 ` collegamento input a lista
    _WLISTNFD = 2 ` colonna da prelevare in fase di selezione
    _WITYPE$ = "TESTO"
    _WIPARENTID = IDF_FORML1
    _WIDES$ = "Descrizione articolo"
    _WIHINT$ = "<Seleziona articolo>"
    _WIOUTONLY$ = "N"
    _WIVALUE$ = INPUTF1$
    WCREATEINPUT "INPUTF1" : ID_INPUTF1 = _WIOID ` creazione input
    IF (_ERRWI <> 0) THEN VIMSG 4, "Errore creazione input" + _ERRWI$
    WSHOWFORM ID_FORML1
  ENDF
END
FZ_LISTAFORM1:
`Es. lista collegata ad archivio

```

```

_ERRWL$ = "" : _ERRWL = 0 : AZZVARSYS 42 `categoria _WL
_WLPARENTID = IDF_FORM1 ` riferimento id finestra
_WLPARENTZNS$ = "INPUT" `se legato all'input altrimenti ""
_WLARCNAME$ = "ARTI" `codice archivio altrimenti ""
_WLTOTFLD = 2 `numero colonne
_WLARCFLD$ (1) = ARCOD$ `codice campo archivio prima colonna.
_WLARCFLD$ (2) = ARDES$ + " " + ARAGG$ `codice campo archivio seconda colonna. Può essere composto
_WLDESFLD$ (1) = "Codice articoli" `descrizione prima colonna. Può essere vuota
_WLDESFLD$ (2) = "Descrizione" `descrizione seconda colonna. Può essere vuota
WCREATELIST "Listall"
ID LISTFORM1 = WLOID
IF _ERRWL <> 0 THEN VIMSG 4, "Errore creazione lista " + _ERRWL$
RETURN

```

Esempio:

```

_ERRWF = 0 : _ERRWF$ = "" : AZZVARSYS 38
_WFTITLE$ = "Form e lista Articoli" ` titolo Form
WCREATEFORM "FORM1" : ` preparazione Form
ID FORM1 = WFOID
IFF _ERRWF <> 0
    VIMSG 4, "Errore creazione lista Form1 " + _ERRWF$
ELSEF
    ... Istruzioni input, azioni, pulsanti, ecc
    WSHOWFORM ID_FORM1 ` disegno della finestra
ENDF

```

Esempio di input di tipo pulsante ACTION:

```

_WITYPE$ = "BUTTON" - Tipologia dell'input
_WIPARENTID = ID_FORM - Numero del Form che conterrà l'input
_WIPARENTZNS$ = "ACTION" - Viene disegnato nella toolbar in alto a destra
_WIVALUE$ = "Salva" - Stringa visualizzata sul pulsante

```

Esempio di input di tipo pulsante:

```

_WITYPE$ = "BUTTON" - Tipologia dell'input
_WIPARENTID = ID_FORM - Numero del Form che conterrà l'input
_WIVALUE$ = "Salva" - Stringa visualizzata sul pulsante

```

Esempio di input di tipo testo collegato ad una lista

```

_WITYPE$ = "TEXT" - Tipologia del campo
_WIPARENTID = ID_FORM - Numero del Form che conterrà il campo
_WILISTID = ID_LISTA - Collegamento input a lista
_WILISTNPLD = 2 - Colonna da prelevare dalla lista in fase di selezione
_WIDES$ = "Nome" - Etichetta del campo
_WIHINT$ = "<Nome>" - Placeholder. Testo visualizzato in grigio dentro il campo per aiutare l'utente nella valorizzazione
_WIOUTONLY$ = "" - Indica se il campo deve editabile (Input) oppure no (Output)
_WIVALUE$ = " " - Valore di default da assegnare al campo

```

Esempio di input di tipo testo

```

_WITYPE$ = "TEXT" - Tipologia del campo
_WIPARENTID = IDF_FORM1 - Numero del Form che conterrà il campo
_WIDES$ = "Descrizione articolo" - Etichetta del campo
_WIHINT$ = "<Seleziona articolo>" - Placeholder. Testo in grigio dentro il campo per aiutare l'utente nella valorizzazione
_WIOUTONLY$ = "N" - Indica se il campo è editabile: ' ' o 'S' (Input) 'N' (Output)
_WIVALUE$ = INPUTF1$ - Valore di default da assegnare al campo

```

Esempio di input di tipo data

```

_WITYPE$ = "DATE" - Tipologia del campo
_WIPARENTID = IDF_FORM - Numero del Form che conterrà il campo
_WIDES$ = "Data" - Etichetta del campo
_WIHINT$ = "<Seleziona data>" - Placeholder. Testo in grigio all'interno del campo per aiutare l'utente nella valorizzazione
_WIOUTONLY$ = "" - Indica se il campo è editabile: ' ' o 'S' (Input) 'N' (Output)
_WIVALUE$ = " " - Valore di default da assegnare al campo

```

Esempio di lista Estesa:

```

_ERRWL$ = "" : _ERRWL = 0 : AZZVARSYS 42
_WLPARENTID = ID_FORM
_WLPARENTZNS$ = "INPUT" o "ACTION"
_WLARCNAME$ = "CLFO" ` Clienti/Fornitori

_WLTOTFLDEXT = 2 ` Numero di campi personalizzati
_WLNMFLDEXT$ (1) = "ragclfo"
_WLNMFLDEXT$ (2) = "indclfo" `indirizzo cliente/fornitore
_WLTPFLDEXT$ (1) = "STRINGA" ` tipo campo personalizzato
_WLTPFLDEXT$ (2) = "STRINGA"
_WLORDDR$ (1) = 1 `specifica che l'ordinamento sarà impostato per "ragclfo"
_WLORDDR$ (1) = "A" `ordinamento di tipo ascendente
_WLTOTFLD = 2
_WLARCFLD$ (1) = " _WLNMFLDEXT$ (1) " `CKY_CNT"
_WLARCFLD$ (2) = " _WLNMFLDEXT$ (2) " `CDS_DESCTO"
_WLDESFLD$ (1) = "Ragione Sociale" ` "Cliente"
_WLDESFLD$ (2) = "Indirizzo" ` "Indirizzo"

IF WLCODICELCF$ = "" THEN WCREATELIST " LISTAMAINAPPCF"
ID_LISTCF = WLOID
IF ( _ERRWL <> 0) THEN VIMSG 4, "Errore creazione lista: " + _ERRWL$

ON_ROW_LISTAMAINAPPCF:
_GETROWVALFLD "_PCCOD$",TMP$
_PCCOD$ = TMP$
_GETROWVALFLD "_PCDESS$",TMP$
_PCDESS$ = TMP$
_GETROWVALFLD "_PCIND$",TMP$
_PCIND$ = TMP$
_GETROWVALFLD "_PCLOC$",TMP$

```

```

_PCLOC$ = TMP$
GETROWVALFLD "_PCPRO$",TMP$
_PCPRO$ = TMP$
GETROWVALFLD "_PCTEL$",TMP$
_PCTEL$ = TMP$
_WLVFLDEXT(1) = _PCDES$ + " - " + _PCCOD$
_WLVFLDEXT(2) = _PCIND$ + " , " + _PCLOC$ + " (" + _PCPRO$ + " ) , Tel. " + _PCTEL$ END
END
ON ROWFILTER LISTAMAINAPPCF:
GETROWVALFLD "_PCPF$",TMP$
IFF TMP$ = "S"
_WLROWOK$ = "S"
ELSEF
_WLROWOK$ = "N"
ENDIF
END

```

Esempio di lista dinamica:

```

_WLPARENTID = ID_FORM - ID del Form che conterrà lista.
_WLTOTROW = 10 - Numero di righe della lista
_WLTOTFLD = 2 - Numero di campi di una riga della lista
_WLDESFLD(1) = "Articolo" - Nome di un campo di una riga della lista
_WLDESFLD(2) = "Descrizione" - Nome di un campo di una riga della lista
WCREATELIST "MYLISTDIN"
ID_LISTDIN = _WLOID
IF _ERRWL <> 0 THEN PRINT _ERRWL$
ON ROW MYLISTDIN:
CURRENT_ROW = _WLNUMROW - _WLNUMROW indice della riga corrente
_WLVALFLD(1) = _MMART$( _WLNUMROW ) -- Valore del primo campo della riga
_WLVALFLD(2) = _ARDES$ -- Valore del secondo campo della riga
END

```

Esempio di lista da Archivio:

```

_WLPARENTID = ID_FORM - ID del Form che conterrà lista.
_WLARCNAME$ = "CLFO" " - Nome dell'archivio da visualizzare sulla lista
_WLTOTFLD = 2 - Numero di campi di una riga della lista
_WLARCFLD(1) = "_PCCOD$" - Nome del campo archivio che si vuole visualizzare
_WLARCFLD(2) = "_PCDES$" - Nome del campo archivio che si vuole visualizzare
_WLDESFLD(1) = "Codice Conto" - Descrizione del campo visualizzato
_WLDESFLD(2) = "Descrizione" - Descrizione del campo visualizzato
WCREATELIST "myListDin"
ID_LISTDIN = _WLOID
IF _ERRWL <> 0 THEN PRINT _ERRWL$

```

Esempio di lista archivio collegata ad input:

```

_WLPARENTID = ID_FORM - ID del Form che conterrà lista.
_WLPARENTZNS = "INPUT" - ACTION significa che la lista è collegata ad un input
_WLARCNAME$ = "CLFO" " - Nome dell'archivio da visualizzare sulla lista
_WLTOTFLD = 2 - Numero di campi di una riga della lista
_WLARCFLD(1) = "_PCCOD$" - Dati dell'archivio da visualizzare nel campo 1
_WLARCFLD(2) = "_PCDES$" + "_PCNAZ$" - Dati dell'archivio da visualizzare nel campo 2
_WLDESFLD(1) = "Codice" - Nome di un campo di una riga della lista
_WLDESFLD(2) = "Descrizione" - Nome di un campo di una riga della lista
WCREATELIST "MYLISTARC"
ID_LISTARC = _WLOID
IF _ERRWL <> 0 THEN PRINT _ERRWL$

```

```

Campo di input in cui la lista sarà visualizzata.
_WITYPE$ = "TEXT" - Tipo di input
_WIPARENTID = ID_FORM - ID del Form che conterrà l'input
_WIDES$ = "Clienti/Fornitori" - Etichetta descrittiva del campo
_WIOUTONLY$ = "N" - N specifica che il campo è editabile
_WILISTID = ID_LISTARC - ID lista collegata all'archivio
_WILISTNFLD = 1 - Campo da visualizzare nell'input dopo aver scelto una riga dalla lista associata
WCREATEINPUT "myInputTesto" 1 - myInputTesto è l'ID scelto dall'utente. Utile a gestire gli eventi
ID_TINPUT = _WIOID
IF _ERRWI <> 0 THEN PRINT _ERRWI$

```

Esempio WCREATEFORM

```

_ERRWF = 0 : _ERRWF$ = "" : AZZVARSYS 38
_WFTITLE$ = "Form e lista Articoli" ' Titolo Form
WCREATEFORM "FORM1" : ' Creazione Form
ID_FORM1 = _WFOID
IFF _ERRWF <> 0
_VIMSG 4, "Errore creazione lista Form1 " + _ERRWF$
ENDIF

```

Esempio WSHOWFORM

```

_ERRWF = 0 : _ERRWF$ = "" : AZZVARSYS 38
_WFTITLE$ = "Form e lista Articoli" ' titolo Form
WCREATEFORM "FORM1" : ' preparazione Form
ID_FORM1 = _WFOID
IFF _ERRWF <> 0
_VIMSG 4, "Errore creazione lista Form1 " + _ERRWF$
ELSEF
... Istruzioni input, azioni, pulsanti, ecc
_WSHOWFORM ID_FORM1 ' disegno della finestra
ENDIF

```

Esempi WSET

```

WSET PROPRIETA ALFANUMERICA
_WOPNAME$ = "_WIVALUE$" - Proprietà da impostare
_WOPVAL$ = VAL$ - Valore Proprietà impostata
WSET IDOBJ - ID componente da impostare

WSET PROPRIETA NUMERICA
_WOPNAME$ = "_WILISTNFLD" - Proprietà da impostare
_WOPVAL = VAL - Valore Proprietà impostata
WSET IDOBJ - ID componente da impostare

```

Esempi WGET

```

WGET PROPRIETA ALFANUMERICA
_WOPNAME$ = "_WIVALUE$" - Proprietà da ottenere
WGET IDOBJ - ID componente da leggere
VAL$ = _WOPVAL$ - Valore Proprietà ottenuta

WGET PROPRIETA NUMERICA
_WOPNAME$ = "_WILENCAR$" - Proprietà da ottenere
WGET IDOBJ - ID componente da leggere
VAL = _WOPVAL - Valore Proprietà ottenuta

```

Esempio WGETOID Collage mobile

```

ON_IN INCT:
WGETOID "INCT" 'etichetta che scatta sul controllo INCT (in questo caso INPUT di test)
ID_IN = _WOPVAL 'controllo di cui si vuole ottenere l'ID
_WOPNAME$ = "_WIVALUE$" 'recupero il testo del campo di input
WGET ID_IN
VIMSG 1,"Entro - testo = " + _WOPVAL$ 'mostro a video il testo del campo di input
END

ON_OUT INCT:
VIMSG 1,"Esco"
_WOPNAME$ = "_WIVALUE$"
_WOPVAL$ = "Uscito"
WSET ID_IN ' Posso fare una WSET utilizzando l'ID letto in precedenza
END

```

Esempio WCREATEDIALOG

```

EA_DIALOG_03:
_ERRWD$="" : _ERRWD=0:
_WDTITLE$ = "Titolo della Dialog 03" ' Titolo della Dialog
_WDMESSAGES$ = "Messaggio della Dialog 03 " +CHR$(10)+"Riga02"+CHR$(10)+"Riga3"+CHR$(10)+"Riga4"+CHR$(10)+"Riga5" 'Mess. Dialog
WCREATEDIALOG "MiaDialog03"
IFF (_ERRWD <> 0) ' _ERRWD errore Dialog
VIMSG 4, "Errore creazione form: " + _ERRWD$
ENDF
IDD_03 = _WDOID

'TASTO BUTTON 01
'Prepara il pulsante 01
_ERRWI$ = "" : _ERRWI = 0 : AZZVARSYS 41
_WITYPE$ = "BUTTON" ' BUTTON/TEXT Solo BUTTON errore Text
_WIPARENTID = IDD_03
_WIPARENTZN$ = "" ' ACTION o Space indifferente
_WIDE$ = "Descr" 'Indifferente
_WIVALUE$ = "01"
WCREATEINPUT "CMD01"
IF (_ERRWI <> 0) THEN VIMSG 4, "Errore creazione pulsante Action salva: " + _ERRWI$
ID_ACT01 = _WIOID

'TASTO BUTTON 02
'Prepara il pulsante 02
_ERRWI$ = "" : _ERRWI = 0 : AZZVARSYS 41
_WITYPE$ = "BUTTON"
_WIPARENTID = IDD_03
_WIPARENTZN$ = "ACTION"
_WIVALUE$ = "02"
WCREATEINPUT "CMD02"
IF (_ERRWI <> 0) THEN VIMSG 4, "Errore creazione pulsante Action salva: " + _ERRWI$
ID_ACT02 = _WIOID

'TASTO BUTTON 03
'Prepara il pulsante 03
_ERRWI$ = "" : _ERRWI = 0 : AZZVARSYS 41
_WITYPE$ = "BUTTON"
_WIPARENTID = IDD_03
_WIPARENTZN$ = "ACTION"
_WIVALUE$ = "03"
WCREATEINPUT "CMD03"
IF (_ERRWI <> 0) THEN VIMSG 4, "Errore creazione pulsante Action salva: " + _ERRWI$
ID_ACT03 = _WIOID

WSHOWDIALOG IDD_03

RETURN

```

Esempio 1 SRVCALL: lettura progressivi di un articolo

```

##### SPRIX MOBILE #####

ON_START SPRIX:
CODART$ = "00000A01"
PUTREM STR "COD DA LEGGERE", CODART$
SRVCALL "colsrl", "GETAR_REMOTA", "LEGGI_GETAR_REMOTA"
'Qui ancora non hai la certezza del dato elaborato. Il dato deve essere gestito in GEST_PROGRESSIVI:

END
-----
LEGGI_GETAR_REMOTA:

PRINT " ERRSRV$ = " + _ERRSRV$
PRINT " ERRGETAR$ = " + _ERRGETAR$

IFF _ERRSRV$ = ""
GETREM_STR "ERRGETAR", _ERRGETAR$

IFF _ERRGETAR$ = ""
GETREM_NUM "ARINQ" , ARINQ
DIM ARINT(16) : GETREM_ARRAY "ARINT" , ARINT

```

```

DIM ARINTXL(16) : GETREM_ARRAY "ARINTXL" , ARINTXL
                  GETREM_NUM  "ARINV" , ARINV
                  GETREM_NUM  "ARTIN" , ARTIN
                  GETREM_NUM  "ARCIN" , ARCIN
                  GETREM_NUM  "ARCAR" , ARCAR
DIM ARCAT(16)  : GETREM_ARRAY "ARCAT" , ARCAT
DIM ARCATXL(16) : GETREM_ARRAY "ARCATXL" , ARCATXL
                  GETREM_NUM  "ARCAV" , ARCAV
                  GETREM_NUM  "ARTCA" , ARTCA
                  GETREM_NUM  "ARCCA" , ARCCA
                  GETREM_NUM  "ARSCA" , ARSCA
DIM ARSCT(16)  : GETREM_ARRAY "ARSCT" , ARSCT
DIM ARSCTXL(16) : GETREM_ARRAY "ARSCTXL" , ARSCTXL
                  GETREM_NUM  "ARSCV" , ARSCV
                  GETREM_NUM  "ARTSC" , ARTSC
                  GETREM_NUM  "ARCSC" , ARCSC

GOSUB GEST_PROGRESSIVI
ELSEF
ERRSTR$ = "Errore GETAR: " + ERRGETAR$
GOSUB GEST_ERRORE_GETARREM
ENDF
ELSEF
ERRSTR$ = "Errore chiamata remota: " + _ERRSRV$
GOSUB GEST_ERRORE_GETARREM
ENDF
END

GEST_PROGRESSIVI:
\...
'Qui gestisco i progressivi recuperati da remoto
\...

RETURN

GEST_ERRORE_GETARREM:
\...
'Qui gestisco l'errore tornato dalla chiamata remota
\...
RETURN

##### COLLAGE SERVER REMOTO (colsr1) #####

GETAR_REMOTA:
GETREM_STR "COD_DA_LEGGERE", CODART$

GETAR CODART$, 0

PUTREM_STR "ERRGETAR", _ERRAR$

IFF _ERRAR$ = ""

PUTREM_NUM "ARINQ" , _ARINQ
PUTREM_ARRAY "ARINT" , _ARINT
PUTREM_ARRAY "ARINTXL" , _ARINTXL
PUTREM_NUM "ARINV" , _ARINV
PUTREM_NUM "ARTIN" , _ARTIN
PUTREM_NUM "ARCIN" , _ARCIN
PUTREM_NUM "ARCAR" , _ARCAR
PUTREM_ARRAY "ARCAT" , _ARCAT
PUTREM_ARRAY "ARCATXL" , _ARCATXL
PUTREM_NUM "ARCAV" , _ARCAV
PUTREM_NUM "ARTCA" , _ARTCA
PUTREM_NUM "ARCCA" , _ARCCA
PUTREM_NUM "ARSCA" , _ARSCA
PUTREM_ARRAY "ARSCT" , _ARSCT
PUTREM_ARRAY "ARSCTXL" , _ARSCTXL
PUTREM_NUM "ARSCV" , _ARSCV
PUTREM_NUM "ARTSC" , _ARTSC
PUTREM_NUM "ARCSC" , _ARCSC
ENDF
END

```

Esempio 2 SRVCALL: test per ricezione ed apertura su mobile di un file.

```

ON_START SPRIX:
FILE_DA_R$ = "PROVA.html"
PUTREM_STR "FILE_SCAMBIO", FILE_DA_R$
IFF ERRPUTREM$ <> ""
VIMSG 5, "ERRORE PUTREM :"+ERRPUTREM$
ELSEF
SRVCALL "COLSR2", "INVIO_FILE", "RICEZ_FILE"
ENDF
END
\-----
RICEZ_FILE:
IFF _ERRSRV$ = ""
GETREM_FILE "FILE_SCAMBIO" , FILE_DA_R$ 'saldo contabile del cliente/fornitore
IFF _ERRGETREM$ = ""
VISUALIZZAFI 1, FILE_DA_R$
IF _ERRFILE$<>"" THEN VIMSG 5, "Viusalizza file :"+_ERRFILE$
ELSEF
VIMSG 5, "ERRORE GETREM : "+_ERRGETREM$
ENDF
ELSEF
'Errore chiamata remota
VIMSG 5, "ERRORE SERVER : "+_ERRSRV$
ENDF
END
\-----
'#####
'# COLLAGE SERVER REMOTO
'#####
INVIO_FILE:
GETREM_STR "FILE_SCAMBIO" , FILE_S$

```

```
PUTREM_FILE "FILE_SCAMBIO", "C:\TEMP\"+FILE_SS
END
```

Esempio SRVSYNCDOC:

```
SRVSYNCDOC _MMSIG$+STR$(_MMSER,0,0)+"/"+STR$(_MMNUM,0,0), "", "ON_SRVSYNCDOC"

ON_SRVSYNCDOC:
  IFF_ERRSSD$ = ""
  VISUALIZZAFI 1, _SSDFILESTP$
  IF _ERRFILE$ <> "" THEN VIMSG 1, "Errore visualizzazione documento: |" + _ERRFILE$
  ELSEF
    VIMSG 1, "Errore lettura documento: |" + _ERRSSD$
  ENDF
END
```

EsempioSalvataggio parametri in aree di memoria

```
PUTLOC_STR "T", "Cliente", CLI$ (Temporaneo)
GETLOC_STR "T", "Cliente", OLDCLI$
PUTLOC_NUM "P", "Operazioni", NUMOP (Persistente)
STORELOC "P", salva in modo persistente i dati in memoria.
```

Esempio: utilizzo APRISTAMP in SRVCALL (Collage Server Remoto) chiamata dalle Form di Sprix Mobile

Dato il seguente Sprix Mobile "spxmob.ss" che utilizza il seguente "colsrvrm.ss", l'applicazione Mobile esegue verifiche su codice articolo, variazione della descrizione e stampa su dispositivo collegato al Server. Il seguente codice dei due sorgenti è un esempio, pertanto non è completo.

SPXMOB.SS

```
-----
ON_START_SPRIX: 'evento << ingresso nella voce di menu >>
-----
'
'
'-----
'FORMATO n.1 - richiesta del Codice Articolo
'-----
AZZCATVARSYS "WF" 'Inizializzo le variabili di struttura WFORM
_WFTITLE$="titolo FINESTRA della FORM 1" 'Inizializzo il Titolo della Finestra
_WFHDRTXT$="descrizione della TESTATA della FORM 1" 'Inizializzo il Testo dell'Header
_WFFTRTXT$="descrizione del PIEDE della FORM 1" 'Inizializzo il Testo del Footer
_WCREATEFORM "FORM 1" 'Creo il FORMATO n.1
IF _ERRWF<>0 THEN VIMSG 1, _ERRWF$ : END 'Termino la routine in caso di errore
ID_FORM1 = _WFOID 'Salvo lo ID della Finestra
'-----
'CAMPO INPUT N.1 - per inserire il Codice Articolo
'-----
AZZCATVARSYS "WI" 'Inizializzo le variabili di struttura WINPUT
_WITYPE$="TEXT" 'Definisco che il campo di Input n.1 e' di tipo Testo
_WIPARENTID=ID_FORM1 'Indico in quale form va inserito il campo di Input n.1
_WIVALUE$="inserire articolo" 'Valore iniziale del campo di Input n.1
_WIHINT$="inserire articolo" 'Valore iniziale della stringa testo grigia
_WIDES$="descrizione del CAMPO della FORM 1" 'Descrizione del campo
_WIOUTONLY$="N" 'Inizializzo campo di NON solo Output
_WILENCAR=20 'Dimensione max in Input (max 20 caratteri)
_WILINE$="" 'Imposta vuoto nell'Input
_WCREATEINPUT "INP 1" 'Creo Input n.1
IF _ERRWI<>0 THEN VIMSG 1, _ERRWI$ : END 'Termino la routine in caso di errore
ID_INP1= _WIOID 'Salvo lo ID del campo di Input n.1
'-----
'FORMATO n.2 - richiesta nuova Descrizione
'-----
AZZCATVARSYS "WF" 'Inizializzo le variabili di struttura WFORM
_WFTITLE$="titolo FINESTRA della FORM 2" 'Inizializzo il Titolo della Finestra
_WFHDRTXT$="descrizione della TESTATA della FORM 2" 'Inizializzo il Testo dell'Header
_WFFTRTXT$="descrizione del PIEDE della FORM 2" 'Inizializzo il Testo del Footer
_WCREATEFORM "FORM 2" 'Creo il FORMATO n.2
IF _ERRWF<>0 THEN VIMSG 1, _ERRWF$ : END 'Termino la routine in caso di errore
ID_FORM2 = _WFOID 'Salvo lo ID della Finestra
'-----
'CAMPO INPUT N.2 - inserimento nuova Descrizione
'-----
AZZCATVARSYS "WI" 'Inizializzo le variabili di struttura WINPUT
_WITYPE$="TEXT" 'Definisco che il campo di Input n.2 e' di tipo Testo
_WIPARENTID=ID_FORM2 'Indico in quale form va inserito il campo di Input n.2
_WIVALUE$="inserire nuova descrizione" 'Valore iniziale del campo di Input n.2
_WIHINT$="inserire nuova descrizione" 'Valore iniziale della stringa testo grigia
_WIDES$="descrizione del CAMPO della FORM 2" 'Descrizione del campo
_WIOUTONLY$="N" 'Inizializzo campo di NON solo Output
_WILENCAR=30 'Dimensione max in Input (30 caratteri)
_WILINE$="" 'Imposta vuoto nell'Input
_WCREATEINPUT "INP 2" 'Creo INPUT n.2
IF _ERRWI<>0 THEN VIMSG 1, _ERRWI$ : END 'Termino la routine in caso di errore
ID_INP2= _WIOID 'Salvo lo ID del campo di Input n.2
'-----
'+-----+
'|PULSANTI DELLE DUE FORM|
'+-----+
'
'-----
'PULSANTE N.1 - Varia Descrizione Articolo
'-----
AZZCATVARSYS "WI" 'Inizializzo le variabili di struttura WINPUT
_WITYPE$="BUTTON" 'Definisco che il campo di Input e' di tipo Pulsante
_WIPARENTID=ID_FORM1 'Associo il Pulsante n.1 al FORM 1
_WIVALUE$="VARIA DESCRIZIONE" 'Descrizione del Pulsante
_WICO$="3" 'Icona del campo di tipo "Modifica"
_WCREATEINPUT "PULS 1" 'Creo il Pulsante n.1
IF _ERRWI<>0 THEN VIMSG 1, _ERRWI$ : END 'Termino la routine in caso di errore
ID_BUT1= _WIOID 'Salvo lo ID del campo Pulsante n.1
'-----
'PULSANTE N.2 - Stampa Descrizione Articolo
```

```

-----
AZZCATVARSYS "WI"
  _WITYPES="BUTTON"
  _WIPARENTID=ID_FORM1
  _WIVALUES="STAMPA DESCRIZIONE"
  _WIICO$="7"
WCREATEINPUT "PULS 2"
  IF _ERRWI<>0 THEN VIMSG 1, _ERRWI$ : END
  ID_BUT2 = _WIOID
-----
'PULSANTE N.3 - Annulla in FORM 1
-----
AZZCATVARSYS "WI"
  _WITYPES="BUTTON"
  _WIPARENTID=ID_FORM1
  _WIVALUES="ANNULLA"
  _WIICO$="2"
WCREATEINPUT "PULS 3"
  IF _ERRWI<>0 THEN VIMSG 1, _ERRWI$ : END
  ID_BUT3 = _WIOID
-----
'PULSANTE N.4 - Conferma Variazione Descrizione
-----
AZZCATVARSYS "WI"
  _WITYPES="BUTTON"
  _WIPARENTID=ID_FORM2
  _WIVALUES="CONFERMA VARIAZIONE"
  _WIICO$="1"
WCREATEINPUT "PULS 4"
  IF _ERRWI<>0 THEN VIMSG 1, _ERRWI$ : END
  ID_BUT4 = _WIOID
-----
'PULSANTE N.5 - Stampa Descrizione Articolo
-----
AZZCATVARSYS "WI"
  _WITYPES="BUTTON"
  _WIPARENTID=ID_FORM2
  _WIVALUES="STAMPA DESCRIZIONE"
  _WIICO$="7"
WCREATEINPUT "PULS 5"
  IF _ERRWI<>0 THEN VIMSG 1, _ERRWI$ : END
  ID_BUT5 = _WIOID
-----
'PULSANTE N.6 - Annulla in FORM 2
-----
AZZCATVARSYS "WI"
  _WITYPES="BUTTON"
  _WIPARENTID=ID_FORM2
  _WIVALUES="ANNULLA"
  _WIICO$="2"
WCREATEINPUT "PULS 6"
  IF _ERRWI<>0 THEN VIMSG 1, _ERRWI$ : END
  ID_BUT6 = _WIOID
-----
WSHOWFORM ID_FORM1
END
-----
ATTIVAZIONE CHIAMATE SU EVENTO/PULSANTE
-----
ON OUT_INP 1:
  'ON_PRESS_PULS 1:
  'ON_PRESS_PULS 2:
  _WOPNAME$=" _WIVALUES$"
  WGET ID_INP1
  VAL_INP1$= WOPVAL$
  PUTREM STR "GET COD",VAL_INP1$
  SRVCALL "ColSrvRm","EXE1","VALORI_DI_RITORNO_GET"
END
-----
ON PRESS CMD01:
  IF MIOERR_GET$="Articolo NON Esistente! Ripetere l'Immissione!" THEN _TF=0 : END ' in base all'esito della lettura
  WCALL "CLOSEFORM", ID_FORM1
  WSHOWFORM ID_FORM2
  VIMSG 1, "Apro il FORM 2 per inserire una nuova Descrizione|oppure per Stampare" 'Messaggio
END
-----
'ON_PRESS_PULS 1:
' IF MIOERR_GET$="Articolo NON Esistente! Ripetere l'Immissione!" THEN _TF=0 : END
'END
-----
ON PRESS_PULS 2:
  IF MIOERR_GET$="Articolo NON Esistente! Ripetere l'Immissione!" THEN END ' _TF=0
  _WOPNAME$=" _WIVALUES$"
  WGET ID_INP1
  VAL_INP1$= WOPVAL$
  PUTREM STR "STP DES",VAL_INP1$
  SRVCALL "ColSrvRm","EXE3","VALORI_DI_RITORNO_STP"
  END
-----
ON_PRESS_PULS 3:
  WCALL "CLOSEFORM",ID_FORM1
  VIMSG 1, "Sono uscito dalle routine"
  END
-----
ON_OUT_INP 2:
  _TF=0
  END
-----
ON_PRESS_PULS 4:
  _WOPNAME$=" _WIVALUES$"
  WGET ID_INP2
  VAL_INP2$= WOPVAL$
  PUTREM STR "VAR DES",VAL_INP2$
  SRVCALL "ColSrvRm","EXE2","VALORI_DI_RITORNO_VAR"
  END
-----
ON_PRESS_PULS 5:

```

```

_WOPNAME$=" _WIVALUE$" 'Proprieta' da leggere
WGET ID INPI 'ID casella di testo
VAL_INPI$= WOPVAL$ 'Salvo il valore di _WOPVAL$
PUTREM STR "STP DES",VAL INPI$ 'Gestione stringhe da Mobile a Server (per Stampa)
SRVCALL "ColSrvRm","EXE3","VALORI_DI_RITORNO_STP" 'Chiamata CollageServerRemoto
END 'Termino la routine

```

'Nota: al secondo pulsante di stampa, il valore VAL_INPI viene azzerato e resta solo la descrizione

```

-----
ON PRESS PULS 6: 'Evento Pulsante 6 (Annulla)
WCALL "CLOSEFORM",ID_FORM2 'Chiudo la FORM 1
VIMSG 1, "Sono uscito dalle routine" 'Messaggio
END 'Termino la routine
-----

```

'GESTIONE FINESTRA DI DIALOGO - PER VISUALIZZARE MIOERR_GET\$

```

-----
DIALOG_GET: 'visualizza MIOERR_GET$ per la conferma
ERRWD$="" : ERRWD=0
_WDTITLE$="Titolo della DIALOG relativa alla fase di GET" 'Titolo della Dialog
_WDMESSAGE$=MIOERR_GET$ ' Messaggio Dialog della GET
WCREATEDIALOG "MiaDialogGet"
IDD_DIAL_GET= WDOID 'Numero univoco della finestra
'Preparazione di un pulsante BUTTON "Ok"
ERRWI$="" : ERRWI=0 : AZZVARSYS 41
_WITYPE$="BUTTON" ' BUTTON/TEXT Solo BUTTON errore Text
_WIPARENTID=IDD_DIAL_GET
'_WIPARENTZN$="" ' ACTION o Space indifferente
'_WIDES$="Descr" 'Indifferente
_WIVALUE$="Ok"
WCREATEINPUT "CMD01"
IF ( _ERRWI<>0) THEN VIMSG 4, "Errore creazione pulsante Action: " + _ERRWI$
WSHOWDIALOG IDD_DIAL_GET
RETURN
-----

```

'GESTIONE FINESTRA DI DIALOGO - PER VISUALIZZARE MIOERR_VAR\$

```

-----
DIALOG_VAR: 'visualizza MIOERR_GET$ per la conferma
ERRWD$="" : ERRWD=0
_WDTITLE$="Titolo della DIALOG relativa alla fase di VAR" 'Titolo della Dialog
_WDMESSAGE$=MIOERR_VAR$ ' Messaggio Dialog della VAR
WCREATEDIALOG "MiaDialogVar"
IDD_DIAL_VAR= WDOID 'Numero univoco della finestra
'Preparazione di un pulsante BUTTON "Ok"
ERRWI$="" : ERRWI=0 : AZZVARSYS 41
_WITYPE$="BUTTON" ' BUTTON/TEXT Solo BUTTON errore Text
_WIPARENTID=IDD_DIAL_VAR
'_WIPARENTZN$="" ' ACTION o Space indifferente
'_WIDES$="Descr" 'Indifferente
_WIVALUE$="Ok"
WCREATEINPUT "CMD02"
IF ( _ERRWI<>0) THEN VIMSG 4, "Errore creazione pulsante Action: " + _ERRWI$
WSHOWDIALOG IDD_DIAL_VAR
RETURN
-----

```

'GESTIONE FINESTRA DI DIALOGO - PER VISUALIZZARE MIOERR_STP\$

```

-----
DIALOG_STP: 'visualizza MIOERR_GET$ per la conferma
ERRWD$="" : ERRWD=0
_WDTITLE$="Titolo della DIALOG relativa alla fase di STP" 'Titolo della Dialog
'_WDMESSAGE$=MIOERR_STP$ ' Messaggio Dialog della STP
'_WDMESSAGE$="Controllare l'esito della stampa" ' Messaggio Dialog della STP
WCREATEDIALOG "MiaDialogStp"
IDD_DIAL_STP= WDOID 'Numero univoco della finestra
'Preparazione di un pulsante BUTTON "Ok"
ERRWI$="" : ERRWI=0 : AZZVARSYS 41
_WITYPE$="BUTTON" ' BUTTON/TEXT Solo BUTTON errore Text
_WIPARENTID=IDD_DIAL_STP
'_WIPARENTZN$="" ' ACTION o Space indifferente
'_WIDES$="Descr" 'Indifferente
_WIVALUE$="Ok"
WCREATEINPUT "CMD03"
IF ( _ERRWI<>0) THEN VIMSG 4, "Errore creazione pulsante Action: " + _ERRWI$
WSHOWDIALOG IDD_DIAL_STP
RETURN
-----

```

GESTIONE ERRORI DELLA CHIAMATA REMOTA

NOTA BENE: il punto di ritorno e' il punto in cui settare i flag di controllo, condizioni e tutto cio' che dipende dalla CALL REMOTA!

```

-----
VALORI_DI_RITORNO_GET: 'Routine di ritorno
IF _ERRSRV<>0 THEN VIMSG 1, _ERRSRV$ : END 'Test errore call remota
GETREM STR "ERRORE GET", MIOERR_GET$ 'Leggo variabile errore
IF _ERRGETREM$<>" " THEN VIMSG 1, " _ERRGETREM$ = " + _ERRGETREM$ : END 'Test errori su GETREM
' NOTARE: non utilizzo la VIMSG altrimenti perdo il FUOCO e il CONTROLLO
' Quindi non utilizzo IF MIOERR_GET$<>"OK" THEN VIMSG 1, MIOERR_GET$
' Meglio invece se utilizzo la DIALOG
IF MIOERR_GET$<>"OK" THEN GOSUB DIALOG_GET 'Apertura messaggi DIALOG della lettura
END 'Termine routine
-----
VALORI_DI_RITORNO_VAR: 'Routine di ritorno
IF _ERRSRV<>0 THEN VIMSG 1, _ERRSRV$ : END 'Test errore call remota
GETREM STR "ERRORE VAR", MIOERR_VAR$ 'Leggo variabile errore
IF _ERRGETREM$<>" " THEN VIMSG 1, " _ERRGETREM$ = " + _ERRGETREM$ : END 'Test errori su GETREM
IF MIOERR_VAR$<>"OK" THEN GOSUB DIALOG_VAR 'Apertura messaggi DIALOG della variazione
END 'Termine routine
-----
VALORI_DI_RITORNO_STP: 'Routine di ritorno
IF _ERRSRV<>0 THEN VIMSG 1, _ERRSRV$ : END 'Test errore call remota
GETREM STR "ERRORE STP", MIOERR_STP$ 'Leggo variabile errore
IF _ERRGETREM$<>" " THEN VIMSG 1, " _ERRGETREM$ = " + _ERRGETREM$ : END 'Test errori su GETREM

```



```

IF MIOERR_STP$ <> "OK" THEN GOSUB DIALOG_STP 'Apertura messaggi DIALOG della stampa
END 'Termine routine
'-----
'-----
'-----
COLSRVRM.SS
'-----
'COLLAGE SERVER REMOTO
'-----
'In base alle selezioni nel dispositivo mobile "Client Sprix Mobile"
'esegue le seguenti operazioni sul SERVER tramite "Collage Server Remoto":
'
'Dopo l'inserimento del codice articolo nel FORM_1, all'uscita del campo di INPUT 1 oppure alla pressione del pulsante 1
'(VARIA DESCRIZIONE) avvia la routine di lettura controllo dell'esistenza e termina la procedura
EXE1: GOSUB GET_COD : END
'Dopo l'inserimento della descrizione nel FORM_2, alla pressione del pulsante 4 (CONFERMA VARIAZIONE) avvia la routine di
'variazione della descrizione e termina la procedura
EXE2: GOSUB VAR_DES : END
'Dopo l'inserimento del codice articolo nel FORM_1 e la pressione del pulsante 2 (STAMPA DESCRIZIONE) oppure alla pressione
'del pulsante 5 (STAMPA DESCRIZIONE) del FORM 2 avvia la routine di stampa del codice e descrizione e termina la procedura
EXE3: GOSUB STP_DES : END
'-----
'ROUTINE DI ELABORAZIONE
'-----
GET_COD: 'routine di lettura del codice articolo e controllo l'esistenza
MIOERR_GET$="OK" 'inizializzo una mia variabile di errore
'NOTARE: non essendo possibile eseguire debug in Collage Server Remoto e' molto utile gestire una propria
'variabile per per testare tutte gli esiti delle operazioni eseguite.
GETREM_STR "GET_COD",VAL_INP1$ 'ricevo il codice articolo dal Client Mobile
'se esiste errore di lettura stringa del mobile valorizza la mia variabile errore e termino la procedura
IF _ERRGETREM$ <> "" THEN MIOERR_GET$ = _ERRGETREM$ : GOSUB ERRORE_GET : END

'LETTURA
GETAR VAL_INP1$,0 'leggo il codice articolo contenuto in VAL_INP1$
'se esiste errore di lettura stringa su Server valorizza la mia variabile errore e termino la procedura
IF _ERRAR <> 0 THEN MIOERR_GET$ = "Articolo NON Esistente! Ripetere l'Immissione!": GOSUB ERRORE_GET : END
'se non esiste errore in lettura nella mia variabile informo dell'esito positivo dell'operazione
IF _ERRAR$="" THEN MIOERR_GET$ = "Articolo Esistente! Attuale Descrizione: "+_ARDES$: GOSUB ERRORE_GET
GOSUB ERRORE_GET
RETURN

VAR_DES: 'routine di variazione della descrizione del codice articolo
MIOERR_VAR$="OK" 'inizializzo una mia variabile di errore
GETREM_STR "VAR_DES",VAL_INP2$ 'ricevo il codice articolo dal Client Mobile
'se esiste errore di lettura stringa del mobile valorizza la mia variabile errore e termino la procedura
IF _ERRGETREM$ <> "" THEN MIOERR_VAR$ = _ERRGETREM$ : GOSUB ERRORE_VAR : END

'VARIAZIONE
_ARDES$=VAL_INP2$ 'inizializzo la variazione con la nuova descrizione che ho in VAL_INPUT2$
PUTAR 0 'registro la variazione del codice articolo nel Server
'se esiste errore in scrittura valorizza la mia variabile errore e termino la procedura
IF _ERRAR <> 0 THEN MIOERR_VAR$ = _ERRAR$ : GOSUB ERRORE_VAR : END
'se non esiste errore in scrittura nella mia variabile informo dell'esito positivo dell'operazione
IF _ERRAR$="" THEN MIOERR_VAR$ = "Descrizione Modificata in: " + VAL_INP2$: GOSUB ERRORE_VAR
GOSUB ERRORE_VAR
RETURN

STP_DES:
MIOERR_STP$="OK" 'inizializzo una mia variabile di errore
GETREM_STR "STP_DES",VAL_INP1$ 'ricevo il codice articolo dal Client Mobile
'se esiste errore valorizza la mia variabile di errore e termino la procedura
IF _ERRGETREM$ <> "" THEN MIOERR_STP$ = _ERRGETREM$ : GOSUB ERRORE_STP : END

'-----
'-----
'-----
ATTENZIONE! IN COLLAGE SERVER REMOTO, PER STAMPARE CON APRISTAMP o APRISTAMP EXT
E' INDISPENSABILE CHE NON ESISTANO INTERAZIONI I/O
'-----
'-----
Dalla versione 2020A (v750) l'istruzione è disponibile anche in Collage Server Remoto per stampe immediate su file e stampe fisiche,
ma esclusivamente in assenza di interazioni I/O e solo per stampe che non utilizzino la modulistica grafica. La stampa con dispositivo
collegato al Server è consentita con stampe immediate su file e fisiche configurate senza la richiesta conferma inizio stampa. Non
sono consentite tutte le condizioni che richiedano un I/O o che rilascino messaggi di qualsiasi forma e contenuto. Esempio: stampa
video, richiesta conferma stampante, scelta dispositivo e/o formato, invio mail in corso, mittente/destinatario, errori di
configurazione mail SMTP. In mancanza dei suddetti requisiti, sono provocati simili errori (con/senza chiusura immediata del Client
Mobile): "eccezione non gestita", "Impossibile accedere alla tastiera - Tastiera non disponibile in modalita senza interfaccia
utente", "Sprix "CollageServerRemoto" - Stampa non disponibile - Routine collage interrotta", ecc... Le combinazioni per l'utilizzo
dalla stampante variano in base al tipo di operazione ed il tipo di configurazione in uso, pertanto non sono elencabili in modo
esaustivo e occorre considerarle empiricamente, come anche le ipotesi di ristampa su file pdf nel momento in cui il medesimo file pdf
sia aperto in consultazione sul Server oppure l'uso di editor di stampa che richiedano esiti oppure conferme dell'elaborato (es. Adobe
PDF) oppure l'inceppamento/esaurimento della carta su una stampante fisica, in quanto si tratta di un evento non intercettato dal
Collage (salvo qualora sia possibile, per esempio, che la stampante sia successivamente in grado di comunicare il suddetto evento ad
un altro Collage che stia "in ascolto" sulla porta), e così via...

' APRISTAMP 73,"" 'apertura stampante Server fisica HP LJ400 formato Drv e richiesta inizio stampa = MAI
' APRISTAMP_EXT 73,"",79 'apertura stampante Server fisica HP LJ400 formato Drv e richiesta inizio stampa = MAI

' APRISTAMP 79,"" 'apertura stampante Server fisica HP LJ400 formato txt e richiesta inizio stampa = MAI
' APRISTAMP_EXT 79,"",79 'apertura stampante Server fisica HP LJ400 formato txt e richiesta inizio stampa = MAI

' APRISTAMP 118,"" 'apertura dispositivo file Server formato C:\Temp.filesilent.txt e richiesta inizio stampa = MAI
' APRISTAMP_EXT 118,"",79 'apertura dispositivo file Server formato C:\Temp.filesilent.txt e richiesta inizio stampa = MAI

APRISTAMP 121,"" 'apertura dispositivo file Server formato C:\Temp.filesilent.pdf e richiesta inizio stampa = MAI
' APRISTAMP_EXT 121,"",79 'apertura dispositivo file Server formato C:\Temp.filesilent.pdf e richiesta inizio stampa = MAI

LPRINT "Cod. Articolo = "+VAL_INP1$+" - "+"Descrizione = "+_ARDES$ 'stringa in stampa (codice + descrizione)
LPRINT "Cod. Articolo = "+_ARCOD$+" - "+"Descrizione = "+_ARDES$ 'stringa in stampa (codice + descrizione)

'Notare: al secondo pulsante di stampa, il valore VAL_INP1 viene azzerato e resta solo la descrizione

CHIUDISTAMP 'chiusura stampante aperta con APRISTAMP
'' CHIUDISTAMP_EXT 'chiusura stampante aperta con APRISTAMP_EXT
'' IF _ERRCHSTAMP <> 0 THEN MIOERR_STP$ = _ERRCHSTAMP$ : 'test CHIUDISTAMP_EXT

GOSUB ERRORE_STP 'gosub di test chiusura stampante aperta con APRISTAMP

```

```
RETURN
'-----
'ROUTINE INVIO ERRORI (lettura, variazione, stampa)
'-----
ERRORE_GET:
  PUTREM_STR "ERRORE_GET", MIOERR_GET$ 'ritorna la variabile di errore GET
RETURN
'-----
ERRORE_VAR:
  PUTREM_STR "ERRORE_VAR", MIOERR_VAR$ 'ritorna la variabile di errore VAR
RETURN
'-----
ERRORE_STP:
  PUTREM_STR "ERRORE_STP", MIOERR_STP$ 'ritorna la variabile di errore STP
RETURN
'-----
```

MXRT

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
RTAPRI	Par: [n]. Funzione apertura Registratore Telematico.	S/C/Cr
RTCHIUDI	Funzione chiusura Registratore Telematico aperto con RTAPRI.	S/C/Cr
RTCHIUSURAGIO	Funzione invio richiesta chiusura giornaliera al Registratore Telematico.	S/C/Cr
RTGETREPARTI	Funzione invio richiesta reparti configurati per Registratore Telematico.	S/C/Cr
RTGETPAGAMENTI	Funzione invio richiesta pagamenti configurati per Registratore Telematico.	S/C/Cr
RTDISPLAY	Par:<str>. Funzione invio informazioni sul display Registratore Telematico.	S/C/Cr
RTELENCODISP	Funzione che restituisce l'elenco dei Registratori Telematici configurati.	S/C/Cr
RTAPRICASSETTO	Funzione per comando apertura cassetto cassa	S/C/Cr
_ERRRT\$	Descrizione esito delle funzioni RT?	S/C/Cr
_RT	Variabili di struttura popolate dalle funzioni RT?	S/C/Cr
Logiche		
<p>Le funzioni RTAPRI e RTCHIUDI aprono e chiudono un Registratore Telematico diverso da quello predefinito per il Client, oppure servono per effettuare più operazioni sullo stesso Registratore Telematico, eseguendo l'apertura e la chiusura solo una volta.</p> <p>Se RTAPRI non viene chiamata per utilizzare una qualsiasi altra funzione SPRIX di gestione del Registratore Telematico, quest'ultima apre e chiude in automatico il Registratore Telematico predefinito per il Client.</p> <p>Tutte le funzioni RT? ritornano eventuali errori nella variabile _ERRRT\$.</p> <p>Per la gestione e la stampa da magazzino, consultare le variabili nella testata della struttura _MM relative ai driver MxEcr e MxRT.</p> <p>Nota bene: la stampa sul dispositivo RT è tutta pilotata in modo trasparente dalla PUTMM, pertanto i dati sono presi per come sono e come tali sono passati. Non esistono variabili _RT in grado di modificarli.</p>		

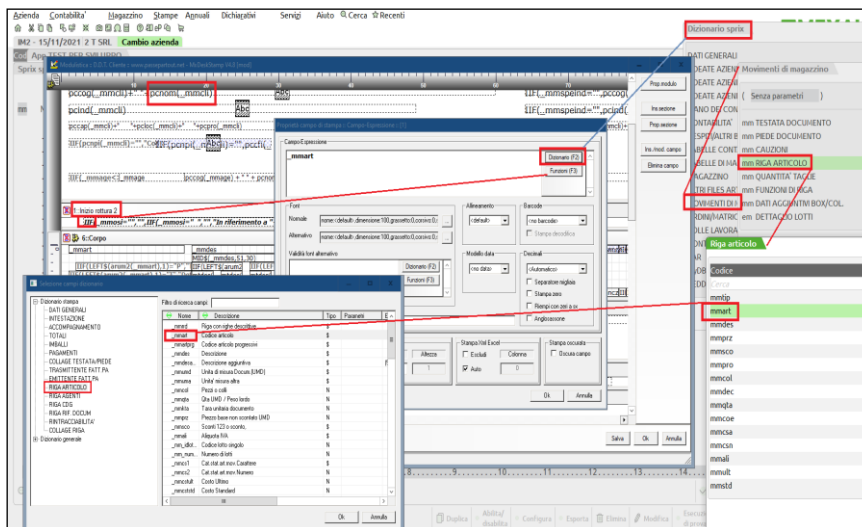
MODULI E ARCHIVI DEL GESTIONALE

Attenzione: Sprix/Collage non dispongono di tutti i **controlli** che il Passepartout è in grado di eseguire sulla congruità dei dati negli archivi, eccetto quelli indispensabili per rispettare i minimi requisiti per eseguire le operazioni. Sprix/Collage non hanno gli stessi **automatismi** del Passepartout, nemmeno le sue capacità di riorganizzazione o ricostruzione dei dati, pertanto, le personalizzazioni che modificano o cancellano archivi devono ricostruire congruenti relazioni fra dati, anche appartenenti a tabelle archivi separate, ma collegate; i moduli come il Controllo di Gestione, Docuvision (archiviazione documentale) e MyDB hanno una propria gestione separata di archivi, pertanto, sono da gestire in modo separato e parallelo rispetto ai moduli di Primanota, Parcellazione, Magazzino e Bolla di Lavoro a cui possono essere collegati. Sprix, infatti, non replica gli stessi automatismi del Gestionale e quindi, se si desiderano ottenere delle relazioni fra questi gruppi di archivi, vanno ricostruiti.

Si ricorda che la struttura dei dati del Gestionale è consultabile nei seguenti modi:

- Le Tabelle e Colonne sono definite nel documento "**ElencoColonneMexalDB.Xls**" allegato al chiarimento soluzione n. 30663 del servizio di assistenza EasyPass accessibile dall'Area Riservata di www.Passepartout.net. I contenuti delle colonne nel file vanno aperte ed estese. Per ulteriori approfondimenti consultare il capitolo MexalDB del Manuale Utente
- I servizi di replica **DBMIRROR** configurabili nei Moduli Aggiuntivi dei Dati Aziendali [F4] del menu Anagrafica Azienda [AN], oppure, nei casi di installazione LIVE (che non consentono configurazioni ODBC) le funzioni Sprix di replica come **ASC MIRROR**. In entrambi i casi, le tabelle di replica sono elencate in "\server\programmi\bin\loc\dbmirror.dat"
- I **Tracciati CSV** sono pubblicati nel chiarimento n. 54452 del supporto on-line. I Tracciati CSV sono verificabili nel menu del Gestionale [Servizi - Trasferimento Archivi - **Export/Import CSV - ASCII**]
- Specifiche dei campi di stampa della **Modulistica documenti grafica** (menu 'ZM) e **Stampe generali** (menu 'ZS') in quanto analoghi ai dati dei campi Dizionario del Gestionale [Servizi - Personalizzazioni - **Stampa Dizionario Sprix**] (menu 'ZZZ')

Esempio su documento BC con Modulistica di Stampa e Dizionario Sprix



Le suddette funzionalità sono considerabili **PROPEDEUTICHE** alla conoscenza della base dei dati del Gestionale, che in ambito Sprix e Collage è gestita con le "VARIABILI STRUTTURA" e "DIZIONARIO" e che tratteremo qui in seguito.

I dati obbligatori sono indicati a video dal Gestionale e le relazioni fra i dati sono descritte nel Manuale Utente.

STRUTTURA DATI E DIZIONARIO

In aggiunta alle variabili personali dell'utente, questo strumento di sviluppo dispone di specifiche variabili collegate ad operazioni (funzioni e istruzioni) sugli archivi del Gestionale. Queste variabili si definiscono di **Struttura** e di **Dizionario** e richiedono una corretta sintassi e dei precisi parametri di gestione.

I dati dell'installazione e degli archivi sono rappresentati da due tipi di variabili caratterizzate da una determinata sintassi e funzionalità:

- variabili di **STRUTTURA**, scritte in maiuscolo e consentono l'accesso ai dati sia in lettura che scrittura;
- variabili di **DIZIONARIO**, scritte in minuscolo e consentono l'accesso ai dati in sola lettura (diretta o ciclata);

Generalmente rappresentano i medesimi archivi di dati, ma non necessariamente la sintassi dei nomi delle variabili è la stessa. Per esempio, il codice del articolo è rappresentato dalla struttura con il nome "_ARCOD\$" e dal dizionario è rappresentato con lo stesso nome "arcod()", mentre invece, il numero record dell'articolo è rappresentato dalla struttura con il nome "_ARNRA" e dal dizionario con il nome "arrec()".

Per la variabile di una determinata struttura non sempre esiste la corrispondente variabile di dizionario e viceversa. Per esempio, il codice pagamento in primanota è previsto dalla struttura con la variabile "_PNPAG", ma non dal dizionario, poichè il dato è in sola scrittura; viceversa, esiste la variabile dizionario relativa al numero di terminale "xster", ma non esiste la corrispondente variabile di struttura.

La gestione dei dati in lettura ed in scrittura, pertanto, richiede **la completa conoscenza delle logiche degli archivi coinvolti**; questi possono variare in base a diverse circostanze quali: il tipo e la configurazione del prodotto; il livello di gestione aziendale; il tipo di registrazione che viene effettuata; le condizioni operative; il setting degli automatismi; ecc...; Se tutto questo viene ignorato dallo sviluppatore, può essere difficile (o non possibile) eseguire corrette operazioni sugli archivi.

Per capire come funziona il **Dizionario** rispetto alla **Struttura** è indispensabile conoscere il funzionamento della Struttura e l'argomento trattato dal manuale utente. Relativamente all'ambiente Sprix, le caratteristiche e le logiche dalle variabili sono documentate nelle tabelle delle Strutture Archivi qui contenute, mentre il dizionario, invece, essendo questo la stessa libreria in lettura non è qui documentato (salvo qualche eccezione), in quanto è stampabile e/o consultabile direttamente dall'editor.

Non essendo possibile ricomprendere esempi per tutte le casistiche, in alternativa, si può sfruttare un semplice stratagemma di "auto apprendimento": consiste nel registrare l'operazione desiderata dai moduli standard del Gestionale e poi stampare (o debuggare) tutta la struttura (con le GET) e/o dizionario (con il ciclotabellare) di tutti i dati interessati dall'operazione; se necessario, azzerate i dati in "cache" che sono da inizializzare, poi, per eventuali ulteriori nuove elaborazioni. Terminata la replica manuale con Sprix delle logiche relative ai dati (gestite in automatico dall'operatività standard), la funzione di Riorganizzazione archivi rilascia informazioni circa la congruità o meno dei dati. E' possibile, infine, verificare quali righe e campi con relativi valori sono coinvolti nell'operazione attraverso esportazioni su MEXALDB oppure le più semplici trasmissioni CSV il cui modulo (Servizi\TrasferimentoArchivi\InvioCSV) consente di editare il file (sul percorso del nome file tramite [Shift+F7]) per verificare le medesime principali variabili di struttura Sprix.

STRUTTURA

Le variabili di **STRUTTURA** consentono di operare negli archivi del gestionale con istruzioni di lettura (GET?), scrittura (PUT?) e cancellazione (DEL?), le quali generalmente terminano con la sigla dell'archivio coinvolto (es. GETPC legge il piano dei conti). Le variabili di struttura hanno sintassi `_XX?$()`:

- `_XX` è l'archivio coinvolto (es. '_PC' archivio piano dei conti);
- `?` è il dato dell'archivio (es. 'PAG' codice pagamento nell'archivio PC);
- `$` è l'eventuale stringa (es. 'COD\$' codice conto nell'archivio PC);
- `()` è l'eventuale array (es. 'NDS\$(1-2)' descriz. delle 2 note del conto);

Dall'esempio si ottengono così le variabili `_PCPAG`, `_PCCOD$`, `_PCNDS$(I)`, tutte variabili relative all'archivio del Piano dei Conti da poter così leggere (GETPC), oppure scrivere (PUTPC), oppure cancellare (DELPC) e infine controllarne gli esiti con la relativa variabile di errore `_ERRPC`.

Tutte le operazioni possibili (non solo GET, PUT, DEL) sono infatti controllate dal gestionale e segnalate con variabili di sistema `_ERR?` che segnalano l'eventuale errore¹³ e non sono forzabili e né aggirabili. Non sempre questi controlli sono attuati dal gestionale, poiché dipende da che cosa è attivo, per esempio, il controllo di congruità della Primanota avviene ovviamente solo se lo scadenzario è attivo.

¹³ In generale, si è affermato che il controllo della corretta esecuzione viene effettuato con il **debugger**, ma può essere fatto anche attraverso l'utilizzo delle variabili che rilasciano errori durante l'esecuzione delle istruzioni associate e la descrizione dell'eventuale messaggio di errore (ovvero le variabili `_ERR?` ed `_ERR?$()`, dove il simbolo '?' va sostituito dalla sigla identificativa dell'argomento trattato). In ambiente Collage, per testare l'errore è invece utilizzata l'etichetta `ON_ERR_[NOME_CAMPO]`.

In aggiunta a queste metodologie ne esiste un'altra: la corretta esecuzione di uno Sprix si verifica in base all'attendibilità delle variabili (di dizionario o delle strutture archivi) coinvolte dal programma, facendone la relativa **controprova**. Ad esempio, per verificare la corretta registrazione di una FE con una PUTPN, la verifica di eventuali errori in lettura è effettuabile con la seguente istruzione "IF _ERRPN<>0 THEN VIMSG 4,_ERRPN\$". Se ad esempio si vuole verificare, se la PUTPN di una FE intraCee non effettua correttamente la registrazione dei dati nella finestra intraCee di primanota, non è sufficiente verificare che in tale finestra vi sia l'assenza dei dati che si sarebbe voluto inserire con PUTPN, ma è necessario verificare la prova di stampa dei valori attribuiti alle variabili struttura archivi che si sarebbe voluto inserire in primanota con la PUTPN; se la stampa avviene, significa che il gestionale passa correttamente i valori a queste variabili e che sono quindi utilizzate dallo Sprix in modo corretto. Effettuare una GETPN della registrazione della FE intraCee, testare l'errore con "IF _ERRPN<>0 THEN VIMSG 4,_ERRPN\$", valorizzare delle variabili (numeriche/stringa) con i valori delle variabili strutture archivi provenienti dalla lettura con GETPN, stampare tali variabili (in alternativa si può scegliere di stampare direttamente le variabili strutture archivi).

I suddetti controlli non eseguiti dal gestionale, specie in presenza di automatismi della procedura standard, incontrano anche un altro importante aspetto da considerare e che apparentemente sembra un vincolo o un limite, ma facilmente superabile con logica e programmazione.

Possono rilevarsi, infatti, delle differenze fra quanto viene letto (GET) e quanto poi viene successivamente scritto (PUT): mentre l'istruzione di GET legge ciò che è scritto sul record (anche i dati di default modificati manualmente), l'istruzione di PUT, invece, registra sempre attivando gli automatismi della registrazione (come ad esempio sono alcuni dati default di anagrafica). Di conseguenza, la PUT che non modifica la struttura caricata da una precedente GET, non significa che si stia rinunciando alla revisione dei dati (come avviene nella procedura standard premendo il tasto di abbandono "Esc"), ma significa, invece, che si sta confermando e registrando l'operazione, compresa l'attivazione di tutto quello che l'istruzione PUT scrive di default (come avviene nella procedura standard premendo il tasto di conferma "F10").

In altri termini, lo Sprix consente di leggere oppure di scrivere, ma il concetto di "revisione" deve essere gestito dal programmatore in base alla logica e gli automatismi della procedura standard del gestionale. Un tipico esempio di questo comportamento è quello della "tara" degli articoli a peso nel documento di magazzino, cioè la variabile `_MMKTA(R)` e i suoi differenti comportamenti fra l'ambito della GETMM e quello della PUTMM, a cui si rimanda per approfondimenti.

In questo manuale le variabili di struttura sono indicate sia nei gruppi di argomento relativi ai "MODULI E ARCHIVI DEL GESTIONALE", ma poi l'elenco dettagliato di tutte le variabili è nella successiva LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI elencata in ordine alfabetico.

In ragione delle implementazioni del Gestionale alcune variabili ri-nascono 'rinnovate' nel nome e nella sostanza, aggiungendosi e affiancandosi nell'elenco a quelle originarie: questo consente di allineare ed aggiornare l'eventuale revisione del vecchio sorgente a versioni più recenti del Gestionale; in altre parole, il Gestionale si evolve ed amplia talune informazioni, ma piuttosto che costringere gli sviluppatori a modificare i propri sorgenti, preferisce implementare aggiungendo e doppiando le variabili sul medesimo gruppo di informazioni, ma di consistenza più estesa e/o più potente (per esempio la necessità di abbandonare la gestione di dati su una array statica in favore di quella dinamica) consentendo, quindi, delle opportunità di scelta sul potere "restare" al vecchio oppure cambiare verso il nuovo".

L'elenco di variabili non riporta tutte le informazioni sulla gestione dei dati coinvolti, pertanto è fondamentale consultare il Manuale Generale dell'utente. Alcuni dati del Gestionale, per esempio quelli tabellati, non hanno proprie variabili di struttura e sono modificabili solo dal gestionale oppure con Collage.

Attenzione: taluni archivi sembrano appartenenti ad una determinata struttura, ma in realtà sono dei "Dati Aggiuntivi" realizzati con strumenti che consentono di realizzare strutture di dati "trasversali" ad altri archivi, fra cui quelli realizzati con "MyDB". Ciò significa che la gestione dei tali archivi non è consentita dai comandi relativi ad altri archivi (ad esempio dai documenti di magazzino "_MM" o gli articoli "_AR" oppure i conti "_PC", ecc.), poichè occorre utilizzare quelli che sono propri dello strumento, ovvero MyDB, pertanto sono le variabili `_MYDB`, la lettura `MYDBGGET`, la scrittura `MYDBPUT`, la cancellazione `MYDBDEL`, ecc.. e con una sintassi invertita.

Nota bene: la ricerca nel manuale on-line tramite [SH+F1] funziona solo sulla radice della struttura (esempio: `_PC`, `_AR`, `_MM`, ecc..).

Per descrivere più dettagliatamente il suddetto esempio, ci si potrebbe porre la questione del perché l'istruzione PUTPN non registri i valori nel modello intraCee: registrando una FE IntraCee numero 1/00001 in data 22/02/2002 da Magazzino, attivando lo Sprix, si vorrebbe registrare sulle FE contabilizzata in primanota, alcuni dati nella finestra dell'Intra, ma senza riuscirci. Prima di effettuare la controprova [analizziamo come opera lo Sprix](#):

```
'leggo la FE e controllo l'esistenza di errori, poi cancello la Fe e controllo l'esistenza di errori
GETPN "20020222",0,"FE","V",0,1,1,"20020222": IF _ERRPN<>0 THEN VIMSG 4,_ERRPN$
DELPN "20020222",0,"FE","V",0,1,1,"20020222": IF _ERRPN<>0 THEN VIMSG 4,_ERRPN$
'attribuisco valori alle variabili
_PNDESS(1)="DESCRIZIONE": _PNINT$="S": _PNIEU$="S": _PNNTR$(1)="": _PNNCM(1)=73181691: _PNMNE(1)=95.25:
'registro la FE e controllo l'esistenza di errori e termino la routine
PUTPN: IF _ERRPN<>0 THEN VIMSG 4,_ERRPN$: END 'termina
```

Per effettuare la controprova, occorre verificare se la stampa dei valori nei campi dell'intraCee avviene correttamente; in caso positivo, significa che il gestionale attribuisce correttamente i valori stabiliti alle variabili, per cui sono utilizzabili dallo Sprix. Eseguo la controprova:

```
'leggo la primanota e controllo l'esistenza di errori
GETPN "20020222",0,"FE","V",0,2,2,"20020222": IF _ERRPN<>0 THEN VIMSG 4,_ERRPN$
'attribuisco alle variabili con il valore letto
AMMO=_PNAOP(1): A$=_PNNTR$(1): B=_PNNCM(1): C=_PNMNE(1): D=_PNAOV(1): E=_PNUSP(1): F=_PNVST(1)
G$=_PNRGM$(1): H$=_PNMTR$(1): I$=_PNPDE$(1): L$=_PNPPOS(1): M$=_PNPDSS(1)
'stampo le variabili con i valori provenienti dalla lettura della primanota
N|_PNAOP(1)=AMMO|_PNNTR$(1)=A$|_PNNCM(1)=B|_PNMNE(1)=C|_PNUSP(1)=E|_PNVST(1)=F
N|_PNAOV(1)=D|_PNRGM$(1)=G|_PNMTR$(1)=H|_PNPDE$(1)=I|_PNPPOS(1)=L|_PNPDSS(1)=M
```

DIZIONARIO

Il **Dizionario** è un insieme di tabelle di dati in sola lettura, relativi all'installazione e ai dati aziendali. Si consulta in modo guidato direttamente dall'Editor: il tasto [F2] che apre un elenco di gruppi di archivi scorribili e selezionabili con Invio, per accedere a sottoelenchi di tabelle archivi contenenti altri sotto elenchi di nomi di variabili di selezionabili con [Invio]. Dove previsto, è possibile visualizzare anche delle note descrittive.

In testa alla tabella possono esserci parametri corrispondenti alle prime voci del dizionario e che rappresentano **chiavi** per identificare quelle successive.

Per esempio il prezzo di listino di un articolo 'arprz' richiede l'indicazione di quale listino (l) e quale articolo arcod(), di conseguenza la relativa sintassi è arprz(l)(arcod()) in cui sia l (l'indice d'array) che arcod() sono indicabili direttamente con un valore, es. l=2, COD_ART\$ = "CODICEARTICOLO" e quindi arprz(2)("CODICEARTICOLO"), o indirettamente con un'altra variabile, anche di Dizionario oppure di Struttura, ad esempio arprz(2)(COD_ART\$) o arprz(2)(afcod()) o arprz(2)(_ARCOD\$). L'importante è che riportino comunque un valore equivalente al parametro richiesto. Si possono fare anche altri esempi come la primanota per leggere la descrizione del piano dei conti pcdes(pncto()), la lettura taglie mmtag(l)() mtdes(mmntt(),l), oltag(l)() o la lettura di scadenze pspim(l)() e così via, fino a sequenze di parametri complesse come sono quelle richieste dalla quantità di inventario aqinv(arcod(),1, sxann) o particolarità prezzo dell'articolo tzprz(1)("P", "", 1, arcod(), 1, sxdat).

A destra di ciascuna voce c'è la relativa lunghezza e la natura numerica o alfanumerica della variabile, con eventuale numero di elementi array fra parentesi e da indicare prima dei parametri; es. il prezzo del secondo listino dell'articolo "ART1" nella tabella 'ar' di anagrafica articoli è dato da arprz(2)("ART1"). **Nota:** per alcune variabili è attivo l'Help in linea [F2].

L'istruzione **AZZCACHE** (per Sprix / Collage / Datatrek) azzerava la CACHE del Dizionario. Nella gestione delle tabelle dizionario, può capitare che sia rilasciato il messaggio "funzione eseguibile solo in modalità inserimento": significa che è attivo il tasto funzione INS (segnalato a video con SSC), pertanto, disattivare la funzione.

Consigli sulle ricerche nel Dizionario

Nell'elenco delle tabelle del Dizionario è disponibile la ricerca Full Text, che agevola la ricerca del dato tramite la descrizione del campo dizionario. Purtroppo è utile solo se si è posizionati nella tabella giusta, pertanto può essere utile il seguente ulteriore metodo.

Il Dizionario è stampabile in PDF da menu Servizi-Personalizzazioni-Stampa Dizionario e consente di ricercare con parola chiave significativa dell'argomento cercato e relativamente alla Descrizione. Essendo la stampa un elenco molto ristretto è bene ricercare su una porzione della parola cercata, per esempio cercare "prov" anziché "provvigioni". Per restringere il campo di ricerca è bene utilizzare parole poco ripetute, per esempio "driver" anziché "nome" per la configurazione di collegamento al DB.

Nota Bene: la stampa del Dizionario è allineata alla libreria della versione in uso ed attiva in relazione al tipo di installazione in uso: la variabile di Dizionario non presente o non disponibile oppure non attiva, dipende dalle funzioni attive e/o attivate nell'installazione della versione in uso e aggiornata.

Qualora vi fosse ancora incertezza sul significato delle variabili o il loro utilizzo, anche come chiave (parametro) del dato, spesso è utile effettuare la registrazione di interesse attraverso i moduli standard, per poi leggere i valori del dizionario tramite ciclotabellare (oppure a lettura diretta quando non sono ciclabili); questo metodo empirico consente, infatti, di conoscere i valori dei parametri che sono necessari per le letture dirette e che sono rappresentati dai primi elementi di tabella. Per esempio, data la tabella "pc" che richiede parametro Par: (conto\$), per conoscere la descrizione [pcdes()] del conto cliente occorre fornire la chiave lettura che è rappresentata dal codice del conto che è pcdes("501.00001"); questo parametro è quindi il codice conto, rappresentato in testa all'elenco della tabella da pccod(). Si ricorda infine, che la nomenclatura e l'ordine di elencazione delle variabili dizionario può NON essere equivalente a quella delle variabili di struttura (_??).

Nota: considerare che **Collage** non sia in grado leggere il Dizionario non è corretto. In realtà **non può utilizzare il ciclotabellare**, pertanto, se c'è la possibilità di accedere al dato univoco di dizionario (esempio tabella "sx") oppure tramite le relative chiavi (esempio tabella "pc") è consentito leggere il dato, nella maggior parte dei casi anche di un altro archivio dove Collage non sta operando; dato che Collage non dispone di ciclotabellare alcune correlazioni fra dati possono richiedere appoggi su file esterne.

Collage, oltre a leggere dati presenti video, può quindi ricercare dati negli archivi tramite Dizionario ad accesso DB diretto o tramite chiave (in assenza di questa si può eseguire uno Sprix tramite SHELLSPRIX).

Per esempio: supponiamo di volere introdurre un controllo in fatturazione da Magazzino relativo a un confronto sul fatturato annuo ed un valore soglia attribuito al medesimo Cliente della Rubrica, con conseguente blocco o avviso in caso di superamento di tale soglia. Questa funzionalità non è prevista nello standard e si potrebbe essere portati a pensare di risolverlo impostando il "Fuori Fido", ma non è corretto, poiché quando il cliente paga modifica le rate e il fuori fido.

Come primo approccio (supponendo erroneamente che Collage non legga il Dizionario, specie di un altro archivio) si penserebbe a questa soluzione: creo una videata personalizzata sul cliente dove inserire il valore soglia; all'uscita dal campo di Magazzino salvo il cliente (_SINP\$) e con SHELLSPRIX lancio Sprix di lettura dei dati anagrafici del conto (con ciclotabellare o con GETPC); confronto i dati e poi eventualmente blocco la conferma, oppure rilascio un messaggio, oppure apro una stampa e-mail, ecc...

Considerando invece che Collage legge il dizionario (se in chiave e non è necessariamente da dovere ciclare, come ad esempio le tabelle "pn" o "mm"), una soluzione potrebbe essere quella di creare la videata numerica (pc1aa), ma poi eseguire tutto con questo semplice Collage su menu MD Magazzino, esempio:

```
ON_IN_MDAGEN : GETATMM : VID=pc1aa(_MMCLI$) : FAT=pctft(sxann+1)(_MMCLI$)
IF VID<FAT THEN VIMSG 1, "VID minore di FAT" '(oppure blocco _TF o APRISTAMP_EMAIL)
END
```

Può accadere che all'improvviso il dizionario smetta di funzionare e che non si riesca più a leggere il dato d'interesse, senza capire le motivazioni, in quanto tutto sembra essere regolare e non sono intervenuti particolari eventi che abbiano modificato lo stato delle cose (archivi e codice sprix). Questa situazione può essere provocata da due fattori, anche fra essi connessi:

- il cliente cambia lo stile di lavoro (attiva parametri o funzioni nuove o adotta logiche nuove e non previste dal programmatore, oppure semplicemente, progredisce nella sua operatività quotidiana);
- il codice realizzato è diventato vecchio, inadeguato allo scopo, cioè non è più in grado di gestire (o prevedere) le operatività del cliente;

Principalmente si è portati a pensare all'anomalia senza verificare il presupposto per la lettura del dizionario, ovvero le chiavi della tabella e le condizioni stabilite nel gestionale per poterle leggere. Esempio: si verifica che leggendo i dati di un articolo quali unità di misura principale, secondaria e minimo fatturabile tramite sprix se l'articolo è strutturato con autocodifica a posizione inferiore alla totale lunghezza dell'articolo, i dati rilevati sono nulli; soluzione: per risolvere il problema bisogna eseguire la lettura dell'articolo tramite una variabile creata appositamente in modo da leggere la lunghezza dell'autocodifica. Nell'esempio sottostante si assegna alla variabile LUN la lunghezza dell'autocodifica e poi si visualizza l'unità di misura principale dell'articolo.

```
ON_OUT_MDCODART:
LUN = mraut(arstr(LEFT$(_SINP$,amdra)))
VIMSG 5,arum1(LEFT$(_SINP$,LUN))
END
```

Il Dizionario può essere letto in modo sequenziale attraverso il cosiddetto **ciclotabellare**. Se la tabella è indicata 'senza parametri' le voci sono a lettura diretta in quanto uniche. Ad esempio, il dato di sistema relativo all'anno di gestione 'sxang' oppure all'anno in linea 'sxann'. Se l'indicazione 'senza parametri' è evidenziata, significa che la lettura dei record della tabella può avvenire solo con Ciclotabellare (non disponibile in Collage), che apre l'archivio assumendo automaticamente i parametri della tabella (e che quindi non sono necessari), lo legge e lo chiude al termine del programma. Il ciclo tabellare viene battezzato con la sigla della tabella: si attiva nell'editor con tasto [F4] e si cancella sulla stessa riga di attivazione con [ShF3]; nel primo campo Cod.Tab' con [F2] si cerca la tabella di interesse da confermare con Invio.

Nei campi successivi non obbligatori si selezionano le variabili Dizionario [F2+Invio+Invio]; nel campo di 'Scelte' si può inserire un'espressione logica fino a 220 caratteri per condizionare la scelta dei record, anche tramite valori di Dizionario prestabiliti fuori dal ciclo, per esempio posta la variabile alfanumerica ART\$="ART1" (codice articolo), la scelta arcod()=ART\$ seleziona solo "ART1" o la scelta mmsg()="FT" AND mmart()=ART\$ seleziona nei movimenti solo fatture con articolo "ART1", oppure ppsaf(pccod(),sxann)>=0) seleziona i saldi positivi dei conti dell'anno in linea, e così' via..

Ciascun campo 'Ordine' di massimo 66 caratteri, consente espressioni fino a nove variabili e/o porzioni di esse, concatenate con '+', per creare gruppi gerarchici decrescenti di ordini di lettura (da 3 a 1) ed ottenere una vista logica di estrapolazione dei record. Senza ordinamenti la lettura è ordinata per chiave primaria del archivio, più rapida di ordinamenti articolati o che ripetono la stessa chiave (ad esempio, inserendo nel ciclo di lettura di documenti di magazzino un ordine 3 con sigla+serie+numero+conto del documento o in modo equivalente la sigla su ordine 3, la serie su 2 e il numero su1). Se l'ordine di lettura non è necessario, non utilizzare l'ordine.

Confermando il ciclotabellare con [F10] si marca la riga dell'editor con la sigla della tabella scelta visualizzata in reverse a sinistra della riga di separazione verticale. Il ciclo è estensibile ad altre righe contigue premendo F4 su di esse, per creare gruppi di istruzioni e/o di campi di stampa da ripetere tante volte per quanti sono i record letti. Ciascuna riga marcata può essere definita una rottura di livello dell'ordinamento con [ShF4].

In ciascun livello è infatti gestita la relativa rottura per condizionare la lettura con opportuni indicatori, che vengono accesi e eseguiti quando si verificano le condizioni presenti nell'espressione indicata nel gruppo gerarchico, per effettuare stampe di totali, salti pagina, calcoli, ecc... nelle righe in cui questi indicatori sono opportunamente posizionati. I livelli di rottura di ordinamento sono quindi tre (3, 2, 1), con due tipi di indicatori ciascuno, di intestazione 'I' (rispetto al record precedente) e totale 'T' (rispetto al record successivo): I3 e T3, I2 e T2, I1 e T1, evidenziati nel editor in reverse a fianco della sigla del ciclotabellare. Nella sequenza di righe di ciclotabellare l'indentazione delle colonne nell'editor deve essere decrescente da sinistra verso destra per le 'I' e crescente da destra verso sinistra per le 'T'. La sequenza dell'indentazione non si può invertire (al termine del paragrafo c'è un esempio che dimostra le rotture relative agli ordini di livello su dati dei documenti di magazzino).

Sul primo record letto sono rotti tutti i livelli di ordinamento ed attivati tutti gli indicatori 'I'. Se nel record letto c'è la rottura di ordinamento rispetto al record precedente sono attivati gli 'I', altrimenti se c'è rottura rispetto a quello successivo sono attivati i 'T'. Per ciascuna rottura di livello superiore si attivano automaticamente le rotture di tutti i livelli inferiori dello stesso tipo (I3, I2, I1) → (I2, I1) → (I1) e (T3, T2, T1) → (T2, T1) → (T1). Sul ultimo record letto sono rotti tutti i livelli d'ordinamento con l'attivazione dei 'T'. Esempio: posto un insieme non ordinato di record di clienti, per contarli e totalizzarli per gruppi gerarchici Comune, Provincia e Regione si cicla la tabella dei clienti con Ordine 3: Regione, Ordine 2: Provincia e Ordine 1: Comune. Confermato il ciclotabellare (eventualmente preceduto da righe di intestazione) l'estensione nel editor di righe marcate come ciclotabellare (es. num. 8 righe), consente di stabilire quanto segue:

1. (R=1) riga capo del ciclo con rottura I3 per settare il contatore Regione (impostato fuori ciclo non verrebbe inizializzato ad ogni record) (a seconda dei casi può differire di +/-1 rispetto ai record)

2. (P=1) riga con rottura I2 per settare il contatore Provincia
3. (C=1) riga con rottura I1 per settare il contatore Comune
4. [-pc???] riga di campi stampa di variabili d'interesse lette
5. (R=R+1 : P=P+1 : C=C+1) riga di incremento contatori per record letto; può contenere anche altri calcoli, totalizzazioni, decrementi (X=X-1) e condizioni IF ... THEN Y=Y+1, ecc...con risultato da riportare poi in stampa alla rottura
6. [-C-1] riga di stampa con rottura T1 del totale per Comune
7. [-P-1] riga di stampa con rottura T2 del totale per Provincia
8. [-R-1] riga di stampa con rottura T3 del totale per Regione

Quando il tipo di ordinamento desiderato non è consentito dal ciclotabellare (in ragione di come sono scritti i dati) è possibile avvalersi di utili concatenazioni tramite la funzione MID\$, cioè una funzione utile anche per ottenere ordinamenti non previsti dal ciclotabellare come può essere la rappresentazione in stringa di valori numerici (per esempio mmsig, mmtsti, mmcds, mmcos,...) oppure tramite l'export dei dati (file, array..) oppure tramite funzioni di SORT.

Con l'istruzione (solo Sprix) **EXITTAB** si forza l'uscita dalla tabella di lettura del ciclotabellare per proseguire con il codice successivo al ciclo. Sono consentiti fino a 99 ciclitabellari per programma. Per evitare letture falsate della tabella si **sconsiglia** di inserire nel ciclotabellare delle operazioni di scrittura (PUT) e cancellazione (DEL), poiché possono interferire con l'operazione ciclica di lettura, per effettuare tali operazioni è consigliato l'utilizzo di ulteriori cicli esterni, per esempio su array in memoria oppure su altro supporto (es. un file) precaricati con ciclotabellare. Esempio: salvare i dati su file random, effettuare GET del file, eseguire GET dell'archivio (es. GETAR) e relativa PUT nell'archivio (es. PUTAR). Il ciclotabellare non deve eseguire operazioni di Mirroring (DBMIRROR, ASCMIRROR, ecc..). L'istruzione GET può essere tollerata in quanto è un'operazione di lettura, ma è comunque bene verificare che non interferisca.

Al termine dell'esecuzione del ciclotabellare, le istruzioni che accedono direttamente ai file degli archivi del gestionale potrebbero non essere attive poiché le operazioni sugli archivi, dirette o ciclata, aprono determinati file del gestionale (esempio: rudt.sig, arti.sig, ecc...) e non li chiudono, né rilasciano, fino al termine dell'esecuzione dell'intero programma SPRIX, ragion per cui, a tutela dei suddetti archivi possono essere rilasciati messaggi relativi all'ineseguibilità dell'eventuale operazione (es. kill, copyfile, ecc..) in quanto il file è aperto.

Ordine di elaborazione dei documenti da parte del ciclo tabellare:

1) IN	10) FC	19) BS	28) PF
2) FF	11) RS	20) BP	29) OF
3) FT	12) SL	21) BC	30) OC
4) FS	13) CL	22) MF	31) OX
5) FD	14) DL	23) MA	32) NO
6) FP	15) RC	24) MX	33) PA
7) NC	16) RF	25) PR	34) NS
8) CO	17) BD	26) PX	35) ND
9) RI	18) BF	27) PC	36) NF

Il Ciclotabellare **legge l'anno dell'azienda aperta**, cioè elabora esclusivamente i dati dell'anno di ingresso per l'apertura dell'azienda¹⁴. Eccetto per alcuni archivi (ad esempio quelli del Controllo di Gestione) il ciclotabellare elabora, quindi, **un solo anno per volta**. Per potere leggere i dati di più anni in linea (anche se pregressi e/o chiusi), occorre eseguire il singolo ciclo su ciascun singolo anno aperto con l'istruzione di apertura azienda ed eseguirne la contestuale memorizzazione dei valori su array oppure l'esportazione su altri supporti (file, db, ecc.) al fine di eseguirne poi delle successive elaborazioni e/o confronti fra i dati prelevati dai vari anni. Per elaborare più anni in linea di archivi non multiannuali (ed esempio quelli del Controllo di Gestione), quindi, occorre avvalersi dell'istruzione **APRIAZIE** per effettuare singole aperture azienda su ciascun anno da ciclare, esportandone i risultati per effettuare le elaborazioni globali (confronti in memoria di valori di cicli tabellari operanti su anni diversi).

I ciclitabellari non sono nidificabili e per estrapolare variabili non appartenenti alla tabella ciclata, occorre indicare nelle stesse i precisi parametri necessari alla lettura. L'indicazione può essere diretta, anche tramite un'altra variabile, come può avvenire anche fuori del ciclotabellare, esempio, dato il ciclotabellare 'pn' sulla primanota e data la variabile conto C\$="201.00001", la variabile 'descrizione del conto' pcdes() può essere indicata pcdes(C\$) o pcdes("201.00001"); l'indicazione può essere anche con ciclotabellare, esempio, dai movimenti di magazzino 'mm' per leggere i parametri necessari alle variabili del piano dei conti pccog(mmccli()) e dell'anagrafica articoli ardes(mmart()) oppure dalla primanota 'pn' alla descrizione del piano dei conti pcdes(pncto()); il caricamento dei parametri può avvenire su array con ciclotabellare su 'ps', esempio ARRAY(l)=psnum(), da ciclare su 'pn' dove scegliere ad esempio, pnndo()=ARRAY di tanti elementi per quanti sono stati caricati col primo ciclo.

In alcuni casi, la lettura di determinati archivi (oppure singoli loro dati) deve essere attivata da apposite istruzioni. Per esempio, Il ciclotabellare su archivio **MyDB**, richiede che prima dell'esecuzione del ciclo sia utilizzata l'istruzione MYDBTAB per dichiarare nello specifico qual è l'archivio su cui deve essere eseguito il ciclo. Altro esempio, le righe descrittive nel corpo del documento di magazzino deve essere attivata antepoendo l'istruzione LEGGIDESC al ciclotabellare (v. esempio), ma anche le operazioni di mirroring (DBMIRROR, ASCMIRROR).

Esempio: documenti trasformati (in Bolla o Fattura) e contenenti descrizioni nel corpo

¹⁴ Questa regola è analoga anche nelle stampe personalizzate, che sono annuali. Per esempio, la stampa dello Scadenziario a Partite (-SCADPAR), nonostante consenta la seguente sintassi pntde(_pndre,_pnrpr) in realtà non legge le intestazioni della primanota dell'anno precedente, in quanto il parametro numero progressivo di primanota (_pnrpr) appartiene al dizionario di stampa (si noti che ha il prefisso underscore) e non al dizionario generale che, invece, sarebbe pnrrn() e che in un ciclotabellare "pd" non sarebbe consultabile in quanto solamente ciclabile.

Azienda Contabilità Magazzino Produzione Stampe Annuali Servizi DR ? Cerca Molto soddisfatto

DEM - 03/12/07 DEMO (e test) Sprix Aziende (Generale): spx1

Tabella
Cod.tab mm MOVIMENTI DI MAGAZZINO - TESTATA DOCUMENTO [9]

Ordine 3 mmtsi()+mtse()+mtnu()
Ordine 2 mmosi()+mmose()+mmonu()
Ordine 1 mmord()+mmart()+mmdsc()
Scelte mmtsi()="BC" OR mmtsi()="FT"

Ordinamenti
Ordine di livello T3
Elenco Ok Annulla

Ordinamenti
Ordine di livello T2
Elenco Ok Annulla

esecuzione Esecuzione in prova
Editor manuale utente Editor moduli di importazione
Cambio modalita' editor

1/1

[4] 1.....1.....2.....3.....4.....5.....6.....7.....8.....9.....10.....

Cerca Salva Annulla

'Documenti trasformati contenenti delle descrizioni nel corpo
LEGGIDESC 1
mm I N BC/FT ~mtnu()
mm I N
mm N OC ~mmonu()
mm N Righe ~mmart() Des. ~mmdsc() Qta ~mmqta() Prez ~mmprz()
mm TOT=TOT+mmprz()
mm T N Tot Prez ~TOT
mm T TOT=0

'Ordinamento corpo documenti BC provenienti dal raggruppamento di piu' documenti al cui interno vi siano
'anche delle righe descrittive (tipo . o ,)
LEGGIDESC 1
mm I N BOLLA ~mtnu()
mm I N ORDINE ~mmonu()
mm N ~mmart() ~mmdsc()

DATI GENERALI – AZIENDA

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
APRIAZIE	Par: <sigla\$>, <data\$>. Istruzione di Apertura Azienda. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder	S/Cr/Sh
APRIAZIE_EXT	Par: <sigla\$>, <data\$>, <codSottAz\$>, [att\$] Istruzione che apre Azienda con sotto azienda. Abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder	S/Cr/Sh
_ERRAZIE	Codice risultato Apertura Azienda	S/Cd
_ERRAZIE\$	Messaggio operazione Apertura Azienda	S/Cd
ON_STARTZA: ON_PUTZA: ON_ENDZA:	Eventi in gestione anagrafica nominative aziendali	C/Cd
esempi e casi dizionario		
utenti connessi	E' disponibile la variabile di dizionario sui <u>terminali</u> attivi "sxnte", ma non è disponibile una sugli <u>utenti</u> attivi, poichè il controllo avviene tramite la console del Server, la console del Client e dai file di Log del Server (per esempio, il file è interrogabile per conoscere quali utenti stanno lavorando e dove, oppure per sapere se l'esecuzione Sprix supera il numero di processi consentiti - diversamente, dedicare un desk utente per procedure riservate a shellsprix oppure testare l'esito della procedura sprix ed eventualmente posticiparla -).	S/C/Cd
utente studio / azienda	Le installazioni hanno strutture differenti, in mancanza di specifica variabile che distingua un utente studio da un utente azienda, ove consentito si verifica in modo indiretto (percorsi di lavoro, livello azienda, livello terminale, ecc..)	
sxpas, akdpa, akcpa, akipa, ajpas	richiede attivazione del parametro "Ottieni password" dal menu di "Amministrazione utenti"	S/C/Cd
sxcsa, sxcai	dell'azienda "G" (gestione sottoazienda) è = al codice "651"	S/C/Cd
sxdag, sxdam, sxdaa, xsdtd, xsxdg, xsxsm, xsxda, xsxsor	dizionario data; le date sono sempre nel formato AAAAMMGG, ma il campo riporta dimensione a 10 caratteri alfanumerici poiché la leggibilità di stampa richiede il comando STRDATA\$	S/C/Sm/Cd
sxmyd	Nome archivio MyDB principale in revisione. IFF sxmyd<>"123456esempi_ANMYDB" ("CodConNomApp_ANMYDB") Note: Il nome dell'App "CodConNomApp" è indispensabile, anche se è contenuta nella medesima App. Il nome dell'App "CodConNomApp" è separato dal nome dell'anagrafica MyDB "ANMYDB" dal carattere "_"; (in alcuni casi richiedeva il carattere "@"). ATTENZIONE: nella stringa "case sensitive" (maiuscole e minuscole) avvalersi delle funzioni CVMAIU(<str\$>) e CVMIN (<str\$>)	S/C/Cd
sxarc	Nome archivio MyDB corrente (cioè usato da <sxmyd>). Revisione di un'anagrafica all'interno di un'altra anagrafica. Nota: vale la stessa relazione al dizionario 'sxmyd'. ON_START: IFF sxarc <> "123456esempi_ANMYDB" STOPCOLLAGE "" ELSEF _DES612\$ = "Pulsante mio" ENDF END ON_612_YMYD001: ON_612_YMYD002: VIMSG 4, "Sono il collage 3" END	S/C/Cd
sxute, sxnut, azpri, azult, ammag	Nome utente in linea, Numero utente in linea, Primo anno gestito azienda in linea, Ultimo anno gestito azienda in linea, Numero magazzini azienda in linea	S/C/Sm
sycpl, sycpp	Credenziali (login e password) di Clicca & Posta. La lettura delle variabili suddette è consentita senza restrizioni al solo utente amministratore. Gli altri utenti possono leggere solo le proprie se autorizzati nella configurazione utenti.	S/C/Cd
apval	Valuta di gestione azienda. Es. apval(sxann)	S/C/Cd
acdti, sdti	Le variabili acdti e sdti vanno considerate solo per la parte riguardante il mese e il giorno, ed è per questo che sono dichiarate per 5 caratteri. Quindi anche se la valorizzazione risulta compresa di anno questo non va considerato	S/C/Cd
<p>Caso: l'Azienda Storicizzata non è modificabile ed è in sola consultazione</p> <p>La storicizzazione pone l'azienda nell'irreversibile sola consultazione e non consente l'utilizzo delle Applicazioni o l'accesso all'ambiente di sviluppo; per gestire i contenuti delle Applicazioni occorre operare da un'altra Azienda oppure da Sistema Operativo.</p> <p>In situazioni obsolete di Gruppi Sprix, per bloccare il messaggio dei collage, duplicare il gruppo sprix a cui appartengono le aziende storicizzate, infine svuotare la directory. Esempio se la storicizzata è nel gruppo sprix 4 (pers\sprix\sprix.004) duplicare la cartella rinominandola in gruppo sprix.005 e cancellare gli sprix presenti nel gruppo 4. Se altre aziende attive lavorano nel gruppo 4, spostarle nel gruppo 5. Se le aziende (storicizzate e non) puntano alla cartella generale attiva (pers\sprix) e quindi non ci sono gruppi sprix.nnn (pers\sprix\sprix.nnn), occorrerebbe svuotare la cartella (pers\sprix) del contenuto attivo in sprix/collage e portarlo in una cartella pers\sprix\sprix.nnn da dedicare alle aziende attive non storicizzate ed a cui far puntare tramite il gruppo (nnn) stabilito in Anagrafica Azienda.</p>		
<p>Caso: per distinguere gli anni aperti da quelli chiusi</p>		


```
ON_START: _INTER=0 : END  
ON_612_CTDCOAN: ESECMENU "AM" : END
```

Secondo Collage su menu AM

```
ON_IN_AMOPERATOR: PUTTAST "#505#505F#501" : END
```

Nota bene: non utilizzare il tasto funzione 622 (poichè lo Sh+F10 è riservato e non documentato fra quelli disponibili nella SETTF)

NUMERATORI

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
GETNUM()	Par: (<Tiposigla\$>,<SiglaDoc\$>,<Anno>,<Serie>) Funzione che legge il Numeratore	S/C/Cr/Cd/Sh
SETNUM	Par: <TipoSigla\$>,<SiglaDoc\$>,<Anno>,<Serie>,<Data\$>,<Numeratore> Istruzione che setta il Numeratore Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
_ERRNUM\$	Messaggio operazione sul Numeratore	S/C/Cd
esempi e casi dizionario		
<p>Esempio: lettura di numeratore e set con nuovo valore (numero e data)</p> <pre>NUM = GETNUM("C","V",2009,1): IF _ERRNUM\$<>" THEN VIMSG 4,_ERRNUM\$ VIMSG 1, STR\$(NUM,0,0) SETNUM "C","V",sxdAA,1,"20090508",129: IF _ERRNUM\$<>" THEN VIMSG 4,_ERRNUM\$ END</pre>		
<p>Caso: SETNUM forza la data se minore di quella sul numeratore</p> <p>L'istruzione SETNUM aggiorna sempre il numero, mentre aggiorna la data solo se maggiore come da gestionale.</p>		
<p>Caso: lettura ultimo numero documento</p> <p>La lettura ultimo documento può avvenire con il dizionario "nu" o GETNUM Es. per Corrispettivo del primo sezionale A=num("CO",sxann,1): VIMSG 4, STR\$(A,0,0) A=GETNUM("C","CO",sxang,1): VIMSG 4, STR\$(A,0,0) La gestione in ipotesi di disallineamento fra numeratori in tabella e gli effettivi documenti presenti nei movimenti, richiede la lettura di questi ultimi</p>		

CALENDARIO (CM)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 40	Istruzione che azzerà variabili struttura di Calendario _CM	S/C/Cr/Cd Sm/Cm/Sh/Cc
AZZCATVARSYS "CM"	Istruzione che azzerà variabili struttura di Calendario _CM (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETCM	Par: <ute\$>,<data\$>.[id\$],[adata\$] Istruzione che legge i dati Calendario dell'Agenda Attività Utente	S/Cr/Cd/Cc
PUTCM	Istruzione che scrive i dati in Calendario dell'Agenda Attività Utente	S/Cr/Cd/Cc
DELCM	Par: <ute\$>,<data\$>.[id\$],[adata\$] Istruzione che cancella i dati in Calendario dell'Agenda Attività Utente	S/Cr/Cd/Cc
_ERRCM	Codice risultato operazioni Calendario dell'Agenda Attività Utente	Cc
_ERRCM\$	Messaggio operazioni su dati Calendario dell'Agenda Attività Utente	Cc
_NCM	Quantità di elementi in Calendario dell'Agenda Attività Utente	Cc
_CM	Variabili struttura archivi Calendario dell'Agenda Attività Utente	Cc

esempi e casi dizionario

Il Calendario richiede l'attivazione del Customer Care. L'agenda a cui si riferisce è richiamabile sia da menu [Azienda \ Agenda] che dalla icona dell'Agenda Attività Utente visualizzata sotto la colonna dei tasti di tipo scheda. Per approfondimenti sulle funzionalità del Calendario si rimanda al manuale del Customer Care.

Esempi:

LETTURA_AGENDA:

```
AZZCATVARSYS "CM"

\ GETCM "ADMIN","20171005","",""           `Lettura agenda dell'utente ADMIN del 05/10/2017
\ GETCM "ADMIN","20170101","","20171004"   `Lettura agenda dell'utente ADMIN dal 01/01/2017 al 04/10/2017
GETCM "ADMIN", "20171002", "15477", ""     `Lettura agenda dell'utente ADMIN del 02/10/2017 con id specifico 15477

IF _ERRCM <> 0 THEN VIMSG 1,"ERRORE|Impossibile leggere l'agenda !|"+_ERRCM$
IFP _ERRCM=0
  FOR I = 1 TO _NCM
    LPRINT "-----"
    LPRINT "ID agenda: "+STR$( _CMNCRE(I),0,0)
    LPRINT _CMDATT$(I)
    LPRINT _CMUTE$(I)
    LPRINT _CMDATA$(I)
    LPRINT _CMSORA$(I)
    LPRINT _CMEORA$(I)
    LPRINT _CMDUR$(I)
    LPRINT _CMNOTA$(I)
    LPRINT _CMCART$(I)
    LPRINT _CMCCLI$(I)
  NEXT
ENDF
END
```

DELETE_AGENDA:

```
AZZCATVARSYS "CM"
DELCM "ADMIN","20171005","","20171006"    `Cancellazione agende dal 05/10/2017 al 06/10/2017
DELCM "ADMIN","20171004","15474","",""     `Cancellazione agenda specifica del 04/10/2017 con ID 15474
IF _ERRCM <> 0 THEN VIMSG 1,"ERRORE|Impossibile cancellare l'agenda !|"+_ERRCM$
END
```

LETTURA_SCRITTURA:

```
GETCM "A","25052016","",""
IF _ERRCM = 1 THEN VIMSG 1,_ERRCM$
FOR I = 1 TO _NCM
  PRINT " "
  PRINT I
  PRINT _CMDATT$(I)
  PRINT _CMUTE$(I)
  PRINT _CMSORA$(I)
  PRINT _CMEORA$(I)
  PRINT _CMDUR$(I)
  PRINT _CMNOTA$(I)
  PRINT _CMCART$(I)
  PRINT _CMCCLI$(I);: PRINT pades(_CMCCLI$(I))
  VIMSG 1,"GET"
  FOR I = 1 TO _NCM
    PUTCM
    IF _ERRCM = 1 THEN VIMSG 1,_ERRCM$
    VIMSG 1,"PUT "+STR$( _CMNCRE(I),0,0)
    VIMSG 1,"QUANTI "+STR$( _NCM,0,0)
  NEXT
```

COMMESSA (CO)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 23	Istruzione che azzerà variabili struttura commessa _CO	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "COMM"	Istruzione che azzerà variabili struttura _CO (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETCOMM	Par: <commessa>. Istruzione che legge i dati commessa Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
GETATTCOMM	Istruzione che legge i dati di Anagrafica Commessa presente a video	C
PUTCOMM	Istruzione che scrive i dati commessa Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
_ERRCO	Codice risultato operazioni commessa	S/Cd
_ERRCO\$	Messaggio operazioni su dati commessa	S/Cd
_CO	Variabili struttura archivi commessa	S/Cd
esempi e casi dizionario		
<p>Esempio: legge una commessa e le sostituisce la commessa sostitutiva</p> <pre> AZZVARSYS 23 : GETCOMM "COMM1A3" : IF _ERRCO\$<>" THEN VIMSG 4, _ERRCO\$ VIMSG 1, "Commessa Sostitutiva = " + _COANN\$: _COANN\$="COMM1A1" PUTCOMM : IF _ERRCO\$<>" THEN VIMSG 4, _ERRCO\$ ELSE GOTO MSG MSG: VIMSG 1, "Ho modificato con nuova Commessa Sostitutiva = " + _COANN\$ END </pre>		
Le cancellazioni si eseguono dai Servizi di Controllo di Gestione		

AREE (AE)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 24	Istruzione che azzerà variabili struttura aree _AE	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "AEREA"	Istruzione che azzerà variabili struttura _AE (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETAREA	Par: <area>. Istruzione che legge i dati aree Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
PUTAREA	Istruzione che scrive i dati aree Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
_ERRAE	Codice risultato operazioni su aree	S/Cd
_ERRAE\$	Messaggio operazioni su aree	S/Cd
_AE	Variabili struttura aree	S/Cd
ON_GETAREA:	Evento relativo alla lettura nella voce di menu (ACAR)	C/Cd
ON_PUTAREA:	Evento relativo alla scrittura nella voce di menu (ACAR)	C/Cd
esempi e casi dizionario		
<p>Esempio: legge un'area e le sostituisce la nota</p> <pre> AZZVARSYS 24 : GETAREA "1A00" : IF _ERRAE\$<>" THEN VIMSG 4, _ERRAE\$ VIMSG 1, "Nota dell'area = " + _AENOT\$: _AENOT\$="Nuova nota" PUTAREA : IF _ERRAE\$<>" THEN VIMSG 4, _ERRAE\$ ELSE GOTO MSG MSG: VIMSG 1, "Ho modificato con nuova Nota =" + _AENOT\$ END </pre>		
Le cancellazioni si eseguono dai Servizi di Controllo di Gestione		

MOVIMENTI DI ANALITICA (CDG)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 19	Istruzione che azzeri variabili movimenti di analitica _CDG	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "CDG"	Istruzione che azzeri variabili struttura _CDG (come azzvarsys)	S/C/Cr/Cd Sm/Cm
AZZCATVARSYS "CGE"	-Istruzione dismessa-	S/C/Cr/Cd Sm/Cm
GETCDG	Par: <TipoDoc\$>,<KeyDoc\$>,<Anno>,<NumRigaDoc>,<NumProgAna> Istruzione che legge dati movimenti di analitica Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
PUTCDG	Istruzione che scrive dati movimenti di analitica Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELCDG	Par: <TipoDoc\$>,<KeyDoc\$>,<Anno>,<NumRigaDoc>,<NumProgAna> Istruzione che cancella dati movimenti di analitica Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
ERRCDG	Codice risultato operazione movimenti di analitica	S/Cr/Cd/Sh
ERRCDG\$	Messaggio operazione movimenti di analitica	S/Cr/Cd/Sh
_CDGNR	Numero di righe lette di movimenti di analitica	S/Cr/Cd/Sh
CDG	Variabili struttura movimenti di analitica	S/Cr/Cd/Sh

Attenzione

1. le suddette istruzioni/variabili sono relative al solo al modulo [Azienda - Controllo di Gestione], non riguardano i movimenti analitici disponibili in modo standard nel modulo Primanota e rappresentati in variabili di struttura _PN; un utilizzo dei suddetti comandi senza il modulo Controllo di Gestione attivo rilascia il messaggio: "Istruzione non disponibile in questa installazione".
2. i suddetti comandi e relative variabili _CDG sono specifiche al modulo [Azienda - Controllo di Gestione - Contabilità Analitica - Inserimento/Revisione movimenti **extra**]; non vengono interessate e/o aggiornate le variabili di altri archivi (esempio _MM, _PN, ...), ovvero i moduli che avessero generato dei movimenti di analitica, pertanto, occorre procedere con le specifiche relative loro istruzioni (es. PUTMM, PUTPN, ...). La gestione dei movimenti di analitica del Controllo di Gestione equivale alla funzione standard "Inserimento movimenti extra", pertanto non dispone degli automatismi del gestionale che consentono di pilotare i movimenti di analitica da primanota, magazzino e produzione. Inoltre, la scrittura di movimenti analitici tramite Array ed una singola PUTCDG non è equivalente alla scrittura di un documento di magazzino (PUTMM) con tante righe, bensì occorre eseguire tante PUTCDG per quante sono le righe del documento che le generano (vedere gli esempi 1 e 2 sottostanti relativi alla "scrittura movimenti analitici da Ordine Cliente").
3. Il collegamento fra movimento di analitica e primanota è segnalato dalla lettera "c" nella descrizione della riga di primanota, ma solo se il movimento di analitica è generato dal modulo di primanota. Il collegamento è per conto di coge e non per riga di primanota. I movimenti di analitica generati successivamente con la PUTCDG, invece, sono di tipo extra, quindi non consentono tale collegamento, ma solo l'inserimento dei Riferimenti documento di primanota, in modo da avere un collegamento "logico" (verificabile anche generando il movimento direttamente dal modulo Inserimento movimenti analitici extra).
4. alcune tabelle di dizionario (esempio la tabella "be" COMMESSE: SALDI) sono predisposizioni per implementazioni future, pertanto non sono attive. La possibilità di ottenere le informazioni desiderate è consentita comunque dall'interrogazione della tabella dei movimenti di analitica.

Le chiavi di CdG.

Composizione della SKEY per movimenti Extra (stKey.cTipo = "E")						
Extra NON DEFINITO	num. progressivo					
stKey.cTipExtra = "0"	stKey.iProgExNocTipExtra					
Extra tipo CESPITE	conto CoGe cespite	codice cespite	anno acq. cespite			
stKey.cTipExtra = "C"	stDoc.iCespCoGe	stDoc.sCespCod	stDoc.iCespAnno			
Extra tipo GENERICO	serie doc	numero doc	data doc			
stKey.cTipExtra = "G"	stDoc.iSerDoc	stDoc.iNumDoc	stDoc.dDtDoc			
Extra tipo MAGAZZINO	sigla doc	serie doc	numero doc	data doc		
stKey.cTipExtra = "M"	stDoc.iSigDoc	stDoc.iSerDoc	stDoc.iNumDoc	stDoc.dDtDoc		
Extra tipo PRIMANOTA	causale reg	tipo protocollo	serie protocollo	numero protocollo	data reg	numero reg
stKey.cTipExtra = "P"	stDoc.iCauPm	stDoc.cProt	stDoc.iProtS	stDoc.iProtN	stDoc.dDtRegPm	stUda.iNumDocPm

Dal modulo standard Revisione Movimenti di Contabilità Analitica si gestiscono tutti i tipi di movimento: Extra NON DEFINITO, Extra da Primanota, Extra da Magazzino, Extra da Cespiti, ecc..

Dal modulo standard Inserimento Movimenti Extra non si può generare più movimenti Extra NON DEFINITO collegati (effettivamente) a un documento reale come se fosse un Extra proveniente da un modulo in grado di generarlo e gestirlo con relativa chiave e determinate regole (per esempio, in magazzino c'è un movimento analitico disponibile su ciascuna riga del documento, in primanota ci sono quattro movimenti analitici disponibili per ciascuna contropartita).

Qualsiasi movimento Extra che non sia generato da Contabilità o da Magazzino o da altro modulo che ne sia capace, viene considerato di tipo Extra NON DEFINITO (cioè Extra in "senso stretto"); nel caso che questi movimenti fossero fra loro identici l'ulteriore chiave è `_CDGPROGETX(N)` e non viene resa disponibile all'utente, in quanto è una chiave di sola lettura generata dal Gestionale.

Ignorare la suddetta regola inserendo più Extra NON DEFINITI e associandoli al tipo di movimento (es: `tipreg$="E"` e `tipext$="M"`) e relativi dati del documento (sig, ser, num, ecc.), significa accodarli in un corpo unico, anche se i movimenti analitici non sono tutti uguali, pertanto non sarebbero selezionabili in caso di `DELCDG`, la quale eliminerebbe tutti i `_CDGNR` elementi "associati" al tipo movimento.

Il modulo standard Revisione Movimenti di Contabilità Analitica, invece, è in grado di selezionare i movimenti da eliminare poichè si basa su una gestione a video dei `_CDGNR` elementi (non c'è un ulteriore porzione di chiave). Per soddisfare questa modalità di gestione degli analitici extra non definiti, ma associati al tipo movimento, occorrerebbe un apposito progressivo.

Esempi e casi dizionario

Esempio Sprix: letture e cancellazioni di movimenti di analitica

```
GETCDG "R", "PNDRE=20060906;PNND0=19; ", 2006, 2, 0
DELCDG "R", "PNDRE=20060906;PNND0=19; ", 2006, 2, 1
GETCDG "M", "MMSIG=SL;MMNUM=7;MMSER=1;MMCLI=501.1;NTMM=1; ", 0, 0, 0
DELCDG "M", "MMSIG=FT;MMNUM=7;MMSER=1;MMCLI=501.1;NTMM=1; ", 0, 1, 0
GETCDG "C", "MMSIG=OC;MMNUM=1;MMSER=1", 0, 0, 0
GETCDG "F", "MMSIG=OF;MMNUM=1;MMSER=1", 0, 1, 0
GETCDG "E", "PRGEXTRA=;DTCOMP=20081231;AREA=A;COMM=C;CODANL=C1;TIPOV=M; ", ...
DELCDG "E", "PRGEXTRA=;DTCOMP=20080909;AREA=A;COMM=C;CODANL=C1;TIPOV=G; ", ...
```

Esempio 1: scrittura movimenti da Ordine Cliente, ma nell'Ordine resta collegata solo la prima riga al movimento di analitica

```
' Lettura Documento e Test Errore
GETMM "OC1/" + NUM_DOC$ : IF _ERRMM<>0 THEN VIMSG 4, _ERRMM$ : END

' Cancellazione movimenti analitica del Documento
NUM_DOC$="MMSIG=OC;MMNUM="+STR$(MMNUM,0,0)+" ;MMSER="+STR$(MMSER,0,0)+" ;"
DELCDG "C", M_DOCUS$, 0, 0, 0 : IF _ERRCDG<>0 THEN VIMSG 4, _ERRCDG$

' Riferimenti Documento
SIGLA_DOC$ = MMSIG$ : SERIE_DOC = MMSER : NUMERO_DOC = MMNUM
DATA_DOC$ = _MMDAT$ : CLI_DOC$ = _MMCLI$

' Azzeramento e Inizializzazione variabili
AZZVARSYS 19 : M_RDOC=0

' Ciclo righe documento per creazione movimenti di analitica
FOR RIG_DOC=1 TO _NMM
  RIG_CDG=M_RIGCDG+1 ' Nota bene: la variabile d'appoggio è solo un richiamo al tipo di movimento, che in magazzino è
                    ' sempre 1 (una riga di movimento = una riga di analitica), mentre in primanota (P_) va da 1 a 4,
                    ' cioè una contropartita = fino a 4 righe di analitica;

  _CDGTIPREG$(RIG_CDG) = "C" ' tipo
  _CDGTIPEXT$(RIG_CDG) = "M" ' generico
  _CDGNS$(RIG_CDG) = "S" ' Considera movimento su CDG S/N
  _CDGDDTCOMP$(RIG_CDG) = M_DATDOC$ ' data competenza
  _CDGSCODANL$(RIG_CDG) = "R01001002" ' codice analitico
  _CDGSCOMM$(RIG_CDG) = "COM1" ' commessa
  _CDGSAREA$(RIG_CDG) = "1010" ' area
  _CDGCTOCG$(RIG_CDG) = arric( MMART$(RIG_DOC) ) ' Codice Contropartita righe dei movimenti
  _CDGCODARMO$(RIG_CDG) = _MMART$(RIG_DOC) ' Codice Articolo Movimento
  _CDGSIGDOC$(RIG_CDG) = SIGLA_DOC$ ' Sigla Documento di magazzino
  _CDGISERDOC$(RIG_CDG) = SERIE_DOC ' Serie Documento di magazzino
  _CDGLNUMDOC$(RIG_CDG) = NUMERO_DOC ' Numero Documento di magazzino
  _CDGDDTDOC$(RIG_CDG) = DATA_DOC$ ' Data Documento di magazzino
  _CDGCTOT$(RIG_CDG) = CLI_DOC$ ' Conto Cliente/Fornitore Documento
  _CDGNTEST$(RIG_CDG) = 1 ' testata documento
  _CDGPROGRIG$(RIG_CDG) = _MMNCRERIGA(RIG_DOC) ' riga documento
  _CDGPROGANA$(RIG_CDG) = RIG_CDG ' riga di analitica
NEXT

' creazione movimenti analitica (notare: la PUTCDG è fuori dal ciclo ed è unica di tanti movimenti)
PUTCDG : IF _ERRCDG=0 THEN GOTO FINE ELSE VIMSG 4, _ERRCDG$

FINE:
END
```

Esempio 2: scrittura movimenti da Ordine Cliente e ciascuna riga del documento ha il rispettivo movimento di analitica

```
' Lettura Documento e Test Errore
GETMM "OC1/" + NUM_DOC$ : IF _ERRMM<>0 THEN VIMSG 4, _ERRMM$ : END

' Cancellazione movimenti analitica del Documento
NUM_DOC$="MMSIG=OC;MMNUM="+STR$(MMNUM,0,0)+" ;MMSER="+STR$(MMSER,0,0)+" ;"
DELCDG "C", M_DOCUS$, 0, 0, 0 : IF _ERRCDG<>0 THEN VIMSG 4, _ERRCDG$

' Riferimenti Documento
SIGLA_DOC$ = MMSIG$ : SERIE_DOC = MMSER : NUMERO_DOC = MMNUM
DATA_DOC$ = _MMDAT$ : CLI_DOC$ = _MMCLI$

' Azzeramento e Inizializzazione variabili
AZZVARSYS 19 : M_RDOC=0

' Ciclo righe documento per creazione movimenti di analitica
FOR RIG_DOC=1 TO _NMM
  RIG_CDG=M_RIGCDG+1

  AZZVARSYS 19
  _CDGTIPREG$(RIG_CDG) = "C" ' tipo
  _CDGTIPEXT$(RIG_CDG) = "M" ' generico
  _CDGNS$(RIG_CDG) = "S" ' Considera movimento su CDG S/N
```

```

_CDGDDTCOMP$(RIG_CDG) = M_DATDOC$           ' data competenza
_CDGSCODANL$(RIG_CDG) = "R01001002"        ' codice analitico
_CDGSCOMM$(RIG_CDG) = "COM1"                ' commessa
_CDGSAREA$(RIG_CDG) = "1010"                ' area
_CDGCTOCG$(RIG_CDG) = arric( MMART$(RIG_DOC)) ' Codice Contropartita righe dei movimenti
_CDGCODARMO$(RIG_CDG) = MMART$(RIG_DOC)     ' Codice Articolo Movimento
_CDGSIGDOC$(RIG_CDG) = SIGLA_DOC$           ' Sigla Documento di magazzino
_CDGISERDOC$(RIG_CDG) = SERIE_DOC           ' Serie Documento di magazzino
_CDGLNUMDOC$(RIG_CDG) = NUMERO_DOC          ' Numero Documento di magazzino
_CDGDDTDOC$(RIG_CDG) = DATA_DOC$          ' Data Documento di magazzino
_CDGCTOT$(RIG_CDG) = CLI_DOC$              ' Conto Cliente/Fornitore Documento
_CDGNTEST$(RIG_CDG) = 1                     ' testata documento
_CDGPROGRIG$(RIG_CDG) = MMNCRERIGA(RIG_DOC) ' riga documento
_CDGPROGANA$(RIG_CDG) = RIG_CDG            ' riga di analitica

' creazione movimenti analitica (ma con tante PUTCDG per quante sono le MMNCRERIGA del documento)
' la PUTCDG è dentro il ciclo

PUTCDG : IF _ERRCDG=0 THEN GOTO FINE ELSE VIMSG 4, _ERRCDG$

NEXT

FINE:
END

```

Caso: valorizzazioni degli array _CDGRDOCSOS e _CDGPROGRIG

Le variabili (array) _CDGRDOCSOS (che rappresenta il numero primo record assegnato (da 1 a 16777214)) e la variabile _CDGPROGRIG (che rappresenta il numero di riga documento) si intendono riferite al valore di RECORD di riga (ricavabile dalla variabile MMNRECRIGA(1-N) tramite l'istruzione GETMM) e vanno quindi impostate in relazione a questa variabile quando si effettuano delle registrazioni di movimenti di controllo di gestione da associare a documenti di magazzino tramite l'istruzione PUTCDG.

Caso: bmimp() valore ad intervallo di date

Il dizionario bmimp() non considera l'intervallo di date come effettua il gestionale. Il dizionario rilascia il valore solo se viene indicata nel parametro la data precisa indicata nella tabella. Per date interne all'intervallo la bmimp() rilascia blank (vuoto)

DOCUMENTI DOCUVISION (DV)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 20	Istruzione che azzerava variabili documenti docuvision _DV	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "DVDOC"	Istruzione che azzerava variabili struttura _DV (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETDVDOC	Par: <numdoc>, <numrev>, [[DIRLOC];[GETPH]]. Istruzione che legge documenti docuvision. Per la Fatturazione Elettronica XML consultare l'annotazione relativa a [[DIRLOC];[GETPH]]. Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
PUTDVDOC	Istruzione che scrive documenti docuvision Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELDVDOC	Par: <numdoc>, <numrev>. Istruzione che cancella documenti docuvision Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
ERRDV	Codice risultato operazione documenti docuvision	S/Cd
ERRDV\$	Messaggio operazione su documenti docuvision	S/Cd
DVDNUMVER	Numero totale revisioni documenti docuvision	S/Cd
DVDPAGXVER	Numero di pagine per revisione	S/Cd
_DV	Variabili di struttura documenti docuvision	S/Cd

La gestione dell'archiviazione documentale non dispone degli stessi automatismi dei moduli standard. Il collegamento fra i documenti archiviati e le transazioni dei moduli standard (primanota, magazzino, produzione) va eseguita con i comandi Sprix di Associazione fra il Gestionale e Docuvision.

Per quanto riguarda le Associazioni Docuvision sono disponibili le seguenti istruzioni per gestire il database Docuvision: PUTDVDOC / GETDVDOC / DELDVDOC (inserisce/aggiorna, legge o elimina i dati del Documento Docuvision relativi alle variabili struttura archivi _DV) e PUTASSOCDV / GETASSOCDV / DELASSOCDV (inserisce/aggiorna, legge o elimina i dati delle associazioni fra il Documento Docuvision e l'Entità del Gestionale, relativi alle variabili di struttura archivi _AD).

Attenzione: la funzionalità è attiva per il modulo "Gestione documento", ma non per il modulo "Fatture elettroniche PA-B2B".

Le modifiche devono rispettare i vincoli di gestione del documento e delle revisioni per come impostati nei Parametri Generali Docuvision:

- L'elaborazione è solo per l'azienda aperta: non sono gestiti i documenti di altre aziende eccetto quelli senza sigla (utilizzabili da tutte le aziende); la scrittura verifica che l'azienda specificata sia quella dell'azienda aperta (diversamente il documento è disponibile per tutte le aziende) e la scrittura del documento con sigla diversa da quella aperta non è consentita (la sigla azienda documento è automatica con quella dell'azienda aperta). L'inserimento di nuove pagine o la modifica di una revisione esistente, richiede variabili correttamente impostate: la scrittura non avviene se non si associa almeno una corretta revisione in 'senso lato' (cioè anche se è il 1° inserimento e senza che siano mai avvenute revisioni in 'senso stretto'). Notare: ulteriori informazioni sulle revisioni sono in calce alle variabili _DV
- Non è inseribile una revisione/pagina nuova o aggiornabile una revisione/pagina del documento, senza contemporaneamente scrivere anche il documento impostandone le variabili, anche se i dati del documento sono letti e non modificati.
- La revisione del documento è fino alla prima in cui non sono specificate le variabili _DVVNUM (aggiornamento revisione esistente) e _DVVFILEPUT\$ (con Id revisione a zero, ossia nuova revisione); in ingresso non è letta la _DVDNUMVER, ma viene calcolata.
- Nell'aggiornamento di un documento indicato come non modificabile (dai Parametri Generali di Docuvision) i valori impostati nelle variabili devono essere identici a quelli presenti sul database, compresa la sezione dei campi aggiuntivi (se previsti).
- La scrittura del documento nuovo senza specificare gli eventuali campi aggiuntivi collegati, genera il documento con campi aventi valori di default se alcuni di questi sono previsti con valori di default (anche in modo non obbligatorio) dalla classe del documento, altrimenti non li genera (se nessuno ha valori di default); se il documento non è nuovo e qualche campo è obbligatorio, la loro non specificazione genera errore, altrimenti, se nessuno è obbligatorio, vengono cancellati.

Nota: [[DIRLOC];[GETPH]] consente la copia immediata del file allegato senza doverlo visualizzare da Gestionale.

Per Fatturazione Elettronica XML la copia immediata è consentita se il **Servizio** "Fattura XML - Sistema ricevente - Conservazione digitale" (vedere [Shift+F11] in Anagrafica Azienda) è configurato per essere eseguito in **autonomia**, altrimenti, occorre eseguire una preventiva visualizzazione del documento allegato.

Infatti, se il servizio di invio della fattura elettronica allo SDI è un servizio Passepartout (All Inclusive o Firma + Invio Ricezione), il file ".xml" corrispondente alla fattura appena inviata (es. IT04345180378_00001.xml) viene **cancellato** e non è presente sul disco locale (e quindi non consentirebbe la copia [DIRLOC]) fino a che non si visualizza la fattura stessa dalla gestione documento ed effettuata **dopo** che il servizio PassHub abbia firmato il file (es. IT04345180378_00001.xml.p7m). La visualizzazione del documento, infatti, esegue la richiesta a PassHub il quale ne restituisce una copia che viene resa disponibile **anche** per l'operazione di copia [DIRLOC].

In tale secondo caso, quindi, per forzare questa richiesta a PassHub al fine di scaricare il file Xml nella [DIRLOC], se disponibile e/o consentito da PassHub, è disponibile un ulteriore elemento da associare nel parametro [DIRLOC], ovvero **[GETPH=SI]**, che significa "prendi il file Xml da PassHub". (dalla v711).

In caso di utilizzo di [GETPH=SI] è **sempre bene specificare la [DIRLOC], ovvero, un percorso locale ed accessibile** (per esempio "<C:\Temp>"), altrimenti, se [DIRLOC] fosse null, il download del file avverrebbe nella "C:\mexalbp\dati\temp\fattxml001\", ovvero, una directory temporanea ad uso esclusivo del Gestionale in cui il numero, in questo caso "001", rappresenterebbe il numero di terminale che ha eseguito la GETDVDOC e che, a fine esecuzione del processo Sprix (mx45.exe), **sarebbe ripulita la cartella** "fattxmlINN" (dato che si tratta di una "temp") e che necessiterebbe, quindi, di una immediata COPYFILE locale, ma che però il programmatore **non può** fare in quanto il percorso non è accessibile.

Il file che viene scaricato è in formato ".xml" oppure ".xml.p7m" (se la fattura transitata dallo SDI è quella firmata). In questo secondo caso è possibile decodificare in semplice Xml attraverso la funzione **DECODP7M()**.

Nota Bene: se il documento è già presente e disponibile in Docuvision, la GETPH non preleva da PassHub, bensì preleva il documento disponibile in Docuvision; la ripetizione dell'operazione GETPH va in sovrascrittura in base al nome del file e senza alcun avvertimento. In ogni caso si consiglia di verificare i file attraverso LEGGIDIR e/o ESISFILE.

Esempio:

AZZVARSYS 20

FOR ID=1 TO 16

GETDVDOC ID,-1, "DIRLOC=<C:\tmpd\;GETPH=SI" : IF _ERRDV<>0 THEN VIMSG 4, _ERRDV\$

VIMSG 1, "Num: "+STR\$(ID,0,0)+" Clas: "+STR\$(_DVDCLASS,0,0)+" Tit: "+_DVRTIT\$+ " Prot: "+_DVDPROTO\$+" Cont: "+_DVCNT\$+" DatDoc: "+_DVDDATA\$+" DatPro: "+_DVDDATPRO\$

NEXT

END

Nota su ALLEGATI

Sprix gestisce gli allegati relativi al Documento di Docuvision (fig. 1), non gestisce gli allegati relativi alla Fattura PA / B2B (fig. 2); il modo per intervenire sul file XML (della fattura elettronica) è consentito all'evento di stampa ON_STP_XML, ma non sono disponibili apposite variabili per codificare l'allegato dentro il file XML poiché occorre un'apposita codifica separata (fig. 2). Si ricorda che altri generici file XML sono gestiti in Sprix tramite apposite istruzioni XML. Per ottenere l'handle del documento in considerazione al fine di eseguirne le opportune modifiche con le istruzioni generali XML, avvalersi della variabile _FTE_DOCXML che consente l'accesso da collage al file _XML della FTE.

Fig. 1 (Allegati DOCUVISION)

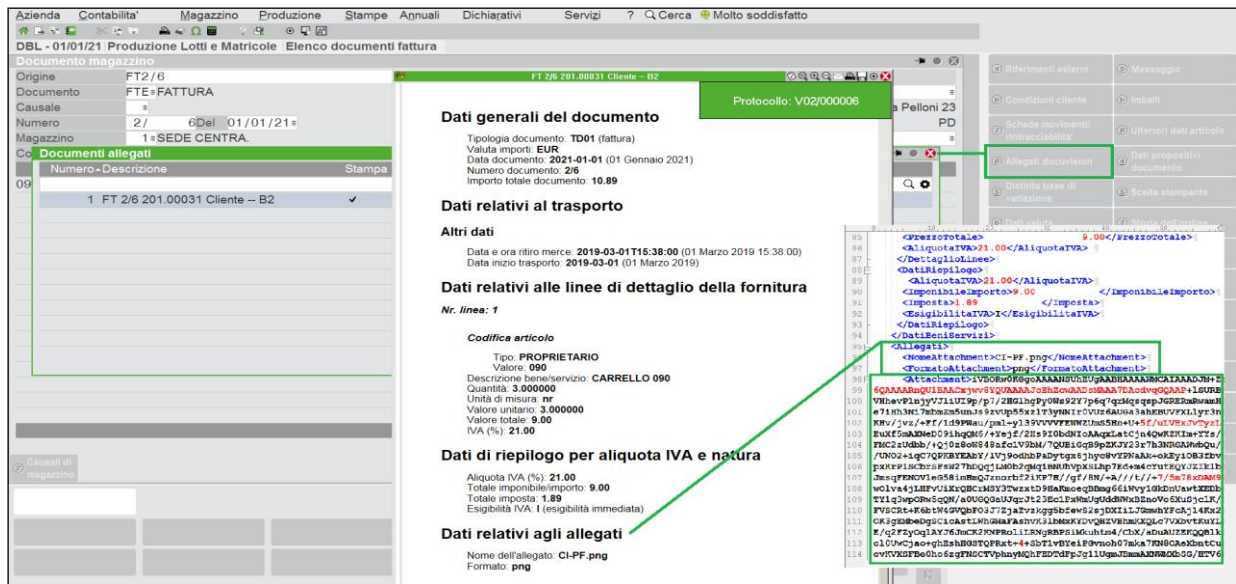
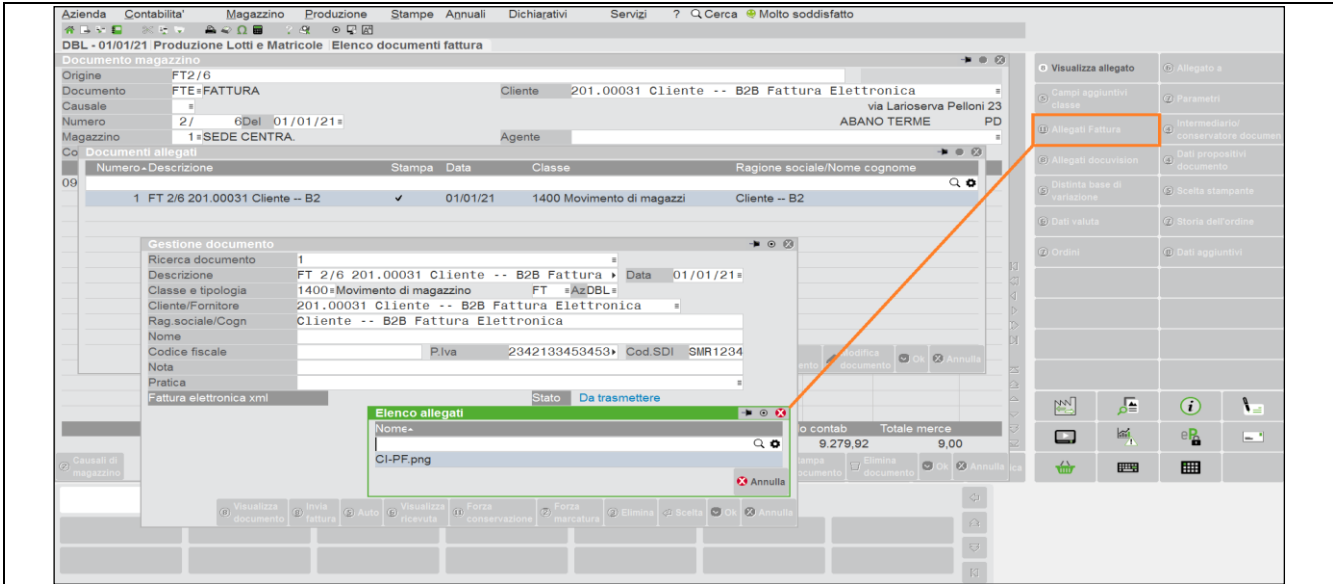


Fig. 2 (Allegati FATTURA)



Esempi e casi dizionario

Esempio: crea nuovo documento sovraziendale (ricercabile in archivio senza azienda) con 2 nuove revisioni; legge e modifica le note delle revisioni (NB: Parametri Docuvison = gestione revisioni attiva e modifica di documento fino ad archiviazione). La lettura delle pagine e la cancellazione non sono attive (vanno impostati i parametri).

```

_DVDID=0 : _DVVNUM(1)=0 : _DVVNUM(2)=0
_DVDCLASS=1400 : _DVDTIT$="Titolo classe" : _DVDDATA$=sxdat
_DVVNOTA1$(1)="nota rev.1":_DVVNOTA2$(1)="ult.nota rev.1"
_DVVFILEPUT$(1)="c:\file1.txt"
_DVVNOTA1$(2)="nota rev.2":_DVVNOTA2$(2)="ult.nota rev.2"
_DVVFILEPUT$(2)="c:\file2.txt"
_PUTDVEDOC : IF _ERRDV<>0 THEN VIMSG 4, _ERRDV$

ID= _DVDID
VIMSG 1,"ha creato un Documento sovraziendale n. "+STR$(ID,0,0)

AZZVARSYS 20
GETDVEDOC ID,-1: IF _ERRDV<>0 THEN VIMSG 4, _ERRDV$
FOR I=1 TO _DVVDNUMVER 'numero globale revisioni
  VIMSG 1,"rev.glob.: "+STR$( _DVVNUM(I),0,0)+" n.rev.: "+STR$( _DVVDNUMVER,0,0)
  VIMSG 1, STR$( _DVDPAGXVER(I),0,0) 'gestione pagine
  VIMSG 1, "DVVNFAG "+STR$( _DVVNFAG(I),0,0) 'gestione pagine
  VIMSG 1, "nome file pagina "+_DVVFILEGET$(I) 'gestione pagine
NEXT

_DVVNOTA1$(2)="NOTA REV.2": _DVVNOTA2$(2)="ULT NOTA REV.2"
_PUTDVEDOC : IF _ERRDV<>0 THEN VIMSG 4, _ERRDV$
'DELDVEDOC 3,0: IF _ERRDV<>0 THEN VIMSG 4, _ERRDV$ 'cancella intero doc.
END
    
```

Esempio:

```

_DVDID = 0
_DVDCLASS = 600
_DVDTIT$ = "TEST"
_DVDDATA$ = sxdat
_DVDPROTO$ = PROT$
_DVDDATPRO$ = DATA$
_DVVESTFIL$(1) = "file"
_DVVFILEPUT$(1) = "<C:\File.txt"
_DVVVAL$(1) = "file.txt"
_DVVNOTA1$(1) = "file.txt"

_DVAVAL$(1) = "xxxxx" ' _DVAVAL$( ) può essere letta e scritta (è il valore del campo aggiuntivo)
' [ _DVANOMECOL$( ) e _DVANCAMPI( ) definiscono la struttura dei campi aggiuntivi ]
' dato il nome colonna CDS_PN

_DVANOMECOL$(1)="CDS_PN" 'letta ma non scritta (va inizializzata alla PUT per specificare quale campo aggiuntivo e ciò che passi)
_DVANCAMPI = 1 ' può solo essere letto

_PUTDVEDOC : IF _ERRDV<>0 THEN VIMSG 4, _ERRDV$
ID= _DVDID
VIMSG 1,"creato documento n. "+STR$(ID,0,0)
    
```

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 25	Istruzione che azzerà variabili associazione gestionale-docuvision _AD	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "ASSOCDV"	Istruzione che azzerà variabili struttura _AD (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETASSOCDV	Par: <tipo>,<keypass>.<iddoc> Istruzione che legge associazione gestionale-docuvision Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
PUTASSOCDV	Par: <tipo>,<keypass>.<iddoc>,<alleg> Istruzione che scrive associazione gestionale-docuvision Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELAASSOCDV	Par: <tipo>,<keypass>.<iddoc> Istruzione che cancella associazione gestionale-docuvision Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
_ERRAD	Codice risultato operazione associazione gestionale-docuvision	S/Cd
_ERRAD\$	Messaggio operazione su associazione gestionale-docuvision	S/Cd
_ADNUMASSOC	Numero totale associazioni gestionale-docuvision	S/Cd
_AD	Variabili struttura associazione gestionale-docuvision	S/Cd

Attenzione: la funzionalità è attiva per il modulo "Gestione documento", ma non per il modulo "Fatture elettroniche PA-B2B".

esempi e casi dizionario

Per conoscere quali documenti sono associati al record della registrazione (anagrafica, primanota, magazzino, articolo, ecc...) si deve leggere la registrazione e poi cercare quali associazioni ha. Per esempio, un documento di magazzino ha un documento docuvision di cui si vuole conoscere l'ID univoco

```
GETMM "FT1/2084" : GETASSOCDV 1400,"FT;1;2084;_MMCLI$;_MMDAT$",0 : VIMSG 1, STR$(_ADNUMASSOC,0,0) + " " +STR$(_ADDVID(1),0,0)
```

Si consiglia di concatenare i parametri riassumendoli in un'unica variabile stringa di appoggio. Esempio: fattura emessa, sezionale 1, numero documento 10, cliente 501.00005, del primo marzo 2019. VAR\$="FT;1;10;501.00005;20190301" oppure
VAR\$=_MMSIG\$ "+" + STR\$(_MMSER,0,0)+" ; "+STR\$(_MNUM,0,0)+" ; "+_MMCLI\$+" ; "+_MMDAT\$: GETASSOCDV 1400,VAR\$,0

Esempi: registrazioni di associazioni Documento (Gestionale - Docuvision)

registrazione di associazione documento Matrice del Gestionale al documento Docuvision

```
AZZVARSYS 25
  _ADMMSIG$(1)="MA": ADMMSER(1)=1: ADMNUM(1)=290
  _ADMMDAT$(1)="01012021": ADMMCLI$(1)="201.00028"
PUTASSOCDV 1500,"MA;1;290;201.00028;20210101",1
  IF _ERRAD<>0 THEN VIMSG 4, _ERRAD$
```

```
AZZVARSYS 25
  _ADMMSIG$(1)="FD": ADMMSER(1)=1: ADMNUM(1)=16
  _ADMMDAT$(1)="12012009": ADMMCLI$(1)="501.00002"
PUTASSOCDV 1400,"FD;1;16;501.00002;20090112",1
  IF _ERRAD<>0 THEN VIMSG 4, _ERRAD$
```

```
`Letture
`GETASSOCDV 600,"20210101;1;FE;V;1;0;0;20210101",1 `(primanota)
`GETASSOCDV 600,"20070101;7;FE;V;1;2;2;20070101",0 `(primanota)
`GETASSOCDV 1100,"FA;1;1;501.00001;20061001",5 `(notula)
`GETASSOCDV 1200,"Art",4 `(articolo)
`GETASSOCDV 1400,"FT;1;27;501.00001",6 `(movimento)
```

Esempio: cambio di allegati Generici (associati ad "ART") in allegati immagine:

```
GETASSOCDV 1200,"ART",0
FOR X=1 TO ADNUMASSOC
PUTASSOCDV 1200,"ART1",_ADDVID(X),"FPALL=I;PRGALL="+STR$(X,0,0)
NEXT
```

Esempio di associazione ciclica in docuvision, di conti, estratto conto, ecc...

```
+-----+
|Cod.Tab.:ps CONTABILITA' - SCADENZARIO NO PARTITE
|Ordine 3:pscto()
+-----+
ps | ESTR$="c:\file.txt" : CRLF$=CHR$(13)+CHR$(10) : C$=pscto()
ps | FOR I=1 TO 10 : TOT=TOT+pspm(I) : NEXT
ps | IF ESISFILE(ESTR$) THEN KILL ESTR$
ps | NF=AUTOFILE() : OPEN "R",NF,ESTR$,32 : FIELD NF,30 AS REC$,2 AS CRLF$
ps | REC$="Estratto " + pscto() + " " + STR$(TOT,2,0) : PUT NF,1 : CLOSE NF
ps | _DVID=0 : _DVVNUM(1)=0 : _DVVNUM(2)=0
ps | _DVDCCLASS=400 : _DVDTIT$="estratto "+pscto() : _DVDDATA$=sxdat
ps | _DVVFILEPUT$(1)="c:\file.txt"
ps | PUTDVDOC : IF _ERRDV<>0 THEN VIMSG 4, _ERRDV$
ps | PUTASSOCDV 400,pscto(), _DVID
```

```
+-----+
|Cod.tab.:pn CONTABILITA' - PRIMA NOTA INTESAZ.
|Ordine 3:pnprn()
+-----+
pn DREG$=pn dre()
pn PRINR=pn prn()
pn CAUS$=pn cau()
pn PROTR$=pn reg()
pn PROTS$=pn sdo()
pn PROTN$=pn pro()
```


Esempio: Associazione di un'entità Allegato Documentale all'entità Anagrafica MyDB, relativa all'Anagrafica Articolo di Magazzino.

Il seguente codice è uno Sprix in quanto è Sprix che è in grado di scrivere; in caso di Collage occorre eseguirlo con ShellSprix (sempre che non si incontrino semaforiche sul record) oppure simulando l'operatore con tante puttast, se consentito.

Prima di eseguire il seguente programma dimostrativo, da Gestionale è stato allegato un qualsiasi documento ad un record di Anagrafica MyDB (MYDB2) relativo ad un articolo "ARTICOLO1"; il documento allegato è visibile sia dall'archivio anagrafico MYDB dell'App, sia dalla Gestione Documento di Docuvision. Poi è stato creato un secondo record di Anagrafica MYDB per l'articolo "ARTICOLO2", ma senza allegare alcunché (perché lo farà SPRIX).

Successivamente, il seguente SPRIX copia l'associazione dell'allegato (cioè il primo record MYDB relativo al primo articolo "ARTICOLO1") per eseguire quindi una nuova associazione, con un nuovo allegato "]Cartel2.xlsx" relativo al secondo articolo "ARTICOLO2".

La logica di base è che c'è una relazione fra l'ENTITA' del Gestionale (MYDB), l'entità Documentale (Documento allegato) e la relativa Associazione fra le due entità.

```
AZZCATVARSYS "MYDB"
MYDBGET "977700mydb_MYDB2",0,5 : VIMSG 4, _ERRMYDB$          '---> leggo il record MYDB di ARTICOLO1
GETDVDOC 98,0,"" : VIMSG 4, _ERRDV$                          '---> leggo il documento allegato di ARTICOLO1
GETASSOCDV 1700,"977700mydb_MYDB2,"+STR$(_MYDBID,0,0),_DVIDID '---> leggo l'associazione fra le due precedenti entità

AZZCATVARSYS "DVDOC"                                         '---> azzero e inizializzo per nuovo documento allegato
_DVIDID=0:_DVENUM(1)=0:_DVIDCLASS=1700:_DVIDTIT$="Titolo classe":_DVIDDATA$=sxdat
_DVVNOTA1$(1)="notal":_DVVNOTA2$(1)="ult.notal":_DVVFILEPUT$(1)="]Cartel2.xlsx"
PUTDVDOC : VIMSG 4, _ERRDV$                                  '---> scrivo la nuova entità (nuovo documento allegato)

MYDBGET "977700mydb_MYDB2",0,6 : VIMSG 4, _ERRMYDB$          '---> leggo il record MYDB di ARTICOLO2 (nuova entità)
KEY$="977700mydb_MYDB2;" +STR$(_MYDBID,0,0)                  '---> compongo chiave per scrivere nuova associazione
PUTASSOCDV 1700,KEY$,_DVIDID,"TPALL=" : IF _ERRAD<>0 THEN VIMSG 4, _ERRAD$ '---> associo le due nuove entità
```

PIANO DEI CONTI – RUBRICA UNICA (PC)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 1	Istruzione che azzera variabili piano dei conti _PC	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "PC"	Istruzione che azzera variabili struttura _PC (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETRUBUNICA	Par: <id>. Istruzione che legge i dati della Rubrica Unica Non consentita nei Livelli di sviluppo (F2, L3, F3) Nota: la replica MexalDB è un concetto completamente aziendale; le tabelle generali (raggruppamenti, rubrica unica rudt, banche, ecc) sono database aziendali, pertanto trattati per ciascuna singola azienda. Es: nelle aziende AAA e BBB con rudt e MexalDB attivo automatico la replica iniziale di entrambe le aziende crea le tabelle AAA_RUDT in "aaa_rp" e BBB_RUDT in "bbb_rp", ma l'aggiornamento del dato di rubrica unica nell'azienda AAA aperta allinea solamente la tabella AAA_RUDT (non la tabella BBB_RUDT) e viceversa; anche in caso di aggiornamento senza alcuna azienda aperta non allinea alcunchè, perchè MexalDB non è attivo. Non è possibile aggiornare una tabella del database bbb_rp tramite l'azienda AAA aperta/chiusa e viceversa. Per allineare le tabelle delle altre aziende, basta aprire l'azienda ed in Servizi - Riorganizzazione archivi aziendali selezionare RUBRICA PC/CLIENTI/FORNITORI e premere F8 (Elabora solo MexalDB), oppure aprire la tabella di raggruppamento e riconfermare un valore.	S/Cr/Cd
GETPC	Par: <codcon\$>. Istruzione che legge anagrafica piano dei conti	S/Cr/Cd/Sm/Sh
GETPC2	Par: <indice\$>, <dato\$>, [tipoconto\$], [tiporicerca\$], [finestra\$], [conto\$] Istruzione che ricerca/legge il conto per Descrizione, P.Iva, Cod. Fisc. Variabili collegate: _CODPICONR, _CODPICO\$() _ERRCODPICO\$	S/Cr/Cd/Sm
GETATTPC	Istruzione che legge il conto contabile a video	C
PUTPC	Istruzione che scrive l'anagrafica del conto nel piano dei conti. Note: nella tecnologia BP, le Aziende che hanno la gestione della RUBRICA UNICA CLI/FOR, se il parametro "All.aut.cli/for rub.unica" dei parametri contabili è impostato ad "S" possono utilizzare PUTPC per modificare e inserire conti di tipo Clienti/Fornitori e per i conti patrimoniali generali (PG) marcati "deposito Bancario" (Ban); se il parametro "All.aut.cli/for rub.unica" dei parametri contabili è impostato ad "N" non si possono inserire o modificare i conti.	S/Cr/Cd/Sm/Sh
DELPC	Par: <codcon\$>. Istruzione che cancella anagrafica piano dei conti	S/Cr/Cd/Sm/Sh
CTRPIVA()	Par: (<partiva\$>). Funzione per il controllo di Partita Iva	S/C/Cr/Cd/Sh
CTRCODFIS()	Par: (<codfis\$>). Funzione per il controllo del Codice Fiscale	S/C/Cr/Cd/Sh
CREACF	Par: [cdfcogn\$], [cdfnome\$], [cdfdatn\$], [cdfsesso\$], [cdfcdcom\$], [cdfdscom\$], [cdfprov\$], [cdfiscale\$] Funzione creazione/scorporo del Codice Fiscale	S/Cd
_CDF	Variabili struttura archivi creazione/scorporo Codice Fiscale	S/C/Sm/Cm/Cd
GETESPOS	Par: <cod\$>, <data\$>. Istruzione di lettura esposizione in primanota Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/C/Cr/Cd
ESPOSIZERR\$	Messaggio relativo all'istruzione GETESPOS	S/C/Cd
ERRPC	Codice risultato operazione su piano dei conti	S/C/Cd
_ERRPC\$	Messaggio operazioni su piano dei conti	S/C/Cd
ERRUBUNI	Codice risultato istruzione GETRUBUNICA	
ERRUBUNI\$	Messaggio istruzione GETRUBUNICA	
PC	Variabili di struttura piano dei conti	S/C/Cd
ON_STARTPC:	Evento di Anagrafica del Piano dei Conti (voce menu CG) richiede uscita dal primo campo	C/Cd
ON_STARTPC: ON_PUTPC: ON_DELPC:	Disponibili nelle seguenti funzioni di Variazioni Contabilità del menu SERVIZI: Anagrafiche piano dei conti; Condizioni clienti/fornitori; Codici bilancio CEE; Conversione da codici bilancio CEE a codici UE;	C/Cd
esempi e casi dizionario		
pcage(pccod())=""	Significa assenza di associazioni nelle condizioni comm.li	S/C/Cd
pcnft() e pctft()	Rappresentano il Fatturato Netto e Lordo Se hanno medesimo valore lordo (con IVA), dal menu 'Aziende\Parametri di base\Contropartite escluse dal fatturato' premere 'IvaAuto' per popolare la tabella a video, e poi, riorganizzare. Nota Bene: l'anno <u>corrente</u> equivale a 'sxann+1' (mentre 'sxann' è quello <u>precedente</u>). A differenza di altri dizionari [per esempio i saldi ppsaf(pccod(),sxann) e ppsal(pccod(),sxann) oppure l'inventario aqinv(arcod(),0,sxann)], cioè di casi in cui 'sxann' (oppure 'sxsol' per le infrannuali) equivale all'anno <u>corrente</u> , nel dizionario pcnft(anno)(conto) e pctft(anno)(conto), invece, è pari all'anno <u>precedente</u> , e pertanto, quello <u>corrente</u> equivale a 'sxann+1' (essendo questo un vettore di 10 elementi, cioè 10 anni). La ragione di questa differenza è che, se nei parametri del dizionario è richiesto l'anno, questo è indicabile con 'sxann' come <u>parametro di lettura</u> , diversamente, l'interrogazione di vettori che corrispondono ad anni contabili la variabile 'sxann' non è più	S/C/Cd

	<p>valida in quanto i valori contenuti nel dizionario 'sxann' vanno da 0 a 9, ma purtroppo, il valore zero come elemento di vettore non è consentito nel linguaggio sprix. Non a caso i suddetti dizionari hanno sintassi dei parametri differenti, esempio: ppsaf(,) e pcnft()().</p> <p>In caso di <u>ciclotabellare</u>, infine, l'ordinamento in base al pcnft / pctft richiede tanti ciclitabellari in base all'anno in corso (es. fino a 10 e poi eventualmente sortare i valori), poichè nel ciclotabellare non è consentito l'utilizzo dell'indice per anno (poichè non consente di inserire costanti)</p>	
pctpp()	Partita Iva secondaria corrispondente alla variabile _PCPIS\$. Assume valore "N" (normale partita iva univoca), "A" (allegati, partita iva non univoca e valore di secondaria = "S"), "P" (privato).	S/C/Cd
pcbcc(), pcbbc()	Le variabili banca relative al codice conto corrente e codice cin sono relative ai conti banca e non a quanto inserito nell'anagrafica del cliente/fornitore	S/C/Cd
patpp(), pctpp()	Conto privato: campo dizionario equivalente alla variabile _PCPRI\$ Il dizionario patpp() e pctpp() riportano il valore equivalente al conto privato o non. Per esempio, _PCPRI\$="S" equivale a patpp()="P"	S/C/Cd
pccra(), pcana()	I Conti di Contabilità Analitica associati ai Conti di Contabilità Generale (Piano dei Conti e Clienti Fornitori)	S/C/Cd
pctfe(), pctfep(), pafep()	Tipo di fattura elettronica PA e B2B. Il dato ??tfe() è su archivio PICO, ??fep() è presente sia su PICO che archivio RUDT; fuori da questi casi i valori sono = "" ("non gestita").	S/C/Cd

Esempio: copia di conto, attribuzione di P.Iva e C.F., controllo+cancellazione

```
AZZVARSYS 1:GETPC "201.00006":IF _ERRPC<>0 THEN VIMSG 5, _ERRPC$:END
_PCCOD$="201.AUTO": PCDES$="Copia Conto": PCNAZ$="I":_PCLIS=1:_PCVAL=2
_PCNPI$="12345678900":_PCCFI$="CPACNT82A03H294J"
PUTPC: IF _ERRPC$<>" " THEN VIMSG 4, _ERRPC$
PI=CTRPIVA(panpi("201.00177")) ` oppure PI=CTRPIVA("12345678900")
IF PI=0 THEN VIMSG 1, "P.Iva errata" ELSE VIMSG 1, "P.Iva corretta"
CF=CTRCODFIS(pacfi(_PCCOD$))
IF CF=0 THEN VIMSG 1, "C.F. errato" ELSE VIMSG 1, "C.F. corretto"
DELPC "201.00177": IF _ERRPC<>0 THEN VIMSG 5, _ERRPC$ : END
```

Una considerazione.

Se una PIVA (partita iva) è trattata con controllo relativo a CF (codice fiscale), può risultare un errore dovuto al fatto che la CTRCODFIS considera il CodFis 'secco' e non accetta p.iva italiane. Una volta verificato che la variabile CodFis è numerico lungo 11, va controllato con la CTRPIVA e non con la CTRCODFIS.

```
Anche se è un CF, il fatto che sia numerico 11 caratteri richiede la CTRPIVA. Esempio
IF LEN(CF$)<=11 THEN CTRPIVA ELSE CTRCODFIS
Per la controverifica se è in realtà tutto numerico (senza doversi avvalere della STRSMP_CI)
FOR I=1 TO LEN(CF$)
X$=MID$(CF$,I,1) : IF X$>="0" AND X$<="9" THEN GOTO CONTINUA ELSE CTRCODFIS : END
CONTINUA:
NEXT
```

Caso: come ottenere il codice dell'ultimo conto codificato con PUTPC

Dopo la scrittura PUTPC si può visualizzare la variabile _PCCOD\$ con l'istruzione VIMSG, come possibile anche in anagrafica articoli con la variabile _ARCOD\$ dopo la PUTAR. Il codice conto è memorizzato nella variabile _PCCOD\$ e il fatto di non visualizzarlo è una scelta effettuata in Sprix, quindi, non è necessariamente da visualizzare con VIMSG. Questo valore _PCCOD\$ è disponibile per ulteriori elaborazioni anche se non è visualizzato. La stessa cosa accade nella GETPC, poichè essa riempie la variabile _PCCOD\$.

Caso: _PCPAE\$ vuota e non valorizzata

In caso di scrittura dell'anagrafica del piano dei conti dove la variabile _PCPAE\$ non viene valorizzata non si verifica nessun errore in fase di scrittura, ma il dato non è presente nel record su disco.

Entrando a video nell'anagrafica in caso il campo Nazionalità contenga il valore "I", "V", "R" automaticamente la procedura visualizza nel campo paese la decodifica standard del valore, ma tale dato non essendo presente su disco non è riportato nella relativa variabile del ciclo tabellare o delle istruzioni GETATTPC e GETPC.

E' necessario entrare nell'anagrafica e confermarla o tramite procedura sprix o procedura standard.

Caso: i saldi provengono dalla Primanota

La scrittura dei saldi non avviene con PUTPC, ma tramite PUTPN. L'importo dei saldi può avvenire con MODPUT, ma la primanota generata è cablata (solo giroconto GR / apertura AP con importo e data e non gestisce lo scadenziario).

Caso: PUTPC [Master/Azienda] non ammessa su tipi conto diversi da C/F/B (_PCSOLOPICO\$)

Messaggio rilasciato quando si lavora nella nuova tecnologia del gestionale a file separati, con aziende di raggruppamento. In questa modalità gli unici conti che la PUTPC processa sono i clienti, i fornitori e le banche (che sono PROPRI della azienda in questione). Tramite la variabile _PCSOLOPICO\$ ="S/N" si ha la possibilità di aggiornare tutte le variabili tipiche del conto aziendale anche per i conti di raggruppamento. I dati del RUDT rimangono invariati. Per i SOLI conti presenti nell'azienda le seguenti variabili possono essere modificati se la variabile: _PCSOLOPICO\$ = "S"

Dati comuni a tutti i conti

_PCDSA\$	Descrizione aggiuntiva
_PCMRA\$()	Stato anagraf. attiva
_PCMRV\$()	Stato anagraf. variata
_PCVDF\$()	campi videata anagraf. non riportabile
_PCVDR\$()	campi videata anagraf. riportabile

Dati conti di tipo banca

_PCNBA\$	Num. banca su archivio BANC	_PCBAPL()	Aliquota tasso passivo
_PCBATC()	Tipo c/c	_PCBARI\$	Riferimento Signor
_PCBACC\$()	n. c/corrente	_PCBASE\$	Riferimento Settore
_PCBASI\$()	Cod. S.I.A)	_PCBATE\$	Riferimento Telefono
_PCBAAZ\$()	Firma Azienda	_PCBANO\$	Riferimento Note
_PCBADI\$	Data iniz.rapporto	_PCBBCIN\$	Cod. Cin su BBAN
_PCBADF\$	Data fine rapporto	_PCIBPAE\$	Cod.paese per codif IBAN
_PCBAAD\$	Data tasso attivo	_PCIBCIN	Cod. cin per codif IBAN
_PCBAAL()	Aliquota tasso attivo	_PCIBCOD\$	Cod libero su codif.IBAN
_PCBAPD\$()	Data tasso passivo	_PCBIC\$	Codice di BIC

Nota: in presenza di raggruppamento, la possibilità di identificare conti come di tipo banca, non passa più solo dall'attribuzione di _PCCLAS\$="B" (che distingueva rispetto a conti di tipo Cassa, per esigenze di flussi finanziari), ma passa anche dal tipo di Mastro, per il quale è attiva l'attribuzione come "Depositi Bancari". Se infatti nel raggruppamento, un Mastro patrimoniale appartenente a un gruppo visibile in Anagrafica Piano dei Conti viene marcato come "Ban" (cioè come Deposito Bancario), questo Mastro (e relativi conti) sarà visualizzato nella Rubrica clienti/fornitori/banche, perchè appartenenti alla specifica azienda in questione. Se poi il raggruppamento e su piano dei conti riservato (80/81), i conti non riservati (quindi personalizzabili) sono quelli con mastro superiore al 69.

PUTPC - Rubrica unica centralizzata - GETRUBUNICA

Per inserire un conto cliente fornitore non presente in rubrica unica si opera in modalità standard, cioè si valorizzano le variabili _PC??\$ e si esegue la PUTPC. Unica particolarità che la variabile _PCCOD\$ sia = "GMM.AUTO"

Se invece si valorizza la variabile _PCCOD\$ = "GMM.00001" se il conto appartiene all'azienda verrà eseguita la normale variazione del conto, se invece, il conto appartiene alla Rubrica Unica ma non alla rubrica aziendale allora si deve utilizzare il comando GETRUBUNICA per leggere i dati della Rubrica unica ed inserirli nella rubrica aziendale. Di seguito è riportato un esempio in cui si inserisce un nuovo conto presente in rubrica unica centralizzata ma non nella rubrica aziendale:

```
_PCCOD$ = " 501.AUTO"
_PCDES$ = "Nuovo cliente"
_PCVAL = 1
_PCLIS = 1
_PCNAZ$ = "IT"
_PCNPIS$ = "12345678901"
```

PUTPC

```
IF _ERRPC$ <> "" THEN IDCONTO$ = MID$( _ERRPC$, INSTR(_ERRPC$, "ID:") + 3, 10) : CODICE$ = MID$( _ERRPC$, INSTR(_ERRPC$, "<") + 1, 9)
```

Se la partita iva è presente in qualunque conto della rubrica unica ma non è presente nella rubrica aziendale, la variabile _ERRPC\$ ritorna, oltre insieme al messaggio di errore, anche il codice di conto che sta utilizzando la partita iva e l'ID che è legato a tale conto. La if precedente indica in linea di massima come il codice sorgente deve essere scritto

GETRUBUNICA IDCONTO\$

```
IF _ERRUBUNIS$ <> "" THEN VIMSG 5, _ERRUBUNIS$ (tale if serve a verificare che la GETRUBUNICA sia andata a buon fine)
```

in caso di esito positivo della GETRUBUNICA le variabili:

_PCDES\$, _PCNPIS\$, _PCCFI\$, _PCFAX\$ etc cioè i dati strettamente della Rubrica, non quelli relativi a pagamenti o condizioni commerciali, sono assegnati alle relative variabili

assegnamo quindi il codice di conto

```
_PCCOD$ = CODICE$
```

PUTPC

```
IF _ERRPC$ <> "" THEN (testiamo il messaggio di errore per verificare cosa possa essere successo)
```

```
se _ERRPC$ = "" il conto è stato inserito nella rubrica aziendale mantenendo le stesse informazioni della rubrica unica.
```

Se la prima PUTPC ritorna un messaggio diverso ci si regola di conseguenza.

Nota: Nota: la replica MexalDB è un concetto completamente aziendale; le tabelle generali (raggruppamenti, rubrica unica rudt, banche, ecc) sono database aziendali, pertanto trattati per ciascuna singola azienda. Es: nelle aziende AAA e BBB con rudt e MexalDB attivo automatico la replica iniziale di entrambe le aziende crea le tabelle AAA RUDT in "aaa rp" e BBB RUDT in "bbb rp", ma l'aggiornamento del dato di rubrica unica nell'azienda AAA aperta allinea solamente la tabella AAA RUDT (non la tabella BBB RUDT) e viceversa; anche in caso di aggiornamento senza alcuna azienda aperta non allinea alcunchè, perchè MexalDB non è attivo. Non è possibile aggiornare una tabella del database bbb rp tramite l'azienda AAA aperta/chiusa e viceversa. Per allineare le tabelle delle altre aziende, basta aprire l'azienda ed in Servizi - Riorganizzazione archivi aziendali selezionare RUBRICA PC/CLIENTI/FORNITORI e premere F8 (Elabora solo MexalDB), oppure aprire la tabella di raggruppamento e riconfermare un valore.

Caso: come intercettare il codice cliente/fornitori nella nuova interfaccia grafica

Di seguito alcune etichette che permettono di conoscere questa informazione:

Nella finestra elenco è attivo l'evento sul campo "CASCECOD":

```
ON_OUT_CASCECOD:
  ATTIVAFIN_NFIN
  VIMSG 5, SCREEN$( _NFIN, GETPOSCURS(2), 40, 9)
END
```

In inserimento di un nuovo codice premendo il tasto F4 sono attivi gli eventi in "CACOD":

```
ON_OUT_CACOD:
  VIMSG 5, "FUORI FINESTRA CACOD"
END
```

Premendo il tasto F3 cerca codice "CKCERCACOD":

```
ON_OUT_CKCERCACOD:
  VIMSG 5, "FUORI FINESTRA CKCERCACOD"
END
```

In qualunque condizione si opera l'evento STARTPC ritorna a seguito della GETATTPC il codice:

```
ON_STARTPC:
  GETATTPC
  VIMSG 5, _PCCOD$
END
```


TABELLE ABI CAP (ARCHVIEW)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 56	Istruzione che azzeri variabili di struttura _ARCHVIEW	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "ARCHVIEW"	Istruzione che azzeri variabili struttura _ARCHVIEW (come azzvarsys)	S/C/Cr/Cd Sm/Cm
ARCHVIEW	Par: <archivio>,<par\$>. Istruzione di finestra elenco tabella ABI e CAP	S/C/Cd
_ERRVIEW	Codice risultato operazione su tabella ABI e CAP	S/C/Cd
_ERRVIEW\$	Messaggio operazione su tabella ABI e CAP	S/C/Cd
_ARCHVIEW	Variabili di struttura archivio tabella ABI e CAP	S/C/Cd

Esempi e casi dizionario

Esempio Sprix ARCHVIEW (Tabella ABI):

```

ESEMPIO_1:
AZZVARSYS 56
ARCHVIEW 1,"" 'Apertura dell'elenco non filtrato delle banche italiane

IFF _ERRVIEW<>0
  IFF _ERRVIEW=1
    VIMSG 5,_ERRVIEW$
  ELSEF
    VIMSG 1,"Nessuna selezione effettuata"
  ENDF
ELSEF
  VIMSG 1,"I valori selezionati sono: ABI="+_ARCHVIEW$(1)+" ,CAB="+_ARCHVIEW$(2)
ENDF

RETURN

ESEMPIO_2:
AZZVARSYS 56
_ARCHVIEW$(1)="01000"
_ARCHVIEW$(6)="ROMA"
ARCHVIEW 1,"" 'La finestra si apre con le sole filiali di BANCA D'ITALIA di ROMA

IFF _ERRVIEW<>0
  IFF _ERRVIEW=1
    VIMSG 5,_ERRVIEW$
  ELSEF
    VIMSG 1,"Nessuna selezione effettuata"
  ENDF
ELSEF
  VIMSG 1,"I valori selezionati sono: ABI="+_ARCHVIEW$(1)+" ,CAB="+_ARCHVIEW$(2)
ENDF

RETURN

ESEMPIO_3:
AZZVARSYS 56
_ARCHVIEW$(1)="01000"
_ARCHVIEW$(2)="99999"
ARCHVIEW 1,"" 'La finestra non si apre perche' non esistone banche per i filtri di selezione impostati _ERRVIEW=1

IFF _ERRVIEW<>0
  IFF _ERRVIEW=1
    VIMSG 5,_ERRVIEW$
  ELSEF
    VIMSG 1,"Nessuna selezione effettuata"
  ENDF
ELSEF
  VIMSG 1,"I valori selezionati sono: ABI="+_ARCHVIEW$(1)+" ,CAB="+_ARCHVIEW$(2)
ENDF

RETURN

ESEMPIO_4:
AZZVARSYS 56
_ARCHVIEW$(1)="01000"
_ARCHVIEW$(2)="03207"
ARCHVIEW 1,"NOVIEW=1" 'La finestra non si apre per via del parametro "NOVIEW=1" e perche' esiste una sola filiale per i codici
ABI/CAB specificati

IFF _ERRVIEW<>0
  IFF _ERRVIEW=1
    VIMSG 5,_ERRVIEW$
  ELSEF
    VIMSG 1,"Nessuna selezione effettuata"
  ENDF
ELSEF
  VIMSG 1,"I valori selezionati sono: ABI="+_ARCHVIEW$(1)+" ,CAB="+_ARCHVIEW$(2)
ENDF

RETURN

```

REFERENTI (CFR)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 33	Istruzione che azzerà variabili referenti _CFR	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "CFR"	Istruzione che azzerà variabili struttura _CFR (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETCFR	Par: <cod\$>,<tipo>,[arch]. Istruzione che legge riferimenti dei referenti Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/Cr/Cd/Sh
GETATTCFR	Istruzione che legge riferimenti dei referenti a video Non consentita nei Livelli di sviluppo (F2, L3, F3)	C
PUTCFR	Istruzione che scrive riferimenti dei referenti Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/Cr/Cd/Sh
DELCFR	Par: <cod\$>,<tipo>,[arch]. Istruzione che cancella riferimenti dei referenti Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/Cr/Cd/Sh
_CFRERR	Codice risultato operazioni su referenti	S/C/Cd
_CFRERR\$	Messaggio operazioni sui referenti	S/C/Cd
_CFR	Variabili di struttura referenti	S/C/Cd
ON_STARTCFR:	Evento relativo all'accesso nei referenti della voce di menu (MR)	C/Cd
ON_ENDCFR:	Evento relativo all'uscita dai referenti della voce di menu (MR)	C/Cd
esempi e casi dizionario		
Esempio Sprix REFERENTI		
Nel Conto 501.00063 sono modificati il nome e la nota del secondo referente relativo alla segreteria		
<pre> AZZCATVARSYS "CFR" GETCFR "501.00063",0,"TP=0" _CFRCOD\$(2)="501.00063" _CFRPRG(2)=2 _CFRPOS(2)=2 _CFRNOME\$(2)="Nome nuovo" _CFRNOTA\$(2)="Nota nuova" *Nota: l'aggiunta di referenti richiede di specificare l'archivio di appartenenza _CFRTP\$(I) PUTCFR IF _CFRERR<>0 THEN VIMSG 1, _CFRERR\$ </pre>		
Esempio:		
<pre> GETCFR "501.00001", 0 Esegue la lettura dei referenti del <u>conto</u> "501.00001" GETCFR "1", 0, "TP=0" Esegue la lettura dei referenti del <u>contatto</u> "1" </pre>		
Il collegamento fra referente registrato e array Sprix è stabilito da _CFRPRG(). Array visibile a video Array Sprix 1 = Direzione 1 = Direzione 2,3,4,5 = " " 2 = Amministrazione 6 = Amministrazione 3 = Assistenza 7,8 = " " 9 = Assistenza		

PRIMANOTA (PN)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 2	Istruzione che azzerava le variabili di primanota _PN	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "PN"	Istruzione che azzerava variabili struttura _PN (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETPN	Par: <dreg\$>, <prinr>, <cau\$>, <protr\$>, <prots>, <protn>, <docn>, <ddoc\$> Istruzione che legge documento di primanota. Non disponibile in Solution	S/Cr/Cd/Sh
GETATTPN	Istruzione che legge primanota a video. Non disponibile in Solution	C
GETNRIPRNVID()	Funzione di lettura numero di riga a video del corpo di primanota	S/C/Cr/Cd/Sm
PUTPN	Istruzione che scrive primanota. Non disponibile in Solution	S/Cr/Cd/Sh
PUTPNX	Istruzione che scrive primanota Lire/Euro. Non disponibile in Solution	S/Cr/Cd/Sh
DELPN	Par: <dreg\$>, <prinr>, <cau\$>, <protr\$>, <prots>, <protn>, <docn>, <ddoc\$> Istruzione che cancella documento di primanota	S/Cr/Cd/Sh
ERRPN	Codice risultato operazione primanota	S/C/Cd
ERRPN\$	Messaggio operazione primanota	S/C/Cd
_NPN	Numero righe documento primanota	S/C/Cd
PN	Variabili di struttura primanota	S/C/Cd
ON_DELPN:	Questo evento non si manifesta al ON_DELM: nel menu MD	C/Cd

logiche

PUTPN/X, GETATTPN e Dizionario di Primanota, non sono utilizzabili da livello di prodotto "Solution".

PUTPN scrive solo nuove registrazioni, per versioni precedenti alla 2023D esclusivamente nella valuta di gestione contabile dell'azienda, e con i dati obbligatori relativi alla versione del gestionale in uso. Le contropartite AVERE sono < 0 (negative) e quelle DARE > 0 (positive).

PUTPN scrive le Rate/Partite in Scadenario in base al Codice Pagamento (_PNPAG) della Tabella Pagamenti e solo se la registrazione è nuova e con contropartite in grado di generare rate significative per lo Scadenario (es. la registrazione di un "PG" con importi nulli non genera rate, nemmeno nulle).

PUTPN non è utilizzabile come funzione di tipo Saldaconto: non aggiorna Primanota/Scadenario (archivio dinamico in sola lettura) della registrazione già esistente. Non aggiorna le Rate/Partite, poichè occorre la registrazione di Primanota "PG" Pagamento Rate/Partite (a *limite con simulazione dell'utente con puttast in Collage*). PUTPN (e GETPN) non gestiscono le variabili scadenario (rate/partite, date, importi, tipi pagamento, ulteriori dati scadenario, banca d'appoggio -eccetto in magazzino con _MMBAP-, ecc.).

PUTPN consente la registrazione di "PG" pagamento di Rate/Partite; nello scadenario a Partite è consentita la gestione delle Partite. Si ricorda che le Note di Credito (NE/NR) si comportano come dei pagamenti e la regola che occorre seguire per saldare un documento è che solo un documento con segno negativo può chiuderne uno con segno positivo, mai viceversa.

PUTPN registra lo Scadenario a Documento dei Clienti/Fornitori presenti nella registrazione: l'importo da scadenzare positivo (fattura) genera una o più scadenze in base al codice pagamento (se _PNPAG=0 è assunto quello di anagrafica Clienti/Fornitori – tranne l'alternativo – e se è ancora zero è emessa un'unica scadenza con rimessa diretta 'M'); non è attivo l'automatismo di pagamento in contanti, occorre registrare un pagamento PG.

In caso di importo negativo (il pagamento) la _PNPCA\$ (causale documento in pagamento) determina il tipo di azione:

- se _PNPCA\$ è valida allora la _PNPSD (per documenti sui Registri protocollo Iva "A","V","S","C"), la _PNPND e la _PNPDD\$ del documento da pagare sono obbligatorie e vengono pagate tutte le rate fino a raggiungere l'importo in pagamento (se il documento non viene trovato o se l'importo da pagare è maggiore della somma delle rate da pagare viene emessa una scadenza negativa con causale rimessa diretta);
- se _PNPCA\$ contiene un tipo di scadenza valido ("C","M","X","Y","Z","R","B","A","T","V") oppure il carattere "***", vengono pagate tutte le rate con quel tipo di scadenza (o rate di qualsiasi tipo se _PNPCA\$ è "***"), iniziando dalla più vecchia fino a raggiungere l'importo da pagare (l'eventuale residuo viene messo a pareggio);
- se _PNPCA\$ è vuoto viene emessa una scadenza negativa con causale rimessa diretta.

Le variabili in sola scrittura _PNPCA\$, _PNPSD, _PNPND e _PNPDD\$ vanno valorizzate per la nuova registrazione con PUTPN; in ogni caso in cui sia prevista l'emissione di una o più scadenze, se _PNPSD, _PNPND o _PNPDD\$ della primanota sono a zero oppure non sono valide, l'importo viene pareggiato.

Nello Scadenario a Partite ciascun documento ha le sue partite. La registrazione del pagamento consente di scegliere le partite da pagare e con lo stesso pagamento si possono pagare più documenti (v. esempi). La possibilità (non consentita nello Scadenario a Documento) di chiudere fino a 250 rate di documenti/partite diversi avviene attraverso: _PNSCT\$(249) Cli/For; _PNSPA\$(249) Partita; _PNSDS\$(249) Data Scadenza; _PNSID(249) ID Scadenza. Nota: _PNPAR\$ non serve per chiudere la partita, ma serve per modificare la partita di default che imposterebbe il programma. Esempio: registro la FE 1/1 nel 2019, il programma assegna alle rate la partita 19FE01/00001 a meno che non valorizzi la partita con altro tramite _PNPAR\$.

PUTPN non aggiorna la Primanota, ma può solo crearla. Per modificare la Primanota occorre: azzerare la struttura (AZZVARSYS) + leggere la registrazione (GETPN) + cancellare la registrazione (DELPN) + re-inserire la nuova registrazione (PUTPN). Considerazioni:

- AZZVARSYS consente di azzerare totalmente la struttura, poichè DELPN/PUTPN mantengono invariate le variabili pregresse eventualmente già inizializzate.
- GETPN consente di leggere la registrazione e valorizza il numero di righe lette nel documento (_NPN). Il parametro data di registrazione <dreg\$> è obbligatorio ed insieme al numero progressivo di registrazione <prinr> forma la **chiave primaria** di primanota: l'omissione di <prinr> (se = 0) rende i restanti parametri necessari per identificare e discriminare le registrazioni con

dati equivalenti (altrimenti il documento processato è il primo che viene trovato in archivio), pertanto, devono assumere valore diverso da 0 (se numerico) o "" (se stringa); l'indicazione di alcuni parametri = 0 o = "" può infatti identificare più di un documento, fra i quali il primo incontrato è quello processato. Il progressivo <prinr> è contenuto nella variabile di struttura di sola lettura _PNPRN (numero progressivo di registrazione) e funziona così: quando si effettuano variazioni sulle registrazioni dal menù contabilità (primanota) il numero viene mantenuto uguale al precedente (viene effettuata una variazione), mentre se si effettua la variazione di un documento da magazzino (emissione revisione documenti) la registrazione "originale" viene cancellata e riscritta con un numero differente. Nella riorganizzazione archivi di primanota viene segnalato il numero di record fisico della registrazione, mentre _PNPRN è il record di creazione che viene assegnato all'inserimento della registrazione. In generale corrispondono, ma se la registrazione viene revisionata, modificata (es. aggiunte di conti, righe, descrizioni, intra ecc.), il record fisico può essere spostato (la registrazione può essere fisicamente spostata), differendo da _PNPRN che invece non cambia in caso di revisione.

- DELPN consente la cancellazione della registrazione. La cancellazione mantiene i dati da re-inserire, ma azzerà l'eventuale collegamento al documento di magazzino (se ha generato la primanota), che serve al Gestionale per l'aggiornamento dal magazzino alla primanota (nel caso replicare la stessa operazione anche per il documento di magazzino, considerando che la cancellazione con Sprix del magazzino non cancella la primanota) e con nuova assegnazione <prinr>. La cancellazione verifica che nello scadenzario i riferimenti del documento (causale + numero + data) non siano stati alterati da eventuali manipolazioni (cancellazioni, pareggi, ecc). La cancellazione verifica la data di registrazione e la presenza di variazione di una registrazione con ritenuta d'acconto già versata" e verifica che non siano avvenuti dei pagamenti (occorre la rimozione manuale di 'P' dallo scadenzario scollegando la rata del documento); la cancellazione non è consentita se è già avvenuto un pagamento poiché vincolato nello scadenzario (il quale non è modificabile da Sprix); eccetto quanto consentito nello scadenzario a **Partite** dove è possibile cancellare la scrittura di pagamento dopo averlo rintracciato (v. variabile _PNPCA\$) per poi riscrivere tutta la registrazione del documento e relativo pagamento, nello scadenzario a **Documento**, invece, la PUTPN non riesce ad identificare il numero o la data della rata in pagamento (non si può selezionare le rate e pagarle), poiché avviene il pagamento dalla prima rata presente nello scadenzario (indipendentemente dalla data di scadenza, in quanto la rata è modificabile dall'utente).

Per ulteriori condizioni consultare _ERRPN\$.

```
Esempio di modifica del conto della prima contropartita contabile;
AZZVARSYS 2 : GETPN "20250427",0,"FE","V",1,0,15,"" : IF _ERRPN<>0 THEN GOTO ERRORE
DELPN "20250427",0,"FE","V",1,0,15,"" : IF _ERRPN<>0 THEN GOTO ERRORE
_PNCTO$(1)="501.00002" : PUTPN : IF _ERRPN<>0 THEN GOTO ERRORE :END
ERRORE : VIMSG 1,_ERRPN$ : END
```

```
Esempio di aggiunta delle descrizioni per ogni contropartita alla fattura FE 1/37 del 01/01/2007 (con Id di primanota _PNPRN=19457) e composta da tre contropartite e pagamento con rimessa diretta
GETPN"20070101",19457,"FE","V",1,37,37,"20070101" : VIMSG 1,_ERRPN$
DELPN"20070101",19457,"FE","V",1,37,37,"20070101" : VIMSG 1,_ERRPN$
_PNDES$(1)="aggiungo 1" : _PNDES$(2)="aggiungo 2" : _PNDES$(3)="aggiungo 3"
PUTPN : VIMSG 1,_ERRPN$
```

Ratei e Risconti

Non sono previste istruzioni e variabili di gestione dei Ratei e Risconti. L'unico modo per potere inserire automaticamente dei ratei è utilizzando un Collage che simuli l'input manuale dell'utente, attraverso una sequenza di routine all'ingresso/uscita dei campi per gestire opportuni _SINP\$ e PUTTAST.

I Ratei/Risconti sono collegati alla primanota se sono registrati direttamente dal modulo di primanota; il collegamento è attraverso l' ID univoco di primanota e dal conto. Se questo è univoco l'informazione è raggiunta se il conto non è univoco non si ha certezza della riga di primanota a cui è collegata. A questo scopo potrebbe aiutare anche l'importo di Rateo/Risconto se non è stato modificato. Dal modulo di immissione dei Ratei/Risconti c'è solo un riferimento alla primanota (non è un collegamento).

Cespiti

Non sono previste istruzioni e variabili di gestione dell'archivio Cespiti. E' consentita la trasmissione dati su file in formato CSV.

Controllo di Gestione

La PUTPN gestisce le sole variabili _PN relative ai movimenti analitici (funzionalità standard); non gestisce il modulo di Controllo di Gestione, per il quale occorre utilizzare le apposite istruzioni GETCDG, PUTCDG, DELCDG.

esempi e casi dizionario

tabella "pn" (primanota)	Non disponibile per Solution	S
tabella "pn" (primanota)	Non gestita la Notula	S
pnord()	sono criteri di ordinamento righe progressivi (3,4,7,...) e non necessariamente in sequenza (1,2,3,4,5...); possono essere riordinati in memoria, per ottenere la stessa sequenza visualizzata a video	S
psnpi(), psdar()	riportano valori solo per documenti superiori al 2001 (anno della loro implementazione); lo scadenzario ha 250 rate, 12 contropartite sospese e 24 rate professionisti; nelle provvigioni 'ps'-'pd' il maturato va calcolato con eventuali arrotondamenti (la provvigione maturata in base all'importo della rata di scadenzario è un valore non memorizzato). Esempio 100 di provvigione su tre rate: P=100/3 : P1=P : P2=P : P3=ARRO(P*100,4,1)/100 T=P1+P2+P3 : VIMSG 1, STR\$(T,2,0)	S

	<p>Nei pagamenti tramite Riba considerare i giorni di esposizione bancaria stabiliti nei parametri contabili; lo scadenziario non visualizza le rate scadute che abbiano superato l'esposizione. Per visualizzarle, operare con data antecedente o prossima alla scadenza Riba e i suddetti giorni.</p>	
<p>psage(), pspro(), psprd(), pspsrp()</p>	<p>Provvigioni. Psage() è il codice dell'agente nella FE al cliente, il pspro() è l'importo provvigione; psprd() è come la pspro(), ma riguarda documenti emessi con valuta documento diversa da quella di gestione (si valorizza qualora vi fossero documenti in valuta). Quando il cliente paga (anche parzialmente) la FE, nella gestione delle provvigioni si valorizza il campo maturato: questo dato non è disponibile in Sprix, non è memorizzato ma ricalcolato in proporzione alle rate pagate della fattura, pertanto, occorre replicare il calcolo (ad es. il maturato sta al totale delle rate pagate, come pspro() sta al totale della fattura); la presenza di più rate (pagate o meno) complica l'elaborazione e può richiedere cicli FOR-NEXT, con controllo sullo stato del pagamento di ogni rata, e la difficoltà può aumentare, nella necessità di considerare anche la tipologia di pagamento che, qualora coinvolgesse le RIBA deve considerare i giorni di esposizione bancaria (salvo attivazione del parametro "Cons.espos.pagata in prov.(S/N):S"). Dopo che l'agente ha emesso la sua fattura per la prestazione effettuata (FF) si può effettuare il PG dalla gestione delle provvigioni, valorizzando di conseguenza il codice psprp(), che corrisponde alla liquidazione della provvigioni. Notare, che all'aumentare di psprp() diminuisce il pspro(), in quanto quest'ultimo assume anche la funzione di provvigione residua da liquidare. In tal modo si risale all'informazione sulla liquidazione totale o parziale della provvigione.</p> <p>Esempio di ciclotabellare per estrarre un elenco di registrazioni di primanota di un determinato agente.</p> <pre> DIM ARRAY () +-----+ Cod.Tab.:ps CONTABILITA' - SCADENZARIO Scelte.: psage()="631.00001" +-----+ ps I=I+1 ps ARRAY(I)=psnum() FOR J=1 TO I +-----+ Cod.Tab.:pn CONTABILITA' - PRIMA NOTA INTESTAZ. Ordine 3:pnndo() Scelte.: pnndo()=ARRAY(J) +-----+ pn N ~pndre() ~pncau() ~pnndo() ~pnndo() ~pncto() NEXT END </pre> <p>Data l'impossibilità di pagamento in blocco con un'unica operazione delle provvigioni agenti in base al residuo maturato, servirebbe un parametro che provveda a precompilare la colonna pagamento con gli stessi importi del residuo maturato, come nella funzione "Stampe/Contabili/Provvigioni agenti" nel caso si utilizzi la combinazione dei parametri di selezione "Provvigioni sul Fatturato/Maturato = M" e "Solo Maturato = N". Questa è una <u>richiesta</u> sviluppabile in Collage, per esempio due Collage: terminata la stampa da menu SCA (con Collage su SCA), alla pressione di F12 si esegue un secondo Collage lanciando in automatico il menu CA con relativo Collage (compila in automatico il codice dell'agente e attende la conferma F10); alla conferma compila il campo col residuo maturato e per ciascun invio, compila i successivi. Il codice in calce è riferito ai due collage, nei rispettivi menu. Il codice può essere reso più articolato, ad esempio coi dovuti controlli che la funzione di liquidazione provvigioni richiede essendo vincolata alle logiche di scadenziario da rispettare.</p> <pre> SCA ON_START: CRLF\$=CHR\$(13)+CHR\$(10) IF ESISFILE ("]\file.txt") THEN KILL ("]\file.txt") NF=AUTOFILE() : OPEN "R",NF,"]\file.txt",14 FIELD NF,12 AS V\$ ALFA, 2 AS CRLF\$: END ON_OUT_SCAAGEN: V\$=LEFT\$(_SINP\$,9) : PUT NF,1 : END ON_ST_LPRAG_GENERICO: I=I+1 : V\$=RIGHT\$(_SST\$,12) : PUT NF,I+1 : END ON_612_ALL: CLOSE NF : ESECMENU "CP" : END CP ON_START: CRLF\$=CHR\$(13)+CHR\$(10) : DIM VALARRO () NF=AUTOFILE() : OPEN "R",NF,"]\file.txt",14 FIELD NF,12 AS V\$ ALFA, 2 AS CRLF\$ NUM=LOF(NF)/14 : NUMREC=NUM-3 END ON_IN_CPAGEN: GET NF, 1 : _SINP\$=V\$: PUTTAST "#501" END ON_OUT_CPDADATA: SETTF "610" : IF _TF=610 THEN I=I+1 END ON_OUT_CPDADATA: SETTF "610" : IF _TF=610 THEN I=I+1 END ON_IN_CPDAPAG: GET NF, I+1 : POS=INSTR(V\$,".") VALORE\$=LEFT\$(V\$,12-POS-1)+MID\$(V\$,POS+1,12-POS) </pre>	<p>S</p>

	<pre> VALORE=VAL (VALORE\$) : VALARRO (I) =ARRO (VALORE, 3, 1) _SINP\$=STR\$ (VALARRO (I) , 0, 0) END ON_OUT_CPDAPAG: IF _TF=512 THEN I=I-1 IF I=NUMREC-1 THEN END IF _TF=1 OR _TF=513 THEN I=I+1 END </pre>	
pspdt()	<p>Varia la data di scadenza in data pagamento. All'atto del pagamento, la data di scadenza viene fisicamente sostituita con quella di pagamento. La data di scadenza resta memorizzata negli ulteriori dati e riportata dalla variabile psudt(). In caso di emissione effetti invece, psudt() assume la data di emissione effetto, poichè pspdt() non viene modificata.</p>	S
psstp()	<p>Caso: perché lo scadenziario cliente con rate in scadenza/scadute non mi fa vedere lo stato del pagamento (e,E,P,...) con psstp()? Eppure, i giorni di esposizione nei parametri contabili, la data di ingresso terminale ed il conto emissione effetti li ho controllati, ma delle prime quattro rate non mi restituisce alcunché. Perché? La risposta consiste nel fatto che la singola rata può avere 250 elementi, pertanto provare a ciclare sulla tabella "ps" una riga di stampa con i seguenti array: "ps" N psstp(1)() - psstp(2)() - psstp(3)() - psstp(4)() - etc....</p>	S
pdids()	<p>Se la variabile dizionario pdids() nonostante sia funzionante non viene stampata, significa che può partire da zero e quindi occorre stabilire nel campo di stampa la possibilità che lo zero possa essere stampato.</p>	S
tabella "ps" o "pd" (scadenziario)	<p>Stampa scadenziario con dettaglio articoli della fattura.</p> <p>Il ciclo di stampa dizionario non cicla anche i movimenti di magazzino (il cui dizionario a sua volta andrebbe ciclato), poichè occorrerebbero cicli nidificati; tuttavia è consentita la stampa di array con una logica simile a questa (o GET di PRN e poi di MM da mandare in stampa).</p> <pre> pd D1\$=pdcau() : N=pdnum() pd IF D1\$="FE" THEN D2\$="FT" pd GETMM D2\$+STR\$(N,0,0) pd FOR I=1 TO _NMM pd LPRINT _MMART\$(I)+" "+D1\$+" "+STR\$(N,0,0) pd NEXT END </pre>	S
<p>Scadenze partita in scadenziario a partite</p> <p>Gli ordinamenti seguono due regole: per PARTITA e per DATA SCADENZA + ID.</p> <p>Il numero ID [pdids(), pdidr()] ha valori da "0" (zero, il campo deve avere attivo il flag di stampa dello zero) ad "n" per quante sono le rate che fra esse riportano, relativamente alla stessa partita, la stessa data di scadenza; rate identiche per lo stesso documento incrementano il valore di pdidr(), pertanto "ID" serve per differenziare le rate in elenco (come se fosse un "3° livello di rottura"). L'ordinamento che viene visualizzato in scadenziario è: [conto + data scadenza + partita + id data scadenza + data pagamento].</p> <p>Un semplice esempio: registrare una FE con pagamento 12 RIBA a scadenza settimanale, modificare la data di una scadenza (facendola coincidere con un'altra), emettere 4 distinte fra le altre 10 restanti, in modo da chiuderle con documento "ED". Impostare ciclotabellare su "pd" con scelta sul un conto pdcto)=CONTOS\$ relativo alle registrazioni (FE e ED). Stabilire in Ordine 3 pdpar() e in Ordine 2 pdpdt(). Stabilire nella prima riga di stampa un livello di rottura I3 relativo alla partita "pdpar()" ed al documento "pdcau()", pdser(), pdnum(), pddat(). Stabilire una seconda riga di stampa separata dalla precedente che riporti: la rata "pdptp()", la data "pdpdt()", l'importo "pdpim()" (NB: attivare il flag di stampa del segno), lo ID "pdids()", la data della distinta "pduidt()".</p> <p>Variante: - Ordine3 "pdpar()", Ordine2 "pdser()+pdnum()", Ordine1 "pdids()" - prima riga di stampa con rottura I3 "pdpar()" - seconda riga di stampa consecutiva e con rottura I2 "pdcau(), pdser(), pdnum(), pddat()" - terza riga di stampa consecutiva "pdptp(), pdpdt(), pdpim()" con stampa segno, pdids() con stampa zero"</p>		
<p>Tabella "pd" non ordina con chiave di primanota. Esempi di ordinamento</p> <pre> pdpar()+pdpdt()+pdids()+pddtp() pdpar()+pddtr()+pdidr()+pddtp() </pre> <p>La chiave pdprn() può essere anche negativa: la chiave negativa indica i casi in cui la partita "pd" NON è collegata alla "pn"; nelle interfacce (e relative scelte di colori) che lo consentono, questi casi sono visualizzati in scadenziario "pd" con una diversa intensità. Le cause dello scollegamento possono essere tante (sia nella conversione "a partite", sia durante la gestione "a partite"). Solo a titolo di esempio, possono trovarsi in questa condizione i seguenti casi: vecchi pareggi convertiti nella "gestione a partite", vecchi pagamenti convertiti in rate di pagamento nella gestione "a partite", rate di operazioni di "pn" eliminate dalla riorganizzazione nella gestione "a partite" poichè danneggiate o rivelatesi incongruenti, pareggi da riorganizzazione nella gestione "a partite", caricamenti csv del solo scadenziario,...)</p>		
tabelle "cs", "ck", "cz"	<p>Ordinamenti: cstab, cscto, cscod, csann, cstip, csdti, ckcto, ckann, ckccr, ckdat, ckprg, czann, cztip, czcto, czccr (la riga di tipo 'A' è la sommatoria</p>	S

	di tutte le righe 'L' – suddivisione per Conto e per Centro di costo/ricavo della spesa sostenuta -). Il valore totale del cespite è la sommatoria delle righe comuni a cscod(), cstab(), cscto(), csann().	
caclò	Il dizionario Sprix relativo al conto corrispettivi lordi caricato nel profilo dell'import viene ignorato. Il programma Sprix di import gestisce un unico conto di corrispettivi lordi e viene letto dai conti automatici del gestionale; non viene letto quello indicato nel profilo di import. Per consentire la possibilità di gestire più conti di corrispettivi lordi (nei conti automatici nella forma GMM.?????, con relativi sospesi con mastro dedicato), occorrerebbe modificare il sorgente dell'import documenti e sostituire _PNCTO\$(1)=caclò con _PNCTO\$(1)=CORRIC\$ e rendendo obbligatorio il campo di input nella configurazione, senza poter prevedere un IF LEFT\$(CORRIC\$,1) = " " THEN _PNCTO\$(1)=caclò (poichè caclò sarebbe uguale a GMM.?????)	S
operazioni (verificare, validare, scansione)	Appartengono a primanota facile (non c'è struttura), ma c'è dizionario 'pnver' (da verificare). La causale differenzia le operazioni da scansione. Da validare è specifica di primanota facile (non disponibile).	S

Esempio: revisione fattura emessa (non pagata) con **SCADENZARIO A DOCUMENTO**; la revisione prevede contestuale pagamento del documento.

```
Data: 03/01/21 Valuta: EUR |CNC S.r.l. CANTIERI NAVALI
Caus: FE FATT. EMESSE |Saldo precedente..: 23.700,00D
Desc: Descrizione FE C.N.C. Co/ric: 1|
Prot: V01/000141 Ndoc: 141 Del: 03/01/21|
001 Codice Descrizione conto Importo (EUR)|Doc.Num. Scadenza Tipo Importo
201.00028 CNC S.r.l. CANTIERI 23700D|FE 14103/02/21M 3.700,00E
803.00001 MERCI C/VENDITE 11.700,00A|FE 14103/03/21M 5.000,00E
805.00001 RICAVI DA PRESTAZ. 9.090,91A|FE 14103/04/21M 5.000,00E
341.00052 IVA VENDITE 2.909,09A|FE 14103/05/21M 10.000,00E
castelletto Iva 803.00001 | castelletto Iva 805.00001
Aliq. Imponibile Imposta | Aliq. Imponibile Imposta
20 10.000,00 2.000,00 | 10 9.090,91 909,09
N74,3 1.700,00 N.I.art.74/c |
```

```
AZZVARSYS 2:
GETPN"20210103",0,"FE","V",1,141,141,"20210103" : IF _ERRPN<>0 THEN GOTO ERR
DELPN"20210103",4236,"",0,0,0,"" : IF _ERRPN<>0 THEN GOTO ERR
```

```
_PNDRS$="20210406": _PNRPR$="V": _PNSPR =1 : _PNNPR=0
_PNCAUS$="FE01" : _PNNDO =0 : _PNDDO$="20210406" : _PNPAG=0 \ (rimessa)

_PNCTO$(1) = "201.00029" : _PNIMP(1) = 1200
_PNCTO$(2) = "803.00001" : _PNIMP(2) = -440
_PNALIS(2,1) = "20" : _PNIME(2,1) = 300 : _PNIVA(2,1) = 60
_PNALIS(2,2) = "N74.3" : _PNIME(2,2) = 140 : _PNIVA(2,2) = 0
_PNCTO$(3) = "805.00002" : _PNIMP(3) = -636.36
_PNALIS(3,1) = "10" : _PNIME(3,1) = 636.36 : _PNIVA(3,1) = 63.64
_PNCTO$(4) = "341.00052" : _PNIMP(4) = -123.64
```

```
PUTPN : IF _ERRPN<>0 THEN GOTO ERR
AZZVARSYS 2
GETPN "20210406",_PNPRN,"",0,0,0,"" : IF _ERRPN<>0 THEN GOTO ERR
NUM = _PNNPR : DAT$ = _PNDRS$
AZZVARSYS 2 \ (altrimenti nel pagamento sarebbero presenti anche conti Iva)
```

```
_PNCAUS$ = "PG" : _PNDRS$ = DAT$
_PNPCA$ = "FE" : _PNFSD = 1 : _PNPND=NUM : _PNPDD$="20210406"
_PNCTO$(1) = "201.00029" : _PNIMP(1) = -1200
_PNCTO$(2) = "324.00001" : _PNIMP(2) = 1200 \ (324.00001 è conto cassa)
```

```
PUTPN : IF _ERRPN<>0 THEN GOTO ERR
ERR: VIMSG 4, _ERRPN$
```

Fine Esempio Scadenzario a DOCUMENTO

Esempio simile a quello sopra appena fatto (primanota con scadenziario a documento), ma con gestione di **SCADENZARIO A PARTITE**, del quale viene pagata una singola rata. Nella primanota con scadenziario a partite viene registrata la seguente FR e si paga la seconda rata di una partita con 3 rate di rimessa diretta dalla data della fattura a fine mese.

```
Data: 03/01/07 Valuta: EUR |ACCIAI srl T
Caus: FR FATT. RICEV. |Saldo precedente..: 4.630,00A
Desc: Co/ric: |Imp. Da scadenzare: 0,00
Prot: A01/000007 Ndoc: 3 Del: 03/01/07|
001 Codice Descrizione conto Importo (EUR)|P a r t i t a Saldo
601.00008 ACCIAI srl 1.200,00A|07FR01/000001 430,00E
702.00051 MERCI C/ACQUISTI 440,00D|07FR01/000002 4.000,00E
702.00024 IMBALLAGGI C/ACQUI 636,36D|07FR01/000003 1.200,00E
212.00001 IVA SU ACQUISTI 123,64D|
```

```
Partita: 07FR01/000003
-----
Scadenza Saldo
28/02/07 400,00E
31/03/07 400,00E
30/04/07 400,00E
```

```
castelletto Iva 702.00051 | castelletto Iva 702.00024
Aliq. Imponibile Imposta | Aliq. Imponibile Imposta
20 300,00 60,00 | 10 636,36 63,64
N38,4 140,00 N.I.art.38qut |
```

```
PARTITA: 07FR01/000003 Saldo: 1.200,00E
Documento del Scadenza Tipo Importo Dt. Pag/Rif Saldo Scadenza
FR 1/ 3 03/01/07 28/02/07 M 400,00E 400,00E
FR 1/ 3 03/01/07 31/03/07 M 400,00E 400,00E
FR 1/ 3 03/01/07 30/04/07 M 400,00E 400,00E
```

```
AZZVARSYS 2:
_PNDRS$="20070103": _PNRPR$="A": _PNSPR =1 : _PNNPR=0
_PNCAUS$="FR" : _PNNDO = 3 : _PNDDO$="20070103"
```

```

PNPAG=9 `(rimessa 30/60/90 data fattura fine mese)
_PNCTOS(1) = "601.00008" : _PNIMP(1) = -1200
_PNCTOS(2) = "702.00051" : _PNIMP(2) = 440
_PNALIS(2,1) = "20" : _PNIMB(2,1)= 300 : _PNIVA(2,1) = 60
_PNALIS(2,2) = "N38.4" : _PNIMB(2,2)= 140 : _PNIVA(2,2) = 0
_PNCTOS(3) = "702.00024" : _PNIMP(3) = 636.36
_PNALIS(3,1) = "10" : _PNIMB(3,1)= 636.36 : _PNIVA(3,1) = 63.64
_PNCTOS(4) = "212.00001" : _PNIMP(4) = 123.64

PUTPN : IF _ERRPN<>0 THEN GOTO ERR
NREG=_PNPRN: AZZVARSYS 2

GETPN "20070103",NREG,"","",0,0,0,"" : IF _ERRPN<>0 THEN GOTO ERR
CONTO$=_PNCTOS(1)
AZZVARSYS 2 `(altrimenti nel PG sarebbero presenti anche conti Iva)
_PPNPAR$="07FR01/000003" (Partita Generale)

PAGAMENTO PER PARTITA: _PNCTOS$+ _PNSPA$+ _PNSDS$+ _PNSID+ _PNSTP$+ _PNSIP
(pagamento della 3^ rata della partita del conto in scadenza 30/04/07 con un
pagamento in data 03/01/2007; eliminando _PNSDS$ si paga dalla rata più vec-
chia; eliminando anche _PNSIP si paga a saldo della primanota della partita
-somma _PNIMP() del _PNCTOS$(CONTO$)- a partire dalla rata più vecchia)

_PNSCT$(1)=CONTO$: _PNSPA$(1)="07FR01/000003": _PNSDS$(1)="20070430"
_PNSIP(1)=400.00
_PNCAUS$="PG" : _PNDRS$="20070331" : _PNRPR$="" : _PNDDO$="20070103"
_PNCTOS(1)="601.00008" : _PNIMP(1)= 400
_PNCTOS(2)="201.00001" : _PNIMP(2)= -400 `(201.00001 è conto cassa)

PAGAMENTO PER RIFERIMENTI DOCUMENTO:
_PNSCT$+ _PNSCA$+ _PNSSE+ _PNSNU+ _PNSDD$+ _PNSDS$+ _PNSID+ _PNSTP$+ _PNSIP
(paga 300 dalla rata più vecchia del doc FR1/8 del 14/01/06 del CONTO)

_PNSCT$(1)=CONTO$: _PNSCA$(1)="FR": _PNSSE(1)=1: _PNSNU(1)=3:
_PNSDD$(1)="20070103": _PNSIP(1)=300.00
_PNCAUS$="PG" : _PNDRS$="20070331" : _PNRPR$="" : _PNDDO$="20070103"
_PNCTOS(1)="601.00008" : _PNIMP(1)= 300
_PNCTOS(2)="201.00001" : _PNIMP(2)= -300 `(201.00001 è conto cassa)

PAGAMENTO PER DATA SCADENZA/TIPO PAGAMENTO: _PNCTOS$+ _PNSDS$+ _PNSID+ _PNSTP$
(paga tutte le rimesse "M" scadenti il 31/03/2007 del CONTO$ per un importo
pari al saldo di primanota; ATTENZIONE l'esempio produce una registrazione
non corretta nello scadenzario, perché un pagamento non può essere superiore
alla rata pagata)

_PNSCT$(1)=CONTO$: _PNSDS$(1)="20070331" : _PNSTP$(1)="M"
_PNCAUS$="PG" : _PNDRS$="20070331" : _PNRPR$="" : _PNDDO$="20070103"
_PNCTOS(1)="601.00008" : _PNIMP(1)= 500
_PNCTOS(2)="201.00001" : _PNIMP(2)= -500 `(201.00001 è conto cassa)

PUTPN : IF _ERRPN<>0 THEN GOTO ERR
ERR: VIMSG 4, _ERRPN$

RISULTATI
RISULTATO DI PAGAMENTO PER PARTITA
Data: 31/03/07 Valuta: EUR |ACCIAI srl A
Caus: PG PAGAMENTO |Saldo precedente..: 5.230,00A
Desc: Co/ric: |Imp. Da scadenzare: 0,00
Numero pagamento...: Del: 03/01/07
001 Codice Descrizione conto Importo (EUR) P a r t i t a Saldo
601.00008 ACCIAI srl400,00D |07FR01/000001 430,00E
201.00001 CASSA400,00A |07FR01/000002 4.000,00E
|07FR01/000003 800,00E

PARTITA: 07FR01/000003 Saldo: 800,00E
Documento del Scadenza Tipo Importo Dt.Pag/Rif SaldoScadenza
FR 1/ 3 03/01/07 28/02/07 M 400,00E 400,00E
FR 1/ 3 03/01/07 31/03/07 M 400,00E 400,00E
FR 1/ 3 03/01/07 30/04/07 M 400,00E
PG 03/01/07 30/04/07 MP -400,00E 31/03/07

Partita: 07FR01/000003
-----
Scadenza Saldo
28/02/07 400,00E
31/03/07 400,00E
30/04/07 0,00E

RISULTATO DI PAGAMENTO PER RIFERIMENTI DOCUMENTO
Data: 31/03/07 Valuta: EUR ACCIAI srl A
Caus: PG PAGAMENTO Saldo precedente..: 5.330,00A
Desc: Co/ric: Imp. Da scadenzare: 0,00
Numero pagamento...: Del: 03/01/07
001 Codice Descrizione conto Importo (EUR) P a r t i t a Saldo
601.00008 ACCIAI srl 300,00D 07FR01/000001 430,00E
201.00001 CASSA 300,00A 07FR01/000002 4.000,00E
07FR01/000003 900,00E

PARTITA: 07FR01/000003 Saldo: 900,00E
Documento del Scadenza Tipo Importo Dt.Pag/Rif SaldoScadenza
FR 1/ 3 03/01/07 28/02/07 M 400,00E
PG 03/01/07 28/02/07 MP -300,00E 31/03/07 100,00E
FR 1/ 3 03/01/07 31/03/07 M 400,00E 400,00E
FR 1/ 3 03/01/07 30/04/07 M 400,00E

Partita: 07FR01/000003
-----
Scadenza Saldo
28/02/07 100,00E
31/03/07 400,00E
30/04/07 400,00E

RISULTATO DI PAGAMENTO PER DATA SCADENZA/TIPO PAGAMENTO
Data: 31/03/07 Valuta: EUR |ACCIAI srl A
Caus: PG PAGAMENTO |Saldo precedente..: 5.130,00A
Desc: Co/ric: |Imp. Da scadenzare: 0,00

```

Numero pagamento...: Del: 03/01/07|

001 Codice	Descrizione conto	Importo (EUR)	Partita	Saldo
601.00008	ACCIAI srl	500,00	07FR01/000001	430,00E
201.00001	CASSA	500,00	07FR01/000002	4.000,00E
			07FR01/000003	800,00E
			07PG03/01/07	-100,00E

DOCUMENTO: PARTITA: 07FR01/000003 Saldo: 800,00E

Documento	del	Scadenza	Tipo	Importo	Dt.Pag/Rif	Saldo	Scadenza
FR 1/	3	03/01/07	28/02/07	M	400,00E	400,00E	
FR 1/	3	03/01/07	31/03/07	M	400,00E		
PG		03/01/07	31/03/07	MP	-400,00E		31/03/07
FR 1/	3	03/01/07	30/04/07	M	400,00E	400,00E	

(notare la registrazione non corretta del pagamento del 03/01/07 di -100,00E)

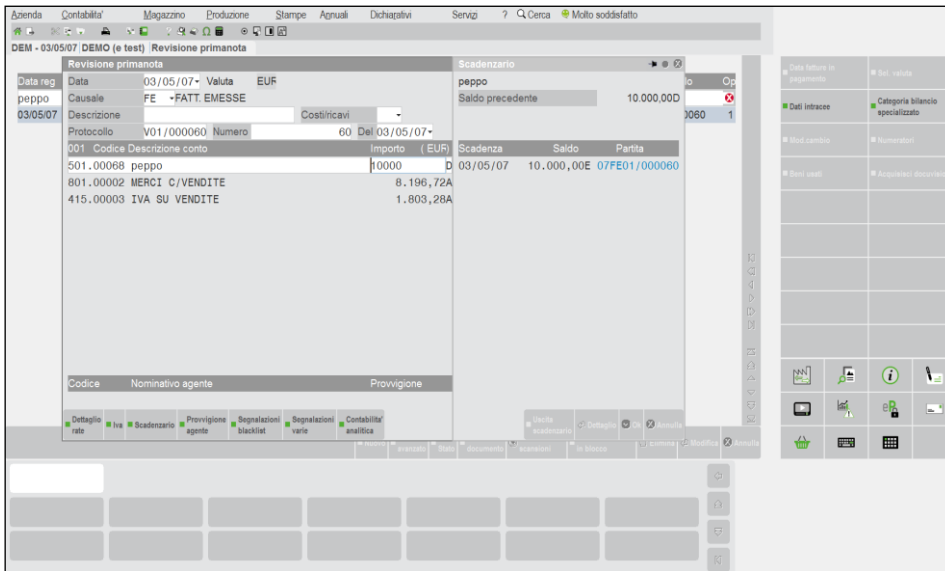
Scadenza	Saldo
28/02/07	400,00E
31/03/07	0,00E
30/04/07	400,00E

Partita: 07FR01/000003

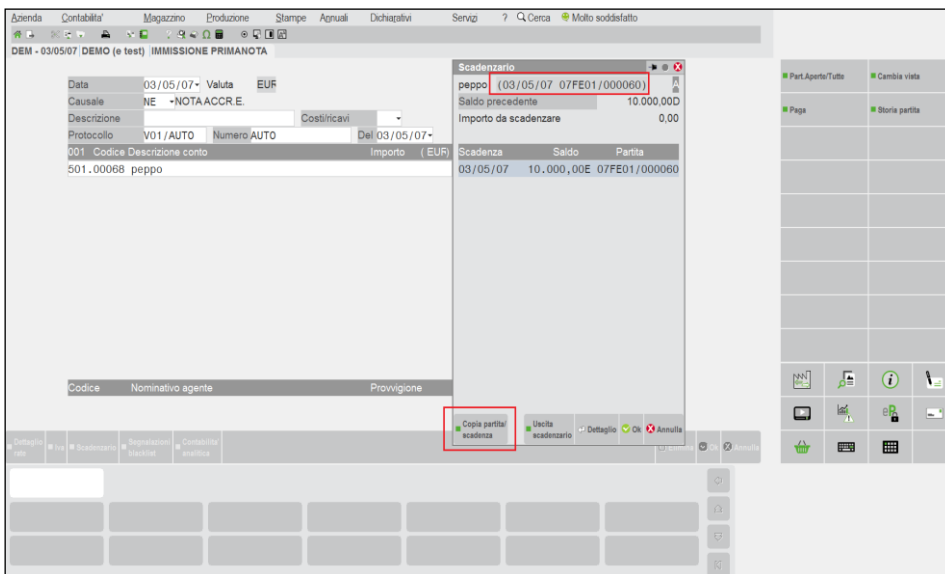
Fine Esempio Scadenzario a PARTITE

Esempio di FE con NE parziale (registrati da primanota) e di PG totale del residuo in scadenzario a partite tramite SPRIX. Per ottenere la corretta chiusura della partita, occorre che vi sia un collegamento alla partita principale e il relativo riferimento ad essa comprensivo di data scadenza nell'operazione in SPRIX

Registro la fattura, 1 scadenza rimessa diretta a vista



Registro la NE e ne collego la partita



Considero la differenza in accreditato

verifico la storia della partita

Eseguo lo sprix per pagare il residuo sulla stessa partita e scadenza

verifico i risultati, cioè, se la partita è chiusa

Codice	Descrizione conto	Importo (EUR)	C/R	Descrizione	Scadenza	Saldo	Partita
501.00068	peppo	10.000,00	D				
801.00002	MERCI C/VENDITE	8.196,72A					
415.00003	IVA SU VENDITE	1.803,28A					

Scadenza	Saldo	Partita
03/05/07	0,00E	07FE01/000060

Documento	del	Scadenza	Tipo	Importo	Data Pag
FE 1/	60	03/05/07	03/05/07 M	10.000,00E	
PG		03/05/07	03/05/07 MP	-5.000,00E	03/05/07
NE 1/	63	03/05/07	03/05/07 MP	-5.000,00E	03/05/07

Caso: PUTPN nelle aziende con Raggruppamento del Piano dei Conti

In presenza di aziende che hanno il raggruppamento del piano dei conti (ad esempio raggruppamento 8) il gestionale genera un file che gestisce le anagrafiche di tutti i conti con sigla raggruppamento disponibile per tutte le aziende che appartengono al raggruppamento di

interesse (in questo caso ru dt.008), anziché generare un file con estensione azienda. Inoltre ciascuna azienda ha un file (pico.sig) che contiene i conti dell'azienda di riferimento, quindi, se all'interno di un'azienda con raggruppamento (ad esempio azienda "OLD") si genera un nuovo conto, questi verrà salvato sia sul file "ru dt.008" sia sul file "pico.old". Se si cambia azienda di riferimento passando ad un'altra azienda che abbia raggruppamento 8 (ad esempio l'azienda OL1) all'interno del file "ru dt.008" vi saranno tutte le informazioni relative all'anagrafica del conto appena creato, tuttavia il file "pico.ol1" dell'azienda aperta non è aggiornato e quindi si può generare un errore durante la scrittura di una registrazione di primanota con Sprix.

La spiegazione di tale comportamento consiste nel fatto che quando si apre il menù "Anagrafica PC/Cli/Forn" e si cerca il codice conto di riferimento questi viene correttamente visualizzato e qualora questi non sia presente nel file "pico.ol1" il file viene aggiornato. Premesso che i conti si possono facilmente importare/esportare con il gestionale tramite il modulo Trasmissione Archivi da menu Servizi, per ovviare a questa particolarità con Sprix si può operare in modi differenti:

- si potrebbe creare il codice conto di riferimento (PUTPC)
- si potrebbe simulare l'apertura della videata "Anagrafica PC/Cli/Forn" utilizzando l'istruzione INPUT ed il campo di tipo "C" che attiva le funzioni del gestionale (in questo caso il file pico.ol1 verrebbe automaticamente aggiornato dalle funzioni standard anziché da Sprix). Di questa soluzione riportiamo un semplice esempio:

```
IF _ERRPN<>0 THEN GOTO CONTR_RAGGR
CONTR_RAGGR:
IF pacod(COD$)=" " THEN VIMSG 1,"CODICE NON PRESENTE!!!":END
PUTTAST "#501"
INPUT "C",COD$,9
END
```

caso: PUTPN di PG di nota d'accredito (NE/NR) su scadenziario a partite

La registrazione di PG relativa a più partite (relative a fatture e note d'accredito) non si comporta come l'operazione saldaconto, la quale consente minore operatività rispetto alla Primanota (e Sprix con PUTPN) in fatto di variazione dei dati delle partite. L'attuale operatività prevede questa differenza di comportamento in sede di registrazione del PG, cioè, se è effettuato da saldaconto le partite si compensano e si eliminano immediatamente, se è effettuato invece da primanota è richiesta la generazione di una partita di storno.

Non è disponibile un'istruzione Sprix che consenta di pagare in automatico le note di credito da immissione primanota (come nella procedura standard per pagare le fatture utilizzando il tasto funzione apposito) senza dover operare manualmente sulle rate con importi negativi e collegare le partite delle fatture che hanno importi positivi (tramite l'attribuzione della stessa partita e stessa scadenza); per la stessa ragione non è disponibile in Sprix una funzione come il Saldaconto cli/for in cui avviene tutto in automatico.

La PUTPN non ha le stesse prerogative sullo scadenziario come hanno le funzioni standard del gestionale come il saldaconto. Le operazioni eseguite dall'utente nel saldaconto possono, infatti, consistere in una semplice rettifica delle rate, e pertanto, questi tipi di operazione non sono consentiti con la PUTPN né con altre procedure Sprix.

Per avvicinarsi a simili risultati di saldaconto nelle operazioni massive di PG sulle partite (siano esse di fatture o note d'accredito) è comunque consentita la preventiva operazione di variazione partite/scadenze, in modo da accoppiare la partita della nota a quella della fattura. In pratica, prima della PUTPN, si opera un accorpamento attraverso il modulo di variazione e poi successivamente si effettua il PG delle partite relative alle fatture.

Caso: messaggio della riorganizzazione dello scadenziario a partite: Data pagamento (0) Valore fuori dai limiti, assunta data di registrazione

La registrazione dell'insoluto "IN" consente 2 tipi di registrazioni: una manuale (con indicazione partita) ed una con aggancio automatico della procedura dal nome "Collega IN". Quest'ultima inserisce la data del pagamento "Data pag" che comunque, in entrambi i casi non è visibile visionando il dettaglio delle rate. La PUTPN dello Sprix si comporta come se fosse una registrazione manuale, sicché, in questi due casi (manuale o da Sprix) non viene generata da programma la data di pagamento, che comunque viene ricostruita dalla riorganizzazione, senza alcuna conseguenza.

Caso: tracciato record archivi extracontabili

Gli archivi extra contabili sono archivi in chiaro. Possono essere letti da sprix con una normale open del file. Il tracciato record del file è il seguente:

```
06 caratteri data nella forma ggmmaa
09 caratteri conto
15 caratteri importo, il segno determina il valore dare/avere
03 caratteri centro di costo
24 caratteri descrizione
01 filler
01 CR
01 LF
```

Lunghezza 60 caratteri. Tale definizione può subire modifiche a seguito di implementazioni della Passepartout. Verificare sempre andando in editing del file che la lunghezza non sia modificata.

Caso: esempio scrittura fattura in doppia valuta

```
_PNDRE$=sxsdt      'data registrazione
_PNRPR$="v"        'protocollo
_PNSPR =1          'serie
_PNCAUS$="FE"      'causale
_PNND0=0           'numero documento
_PNDD0$=sxsdt     'data documento
```

```

_PNPAG=0      'pagamento
'scrittura contabile in dollari
_PNVDO=2
_PNEVE=1.1: _PNDIE=1 ' 1 euro = 1.1 dollari
_PNCTOS(1) = "501.00001" : _PNIMP(1) = 1500 : _PNIMD(1) = 1500*_PNEVE:
_PNCTOS(2) = "801.00011" : _PNIMP(2) = -1280 : _PNIMD(2) = -1280*_PNEVE:
_PNCTOS(3) = "415.00012" : _PNIMP(3) = -220: _PNIMD(3) = -220*_PNEVE:
'castelletto iva
_PNALIS(2,1)= "22" : _PNIMB(2,1)= 1000 : _PNIVA(2,1) = 220: _PNMDO(2,1)= 1000*_PNEVE : _PNIVD(2,1) = 220*_PNEVE:
_PNALIS(2,2)= "N71" : _PNIMB(2,2)= 280 : _PNIVA(2,2) = 0: _PNMDO(2,2)= 280*_PNEVE : _PNIVD(2,2) = 0:
'provvigioni monoagente
_PNAGES="601.00009" : _PNPRO=100: _PNPRD=_PNPRO*_PNEVE
'provvigioni multiagente
'_PNPRA$(1)="601.00009" : _PNPRI(1)=100 : _PNMPD(1)=_PNPRI(1)*_PNEVE
'contabilita' analitica (no CDG!)
_PNSCOS(2,1)="PIPP0" : _PNSIM(2,1)=_PNIMP(2) : _PNSIV(2,1)=_PNIMD(2)*-1

PUTPN : IF _ERRPN<>0 THEN VIMSG 1,_ERRPN$
END

```

Caso: esempio scrittura pagamento fattura in doppia valuta

```

_PNCAU$="PG"      'causale
_PNDRE$=sxsdt    'data registrazione
'scrittura contabile in dollari
_PNVDO=2
_PNEVE=1.1: _PNDIE=1
_PNCTOS(1)="501.00001" : _PNIMP(1)=-1500 : _PNIMD(1)=_PNIMP(1)*_PNEVE
_PNCTOS(2)="201.00001" : _PNIMP(2)=1500 : _PNIMD(2)=_PNIMP(2)*_PNEVE
'***** scadenziario a documento
_PNPCA$="FE" : _PNPSD=1 : _PNPND=19 : _PNPDD$=sxsdt
'***** scadenziario a partite, _PNSIP() va valorizzata con importo in valuta documento
'_PNSCT$(1)="501.00001": _PNSPA$(1)="22FE01/000003": _PNSDS$(1)="20221231"
'_PNSIP(1)=1500*_PNEVE '1650$

PUTPN : IF _ERRPN<>0 THEN VIMSG 1,_ERRPN$
END

```

AGENDA DI LAVORO (AGP)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 34	Istruzione che azzeri variabili agenda di lavoro _AGP;	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "AGP"	Istruzione che azzeri variabili struttura _AGP (come azzvarsys);	S/C/Cr/Cd Sm/Cm
GETAGP	Par: <nrec>,<dal\$>,<ala\$>,<cli\$>,<pre\$>,<ute\$>,<sta\$>,<ori\$>,<azi\$> Istruzione che legge agenda di lavoro; Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/Cr/Cd
PUTAGP	Par: [par1]. Istruzione che scrive agenda di lavoro; Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/Cr/Cd
DELAGP	Par: <nrec>,<dal\$>. Istruzione che cancella agenda di lavoro; Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/Cr/Cd
_ERRAGP	Codice risultato operazione agenda di lavoro;	S/Cd
_ERRAGP\$	Messaggio operazioni su agenda di lavoro;	S/Cd
_AGP	Variabili di struttura agenda di lavoro;	S/Cd

esempi e casi dizionario

Esempio: lettura

```
FOR K = 1 TO 100
  AZZVARSYS(34)
  GETAGP K,"", "", "", "", "", "", "", ""
  LPRINT _ERRAGP$+" ---- "+STR$(K,0,0)+" --- "
  LPRINT _ERRAGP
  FOR I = 1 TO _AGPNELE
    LPRINT _AGPDTLAV$(I)
    LPRINT _AGPID(I)
    LPRINT _AGPSTATOS$(I)
    LPRINT _AGPPRCOD$(I)
  NEXT
NEXT
```

Esempio: lettura e modifica

```
FOR K = 1 TO 7
  AZZVARSYS(34)
  GETAGP K,"", "", "", "", "", "", "", ""
  LPRINT "KAPPA="+STR$(K,0,0)
  LPRINT _ERRAGP$+" ---- "+STR$(K,0,0)+" --- "
  LPRINT _ERRAGP
  FOR I = 1 TO _AGPNELE
    LPRINT _AGPDTLAV$(I)
    LPRINT _AGPID(I)
    LPRINT _AGPSTATOS$(I)
    LPRINT _AGPPRCOD$(I)
  NEXT
NEXT

_AGPNELE=8 `Elementi trovati con la GETAGP
_AGPDTLAV$(8)=sxdat `Data lavori (campo chiave)
_AGPSTATOS$(8)="A" `Stato del lavoro A/F/P
_AGPPRCODS(8)="003" `Codice della prestazione
_AGPDATAAPA$(8)="S" `Riporta data Prestazione su notula/Parcella S/N
_AGPCODCLI$(8)="301.00004" `Codice Cliente
_AGPPRZ(8)=2000 `Prezzo
_AGPQTA(8)=2 `Quantita

PUTAGP 8
LPRINT _ERRAGP$+" ---- "+STR$(K,0,0)+" --- "
LPRINT _ERRAGP
```

Esempio Agenda di Lavoro (scrittura, modifica e cancellazione)

```
`Scrittura del record
AZZVARSYS 34
  _AGPSTATOS(1) = "F"
  _AGPDTLAV$(1) = "20140128"
  _AGPDATAAPA$(1) = "N"
  _AGPPRCODS(1) = "007"
  _AGPQTA(1) = 1
  _AGPPRZ(1) = 150
  _AGPCODCLI$(1) = "301.00003"
  _AGPNOTE$(1) = "nota XXXX"
  _AGPNOTAPA$(1) = "."
PUTAGP 1
VIMSG 1, _ERRAGP$

` Modifica del record
AZZVARSYS 34
GETAGP 0, "20140128", "20140128", "301.00003", "", "", "", "", ""
VIMSG 1, _ERRAGP$
VIMSG 1, STR$( _AGPNELE, 0, 0) + " record estratti"
ID = _AGPID(1)
VIMSG 1, "ID primo record: " + STR$(ID, 0, 0)
` modifiche da apportare
_AGPPRCODS(1) = "006"
_AGPNOTE$(1) = "nota YYYY"
PUTAGP 1
VIMSG 1, _ERRAGP$

` Cancellazione record modificato
AZZVARSYS 34
DELAGP 16, "20140128" ` ID e valore estratto da _AGPID()
VIMSG 1, _ERRAGP$
```


TABELLA PAGAMENTI (PA)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS	Istruzione che azzeri variabili tabella pagamenti _PA	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "PA"	Istruzione che azzeri variabili struttura _PA (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETPAG	Par: <codpag>. Istruzione che legge i dati della tabella pagamenti	S/Cr/Cd
PUTPAG	Par: [g\$] Istruzione che scrive il codice pagamento nella tabella pagamenti.	S/Cr/Cd
DELPAG	Par: <codpag> Istruzione che cancella il codice pagamento nella tabella pagamenti	S/Cr/Cd
_ERRPA	Codice risultato operazione sulla tabella pagamenti	S/Cd
_ERRPA\$	Messaggio operazione sulla tabella pagamenti	S/Cd
_PA	Variabili di struttura tabella pagamenti	S/Cd
esempi e casi dizionario		

ANAGRAFICA ARTICOLI (AR)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 3	Istruzione che azzeri variabili anagrafica articoli _AR	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "AR"	Istruzione che azzeri variabili struttura _AR (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETAR	Par: <codart\$>,<nummag>. Istruzione che legge anagrafica articoli	S/Cr/Cd/Sm/Sh
GETATTAR	Istruzione che legge anagrafica articoli a video	C
PUTAR	Par: <nummag>. Istruzione che scrive anagrafica articoli	S/Cr/Cd/Sm/Sh
DELAR	Par: <codart\$>. Istruzione che cancella anagrafica articoli	S/Cr/Cd/Sm/Sh
GETNUMSTRUT	Par: <codart\$>. Funzione che torna il numero di struttura di un articolo	S/C/Cr/Cd/Sh
GETARVAR	Par: [codart\$],<dadata\$>[,adata\$] Istruzione che legge anagrafica articoli variata Non consentita nei Livelli di sviluppo (F2, L3, F3)	S/Cr/Cd/Sh
_CODARVAR\$	Variabile prossimo articolo variato letto da GETARVAR	S/Cd
_ERRAVAR	Codice risultato operazione GETARVAR	S/Cd
_ERRAVAR\$	Messaggio operazione GETARVARI	S/Cd
DESART\$()	Par: (<codart\$>). Funzione che ritorna la descrizione totale dell'Articolo	S/C/Cr/Cd/Sh
GETCODART	Par: <tipo\$>,<codice\$>[,parametri\$]. Istruzione che trova i codici di Anagrafica Articoli accomunati dal valore del parametro in Input. es.	S/Cr/Cd/Sm/Sh
_ERRCODART\$	Eventuale messaggio d'errore di GETCODART	S
_CODARTNUM - _CODART\$()	Variabile Numero di codici articoli trovati - Array dei codici articoli trovati	S
ARTDBP()	Par: (<codart\$>). Funzione che controlla se l'articolo è di produzione	S/C/Cr/Cd/Sh
_ERRAR	Codice risultato operazione anagrafica articoli	S/C/Cd
_ERRAR\$	Messaggio operazione anagrafica articoli	S/C/Cd
_AR	Variabili struttura anagrafica articoli	S/C/Cd
esempi e casi dizionario		
tabella "ax"	Listini estesi in tabella "ax" in caso di lettura diretta o ciclotabellare. In seguito all'introduzione dei 999 listini, il secondo parametro (Listino) delle variabili di dizionario ax (Codice\$,Listino) è da interpretarsi come posizione listino (dei 18 visualizzabili nell'anagrafica articolo) e non come numero assoluto listino .	S/C/Cd
arum2 in articoli a taglie	La variabile arum 2 ritorna due caratteri. In articoli a taglie puo' essere: T1, T2, T3,...sino a T9, oppure: 10, 11, 12, 13, per le taglie che sono collegate alla serie da 10 in poi. Per valutare se un'articolo è a taglie è possibile effettuare un test se il primo carattere corrisponde a T o se la stringa è formata da numeri compresi da 10 a 99.	S/C/Cd
abcar(), absca(), abult()	Attenzione: consultare gli effetti della configurazione dei parametri dei documenti (menu MZD + [F5]) e del servizio di variazione magazzino di variazione parametri documenti (menu ZVMP). Se la absca() non viene visualizzata e/o aggiornata, ne dalla riorganizzazione, ne riconfermando il documento, ne con la putmm, ma solo registrando un nuovo documento, significa che i documenti che avrebbero dovuto visualizzare/incrementare la data sono stati generati con flag di aggiornamento data disattivato (MZD+[F5]); di conseguenza i loro mmagd() e mmaga() sono = 'N' (anche per var. di struttura), quindi risultano non modificabili nemmeno dalla riorganizzazione (che prova ad eliminare e ricostruire). La riorganizzazione si adegua alle "marcature" sui documenti, conformi ai parametri stabiliti all'epoca di generazione del documento. Questa situazione, può accadere nei casi di errore di configurazione o	S/C/Cd

	<p>"ripensamenti" sui parametri, oppure manipolazioni dei documenti senza considerare i riferimenti _MMNTE(), _MMMSI\$, _MMOSI\$(). Consultare indicazioni della PUTMM</p> <p>I menu ZVMP viene in soccorso per recuperare la situazione, elaborando gli ultimi progressivi dei due anni aperti; per i precedenti anni occorre una rinnovare la registrazione (duplicando o importando il contenuto del documento, poi da cancellare, per poi modificare il numeratore del nuovo documento).</p> <p>Per la valorizzazione della variabile abcar() su azienda infrannuale è necessario stabilire a priori quale anno si intende interrogare, se l'ultimo gestito o il penultimo. Esistono infatti due distinte variabili da utilizzare: se si intende interrogare l'ultimo anno gestito, si utilizza la variabile "abcar()", se invece interessa il penultimo, la variabile è "abult()". Può capitare che in Sprix nel ciclo tabellare su tabella 'ar', la variabile abcar() risulti vuota, anche se in anagrafica articolo la data è visualizzata correttamente. Il fatto che un valore sia visualizzato, non significa necessariamente che sia anche registrato su record. Premessa la sintassi abcar(arcod(),mag), ciò è dovuto dal fatto che la data di ultimo carico è da individuarsi per anno corrente (abcar) o anno precedente (abult); se nel ultimo anno non è effettuato alcun movimento, Sprix ritorna quello del anno precedente.</p> <p>Le date dei progressivi di ultimo carico a video non corrispondono alle date di ultimo carico del dizionario. I dizionari abult() e abcar() si riferiscono all'anno precedente (abult) e all'anno corrente (abcar). Il funzionamento è riassunto nello schema sottostante (tre anni in linea, ultimi due aperti con il 2016 appena creato e senza alcuna data di carico).</p> <pre> 2014 - 2015 - 2016 2014 abcar -> abult 2015 abcar -> abult 2016 abcar </pre> <p>Quando viene creato il nuovo anno, abcar() si deve azzerare per inicializzarsi per il nuovo anno (2016) per cederne il valore ad abult() relativo all'anno precedente (2015). Nel caso in cui nel 2014 sono avvenuti movimenti, ma non nel 2015, abult() del 2015 deve comunque riportare valori del 2014 (l'ultimo carico effettuato, anche se non relativo all'anno immediatamente precedente all'ultimo aperto).</p>	
awinv(), awcar(), awzca()	<p>Dizionario relativo ai progressivi (valore di inventario, valore di carico, valore di scarico). Attenzione: tali valori sono influenzati in base all'anno richiesto (l'anno chiuso o l'anno fra gli ultimi due aperti) dai magazzini coinvolti e/o il totale dei magazzini (non consentito come parametro nel dizionario), ma soprattutto, in base alle logiche di carico e di scarico dei documenti che sono stabilite nei Parametri documenti [tasto F5] delle Tabelle Aziendali di Magazzino relative alle Descrizioni dei documenti (ad esempio: le bolle di deposito fra magazzini). Consultare i meccanismi di Magazzino e relativi effetti sulle risultanze dei dati.</p>	S/C/Cd
<p>Esempio: lettura e modifica di un articolo di magazzino</p> <p>Quando si esegue la lettura di un articolo con l'istruzione GETAR codart\$,nummag è consigliabile eseguire una preventiva istruzione AZZVARSYS 3 per svuotare l'eventuale contenuto delle variabili _AR? lette in precedenza, altrimenti si potrebbe incorrere nel conservare dei contenuti di alcune variabili _AR? che non sono state aggiornate dall'istruzione GETAR.</p> <p>Fatto questo ed eseguita l'istruzione GETAR codart\$,nummag si dovrebbe controllare lo stato della variabile di sistema _ERRAR al fine di verificare se la lettura dell'articolo di riferimento sia, o meno, avvenuto correttamente. Infatti, se si dovesse ricercare un codice articolo inesistente in anagrafica articoli, oppure se si cercasse un articolo utilizzando un codice errato _ERRAR verrebbe valorizzata, viceversa è sempre uguale a zero. Inoltre _ERRAR è indispensabile anche quando si vuole creare o modificare un articolo (PUTAR). In sintesi detta variabile mostra se si stanno eseguendo tutti i passaggi relativi ai dati di magazzino in maniera corretta o meno.</p> <p>Di seguito riportiamo un semplice esempio:</p> <pre> ART\$="911RS" NUM MAG=1 AZZVARSYS 3 GETAR ART\$,NUM MAG IF _ERRAR<>0 THEN VIMSG 1,_ERRAR\$END _ARDES\$="Pluto" PUTAR NUM_MAG IF _ERRAR<>0 THEN VIMSG 1,_ERRAR\$ END </pre>		
<p>Esempio: Scrive l'articolo, lo rilegge e ne cambia la descrizione, eseguendone anche la precancellazione dell'articolo con articolo sostitutivo (utile per eseguire successive cancellazioni fisiche con DELAR oppure in alternativa tramite la funzione standard di servizio per la cancellazione fisica degli articoli precancellati)</p> <pre> ART\$="MELE" : MAG=1 AZZVARSYS 3 _ARCOD\$=ART\$: ARTIP\$="A" : ARDES\$="GOLDEN" : ARIVA\$="20" _ARUM1\$="kg" : ARDEC=2 : ARVST=2 : ARVUL=2 : ARVUP=2 PUTAR 1: IF _ERRAR<>0 THEN VIMSG 4, _ERRAR\$ AZZVARSYS 3 GETAR ART\$,MAG : IF _ERRAR<>0 THEN VIMSG 4, _ERRAR\$ </pre>		

```

_ARDESS="STARK" : _ARANN$="S" : _ARSOS$="MELE DI SCARTO"
PUTAR MAG: IF _ERRAR<>0 THEN VIMSG 1,_ERRAR$
END

.... DELAR ART$

```

Caso: Precancellazione, differenza tra arpre() e arann(), _ARPRE\$ e _ARANN\$

La variabile arpre(), _ARPRE\$ rappresenta il valore se precancellato o meno riferito agli articoli "figli" (cod, strutturati), nell'anagrafica articoli è rappresentato il valore "ANN."

La variabile arann(), _ARANN\$ rappresenta la precancellazione logica articoli. Quando si ha la necessità di eseguire una precancellazione logica di un articolo di magazzino da Sprix occorre impostare la variabile struttura _ARANN\$="S" ed eventualmente definire l'articolo sostitutivo (utilizzando la variabile struttura _ARSOS\$). A questo punto per confermare la precancellazione logica basta eseguire l'istruzione PUTAR. Se viene impostato l'articolo sostitutivo nell'anagrafica articoli sarà presente il valore "CAN+" altrimenti sarà visibile il valore "CAN-"

Di seguito viene visualizzato un semplice esempio:

```

GETAR "748_YELL",1
_ARANN$="S"
_ARCOS$="748_RED"
PUTAR 1
IF _ERRAR<>0 THEN VIMSG 1,"ERRORE NUMERO "+STR$( _ERRAR,0,0)
END

```

Caso: DELAR non effettua la cancellazione definitiva se l'articolo è stato movimentato ma non ha i progressivi

La DELAR non effettua la cancellazione di un articolo precancellato, che è stato movimentato ma non ha piu' progressivi (cioè è stato sostituito da un'altro articolo nei movimenti). Tale comportamento è una caratteristica del programma, per cancellare occorre utilizzare la funzione standard di cancellazione presente nei Servizi del gestionale.

Caso: DELAR non effettua la cancellazione definitiva se l'articolo è stato duplicato da altro movimentato ma non è stato a sua volta movimentato

Un articolo creato da duplicazione di un articolo esistente con la sequenza GETAR/cambio codice/PUTAR, può risultare essere non cancellabile pur non avendo dei movimenti a lui associati. In questo caso specifico si può procedere alla cancellazione da sprix avendo cura di azzerare le variabili dei progressivi

```

GETAR "A01",0 'articolo di partenza con movimenti e progressivi associati
IF _ERRAR<>0 THEN VIMSG 5,_ERRAR$:END
'azzeramento variabili dei progressivi
_ARPRU$=""
_ARINO=0
_ARINV=0
_ARTIN=0
_ARCIN=0
_ARCAR=0
_ARCAV=0
_ARTCA=0
_ARCCA=0
_ARSCA=0
_ARSCV=0
_ARTSC=0
_ARCSC=0
_ARORF=0
_ARCLI=0
_ARCLS=0
_ARCLA=0
_ARPRD=0
FOR A=1 TO 16
_ARINT(A)=0
_ARINTXL(A)=0
_ARCAT(A)=0
_ARCATXL(A)=0
_ARST(A)=0
_ARSTXL(A)=0
_ARORT(A)=0
_ARORTXL(A)=0
_ARCTI(A)=0
_ARCTIXL(A)=0
_ARCTS(A)=0
_ARCTSXL(A)=0
_ARCTA(A)=0
_ARCTAXL(A)=0
_ARCTP(A)=0
_ARCTPXL(A)=0
NEXT
'duplicazione articolo dopo cambio di codice
_ARCOD$="CANC"
PUTAR 0
IF _ERRAR<>0 THEN VIMSG 5,_ERRAR$:END
'cancellazione articolo duplicato
DELAR "CANC"
IF _ERRAR<>0 THEN VIMSG 1,_ERRAR$
END

```

Esempio GETARVAR: Legge gli articoli variati dal 01/01/2011

```

_CODARVAR$="" : DADATA$="20110101" : ADATA$="" : I=0
_LOOP:
GETARVAR _CODARVAR$,DADATA$
I=I+1
LPRINT STR$(I,0,0)+" ("+_ERRARVAR$+"): "+_CODARVAR$
IF _CODARVAR$<>"" THEN GOTO LOOP

```

Esempio lettura e stampa listini estesi

```

GETAR ART$,1 : VIMSG 1, _ERRAR$
GOSUB STAMPA
' GOSUB SCRIVI

```



```

END

STAMPA:
FOR I=1 TO 18
  LPRINT " _ARPRZE () "+" I="+STR$(I,0,0)+"="+STR$( _ARPRZE ( _ARNUMLISCON (I) ),2,0) + " _ARNUMLISCON"+"
I="+STR$(I,0,0)+"="+STR$( _ARNUMLISCON (I),0,0)
NEXT
  LPRINT "-----"
  LPRINT "per i listini dal 10 al 999"
  LPRINT "-----"
FOR I=1 TO 10
  LPRINT " _ARSCOES () "+" I="+STR$(I,0,0)+"="+_ARSCOES (I)
NEXT
FOR I=1 TO 10
  LPRINT " _ARSCONES () "+" I="+STR$(I,0,0)+"="+_ARSCONES (I)
NEXT
  LPRINT "-----"
  LPRINT "per gli sconti su listino dal 9 al 18"
  LPRINT "-----"
FOR I=1 TO 10
  LPRINT " _ARPROTPE$ () "+" I="+STR$(I,0,0)+"="+_ARPROTPE$ (I)
NEXT
FOR I=1 TO 10
  LPRINT " _ARPROFORME$ () "+" I="+STR$(I,0,0)+"="+_ARPROFORME$ (I)
NEXT
FOR I=1 TO 10
  LPRINT " _ARPROAGENE$ "+" I="+STR$(I,0,0)+"="+_ARPROAGENE$ (I)
NEXT
FOR I=1 TO 10
  LPRINT " _ARPROCONDE () "+" I="+STR$(I,0,0)+"="+STR$( _ARPROCONDE (I),2,0)
NEXT
FOR I=1 TO 10
  LPRINT " _ARPROE () "+" I="+STR$(I,0,0)+"="+STR$( _ARPROE (I),2,0)
NEXT
  LPRINT "-----"
  LPRINT "Per le provvigioni su listino dal 9 al 18"
  LPRINT "-----"
FOR I=1 TO 10
  LPRINT " _ARPERE () "+" I="+STR$(I,0,0)+"="+STR$( _ARPERE (I),2,0)
NEXT
  LPRINT "-----"
  LPRINT "Per variazione prezzo su listino dal 9 al 18"
  LPRINT "-----"
RETURN

SCRIVI:
  _ARPRZE(10)=21.11
  _ARSCONES(10)="22,12"
  _ARPROFORME$(10)="23,13"
  _ARPROE(10)=24.14
  _ARPERE(10)=25.15
PUTAR 1

  VIMSG 1, "Scrivi "+_ERRAR$
RETURN

```

Esempio: estrae su un file esterno l'elenco di tutti gli articoli di tipo "A", quindi (tramite un secondo ciclo tabellare) stampa l'elenco di tutti i codici articolo (arcod()) e tutti i codici trovati nel file esterno.

```

|N_FILE=AUTOFILE ()
|FILE$="C:\ARTICOLI.TXT"
|KEY$="C:\ARTICOLI.K1"
|IF ESISFILE(FILE$) THEN KILL FILE$
|IF ESISFILE(KEY$) THEN KILL KEY$
|OPEN "K",N_FILE,FILE$,KEY$,1;amdco,amdco+2
|FIELD N_FILE,amdco AS COD$
|FIELD N_FILE,2 AS EOL$
|
|'CREAZIONE DEL FILE ESTERNO
|+-----+
|Cod.Tab.:ar ARTICOLO - ANAGRAFICA
|Ordine 3:ardes()
|Scelte..:artip()="A"
|+-----+
ar |COD$=LEFT$(arcod()+STRING$(amdco," "),amdco)
ar |EOL$=CHR$(13)+CHR$(10)
ar |INSKEY N_FILE
|
|'LETTURA DEL FILE ESTERNO E STAMPA A VIDEO
|+-----+
|Cod.Tab.:ar ARTICOLO - ANAGRAFICA
|+-----+
ar |COD$=LEFT$(arcod()+STRING$(amdco," "),amdco)
ar |GETKEY N_FILE,1,COD$
ar |IF _ERRIND<>0 THEN COD$=""
ar |N ~arcod()\\ - ~COD$\\

```

Esempio: tramite ciclo tabellare carica su un file esterno il prezzo del primo listino di un articolo; quindi importa tale prezzo sul listino numero 3 memorizzandolo sulla variabile struttura _ARPRZ(3).

```

|N_FILE=AUTOFILE ()
|FILE$="C:\ARTICOLI.TXT"
|IF ESISFILE(FILE$) THEN KILL FILE$
|OPEN "R",N_FILE,FILE$,26
|FIELD N_FILE,12 AS ART$
|FIELD N_FILE,12 AS PRZO NUM4
|FIELD N_FILE,2 AS EOL$
|EOL$=CHR$(13)+CHR$(10)
|+-----+
|Cod.Tab.:ar ARTICOLO - ANAGRAFICA
|+-----+
ar |ART$=arcod()
ar |PRZO=arprz (1) ()
ar |NUMREC=NUMREC+1
ar |PUT N_FILE,NUMREC

```

```

|FOR I=1 TO NUMREC
|GET N_FILE,I
|AZZVARSYS 3
|GETAR ART$,1
|_ARPRZ(3)=PRZO
|PUTAR 1
|IF _ERRAR<>0 THEN VIMSG 1,_ERRAR$
|NEXT
|CLOSE N_FILE
|END

```

Caso: determinazione riferimenti in finestra codici strutturati (esempio MATIP)

Per determinare il riferimento alla struttura in uso e al campo in input corrente nelle finestre di tipo "input strutturato" vanno utilizzate le variabili "sxstr" e "sxcmp", come alternativa alle istruzioni SCREEN\$ e GETPOSCURSY() poiché tali finestre potrebbero trattare l'indice di riga in modo differente in base al tipo di interfaccia (desk, lite, ecc.). Le due variabili ritornano rispettivamente il numero della struttura e il numero del campo video corrente. Tali finestre trattano l'indice di riga in modo differente tra l'interfaccia MexalDesk rispetto quella a carattere o MexalLite.

Se GETPOSCURSY sulla nuova interfaccia MXI non è allineata di un campo, occorre considerare che la GETPOSCURSY ritorna il valore assoluto della coordinata di riga relativa alla posizione video del cursore. E' bene tenere presente che il numero di righe di una stessa videata rappresentata con l'interfaccia MXIDesk può differire da quella a MXILight per via di alcuni aspetti grafici utilizzati in formattazione grafica (vedi per esempio fincature o effetti di sovrapposizione riga). Il numero ritornato dalla funzione in oggetto è sempre relativo all'interfaccia MXILight in quanto viene conteggiato il reale numero di righe presenti nella videata e non quanto appare con l'impiego dell'interfaccia MXIDesk. Per ritornare porzioni di stringhe video, è necessario utilizzare la funzione SCREEN\$ passando il valore di riga ritornato dalla GETPOSCURSY.

Caso: anni a cui si riferiscono i valori di _ARCUL _ARCUP _ARCUC

Supponendo gli anni 98/99/00/01

```

_ARCUL costo ultimo 01
_ARCUP costo ultimo riferito 98/99/00
_ARCUC (sola lettura) costo ultimo riferito all'anno in cui si è entrati (del 98 se entrato in 98, del 99 se entrato in 99, del 00 se entrato in 00)

```

Sono gestiti solo i costi ultimi due anni (i precedenti assumono quello del penultimo).
_ARCUC dipende dall'anno d'ingresso; nel penultimo gestito _ARCUC = _ARCUP.

Caso: PUTTAST su costo ultimo

Il seguente Collage intercetta l'evento, ma non aggiorna il costo ultimo dell'anno seguente: ON_OUT_MACOULT: PUTTAST "#602" : END
Occorre pertanto sostituire il PUTTAST "#602" con _TF=602, ma solo la prima volta:
ON_STARTAR: IABI=1 : END
ON_OUT_MACOULT: IF IABI THEN _TF=602: IABI=0

Caso: richiamare la struttura articolo dal codice alternativo senza ciclo tabellare

La suddetta operazione non è consentita. Occorre effettuare il ciclotabellare su "ar" e nella scelte stabilire aralt() = CODICE\$. La variabile da associare alla GETAR può essere solo il codice principale univoco; per sfruttare il codice alternativo come parametro di ricerca in anagrafica è necessario un preventivo ciclotabellare sulla tabella "ar" per esportare arcod() e aralt() su file ad indici, in modo da ottenere il codice principale dell'articolo (da passare alla GETAR) partendo dal codice alternativo.

Caso: come distinguere di un articolo strutturato, l'articolo padre dagli articoli figli

La possibilità di distinguere direttamente un padre da un figlio non è consentita da ciò che li accomuna (appartenenza a una struttura), ma di ciò che li differenzia (la lunghezza codice oltre la radice minima di base). E' proprio l'individuazione di un codice superiore alla radice minima di base, che distingue un figlio dal padre, ed è l'unica discriminante di cui ci si può avvalere.

```

ar   |IF LEN(arcod()) <= amdra THEN GOTO STAMPA ELSE GOTO SALTA
ar   |STAMPA:
ar   |N|~arcod()
ar   |SALTA:

```

Caso: inventario alla data

Il concetto consiste nel considerare tutta la situazione inventariale di tutti gli articoli all'inizio dell'anno, e poi, di sommare e sottrarre tutti i movimenti stabiliti dalla data in questione. In Sprix occorre effettuare due cicli: inizialmente uno sulla tabella ar, dove il codice fondamentale è aginv(arcod(),0,sxann) ... e tale ciclo non è necessariamente da ripetere ogni volta che si esegue la stampa (salvo siano intervenute rettifiche di inventario, i dati potrebbero essere precedentemente memorizzati per elaborazioni successive); successivamente occorre ciclare sulla tabella mm selezionando solo i movimenti compresi entro i parametri di selezione "dalla data" "alla data". A seconda del tipo documento, tali movimenti andranno a sommarsi o sottrarsi ai valori ottenuti dal primo ciclo (o precedentemente memorizzati).

Caso: come intercettare la variazione o l'inserimento di una anagrafica

La possibilità di intercettare il salvataggio di un nuovo articolo (ad esempio avendo la necessità di inserire un numero progressivo ogni volta che si salva un nuovo articolo si renderebbe necessario intercettare alla pressione del tasto F10 se l'anagrafica esiste o è nuova) oppure di uno già esistente è costituita dalla seguente considerazione: non esiste differenza tra la variazione o inserimento di un'anagrafica al momento della pressione del tasto F10.

L'unico modo per capire quale dei due eventi si sta gestendo, è durante la digitazione di un codice articolo intercettare con la SCREEN\$ l'area video sopra la riga della descrizione che in caso d'inserimento riporta la dicitura "I0". In questo modo tramite propria programmazione si può gestire un flag che testato alla conferma con il Tasto F10 permetterà di capire quali dei due eventi si sta gestendo.

Tabella "lw" DATI WEB ARTICOLI

I parametri fondamentali sono codice articolo e numero di lingua da 0 (l'italiano) a 9 ed è consentito effettuare anche la lettura ad accesso diretto inserendo il codice articolo e il numero di lingua come parametro dei dati WEB. Nella versione 2006A la lettura è consentita solo per la modalità di IMMISSIONE (la modalità AUTOMATICA da anagrafica è prevista per futura implementazione); quindi, se nella parte iniziale della maschera relativa ai dati web è impostato il flag "A" (dati automatici da anagrafica), non avviene la lettura dei dati.

```

ar AS=arcod()
ar FOR I=0 TO 9

```

```

ar N Diretto CODICE ARTICOLO ~A$\\
ar N ~lwart(A$,I)
ar N Codice ~lwcod(A$,I)\\ lingua: ~ Desc ~lwdes(A$,I)\\
ar N UM ~lwumi(A$,I)\\ prezzo ~lwprz(A$,I)\\
ar N -----
ar N Marca ~lwmar(A$,I)\\
ar N Modello ~lwmod(A$,I)\\
ar N Scheda ~lwfsch(A$,I)\\
ar N D1 ~lwfa1(A$,I)\\
ar N D2 ~lwfa2(A$,I)\\
ar N D3 ~lwfa3(A$,I)\\
ar N D4 ~lwfa4(A$,I)\\
ar N Email ~lwema(A$,I)\\
ar N Inter ~lwint(A$,I)\\
ar N Dt cre ~lwdtc(A$,I)
ar N Dt varia ~lwdtv(A$,I)
ar N -----
ar N NEXT

```

Caso: Immagine, Catalogo, Icona

La definizione della variabile _ARIMM\$ o _ARCAT\$ o _ARICO non testa che il file definito nella variabile debba esistere o meno. Se a seguito della GETAR si interroga una delle variabili queste saranno valorizzate con il percorso e il nome del file che passepertout si aspetta, indipendentemente che il file esiste o meno o che l'immagine sia associata all'articolo o meno. Se indichiamo ad una delle variabili un valore, cioè il percorso ed il nome file immagine, se la procedura trova l'immagine l'associa all'articolo copiandola nella cartella immagini, altrimenti ignora l'operazione. Se l'articolo ha un'immagine associata e si azzerava la variabile l'associazione viene cancellata. Per verificare l'esistenza del file si può eseguire il comando ESISFILE prima di assegnare la variabile.

Errore non correggibile Tipo rec.prog. diverso da 'C' (*) Riferimenti: mx45.exe eseg_sprix_mydb() pxserv3d.c spcanc()

Alcune modifiche (più o meno complete/corrette) ai documenti di magazzino che generano dei movimenti, nel tempo potrebbero "rovinare" il file dei progressivi Articoli. In particolare, il messaggio si riferisce al fatto che il programma, nella lettura dei progressivi, invece di trovare un record per i colli si trova un record cancellato.

Occorre pertanto riorganizzare gli ARTICOLI

Caso: variabili array _ARSCQ\$(), _ARSCQN(), _ARSCO\$(), _ARSCON\$() (consultare anche il caso: variabili array _PARTPRZ\$(), _PARTPRZN\$())

Se negli articoli sono state utilizzate valorizzazioni che sono riferite alla nuova gestione degli sconti a cascata (versione 2006C), per valorizzare i dati occorre utilizzare le nuove variabili ARSCQN\$() al posto delle vecchie ARSCQ\$. Se l'articolo è stato modificato con nuove variabili, questo non può essere aggiornata con le vecchie variabili. Lo stesso per le array ARSCOS(), ARSCON\$() e le particolarità PARTPRZ\$() e PARTPRZN\$().

Quando la rappresentazione dell'informazione non è più possibile, le vecchie variabili riportano valore "*****". L'array ARSCQN\$() appartiene al nuovo formalismo degli sconti: le nuove variabili sono sempre impostate con il nuovo formalismo, mentre le vecchie tornano lo sconto nel vecchio formalismo e con "*****" se impostati più sconti con decimali. Esempio: con sconto 10,20+20 le vecchie variabili riportano "*****", le nuove variabili riportano 10,20+20.

Se lo sconto ARSCQ\$() è rappresentabile, nonostante ARSCQN\$() sia correttamente impostato potrebbe venire ignorato rispetto a ARSCQ\$(), in quanto quest'ultimo prevarrebbe se non venisse azzerato, ovvero inizializzando ARSCQ\$()="".

Se lo sconto è rappresentabile in entrambe le variabili, per azzerare lo sconto potrebbe non essere sufficiente azzerare una sola fra le due variabili, bensì potrebbe essere necessario azzerarle entrambe, ovvero inizializzando ARSCQ\$(X,Y)="" e ARSCQN\$(X,Y)=". **Notare: quanto detto non è limitato al solo azzeramento ma anche alla eventuale variazione del valore già presente.**

Esempi: gestione listini estesi

Nell'evoluzione del gestionale, determinate funzioni standard possono essere ampliate ed estese.

Per esempio, il potenziamento della gestione dei listini è una fra le implementazioni che ha richiesto una modifica dell'operatività a video del cursore, pertanto, ciò che in passato si poteva fare con una determinata metodologia (per esempio una concatenazione di tasti) potrebbe non essere più utilizzabile in una nuova funzione più estesa. Per esempio, la nuova gestione dei listini in Collage non è più gestita con un stringa del tipo PUTTAST "#608#608#501123#501321#610#610" (ad esempio per inserire il valore 123 nel "listino 1" ed il valore 321 nel "listino 2"), poiché potrebbe interferire o sovrapporsi a quanto il gestionale stia già eseguendo "under ground" nella nuova maschera di gestione per fare scorrere il cursore fra i campi. In tali casi, conviene spezzare la PUTTAST in più routine.

ESEMPIO (1) COLLAGE SU MENU 'MA'

```

ON IN_MALISTIN:
IF IDX = 0 THEN END ' se contatore zero fermo tutto è acceso dalla pressione del tasto F12
IF IDX >= 9 THEN END ' se il numero del contatore è maggiore o uguale ai listini fermo tutto
IDX = IDX +1 ' incremento del listino
PRZ = PRZ +2 ' valore del listino che volete calcolare
_SINP$ = STR$(PRZ,0,0) ' imposto l'input
PUTTAST "#501" ' vado al campo successivo cioè vado in loop per il numero dei listini.
END

```

```

ON 612_MALISTIN:
IDX = 1
PRZ = PRZ +2
_SINP$ = STR$(PRZ,0,0)
PUTTAST "#501"
END

```

ESEMPIO (2) COLLAGE SU MENU 'MD' / 'MA'

```

-----
'Collage di REVISIONE LISTINO dal campo prezzo della BOLLA FORNITORE
'Uscire dal campo presso con INVIO, modificare il Listino 2 e confermare con F10
'Codificare almeno 2 listini da menu 'MZL'.
'
' SONO PRESENTI 3 METODI DI REVISIONE: DECOMMENTARE QUELLO DI INTERESSE E
' DECOMMENTARNE IL CORRISPONDENTE CODICE (REVISIONE_1A, REVISIONE_1B, REVISIONE_2)
-----

ON_START:
DIM LIS(2) 'dimensiono array (due listini)
DIM SCOART(10)
DSL$="Listino "+LEFT$(mldes(1,1)+".",7)+"": "

```

```

DS2$="Listino "+LEFT$(mlds(2,1)+".....",7)+": "
NDP=amndp
IF NDP=0 THEN TPI$="N": GOTO SLT_ST ELSE TPI$="N"+STR$(NDP,0,0)
ARR=1: FOR I=1 TO NDP: ARR=ARR/10: NEXT
SLT_ST:
REV=0: E_BF=0
'
_INTER=0
END

ON_ENDMM:
REV=0: E_BF=0
END

ON_OUT_MDRICCOD:
IF DOC$="BF" THEN E_BF=1 ELSE E_BF=0
END

ON_OUT_MDRIDOC:
IF _SINP$<>" " THEN REV=1 ELSE REV=0
END

ON_OUT_MDTDOC:
DOC$=LEFT$( _SINP$,2)
IF DOC$="BF" THEN E_BF=1 ELSE E_BF=0
END

ON_OUT_MDCODART:
RIC=0: IVA=0: AZZARRAY SCOART: ISCO=0
IF E_BF=0 THEN END
ART$=_SINP$
CS$=arsta(ART$): CN=arstn(ART$)
RICN=CN
IF CS$="A" THEN RICN=RICN+100
RIC=FIX((100.+RICN) / 100.,2)
IVA$=ariva(ART$): IF LEFT$(IVA$,1)=" " THEN IVA=VAL(MID$(IVA$,2,4))
PIVA=FIX((100. + IVA) / 100.,2)
SCO$=afscn(4)(ART$,1): GOSUB CALC_SCO
END

CALC_SCO:
IF SCO$="" THEN GOTO E_CALC_SCO
ISCO=ISCO+1
I=INSTR(SCO$,"+"): IF I=0 THEN I=LEN(SCO$)+1
SC$=LEFT$(SCO$,I-1): SCO$=MID$(SCO$,I+1,100)
SCOART(ISCO)=VAL(SC$): GOTO CALC_SCO
E_CALC_SCO:
RETURN

ON_OUT_MDPZZO:
IF E_BF=0 THEN END
'
PRZ=VAL( _SINP$)
FOR I=1 TO ISCO
PRZ=ARRO(PRZ * ((100 - SCOART(I)) / 100),2,ARR)
NEXT
RI=GETPOSCURS(2): CO=GETPOSCURS(2)
APRIFIN 1,1, NFIN,RI+1,CO,4,50
TIT$="LISTINI VENDITA (modifica, premi Invio e poi F10)":
LOCATE 1,(50-LEN(TIT$))/2: COLOR 2: PRINT TIT$;
COLOR 0
LOCATE 2,1: PRINT "Ricarico: ";: PRINT RICN;: PRINT "%";
LOCATE 3,34: PRINT "+RIGHT$( " "+STR$(arprz(1)(ART$),NDP,1,14)+" " );
LOCATE 4,34: PRINT "+RIGHT$( " "+STR$(arprz(2)(ART$),NDP,1,14)+" " );
' LIS(1)=ARRO(PRZ * RIC / 100.,3,ARR)
LIS(2)=ARRO(PRZ * RIC * PIVA,3,ARR)
SETTF "505Uscita,610Conferma"
_SCORRI=1: _TF=1

INP01:
LIS(1)=ARRO(LIS(2) / PIVA,2,ARR)
LOCATE 3,1: PRINT DS1$;
PRINT RIGHT$( " "+STR$(LIS(1),NDP,1,14);

INP02:
LOCATE 4,1: PRINT DS2$;
INPUT TPI$,LIS(2),14
IF _TF=-1 THEN GOTO INP01
IF _TF=505 THEN GOTO USCITA

'TRE METODI DI REVISIONE

IF _TF=610 THEN GOTO REVISIONE_1A 'CON LE PUTTAST DALLA INPUT 'M' (si esce dal controllo eventi, perchè sarebbe il "Collage di un Collage")
IF _TF=610 THEN GOTO REVISIONE_1B 'CON LE PUTTAST DALLA INPUT 'M' (si esce dal controllo eventi, perchè sarebbe il "Collage di un Collage")
IF _TF=610 THEN GOTO REVISIONE_2 'CON IL CAMPO MONITOR DI 'MD' (che continua a gestire gli eventi) equivalente a 'MA'

_SCORRI=0: GOTO INP01

USCITA:
CHIUDIFIN 1
_TF=501
END

'#####
REVISIONE_1A: 'metodo da campo INPUT
'#####
'(PUTTAST posizionate prima di APRIFIN e INPUT, ma posizionabili anche fra APRIFIN e INPUT)
PUTTAST "#604#608#608#501"
PUTTAST STR$(LIS(1),2,0)+"#501"+STR$(LIS(2),2,0) '+'#501'+ etc....
PUTTAST "#610#610#610#501" ' conferma e uscita
APRIFIN 2,2, NFIN,0,0,1,41
LOCATE 1,1: INPUT "M",ART$,41
CHIUDIFIN 2
CHIUDIFIN 1
END

'#####

```

```

' REVISIONE_1B: 'metodo da campo INPUT
' #####
' (serie di PUTTAST posizionate fra APRIFIN e INPUT)
' PUTTAST "#604#608#608#501" ' (prima di APRIFIN e INPUT, ma posizionabile anche fra APRIFIN e INPUT)
' FOR I=1 TO 3
'     IF I >= 3 THEN GOTO VAI_INPUT
'         PUTTAST STR$(LIS(I),2,0)
'         PUTTAST "#501"
'     NEXT
' VAI_INPUT:
'     PUTTAST "#610#610#610#501"
'     APRIFIN 2,2, _NFIN,0,0,1,41
'     LOCATE 1,1: INPUT "M",ART$,41
'     CHIUDIFIN 2
'     CHIUDIFIN 1
' END
' #####
' REVISIONE_2: 'metodo da campo MONITOR (F2 sul campo ricerca)
' #####
'ON_???_MA???: PUTTAST "#608#608" : END ' (in caso di revisione listini da menu 'MA')
' CHIUDIFIN 1
' _TF=-1
' PUTTAST "#504#504#504#604#608#608"
'END
'
'ON_IN_MAGESTLIST: 'all'apertura listini
'     IF IDX >= 2 THEN PUTTAST "#610#610#610" ELSE PUTTAST "#501"
'     PUTTAST "#501"
'END
'
'ON_IN_MALISTIN: 'giro sui listini
'     IF IDX >= 2 THEN PUTTAST "#610#610#610" : GOTO FINE ' se il numero del contatore e maggiore o uguale ai listini
fermo tutto
'     IDX=IDX+1 ' incremento del listino
'     _SINP$ = STR$(LIS(IDX),NDP,0) ' imposto l'input
'     PUTTAST "#501" ' vado al campo successivo cioe vado in loop per il numero dei listini
'FINE:
'END

```

Esempio: GETCODAR

```

' Cerco il codice INPUT nei codici ALTERNATIVI
IF (amalt="S") THEN GETCODART "S",LEFT$(COD$,16) : IF (_ERRCODART$<>"") THEN ERR=1: ERR$=_ERRCODART$:
IFF (ERR=0)
IF (_CODARTNUM>0) THEN TIP_T$="S":
IFF (_CODARTNUM=0)

' Cerco il codice INPUT nei codici PRIMARI
GETCODART "P",LEFT$(COD$,amdco) : IF (_ERRCODART$<>"") THEN ERR=1: ERR$=_ERRCODART$:
IFF (ERR=0)
IF (_CODARTNUM>0) THEN TIP_T$="P":
IFF (_CODARTNUM=0)

' Cerco il codice INPUT nei codici FORNITORE
GETCODART "F",LEFT$(COD$,30) : IF (_ERRCODART$<>"") THEN ERR=1: ERR$=_ERRCODART$:
IFF (ERR=0)
IF (_CODARTNUM>0) THEN TIP_T$="F":
IFF (_CODARTNUM=0)

' Cerco il codice INPUT nei codici LOTTO/MATRICOLA (se lotti/matricole gestite)
IFF (amlot="B" OR amlot="E" OR amlot="M")
GETCODART "L", " "+LEFT$(COD$,50) : IF (_ERRCODART$<>"") THEN ERR=1: ERR$=_ERRCODART$:
IFF (ERR=0)
IF (_CODARTNUM>0) THEN TIP_T$="L":
ENDEF
ENDEF
ENDEF
ENDEF
ENDEF
ENDEF
ENDEF
ENDEF
ENDEF

```

ANAGRAFICA UNICA (ZA)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 36	Istruzione che azzerava variabili anagrafica unica _ZA	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "ZA"	Istruzione che azzerava variabili struttura _ZA (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETZA	Par: <id_anagrafica>, <data> Istruzione che legge l'anagrafica unica alla data	S/Cr/Cd
PUTZA	Par:[flag_storicizza] Istruzione che scrive l'anagrafica unica e storico anagrafica	S/Cr/Cd
DELZA	Par: <id_anagrafica>,<data> Istruzione che cancella lo storico anagrafica unica	S/Cr/Cd
_ERRZA	Codice risultato operazione anagrafica unica	S/Cr/Cd
_ERRZA\$	Messaggio operazione anagrafica unica	S/Cr/Cd
_ZA	Variabili di struttura anagrafica unica	S/Cr/Cd

esempi e casi dizionario

Esempio: scrittura nuova anagrafica unica in archivio.

```
'i dati obbligatori/minimi per la scrittura di una anagrafica sono:
' - tipo soggetto
' - codice fiscale, per le persone fisiche
' - partita iva, per le societa' di capitali
' - ragione sociale o nome+cognome
AZZVARSYS 36
_ZAPFI$="S"
_ZASES$="M"
_ZANOM$="PAOLO"
_ZACGN$="ROSSI"
_ZACFS$="PLARSS60A01H294E"
PUTZA
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$
END
```

Esempio: aggiornamento anagrafica unica corrente presente in archivio.

```
AZZCATVARSYS "ZA"
ID=3 'PAOLO ROSSI
GETZA 3,""
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$:END
_ZANOT$(1)="Inserimento nota come aggiornamento anagrafica corrente"
PUTZA
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$
END
```

Esempio: aggiornamento anagrafica unica corrente presente in archivio e scrittura storico del dato variato.

```
AZZCATVARSYS "ZA"
ID=3 'PAOLO ROSSI
DATA$="20231018"
GETZA 3,""
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$:END
'salvo la nota attuale prima di aggiornarla in una variabile che utilizzerò nella scrittura dello storico
V_NOTA$= ZANOT$(1)
_ZANOT$(1)="Nota attuale"
PUTZA
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$:END
_ZA_NNOT$="Nota relativa alla storicizzazione"
_ZA_NDAT$=DATA$
_ZANOT$(1)=V_NOTA$
PUTZA "STOR=1"
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$
END
```

Esempio: lettura dati dallo storico.

```
'attenzione se data di storicizzazione = data attuale, verra' recuperata l'informazione dall'anagrafica corrente
AZZCATVARSYS "ZA"
ID=3
DATA$="20231018"
GETZA ID,DATA$
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$ ELSE VIMSG 1, _ZANOT$(1)
END
```

Esempio: cancellazione dati dallo storico.

```
'cancello tutte le storicizzazioni esistenti per l'id 3 con data 18/10/2023
'DELZA consente la cancellazione dei soli dati dello storico
'se si desidera eliminare definitivamente un'anagrafica è necessario annullarla logicamente ( _ZAANN$="S") e procedere alla
'cancellazione definitiva dal menù Servizi > Cancellazioni > Cancella anagrafica unica (Alt+ZNN)

ID=3
DATA$="20231018"
DELZA ID,DATA$
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$
END
```

Esempio: aggiornamento anagrafica corrente e scrittura automatica storicizzazione.

```
AZZCATVARSYS "ZA"  
ID=4  
DATA$="20231201"  
GETZA ID,""  
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$:  
_ZA_NNOT$="Da storicizzazione automatica"  
_ZA_NDAI$=DATA$  
_ZANOT$(1)="Nuova nota inserita in data "+RIGHT$(DATA$,2)+"/"+MID$(DATA$,5,2)+"/"+LEFT$(DATA$,4)  
PUTZA "STOR=R"  
IF _ERRZA<>0 THEN VIMSG 1,_ERRZA$  
END
```

ANAGRAFICA UBICAZIONI (PU)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 72	Istruzione che azzerava variabili anagrafica ubicazioni _PU	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "PU"	Istruzione che azzerava variabili struttura _PU (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETPU	Par: <idubi>, <prog\$,[idart]> Istruzione che legge anagrafica ubicazioni	S/Cr/Cd/Sh
GETATTPU	Istruzione che legge a video i dati dell'anagrafica ubicazioni	C
PUTPU	Istruzione che scrive anagrafica ubicazioni	S/Cr/Cd/Sh
DELP	Par: <idubi> Istruzione che cancella anagrafica ubicazioni	S/Cr/Cd/Sh
CTRARTTOUBI	Par: <CodArt\$>, <Idubi>, <Idlotto\$>, [quantità] Istruzione per controlli inserimento ubicazione su articolo	S/Cr/Cd/Sh
_ERRPU	Codice risultato operazione anagrafica ubicazioni	S/Cd
_ERRPU\$	Messaggio operazione anagrafica ubicazioni	S/Cd
_PU	Variabili di struttura anagrafica ubicazioni	S/Cd

esempi e casi dizionario
Nota:

Nelle tabelle "pq", "pg", "po" se Id Lotto = -1 viene ritornato il totale progressivi per Articolo, Id Ubicazione, Magazzino, Anno
 Altro dizionario coinvolto per ciclitabellari: "mmdub", "ocrau", "ofrau", "omrau", "oprau", "orrau"

Esempio: registra un'ubicazione, un articolo e un movimento

```

_PUMAG=1 : _PUCOD$="CXCX8888"
PUTPU : IF _ERRPU$<>"" THEN VIMSG 1, _ERRPU$
VIMSG 1, STR$(_PUID,0,0) : ID=_PUID

_ARCOD$="ARTCONUBIC":_ARTIP$="A":_ARDES$="descrizione":_ARALT$="ARTCONUBIC":_ARIVA$="22":_ARUM1$="pz"
_ARVST=2:_ARVUP=2:_ARVUL=2:_ARPRZ(1)=100:_ARUID=ID
PUTAR 1 : IF _ERRAR$<>"" THEN VIMSG 1, _ERRAR$

_MMSIG$="BC":_MMSER=1:_MMNUM=I:_MMDAT$=sxdat:_MMCLI$="201.00011":_MMMAG=1:_MMNOT$(1)="Nota"
_MMTPR$(1)="R":_MMART$(1)=_ARCOD$: _MMQTA(1)=10:_MMALI$(1)="22":_MMPRZ(1)=101:_MMRUBI(1)=ID

PUTMM 0 : IF _ERRMM<>0 THEN VIMSG 1, _ERRMM$
END
  
```


CODICI ALIAS (AL)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 21	Istruzione che azzeri variabili codici alias _AL	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "ALIAS"	Istruzione che azzeri variabili struttura _AL (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETALIAS	Par: <codart\$>,<codalis\$>,<nprog> Istruzione che legge anagrafica codici alias Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
PUTALIAS	Istruzione che scrive anagrafica codici alias Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELALIAS	Par: <codart\$>,<codalis\$>,<nprog> Istruzione che cancella anagrafica codici alias Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
ERRAL	Codice risultato operazione codici alias	S/Cd
ERRAL\$	Messaggio operazione su codici alias	S/Cd
_NAL	Variabile array con numero dei codici alias	S/Cd
_AL	Variabili di struttura codici alias	S/Cd
esempi e casi dizionario		
<p>Dalla versione 2018E sono stati inseriti controlli per migliorare la gestione degli Alias: se il codice Alias (che è un numero) è presente sulle righe del documento di magazzino, durante la PUTMM viene fatto un test per verificare che questo ID Alias esista veramente nella relativa tabella Alias di quell'articolo di riga.</p> <p>Durante la GETMM, se la gestione degli Alias è abilitata (<u>non necessariamente utilizzata</u>), viene assegnato a Sprix quanto è registrato su disco; successivamente, in fase di controllo PUTMM del documento di magazzino, se la gestione degli Alias è abilitata e lo ID Alias è diverso da 0 questo viene ricercato negli Alias e se non esiste torna l'errore "ID codice alias non trovato"; se esiste, invece, viene confrontato il codice articolo dell'Alias con il codice articolo di riga e se non coincide ritorna l'errore "Codice alias non coincide con codice articolo di riga". In fase di PUTMM, quindi, l'ID Alias potrebbe non corrispondere (verificare in debug) ed eventualmente azzerare <code>_MMIDALIAS(I)=0</code>. Nei restanti casi viene eseguita la PUTMM.</p>		
<p>Esempio: Scrive un Codice Alias per un articolo, lo legge e gli modifica le caratteristiche e cancella tutti gli Alias di un altro articolo</p> <pre> AZZVARSYS 21 _ALART\$(1)="ARTICOLO 1" : _ALCOD\$(1)="ALIAS" _ALDES\$(1)="Caratteristiche Alias" : _ALPRO(1)=-1 PUTALIAS : IF _ERRAL<>0 THEN VIMSG 4, _ERRAL\$ GETALIAS "", "", -1 : IF _ERRAL<>0 THEN VIMSG 4, _ERRAL\$ FOR I=1 TO _NAL IF _ALCOD\$(I)="ALIAS" THEN GOTO CAMBIA NEXT CAMBIA: _ALDES\$(I)="Nuove Caratteristiche Alias" PUTALIAS : IF _ERRAL<>0 THEN VIMSG 4, _ERRAL\$ DELALIAS "ARTICOLO 2", "", -1: IF _ERRAL<>0 THEN VIMSG 4, _ERRAL\$: END </pre>		

CODICI ABBINATI (AB)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 22	Istruzione che azzerava variabili codici abbinati _AB	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "ABBIN"	Istruzione che azzerava variabili struttura _AB (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETABBIN	Par: <codart\$>,<codab\$>,<nprog>,<cat\$> Istruzione che legge anagrafica codici abbinati Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
PUTABBIN	Istruzione che scrive anagrafica codici abbinati L'istruzione non scrive le categorie abbinamenti, occorre che siano già codificate in tabella. Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELABBIN	Par: <codart\$>,<codab\$>,<nprog>,<cat\$> Istruzione che cancella anagrafica codici abbinati Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
_ERRAB	Codice risultato operazione codici abbinati	S/Cd
_ERRAB\$	Messaggio operazione codici abbinati	S/Cd
_NAB	Variabile array con numero dei codici abbinati	S/Cd
_AB	Variabili di struttura codici abbinati	S/Cd

esempi e casi dizionario

Esempio: scrive un Codice Abbinato per un articolo, legge e scrive a video tutti i codici Abbinati e infine cancella tutti quelli di un altro articolo

```
AZZVARSYS 22
_ABART$(1)="ARTICOLO 1" : _ABCOD$(1)="CODICEABBINATO"
_ABPRO(1)=-1 : _ABCAT$(1)="" : _ABCATD$(1)=""
PUTABBIN
IF _ERRAB<>0 THEN VIMSG 5, _ERRAB$
GETABBIN "", "", -1, ""
IF _ERRAB<>0 THEN VIMSG 5, _ERRAB$
FOR I=1 TO _NAB
LPRINT "CODART      : " ; : LPRINT _ABART$(I)
LPRINT "CODAB       : " ; : LPRINT _ABCOD$(I)
LPRINT "CATEGORIA    : " ; : LPRINT _ABCAT$(I)
LPRINT "DESCRIZ CATEG : " ; : LPRINT _ABCATD$(I)
LPRINT "PROG        : " ; : LPRINT _ABPRO(I)
LPRINT "DATA CREAZ  : " ; : LPRINT _ABDTC$(I)
LPRINT "DATA ULT MODIF : " ; : LPRINT _ABDTU$(I)
NEXT
DELABBIN "ARTICOLO 2", "", -1, "" : IF _ERRAB<>0 THEN VIMSG 5, _ERRAB$ : END
```

Esempio: ottimizzazione scrittura

Dato che la GET carica comunque tutti gli elementi, è utile una forma di codice che ottimizzi la scrittura alla sola aggiunta di abbinati, anziché la riscrittura di tutti gli abbinati con l'aggiunta del/i nuovo/i abbinati, in quanto l'operazione massiva di riscrittura potrebbe provocare un errore causato da "troppi blocchi sullo stesso semaforo".

```
AZZVARSYS 22
GETABBIN "ABB0", "", -1, ""
IFF _ERRAB = 0 OR _ERRAB = 2
  I = _NAB + 1 'nota: questa forma non ottimizza

  I = 1

  AZZVARSYS 22
  _ABPRO(I) = -1
  _ABCOD$(I) = "ABB1"
  _ABART$(I) = "ABB0"
  PUTABBIN ' (in questo modo la put scarica uno solo)

  IFF _ERRAB

  VIMSG 1, _ERRAB$

  ENDF
ENDF
```

AVANZAMENTO DI PRODUZIONE (APF)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 73	Istruzione che azzeri variabili codici abbinati _APF	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "APF"	Istruzione che azzeri variabili struttura _APF (come azzvarsys)	S/C/Cr/Cd Sm/Cm
AVANZAPF	Par: <bl>,<sbl>,<ripfnn>,<operazione\$ >,<qtapiu>,<datadoc\$> Istruzione che esegue l'avanzamento delle fasi di produzione	S
_ERRAPF	Codice risultato operazione AVANZAPF	S
_ERRAPF\$	Messaggio operazione AVANZAPF	S
_NLOTTIAPF	Elementi caricati per il lotto – solo input	S
_NAPF	Elementi caricati in struttura – solo output	S
_APF	Variabili di struttura AVANZAPF	

esempi e casi dizionario

Esempio AVANZAMENTO DI PRODUZIONE, scarico fino a fase 99:

```

BL=32
SBL=0
RIFPFN=1
OPERA$="S"
FASE=99
QTAPIU=0
DATA$=""
AZZCATVARSYS "APF"
AVANZAPF BL,SBL,RIFPFN,OPERA$,FASE,QTAPIU,DATA$
IF _ERRAPF<>0 THEN VIMSG 4, _ERRAPF$ :END
IF _NAPF>0
  R$="Sono stati creati " + STR$_NAPF,0,0)+" documenti di scarico:"
  FOR A=1 TO _NAPF
    R$=R$+"|- SL"+STR$_APFSER,0,0)+" /"+STR$_APFNUM(A),0,0)
  NEXT
  VIMSG 1, R$
ENDF
END

```

Esempio AVANZAMENTO DI PRODUZIONE, carico prodotto finito:

```

BL=32
SBL=0
RIFPFN=1
OPERA$="C"
FASE=0
QTAPIU=0
DATA$=""
AZZCATVARSYS "APF"
_APFIDLOT(1)=11 : _APFLQTA(1) = 800 ' _APFLQTATGL(1,1) = 800
_APFIDLOT(2)=21 : _APFLQTA(2) = 400 ' _APFLQTATGL(2,2) = 400
_NLOTTIAPF=2
AVANZAPF BL,SBL,RIFPFN,OPERA$,FASE,QTAPIU,DATA$
IF _ERRAPF<>0 THEN VIMSG 4, _ERRAPF$ :END
IF _NAPF>0
  R$="Sono stati creati " + STR$_NAPF,0,0)+" documenti di carico:"
  FOR A=1 TO _NAPF
    R$=R$+"|- CL"+STR$_APFSER,0,0)+" /"+STR$_APFNUM(A),0,0)
  NEXT
  VIMSG 1, R$
ENDF
END

```

Esempio AVANZAMENTO DI PRODUZIONE: scrittura BL e impegni da sprix, avanzamento bolla operazione impegno a fase 99 per scarico a fase 99 senza sviluppo distinta base.

```

'scrittura bolla
_OBNUM=66
_OBMAT=1
_OBMAG=1
_OBNSP = -1
_OBDAT$=sxsdt
_OLTPR$(1)="R"
_OLLAV(1)=32000
_OLART$(1)="PIZZA"
_OLPFF$(1)="PIZZA"
_OLRIF(1)=16000001
_OLQTF(1)=10
PUTBL
IF _ERRBL<>0 THEN VIMSG 1,_ERRBL$:END

'scrittura impegni
_ILNDO(1)=66
_ILNSD(1)=0
_ILNRB(1)=1
_ILFRS(1)=1
_ILMAG(1)=1
_ILAMP$(1)="ACQUA"
_ILTQT(1)=101
_ILQTA(1)=11.11
_ILNDO(2)=66
_ILNSD(2)=0
_ILNRB(2)=1
_ILFRS(2)=2
_ILMAG(2)=1
_ILAMP$(2)="A01"
_ILTQT(2)=101
_ILQTA(2)=22.22
_ILNDO(3)=66
_ILNSD(3)=0
_ILNRB(3)=1
_ILFRS(3)=3
_ILMAG(3)=1
_ILAMP$(3)="MERCE--1"

```

```
_ILTQT(3)=101
_ILQTA(3)=33.33
PUTIL
IF _ERRIL<>0 THEN VIMSG 1,_ERRIL$:END

'aggiornamento bolla per impegno a fase 99
GETBL 66,0
IF _ERRBL<>0 THEN VIMSG 1,_ERRBL$:END
_OLFSI(1)=99
PUTBL
IF _ERRBL<>0 THEN VIMSG 1,_ERRBL$:END

AVANZAPF 66,0,1,"S",99,0,""
IF _ERRAPP<>0 THEN VIMSG 1,_ERRAPP$
END
```

DISTINTA BASE PRIMARIA (TP)

Le funzioni e variabili di Produzione, non sono utilizzabili dai livelli di prodotto che non sono di Produzione		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 8	Istruzione che azzerava variabili _TP	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "DB"	Istruzione che azzerava variabili struttura _TP (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETDB	Par: <codart\$>,<numfase>. Istruzione che legge della DBP	S/Cr/Cd/Sh
PUTDB	Par: <numfase>. Istruzione che scrive dati di DBP	S/Cr/Cd/Sh
DELDB	Par: <codart\$>,<numfase>. Istruzione che cancella dati della DBP	S/Cr/Cd/Sh
_ERRDB	Codice risultato operazioni su DBP	S/Cd
ERRDB\$	Messaggio operazioni su DBP	S/Cd
_NCOMP	Numero di componenti in operazioni su DBP	S/Cd
_NFASI	Numero delle fasi in operazioni su DBP	S/Cd
SVILDB	Par: <codart\$>,[codpad\$],[codcli\$],[nrord],[qta1],[qta2],...,[qta32] Istruzione che sviluppa una Distinta Base Primaria	S/C/Cr/Cd/Sh
ERRSVILDB\$	Messaggio operazione con SVILDB	S/C/Cd
_NSVILDB	Numero componenti sviluppati da SVILDB	S/C/Cd
_MSGSVILDB	Quanti messaggi da SVILDB	S/C/Cd
_MSGSVILDB\$()	Par: (). Descrizione messaggi da SVILDB	S/C/Cd
_TP	Variabili di struttura files articoli DBP	S/C/Cd
esempi e casi dizionario		
<p>Notare: a seguito di implementazioni di funzionalità possono avvenire cambiamenti di sequenza degli eventi, ad esempio, come è avvenuto fra gli eventi delle etichette ON_OUT_MAEVID e ON_OUT_MACTTIP.</p> <p>Il parametro di Apertura Automatica Finestra Condizioni (parametro di Produzione presente in Anagrafica Azienda - [F4] - Parametri Produzione - Apertura automatica finestra condizioni) influenza il comportamento a video.</p> <p>La finestra dei componenti della DBP è cambiata, in quanto l'apertura automatica della finestra condizioni è stata spostata al termine di riga (cioè non più dopo la colonna Evi, bensì dopo la colonna Ctr) e per eseguire l'avanzamento alla riga successiva, dopo la conferma F10 di quella finestra viene forzata in background una PUTTAST "#501". Se non si tiene conto di questo, l'evento di interesse all'uscita di quella finestra viene ignorato e saltano tutte le routine di comandi che lo riguardassero.</p>		
<p>Esempio: Dato l'articolo ARTDBP con Fase 1 a due Componenti (A e B) e Fase 2 a due Componenti (C e D), legge la DBP e ne cancella l'ultima Fase per ricrearla con due altri componenti (E e F).</p> <pre> FOR FAS=1 TO _NFASI GETDB "ARTDBP",FAS : IF _ERRDB<>0 THEN VIMSG 4, _ERRDB\$ FOR IND=1 TO _NCOMP PRINT "Fase "+STR\$(FAS,0,0)+" Compo. "+STR\$(IND,0,0)+" :"+_TPART\$(IND) NEXT NEXT VIMSG 1, "Cancella e sostituisce l'ultima Fase con componenti E - F" DELDB "ARTDBP", _NFASI : IF _ERRDB<>0 THEN VIMSG 5, _ERRDB\$ _TPDES\$="NUOVA FASE": _TPULT\$="ULT. DESCRIZ.": _TPDOC\$="C": _NCOMP=2 _TPIND\$(1)="5": _TPART\$(1)="COMP E": _TPQTA\$(1)="5": _TPUNM\$(1)="1" _TPIND\$(2)="6": _TPART\$(2)="COMP F": _TPQTA\$(2)="5": _TPUNM\$(2)="1" PUTDB _NFASI : IF _ERRDB<>0 THEN VIMSG 4, _ERRDB\$: END </pre>		
<p>Esempio: registrazione di una fase e relativi componenti di una distinta base</p> <pre> _TPCOD\$="DBP" 'cod. articolo con distinta base _TPCTR\$="S" 'controllo esistenza disponibilita' _TPUSA\$="N" 'usa DBVCBVO _TPUMP\$="P" 'tipo unita' di misura utilizzato _TPDEC=2 'numero di decimali _TPKOE=1 'coefficiente 'DATI FASE DISTINTA BASE _TPDES\$="DESCRIZIONE_FASE" 'descrizi one della fase _TPULT\$="ULTER DESCRIZ" 'ulteriore descriz one della fase _TPDOC\$="C" 'documenti in stampa _TPIND\$(1)="INDICE" 'indice _TPART\$(1)="ART3" 'cod. componente _TPQTA\$(1)="3" 'quantita' _TPUNM\$(1)="1" 'unita' di mi ura utilizzata PUTDB 3 'scrittura IF _ERRDB<>0 THEN VIMSG 4, _ERRDB\$ 'controllo errore END </pre>		
<p>Esempio: sviluppo ARTDBP per cliente "201.00001", q.ta=10*3,5 (il componente di DBP con condizione cliente viene sviluppato, diversamente no)</p> <pre> SVILDB "ARTDBP","", "201.00001",0,10,3.5: IF _NSVILDB=0 THEN GOTO ERR1 FOR I=1 TO _NSVILDB IF osval(I)=0 THEN GOTO PROSSIMA1 PRINT " Fase: "+STR\$(osfas(I),0,0)+" , Comp.: "+osart(I);; PRINT " , Q.ta: "+STR\$(osqta(I),0,0) '--> eventuali righe di calcolo PROSSIMA1: NEXT PRINT "" VIMSG 1, "termine sviluppo DBP" ERR1: IF _NSVILDB=0 THEN VIMSG 1, _ERRSVILDB\$ </pre>		
<p>Sviluppo ARTDBPTAG (a Taglie) per cliente "201.00001" con aggancio a DBVO della riga ordine [ocnri()] per una q.tà di 75 (10 per la 1^</p>		

Tg. 20 per la2^ Tg. E 45 per la 4^ Tg.). L'OC su cui è inserita una 'DBP a taglie (che preveda la possibilità di DBV) deve ricevere la modifica di un componente. Lo sviluppo moltiplica le quantità per quella di prodotto finito su riga.

```
SVILDB "ARTDBPTAG", "", "201.00001", 631, 0, 10, 20, 0, 45
IF _NSVILDB=0 THEN GOTO ERR2
FOR I=1 TO _NSVILDB
IF osval(I)=0 THEN GOTO PROSSIMA2
PRINT " Fase: "+STR$(osfas(I),0,0)+" , Comp.: "+osart(I);
PRINT " , Q.tà: "+STR$(osqta(I),0,0)
'--> eventuali righe di calcolo
PROSSIMA2: NEXT
PRINT ""
VIMSG 1, "termine sviluppo DBVO su articolo con DBP a taglie"
ERR2: IF _NSVILDB=0 THEN VIMSG 1, _ERRSVILDB$
```

PUTDB: perde il collegamento alla tabella abbinamenti o presenta unità di misura errata

La GETDB effettua l'agancio della tabella abbinamenti, ma se nella PUTDB non si specifica la valorizzazione di _TPAGG\$=" " si perde quel riferimento (viene azzerata). L'unità di misura è errata perché il campo _TPUNM\$() non deve contenere l'unità di misura (pz) ma 1 se viene utilizzata l'unità di misura primaria - 2 se viene utilizzata l'unità di misura secondaria.

SVILDB: sviluppo articoli con quantità a zero

Lo sviluppo della DBP non consente lo sviluppo di articoli con quantità inesistente in immissione nel documento. La stampa invece, ne consente l'evidenziazione poiché rileva quanto è stato immesso nella DBP e non nel documento (verificabile nell'apposita funzione di stampa distinte basi).

Caso: variabile struttura _TPUNM\$() unità di misura componente

A differenza della variabile _TPUNP\$(tipo unità di misura componente) che accetta valori di P o S, la variabile _TPUNM\$() che rappresenta l'unità di misura scelta per il componente accetta valori 1 o 2 (sempre in formato stringa).

Messaggio: Art. non valido (Svil.DB:)

Il messaggio è un warning e nasce da un controllo preventivo nella codifica e/o aggiornamento di articoli (mp, sml, pf) tipici della PRODUZIONE (titolari struttura, cartella, ...) e quindi suscettibili di DB e, soprattutto, di SVILDB. Il programma controlla in anticipo se ciò che si sta registrando è presente in anagrafica ed informa l'utente se ciò che egli ha scritto è idoneo per SVILDB.

SVILDB è utilizzabile in Shaker

Le variabili di dizionario sono lette col metodo .DIZ a cui viene passato il nome della variabile di dizionario il relativo dizionario appartiene alla tabella "os"

Caso: modifica descrizione fase DBP

Con il rilascio della versione 2022I e relativa introduzione della tabella "gd", FASI DI LAVORAZIONE, è stato modificato il comportamento della variabile _TPDES\$ che diventa dipendente alla valorizzazione del contenuto della variabile _TPCICLO. Nella nuova versione la scrittura di un nuovo record in tabella viene eseguita, in maniera del tutto trasparente all'utente, quando il dato presente in _TPDES\$ non esiste già e _TPCICLO vale 0. Riguardo l'aggiornamento della descrizione della fase della DBP quando la descrizione esiste in tabella, ci sono due modalità operative possibili: 1) azzeramento della variabile _TPCICLO e valorizzazione di _TPDES\$ con la descrizione della fase; 2) si ignora il contenuto di _TPDES\$ e si valorizza la sola _TPCICLO con il nuovo valore. Nota bene: per garantire la retrocompatibilità con le personalizzazioni esistenti rimane l'obbligo di valorizzare in ogni caso il contenuto di _TPDES\$.

DOCUMENTI DI MAGAZZINO (MM)

Le seguenti funzionalità non sono disponibili per i documenti NOTula e PARcella di Parcellazione Studio, nemmeno in forma di lettura ciclotabellare; sono invece consentite le scritture di attività in agenda di lavoro (AGP), da parcellare tramite la gestione riepilogativa		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 4	Istruzione che azzerà variabili documenti di magazzino _MM	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "MM"	Istruzione che azzerà variabili struttura _MM (come azzvarsys)	S/C/Cr/Cd Sm/Cm
AZZVARSYS 15	Istruzione che azzerà variabili lotti _LA	S/C/Cr/Cd Sm/Cm/Sh
LEGGIDESC	Par: <par>. Istruzione che nel ciclotabellare attiva la lettura delle righe descrittive articolo ??tds() e ??dsc() nel corpo del documento di magazzino. L'attivazione della lettura è valida sia nel ciclotabellare che nelle operazioni di mirroring. Si ricorda che le righe di tipo testo sono qui intese come quelle contenenti il punto e la virgola che possono essere inserite da articoli di tipo testo variabile. Sono escluse le righe di tipo testo fisso.	S/C/Cr/Cd/Sh
GETMM	Par: <doc\$>. Istruzione che legge documenti di magazzino.	S/Cr/Cd/Sm
GETMM_EXT	Par: <doc\$>[,conto\$]. Istruzione di lettura estesa dei documenti di magazzino	S/Cr/Cd/Sm/Sh
GETATTMM	Istruzione che legge documenti di magazzino a video	C
GETATTRIGMM	Calcolo dei totali di riga	C
GETATTTOTMM	Calcolo dei totali documento	C
GETNRIDOCVID()	Funzione di lettura numero di riga a video del corpo documento di magazzino	S/C/Cr/Cd/Sm
DOCMODCONT()	Funzione per conoscere lo stato di aggiornamento contabile del documento in revisione	S/C/Cr/Cd/Sm
SETPARSCRDOCNM	Par: <str\$> Istruzione che forza la modifica di documento il cui Xml è già inviato allo SDI	S/Cr/Cd/Sh
PUTMM	Par: <scprn>. Istruzione che scrive documento di magazzino. Consultare le logiche in calce a questa tabella	S/Cr/Cd/Sm/Sh
RIGMM	Par:<numriga>,<modo>,<numdecart>. Istruzione che calcola Valori di Riga del documento di magazzino Richiede requisiti equivalenti a quelli della PUTMM Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
TOTMM	Istruzione che calcola i Totali del documento di magazzino Richiede requisiti equivalenti a quelli della PUTMM Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELMM	Par: <doc\$>. Istruzione che cancella documenti di magazzino. Non cancella la Primanota (occorre utilizzare DELPN), vedere le logiche di Primanota. Per aprire la finestra di selezione di documenti uguali ma con codice cliente/fornitore diverso occorre utilizzare la DELMM_EXT	S/Cr/Cd/Sm
DELMM_EXT	Par: <doc\$>[,conto\$][,cancpn]] Istruzione di cancellazione estesa documenti di magazzino	S/Cr/Cd/Sm/Sh
_ERRMM	Codice risultato operazione documenti di magazzino	S/C/Cd
_ERRMM\$	Messaggio operazioni su documenti di magazzino	S/C/Cd
_NMM	Numero Righe documenti di magazzino	S/C/Cd
_NTMM	Numero Testate documenti di magazzino	S/C/Cd
_MM	Variabili di struttura documenti di magazzino	S/C/Cd
_MML	Variabili si struttura lotti matricole	S/C/Cd
_LA	Variabili LOTTI (evento ON_LISTALOTTI)	C/Cd
_LISTADOCMM\$()	Par: (). Vettore lista (evento ON_LISTADOCMM)	C/Cd
SETPARCREAZMATR	Par: <str\$>. Istruzione per la gestione della creazione della Matrice (modifica il default presente nei parametri dell'anagrafica azienda). Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
SETPARSOSTMATR	Par: <str\$>. Istruzione per la gestione della creazione della Matrice (modifica il default presente nei parametri dell'anagrafica azienda) Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
ON_LISTADOCMM:	Evento elenco documenti (vedere _LISTADOCMM\$)	S/Cr/Cd/Sh
ON_LISTALOTTI:	Evento in lista lotti (vedere variabili _LA). Nota: non usare interazioni a I/O video; il 20° elemento del dizionario lamov() riporta se il lotto ha subito carichi multipli. Es: lamov(20)(1)= se il lotto 1 ha carichi multipli da più documenti. Es: FOR I=1 TO _LANUM : I\$=STR\$(_LIDL(I),0,0); R\$=STR\$(_LARIG,0,0); N\$=STR\$(_LANUM,0,0) : NEXT	C/Cd
ON_MSGARTMM:	Evento prima del dettaglio articoli in menu MD	C/Cd
ON_PUTMM:	Nel menu ME (emissione differita documenti) questo evento c'è solo con l' effettiva scrittura , esempio la trasformazione, pertanto, non viene processato in fase di ristampa! ovvero con " Emissione di prova = S " (è attivo ON_STARTMM)	C/Cd
ON_DELMM:	Nel menu MD questo evento non avvia l'evento ON_DELPN	C/Cd
ON_STP_XML:	Evento per la modifica del file XML della fattura elettronica. es.	C/Cd
_FTE	Variabili utilizzate dall'evento ON_STP_XML es. Nota: non è possibile intervenire sullo stesso Tag che riporta più informazioni (esempio: lotti spezzati sulla stessa riga articolo), in tal caso occorre gestire l' intero documento con le istruzioni generali XML. Per ottenere l'handle del documento in considerazione al fine di eseguirne le opportune modifiche con le istruzioni generali XML, avvalersi della variabile _FTE_DOCXML che consente l'accesso da collage al file _XML della FTE.	C/Cd
STORD	Par: <stampante>,<serie>,<numero>,<anno1>,<anno2> Istruzione che stampa la Storia Ordine. Non utilizzabile da livello di prodotto "Junior" (prodotto obsoleto non più distribuito)	S/Cr/Cd/Sh
_ERRSTORD	Codice risultato operazione Storia Ordine	S/Cd
_ERRSTORD\$	Messaggio operazione Storia Ordine	S/Cd

LOGICHE DELLA SCRITTURA (PUTMM)

Attenzione:

- La PUTMM con parametro che esegue anche la stampa, richiede interazione I/O, pertanto non è schedulabile in operazioni pianificate di tipo batch oppure con mxrsbatch.exe. Per la stampa, avvalersi di apposita APRISTAMP
- L'istruzione scrive in archivi MYDB attraverso _MMMYDBT(T) e _MMMYDBR(R) (vedere le note in calce alla tabella dei comandi MyDB); MyDB non è gestito dal modulo Produzione e gli eventuali archivi possono essere riportati nei documenti CL, SL, DL solo attraverso la procedura standard dal modulo Magazzino;
- Non scrive movimenti di analitica del Controllo di Gestione (serve PUTCDG);
- Genera documenti CL/SL (articoli con DBA) solo con livello Produzione e non consentita nel Livello di sviluppo (F2), non consentita in Passcom o Live in Server Farm.
- La gestione dei documenti di produzione SL, CL, DL e DBA non sono disponibili nel livello Solution, ma solo in Aziendale e Produzione. I documenti di rettifica dei progressivi di magazzino in Solution sono: 'IC' = Rettifica di Carico e 'IS' = Rettifica di Scarico
- La stampa non è consentita ai livelli Contabile, Junior, Solution senza MDS;
- In materia di Fatturazione Elettronica consultare le note della tabella DOCUMENTI DOCUVISION (DV)

Sprix consente di registrare e stampare la Fattura Xml e presentarle nel cruscotto del documentale per la PA ed il B2B (vedere menu AVF di Docuvision), ma la procedura di invio deve proseguire con le funzioni standard. Shaker, invece, non può stampare perché consiste in una componente esterna che dialoga con Passepartout in modo silente senza interazioni I/O pertanto non può "stampare", ma può predisporre la registrazione da stampare successivamente.

La scrittura necessita dei dati obbligatori richiesti dalla versione del gestionale in uso. **La revisione di un documento fiscale elettronico potrebbe richiedere SETPARSCRDOCNM, ovvero l'istruzione che, richiamata prima della PUTMM, consente di forzare la scrittura di documento non modificabile se l'elettronico è già stato trasmesso al sistema di interscambio (SDI).**

L'istruzione non è consentita per la Notula. L'istruzione non scrive movimenti di analitica del Controllo di Gestione (utilizzare la PUTCDG). La generazione dei documenti SL/CL per articoli con DBA è consentita solo con livello Produzione.

La primanota è registrata se prevista dal documento, che sia nuovo e diverso dalla FF; se il destinatario del documento gestisce la fattura elettronica o la fattura PA (Pubblica Amministrazione), il numero di serie _MMSER è prelevato in automatico.

Sprix consente di registrare e stampare la Fattura Xml e rappresentarle nel cruscotto del documentale per la PA ed il B2B (vedere menu AVF di Docuvision), ma la procedura di invio deve poi proseguire tramite le funzioni standard del Gestionale. Shaker, invece, non può stampare perché consiste in una componente esterna che dialoga con Passepartout in modo silente senza interazioni I/O pertanto non può "stampare", ma può predisporre la registrazione da stampare successivamente.

Nella scrittura dell'ordine, la creazione o sostituzione della matrice si controlla con istruzioni SETPARCREAZMATR e SETPARSOSTMATR.

Sono consentite fino a circa 65.000 righe di articoli e descrizioni sul documento, che sia di tipo sospeso (preventivo/ordine) oppure di tipo movimento (bolla/fattura); precisamente 65500 per i documenti sospesi e 65535 per i movimenti. Le prestazioni dipendono dalla memoria del sistema e sono inversamente proporzionali alla quantità di righe.

Il numero massimo di righe (R) deve comunque rientrare nei limiti dell'array numerico (max 131040) e dipende dal tipo di installazione in uso ed è inversamente proporzionale alla quantità di informazioni aggiuntive alla riga: videate personalizzate (limite a 10080), taglie (limite 8190), imballi, lotti, modifiche alle descrizioni degli articoli, variazioni di aliquota iva, ecc.. Incidono anche le trasformazioni/fusioni e le aggiunte/cancellazioni di righe, che diventano maggiormente più lente rispetto ai casi di quantitativi inferiori di righe sul documento, e ancor più lente in caso sia collegato MexalDB in tempo reale.

In ogni caso, effettuare i test con quantità crescenti (1000, 2000, ecc) in varie condizioni di ulteriori dati. L'array (R) riferito alle righe del documento contiene inoltre i dati di testata dei (T) possibili documenti d'origine (max. 250) che possono comporre il documento. **Nota:** nella composizione del documento, la valorizzazione delle variabili di testata (quelle riferite a _NTMM) devono essere valorizzate in modo uniforme e omogeneo, altrimenti non sono stampate (vedere approfondimenti sull'operatività come riportato dal manuale utente). In ogni caso, Emissione Revisione Documenti può visualizzare fino a 8191 righe.

NOTARE BENE

Prestare attenzione al tipo di documento si sta elaborando, poiché la modifica della chiave del documento comporta una nuova emissione del documento e non una revisione dello stesso. Per i documenti che non comportano movimentazioni (ordini, preventivi, matrici, ecc..) la chiave è costituita da Sigla+Serie+Numero, mentre per i documenti che comportano movimentazioni e/o registrazioni in primanota (fatture, bolle, documenti di carico e scarico, ecc) la chiave è: Sigla+Serie+Numero+Data+Conto.

La **trasformazione** non può avvenire come nel Gestionale (in cui basta cambiare la _MMSIG\$ e confermare), perché occorre leggere il documento, modificarne gli estremi e registrarlo "ex-novo", per poi cancellare l'originario documento.

L'operazione è corretta se le variabili sono impostate con gli accorgimenti richiesti dai documenti e in base alle condizioni di trasformazione (operazioni invece automatiche nello standard), che vengono controllate corrette nella riorganizzazione archivi con messaggi di tipo "Doc. di scarico errato, corretto"; i disallineamenti di archivi sono provocati da grossolane trasformazioni di variabili lette, ma non trattate in scrittura e quindi non adatte a nuovi documenti.

Se i **riferimenti** del documento non vengono aggiornati, si riferirebbero al documento originario (da cancellare); **Sprix non aggiorna i riferimenti**, quindi occorre verificare il riferimento per ciascuna riga del documento trasformato: gli array della testata sono _MM?SI\$(), _MM?SE(), _MM?NU(), _MM?DA\$() ed il collegamento di questi riferimenti alle righe è tramite l'array _MMNTE() cioè il riferimento della testata che indica il documento di origine di ogni riga.

I riferimenti al documento di origine _MM?SI\$() sono letti solo se nell'array _MMNTE() è indicato l'indice corrispondente ai riferimenti alle testate in modo progressivo crescente e continuo, dato che c'è un collegamento diretto fra le righe di ciascun documento originario e i riferimenti (riportati in testata) nel documento trasformato.

L'indice deve quindi essere indicato in modo progressivo crescente e contiguo senza vuoti, poiché c'è un collegamento fra le righe di ogni documento originario (ordine) ed i riferimenti (riportati in testata) nel documento prodotto (bolla). Esempio OC1 (ha 10 righe) OC2 (ha 3 righe) OC3 (ha 1 riga) e gli array(1-200) di testata del documento BC saranno riferiti all'indice 1, 2, e 3. Ma ciò servirà solo se indichiamo anche che la prima serie di 10 righe è riferita all'indice 1, la seconda serie di 3 righe (dall'undicesima alla quattordicesima) all'indice 2 e l'ultima riga (la quindicesima) all'indice 3, in modo progressivo crescente e senza vuoti.

Nella **lettura e trasformazione di più documenti**, le variabili non subiscono lo stesso trattamento: per alcune possono continuare a valere i valori letti, ma per altre possono avvenire azzeramenti di default prima della scrittura, esempio la _MMVAS() se azzerata, causa l'assenza delle spese di trasporto in primanota, pertanto è utile separare la lettura dalla scrittura salvando i dati su variabili d'appoggio.

Non è consentita la fusione di più documenti sospesi in un unico dello stesso tipo (è rilasciato il messaggio "per i docum. Prev/ord/matr non è ammessa più di una testata"): occorre trasformarli in uno di movimento (**azzerando** _MMMSI\$(I) per _NTMM righe per impostare correttamente il documento di scarico (non trovato con il documento origine e con il seguente messaggio "Numero di riga univoco utilizzato per la riga n (riga n-1)").

Revisionando un CL/SL non intestato, l'aggiunta del conto cliente/fornitore modifica la chiave del documento, pertanto non risulta essere più una revisione, ma

una **nuova** immissione e ciò costringe alla DELMM del documento originario. Se non viene presa in considerazione la corretta gestione dei riferimenti, si possono creare dei disallineamenti rispetto alla parametrizzazione dei documenti (v. menu MZD+[F5]) relativamente all'aggiornamento dell'ultima data di carico/scarico, in quanto, dall'originaria impostazione a "S" (cioè _MMAGDTMAG\$="S", _MMAGDTAMA\$="S", mmpa()="S" e mmpd()="S") può provocarsi una modifica di detti parametri ad "N" (i documenti vengono marcati "N", cioè senza aggiornamento delle date); di conseguenza, dato che la riorganizzazione cancella e ricostruisce i progressivi in base alle condizioni che trova sul documento, se il documento è stato creato marcato con flag = "N" (data non aggiornabile), la riorganizzazione non la aggiorna, azzerando la data nei progressivi di anagrafica articolo (la data non è più visibile), conformemente a quanto il documento SL riportava (cioè "N"). Per rimediare a simili situazioni il menu [ZVMP] può intervenire solo sugli ultimi progressivi dei due anni aperti; per i precedenti anni occorre una rinnovare la registrazione (duplicando o importando il contenuto del documento, poi da cancellare, per poi modificare il numeratore del nuovo documento). Il suddetto comportamento sulla modifica dei flag di aggiornamento, può essere causato dal non azzeramento dei riferimenti, come invece non accadrebbe se fossero azzerati. Esempio: SL123 di un articolo solo scaricato dalla DBA di una BC, a cui aggiungere il conto cliente (e che quindi modifica la chiave del documento).

```
GETMM "SL123" : IF _ERRMM<>0 THEN VIMSG 1, _ERRMM$
_MMCLI$="501.00123"
_MMNTE(1)=0
_MMMSI$(1)=" " (: _MMMSE(1)=0 : _MMMN(1)=0 : _MMMDA$(1)="")
_MMOSI$(1)=" " (: _MMOSE(1)=0 : _MMONU(1)=0 : _MMODA$(1)="")
PUTMM 0
DELMM "SL123"
```

Aggiornamento Primanota e Scadenziario

La **primanota** viene registrata **se prevista dal documento** e che questo sia **nuovo** e **diverso** dalla Fattura Fornitore. Fuori da queste condizioni la modifica al documento di magazzino **non aggiorna la primanota né tanto meno lo scadenziario** (v. logiche PUTPN). Possono anche essere previsti casi in cui alcuni valori non sono gestiti, per esempio nell'accorpamento di più bolle in una fattura le spese di trasporto potrebbero non essere contabilizzate in Primanota, pertanto, ciò richiederebbe di richiamare il documento da Emissione Revisione Documenti e salvandolo confermando l'aggiornamento della Primanota.

Un modo che potrebbe essere consentito per stabilire da magazzino una determinata data relativa alla scadenza del pagamento (es. Rimessa) di un nuovo documento è attraverso la procedura Collage, in modo da forzare il campo **decorrenza** (MDDDEC) sulla base di un determinato calcolo di giorni parametrizzato dall'utente. Ad esempio, per portare le scadenze a fine mese.

La lettura della registrazione di Primanota da modulo Emissione Revisione Documenti di Magazzino è consentita sia da Sprix (getpn eseguita da shellsprx) che da Collage (getatpn al evento on_putpn durante la registrazione del documento di magazzino tramite collage configurato su menu MD che CR). La modifica della primanota tramite sprix sarebbe con delpn+putpn ma **si perderebbe il collegamento di _MM sulla _PN** (il flag presente nei Totali del documento di magazzino che consente l'aggiornamento della primanota da magazzino); l'unico modo per conservare questo collegamento è simulare l'utente tramite Collage che operi in Revisione Primanota tramite la finestra Shift+F5 (Documento Precedente). Esempio di Collage su menu MD CR.

```
ON_STARTMM: _INTER=1 : DISVID 0 : END
ON_IN_MDDDOC:
IF F2=1 AND F1=0 THEN VIMSG 1, STR$( _PNPRN,0,0) : F3=1
IF F3=1 THEN END
IF F2=0 AND F1=0 THEN END ELSE PUTTAST "#617#617" : DISVID 1 : END
ON_ENDMM: F1=1 : F2=0 : F3=0 : END
ON_STARTPN: DISVID 1 : GETATTPN : PUTTAST "#505" : F1=0 : F2=1 : F3=0 : DISVID 0 : END
```

La variabile _ERRMM\$ non controlla la validità di tutte le variabili _MM? durante la scrittura PUTMM

Sprix non ha gli stessi automatismi e controlli del gestionale, pertanto vanno tutti considerati o ricostruiti. La maggiore libertà di Sprix richiede un accurato controllo nel trattamento dei dati. Come la maggior parte di variabili, ad esempio, la valuta non è obbligatoria per registrare un documento (infatti non è un requisito minimo) e l'inizializzazione della _MMEST con un valore inesistente non è (e non deve essere) di ostacolo alla registrazione quando sono comunque soddisfatti i requisiti minimi obbligatori per registrare, poiché non è possibile prevedere i motivi per cui lo sviluppatore decide di operare in un certo modo (volontario o accidentale) consentito nonostante non sia completamente corretto. La stessa riorganizzazione, infatti, avverte che comunque la registrazione è stata eseguita con valuta inesistente, ma non elimina il documento e quindi lo accetta, poiché accetta le condizioni pervenute dallo sviluppatore nonostante che queste non siano tutte corrette. Quanto detto è paragonabile ad una semplice GETMM di BF e contestuale PUTMM di FF senza la gestione (il controllo) di tutte le variabili in esse contenute. L'operazione è consentita, ma non è una trasformazione di documenti che, peraltro, sono anche sostanzialmente molto diversi.

La trasformazione automatica del documento è possibile?

Considerando le indicazioni riportate nel dettaglio della PUTMM è evidente che la trasformazione dei documenti non è un'operazione semplice, poiché ricorrono importanti aspetti da considerare; non basta leggere il documento (GETMM) per cambiargli la sigla _MMSIG\$ (esempio da OC a BC) e poi eseguirne una PUTMM, poiché il risultato che si otterrebbe non è la trasformazione del documento OC, ma la generazione di una BC con le stesse sembianze dell'OC letto. Nella procedura standard, ogni qual volta che si esegue una trasformazione di un documento da "Emissione/Revisione documenti", il Gestionale aggiorna automaticamente tutti gli elementi ed i riferimenti che collegano i documenti fra di loro, pertanto, replicare questo in Sprix significa dovere ricostruire tutte le logiche ed i meccanismi (per esempio, l'evadibilità o meno di una riga da spezzare). Se non si desidera gestire le relazioni si potrebbe creare una sorta di automatismo tramite Collage che carica i documenti OC evadibili di riferimento e lasci all'utente la possibilità, eventualmente, di potere modificare il documento prima della trasformazione finale che, comunque, resta in carico al Gestionale che controlla se sussistono i requisiti per la trasformazione e conseguente evadibilità del documento.

Per quanto riguarda l'evadibilità ordine cliente, la possibilità di evadere l'ordine cliente non è consentita iniziando le variabili di struttura _MMEV\$ e/o _MMEVD\$ in quanto queste, stabiliscono regole con cui il documento deve essere poi gestito una volta che viene aperto da gestionale. Quindi, una volta stabilite, il documento consentirà trasformazioni se le righe sono interamente evase e/o consentirà l'aggiunta di nuove righe direttamente evadibili. Chi "comanda" quindi, sono le righe del documento che devono essere gestite impostando la variabile _MMORD\$

Gestione della DBA (Distinta Base Automatica)

Non esistono comandi o strutture di gestione della DBA, eccetto la lettura da ciclotabellare (tabella "ta"), per esempio è consentito leggere i componenti se l'articolo ART\$ ha una _ARTDBA\$, es. COMP\$=taart(ARTDBA\$,I).

Relativamente a GETMM e PUTMM la gestione della DBA o DBVO collegata all'uso di _MMNRECRIGA(1-N).

Solo le funzioni standard del gestionale rintracciano i documenti SL/CL dal documento BC/FT e questo non è consentito in sprix (perché BC/FT possono originare più documenti).

Se il terminale è di tipo **Produzione**: è consentito procedere alla rintracciabilità da SL/CL attraverso la struttura _MMRINTR???, inoltre, l'esecuzione della PUTMM se impostata correttamente la variabile _MMNRECRIGA(1-N) genera gli SL e CL per articoli con DBA.

Se il terminale è di tipo **Aziendale** o **Solution**: per risalire ai documenti SL/CL generati dalla DBA è possibile interrogare _MMNOT\$ in quanto riporta la traccia del documento origine.

Come riconoscere se il documento è nuovo in immissione oppure vecchio in revisione.

Data l'assenza di specifiche variabili occorrono statagemmi da scegliere in base alle circostanze (che vanno dal tipo di documento al tipo di trattamento che lo riguarda e che possano rilasciare "indizi" negli archivi, fino all'operatività eseguita con Sprix oppure Collage. Indichiamo alcuni suggerimenti.

Per controllare se il documento di magazzino è in modalità inserimento o in variazione, si può testare una variabile d'appoggio (se vuota si è in inserimento) basata sul valore prelevato dal campo ricerca (in variazione documento questo risulta necessariamente valorizzato).

In Collage è possibile stabilire che il documento è in immissione quando è vuoto (_SINP\$="") il campo di ricerca documento MDRIDOC, diversamente il documento è in revisione; nella modalità di gestione con la "finestra elenco", questa possibilità è data dall'evento ON_LISTADOCMM dove se il vettore _LISTADOCMM\$() è vuoto significa immissione, diversamente il documento è in revisione (notare: l'evento e l'array non sono nati per questa funzione, l'evento equivale alla tradizionale uscita del vecchio campo di ricerca documenti e non si attiva su documento nuovo, pertanto l'array va azzerato manualmente all'uscita della revisione).

Un'altra possibilità consiste nella lettura del documento e testare IF _MMNUM=0 THEN VAR\$="documento nuovo". Esempio

```
ON_STARTMM: GETATMM : END
ON_PUTMM: IF _MMNUM<>0 THEN D$="revisione" ELSE D$="nuovo": END
```

In Sprix, invece, altre modalità possono consistere nel verificare le variabili di struttura relative alle proprietà del documento (per esempio l'utente di ultima modifica `_MMUTEMOD$`, oppure data creazione `_MMDATCRE$`, ecc..), altre ancora possono consistere nel verificare se i documenti hanno originato una scrittura di primanota ciclando il alcuni casi di dizionario "mmprd" e "mmprr", oppure hanno una matrice collegata o un altro documento di riferimento.

Collegamento ordine cliente e fornitore (il collegamento avviene dall'OC e non dall'OF).

GETMM "OC1" : `_MMORD$(1)="F" : _MMSOF(1)=serie OF : _MMNOF(1)=numero OF : _MMNCRIGAO(1)=_MMNCRERIGA()` della riga OF da collegare : PUTMM 2
 Attenzione: l'operazione è consentita dal suddetto esempio, ma richiede comunque attenta valutazione sugli effetti provocati ai collegamenti nel gestionale; per esempio, se si sbaglia l'assegnazione `_MMNCRIGAO(1) = _MMNCRERIGA()`, cioè la riga OF non è quella giusta, il documento OF risulta collegato ma in realtà non lo è, pertanto, non è più possibile cancellare i documenti (ne da sprix ne da procedura) e gli eventuali successivi interventi sul OF collegato sono vincolati, e l'unica soluzione è la riorganizzazione degli archivi.. Si consiglia, inoltre, di consultare la richiesta 13959.

Righe OC e OF fra esse collegate

Il riconoscimento di righe d'ordine fra esse collegate non avviene dall'ordine fornitore all'ordine cliente, ma si esegue dalla riga d'ordine cliente al documento ordine fornitore (vedere dizionario ocndo()).

Utilizzo dei colli

Indipendentemente dal fatto che in "Anagrafica Articoli" un articolo abbia, o meno, valorizzato il campo confezione, se si inserisce in "Emissione/revisione documenti" una quantità di articolo nella forma V*F (3*6 ad esempio), i progressivi di magazzino e/o i progressivi degli ordinativi vengono maggiorati della quantità 3*6=18. La stessa cosa avviene in Sprix, dal momento che `_MMCOL(1-200)` identifica il numero di colli e che `_MMQTA(1-200)` identifica la quantità (in questo caso la quantità per collo).

Descrizione aggiuntiva nel documento di magazzino

La dimensione in quantità di caratteri del codice articolo (descrizione e descrizione aggiuntiva), presenta aspetti differenti in base al contesto: nella procedura standard l'estensione è interamente visibile, mentre in Sprix le istruzioni di lettura (GETMM e GETAR) e cicli tabellari ("mm" e "ar") richiedono considerazioni, in quanto si distingue fra ciò che riguarda i primi 40 caratteri ed i successivi 20.

La dimensione della descrizione principale più la dimensione del codice principale, infatti, è 40 caratteri, e fra queste c'è una interdipendenza ad intervalli di 4 caratteri: la lunghezza codice principale può essere di 8, 12, 16, 20, 24, 28, fino a 32 caratteri e la lunghezza della descrizione articolo può essere, a sua volta, di 32, 28, 24, 20, 16, 12, fino a 8 caratteri in base alle dimensioni del codice principale; inoltre, è consentita una descrizione aggiuntiva di 20 caratteri in apposito campo fino a identificare l'articolo con 60 caratteri. Spesso, però, si è portati a pensare che quest'ultima sia parte integrante della descrizione principale, ma in realtà è un elemento indipendente e trattato da Sprix in modo differente rispetto a quanto accade nel gestionale standard. Relativamente all'anagrafica articolo (GETAR, ciclotabellare "ar") consente l'intera visualizzazione: sono previste le variabili strutture archivi `[_ARCOD$ e _ARDESS$ fino a 40 caratteri]` e `_ARAGGS$` fino a 20 caratteri e sono previste, con le stesse caratteristiche, le variabili dizionario `arcod()`, `ardes()` ed `aragg()`. Relativamente al movimento di magazzino (GETMM, ciclotabellare "mm") è consentita la visualizzazione dei 40 caratteri ripartiti fra codice principale e descrizione principale: sono previste le variabili strutture archivi `[_MMART$(R) e _MMDESS$(R) fino a 40 caratteri]` e sono previste, con le stesse caratteristiche, le variabili dizionario `mmart()` ed `mmdes()`. Per aggiungere nel documento l'intera estensione della descrizione aggiuntiva è necessario leggerla dall'anagrafica, cioè `aragg(_MMART$(R))` e nel ciclotabellare "mm" è `aragg(mmart())`.

Occorre considerare anche che il dizionario e la variabile di struttura relative alla descrizione con 72 caratteri, si riferiscono alla descrizione articolo scritta nel documento. Il dizionario relativo alla descrizione di 72 caratteri `[ocdes(), mmdes() e simili]` e la variabile di struttura `_MMDESS()` ritornano la descrizione fino a 72 caratteri (della descrizione totale) solo se la stessa è stata modificata nel documento (modifica della descrizione proposta dall'anagrafica articolo), altrimenti, se la descrizione non è stata modificata, ritornano la sola descrizione principale dell'articolo, fino a 40 caratteri (comprensivi della dimensione del codice principale). Anche se non modificata, la descrizione estesa è comunque prelevabile `D$=ardes(ocart()+aragg(ocart()))` che può essere confrontata con `stringa oc/mmdes()` per sapere se è stata o meno modificata.

Per comprendere meglio il comportamento delle suddette variabili basta fare un documento con un articolo che abbia una lunga descrizione di anagrafica (in modo che possa completare il campo della descrizione del documento), poi avanzare alla riga successiva e quindi tornare sulla riga per modificare totalmente quella descrizione anagrafica e avanzare nuovamente alla riga successiva: noterete che il programma visualizza due elementi in più, ovvero la descrizione di anagrafica e quella del documento.

Qualora si rendesse necessario stampare le descrizioni aggiuntive dei componenti della DBA (in presenza di attivazione del flag di modulistica relativo ai componenti DBA), occorre avvalersi delle apposite variabili Collage di modulistica debitamente riempite all'atto della registrazione/stampa del documento, attraverso variabili precedentemente memorizzate con i valori relativi alle descrizioni aggiuntive lette attraverso la GETATTAR. Soluzione alternativa potrebbe anche consistere, sempre attraverso la GETATTAR in una memorizzazione della descrizione nelle videate riportabili personalizzate, da stampare direttamente nel documento (come le variabili Collage di modulistica)

Esecuzione di istruzioni e valorizzazione di variabili in Collage

Essendo prevista nel medesimo modulo sia la possibilità di immissione del documento che di revisione, le variabili sono valorizzate a partire dall'etichetta `ON_STARTMM`: la quale viene attivata dopo l'etichetta `ON_OUT_MDTDOC`: e solo la prima volta che si esce dal campo `MDTDOC` (sigla documento). Per lo stesso motivo, l'attivazione dell'etichetta `ON_STARTMM`: è determinante per la corretta esecuzione degli eventi e/o istruzioni che vengono successivamente eseguite (`ON_DELM`., `GETATTMM`, ecc..).

Per capire meglio la possibilità di eseguire etichette e l'ordine in cui vengono eseguite è sufficiente testare l'esecuzione delle etichette stesse con un semplicissimo Collage. Per esempio, nel menu di Emissione Revisione Documenti è possibile verificarle in questo modo:

```
ON_START: VIMSG 1, "parte evento START": END
ON_IN_MDRICDOC: VIMSG 1, "parte evento RICDOC": END
ON_IN_MDTDOC: VIMSG 1, "parte evento IN_MDTDOC": END
ON_STARTMM: VIMSG 1, "parte evento STARTMM": END
ON_IN_ALL: VIMSG 1, "parte evento ON_IN_": END
ON_OUT_ALL: VIMSG 1, "parte evento ON_OUT_": END
ON_ERR_ALL: VIMSG 1, "parte evento ON_ERR_": END
```

Occorre evidenziare che all'uscita dal campo la GETATTMM deve "avere il tempo" di potere valorizzare la variabile richiesta all'uscita del campo (a meno che non sia già valorizzata). Questo lo si può notare, ad esempio, con due campi (descrizione e sconto relativi alla riga). Si esegua il seguente esempio, registrando una riga di articolo senza sconto, e poi si riproccesi la stessa riga inserendo lo sconto, e poi si riproccesi la riga per una terza volta.

```
ON_OUT_MDCODART:
ON_OUT_MDSCON:
GETATTMM
VIMSG 1, "descrizione =" + _MMDESS(1)
VIMSG 1, "sconto =" + _MMSCO(1)
END
```

Collage: verifica abbandono e uscita dal documento

```
ON_IN_2CONFUSC:
VIMSG 1, "_SINP$ =" + _SINP$ + "|" + "_TF =" + STR(_TF,0,0)
END
```

```
ON_OUT_2CONFUSC:
VIMSG 1, "_SINP$ =" + _SINP$ + "|" + "_TF =" + STR(_TF,0,0)
END
```

La gestione della finestra di uscita dall'immissione/revisione di un documento è uniformata con etichetta 2CONFUSC. Si testa `_SINP$` ed eventuale ulteriore dettaglio di scelta con `_TF`.

Collage: scrive la nota del documento nel campo descrizione della primanota

```
ON_PUTMM:
GETATTMM
NOT$=_MMNOT$(1)
END
```

```
ON_ENDMM:
DISVID 1
PUTTAST "#617,#617"
END
```

```
ON_IN_CICAUS:
PUTTAST "#501"
PUTTAST NOT$
PUTTAST "#610"
END
```

Collage: visualizzazione Partita Iva nell'elenco documenti

```
ON_IN_MDRICDOC:
SETTF "512.513" : T$="#"+STR$(_TF,0,0)
IF F=1 THEN LOCATE RIGA,86 : APRIFIN 1,1,-1,10,86,1,15 : INPUT"A",PIVA$,15 : CHIUDIFIN 1 : PUTTAST T$ : F=0
IF F=0 THEN PUTTAST "#501#501"
END
ON_IN_MDTDOC: PIVA$="" : PUTTAST "#501" : END
ON_IN_MDCAU: GETATTMM : PIVA$=pcnpi(_MMCLIS) : F=1 : PUTTAST "#505" : END
```

Nota: l'esempio è grossolano, richiede affinamento e/o modalità diverse di visualizzazione dell'output a seconda delle esigenze. L'esempio è idoneo per le nuove modalità di ricerca documenti tramite finestra elenco.

Anche se sconsigliato personalizzare su versioni obsolete, simili operazioni sono consentite anche in modalità di layout obsolete (ad esempio l'obsoleta modalità di ricerca tramite maschera di ricerca). Prendendo spunto dall'esempio relativo a _NCAM\$ si può ottenere quanto segue.

```
ON_START:
FLAG=1
END
```

```
ON_IN_MDRICDOC:
IF FLAG=1 THEN GOTO SALTO1
GOSUB FINESTRA
```

```
SALTO1:
END
```

```
ON_OUT_MDRICDOC:
IF FLAG=1 THEN GOTO SALTO2
```

```
SALTO2:
FLAG=0
END
```

```
FINESTRA:
X=GETPOSCURS(2) : Y=GETPOSCURS(2)
COD$=SCREEN$(-1,Y,23,9) : PIVA$=pcnpi(COD$)
LOCATE Y,105 : COLOR 3 : PRINT PIVA$;
RETURN
```

Totalizzazioni in Collage

Non essendo possibile utilizzare le RIGMM e TOTMM (eccetto se eseguite con Sprix da Shellsprix), per ottenere totali di riga e documento di valori che non siano in campi di input è necessario leggere a video (SCREEN\$) oppure ricostruire il calcolo.

Caso: Riferimenti Esterni

In testata movimenti di magazzino esistono 3 riferimenti esterni, con che criteri vengono caricati?
A volte sono caricati i primi due con valori uguali in altre viene caricato il terzo valore.

Motivo....

I riferimenti esterni contenuti nelle variabili di dizionario mmess(), mmesn(), mmesd()...che sono array di 3 elementi. Questi hanno una specifica logica:
Il primo valore dell'array ad es. mmess(1)() contiene i riferimenti letti e contenuti nel documento attuale.
Nel secondo sono contenuti gli eventuali riferimenti immessi nel documento che ha generato l'eventuale movimento di magazzino (carico-scarico)
Nel terzo sono contenuti gli eventuali riferimenti immessi nel documento ordine trasformato successivamente.

Esempio

OC1 trasformato in BC1 e BC1 trasformata in FT1

lettura valori su FT
mmess(1)() contiene i valori dei riferimenti immessi in fattura (FT1)
mmess(2)() contiene i valori dei riferimenti immessi nel documento (BC1)
mmess(3)() contiene i valori dei riferimenti immessi nel documento (OC1)

Caso: terminale factory, impossibilità di leggere/scrivere documenti FT, leggere e scrivere dati contabili o Docuvision.

Gli utenti associati a terminali di questo tipo non possono leggere e scrivere documenti di magazzino associabili alla prima nota e non hanno accesso ai dati di prima nota e la stessa cosa vale per Docuvision quando gli allegati non rientrano tra le tipologie ammesse. Il comportamento è quello previsto per questo tipo di terminale; se il terminale deve poter operare su questa tipologia di documenti, in prima nota o sulle classi docuvision non ammesse deve essere aperto con un livello diverso.

Messaggio autocodifica: articolo non trovato in archivio articoli

Sprix non effettua autocodifica (PUTMM, PUTIL,...), occorre una pre-codifica (PUTAR). L'autocodifica è consentita se gli articoli sono strutturati (es. struttura con automatica = C)

Messaggio: tipo documento obbligatorio

La registrazione di un tipo di documento non attivo o non previsto dal modulo, rilascia il messaggio: tipo documento obbligatorio. Questo perchè il programma non sarebbe in grado di identificare il tipo di registrazione non attivata o non consentita che si vuole effettuare.

Evento "Articolo non trovato" oppure "Articolo non trovato. Vuoi crearlo nuovo"

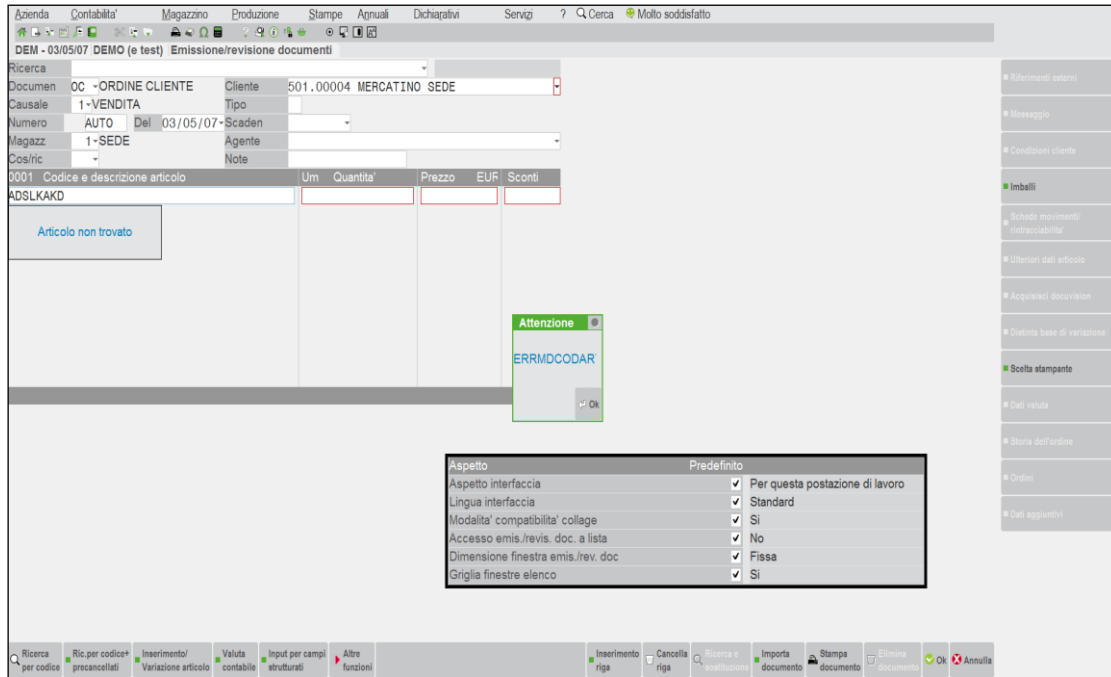
A seguito di implementazioni di funzionalità possono avvenire cambiamenti di sequenza degli eventi, ad esempio, come è avvenuto per

l'evento ON_ERR_MDCODART: L'inserimento di un articolo inesistente può dare il messaggio "articolo non trovato. Vuoi crearlo nuovo" anziché il messaggio "Articolo non trovato" generato dall'evento ON_ERR_MDCODART:

Ci sono metodi per ripristinare la vecchia situazione replicando una certa operatività che generi quel vecchio messaggio, oppure, potrebbe essere possibile ripristinare la vecchia sequenza originaria impostando una certa configurazione del client.

Nel primo metodo si tratta di implementare dei controlli nel codice per ottenere (replicare) la stessa condizione di evento errore. L'evento ON_ERR_MDCODART: (articolo non trovato) avviene quando la ricerca estesa è attiva ed avviene l'inserimento di un codice inesistente e la successiva sequenza di tasti Invio (apre l'elenco), poi Esc e ancora Esc. Si replica questa sequenza entrando in ON_IN_MDCODART per verificare il valore del tipo di ricerca attivo (dizionario sxrea) e disattivare la ricerca estesa con PUTTAST. L'evento ON_OUT_MDCODART apre la finestra che, presumibilmente, significa che non sta trovando l'articolo. Eseguendo la PUTTAST dei tasti Esc genera l'errore.

Nel secondo metodo si tratta di configurare il client in modalità old style, per esempio, come nelle seguenti immagini.

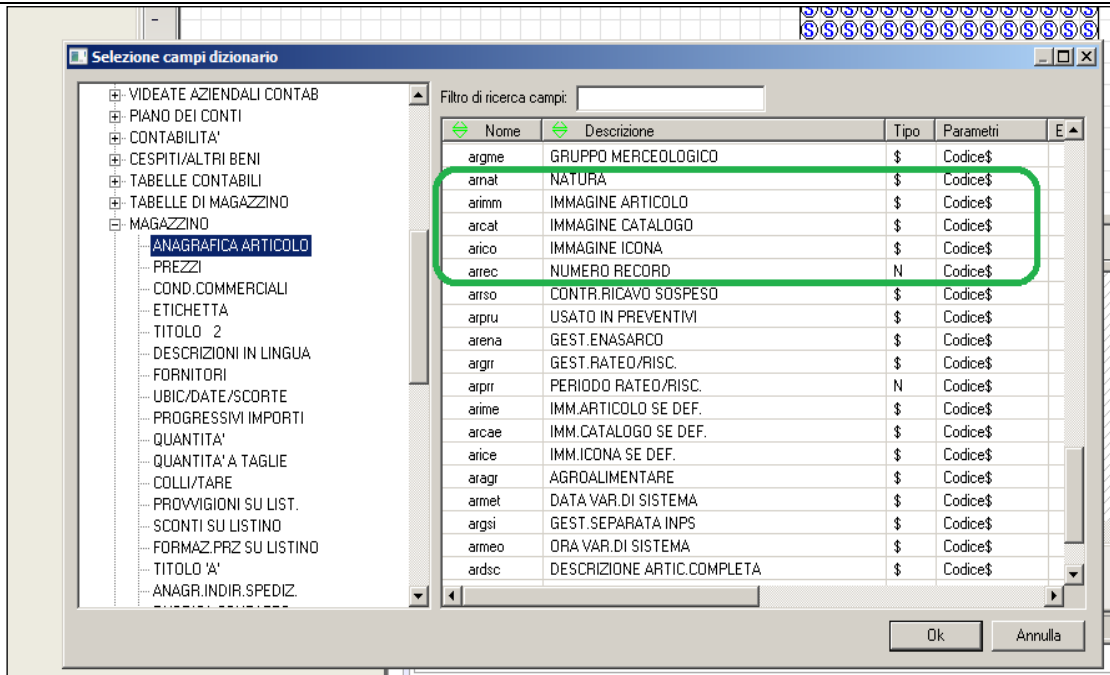


Esempi e casi dizionario

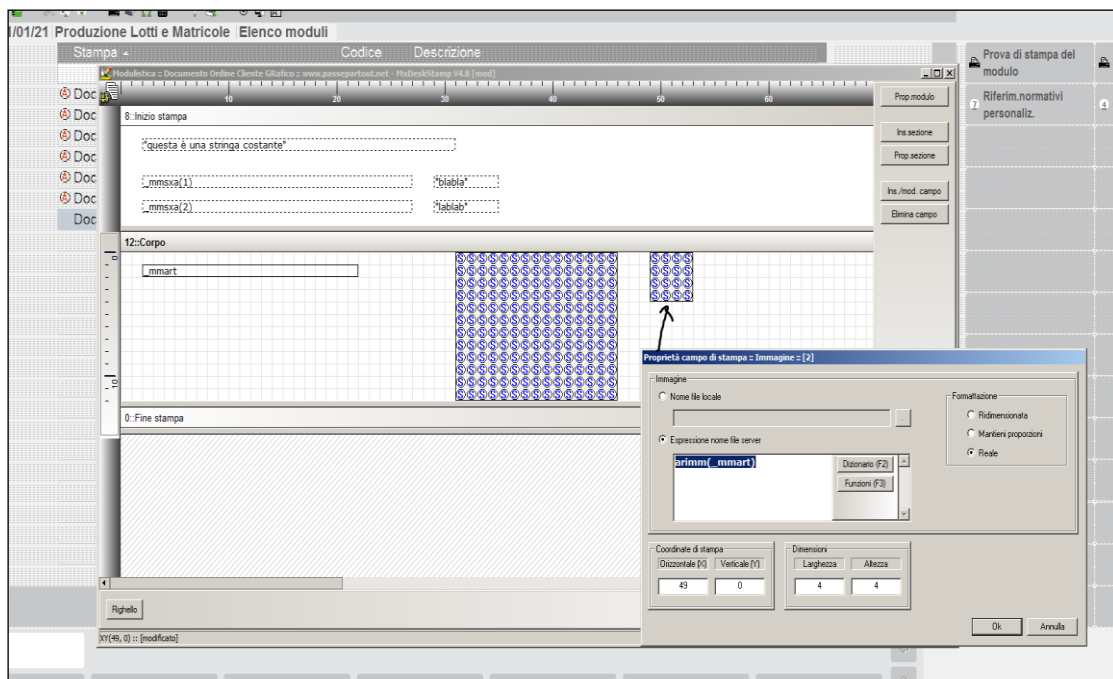
??tds() e ??dsc()	Per leggere righe descrittive in ciclotabellare occorre attivare l'istruzione LEGGIDESC	S/C/Cr/Cd/Sh
mmprd	Data di registrazione primanota. Nota: è però la cosiddetta "ultima foglia"	S
mmcds(), mmcos()	Sono la data e ora di sistema della creazione. Sono valori numerici, il loro ordinamento può essere ottenuto anche tramite la funzione MID\$(in quanto funzione utile per ottenere ordinamenti non previsti dal ciclotabellare). Esempio: Ordine 3 : MID\$(mmcds(),1,4)+MID\$(mmcds(),5,2)+MID\$(mmcds(),7,2) Ordine 2 : MID\$(mmcos(),1,2)+MID\$(mmcos(),4,2)	S
mmppr	Progressivo di registrazione primanota. Nota: come mmprd	S
mmord()	Sono criteri di ordinamento righe progressivi (3,4,7,...) e non necessariamente in sequenza (1,2,3,4,5...); possono essere riordinati in memoria a partire da 1, per ottenere la stessa sequenza visualizzata a video. La mmord() ritorna il numero di ordinamento di riga, ma il punto di partenza e l'ordine progressivo non sono rilevanti, perchè servono a garantire la sequenza fisica delle righe (3,4,7,9...), ma non l'esatto ordine progressivo (1,2,3,4,5...) delle righe visibili a video e numerate con il contatore di riga. Questo è dovuto ad una combinazione fra la logica di memorizzazione di dati relativi al record riga e di lettura della variabile dizionario ??ord(), nei casi di <u>trasformazioni e fusioni</u> dei documenti oppure nei casi di <u>modifiche alle descrizioni</u> delle righe articolo, variazioni di aliquota iva, o <u>aggiunte/cancellazioni di righe</u> (F7/F8) specialmente se descrittive, oppure nuove attivazioni di funzioni come le <u>taglie</u> , <u>imballi</u> , <u>videate riportabili</u> , <u>lotti</u> , ecc... Questo meccanismo riguarda anche i documenti sospesi e relativa variabile ??ord(): un OC nuovo dalla cui ocord() inizialmente si rileva una progressione delle righe (1,2,3,4,...), successivamente alle suddette operazioni può rivelare una sequenza diversa (es. 2,3,5,7,...) e crescente. La presenza di un qualsiasi ordinamento nel ciclo tabellare peggiora la situazione, provocando una lettura disordinata (che sembra casuale, es. 5,2,7,3,...), in quanto falsa l'esito del sort stabilito con ordinamento di default; in presenza di ordinamenti nel ciclo tabellare quindi, per normalizzare la sequenza è necessario ripristinare l'ordinamento su ??ord() dopo qualsiasi altro ordinamento effettuato, esempio Ordine 3 → ocsig()+ocser()+ocnum() Ordine 2 → ocord() La sequenza fisica viene riordinata (es. 2,3,5,7,...) e per avere anche una corrispondenza di quanto risulta a video (ordine progressivo di numero di riga:1,2,3,4,...)	S

	<p>può essere utile rinumerare in base alle FOR I=1 TO _NMM righe lette.</p> <p>2-1 3-2 5-3 7-4 ecc.</p> <p>La sequenza rilasciata ??ord() è crescente ma non progressiva, perchè considera l'ordine della sequenza dei record principali (es. codice articolo), agganciati ai quali ci sono altri record (relativi a videate, taglie, lotti,...). Rinumerando da 1 questi valori ritornati da ??ord si ottiene l'ordine di riga visualizzato. Esempio:</p> <p>CORPO DOCUMENTO</p> <table border="0"> <tr> <td>n.video</td> <td>??ord()</td> <td>??art()</td> </tr> <tr> <td>0001</td> <td>2</td> <td>- record cod. articolo A 3° record taglie 4° record lotti</td> </tr> <tr> <td>0002</td> <td>5</td> <td>- record cod. articolo B</td> </tr> <tr> <td>0003</td> <td>6</td> <td>- record (./) desc. aggiuntiva (è come un cod art descrittivo) 7° record desc. aggiuntiva 8° record videata</td> </tr> <tr> <td>0004</td> <td>9</td> <td>- record cod. articolo C 10° record imballi</td> </tr> </table> <p>Rinumerando la sequenza dei record 2, 5, 6, 7 in 1, 2, 3, 4 si ottiene la stessa sequenza di numerazione a video (n. video).</p> <p>Per quanto riguarda il numero di riga documento magazzino, a meno che non sia già disponibile una apposita variabile dizionario che riporti il numero di riga del documento, tale numero è ricavabile in vari modi fra cui la lettura diretta del campo a video (SCRREN\$) oppure la combinazione fra il campo ??nri() e ??ord(), di cui la prima è il numero di record (tanti numeri record appartenenti a uno stesso documento ??ser()+??sig()+??num() fanno la quantità di righe di quel documento) mentre la seconda è l'ordinamento di questi record (e il maggiore valore traibile per uno stesso ??ser()+??sig()+??num() è equivalente alla quantità di righe/record ciclata su quel documento, o l'equivalente di ??ulr() o l'equivalente di _NMM).</p>	n.video	??ord()	??art()	0001	2	- record cod. articolo A 3° record taglie 4° record lotti	0002	5	- record cod. articolo B	0003	6	- record (./) desc. aggiuntiva (è come un cod art descrittivo) 7° record desc. aggiuntiva 8° record videata	0004	9	- record cod. articolo C 10° record imballi	
n.video	??ord()	??art()															
0001	2	- record cod. articolo A 3° record taglie 4° record lotti															
0002	5	- record cod. articolo B															
0003	6	- record (./) desc. aggiuntiva (è come un cod art descrittivo) 7° record desc. aggiuntiva 8° record videata															
0004	9	- record cod. articolo C 10° record imballi															
mmsig()	<p>Le sigle documento sono memorizzate su disco in formato numerico, pertanto, nel ciclottabellare il singolo ordinamento in base a mmsig() non è sufficiente ad ottenere un elenco in ordine alfabetico di mmsig(). Occorre indicare nel seguente modo tramite la funzione MID\$ (funzione utile anche per ottenere ordinamenti non previsti dal ciclottabellare):</p> <pre> +-----+ Cod.Tab.:mm MOVIMENTI DI MAGAZZINO - TESTATA DOCUMENTO Ordine 3:MID\$(mmsig(),1,2) +-----+ </pre>	S															
mmrig()	<p>mmrig() coincide con MMNCRERIGA(1-N); rappresentano il numero univoco di creazione della riga; Cambia se si sostituisce la riga articolo con un altro articolo o la si cancella e si reinserisce. MMNRECRIGA rappresenta il numero record su disco della riga. Questo può cambiare quando si cancella e riscrive il documento.</p>	S															
mmtsd(), mmdsc(), ?tds(), ?dsc()	<p>Relative alla lettura di righe descrittive dei documenti magazzino di tipo punto(.) e virgola(.) sono attivate in Sprix con LEGGIDESC <par> (se <par>=1 sono abilitate, il default è <par>=0 cioè disabilitate) per leggere il secondo record fisico del corpo documento e quindi, non ordinabile sulla stessa linea del record principale in caso di letture sequenziali (ma solo sulla successiva). L'attivazione della lettura vale sia per il ciclottabellare che per le operazioni di mirroring.</p>	S															
documento trasformato	<p>I campi di dizionario che contengono le principali informazioni sulla trasformazione dei documenti di magazzino sono mmtsi(), mmtse(), mmtnu(), mmtda(). La trasformazione del documento di magazzino (es. bolla in fattura) rende utile mantenere il riferimento al documento di trasformazione. I campi di dizionario mmt?() contengono le informazioni relative al documento trasformato. Se quindi si trasforma la bolla (es BC5) in fattura (es. FT13), dentro alla bolla (mmsig)="BC" e mmnum()=... il dizionario mmtsi() è "FT" e il dizionario mmtnu() è = 13. Inoltre, se si vogliono conoscere anche i riferimenti relativi il documento d'ordine di riferimento bisogna utilizzare i campi di dizionario mmosi() e mmonu(). In altri termini, le variabili mmsig() mmser() mmnum() differiscono dalle voci mmtsi() mmtse() mmtnu() in quanto, in caso di semplice emissione di un documento che movimenti il magazzino (es. BC) e senza alcuna sua trasformazione in altro documento, tutte le voci si riferiscono ad esso; nel caso fosse intervenuta anche una sua trasformazione in altro documento (es. FT), mmsig() mmser() mmnum() si riferiscono ai dati del documento originario che ha movimentato il magazzino (BC) e mmtsi() mmtse() mmtnu() si riferiscono ai dati dell'ultimo documento prodotto dalla trasformazione (FT).</p> <p>Per verificare tramite le variabili di struttura il riferimento ai documenti di origine per gli articoli inseriti dei documenti di magazzino, considerare le seguenti variabili:</p> <ul style="list-style-type: none"> - [_MMMSI\$(1-200); _MMMSE(1-200); _MMMNNU(1-200); _MMMDA\$(1-200)] se il riferimento sono documenti di magazzino - [_MMOSI\$(1-200); _MMOSE(1-200); _MMONU(1-200); _MMODA\$(1-200)] se il riferimento sono documenti d'ordine <p>Tali variabili strutture archivi riguardano la testata del documento, quindi i riferimenti verranno inseriti una sola volta. Per collegare tali riferimenti agli articoli inseriti nel corpo del documento va utilizzata la variabile del corpo del documento _MMNTE(1-200) che indica il riferimento di testata a cui fare riferimento per determinare il documento di</p>	S															

	<p>origine di un articolo. Es. sintetico di come determinare tali riferimenti:</p> <pre>AZZVARSYS 4 GETMM "BC45" FOR I=1 TO _NMM VMSG 1,_MMART\$(I)+" *** "+_MMOSIS(_MMNTE(I))+STR\$_MMONU(_MMNTE(I)),0,0) NEXT END</pre> <p>In trasformazione documenti (più OF e le trasformo in una FF) e revisione di documenti con riferimenti di testata appare il messaggio PUTMM: 'I numeri di riferimenti alle testate devono essere consecutivi e progressivi'. Per evitarlo occorre riempire le variabili relative alle sottotestate e non ci devono essere in queste buchi nell'array. Le sottotestate sono a esempio i dati di testata degli ordini nella fattura, che vengono inseriti come riferimento. Sono pertanto le variabili che iniziano con _MMOxxx (quelle relative agli ordini) e _MMMxxx (relative ad eventuali mov. magazzino originari). L'eventuale aggiunta di righe in bolla, fa assumere la _MMNTE della riga precedente (la riga non assume la _MMNTE della bolla, ma della riga precedente proveniente dal ordine)</p>	
tabella "or"	Riguarda documenti OC, OF, PR	S
odtip()	Campo relativo agli impegni DBA e condizionato dal setting del parametro di magazzino "Sost.Autom.ord.matrice". Se questo è impostato "N" può rendere necessario il criterio odtip()="C" anziché "A".	S
ocsta()	La variabile non riporta lo stato "S" sospeso, perché il valore è riportato da octip() ed è un valore numerico (per esempio Evadibile = 4). Il dizionario ocsta() è lo stato di riga relativo alla tipologia di riga octip(). Per maggiori dettagli, vedere fra i primi campi della stampa standard di magazzino "ordini preventivi matrici". Si può notare che c'è il campo Stato (ocsta) e Tipo riga d'ordine (octip).	S
indirizzo spedizione	Bolle/fatture ordinate per indirizzo di spedizione: questi documenti sono ordinate per numero progressivo sezionale e le bolle in fattura per data documento e codice indirizzo spedizione (partendo dalle null ""). Per ottenere la stessa numerazione disattivare "raggruppa documenti". Alternativa: statistiche movimenti articolo (non per indirizzo); ciclotabellare movimenti ordinati per mmspe()	S
mfsmx() e mfbcs()	Sono il massimo numero di serie per la sigla documento voluta e la descrizione dell'esimo numero di serie; i dati arrivano sull'Applicazione Sprix Mobile in un archivio a sola lettura chiamato "desdoc".	S/C/Sm
mqscn() e mqqta()	Sono relativi all'archivio sconti a quantità chiamato "sconqua"	S/C/Sm
mmpsi(), mmpnu() mmpdt() mmpse()	Variabili relative ai dati del preventivo origine valorizzate solo con documenti PC che sono gestiti solo se attivato il modulo di controllo di gestione. Il documento PC è il documento "preventivo cliente" (non il PR) che è stato aggiunto nella lista dei documenti di magazzino per il modulo controllo di gestione.	S/C/Sm
mmdima(), mmamm(), mmmag(), mmama()	Variabili mmdma() DA MAGAZZINO SULLA RIGA e mmamm() A MAGAZZINO SULLA RIGA: sulla riga viene sempre riportato il magazzino, quindi queste variabili sono sempre valorizzate (se impostate). Variabili mmmag() NUMERO MAGAZZINO IN TESTATA e mmama() NUMERO A MAGAZZINO IN TESTATA: sono uguali a quelle di riga, se non si usa il magazzino sulla riga o se risultano tutti uguali; ritornano valore 0 invece se sono stati utilizzati magazzini diversi sulle righe.	S/C/Sm
mmtip()	Per leggere i movimenti con il valore della seconda unità di misura. Tra le funzioni di riga c'è ad esempio la qta UM documento e se il tipo riga (mmtip) è uguale a 2 (vale UM2) è quindi riferita alla seconda unità.	S/C/Sm
Struttura articolo	La lettura delle strutture da documento può avvenire attraverso l'apposita funzione GETNUMSTRUT, oppure attraverso il dizionario in base ad una sintassi simile alla seguente. Ad es. in caso di ordini fornitore: FOR I=1 TO 9 : mrctp(I)(arstr(ofart(I))) : NEXT	S/C/Sm
mvval()	Il cambio mvval() non contro Lira è memorizzato in negativo. Per ottenere il valore positivo occorre utilizzare ABS del campo o moltiplicarlo per -1.	S/C/Sm
Dizionario per riportare immagini / articolo / catalogo / icona nel corpo del modulo documento		



Quando viene selezionato [F2] per inserire il campo, scegliere "Immagine"



Esempio: registra un OC e ne legge alcuni totali. Registra un secondo OC con DBA e aggiorna il primo cancellando+aggiungendo una riga articolo. Evade i due OC in BC aggiungendo la provvigione. Visualizza riferimenti delle righe della BC e cancella gli OC; infine procede alla fatturazione con cancellazione di BC e visualizzazione dei riferimenti del movimento.

```

_MMSIG$="OC";_MMSER=1;_MMNUM=0;_MMDAT$=sxdat;_MMCLI$="201.00001";_MMMAG=1
_MMTPR$(1)="R";_MMART$(1)="04M-MI";_MMQTA(1)=1;_MMPRZ(1)=100;_MMALIS(1)="20"
_MMTPR$(2)="R";_MMART$(2)="04M-SI";_MMQTA(2)=2;_MMPRZ(2)=200;_MMALIS(2)="20"
_MMRICSTP=1;PUTMM 2:IF _ERRMM<>0 THEN GOTO ERRORE
RIGMM 1,2,2:TOTMM
VIMSG 1,"Prez."+STR$( _MMR_PREZZO,2,0)+"Imponib."+STR$( _MMR_IMPON,2,0)
VIMSG 1,"Totale Imponibile "+STR$( _MMT_TIMPON,2,0)
OSIG1$=_MMSIG$;OSER1$=STR$( _MMSER,0,0);ONUM1$=STR$( _MMNUM,0,0)

```

```

AZZVARSYS 4
_MMSIG$="OC";_MMSER=1;_MMNUM=0;_MMDAT$=sxdat;_MMCLI$="201.00001";_MMMAG=1
_MMTPR$(1)="R";_MMART$(1)="PCDESK";_MMQTA(1)=1;_MMPRZ(1)=300;_MMALIS(1)="20"
_MMNRECRIGA(1)=-1;PUTMM 2:IF _ERRMM<>0 THEN GOTO ERRORE
OSIG2$=_MMSIG$;OSER2$=STR$( _MMSER,0,0);ONUM2$=STR$( _MMNUM,0,0)

```

```

GETMM_EXT OSIG1$+OSER1$+" "+ONUM1$,"201.00001"
IF _ERRMM<>0 THEN GOTO ERRORE
_MMTPR$(1)=""
_MMTPR$(3)="D";_MMTSD$(3)="";_MMDEE$(3)="Riga descrittiva"
SETPARSOSTMATR "S":PUTMM 0:IF _ERRMM<>0 THEN GOTO ERRORE

```

```
AZZVARSYS 4
GETMM OSIG1$+ONUM1$+"+"+OSIG2$+ONUM2$ : IF _ERRMM<>0 THEN GOTO ERRORE
_MMMSG$="BC" : _MMSER=1 : _MMNUM=0 : _MMAGE$="401.00001"
FOR I=1 TO 3 : _MMORD$(I)="E" : _MMPRO(I)=5 : NEXT
PUTMM 2 : IF _ERRMM<>0 THEN GOTO ERRORE
BSIG$=_MMMSG$ : BSER$=STR$_MMSER,0,0 : BNUM$=STR$_MMNUM,0,0
```

```
AZZVARSYS 4
GETMM BSIG$+BNUM$
VIMSG 1, "visualizzo i Riferimenti delle righe della bolla"
FOR I=1 TO _NMM
VIMSG 1,_MMART$(I)+"->"+_MMOSIS$_MMNTE(I))+STR$_MMONU$_MMNTE(I),0,0
NEXT
```

```
AZZVARSYS 4
DELMM_EXT OSIG1$+OSER1$+"/" +ONUM1$+"+"+OSIG2$+OSER2$+"/" +ONUM2$, "201.1"
IF _ERRMM<>0 THEN GOTO ERRORE
```

```
AZZVARSYS 4
GETMM BSIG$+BSER$+"/" +BNUM$ : IF _ERRMM<>0 THEN GOTO ERRORE
_MMMSG$="FT" : _MMSER=1 : _MMNUM=0
PUTMM 1 : IF _ERRMM<>0 THEN GOTO ERRORE
```

```
AZZVARSYS 4
GETMM BSIG$+BNUM$
VIMSG 1, "visualizzo i Riferimenti delle righe della fattura"
FOR I=1 TO _NMM
VIMSG 1,_MMART$(I)+"->"+_MMMSIS$_MMNTE(I))+STR$_MMMNU$_MMNTE(I),0,0
NEXT
```

```
DELMM BSIG$+BSER$+"/" +BNUM$ : IF _ERRMM<>0 THEN GOTO ERRORE
ERRORE: IF _ERRMM<>0 THEN VIMSG 4, _ERRMM$
```

Esempio: inserimento di riga descrittiva con testo variabile di articolo di tipo "V"

Per inserire nel documento il testo variabile di un articolo di tipo "V" occorre definire la tipologia della riga (_MMTPR(1-200)="D") e decidere se portare nei documenti successivi la descrizione inserita nel documento che si sta creando/modificando (_MMTSD(1-200)="") oppure _MMTSD(1-200)=",". Infine occorre inserire il testo variabile di un articolo all'interno della variabile _MMDEE\$(1-200) che rappresenta la descrizione della riga in questione. _MMDEE\$(1-200) può contenere sia una stringa di testo qualunque, cosicché se si vuole inserire la descrizione variabile di un articolo di testo variabile (tipo "V") basta utilizzare il campo di dizionario tetes(codice\$,riga).

Dato che si sta inserendo un commento all'interno del documento di magazzino che si sta creando/modificando, non è necessario inserire il codice articolo (_MMART\$(1-200)) in quanto la variabile _MMDEE\$(1-200) rappresenta esclusivamente una descrizione che si inserisce all'interno di una riga. Esempio:

```
AZZVARSYS 4

'DATI DI TESTATA
_MMMSG$="OC" : _MMSER=1 : _MMNUM=0 : _MMDAT$="20210427" : _MMCLI$="501.00003" : _MMMAG=1

'DATI DI RIGA
'RIGA NUMERO 1
_MMTPR$(1)="R" : _MMART$(1)="748_RED" : _MMALI$(1)="20" : _MMQTA(1)=741

'RIGA NUMERO 2
_MMTPR$(2)="D" : _MMDEE$(2)=tetes("DESCR V",1) : _MMTSD$(2)=","

PUTMM 0
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Caso: PUTMM di una fattura proforma (FP)

Il documento proforma risulta una funzionalità di emissione revisione documenti e non risulta gestibile tramite l'istruzione PUTMM. Per poter gestire questa casistica si può valutare questa possibilità: Valorizzare le variabili _MM analogamente ad una fattura da emettere (FT); Creare un formato di stampa (nelle modulistica documenti, anche tramite duplicazione di un formato esistente) specifico per queste stampe (es. FTP), dove all'interno dichiarare come sigla e descrizione documento (tramite utilizzo delle costanti VCOS) FP Fattura Proforma, Utilizzare con _MMMSG\$="FT" e _MMMODSTP\$="P" (nel caso sia stabilito questo formato FTP); Utilizzare la PUTMM 4 (non memorizza il documento, lo stampa solamente).

Esempio: PUTMM, utilizzo delle variabili _MMTANNOLET(), _MMTPROGLET(), _MMANNOLETT(), _MMPROGLETT(), nel caso di documenti emessi con dichiarazione d'intento

'Creo una BC per un cliente con dichiarazioni d'intento 2022/1 e 2022/2 attive.
'Scarico il plafond disponibile dalla dichiarazione 2022/1 per la riga articolo e 2022/2
'per le spese di trasporto inserite nel piede del documento.

```
'Dati generici da sostituire con i propri
DATA$=xsdt
SERIE=1
MAGAZZINO=1
CLIENTE$="501.2"
ARTICOLO$="PIZZA"
QUANTITA=50
PREZZO=15
C_TRASPORTO=50

'Dati di testata
_MMMSG$="BC"
_MMSER=SERIE
_MMDAT$=DATA$
_MMMAG=MAGAZZINO
_MMCLI$=CLIENTE$
_MMALISPTRA$="N08,3" 'codice di esenzione per acquisto in regime di dichiarazione d'intento. va obbligatoriamente inserito altrimenti non vengono salvate
'le informazioni relative alla dichiarazione d'intento (visualizzabile tramite apposito tasto dal piede del documento)

'Riga articolo
_MMANNOLETT(1)=22 : _MMPROGLET(1)=1
_MMTPR$(1)="R" : _MMART$(1)=ARTICOLO$ : _MMQTA(1)=QUANTITA : _MMPRZ(1)=PREZZO : _MMALI$(1)="N08,3"

'Piede, spese di trasporto
_MMTANNOLET(1)=22 : _MMTPROGLET(1)=2
_MMTSR$(1)="V" : _MMPORS$(1)="D" : _MMTPSS$(1)="V" : _MMVAS(1)=C_TRASPORTO
'Scrittura documento
PUTMM 0
IF _ERRMM<>0 THEN VIMSG 1, _ERRMM$ ELSE VIMSG 1, "EMESSO BC1/" +STR$_MMNUM,0,0
END
```

Caso: PUTMM non aggiunge la riga articolo e non funziona con il solo codice alternativo

E' fondamentale aggiungere il tipo riga _MMTPRS(1-200). Nella registrazione è inoltre necessario indicare il codice principale (che deve essere già codificato in anagrafica); prima di effettuare la PUTMM, occorre leggerlo da anagrafica con ciclotabellare sulla base del codice alternativo che viene "passato".

Esempio Collage: all'uscita del campo Iva confronta il prezzo unitario di riga scontato con quello di listino anagrafico e rilascia messaggio se è inferiore

ON_OUT_MDALI:

```
RIG$=SCREEN$(-2,7,1,4) : NRIG=VAL(RIG$) : GETATTMM
PRZSCO=_MMPRZ(NRIG)-((MMPRZ(NRIG)*VAL(_MMSCO$(NRIG)))/100) : VIMSG 1, STR$(PRZSCO,4,0)
IF PRZSCO<arprz(1)(_MMART$(NRIG)) THEN VIMSG 1, "inferiore"
END
```

Esempio Collage: variante del precedente, ma che sfrutta la quantità e gli importi minimi delle condizioni commerciali di anagrafica articolo (ponendo come quantità 1 per rendere l'importo fatturabile come se fosse un prezzo unitario). NOTA BENE: questo utilizzo delle condizioni minime commerciali è improprio e non impedito, ma fortemente sconsigliato in quanto viene persa la funzionalità dei messaggi sul minimo/massimo fatturabile (notare che viene disattivato il messaggio di minimo fatturabile non raggiunto).

ON_IN_MDCODART:

```
'catturo la riga precedente
RIG$=SCREEN$(-2,7,1,4) : NRIG=VAL(RIG$)-1
IF NRIG=0 THEN END ELSE GOSUB ELABORA
END
```

ON_IN_MDVMINF: ' nascondo il messaggio di minimo fatturabile

```
DISVID 1
PUTTAST "#501"
END
```

ELABORA:

```
RIG$="" : NRIG=0 : PRZSCO=0 : PRZUNI=0
AZZVARSYS 4
GETATTMM
' prezzo unitario di riga scontato
PRZSCO=_MMPRZ(NRIG)-((MMPRZ(NRIG)*VAL(_MMSCO$(NRIG)))/100)
' calcolo il prezzo unitario delle condizioni commerciali [Importo(minimo) DIVISO la Quantita(minima)]
PRZUNI=armin(_MMART$(NRIG))/armf(_MMART$(NRIG))
' se il prezzo unitario di riga e' inferiore a quello di condizioni commerciali rilancio il messaggio
IF PRZSCO<PRZUNI THEN VIMSG 1, "prezzo unitario di riga scontato " + STR$(PRZSCO,4,0) + " | inferiore al prezzo unitario " + STR$(PRZUNI,4,0)
RETURN
```

Caso: comparazione documento in revisione

Non sono disponibili in sprix funzioni di comparazione della ristampa del documento (come può effettuare la stampante Docuvision). E' disponibile invece il confronto binario fra file.

Esempio: Lotti

FOR R=1 TO _NMM

```
LPRINT " _MML_INIZLOT :"+ STR$( _MML_INIZLOT(R),0,0);
LPRINT " _MML_MXQTLOT :"+ STR$( _MML_MXQTLOT(R),0,0)
```

```
IF ( _MML_INIZLOT(R) = 0) THEN GOTO SALTA
FOR L=_MML_INIZLOT(R) TO ( _MML_INIZLOT(R)+ _MML_MXQTLOT(R) - 1)
LPRINT " _MML_IDLOT(" + STR$(L,0,0) + ") :"+ STR$( _MML_IDLOT(L),0,0);
LPRINT " _MML_NDQTA(" + STR$(L,0,0) + ") :"+ STR$( _MML_NDQTA(L),0,0);
LPRINT " _MML_LQTA (" + STR$(L,0,0) + ") :"+ STR$( _MML_LQTA(L),0,0);
LPRINT " _MML_LCOLLI (" + STR$(L,0,0) + ") :"+ STR$( _MML_LCOLLI(L),0,0)
FOR TG=1 TO 32
LPRINT " _MML_LQATGL(" + STR$(TG,0,0) + ") :"+ STR$( _MML_LQATGL(L,TG),0,0)
NEXT
SALTA:
NEXT
NEXT
```

Scrittura

Quando vengono estese variabili (come dalla gestione Lotti alla gestione "M" Lotti/Matricole), significa che concorrono ulteriori elementi da considerare. In questo caso la struttura lotti e matricole si articola diversamente da quella originaria (Lotti) e la logica risulta diversa. In ogni caso, come buona prassi per capirne le logiche, prima di ottenere risultati in scrittura occorre ottenerne in lettura.

In questo modo, si capisce che le nuove variabili sono indicizzate (es. _MML_IDLOT(L)), ma non hanno per indice la riga bensì un indice progressivo (L) che va da uno (1) alla somma di tutti i lotti inseriti nel documento. Questo consente di inserire eventuali ulteriori elementi gestendole con criterio.

Esempio: come inserire un lotto in una riga di un documento che ha la riga 1 con _MML_INIZLOT=1 e _MML_MXQTLOT = 3 (lotto 1 2 3) la riga 2 con _MML_INIZLOT=4 e _MML_MXQTLOT = 3 (lotto 4 5 6) come si inserisce un quarto lotto riferito alla riga 1?

Soluzione: per inserire una nuova matricola sulla riga 1 occorre reimpostare le relative variabili, quelle relative alla riga uno, dove viene aggiunto un lotto (su L=4), e quelle della riga due i cui indici risultano differenti (L parte da 5). In altri termini occorre scalare di una posizione i vettori che usano la L a partire dalla posizione 4 e reimpostare tutte le variabili _MML_INIZLOT(R) di tutto il documento e quindi compilare i vettori L nella posizione 4 e reimpostare le variabili _MML_MXQTLOT(R) della riga su inserire il nuovo lotto. In ogni caso, tutte le variabili che devono essere aggiornate (anche la quantità della riga documento, non solo le matricole).

E' consigliabile effettuare una prova valorizzando un documento con i valori da gestire, e poi effettuare la lettura di tutte le variabili interessate da sprix, specificate nel manuale, per comprendere meglio come assegnarle.

Messaggistica lotti

Alcuna messaggistica lotti, es. "Lotto caricato da altro documento" ... "Lotto n., Codice Lotto non corretto, Data di carico diversa da quella del documento (riga n)",... possono essere relazionati alla configurazione dei lotti (es. parametri tipo lotto [F8]), modalità di creazione anagrafiche lotto e relative operatività di utilizzo dei lotti (corretta o non corretta rispetto a quanto configurato).

Caso: tipo lotto errato in sostituzione articolo

Se si prova a caricare un documento al cui interno c'è un articolo di tipo lotto e si prova a sovrascrivere la riga con un altro articolo il cui tipo lotto è differente, al momento della PUTMM la procedura rilascia questo errore: Errore :Tipo lotto errato riga X

```
Esempio :
GETMM "BC105"
```

vengono cambiati _MMART\$(1)="1260" : _MMIDLOT(1,1)=1
 in _MMART\$(1)="PIPP0" : _MMIDLOT(1,1)=4
 PUTMM 0

"1260" E "PIPP0" sono articoli che appartengono a tipi lotti diversi, se invece di "PIPP0" si utilizza un articolo con lo stesso tipo di lotto, tutto funziona correttamente

Il motivo di questo comportamento è che non viene azzerata l'intera struttura di riga (1), e quindi nemmeno il tipo lotto (che "sta sotto"). E' noto che Sprix non ha necessariamente gli stessi automatismi del Gestionale (es. anche PUTPN non aggiorna la primanota se non si cancella la vecchia registrazione, oppure PUTMM in trasformazione documento se non si cancella il documento origine)

Procedura corretta:

```
GETMM "BC105"
_MMTPR$(2)=_MMTPR$(1)
_MMART$(2)="PIPP0"
_MMIDLOT(2,1)=4
_MMQTA(2)=_MMQTA(1)
_MMALIS$(2)=_MMALIS$(1)
_MMLQTA(2,1)=_MMLQTA(1,1)
_MMTPR$(1)="<<<<<<(notare)
PUTMM 0
```

Il programma ricompone il documento, assegnando la riga due al posto della riga uno (elimina la riga uno - azzerata - e la sostituisce con la due, compattando il corpo del documento)

Caso: dati tipo lotto obbligatori riga N non impostati

Impostando _MMIDLOT(R,L)=-1 nella PUTMM significa autocodifica del lotto. Il messaggio avverte che tra i campi personalizzati del lotto c'è un dato obbligatorio che non è stato valorizzato. Sprix non prevede automatismi, pertanto, vanno replicate in sprix le stesse operazioni che si fanno nella sessione di lavoro.

Di conseguenza, le alternative sono:

- togliere i dati obbligatori per consentire l'autocodifica del lotto (fra i campi personalizzati impostare valore di default)
- creare il lotto tramite le sue variabili di struttura e poi, successivamente, assegnarlo

Caso: valore di elemento "L" in autoprelievo -2

Scarico automatico lotti: il valore dell'elemento di riga lotto (L) non influisce nell'autoprelievo "-2" (eccetto se non valorizzata in quanto eccede i limiti dell'array), in quanto scarica fino a raggiungere la quantità della riga Articolo (R). L'elemento (L) è utile quando si vuole attribuire un determinato idlotto a ciascuna riga lotto (L).

Casi: fattura totale del documento di magazzino e fatturato per codice/indirizzo del destinatario

Il fatturato totale del documento di magazzino non è un dato memorizzato, ma è un dato calcolato. Questo dato non esiste nella struttura né nel dizionario, tuttavia sono disponibili le "Funzioni di Riga" del dizionario, la possibilità di eseguire letture a video del dato di interesse ed istruzioni di computazione dei totali di riga (RIGMM) e documento (TOTMM).

Per calcolare il Fatturato per codice/indirizzo del destinatario, si possono scegliere due strade distinte: utilizzare un ciclo tabellare oppure utilizzare un file esterno. Il ciclo tabellare è una soluzione semplice ed immediata: filtrando i documenti sulla base del codice di destinazione (mmspe(<>"")) ed inserendo tale codice come rottura del ciclo, si possono eseguire tutte le elaborazioni di interesse. Tuttavia, non esiste un campo di dizionario che contenga il totale del documento e quindi si rende necessario effettuare la sommatoria dei valori provenienti dalle funzioni di riga e l'introduzione di approssimazioni nel calcolo del totale. Vedere anche le istruzioni RIGMM e TOTMM.

Un metodo alternativo potrebbe consistere nell'utilizzo di un file esterno gestito contemporaneamente da un Collage ed uno Sprix. Con Collage, al momento dell'emissione di un documento di magazzino, si potrebbe verificare che il codice di destinazione sia pieno e, in questo caso, si dovrebbe aggiornare un file esterno che contiene le poche informazioni di interesse (ad esempio sigla documento, serie documento, numero documento, codice di destinazione, totale del documento). In particolare, poiché non esiste una variabile che contenga l'importo totale del documento è utilizzabile la lettura a video SCREEN\$; una volta memorizzate le informazioni in un file esterno (possibilmente a chiavi), questo potrebbe essere riaperto da uno Sprix che elabori tutte le informazioni di interesse (ad esempio le somme dei totali dei documenti per codice di destinazione).

Esempio: visualizza totali del documento tramite funzione RIGMM / TOTMM (nota: per entrambe valgono le stesse regole della PUTMM)

'Azzerare le variabili di sistema del documento di Magazzino
 AZZVARSYS 4

'Definizione dei dati di testata del documento (NB: e' attivo il magazz. di riga)
 _MMSIG\$="FT" : _MMSER=1 : _MMNUM=0 : _MMDAT\$="20210101" : _MMCLI\$="201.00001" : _MMMAG=1

'Prima riga documento (2 articoli a 111 Euro)
 _MMRMAG(1)=1 : _MMTPR\$(1)="R" : _MMART\$(1)="04M-MI" : _MMQTA(1)=2 : _MMPRZ(1)=111 : _MMALIS\$(1)="20"

'Seconda riga documento (1 articolo a 111 Euro)
 _MMRMAG(2)=1 : _MMTPR\$(2)="R" : _MMART\$(2)="04M-SI" : _MMQTA(2)=1 : _MMPRZ(2)=111 : _MMALIS\$(2)="20"

'Registrazione documento e verifica di errori
 PUTMM 0
 IF _ERRMM<>0 THEN VIMSG 4, _ERRMMS

'Esecuzione totali della prima riga e loro visualizzazione
 RIGMM 1,2,2
 VIMSG 4, "totale prezzo della prima riga "+STR\$(_MMR_PREZZO,2,0)
 VIMSG 4, "totale imponibile della prima riga "+STR\$(_MMR_IMPON,2,0)

'Esecuzione totale documento e visualizzazione
 TOTMM
 VIMSG 4, "totale imponibile del documento "+STR\$(_MMT_TIMPON,2,0)

'Dati per la verifica in Emissione revisione Documenti di Magazzino
 VIMSG 4, "documento "+_MMSIG\$+" "+STR\$(_MMSER,0,0)+"/"+STR\$(_MMNUM,0,0)

'Termine incondizionato della routine
 END

Caso: TOTMM con rivalsa di Iva sconto 130

In caso di movimenti con sconto 130, solo contabilmente i dati vengono girati per azzerare il valore della merce, all'interno del documento il valore dell'imponibile è presente. Eseguendo la TOTMM _MMT_IMPON riporta il valore corretto dell'imponibile. E' presente la variabile _MMT_OMA130 che riporta il totale della merce che viene conteggiata per le righe interessate dalla rivalsa d'iva. Per avere il totale del documento in presenza della rivalsa d'iva la formula deve essere: _MMT_TIMPON + _MMT_TIMPOS - _MMT_OMA130
 Tale totale funziona anche per quei documenti che non hanno righe con rivalsa d'iva.

Caso: ripristino di un ordine cliente (OC) a seguito della cancellazione di una bolla cliente (BC)

Si supponga una BC che proviene dalla trasformazione di un OC da ripristinare a seguito della cancellazione della medesima BC

GETMM "BC1/345" (carico tutta la struttura del documento bolla)

```
VIMSG 1, _ERRMM$ (verifico se l'ho caricata)
SIG_BC$_MMSG$: SER_BC=_MMSER : NUM_BC=_MMNUM '(memorizzo la chiave del documento BC)
SIG$_MMOSI(1) : SER=_MMOSE(1) : NUM=_MMONU(1) '(memorizzo la chiave del documento OC di origine)
DELMMSG SIG_BC$+STR$(SER_BC,0,0)+"/"+STR$(NUM_BC,0,0) '(cancello il documento BC, ma restano caricate le variabili poichè non eseguo un AZZVARSYS)
VIMSG 1, _ERRMM$ (verifico se l'ho cancellato)
_MMSG$_MMSG$: _MMSER=SER : _MMNUM=NUM '(inizializzo la chiave del documento con quella dell'ordine)
PUTMM 0 '(registro senza primanota)
VIMSG 1, _ERRMM$ (verifico l'esito della registrazione)
END
```

Attenzione: l'esempio è molto semplice e rudimentale; la complessità di questa operazione può essere molto superiore poichè dipende dall'operatività sui documenti in questione. L'esempio, infatti, non considera molti aspetti importanti, fra cui le questioni di evadibilità delle righe, i riferimenti di testata delle righe in caso di fusione e trasformazioni multiple, la confluenza di valori "vari" nei campi che li prevedono (es. le note), evasioni parziali con residuo, gestione matrici, protocolli, presenza di DBA, ecc....

Caso: calcolo dell'imposta del documento di magazzino

Nel calcolo dell'imposta del documento di magazzino sono coinvolti molteplici fattori, pertanto non è possibile predeterminare una procedura di calcolo che esegua il totale dell'imposta. Tuttavia, sono comunque disponibili le tabelle "Funzioni di Riga" e "Riga Articolo" che contengono le informazioni necessarie (quantità, valore, aliquota Iva, ecc..) per l'opportuna formula di calcolo in base alle esigenze di ciascun caso in relazione alle circostanze fiscali. In alternativa, valutare la possibilità di eseguire letture a video ed istruzioni di totalizzazioni di riga (RIGMM) e documento (TOTMM).

Caso: contenuto della variabile mmruv() e valore merce depurato degli sconti pagamento

La variabile del valore movimentato nell'unità di misura primaria um1, cioè mmruv(), contiene il valore relativo alla merce imputata in riga al netto degli sconti di riga.

ESEMPIO:

```
FATTURA EMESSA
CODICE E DESCRIZIONE  UM  QTA  PREZZO  SCONTO  IVA
ARTICOLO              PZ   1    100     30     20
```

PAGAMENTO: RIMESSA CON SCONTO M 4%

Per cui, TOTALE MERCE 67,2 (70 - (4% SU 70))

La variabile in questo caso contiene 70 il valore riferito alla riga e non 67,2 che è un dato relativo anche ad eventuali altre righe. Lo sconto condizionato di tipo "merce" rappresenta una delle possibili "rettifiche di valore" effettuabili sul totale della merce movimentata. Ci sono altre imputazioni effettuabili a livello di totali, che si possono effettuare sul cosiddetto piede del documento, come le spese addebitate per il trasporto, le spese banca ed altro. Sarebbe condivisibile l'esigenza di avere una variabile che riporti il valore totale merce movimentato in riga per finalità statistiche, ma per le ragioni citate sarebbe impropria la creazione di una ulteriore variabile di riga documento, in quanto certi valori assunti sono relativi a tutte le righe. Assunto questo, è valutabile la creazione di una variabile di appoggio per gestire questa informazione, in quanto tutti i valori di cui sopra sono ricavabili attraverso le variabili del documento e un calcolo basato sulla lettura di questi ed un determinato criterio di "ripartizione" per assegnare alle singole righe. $N = mmruv() - (mmruv() * cpsco(mmpag()) / 100)$

Caso: argomentazioni sulle variabili mmruv() e mmrdv()

Le variabili mmruv() ed mmrdv() di distinguono in base alla presenza dell'eventuale seconda unità di misura dell'articolo nella generazione del documento di movimento magazzino. L'utilizzo in fatturazione della seconda unità di misura (con relativo coefficiente di conversione), può provocare differenze negli importi: passare dall'unità di misura primaria a quella secondaria, comporta la differenza di qualche centesimo di decimale fino all'intero decimale stesso, poichè non è possibile poter prescindere (a livello matematico) dall'utilizzo dell'una o dell'altra unità di misura (semprechè nella valorizzazione, non si possano utilizzare tutti i decimali e non sia coinvolto il coefficiente di conversione), e dato che avviene un arrotondamento in medio-eccesso al secondo decimale, la differenza non può non esistere.

Per vedere eventuali differenze è possibile provare ad effettuare il cambio di unità di misura una prima volta, lasciando il cambio di prezzo che il programma propone (perchè effettua la conversione in base al coefficiente stabilito a monte!) e si nota che mmruv() e mmrdv() inizialmente coincidono; cercando successivamente di effettuare (in un'altra registrazione e con gli stessi dati) il cambio di unità di misura una seconda volta, modificando il prezzo unitario si nota che mmruv() e mmrdv() non coincidono più poichè, per lo stesso articolo è stato utilizzato lo stesso prezzo unitario per due sistemi di misura differenti (che hanno un coefficiente di conversione che le mette in relazione).

Occorre quindi prestare attenzione a queste due considerazioni:

- 1) la valorizzazione dello scarico di magazzino è sempre in unità di misura principale (mmruv());
- 2) la registrazione di primanota avviene in base alla valorizzazione scelta sulla riga del documento di magazzino (mmruv() o mmrdv()) a seconda dell'unità di misura utilizzata;

Qualora la riga articolo risultasse movimentata in base alla seconda unità di misura, la valorizzazione non dovrebbe effettuarsi in base alla mmruv(), bensì in base alla mmrdv(), ricordandosi comunque che in entrambi i casi, c'è un arrotondamento per medio-eccesso.

La variabile mmrdv() dovrebbe essere stata creata, proprio per tener conto dell'eventuali differenze che possono emergere e che a livello contabile potrebbero manifestarsi, dato che la registrazione di primanota avviene appunto in base alla valorizzazione scelta.

Quando si operava in Lire ciò non era molto "rilevante", ma in Euro può avere effetti, in sede di confronti fra bilancio e valorizzazioni di magazzino. In un documento movimentato con seconda unità di misura, la valorizzazione con mmruv() anziché con mmrdv() potrebbe provocare differenze, in sede di confronti fra le valorizzazioni "complessive" di contabilità e di magazzino.

La variabile mmrdv() dovrebbe essere stata creata proprio per tener conto della differenza in questione e che a livello contabile potrebbe manifestarsi, dato che la registrazione di primanota avviene, appunto, in base alla valorizzazione scelta.

Caso: ordinamento delle bolle per indirizzo di spedizione

Bolle/fatture vengono ordinate in base alla numerazione progressiva del sezionale impostato. Tale numerazione, però, non segue un ordinamento preciso delle bolle presenti in fattura in quanto, indipendentemente dalla loro numerazione, principalmente sono adottati due criteri di selezione basati prima sulla data del documento e poi sul codice dell'indirizzo di spedizione; entrambi questi sistemi, ordinano le bolle a partire da quelle che sono sprovviste di tali elementi. Per ottenere la stessa numerazione, oltre a quanto detto, occorre disattivare il "raggruppa documenti".

Alternative per ottenere determinati ordinamenti si possono ottenere con la stampa delle statistiche movimenti articolo (ma non per indirizzo); una semplice soluzione potrebbe consistere nell'utilizzo di una stampa Sprix con ciclo tabellare dei movimenti di magazzino ordinati per la variabile dizionario mmspe() C.CONTO IND.DI SPED. 9. Situazione più complessa sarebbe invece, quella di agire con Sprix su una stampa di Passepartout.

Caso: Stampa dei totali di riga (per colore) e di colonna (per taglia)

Attenzione, questo esempio è un work around adottato per ottenere un determinato risultato, partendo da una struttura di codifica archivi impostata ed utilizzata in modo "personale". L'esempio, infatti, prende in considerazione gli ordini per determinati articoli di abbigliamento (che hanno due serie di tre tipi di taglie, ovvero la prima con "0", "3", "6" e la seconda con "2", "3", "4"), i quali articoli hanno tre tipi di colore (colore "1", colore "2", colore "3") e sono inseriti nella codifica anzichè esserne fuori oppure fare parte di un'apposita tabella di attributi.

Pur essendo un elemento statistico, il colore viene autocodificato, anzichè essere un semplice attributo di un articolo reale. La seguente immagine di struttura, identifica dove è stato posizionato il colore, ovvero al 13° carattere.

The screenshot shows a table with columns: Num, Descrizione, Autocodifica, Prog. a elem., SviluppoQta, InputQta. The table contains items 1 through 9. A dialog box titled 'PARAMETRI STRUTTURA' is open, showing various configuration options. The 'A posizione' field is highlighted with a red box and contains the value '20'. The 'Automatica' field contains 'C A T N Creazione'.

Di seguito il codice sorgente

```

IF DAORD=0 THEN DAORD=1 : IF X=0 THEN X=12 ELSE X=X
NUM=0 : TGEN=0:TG1=0:TG2=0:TG3=0 : TQCGEN=0:TQC1=0:TQC2=0:TQC3=0
G Scheda Produzione Articolo come da situazione al ~STRDATAS$(sxdat,2,"/")
G
G
or
or TG=VAL(RIGHT$(arum2(ortart()),1)) : NUM=NUM+1 : COD_BS=LEFT$(ortart(),X)
or IF COD_AS="" OR COD_AS=COD_BS THEN GOTO SALTA
or G TOT.TAGLIA ~TQC1 | ~TQC2 | ~TQC3 | | ~TQCGEN
or
or =====
or
or N
or
or TQCGEN=0 : TQC1=0 : TQC2=0 : TQC3=0
or
or SALTA:
or I G Articolo | Colore | Descrizione: ~ardes(ortart()+aragg(ortart())) | TOT.COLORE
or I N
or I N ~ortart() | Taglia | ~mtdes(TG,1) | ~mtdes(TG,2) | ~mtdes(TG,3) |
or I TGEN=0:TG1=0:TG2=0:TG3=0
or I
or
or COD_AS=LEFT$(ortart(),X)
or TQTA=ortag(1)()+ortag(2)()+ortag(3)() : TGEN=TGEN+TQTA
or TG1=TG1+ortag(1)():TG2=TG2+ortag(2)():TG3=TG3+ortag(3)()
or 'totali quantita' di riga
or T N |Quantita| ~TG1 | ~TG2 | ~TG3 | | ~TGEN
or T N
or T TQCGEN=TQCGEN+TGEN : TQC1=TQC1+TG1 : TQC2=TQC2+TG2 : TQC3=TQC3+TG3
or T
or
or G TOT.TAGLIA ~TQC1 | ~TQC2 | ~TQC3 | | ~TQCGEN
or
or =====
or
or N
    
```

Sono previsti anche i parametri di selezione, con possibilità di scelta fino a quale carattere

The 'Parametri di esecuzione' dialog box shows a table with the following data:

Nome	Parametri	Descrizione	Tipo	Ln	Dec
CODART\$	Dal Codice Art. Base		20 M		
CODART_AS	A1 Codice Art. Base		20 M		
X	Tot. fino carattere		2 0		

Il ciclotabellare riguarda le testate degli ordini, ha la seguente selezione:
 $CVAS(LEFT$(ortart(),X),X) \geq CVAS(LEFT$(CODART$,X),X)$ AND $CVAS(LEFT$(ortart(),X),X) \leq CVAS(LEFT$(CODART$,X),X)$ AND $ornum() \geq DAORD$ AND $orsig() = "OC"$ AND $ortip() = "3"$

The 'Tabella' dialog box shows a table with the following data:

Cod.tab	Ordine 3	Ordine 2	Ordine 1	Scelte
ORDINIMATRICI/PIVVENT. - TESTATA OC-OF-PR	ortart()			$CVAS(LEFT$(ortart(),X),X) \geq CVAS(LEFT$(CODART$,X),X)$ AND $CVAS(LEFT$(ortart(),X),X) \leq CVAS(LEFT$(CODART$,X),X)$

L'esecuzione prende un range di articoli senza indicare la posizione del carattere, in quanto il codice è impostato al 12° se non viene indicato

Il ciclo a estrapola gli ordini per colore e taglie ed esegue i totali di riga e colonna

Scheda Produzione Articolo come da situazione al 03/05/18

Articolo	Colore	Descrizione: CARDIGAN					TOT.COLORE
BA10799	1	Taglia	0	3	6	13	21
		Quantita		8			
Articolo	Colore	Descrizione: CARDIGAN					TOT.COLORE
BA10799	2	Taglia	0	3	6		
		Quantita					
Articolo	Colore	Descrizione: CARDIGAN					TOT.COLORE
BA10799	3	Taglia	0	3	6	1	1
		Quantita					
TOT.TAGLIA				8	14		22
Articolo	Colore	Descrizione: SCARPA					TOT.COLORE
BC10799	1	Taglia	2	3	4	1	2
		Quantita		1			
Articolo	Colore	Descrizione: SCARPA					TOT.COLORE
BC10799	3	Taglia	2	3	4	1	2
		Quantita		1			
TOT.TAGLIA				2	2		4

Anche una stampa standard poteva avvicinarsi a questo tipo di stampa

Caso: prezzo particolare al netto di Iva con listino Ivato

Salvo l'imputazione manuale del prezzo, il gestionale considera la particolarità prezzo con precedenza assoluta rispetto ad altri valori possibili; nel manuale di magazzino c'è una graduatoria di importanza. Il prezzo che viene proposto, se non si è a posteriori del calcolo del gestionale, in Sprix è necessario ricavarlo con un calcolo (non ci sono variabili che riportino tale prezzo). Date le seguenti condizioni:

- Listino Ivato (+I)
- PF=prezzo particolare fisso (fra Cliente e Articolo)
- A=aliquota (non in %)

Il Prezzo al netto dell'Iva (PN) si ottiene con la seguente formula $PN=PF/(1+(A/100))$. Ad es. prezzo 8,00 Euro ed Iva al 20%, il prezzo è $6,666666=8/1,20$.

Tenere presente che per l'arrotondamento dei decimali esiste una funzione "ARRO" con troncamento all'ultimo decimale di riferimento (che comunque non può essere oltre il 6"). Nell'esempio, essendo presente un risultato periodico, in caso di 2 decimali si potrebbe ottenere un risultato arrotondato $R=ARRO(PN,3,0.01)$ pari a 6,67.

ON_SMSGARTMM - finestra dettaglio articoli in emissione revisione documenti

L'evento ON_SMSGARTMM: si attiva quando entrando nel campo codice articolo di emissione revisione documenti si apre automaticamente la finestra di dettaglio.

Tale finestra non sempre appare ma l'utilizzo viene determinato se presenti informazioni aggiuntive.

La presenza di tale finestra può creare dei problemi se dei collage antecedenti, alla versione 2014E4, nel punto indicato aprono finestre proprie di programmazione, in quanto l'apertura di una finestra passepartout modifica la sequenza delle finestre aperte da collage non permettendone una gestione corretta, con il verificarsi dell'errore "... Impossibile chiudere la finestra. Routine collage interrotta".

L'evento nasce quindi per poter intervenire prima dell'apertura della finestra di passepartout e mantenere la sequenze delle finestre proprie inalterato.

Esempio:

ON_IN_MDCODART:

```
GETATTMM
IF APRI = 1 THEN CHIUDIFIN FINESTRA: APRI = 0
IF _MMART$(RIGA) <> "" AND LEFT$( _SINP$,1) <> "." AND APRI = 0 THEN GOSUB MESSAGGIO
END
ON_SMSGARTMM:
IF APRI = 1 THEN CHIUDIFIN FINESTRA: APRI = 0
END
```

```
MESSAGGIO:
FINESTRA=AUTOFIN()
APRIFIN FINESTRA,12,_NFIN,26,28,1,24:
MESS$="FINESTRA "+STR$(FINESTRA,0,0)
LOCATE 1,1:PRINT MESS$:
APRI = 1
RETURN
```

Caso: dizionario "gs" SCONTI/ABBUONI/GARANZIE variabili non valorizzate

Le variabili relative alla tabella "gs" (Magazzino \ Tabelle Aziendali \ Descrizione Tabelle \ Omaggi/Abbuoni/Garanzie), non vengono tutte valorizzate tramite una lettura diretta o da ciclo tabellare se non si accede alla tabella e la si conferma (tasto funzione [F10]).

Caso: totali di riga e documento, quali logiche venivano impiegate prima del rilascio delle apposite istruzioni di totalizzazione (RIGMM e TOTMM)?

Premessa importante: si rilascia a scopo esclusivamente didattico la seguente metodologia di calcolo che veniva utilizzata in passato per cocolare i totali di riga e documento. La seguente metodologia non è più utilizzabile nelle recenti versioni nel calcolo dei totali del documento di magazzino, poiché ci sono apposite istruzioni RIGMM e TOTMM che considerano elementi che in passato non erano ancora presenti, ma potrebbe offrire degli spunti di riflessione per elaborazioni in altri contesti o per altre esigenze di calcolo.

A parte le variabili dizionario di sola lettura (nelle tabelle dei movimenti di magazzino, ordini, preventivi, matrici) che comunque richiedono attenzione nella loro valutazione (si pensi ai differenti tipi di arrotondamento relative alle variabili dizionario mmruv() ed mmrdv()), in passato non esistevano delle variabili struttura archivi in grado di riportare i totali di riga o del documento, in quanto le condizioni di registrazione erano molteplici e non erano sintetizzabili nelle stesse variabili (si pensi alle tipologie di valuta, i decimali, gli arrotondamenti, gli sconti, le provvigioni, le particolarità, ecc.); si rendeva quindi necessario ricostruire i calcoli relativi ai totali di riga e totali del documento durante la sua emissione, a seconda delle varie esigenze.

Riportiamo, quindi, i criteri generali per i calcoli di riga e totali documento valide fino all'anno 2003, addirittura con la differenziazione, in termini di arrotondamenti, fra la gestione in Lire e la gestione in valuta diversa.

Di questi criteri ne anticipiamo i presupposti:

1) fissiamo alcune variabili relative ai decimali di riferimento

DECQTA = numero decimali con cui è gestita la quantità;
 DECPRZ = numero decimali con cui è gestito il prezzo di riga articolo nel documento;
 DECIMP = numero decimali con cui è gestito l'importo di riga ed i totali di documento;
 l'importo di riga è visualizzato nel documento in basso a destra)

2) ricordiamo i tipi di arrotondamento possibili:

E = arrotondamento per eccesso al numero di decimali indicato

es. numero decimali prezzo = 0
 numero = 12,50 E(12.50,DECPRZ) = 13
 numero = 12,49 E(12.49,DECPRZ) = 13

M = arrotondamento per difetto/eccesso dei 4/5 (da 1 a 4 per difetto, da 5 a 9 per eccesso) al numero di decimali indicato

es. numero decimali prezzo = 0
 numero = 12,50 M(12.50,DECPRZ) = 13
 numero = 12,49 M(12.49,DECPRZ) = 12

D = arrotondamento per difetto al numero di decimali indicato (troncamento)

es. numero decimali prezzo = 0
 numero = 12,50 D(12.50,DECPRZ) = 12
 numero = 12,49 D(12.49,DECPRZ) = 12

Dopo l'ultimo decimale di riferimento che è stato impostato, avviene un troncamento; tale operatività è una modalità di calcolo del gestionale, in quanto è richiesto al programma di poter lavorare tanto con i numeri grandi quanto con quelli piccoli. Il coefficiente di conversione (Koeff.) può contribuire a rendere difficile l'eventuale interpretazione dei progressivi, se i decimali nei prezzi non fossero coerenti al tipo di gestione in atto (i decimali in prezzo sono impostabili dal menu Servizi). Gli arrotondamenti sono influenzati anche dal rapporto fra il tipo di arrotondamento impostato sui listini ed il numero di decimali che sono stati impostati.

I decimali presenti nel corpo fattura possono essere più di due e nel totale della fattura al massimo due (come accadeva precedentemente con la gestione in Lire). L'importo sulla stampa della fattura non può essere diversa da quella proposta a video (quanto emesso deve corrispondere a quanto registrato): possibili differenze potrebbero imputarsi all'utilizzo di decimali, di valuta straniera, di VFOR in modulistica.

Arrotondamento nella determinazione dell'imposta: a partire dalla versione 12.0, il tipo di arrotondamento da applicare è in funzione della valuta di gestione dell'azienda (valuta in Lire oppure diversa da Lire):

a) con valuta = Lire, tipo di arrotondamento utilizzato nella determinazione dell'imposta è di tipo E (per eccesso al numero di decimali indicato);
 b) con valuta diversa da Lire, il tipo di arrotondamento utilizzato nella determinazione dell'imposta è di tipo M (eccesso/difetto al numero di decimali indicato. Dopo la conversione all'Euro, il tipo di arrotondamento utilizzato è = M;

Arrotondamento nello scorporo dell'imposta dal totale lordo (quantità * importo): a partire dalla versione 12.0, il tipo di arrotondamento da applicare è in funzione della valuta di gestione dell'azienda (valuta in Lire oppure diversa da Lire):

a) con valuta = Lire, il calcolo dell'imponibile netto avviene con arrotondamento di tipo D (per difetto al numero di decimali indicato); il calcolo dell'imposta è come differenza fra il totale lordo e l'imponibile netto.
 b) con valuta diversa da Lire, il calcolo dell'imponibile netto avviene con arrotondamento di tipo M (per eccesso/difetto al numero di decimali indicato); per il calcolo dell'imposta, viene utilizzato l'imponibile netto prima calcolato. Controllare che l'imponibile netto più l'imposta dia ancora il totale lordo: se il totale lordo è maggiore, viene ricalcolato l'imponibile netto come differenza tra totale lordo di partenza e imposta calcolata; se il totale lordo è minore, viene ricalcolato l'imposta come differenza tra totale lordo di partenza e imponibile netto calcolato

Gli stessi calcoli vengono utilizzati per determinare i valori della ritenuta d'acconto e della cassa professionisti.

A partire dalla versione 12.3, la determinazione del prezzo di riga articolo, dell'imponibile e loro elaborazioni nel documento di magazzino, utilizza un tipo di arrotondamento in base alla seguente logica: se i valori (prezzi ed imponibili) sono espressi in valuta Lire, viene utilizzato l'arrotondamento di tipo D; se i valori (prezzi ed imponibili) sono espressi in valuta diversa da Lire, viene utilizzato l'arrotondamento di tipo M (in caso di emissione di nuovi documenti, trasformati da un tipo ad un altro e accorpando due o più documenti in uno solo) oppure l'arrotondamento di tipo D (in caso di revisione o consultazione di documenti emessi con versioni precedenti alla 12.3).

NB: indichiamo con ARRO l'arrotondamento che segue la logica sopra riportata; ARRO può quindi essere di tipo D oppure M. L'ordine matrice segue la tipologia dell'arrotondamento utilizzato nell'ordine di riferimento.

CALCOLO DELLA QUANTITA'

QTA = quantità immessa nel documento
 COLLI = colli (immessi nel documento)
 TARA = tara articolo

Articoli movimentati a colli: $QTA_REALE = D(QTA * COLLI, DECQTA)$

Articoli a peso netto: $TARA_REALE = D(TARA * COLLI, 3)$

se $qta_reale > 0$: $QTA_REALE = D(QTA_REALE - TARA_REALE, DECQTA)$

se $qta_reale < 0$: $QTA_REALE = D(QTA_REALE - (-TARA_REALE), DECQTA)$

CALCOLO DEL PREZZO INDICATO NELLA RIGA ARTICOLO

Il prezzo indicato a video nel campo prezzo è quello fornito dell'eventuale particolarità, oppure quello calcolato a partire da quanto specificato nell'anagrafica articolo. Tale prezzo lo indichiamo con PRZ_REALE.

Logica con cui viene fornito dall'anagrafica articoli il prezzo dell'articolo senza DBA e che dipende dal tipo di documento che si sta emettendo:

- documenti BD e IN: prezzo medio di carico – prezzo medio d'inventario, relativi al magazzino del documento, costo ultimo – costo standard;
- documenti CL: in base al parametro di magazzino "Car.costo Prod.Fin (U/S/M/V/PL)"; se il parametro non è gestito, viene preso il costo standard;
- documenti DL e SL: in base al parametro di magazzino "Scar.costo Mat.Prime (U/S/M/V/P)"; se il parametro non è gestito: prezzo medio di carico - prezzo d'inventario, relativi al magazzino del documento, costo ultimo - costo standard;
- altri documenti d'acquisto intestati a un codice fornitore: prezzo d'acquisto per il fornitore – Costo ultimo – Costo standard;
- altri documenti d'acquisto non intestati (BS): se si gestiscono più listini 0, altrimenti il listino 1;
- altri documenti di vendita intestati ad un cliente: listino del cliente;
- altri documenti di vendita non intestati (CO RI FC BS PR PX): se si gestiscono più listini 0, altrimenti il listino 1.

CALCOLO DEL VALORE DELL'IMPORTO RIGA ARTICOLO DEL DOCUMENTO

Il valore dell'importo di riga (che viene calcolato ed visualizzato a video in basso a destra, gravato da eventuali sconti), lo chiamiamo VALORE_REALE ed è prodotto da:

$VALORE_REALE = ARRO(QTA_REALE * PRZ_REALE, DECIMP)$

CALCOLO DEGLI SCONTI SULLA RIGA ARTICOLO DEL DOCUMENTO

Si ricorda che il valore dello sconto presente a video segue la seguente priorità, dalla prima all'ultima la priorità è decrescente: particolarità sconto, sconto incondizionato del cliente ed infine, sconto dato dalle impostazioni dell'articolo. Si ricorda che questi automatismi scattano solo all'atto dell'inserimento dell'articolo, e se il campo viene modificato manualmente, quello inserito è prioritario. Definiamo i valori degli sconti:

SC100 (sconto merce):

? $VALORE_SC100 = VALORE_REALE$

SC110 (abbuono dell'imponibile):

? $SC110 = VALORE_REALE$

se il documento ha righe articolo con importo comprensivo di IVA, ALQ = aliq iva

? $VALORE_SC110 = SC110$ scorporato dell'imposta utilizzando ALQ e DECIMP

SC120 (abbuono imponibile e iva):

? $SC120 = VALORE_REALE$

se il documento ha righe articolo con importo NON comprensivo di IVA, ALQ = aliq iva

Y = imposta calcolata a partire da SC120 utilizzando ALQ e DECIMP

? $VALORE_SC120 = D(SC120 + Y, DECIMP)$

SC200 (garanzia):

? $VALORE_SC200 = VALORE_REALE$

? VALORE_REALE viene azzerato.

SCONTO (a valore):

X = Sconto a valore riferito alla quantità unitaria

? $VALORE_SCONTO = ARRO(QTA_REALE * X, DECIMP)$

SCONTO (in percentuale)

X = percentuale dello sconto

? $VALORE_SCONTO = ARRO(VALORE_REALE * X / 100, DECIMP)$

SCONTO (in cascata, al massimo 3 sconti):

? $XVAL = VALORE_REALE$

? $VALORE_SCONTO = 0$

Calcolo per documenti in lire o emessi con versione precedente la 12.3

Ciclo su max 3

Y = $D(XVAL * X / 100, DECIMP)$

? $VALORE_SCONTO = D(VALORE_SCONTO + Y, DECIMP)$

? $XVAL = D(XVAL - Y, DECIMP)$

Fine ciclo

Calcolo per documenti in valuta diversa da lire dalla versione 12.3 compresa

Operazioni da fare per ogni valore che compone lo sconto in cascata:

Y = $D((XVAL * X / 100), 5)$

VALORE_SCONTO = $D((VALORE_SCONTO + Y), 5)$

XVAL = $D((XVAL - Y), 5)$

In caso di arrotondamenti su sconti in cascata, nel primo sconto l'arrotondamento avviene anche a livello di decimali per difetto al 5° e l'ultimo per difetto/eccesso sul numero di DECIMP

Ultimato il calcolo dello sconto in cascata lo sconto finale è così calcolato:

$VALORE_SCONTO = M(VALORE_SCONTO, DECIMP)$

CALCOLO DELL'IMPORTO DI RIGA SCONTATO

? $VALORE_REALE = D(VALORE_REALE - VALORE_SCONTO, DECIMP)$

CALCOLO DELLA PROVVISGIONE

X = Percentuale provvigione

Se il VALORE_REALE comprensivo di Iva, scorporare l'Iva

? $VALORE_PROV = ARRO(VALORE_REALE * X / 100, DECIMP)$

CALCOLI RELATIVI ALLA FINESTRA DEI TOTALI DOCUMENTI

Pagamento con sconto merce (SCMERCE): è applicato sul VALORE_REALE dei singoli articoli immessi, ad esclusione di quelli di tipo S

X = percentuale dello sconto

VALORE_SCMERCE = $ARRO(VALORE_REALE * X / 100, DECIMP)$; per riga articolo

VALORE_REALE = $D(VALORE_REALE - VALORE_SCMERCE, DECIMP)$

VALORE_SCMERCE += $D(VALORE_SCMERCE, DECIMP)$

TOTALE_SCONTO += $D(TOTALE_SCONTO + VALORE_SCMERCE, DECIMP)$

Pagamento con sconto abbuono (SCABB): è applicato sul totale documento

X = percentuale dello sconto

VALORE_SCABB = $ARRO(TOTALE_DOCUMENTO * X / 100, DECIMP)$;

Il valore viene visualizzato nella finestra totali

Spese banca (SPBAN): spese per ricevute bancarie

X = Valore spese banca per singola rata

J = Numero delle rate impostate nel pagamento selezionato
 SPBAN = ARRO(J * X, DECIMP)
 L'importo delle spese viene ripartito in base alle aliquote I.V.A. presenti nel documento ed in proporzione all'importo merce per singola aliquota.

Tutte le somme per ottenere i totali utilizzano l'arrotondamento di tipo D.

CALCOLO PREZZO DI VENDITA ARTICOLO CON DBA

Il prezzo di vendita dell'articolo con dba si ottiene a partire dai componenti della dba visualizzati in emissione documenti. I componenti riportano i prezzi nella valuta del listino del articolo con dba. Il prezzo del componente viene convertito nella valuta di gestione azienda e moltiplicato per la sua quantità, il risultato viene totalizzato insieme a quello degli altri componenti. Il totale così ottenuto in valuta di gestione azienda, viene diviso per la quantità dell'articolo con dba per ottenere il prezzo di vendita, tale prezzo viene convertito da valuta azienda a valuta documento.

ANAGRAFICA ARTICOLI

Ricarico Listino prezzo articolo

Listino senza iva: viene aggiunta la percentuale di ricarico al costo impostato di partenza;

Listino con iva: per ottenere il prezzo di listino con iva, si parte dal costo impostato si aggiunge l'iva e successivamente viene aggiunta la percentuale di ricarico;

Casi: Personalizzazioni del file XML della fattura elettronica

Premessa: il file Xls di "Rappresentazione tabellare del tracciato fattura ordinaria" è consultabile al link <https://www.fatturapa.gov.it/export/fatturazione/it/normativa/f-2.htm>. si ricorda che la personalizzazione del file Xml è consentita anche tramite la programmazione dei campi del modulo di stampa. I campi prevedono delle IIF di condizione e possono esserne aggiunte ulteriori, ad esempio, per condizionare il contenuto dei campi "dati aggiuntivi" di ripo "2.2.1.3.1" (Codice tipo) e 2.2.1.3.2 (Codice valore).

Detto questo, la modifica/integrazione del file XML della fattura elettronica avviene tramite l'utilizzo dell'etichetta collage "ON_STP_XML".

Se il tag che si vuole gestire è presente nel file XML e tale tag può essere presente una sola volta, dichiarato nelle variabili, se presente sarà sostituito, se assente sarà inserito. Se il tag invece può essere ripetuto più volte nel file XML, dichiarato nelle variabili, sarà aggiunto.

_FTENODO\$(RIGA,100); _FTEVALNODO\$(RIGA,100), _FTEOPNODO\$(RIGA,100), queste variabili bidimensionali servono a raggruppare per nodo tutti i figli ad esso collegati posizionati su un elemento del vettore (RIGA), mentre il secondo elemento serve a specificare tutti i tag che devono essere racchiusi nell'elemento.

Nell'utilizzo dei tag della fattura elettronica vanno dichiarati i tag di ultimo livello che si vogliono gestire.

Per i formati "DATA" presenti all'interno del file XML devono essere dichiarati "AAAAMMGG" senza separatori.

Particolarità della variabile _FTERIGMM(ELEMENTO): tale variabile indica a quale riga della linea si vuole gestire l'informazione, per identificare la linea occorre contare le righe diverse dalle descrizioni, nel documento.

Esempio se la prima riga è un articolo e la seconda riga è descrittiva e la terza riga è un articolo e su questa si vogliono gestire le informazioni la variabile _FTERIGMM(ELEMENTO) sarà uguale a 2

_FTENUMELE identifica il numero del vettore caricato nella struttura.

Per inserimenti multipli riferiti alla stessa riga articolo tale valore deve essere incrementato per quanti nodi padre devono essere inseriti.

Caso di modifica di un Tag non ripetuto presente o meno nel documento XML

ON_STP_XML:

```
ELEMENTO = ELEMENTO +1 'incremento di un contatore riga della struttura
_FTENODO$( ELEMENTO,1)="1.1.1.1" 'nodo monovalore del documento XML corrisponde al tag <IdPaese> del padre <DatiTrasmissione>
_FTEVALNODO$( ELEMENTO,1)="IT" ' valore che si vuole attribuire al nodo della riga
_FTENODO$( ELEMENTO,2)="1.1.1.2" 'nodo monovalore del documento XML corrisponde al tag <IdCodice> del padre <DatiTrasmissione>
_FTEVALNODO$( ELEMENTO,2)="ABCDEFGHIJKLMNO" ' valore che si vuole attribuire al nodo della riga
_FTENUMELE = ELEMENTO ' numero dell'elemento caricato nella struttura.
END
```

Dettaglio partenza:

```
<DatiTrasmissione> [1.1]PADRE
<IdTrasmittente> [1.1.1]
<IdPaese>IT</IdPaese> [1.1.1.1]TAG paese (tag perchè chiuso dalla barra /) - valore del NODO
<IdCodice>23434123424</IdCodice></IdTrasmittente> [1.1.1.2]TAG codice (tag perchè chiuso dalla barra /) - valore del NODO
```

Dettaglio modifica:

```
<DatiTrasmissione>
<IdTrasmittente>
<IdPaese>IT</IdPaese>
<IdCodice>ABCDEFGHIJKLMNO</IdCodice></IdTrasmittente>
```

Caso di modifica di un tag valore ripetuto più volte nel documento XML

In questo esempio inseriamo per una riga numero del documento di magazzino, tipo riga articolo, due tag ripetuti due volte con due dati differenti. Caso tipico della GDO che vuole per ogni codice articolo altri due codici

ON_STP_XML:

```
GETATTMM
FOR X=1 TO _NMM
  IFF _MMTPR$(X) = "R"
    RIGA = RIGA +1
  ENDF
```

' stiamo **inserendo** un gruppo di tag per la stessa riga articolo del documento di magazzino

```
IFF _MMART$(X) = "01" AND _MMPRZ = 100
  ELEMENTO = ELEMENTO +1
  _FTENODO$( ELEMENTO,1)="2.2.1.3.1"
  _FTEVALNODO$( ELEMENTO,1)="EAN"
  _FTEOPNODO$( ELEMENTO,1)="I"
  _FTERIGMM(ELEMENTO) = RIGA
  _FTENODO$( ELEMENTO,2)="2.2.1.3.2"
  _FTEVALNODO$( ELEMENTO,2)=aralt(_MMART$( _NMM))
  _FTEOPNODO$( ELEMENTO,2)="I"
  _FTERIGMM(ELEMENTO) = RIGA
  _FTENUMELE = ELEMENTO

  ELEMENTO = ELEMENTO+1
  _FTENODO$( ELEMENTO,1)="2.2.1.3.1"
  _FTEVALNODO$( ELEMENTO,1)="Codice Uso Fornitore"
  _FTEOPNODO$( ELEMENTO,1)="I"
  _FTERIGMM(ELEMENTO) = RIGA
  _FTENODO$( ELEMENTO,2)="2.2.1.3.2"
```



```

_FTEVALNODOS( ELEMENTO,2)=afcof(_MMART$( _NMM),1)
_FTEOPNODOS( ELEMENTO,2)="I"
_FTERIGMM(ELEMENTO) = RIGA
_FTENUMELE = ELEMENTO
ENDF

```

' stiamo **variando** un'informazione per un codice articolo di magazzino

```

IFF _MMART$(X) = "01" AND _MMPRZ = 300
ELEMENTO = ELEMENTO +1
_FTENODOS( ELEMENTO,1)=" 2.2.1.14"
_FTEVALNODOS( ELEMENTO,1)="N6"
_FTEOPNODOS( ELEMENTO,1)="V"
_FTERIGMM(ELEMENTO) = RIGA
_FTENUMELE = ELEMENTO
ENDF
NEXT
END

```

Esempio di Modifica TAG (2.2.1.3.1) e (2.2.1.3.2).

```

ON_STP_XML:
ELEMENTO=ELEMENTO+1
_FTENODOS(ELEMENTO,1)="2.2.1.3.1"
_FTEVALNODOS(ELEMENTO,1)="pippo"
_FTEOPNODOS(ELEMENTO,1)="U"
_FTENODOS(ELEMENTO,2)="2.2.1.3.2"
_FTEVALNODOS(ELEMENTO,2)="pluto"
_FTEOPNODOS(ELEMENTO,2)="U"
_FTENUMELE=ELEMENTO
END

```

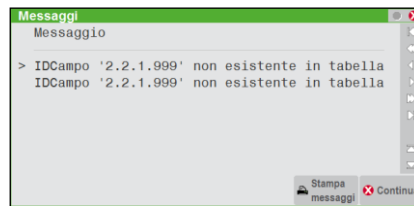
Risultato

```

<CodiceArticolo>
  <CodiceTipo>pippo</CodiceTipo>
  <CodiceValore>pluto</CodiceValore></CodiceArticolo>

```

Nel caso in cui si indicassero campi nodo non esistenti viene rilasciato apposito messaggio



Nota: La personalizzazione dei tracciati generali dei moduli periodicamente rilasciati da Passepartout (esempio fattura elettronica FTE) tramite le modifiche e/o l'aggiunta di altre informazioni non previste è incompatibile con il suddetto aggiornamento automatico, poiché richiederebbe sia la manutenzione della combinazione di una moltitudine di informazioni (per esempio le descrizioni lotti) sia la valutazione di opportunità circa la presenza di dati non graditi alle convenienze e/o privacy di tutti gli utenti. Il formato Xml consente di distinguere le personalizzazioni in colore diverso per riconoscerle in caso di eventuali e manuali aggiornamenti dei tracciati.

La personalizzazione tramite le variabili _FTE all'evento ON_STP_XML: non consente l'intervento sul medesimo Tag, anche se apparentemente sembra composto da più elementi (come ad esempio la presenza di più lotti spezzati relativi alla stessa riga articolo). Questo genere di intervento è consentito solo con la gestione dell'intero modulo tramite i comandi XML e che fra poco descriviamo con alcuni esempi. Le informazioni/espressioni sono nella sezione Lotti riga articolo del modulo di stampa utilizzato (es. FTE) e sono riportate nella SEZIONE <AltriDatiGestionali> ("2.2.1.16") e sono: nel tag <TipoDato> ("2.2.1.16.1") la stringa DATI LOTTO nel tag <RiferimentoTesto> ("2.2.1.16.2") il codice lotto e la relativa quantità (se valorizzata anche la Data di scadenza). Esempio FTE riporta il campo lotto: "<RiferimentoTesto>Id: "+STR\$(_mm_gidlot,-1,0)+IF(ladv(_mm_gidlot)="", "", " - Scadenza: "+STRDATAS\$(ladv(_mm_gidlot),3,"/"))+" - Qtà: "+STR\$(_mm_glqtumd,-1,0)+"</RiferimentoTesto>". Per non riportare il campo ID LOTTO disabilitare la sezione Lotto riga articolo del modulo.

Per ottenere modifiche sul medesimo tag che apparentemente sembra essere composto da più elementi, è possibile utilizzare la variabile handle del documento XML in considerazione, ovvero la variabile _FTE_DOCXML al fine di eseguirne le opportune modifiche con le istruzioni generali XML. La variabile _FTE_DOCXML consente infatti l'accesso da collage al file _XML della FTE. L'handle consente di avere un punto di partenza per esaminare tutto il file, scorrendo fino a trovare i nodi di interesse al fine di modificare e/o aggiungere elementi: quando il nodo viene trovato lo si può modificare (anche con l'aggiunta di altri elementi appartenenti a quella radice) oppure, se non c'è il nodo, lo si può aggiungere. Nell'eventuale coesistenza della gestione tramite le istruzioni XML con quella tramite le variabili _FTE, si consideri che la gestione dell'handle _FTE_DOCXML avviene prima di quella delle variabili _FTE.

Esempio 1:

```

ON_STP_XML:
XML=_FTE_DOCXML

'Ricerca del nodo radice
R=XMLGETROOT(XML)
IF R=0 THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo radice|" + _ERRXML$: GOTO ERR:

N_211 = XMLFINDELEM(R,"DatiGeneraliDocumento")
IF N_211=0 THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo 2.1.1 <DatiGeneraliDocumento>|" + _ERRXML$: GOTO ERR:

END

```

Esempio 2:

```

ON_STP_XML:
_INTER=0
XML=_FTE_DOCXML

'Ricerca del nodo radice
R=XMLGETROOT(XML)
IF R=0 THEN VIMSG 5, "ERRORE|Non c'è il nodo radice|" + _ERRXML$: GOTO ERR:

'Supponiamo di eseguire calcoli relativi alla RITENUTA e che si vogliono modificare/aggiungere in file XML
' .....calcoli.....

'Valorizzazione blocco 2.1.1.5 <DatiRitenuta>
' 2.1.1.5.1 <TipoRitenuta>
' 2.1.1.5.2 <ImportoRitenuta>
' 2.1.1.5.3 <AliquotaRitenuta>

```

2.1.1.5.4 <CausalePagamento>

```
'Ricerca nodo 2.1.1 <DatiGeneraliDocumento>
N_211=XMLFINDELEM(R,"DatiGeneraliDocumento")
IF N_211=0 THEN VIMSG 5, "ERRORE|Non c'e' il nodo 2.1.1 <DatiGeneraliDocumento>|"+_ERRXML$: GOTO ERR:

'Ricerca nodo 2.1.1.5 <DatiRitenuta>
N_2115=XMLFINDELEM(N_211,"DatiRitenuta")
IF N_2115<>0 THEN VIMSG 5, "ATTENZIONE|Il nodo 2.1.1.5 <DatiRitenuta> esiste gia!": GOTO END_INSERT_2115:

'Inserimento nodo 2.1.1.5 <DatiRitenuta>
INSERT_2115:

  N_2115=XMLADDELE(N_211,"DatiRitenuta",")
  IF N_2115=0 OR _ERRXML$<>" THEN VIMSG 5, "ERRORE|Impossibile aggiungere il nodo 2.1.1.5 <DatiRitenuta>|"+_ERRXML$: GOTO END_INSERT_2115:

  N_21151=XMLADDELE(N_2115, "TipoRitenuta", "RT01")
  IF N_21151=0 OR _ERRXML$<>" THEN VIMSG 5, "ERRORE|Impossibile aggiungere il nodo 2.1.1.5.1 <TipoRitenuta>|"+_ERRXML$: GOTO END_INSERT_2115:

  N_21152=XMLADDELE(N_2115, "ImportoRitenuta", STR$(IMP_RIT,2,0))
  IF N_21152=0 OR _ERRXML$<>" THEN VIMSG 5, "ERRORE|Impossibile aggiungere il nodo 2.1.1.5.2 <ImportoRitenuta>|"+_ERRXML$: GOTO END_INSERT_2115:

  N_21153=XMLADDELE(N_2115, "AliquotaRitenuta", ALIQ_RIT$)
  IF N_21153=0 OR _ERRXML$<>" THEN VIMSG 5, "ERRORE|Impossibile aggiungere il nodo 2.1.1.5.3 <AliquotaRitenuta>|"+_ERRXML$: GOTO END_INSERT_2115:

  N_21154=XMLADDELE(N_2115, "CausalePagamento", CAU_RIT$)
  IF N_21154=0 OR _ERRXML$<>" THEN VIMSG 5, "ERRORE|Impossibile aggiungere il nodo 2.1.1.5.4 <CausalePagamento>|"+_ERRXML$: GOTO END_INSERT_2115:

END_INSERT_2115:

'-----

'Variazione Totale Documento '2.1.1.9 <ImportoTotaleDocumento>
'N_2119=XMLFINDELEM(N_211,"ImportoTotaleDocumento")
'IF N_2119=0 THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo 2.1.1.9 <ImportoTotaleDocumento>"+_ERRXML$: GOTO ERR:

'INSERT_2119:
'  XMLADDELE(N_2115, "CausalePagamento", CAU_RIT$)
'  XMLSETNODVAL(N_2119, STR$(TOT_DOC,2,0))
'  IF N_21154=0 THEN VIMSG 5, "ERRORE|Impossibile aggiungere il nodo 2.1.1.5.4 <CausalePagamento>"+_ERRXML$: GOTO END_INSERT_2115:

'END_INSERT_2119:

'-----

'Valorizzazione e aggiunta nel dettaglio riga della sezione 2.2.1.13 <Ritenuta>

'Ricerca primo nodo 2.2.1 <DettaglioLinee>
N_221=XMLFINDELEM(R,"DettaglioLinee")
IF N_221<>0 THEN N_DTG_LINEE=N_221:
IF N_221=0 THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo 2.2.1 <DettaglioLinee>|"+_ERRXML$: GOTO ERR:

INSERT_221113:

'..... altre operazioni.....

  IFF RIG_RDA
  N_221113=XMLFINDELEM(N_DTG_LINEE,"Ritenuta")
  IFF N_221113<>0
  XMLSETNODVAL XMLGETCHILD(N_221113),"SI":
  IF _ERRXML$<>" THEN VIMSG 5, "ERRORE|Impossibile valorizzare il nodo 2.2.1.13 <Ritenuta>|"+_ERRXML$: ERR=1: ERR=1: CRD=_NMM+1:

  ELSEIF
  N_221113=XMLADDELE(N_DTG_LINEE, "Ritenuta", "SI")
  IF N_221113=0 OR _ERRXML$<>" THEN VIMSG 5, "ERRORE|RIG_ART: "+STR$(RIG_ART_XML,0,0)+"|RIG_DOC: "+STR$(RIG_DOC,0,0)+"|Impossibile aggiungere il
  nodo 2.2.1.13 <Ritenuta>|"+_ERRXML$: 'ERR=1: CRD=_NMM+1:
  ENDF
  ENDF

  IF ERR<>0 THEN GOTO END_INSERT_221113:

'Ricerca prossimo nodo 2.2.1 <DettaglioLinee>
WHILE_FIND_DTG_LINEE:
  N_221=XMLGETSIBL(N_DTG_LINEE):
  IF N_221<>0 THEN IF (XMLGETELTYPE(N_221)=1 AND XMLGETELNAM$(N_221)="DettaglioLinee") THEN
  N_DTG_LINEE=N_221: GOTO INSERT_221113: ELSE N_DTG_LINEE=N_221: GOTO WHILE_FIND_DTG_LINEE:
  IF N_221=0 OR _ERRXML$<>" THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo fratello 2.2.1 <DettaglioLinee>|"+_ERRXML$: ERR=1: CRD=_NMM+1:
  WEND_FIND_DTG_LINEE:

END_INSERT_221113:

'.....eccetera, eccetera.....

END_ON_STP_XML:
END
```

Esempio 3:

Data una fattura proveniente da due bolle con descrizioni, si desidera riportare tali descrizioni nei riferimenti alle bolle di ciascuna parte di fattura.

Notaire.

L'operazione può essere eseguita tramite un collage, considerando che in sprix, a differenza del mondo java dove ciò che si identifica tale è, le istruzioni che gestiscono XML sono delle primitive di ben più "basso livello", pertanto è importante una particolare attenzione nell'analisi semantica del documento XML, di conseguenza, può rendersi utile capire il tipo di elemento

che appartiene al nodo (esempio con XMLGETELTYPE) e se, effettivamente, questo nodo abbia dei "fratelli" su cui dover andare ad elaborare per non ricadere sempre nello stesso nodo a causa di un ciclo di codice non correttamente impostato.

```

ON_PUTMM:
  GETATTMM
  GOSUB SCREEN
END

ON_STP_XML:
  GOSUB STAMPA_XML
END

SCREEN:
'RIF VARIABILE CHE INTERCETTA L"**** DI PRESENZA RIFERIMENTI ESTERNI
RIF$=SCREEN$(0,4,31,1)
'INTERCETTOLA SIGLA DOCUMENTO MMSIG$
MMSIG$=SCREEN$(0,2,10,2)
'NUMERO DEL DOCUMENTO
MMNUM$=SCREEN$(0,4,12,6)
'SERIE DEL DOCUMENTO
SERIE=VAL(SCREEN$(0,4,9,2))
'NR RIGA
MMNRIG$=SCREEN$(0,7,1,4)
'CODICE CLIENTE
MMCLI$=SCREEN$(0,2,41,9)
RETURN

'-----
'Gestione personalizzazione fattura XML
'-----

STAMPA_XML:
XML=_FTE_DOCXML

'Ricerca del nodo radice
R=XMLGETROOT(XML)
IF R=0 THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo radice|" + _ERRXML$

N_218 = XMLFINDELEM(R,"DatiDDT")
IF N_218=0 THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo 2.1.8 <DatiDDT>|" + _ERRXML$

  FRASE$="NOTE NR."
  BC$=""
  DATABC$=""
  NRBC$=""

  FOR I=1 TO _NMM
    IFF (_MMTPRS(I)="D" AND INSTR(_MMDEE$(I),FRASE$)<>0)
      GOSUB RECUPERA_BC : GOSUB NODO_2_1_8

      N_218 = XMLFINDELEM(N_218,"DatiDDT")
      IF N_218=0 THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo 2.1.8 <DatiDDT>|" + _ERRXML$

      'Valorizza il nodo 2181 esistente o ne aggiunge uno nuovo
      N_2181=XMLFINDELEM(N_218,"NumeroDDT")
      IFF N_2181>0 '(ESISTE)
        TEST$=XMLGETNCONT$(XML,N_2181)
        'VIMSG 5,STR$(XMLGETELTYPE(N_218),0,0)
        XMLSETNODVAL XMLGETCHILD(N_2181),NRBC$
      ELSEF
        N_2181=XMLADDELE(N_218,"NumeroDDT",NRBC$)
        IF N_2181=0 THEN VIMSG 5, "ERRORE|Impossibile aggiungere il nodo 2.1.8.1 "+ _ERRXML$:
      ENDF

      'Valorizza il nodo 2182 esistente o ne aggiunge uno nuovo
      N_2182=XMLFINDELEM(N_218,"DataDDT")
      IFF N_2182>0 '(ESISTE)
        XMLSETNODVAL XMLGETCHILD(N_2182),DATABC$
      ELSEF
        N_2182=XMLADDELE(N_218,"DataDDT",DATABC$)
        IF N_2182=0 THEN VIMSG 5, "ERRORE|Impossibile aggiungere il nodo 2.1.8.2 "+ _ERRXML$:
      ENDF

      'RICERCA PROSSIMO NODO "FRATELLO"
      N_218=XMLGETSIBL(N_218):
      IF N_218=0 THEN VIMSG 5, "ERRORE|Impossibile trovare il nodo 2.1.8 <DatiDDT>|" + _ERRXML$

    ENDF
  NEXT

'RET = XMLDUMPFIL (XML, "<C:\TMP\FatXML_NEW.xml", "UTF-8")
'IF RET=0 OR _ERRXML$<>" THEN VIMSG 5, "Impossibile scrivere il file:|" + _ERRXML$:

RETURN

'INTECETTO DAL TESTO LA DATA E IL NR DELLA BC
RECUPERA_BC:
  DATA$=RIGHT$( _MMDEE$(I),8)
  DAPOS=LEN(FRASE$)+1
  APOS=INSTR(_MMDEE$(I),"DEL")
  NRBC$=MID$( _MMDEE$(I),DAPOS,APOS-DAPOS)
  DATABC$="20"+RIGHT$(DATA$,2)+MID$(DATA$,4,2)+LEFT$(DATA$,2)
RETURN

```

Altri esempi:

Ulteriori esempi (o frazioni di esempio) sono disponibili nel capitolo XML della sezione dei COMANDI PER ARGOMENTO.

Caso: nella "duplicazione" documento, avviene una perdita dell'allegato documentale di Docuvision dal documento origine oppure un trasferimento al documento generato?

La duplicazione del documento che comporti, in realtà, la generazione di un nuovo documento (per esempio la GET di un ordine e successiva PUT di preventivo) trasferisce l'allegato dal documento origine al documento generato. L'allegato non viene perso, ma semplicemente attribuito al nuovo documento. Tale trasferimento è corretto, in particolare poiché il Gestionale non sta duplicando la situazione, piuttosto ne crea una nuova (in quanto ha un'origine diversa) e pertanto il Gestionale tratta l'allegato Docuvision analogamente alle trasformazioni dei documenti portando l'allegato al documento nuovo. L'effetto che quindi ne deriva, pertanto, è un trasferimento e per ottenere ANCHE il mantenimento del medesimo allegato anche sul documento origine, occorre attribuirglielo nuovamente (con la conseguenza che due diversi documenti che non hanno alcun collegamento fra essi puntano al medesimo allegato. A

dimostrazione di quanto descritto, si allega esempio:

```
GETMM "OC31"
VIMSG 1, _ERRMM$

'Leggi Associazione in magazzino: <tipo>=1500
VAR_1$="OC;1;31;20210703"
GETASSOCDV 1500,VAR_1$,0
VIMSG 1, _ERRAD$
' Quale e' l'ID di associazione?
IDASSOC = _ADDVDID(1) : VIMSG 1, STR$(IDASSOC,0,0)

ASSOMIA=IDASSOC

_MMSIG$="PC"
PUTMM 0
VIMSG 1, _ERRMM$

PUTASSOCDV 1500,VAR_1$,ASSOMIA,""
VIMSG 1, _ERRAD$

END
```

Esempio trasformazione nativa documento OC in BC con evasione di tutte le righe ordine:

La trasformazione nativa documento si basa sulla variabile _MMSIGTRAS\$ per specificare il documento di destinazione e di variabili colli, qualità, quantità taglia, colli lotto, quantità lotto e quantità taglia lotto per specificare nei documenti creati eventuali evasioni o trasformazioni parziali.

```
SIGLA$="OC"
SIGLATRA$="BC"
SERIE=1
NUMERO=22
'DOCUMENTO$="OC1/10+OC1/15+OC1/19"
DOCUMENTO$=SIGLA$+STR$(SERIE,0,0)+"/"+STR$(NUMERO,0,0)
GETMM DOCUMENTO$
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$:END
_MMSIGTRAS$=SIGLATRA$
_MMNUM=0
_MMSER=1
_MMDAT$=sxdat
IF _NMM<1 THEN VIMSG 1,"IL DOCUMENTO NON CONTIENE RIGHE" :END
FOR R=1 TO _NMM
  IF _MMTPR$(R)="R" THEN _MMORD$(R)="E" 'evasione totale di riga
NEXT
PUTMM 2
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Esempio trasformazione nativa documento OC in BC con evasione parziale di riga ordine articolo con taglie:

La situazione di partenza dell'esempio contempla un ordine che contiene in riga 1 un articolo a taglie (evadiamo taglie 1,2 e lasciamo in ordine le altre).

```
SIGLA$="OC"
SIGLATRA$="BC"
SERIE=1
NUMERO=40
DOCUMENTO$=SIGLA$+STR$(SERIE,0,0)+"/"+STR$(NUMERO,0,0)
GETMM DOCUMENTO$
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$:END
_MMSIGTRAS$=SIGLATRA$
_MMNUM=0
_MMSER=1
_MMDAT$=sxdat
_MMQTAR(1)=2*10 'qta evase riga, equivale a 2 taglie x qta 10 per taglia
_MMTAGR(1,1)=10 : _MMTAGR(1,2)=10 'qta evase per taglia
PUTMM 2
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Esempio trasformazione nativa documento OC in BC con evasione parziale di riga ordine articolo con lotti (variabili di gestione base/estesa):

```
SIGLA$="OC"
SIGLATRA$="BC"
SERIE=1
NUMERO=41
DOCUMENTO$=SIGLA$+STR$(SERIE,0,0)+"/"+STR$(NUMERO,0,0)
GETMM DOCUMENTO$
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$:END
_MMSIGTRAS$=SIGLATRA$
_MMNUM=0
_MMSER=1
_MMDAT$=sxdat
_MMQTAR(1)=50 'qta evase riga
_MMLQTAR(1,1)=50 'assegno qta evasa al dato di riga del lotto
PUTMM 2
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Esempio trasformazione nativa documento OC in BC con evasione parziale di riga ordine articolo con lotti (variabili di gestione estesa/matricole):

```
SIGLA$="OC"
SIGLATRA$="BC"
SERIE=1
NUMERO=27
DOCUMENTO$=SIGLA$+STR$(SERIE,0,0)+"/"+STR$(NUMERO,0,0)
GETMM DOCUMENTO$
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$:END
_MMSIGTRAS$=SIGLATRA$
_MMNUM=0
_MMSER=1
_MMDAT$=sxdat
_MMQTAR(1)=50 'qta parziale evasione di riga
_MML_LQTAR(1)=50 '50 pz del lotto 1
PUTMM 2
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Esempio trasformazione nativa documento BC in FT:

```
SIGLATRAS="FT"
DOCUMENTO$="BC1/2+BC1/3"
GETMM DOCUMENTO$
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$:END
_MM$IGTRAS$=SIGLATRAS$
_MMNUM=0
_MM$SER=1
_MM$DAT$=sxdat
PUTMM 3
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Esempio di scrittura documento di magazzino, registrazione di primanota e CDG:

```
_MMSIG$="FT"
_MM$SER=1
_MM$DAT$=sxsdt
_MM$MAG=1
_MM$CLIS$="501.10"
_MM$DTANL$="" "DATA COMPETENZA"
_MM$COMMANL$="00000001" "COMMESSA"
_MM$AREANL$="0002" "AREA"
_MM$TPR$(1)="R"
_MM$ART$(1)="00000001"
_MM$QTA(1)=5
_MM$PRZ(1)=111
_MM$ALIS(1)="22"
PUTMM 15
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Esempio di scrittura documento di magazzino, registrazione di primanota, movimento bene usato e CDG:

```
_MMSIG$="FT"
_MM$SER=1
_MM$DAT$=sxsdt
_MM$MAG=1
_MM$CLIS$="501.10"
_MM$DTANL$="" "DATA COMPETENZA"
_MM$COMMANL$="00000001" "COMMESSA"
_MM$AREANL$="0002" "AREA"
_MM$TPR$(1)="R"
_MM$ART$(1)="00000001"
_MM$QTA(1)=5
_MM$PRZ(1)=111
_MM$ALIS(1)="U22"
PUTMM 16
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Esempio di cancellazione di documento di magazzino, registrazione primanota, movimento bene usato e CDG

```
DELMM_EXT "FT1","501.10",16
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$
END
```

Esempio GETMM documenti tramite variabile _MMRICDOC\$()

```
_MMRICDOC$(1)="BC" 'le variabili vengono utilizzate per creare un'unica stringa con lo scopo di superare il limite della stringa parametro dell'istruzione
_MMRICDOC$(2)="260+"
_MMRICDOC$(3)="BC261+BC262"
_MMRICDOC$(4)="BC263"
GETMM ""
IF _ERRMM<>0 THEN VIMSG 1,_ERRMM$:END
VIMSG 1,_NTMM: "+STR$(_NTMM,0)+"|_NMM: "+STR$(_NMM,0,0)
END
```

Esempio di conservazione dell'informazione relativa al collegamento di riga OF>OC.

```
' Leggo l'OF, lo trasformo in movimento e lo salvo
GETMM "OF1/235"
_MM$SIG$ = "BF"
_MM$NUM = 1235
PUTMM 0 'Salvo il movimento
IF _ERRMM <> 0 THEN VIMSG 1, _ERRMM$:END
' Leggo l'OC collegato, e modifico le variabili legate all'ordine collegato
GETMM "OC1/2704"

' Per ogni riga dell'OC
FOR R = 1 TO _NMM
' Popolo le variabili legate all'OF precedentemente collegato
_MM$TDOCP$(R) = "OF" ' Tipo documento precedentemente collegato alla riga (fisso a "OF")
_MM$SOF(R) = _MMSOF(R) ' Serie documento OF precedentemente collegato alla riga
_MM$NOFP(R) = _MMNOF(R) ' Numero documento OF precedentemente collegato alla riga

' Azzero le variabili legate all'OF attualmente collegato
_MM$ORD$(R) = "E" ' Stato riga OC: Evadibile
_MM$SOF(R) = 0 ' Serie documento OF attualmente collegato (azzerata)
_MM$NOF(R) = 0 ' Numero documento OF attualmente collegato (azzerato)
_MM$NCRIGAOF(R) = 100001 ' Numero creazione riga documento OF attualmente collegato (valore che identifica il collegamento precedente)
NEXT

' Salvo l'OC
PUTMM 0
IF _ERRMM <> 0 THEN VIMSG 1, _ERRMM$

' Elimino l'OF d'origine
DELMM "OF1/235"
IF _ERRMM <> 0 THEN VIMSG 1, _ERRMM$
```

Caso: utilizzo _MMCMR() / _MMCAURIGA()

Prima del rilascio della versione 2024C la variabile _MMCMR() veniva utilizzata per contenere 2 distinti valori a seconda che la tabella delle contropartite, presenti in Magazzino > Tabelle generali > Contropartite (Alt+MGT), fosse stata compilata o meno. Se la tabella non era compilata la variabile veniva utilizzata per contenere il dato della causale di riga, diversamente per la contropartita di riga.

A partire dalla versione 2024C, con l'introduzione della variabile `_MMCAURIGA()` destinata a contenere in modo esclusivo l'eventuale causale di riga, si possono verificare i seguenti casi:

- se la tabella delle contropartite non è mai stata compilata, e con lo scopo di mantenere un comportamento coerente con il passato per le procedure che ne fanno uso, `_MMCMR(R)` e `_MMCAURIGA(R)` ritornano lo stesso valore facente riferimento alla causale;
- se la tabella delle contropartite era stata popolata precedentemente all'aggiornamento, la variabile `_MMCMR(R)` continuerà a restituire il valore della contropartita di riga e `_MMCAURIGA(R)` potrà essere utilizzata per le contropartite;
- una volta che la tabella delle contropartite precedentemente non compilata viene compilata una prima volta, anche se eventualmente svuotata in un momento seguente, `_MMCMR(R)` ritornerà sempre ed esclusivamente il valore della contropartita e `_MMCAURIGA(R)` quello della causale, facendo sì che il comportamento diverga per non coincidere mai più.

Riguardo la variabile dizionario `mmcmr`, che da sempre ritorna il dato di riga e qualora non presente quello di testata, dopo l'aggiornamento alla versione 2024C:

- nel caso che la tabella delle contropartite non è compilata ritornerà il contenuto della nuova variabile della causale di riga o del dato di testata della causale;
- dopo la prima compilazione della tabella delle contropartite, quando verrà accesa in modo definitivo la nuova gestione, conterrà sempre e solo il dato della contropartita di riga o di testata.

INDIRIZZI DI SPEDIZIONE (DA)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 12	Istruzione che azzerà variabili indirizzi di spedizione _DA	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "IND"	Istruzione che azzerà variabili struttura _DA (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETIND	Par: <codice\$>, [codice_conto\$] Istruzione che legge anagrafica indirizzi di spedizione Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sm/Sh
GETATTIND	Istruzione che legge i dati degli Indirizzi di Spedizione presenti a video	C
PUTIND	Istruzione che scrive anagrafica indirizzi di spedizione Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sm/Sh
ERRDA	Codice risultato operazioni su indirizzi di spedizione	S/Sm/Cd
_ERRDA\$	Messaggio operazioni su indirizzi di spedizione	S/Sm/Cd
DA	Variabili di struttura indirizzi di spedizione	S/Sm/Cd
ON_STARTINDSPED:	Evento all'ingresso nella voce di menu (MI)	C/Cd
ON_PUTINDSPED:	Evento alla scrittura/conferma nella voce di menu (MI)	C/Cd
ON_DELINDSPED:	Evento nella voce di menu (MI) non è attivo poiché c'è solo cancellazione logica (che è un PUT di trasformazione)	C/Cd
ON_ENDINDSPED:	Evento all'uscita dalla voce di menu (MI)	C/Cd
esempi e casi dizionario		
<p>Esempio: registra un Anagrafica Indirizzo di Spedizione e poi la aggiorna</p> <pre> _DACIN\$="AUTO" : _DACTO\$="201.00002" _DADES\$="descrizione indirizzo":_DANAZ\$="I":_DAPAE\$="IT" PUTIND : IF _ERRDA<>0 THEN VIMSG 1, _ERRDA\$ CODIND\$= _DACIN\$ IF _ERRDA=0 THEN VIMSG 1, "dopo la scrittura aggiorno la descrizione" AZZVARSYS 12: GETIND CODIND\$,": IF _ERRDA<>0 THEN VIMSG 1, _ERRDA\$ _DADES\$="NUOVA Descrizione" PUTIND : IF _ERRDA<>0 THEN VIMSG 1, _ERRDA\$ END </pre>		

ANAGRAFICA CONTATTI (DC)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 13	Istruzione che azzerava variabili anagrafica contatti _DC	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "CON"	Istruzione che azzerava variabili struttura _DC (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETCON	Par: <codice\$>. Istruzione che legge anagrafica contatti Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
GETATTCON	Istruzione che legge anagrafica contatti a video	C
PUTCON	Istruzione che scrive anagrafica contatti Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELCON	Par: <codice\$>. Istruzione che cancella anagrafica contatti Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
_ERRDC	Variabile codice risultato operazioni anagrafica contatti	S/Cd
_ERRDC\$	Variabile messaggio operazioni su anagrafica contatti	S/Cd
_DC	Variabili di struttura anagrafica contatti	S/Cd
ON_STARTCONT:	Evento relativo all'accesso nella voce di menu (MR)	C/Cd
ON_PUTCONT:	Evento relativo alla scrittura/conferma nella voce di menu (MR)	C/Cd
ON_DELCONT:	Evento relativo alla cancellazione nella voce di menu (MR)	C/Cd
ON_ENDCONT:	Evento relativo all'uscita dalla voce di menu (MR)	C/Cd
esempi e casi dizionario		
<p>Esempio: registra un Anagrafica Contatto e lo duplica modificandone il nome di ricerca e infine cancella il vecchio contatto</p> <pre> _DCCTO\$="AUTO" : _DCNAZ\$="IT" : _DCDES\$="Descrizione Contatto" _DCRIC\$="Nome di Ricerca" PUTCON : IF _ERRDC<>0 THEN VIMSG 4, _ERRDC\$ CODCONT\$=_DCCTO\$: AZZVARSYS 13 VIMSG 1, "duplica il contatto e lo modifica poi cancella l'originario" GETCON CODCONT\$: IF _ERRDC<>0 THEN VIMSG 4, _ERRDC\$ _DCCTO\$="AUTO" : _DCNAZ\$="IT" : _DCDES\$="Contatto Duplicato" _DCRIC\$="NUOVO NOME DI RICERCA" PUTCON : IF _ERRDC<>0 THEN VIMSG 4, _ERRDC\$ DELCON CODCONT\$: IF _ERRDC<>0 THEN VIMSG 4, _ERRDC\$ END </pre>		

ANAGRAFICA LOTTI (LOT)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 17	Istruzione che azzerà variabili anagrafica lotti _LOT	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "LOT"	Istruzione che azzerà variabili struttura _LOT (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETLOT	Par: <idlot>. Istruzione che legge anagrafica lotti Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
GETMULTILOT	Par: <[TipoLotto],[Codart]>,[TipoCodUt],[CodUtente],<Includi>,<Mag> Istruzione di lettura mirata anagrafica lotti (v. variabili _LOT_M) Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
PUTLOT	Istruzione che scrive anagrafica lotti Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELLOT	Par: <idlot>. Istruzione che cancella anagrafica lotti Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
_ERRLOT	Codice risultato operazione anagrafica lotti	S/Cd
_ERRLOTS	Messaggio operazione su anagrafica lotti	S/Cd
_LOT	Variabili struttura anagrafica lotti	S/Cd
_LOT_M	Variabili struttura anagrafica lotti per GETMULTILOT	S/Cd

esempi e casi dizionario

GETLOT

GETLOT "codice lotto" valorizza tutte le variabili _LOT??????.

_NDULOT=N, ritorna il numero dei campi presenti nei dati personalizzati, che sono stati dichiarati nel tipo lotto.
_LOTCTPL\$ contengono il tipo lotto del codice lotto letto.
_LOTDUCD\$(1-n) contiene il codice campo dato utente personalizzato.
_LOTDUTP\$(1-n) contiene il tipo dato utente personalizzato.
_LOTDUVL\$(1-n) contiene il valore dato utente personalizzato.

Esempio

```
GETLOT 1
_LOTCTPL$="aa"
_LOTCART$="CODICE ARTICOLO"
_LOTDTCA$="Data di carico"

_LOTDUCD$(1)="a"
_LOTDUTP$(1)="N"
_LOTDUVL$(1)="1500"

_LOTDUCD$(2)="b"
_LOTDUTP$(2)="A"
_LOTDUVL$(2)="Fornitore Italia"
```

Per conoscere il nome dei campi delle videate personalizzate tramite il comando GETTLOT _LOTCTPL\$ si hanno a disposizione tutte le variabili _TLOT??? che permettono di gestire le informazioni.

Esempio: Scrive lotto per articolo "04M-MI", lo legge e blocca e lo rilegge con GETMULTILOT. Infine lo cancella.

```
_LOTSTAT$="C" : _LTCART$="04M-MI" : _LOTCTPL$="ab"
_LOTCFOR$="401.00001" : _LOTDTCA$="20210101" : _LOTDTVA$="20210101"
_LOTDUCD$(1)="xx" : _LOTDUTP$(1)="A" : _LOTDUVL$(1)="For.1"
_LOTDUCD$(2)="yy" : _LOTDUTP$(2)="A" : _LOTDUVL$(2)="Cli.2"
_LOTDUCD$(3)="zz" : _LOTDUTP$(3)="A" : _LOTDUVL$(3)="Serie 3"
```

```
PUTLOT: IF _ERRLOT<>0 THEN VIMSG 4, _ERRLOTS
VIMSG 1, "Ha registrato il lotto "+STR$( _LOTID,0,0)+" per "+_LTCART$
IDLOT= _LOTID
AZZVARSYS 17
```

```
GETLOT IDLOT : IF _ERRLOT<>0 THEN VIMSG 1, _ERRLOTS
_LOTSTAT$="B": LOTMMSG$="messaggio per il blocco"
PUTLOT: IF _ERRLOT<>0 THEN VIMSG 4, _ERRLOTS
VIMSG 1, "Ha bloccato il lotto "+STR$( _LOTID,0,0)+" di "+_LTCART$
```

```
GETMULTILOT "", "04M-MI", 0, "", 1, 1 : IF _ERRLOT<>0 THEN VIMSG 1, _ERRLOTS
FOR I=1 TO _LOTNID_M
PRINT "Id lotto "+STR$( _LOTID_M(I),0,0)
PRINT "Codice Tipo lotto "+ _LOTCTPL_M$(I)
PRINT "Codice Articolo "+ _LTCART_M$(I)
PRINT "Codice Utente lotto "+ _LOTCTE_M$(I)
NEXT
VIMSG 1, "legge con GETMULTILOT, confermando il messaggio cancella"
DELLOT IDLOT : IF _ERRLOT<>0 THEN VIMSG 1, _ERRLOTS : END
```

Particolarità relativa a variabili tabellari di anagrafica lotti ("lq", "le", "lc")

Il campo "lb001", nella lettura ad accesso diretto, ritorna il codice del primo campo dei dati aggiuntivi. Es:

```
"aaa" "Primo campo"
"aab" "Secondo campo"
"aac" "Terzo campo"
```

lb001(1) ritornerà sempre "aaa". Nella lettura sequenziale, verranno riportate 3 righe in cui conterrà di volta in volta il valore corretto.

Caso: variabile di dizionario contenente il valore dei campi utente all'interno dell'anagrafica lotto

Affinché sia possibile leggere il valore è necessario che il codice campo sia di 3 caratteri.

Le variabili di dizionario, che riportano il valore dei campi utente all'interno della anagrafica lotto, le trova sotto la voce di dizionario "Videate/Dati Aggiuntivi Lotto" e sono:
"lc000" -> ID LOTTO

```
"lc001" -> CODICE CAMPO
"lc002" -> VALORE CAMPO ALFANUMERICO
"lc003" -> VALORE CAMPO NUMERICO
```

per poter risalire al valore specificato nell'Anagrafica Lotto di:
 - tipo alfanumerico assegnato al campo, ad esempio "aa1", la variabile da valorizzare è la lc002(Id lotto,"aa1");
 - tipo numerico la variabile da utilizzare è la lc003(id lotto,"aa1").

Le riporto parte di manuale "Anagrafica Tipi Lotti":

"COD. - Codice del campo composto da quattro caratteri di cui i primi due, impostati dal programma, sono derivati dal codice tipo lotto. I rimanenti due caratteri sono a discrezione dell'utente. In particolare i codici che hanno il quarto carattere vuoto sono gestibili da modulistica documenti e Sprix, mentre quelli che hanno anche il quarto carattere sono gestiti solo in anagrafica lotto."

Esempio. Supponendo che il tipo lotto prevede un campo senza chiave "afr" (insieme ad altr idue campi con chiave), la possibilità di visualizzarlo in anagrafica articolo (entrando in magazzino [F7] e selezionando Lotti [F6]), senza dover passare dalla rintracciabilità (il seguente codice è a titolo di esempio su un lotto, e va riadattato nel caso si decidesse di gestire lo scorrimento di riga col cursore).

```
ON IN_MLSCLEOTTO:
  ID=VAL(SCREEN$(-3,4,5,7))
  FOR I=1 TO 3
    A$=lc002(ID,lucod(latip(ID),I)) ' se non conosciuto il codice campo occorre passare dal tipo lotto
    ' A$=lc002(ID,"afr") se conosciuto il codice (tre caratteri)
    VIMSG 1, A$
  NEXT
END
```

Caso: _ERRLOT\$="Codice lotto non valido o non trovato (lotto n)"

LA PUTLOT effettua autocodifica del lotto. Se la PUTLOT è effettuata con _LOTID valorizzato rilascia il seguente errore, poichè deve avvenire una GETLOT che, se effettuata su un'azienda in cui non trova corrispondenza di lotto, non lo trova.

Caso: _LOTTCCA\$ modifica del controllo del carico lotti già movimentati

La variabile _LOTTCCA\$ è relativa a nuove imputazioni, ma per modificare il controllo del carico di lotti già movimentati (per esempio da "documento" a "fornitore") è necessario impostare la variabile _LOTTCCANEWS

Cancellazione del lotto

Il lotto non è cancellabile se è stato movimentato. La cancellazione del Lotto necessita di scollegamento dalle righe movimentate, pertanto occorre movimentare la riga senza lotto. Se la GETMM incontra limiti di dimensione del documento (es. per array fuori dai limiti) non c'è modo di potere selezionare i documenti a cui cancellare il lotto dalla riga. Occorre procedere manualmente sui documenti fino ad azzerare i movimenti dei lotti per poi cancellare il lotto e infine riorganizzare gli archivi, oppure, in alternativa c'è una procedura simile (da testare e verificare poiché non è ufficialmente riconosciuta) e che consiste nei seguenti passaggi:

- 1) esportare in CSV i documenti di interesse (selezionabili in vari modi, ma non per lotto, quindi trovate altro che li accomuni)
- 2) azzerare il file molo_dem.csv
- 3) selezionare e cancellare i documenti a blocchi da emissione revisione documenti
- 4) importare ex-novo i documenti da CSV
- 5) cancellare il lotto
- 6) riorganizzare

Gestione storico archivio lotti da Sprix

A partire dalla versione 2024A (v860) è stato introdotto l'archivio storico dei lotti. In seguito a questa introduzione si riportano i comportamenti delle varie istruzioni Sprix riguardo ai dati e le nuove variabili introdotte.

Le istruzioni GETLOT e DELLOT continueranno a funzionare come prima: le istruzioni operano sia sull'archivio corrente che quello storico in modo trasparente. Restano valide le limitazioni in cancellazione dei lotti movimentati anche per quelli inseriti nello storico.

L'istruzione PUTLOT non può scrivere o aggiornare dati nell'archivio dei lotti storicizzati.

L'istruzione GETMULTILOTT viene estesa con quattro nuovi valori del parametro "Includi":

- 2) = come parametro 0 ma solo dei dati nello storico;
- 3) = come parametro 1 ma solo dei dati nello storico;
- 4) = come parametro 0 ma dei dati di corrente+storico;
- 5) = come parametro 1 ma dei dati di corrente+storico;

L'istruzione GETCODART è stata estesa con un nuovo parametro, "O", per la ricerca degli articoli con lotto associato nell'archivio storico.

Riguardo le tabelle dizionario sono state introdotte le tabelle "ld" e "lf" come replica delle tabelle "la" e "lb", mentre per l'accesso diretto è stata introdotta la tabella "lh" come replica della tabella "lc". Contrariamente al ciclo tabellare, dove i dati sono separati per tipo archivio corrente/storico, l'accesso diretto da dizionario ai dati storicizzati può essere effettuato anche utilizzando i membri della tabella dei dati correnti.

Sono state aggiunte le variabili flag di sola lettura _LOTSTOR\$ e _LOTSTOR_M\$() che specificano se il lotto è stato storicizzato.

PARTICOLARITA' (PART)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 7	Istruzione che azzerà variabili particolarità _PART	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "PART"	Istruzione che azzerà variabili struttura _PART (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETPART	Par: <tpart\$>,<codcon\$>,<catcon>,<codart\$>,<catart>,<dtvalda\$> Istruzione che legge particolarità	S/Cr/Cd/Sm/Sh
GETPARTICOL	Par: <tpart\$>,<codcon\$>,<catcon>,<codart\$>,<catart>,<dtvalda\$>,[mod] Istruzione che legge particolarità con inizio ricerca Non consentita nel Livello di sviluppo (F2)	S/C/Cr/Cd/Sm/Sh
GETATTPART	Istruzione che legge particolarità a video	C
PUTPART	Istruzione che scrive particolarità	S/Cr/Cd/Sm/Sh
DELPART	Par: <tpart\$>,<codcon\$>,<catcon>,<codart\$>,<catart>,<dtvalda\$> Istruzione che cancella particolarità	S/Cr/Cd/Sm/Sh
_ERRPART	Codice risultato operazione particolarità	S/C/Cd
_ERRPART\$	Messaggio operazioni su particolarità	S/C/Cd
_PART	Variabili di struttura particolarità	S/C/Cd
ON_STARTTZ:	Eventi nella gestione particolarità prezzo	C/Cd
ON_DELTZ:		
ON_PUTTZ:		
ON_ENDTZ:		

esempi e casi dizionario

Caso: GETATTPART in Anagrafica Articoli o Rubrica Clienti Fornitori

L'istruzione GETATTPART valorizza le variabili _PART? con il contenuto a video delle particolarità e non è in grado di predeterminare se un articolo (oppure un cliente) abbia o meno una o più particolarità. Le variabili _PART? richiedono quattro parametri per individuare una specifica particolarità (Tipo particolarità; Codice/Categoria articolo; Codice/Categoria conto; Data da).

L'utilizzo di GETATTPART nella videata "Anagrafica Articoli" opera fra l'etichetta ON_STARTAR e ON_ENDAR, tuttavia, le variabili _PART? restano vuote finché non si specificano i quattro parametri aprendo la videata delle particolarità [F1] ed entrando nel corpo del "Tipo di particolarità" [F5]. Questo è il modo per determinare se esistono delle particolarità per il tipo di particolarità scelto.

```
ON 612 ALL:
GETATTPART
VIMSG 1, PARTTIP$+""+ _PARTCLI$+""+ _PARTART$+""+ _PARTDDT$
FOR I=1 TO 3
VIMSG 2, STR$( _PARTFIN(I),0,0)+"/"/"/" + _PARTPRZ$(I)
NEXT
END
```

Per determinare una singola particolarità o l'elenco di particolarità di un dato articolo o un dato cliente, valutare l'esecuzione di uno Sprix tramite SHELLSPRIX che esegua l'istruzione GETPARTICOL oppure che esegua un ciclotabellare (tabella "tz") salvando le info su file.

Esempio: scrive particolarità Prezzo a Quantità del Cliente, legge e ne cancella una e ne varia un'altra Infine, legge con GETPARTICOL.

```
AZZVARSYS 7 : _PARTTIP$="P"
_PARTCLI$="201.00028" : _PARTCAC=0 : _PARTART$="ART001" : _PARTCAA=0
_PARTDDT$="20210315" : _PARTADT$="20211231" : _PARTVAL=2 : _PARTPF$="Q"
FOR I=1 TO 3 : _PARTFIN(I)=I : _PARTPRZ$(I)=STR$(I*33,0,0) : NEXT
_PARTPRZ$(4)="100" : PUTPART : IF _ERRPART<>0 THEN VIMSG 4, _ERRPART$ : END
```

```
-----
Tipo particolarità.(P/S/V/A): P Prezzi
Cod.conto...: 201.00028 CNC S.r.l. CANTIERI NAVALI CLAS
Listino ven.: ListCTO: 1 UNO
Cod.articolo: ART001 Articolo 001
Cat.form.prz:
Validità da data..: 15/03/2021
-----
```

```
Fino a data: 31/12/2021 Val.: 2 EUR EURO
Fino a: Q Quantità Prezzi
1,000 33,00
2,000 66,00
3,000 99,00
oltre 100,00
```

Esempio: legge Particolarità Prezzo a Quantità del Cliente e visualizza il valore del terzo scaglione. Cancella la particolarità, la reinserisce e legge con GETPARTICOL indicando solo il conto del Cliente

```
GETPART "P","201.00028",0,"ART001",0,"20210315":IF _ERRPART<>0 THEN GOTO ERR
VIMSG 1, _PARTPRZ$(3)
DELPART "P","201.00028",0,"ART001",0,"20210315":IF _ERRPART<>0 THEN GOTO ERR
```

```
AZZVARSYS 7 : _PARTTIP$="P"
_PARTCLI$="201.00028" : _PARTCAC=0 : _PARTART$="ART001" : _PARTCAA=0
_PARTDDT$="20210315" : _PARTADT$="20211231" : _PARTVAL=2 : _PARTPF$="Q"
FOR I=1 TO 3 : _PARTFIN(I)=I : _PARTPRZ$(I)=STR$(I*33,0,0) : NEXT
_PARTPRZ$(4)="100" : PUTPART : IF _ERRPART<>0 THEN VIMSG 4, _ERRPART$
```

```
AZZVARSYS 7 : IMODO=0: CONTO$="201.00028": ART$="ART001"
LOOP:
GETPARTICOL "P",COD$,CATC,ART$,CATA,"",IMODO: IF _ERRPART<>0 THEN GOTO ERR
IF _PARTCLI$="" AND _PARTART$="" AND _PARTCAC=0 AND _PARTCAA=0 THEN GOTO FIN
```

```
PRINT _PARTCLI$+ " : fino a "+STR$( _PARTFIN(1),0,0)+": valore "+ _PARTPRZ$(1)
PRINT _PARTCLI$+ " : fino a "+STR$( _PARTFIN(2),0,0)+": valore "+ _PARTPRZ$(2)
PRINT _PARTCLI$+ " : fino a "+STR$( _PARTFIN(3),0,0)+": valore "+ _PARTPRZ$(3)
PRINT _PARTCLI$+ " : oltre "+STR$( _PARTFIN(3),0,0)+": valore "+ _PARTPRZ$(4)
VIMSG 1, "visualizzati a video|gli scaglioni della particolarità"
```

```
AZZVARSYS 7 : IMODO=1 : GOTO LOOP
```

```
VIMSG 1, "visualizzati a video|gli scaglioni della particolarità"
ERR: IF _ERRPARTS<>" THEN VIMSG 4, _ERRPARTS
FIN: END
```

Caso: variabili array _PARTPRZ\$() e _PARTPRZN\$()

(consultare anche il caso: variabili array _ARSCQ\$(), _ARSCQN(), _ARSCO\$(), _ARSCON\$())

La variabile _PARTPRZN\$() è per gestire gli sconti in cascata fino a max 53 caratteri, ma essendo un archivio unico quello delle particolarità, entrambe le variabili (" _PARTPRZ\$()" e " _PARTPRZN\$()") valorizzano i medesimi campi per tutte le particolarità, che siano di tipo "Sconto", "Prezzo", "Provvigione" e "Articolo". Gli eseguibili compilati con versioni precedenti alla 2006C sono utilizzabili anche nelle ultime versioni, ma nel momento in cui tali sprix vengano modificati, e quindi necessitano di una nuova compilazione, sarà necessario sostituire la variabile " _PARTPRZ\$()" con la nuova variabile " _PARTPRZN\$()" . La procedura richiede:

- in fase di scrittura, l'istruzione PUTPART permette di poter valorizzare sia la variabile " _PARTPRZ\$()" per la vecchia gestione, che la " _PARTPRZN\$()" per la nuova gestione.
- in fase di revisione di una o più particolarità, la procedura richiede la valorizzazione della nuova variabile " _PARTPRZN\$()" in quanto l'istruzione GETPART, eseguita prima della PUTPART, necessita obbligatoriamente la valorizzazione della nuova variabile.

Si ricorda che la nuova variabile è prioritaria rispetto alla vecchia, per cui qualora venga utilizzata la vecchia occorre comunque azzerare la nuova. Ogni qual volta si volesse leggere un valore superiore ad 8 caratteri con la vecchia variabile " _PARTPRZ\$()" verranno restituiti una serie di asterischi. Ciò si verifica anche valorizzando le variabili di dizionario "tzprn()", vecchia variabile, anziché ricorrere alla nuova "tzprn()".

Caso: estrarre "particolarità" da un ciclo tabellare diverso dalla tabella "tz"

Per potere estrarre informazioni inerenti le "particolarità" di un articolo da un ciclo tabellare differente da "tz" è necessario introdurre i sei parametri di riferimento. Tali parametri sono indispensabili in quanto non è possibile altrimenti selezionare un record univoco che individui la particolarità di interesse. Se non si conoscono i valori di tali parametri ma si ha comunque la necessità di estrarre tali particolarità si possono eseguire due cicli tabellari in cascata. In tal modo nel primo ciclo tabellare (eseguito su "tz") si estraggono tutte le informazioni inerenti i valori dei parametri da estrarre (tali valori dovranno ovviamente essere memorizzati all'interno di una variabile, di un array oppure di un file esterno) e nel secondo ciclo tabellare, eseguito su una tabella di dizionario differente da quella delle particolarità, si estraggono tutte le informazioni relative con riferimento al primo ciclo tabellare. Di seguito viene mostrato un esempio:

```
|DIM PART$(,4)
|'INSERISCO IN UN ARRAY TUTTE LE PARTICOLARITA' ESISTENTI
+-----+
|Cod.Tab.:tz ALTRI FILES ARTICOLO - PREZZI CLIENTE/ARTICOLO
+-----+
tz |NUM=NUM+1
tz |PART$(NUM,1)=tztip()'TIPOLOGIA DELLA PARTICOLARITA'
tz |PART$(NUM,2)=tzcli()'CODICE CLIENTE
tz |PART$(NUM,3)=tzart()'CODICE ARTICOLO
tz |PART$(NUM,4)=tzddt()'DATA DI VALIDITA' DA...
|
|'DATO UN ARTICOLO, SE HA UNA PARTICOLARITA' LA STAMPO...
N +-----+
N |C. ARTICOLO QUANTITA' PREZZO |
N +-----+
+-----+
|Cod.Tab.:ar ARTICOLO - ANAGRAFICA
+-----+
ar | FOR I=1 TO NUM
ar | IF PART$(I,3)<>arcod() THEN GOTO PROSSIMO
ar | FOR J=1 TO 4
ar N |~arcod() ~tzfin(J) (PART$(I,1),PART$(I,2),0,arcod(),0,PART$(I,4) ) ~tzprz(J) (PART$(I,1),PART$(I,2),0,arcod(),0,PART$(I,4) )
ar | NEXT
ar | PROSSIMO:
ar | NEXT
N | |
N +-----+
|END
```

Caso: l'evento ON_PUTAR: non valorizza la variabile _PART?

Effettuando la GETATTPART all'evento ON_PUTAR:, questa non ritorna valori corretti delle variabili _PART? (per esempio 00000000 anziché la data inserita nel campo a data validità della _PARTADT\$). Occorre avvalersi dell'evento più specifico di scrittura particolarità ON_PUTTZ:, in quanto in quanto le variabili di tipo diverso da _AR? non tutte risultano valorizzate all'evento ON_PUTAR:.

Caso: particolarità Prezzi

Tramite l'utilizzo della funzione GETPART vengono valorizzate le variabili nel seguente modo:
 _PARTPRZ\$(1-4) particolarità semplici in caso di sconti non sono gestiti quelli in cascata.
 _PARTPRZN\$(1-4) introdotte con gli sconti in cascata, sostituiscono integralmente le _PARTPRZ\$().
 _PARTPRZN\$(4-8) nuovi scaglioni impostano gli elementi del 4,5,6,7,8 scaglione.

Il nono scaglione cioè l'OLTRE viene sempre impostato tramite la variabile _PARTPRZ\$(4) o _PARTPRZN\$(4) impostando il valore 4. Per la PUTPART valgono le stesse regole.

Nota: _PARTPRZ\$(4) o _PARTPRZN\$(4) nella finestra degli scaglioni leggono o scrivono sempre il 9 elemento cioè l'OLTRE

Caso: aggiornamento particolarità provvigione su sconto in caso di revisione dello sconto di riga

La revisione della riga (di documento già esistente oppure di documento nuovo di cui si ritorna sulla riga e la si modifica) non aggiorna il dato relativo alla particolarità (es. provvigione in base allo sconto), pertanto, in caso di revisione occorre aggiornare manualmente. Nel caso non si volesse lasciare questa incombenza all'utente, possono essere impiegati vari stratagemmi, fra cui i seguenti due esempi.

Il primo consiste in un Collage che possa eseguire controlli direttamente sulla riga

```
ON IN MDSCON: SCO$= _SINP$ : END
ON OUT MDSCON: IF _SINP$<>SCO$ THEN GOTO VARIA ELSE END : END
VARIA: _TF=0 : PUTTAST "#626" : PUTTAST "33,00" ' (oppure si esegue una GETPARTICOL per leggere la percentuale da inserire)
END
```

Il secondo consiste in un Collage che controlla alla conferma della revisione del documento, ma è più articolato e richiederebbe più risorse. Per esempio, ho tre particolarità provvigioni su sconto e ho un articolo su documento di cui revisiono lo sconto:

```
ON IN MDCAU: 'memorizzo sconti di riga
```

```
  DIM SCO_DOC$( )
  GETATTMM
  FOR X=1 TO NMM
    SCO_DOC$(X)= MMSCO$(X)
```

```

NEXT
END
ON PUTMM:
GETATTMM
FOR X=1 TO NMM
  AZZVARSYS 7
  GETPARTICOL "V",_MMCLI$,0,_MMART$(X),0,"",0
  IF _ERRPART<>0 THEN VIMSG 1,_ERRPART$
  DIM SCOPART$( ) : DIM PROPART$( )
  FOR Y=1 TO 3 'carico le prime tre particolarita'
    SCOPART$(Y)=STR$( _PARTFIN(Y),2,0) : PROPART$(Y)=_PARTPRZ$(Y)
  NEXT
  IF _MMSCO$(X)=SCO_DOC$(X) THEN END ELSE GOSUB VARIA
NEXT
MSG1$="Hai modificato lo sconto!! Modifica la provvigione particolare su sconto|"
MSG2$=" che risulterebbe dover essere del "+STR$( _MMPRO(K),2,0)+" %|"
MSG3$=" oppure lo faccio io se mi esegui una SHELLSPRIX"
VIMSG 1,MSG1$+MSG2$+MSG3$
END
VARIA:
FOR Z=1 TO 3 'scelgo fra le prime tre particolarita'
  IF _MMSCO$(X)=STR$( _PARTFIN(Z),2,0) THEN _MMPRO(X)=VAL(PROPART$(Z))
  K=X
NEXT
RETURN
  
```

Esempio: date le seguenti combinazioni di elementi anagrafici e tabellari relativi alla particolarità Prezzo Fisso senza data di validità

Codice conto cliente (tzcli)	Codice articolo (tzart)	Listino vendita (tzcac)	Categoria prezzi (tzcaa)	Cat. stat. cliente (tzstc)	Zona (tzzon)	Cat. stat. articolo (tzsaa+tzsan)	Gruppo merceologico (tzgrm)	Natura (tznat)	Prezzo fisso €
	"PARTICOLART"					"A"+02			20,00
							1		26,00
			1						27,00
									25,00
								"N03"	28,00
"201.00047"	"PARTICOLART"								10,00
"201.00047"						"A"+02			11,00
"201.00047"							1		12,00
"201.00047"			1						13,00
"201.00047"								"N03"	14,00
	"PARTICOLART"	5							29,00
		5				"A"+02			30,00
		5					1		31,00
		5	1						36,00
		5						"N03"	32,00
	"PARTICOLART"			2					15,00
				2		"A"+02			17,00
				2			1		18,00
			1	2					16,00
				2				"N03"	19,00
	"PARTICOLART"				3				33,00
					3	"A"+02			22,00
					3		1		23,00
			1		3				21,00
					3			"N03"	24,00

VISTA GESTIONALE

T-Data	A Data	Articolo	cA	Stat.A	Grp.Merc	Nat	Conto	cC	Stat.C	Zona	Tip
P		PARTICOLART					Tutti i clienti				Fis
P		Tutti gli articoli		A02			Tutti i clienti				Fis
P		Tutti gli articoli			1		Tutti i clienti				Fis
P		Tutti gli articoli				1	Tutti i clienti				Fis
P		Tutti gli articoli					N03 Tutti i clienti				Fis
P		PARTICOLART					PARTICOLATO SRL				Fis
P		Tutti gli articoli		A02			PARTICOLATO SRL				Fis
P		Tutti gli articoli			1		PARTICOLATO SRL				Fis
P		Tutti gli articoli				1	PARTICOLATO SRL				Fis
P		Tutti gli articoli					N03 PARTICOLATO SRL				Fis
P		PARTICOLART						5			Fis
P		Tutti gli articoli		A02				5			Fis
P		Tutti gli articoli			1			5			Fis
P		Tutti gli articoli				1		5			Fis
P		Tutti gli articoli					N03	5			Fis
P		PARTICOLART					Tutti i clienti		2		Fis
P		Tutti gli articoli		A02			Tutti i clienti		2		Fis
P		Tutti gli articoli			1		Tutti i clienti		2		Fis
P		Tutti gli articoli				1	Tutti i clienti		2		Fis
P		Tutti gli articoli					N03 Tutti i clienti		2		Fis
P		PARTICOLART					Tutti i clienti			3	Fis
P		Tutti gli articoli		A02			Tutti i clienti			3	Fis
P		Tutti gli articoli			1		Tutti i clienti			3	Fis
P		Tutti gli articoli				1	Tutti i clienti			3	Fis
P		Tutti gli articoli					N03 Tutti i clienti			3	Fis

COMBINAZIONI PREZZO FISSO

```

-----
N   Articolo      cA   -   StatA   -   GrpMer   -   Nat   -   Conto   -   cC   -   StatC   -   Zona   -   Part.   Prezzo Fisso
N   tzart         tzcaa -   tzsaa+tzsan -   tzgrm   -   tznat   -   tzcli   -   tzstc   -   tzcac   -   tzzon   -   tzpr8
-----
N   -----
      |Cod.tab.tz ALTRI FILES ARTICOLO - PREZZI CLIENTE/ARTICOLO
      |-----
N   ~tzart()      -   ~tzcaa - ~tzsaa~tzsan - ~tzgrm - ~tznat - ~tzcli() - ~tzstc - ~tzcac - ~tzzon - ~tzpr8
-----
AZZCATVARSYS "PART"

CLI$="201.00047" 'tzcli - Conto - (Codice conto)
STATC=2         'tzstc - StatC - (Cat.Stat. cliente)
ZON=3          'tzzon - Zona - (Zona conto)
LIS=5          'tzcac - cC - (Listino vendita) - (parametro key principale 'CC' <catcon>)
  
```

```

ART$="PARTICOLART"      'tzart - Articolo - (Codice articolo)
CATAR$="A"              'tzsaa - StatA - (Cat.stat.articolo Alfanumerica)
CATAR=02                'tzsan - StatA - (Cat.stat.articolo Numerica)
GRU$="1"                'tzgrm - GrpMerc - (Gruppo merceologico)
CATPRE=1                'tzcaa - cA - (Categoria prezzi formazione articolo) - (parametro key principale 'Ca' <catart>)
NAT$="N03"              'tznat - Nat - (Natura)

```

```

'GETPART <tpart$>,<codcon$>,<catcon$>,<codart$>,<catart$>,<dtvalda$>
'GETPARTICOL <tpart$>,<codcon$>,<catcon$>,<codart$>,<catart$>,<dtvalda$>,[mod]
'tzpr8(9) (Tp$,Cto$,CC,Art$,Ca,Dt$)

```

```

' codice articolo tzart (20,00) -----
GETPART "P", "", 0, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P", "", 0, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P", "", 0, ART$, 0, "") : VIMSG 1, "tzpr8 - P = " + TZPR8$

' cat.sta.articolo tzsaa + cat.sta.articolo tzsan (26,00) -----
GETPART "P E", "", 0, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P E", "", 0, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P E", "", 0, CATAR$, CATAR, "") : VIMSG 1, "tzpr8 - P E = " + TZPR8$

' gruppo merceologico tzgrm (27,00) -----
GETPART "P M", "", 0, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P M", "", 0, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P M", "", 0, GRU$, 0, "") : VIMSG 1, "tzpr8 - P M = " + TZPR8$

' categoria formazione prezzi tzcaa (25,00) -----
GETPART "P", "", 0, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P", "", 0, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P", "", 0, "", CATPRE, "") : VIMSG 1, "tzpr8 - P = " + TZPR8$

' natura tznat (28,00) -----
GETPART "P U", "", 0, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P U", "", 0, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P U", "", 0, NAT$, 0, "") : VIMSG 1, "tzpr8 - P U = " + TZPR8$

' codice articolo tzart + conto cliente tzcli (10,00) -----
GETPART "P", CLI$, 0, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P", CLI$, 0, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P", CLI$, 0, ART$, 0, "") : VIMSG 1, "tzpr8 - P = " + TZPR8$

' cliente tzcli + categoria statistica articolo tzsaa+tzsan (11,00) -----
GETPART "P E", CLI$, 0, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P E", CLI$, 0, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P E", CLI$, 0, CATAR$, CATAR, "") : VIMSG 1, "tzpr8 - P E = " + TZPR8$

' cliente tzcli + gruppo merceologico tzgrm (12,00) -----
GETPART "P M", CLI$, 0, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P M", CLI$, 0, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P M", CLI$, 0, GRU$, 0, "") : VIMSG 1, "tzpr8 - P M = " + TZPR8$

' cliente tzcli + categoria statistica prezzi (13,00) -----
GETPART "P", CLI$, 0, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P", CLI$, 0, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P", CLI$, 0, "", CATPRE, "") : VIMSG 1, "tzpr8 - P = " + TZPR8$

' cliente tzcli + natura tznat (14,00) -----
GETPART "P U", CLI$, 0, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P U", CLI$, 0, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P U", CLI$, 0, NAT$, 0, "") : VIMSG 1, "tzpr8 - P U = " + TZPR8$

' cat.sta.cliente tzstc + articolo tzart (29,00) -----
GETPART "P", "", STATC, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P", "", STATC, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P", "", STATC, ART$, 0, "") : VIMSG 1, "tzpr8 - P = " + TZPR8$

' listino tzcac + articolo tzart (29,00) -----
GETPART "P", "", LIS, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P", "", LIS, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P", "", LIS, ART$, 0, "") : VIMSG 1, "tzpr8 - P = " + TZPR8$

' listino tzcac + cat.sta.articolo tzsaa+tzsan (30,00) -----
GETPART "P E", "", LIS, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPART "PNE", "", LIS, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P E", "", LIS, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
GETPARTICOL "PNE", "", LIS, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P E", "", LIS, CATAR$, CATAR, "") : VIMSG 1, "tzpr8 - P E = " + TZPR8$
TZPR8$=tzpr8(9) ("PNE", "", LIS, CATAR$, CATAR, "") : VIMSG 1, "tzpr8 - PNE = " + TZPR8$

' listino tzcac + gruppo merceologico tzgrm (31,00) -----
GETPART "P M", "", LIS, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPART "PNM", "", LIS, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P M", "", LIS, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
GETPARTICOL "PNM", "", LIS, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P M", "", LIS, GRU$, 0, "") : VIMSG 1, "tzpr8 - P M = " + TZPR8$
TZPR8$=tzpr8(9) ("PNM", "", LIS, GRU$, 0, "") : VIMSG 1, "tzpr8 - PNM = " + TZPR8$

' listino tzcac + categoria prezzi articolo tzcaa (36,00) -----
GETPART "P N", "", LIS, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPART "P T", "", LIS, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPART "PNT", "", LIS, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P N", "", LIS, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P T", "", LIS, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
GETPARTICOL "PNT", "", LIS, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P N", "", LIS, "", CATPRE, "") : VIMSG 1, "tzpr8 - P N = " + TZPR8$
TZPR8$=tzpr8(9) ("P T", "", LIS, "", CATPRE, "") : VIMSG 1, "tzpr8 - P T = " + TZPR8$
TZPR8$=tzpr8(9) ("PNT", "", LIS, "", CATPRE, "") : VIMSG 1, "tzpr8 - PNT = " + TZPR8$

' listino tzcac+ natura tznat (32,00) -----
GETPART "P U", "", LIS, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPART "PNU", "", LIS, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P U", "", LIS, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
GETPARTICOL "PNU", "", LIS, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)
TZPR8$=tzpr8(9) ("P U", "", LIS, NAT$, 0, "") : VIMSG 1, "tzpr8 - P U = " + TZPR8$
TZPR8$=tzpr8(9) ("PNU", "", LIS, NAT$, 0, "") : VIMSG 1, "tzpr8 - PNU = " + TZPR8$

' cat.sta.cliente tzstc + codice articolo tzart (15,00) -----
GETPART "P S", "", STATC, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART P ART$ = " + PARTPRZ$(4)
GETPARTICOL "P S", "", STATC, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL P ART$ = " + PARTPRZ$(4)

```

```

TZPR8$=tzpr8(9) ("PS", "", STATC, ART$, 0, "") : VIMSG 1, "tzpr8 - PS= " + TZPR8$

' cat.sta.cliente tzstc + cat.sta.articolo tzsaat+zsan (17,00) -----
GETPART "PSE", "", STATC, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PSE", "", STATC, CATAR$, CATAR, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PSE", "", STATC, CATAR$+RIGHT$( "00"+STR$(CATAR,0,0),2),0, "") : VIMSG 1, "tzpr8 - PSE= " + TZPR8$

' cat.sta.cliente tzstc + gruppo tzgrm (18,00) -----
GETPART "PSM", "", STATC, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PSM", "", STATC, GRU$, 0, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PSM", "", STATC, GRU$, 0, "") : VIMSG 1, "tzpr8 - PSM= " + TZPR8$

' cat.sta.cliente tzstc + categoria prezzi articolo tzcaa (16,00) -----
GETPART "PST", "", STATC, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PST", "", STATC, "", CATPRE, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PST", "", STATC, "", CATPRE, "") : VIMSG 1, "tzpr8 - PST= " + TZPR8$

' cat.sta.cliente tzstc + natura tznat (19,00) -----
GETPART "PSU", "", STATC, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PSU", "", STATC, NAT$, 0, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PSU", "", STATC, NAT$, 0, "") : VIMSG 1, "tzpr8 - PSU= " + TZPR8$

' articolo tzart + zona tzzon (33,00) -----
GETPART "PZ", "", ZON, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPART "PZ", "", ZON, ART$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PZ", "", ZON, ART$, 0, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PZ", "", ZON, ART$, 0, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PZ", "", ZON, ART$, 0, "") : VIMSG 1, "tzpr8 - PZ= " + TZPR8$
TZPR8$=tzpr8(9) ("PZ", "", ZON, ART$, 0, "") : VIMSG 1, "tzpr8 - PZ= " + TZPR8$

' cat.sta.articolo tzsaat+zsan + zona tzzon (22,00) -----
GETPART "PZE", "", ZON, CATAR$, CATAR, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PZE", "", ZON, CATAR$, CATAR, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PZE", "", ZON, CATAR$+RIGHT$( "00"+STR$(CATAR,0,0),2),0, "") : VIMSG 1, "tzpr8 - PZE= " + TZPR8$

' gruppo tzgrm + zona tzzon (23,00) -----
GETPART "PZM", "", ZON, GRU$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PZM", "", ZON, GRU$, 0, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PZM", "", ZON, GRU$, 0, "") : VIMSG 1, "tzpr8 - PZM= " + TZPR8$

' cat.prezzi.articolo tzcac + zona tzzon (21,00) -----
GETPART "PZ", "", ZON, "", CATPRE, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PZ", "", ZON, "", CATPRE, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PZ", "", ZON, "", CATPRE, "") : VIMSG 1, "tzpr8 - PZ= " + TZPR8$

' natura tznat + zona tzzon (24,00) -----
GETPART "PZU", "", ZON, NAT$, 0, "" : IF _ERRPART=0 THEN VIMSG 1, "GETPART_P_ART$ = " + _PARTPRZ$(4)
GETPARTICOL "PZU", "", ZON, NAT$, 0, "", 0 : IF _ERRPART=0 THEN VIMSG 1, "GETPARTICOL_P_ART$ = " + _PARTPRZ$(4)
TZPR8$=tzpr8(9) ("PZU", "", ZON, NAT$, 0, "") : VIMSG 1, "tzpr8 - PZU= " + TZPR8$

```

RISULTATO CICLOTABELLARE

Articolo tzart	cA tzcaa	StatA tzsaat+zsan	GrpMerc tzgrm	Nat tznat	Conto tzcli	cC - tzstc	StatC - tzcac	Zona - tzzon	Part. - tzpr8	Prezzo Fisso
PARTICOLART	-	-	-	-	-	-	-	-	-	20,00
-	-	A	2	-	-	-	-	-	-	26,00
-	-	-	1	-	-	-	-	-	-	27,00
-	1	-	-	-	-	-	-	-	-	25,00
-	-	-	-	N03	-	-	-	-	-	28,00
PARTICOLART	-	-	-	-	201.00047	-	-	-	-	10,00
-	-	A	2	-	-	-	-	-	-	11,00
-	-	-	1	-	-	-	-	-	-	12,00
-	1	-	-	-	201.00047	-	-	-	-	13,00
-	-	-	-	N03	201.00047	-	-	-	-	14,00
PARTICOLART	-	-	-	-	-	-	5	-	-	29,00
-	-	A	2	-	-	-	5	-	-	30,00
-	-	-	1	-	-	-	5	-	-	31,00
-	1	-	-	-	-	-	5	-	-	36,00
-	-	-	-	N03	-	-	5	-	-	32,00
PARTICOLART	-	-	-	-	-	2	-	-	-	15,00
-	-	A	2	-	-	2	-	-	-	17,00
-	-	-	1	-	-	2	-	-	-	18,00
-	1	-	-	-	-	2	-	-	-	16,00
-	-	-	-	N03	-	2	-	-	-	19,00
PARTICOLART	-	-	-	-	-	-	-	3	-	33,00
-	-	A	2	-	-	-	-	3	-	22,00
-	-	-	1	-	-	-	-	3	-	23,00
-	1	-	-	-	-	-	-	3	-	21,00
-	-	-	-	N03	-	-	-	3	-	24,00

ANAGRAFICA TIPO LOTTO (TPLOT)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 18	Istruzione che azzerà variabili anagrafica tipo lotto _TPLOT	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "TPLOT"	Istruzione che azzerà variabili struttura _TPLOT (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETTPLOT	Par: <tpplot\$>, [ord]. Istruzione che legge anagrafica tipo lotto Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
PUTTPLOT	Istruzione che scrive anagrafica tipo lotto Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
DELTTPLOT	Par: <tpplot\$>. Istruzione che cancella anagrafica tipo lotto Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd/Sh
_ERRTPLOT	Codice risultato operazione anagrafica tipo lotto	S/Cd
_ERRTPLOT\$	Messaggio operazione anagrafica tipo lotto	S/Cd
_NDUTPLOT	Numero di dati utente gestiti nei tipi lotto	S/Cd
_TPLOT	Variabili di struttura anagrafica tipo lotto	S/Cd

esempi e casi dizionario

Esempio: scrive tipo lotto generico, lo duplica personalizzandolo e infine cancella l'originario tipo lotto

```

_TPLOTCOD$="***" : _TPLOTDESC$="Descriz. Tipo lotto"
_TPLOTSTAT$="C" : _TPLOTUNI$="S"
PUTTPLOT : IF _ERRTPLOT<>0 THEN VIMSG 4, _ERRTPLOT$
VIMSG 1, "Ha scritto il tipo lotto generico|" + _TPLOTCOD$
TPL$= _TPLOTCOD$
AZZVARSYS 18 : GETTPLOT TPL$,2 : IF _ERRTPLOT<>0 THEN VIMSG 4, _ERRTPLOT$
_TPLOTCOD$="***" : _TPLOTDESC$="Descriz. NUOVO tipo lotto"
_TPLOTSTAT$="C" : _TPLOTUNI$="S"
_TPLOTAPGSS$="E" : _TPLOTAPTP$="F" : _TPLOTAPCO$="S" : _TPLOTAPCL$="S"
_TPLOTAPSL$="N" : _TPLOTAPQR$="N" : _TPLOTSCAQ$="N" : _TPLOTATP$="M"
_TPLOTDUCD$(1)="** 3" : _TPLOTDUET$(1)="Etichetta" : _TPLOTDUTP$(1)="A"
_TPLOTDUDI(1)=20 : _TPLOTDUOB$(1)="S" : _TPLOTDUOR(1)=1
_TPLOTDUVL$(1)="Valore iniziale": _TPLOTDUCH(1)=4
PUTTPLOT : IF _ERRTPLOT<>0 THEN VIMSG 4, _ERRTPLOT$
VIMSG 1, "Ha duplicato il tipo lotto generico| e lo ha personalizzato"
DELTTPLOT TPL$ : IF _ERRTPLOT<>0 THEN VIMSG 4, _ERRTPLOT$
VIMSG 1, "Ha cancellato l'iniziale tipo lotto generico"

```

ANAGRAFICA GRUPPI MERCEOLOGICI (GM)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 37	Istruzione che azzerà variabili struttura gruppi merceologici _GM	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "GM"	Istruzione che azzerà variabili struttura _GM (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETGM	Par: <gruppo\$>. Istruzione che legge gruppi merceologici	S/Cr/Cd/Sh
PUTGM	Istruzione che scrive gruppi merceologici	S/Cr/Cd/Sh
_ERRGM	Codice risultato operazione	S/Cd
_ERRGM\$	Descrizione del codice risultato	S/Cd
_GM	Variabili struttura	S/Cd
esempi e casi dizionario		
<p>Esempio: Scrittura di una anagrafica dei GRUPPI MERCEOLOGICI</p> <pre> _GMCOD\$ = "GRUPPO1" `Codice _GMANN\$ = "N" `Flag Annullamento logico _GMDESS\$ = "Primo gruppo merceologico" _GMDTCS\$ = sxdat `Data Creazione _GMDTMS\$ = sxdat `Data ultimo aggiornamento _GMUM1\$ = "Nr" `Unita di misura primaria _GMRIC\$ = "" `Centro di Ricavo _GMCOSS\$ = "" `Centro di Costo _GMCDP\$ = "" `Gruppo merc. Padre PUTGM IF _ERRGM <> 0 THEN VIMSG 4,_ERRGM\$ ` *** Azzeramento della struttura AZZVARSYS 37 `Azzero la struttura ` *** Lettura di una anagrafica dei GRUPPI MERCEOLOGICI GETGM "GRUPPO1" IF _ERRGM <> 0 THEN VIMSG 4,_ERRGM\$ </pre>		

BOLLA DI LAVORAZIONE (OB-OL)

Le funzioni e variabili di Produzione, non sono utilizzabili dai livelli di prodotto che non sono di Produzione		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 6	Istruzione che azzerava variabili bolla di lavorazione _OB e _OL	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "BL"	Istruzione che azzerava variabili struttura _OB e _OL (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETBL	Par: <numeroBL>,<numeroSottoBL>. Istruzione che legge bolla di lavorazione La variabile dello scarico automatico nella bolla di lavoro, non è disponibile poichè è un dato non memorizzato in archivio. Viene stabilito nella gestione a video al momento della formazione della BL. Collage può intervenire nella gestione a video. Non è disponibile una variabile _OB relativa all'ultimo numero della sottobolla generata; valutare la possibilità di leggere attraverso il ciclotabellare.	S/Cr/Cd/Sh
GETATTBL	Istruzione che legge bolla di lavorazione a video	C
GETNRIBLVID()	Funzione di lettura numero di riga a video del corpo della bolla di lavoro	S/C/Cr/Cd/Sm
PUTBL	Istruzione che scrive bolla di lavorazione. L'istruzione non scrive in alcun modo archivi MYDB (vedere le note in calce alla tabella dei comandi MyDB); gli eventuali archivi possono essere riportati nei documenti CL, SL, DL solo attraverso la procedura standard dal modulo Magazzino. La PUTBL non scrive Controllo di Gestione, utilizzare la PUTCDG. Se non tutte le variabili di riga sono impostate oppure il numero della riga in produzione per il magazzino non è valorizzato, la PUTBL genera la testata e non il relativo dettaglio, senza alcun messaggio di errore. La non corretta inizializzazione delle variabili _OLTPR\$ (tipo riga) _OLLAV(numero sottoBL -stottobolla -) genera una bolla di lavoro (BL) senza il flag SML sospesi nonostante nella distinta siano previsti (non aggancia correttamente i semilavorati). Nel piede dell'elenco delle variabili _OL ci sono due importanti note, che riportano le seguenti condizioni operative: (*a) - "R"=normale; "r"=normale bloccata -BL lavorante-; "P"=PF sospeso -solo con sottoBL-; "S"=SML semilavorato sospeso; (*b) - =0 SML sospeso; >0 in lavorazione (32000 per le BL x.0000); <0 già caricato (32000 per le BL x.0000) La PUTBL non ha automatismi e va ricostruita. Se non si usano più elementi di riga il programma non ha a disposizione abbastanza vettori per caricare sia la bolla che il SML. Per capire come si deve ricostruire una corretta PUTBL, basta eseguire la stessa operazione dalla procedura standard e poi leggere con GETBL tutta la struttura (ma con più vettori).	S/Cr/Cd/Sh
DELBL	Par: <numeroBL>,<numeroSottoBL>. Istruzione che cancella bolla di lavorazione Per la cancellazione di una riga PF da una bolla o piano di lavoro occorre azzerare le seguenti variabili di struttura: _OLTPR\$(N_RIGA_BL)="", _OLLAV(N_RIGA_BL)=0, _OLART\$(N_RIGA_BL)=""	S/Cr/Cd/Sh
_ERRBL	Codice risultato operazioni su bolla di lavorazione	S/Cr/Cd/Sh
_ERRBL\$	Messaggio operazioni su bolla di lavorazione	S/Cr/Cd/Sh
_OB	Variabili di struttura testata della BL	S/Cr/Cd/Sh
_OL	Variabili di struttura righe della BL	S/Cr/Cd/Sh
_NBL	Numero di righe bolla di lavorazione	S/Cr/Cd/Sh
COLOCBL	Par: <NumBL>,<NumPF>,<NumRecRigaOC>. Istruzione di collegamento riga ordine ad una BL. Nota: fino che non viene prevista un'operazione inversa alla COLOCBL, l'unico modo per eseguire uno scollegamento (scolocbl) è quello di cancellare la bolla (compreso sottobolle ecc.), replicarla e infine ricollegarla.	S/Cr/Cd/Sh
_ERRCOLOCBL	Codice risultato operazione di collegamento	S/Cr/Cd/Sh
_ERRCOLOCBL\$	Messaggio operazione di collegamento	S/Cr/Cd/Sh
SPRIGBL	Par:<NumBL>,<NumSottoBL>,<CodPF>,<NumRifPF>,<qta1>,<qta2>,...,<qta32> Istruzione che spezza riga di una bolla di lavorazione	S/Cr/Cd/Sh
_ERRSPRIGBL	Codice risultato operaz. Spezza Riga	S/Cd
_ERRSPRIGBL\$	Messaggio operazione Spezza Riga	S/Cd
_ERWSPRIGBL\$	Variabile di sola lettura valorizzata solo in esecuzione di SPRIGBL da terminali senza interfaccia utente, quando i lotti associati alle materie prime vengono azzerati dall'operazione di spezzamento riga.	S/Cr/Sh
ON_STARTBL:	Evento che richiede uscita dal primo campo della Bolla di Lavorazione (voce di menu PB)	C/Cd
ON_DELBL:	Evento di cancellazione bolla, consente lettura GETATTBL	C/Cd
Logiche, esempi e casi dizionario		
tabella "ob" e "ol"	Nella bolla di lavoro "ob" se non è specificato l'ordine, l'ordinamento è per obnum(),obnso(). La prima sottobolla di ogni bolla è la numero -1 bolla padre, fittizia e creata in automatico da programma. Nella tabella "ol" se l'ordine non è specificato, l'ordinamento è olnum(),ollav(),olart(). Nota: ollav() utilizza il valore 32000 quando fa riferimento alla BL principale e di conseguenza -32000 = Pronto, 0 = PF sospeso, 32000 = in lavorazione. Per trattare tutte le righe di una determinata BL, selezionare tutte le righe con campo ollav in comune (sia positivo che negativo). In tabella "oi" l'assenza di ordine genera ordinamento per oinum(),oisot(),oifif(),oifas.	S/C

oimag.

La variabile olrif() indica se collegata o meno all'ordine cliente: se il numero va da 1 a 16000000 significa che è collegata a OC, se il numero è superiore a 16000000 significa che non è collegata.

Esempio: variabili minime per PUTBL

```
_OBPIA$="N"
_OBNUM=10
_OBMAT=1
_OBMAG=1
_OBNSP = -1
_OLTPR$(1)="R"
_OLLAV(1)=32000
_OLART$(1)="Codice articolo"
_OLFFP$(1)="Codice articolo"
_OLRIF(1)=16000001
_OLQTF(1)=10
_OLNRX(1)=1
```

Sprix non ha gli stessi automatismi di avanzamento della Produzione del Gestionale E' tutto manuale e la procedura di avanzamento va ricostruita, in ogni caso non è possibile creare sottobolle.

Procedura:

- leggere i prodotti con distinta base della bolla e delle relative sottobolle (GETBL)
- leggere gli impegni se presenti (GETIL), altrimenti SVILDB
- caricare la struttura in _MM per effettuare SL (PUTMM)
- cancellare gli impegni se presenti (DELIL)
- caricare la struttura in _MM per effettuare CL (PUTMM)
- avanzare la bolla per chiuderla (PUTBL) sulle fasi d'interesse (es. _OLFAS=100 -CL- e altre variabili _OL che sono necessarie
- dalla v815 è possibile con un terminale Produzione scrivere le variabili della rintracciabilità di riga del movimento di magazzino

Esempio: semplice PUTBL

```
_OBNUM=AUTO
_OBMAG=1
_OBMAT=1
_OBDAT$=sxdat
_OBNSP=-1
I=1
_OLTPR$(I)="R" riga normale
_OLLAV(I)=32000 per la bolla principale
_OLART$(I)="ART_PF" CODICE PRODOTTOFINITO
_OLRIF(I)=16000001 produzione per il magazzino
_OLNTG(I)=0 articolo non a taglie altrimenti 1
_OLQTF(I)=33 quantità da produrre
PUTBL
IF _ERRBL THEN VIMSG 4, _ERRBL$ ELSE VIMSG 4, STR$(_OBNUM,0,0)
```

Esempio: scrive una BL, la legge e la spezza, aggiunge una riga da collegare a un ordine cliente

```
_OBNUM=25: _OBNSP=-1: _OBMAG=2: _OBMAT=1: _OBDAT$="20210101"
_OLLAV(1)=32000: _OLRIF(1)=16000001: _OLRII(1)=1
_OLCLIS(1)="201.29": _OLTPR$(1)="R": _OLART$(1)="ARTDBPTAG":
_OLNTG(1)=2: Q=7: FOR T=1 TO 4: _OLTAG(1,T)=Q: NEXT
_OLLAV(2)=32000: _OLRIF(2)=16000002: _OLRII(2)=2
_OLCLIS(2)="201.28": _OLTPR$(2)="R": _OLART$(2)="MON-CTX14": _OLQTF(2)=3
PUTBL: IF _ERRBL<>0 THEN VIMSG 5, _ERRBL$
VIMSG 1, "Ha creato la Bolla di Lavoro 25.0000"

AZZVARSYS 6
GETBL 25,0: IF _ERRBL<>0 THEN VIMSG 5, _ERRBL$
FOR I=1 TO _NBL
'REM if _OLLAV(I)=0 then salta (...con SML sospesi, considerati in stampa)
VIMSG 1, "Riga "+STR$(I,0,0)+" Art: "+_OLART$(I)+" Qtà:"+STR$( _OLQTF(I)
NEXT

SPRIGBL 25,0,"ARTDBPTAG",olrii(25,-1,1),2,2,2,2
IF _ERRSPRIGBL<>0 THEN VIMSG 1, _ERRSPRIGBL$
VIMSG 1, "Ha spezzato la riga del PF a taglie della BL"
AZZVARSYS 6

GETBL 25,0: IF _ERRBL<>0 THEN VIMSG 5, _ERRBL$
NR= _NBL+1
_OLLAV(NR)=32000: _OLRIF(NR)=(16000000+NR): _OLRII(NR)=NR
_OLTPR$(NR)="R": _OLART$(NR)="ARTDBP": _OLQTF(NR)=15
_OLSDO(NR)=1: _OLORD(NR)=290: _OLDAT$(NR)="20210101"
_OLSCOS(NR)="20210424": _OLSCA$(NR)="20210424"

PUTBL: IF _ERRBL<>0 THEN VIMSG 5, _ERRBL$
VIMSG 1, "Ha aggiunto una riga di PF da collegare a un OC"
IF ocnum()=290 THEN N=ocnri() \ (NB: da ciclotabellare su tabella OC)
COLOCBL 25,NR,N: IF _ERRCOLOCBL<>0 THEN VIMSG 4, _ERRCOLOCBL$
VIMSG 1, "Ha collegato la riga PF all'OC | ora cancella una BL"
DELBL 22,0: IF _ERRBL<>0 THEN VIMSG 5, _ERRBL$
```

Esempio: collegamento ordine cliente (OC) a bolla (BL)

L'istruzione COLOCBL permette il collegamento passaggio dello stato di riga OC da "S" ad "L". La variabile _OLRIF() dove passare da un numero > 16000000 ad un numero inferiore e corrispondente al numero di riga collegata. La PUTBL può creare solo per il magazzino e quindi _OLRIF(n) > 16000000.

La creazione della bolla non consente il collegamento (nemmeno inserendo tutti i dati relativi all'ordine cliente sulla riga della bolla). Solo successivamente è possibile collegare l'ordine alla bolla, con COLOCBL agganciando la riga del prodotto finito con il numero record della riga dell'ordine cliente (variabile dizionario ocnri). Es. OC n.1/31 con prodotto finito e record riga ocnri=186; su bolla (ex novo, oppure appena letta come nell'esempio).

```
AZZVARSYS 6: GETBL 15,0: IF _ERRBL<>0 THEN VIMSG 4, _ERRBL$: VIMSG 4, STR$( _OLRIF(1),0,0)
_OLCLIS(1)="201.00006": _OLSDO(1)=1: _OLORD(1)=31: _OLDAT$(1)="20040101": _OLSCOS(1)="20040101": _OLSCA$(1)="20040101"
PUTBL: IF _ERRBL<>0 THEN VIMSG 4, _ERRBL$
'END
COLOCBL 15,1,186: IF _ERRCOLOCBL<>0 THEN VIMSG 4, _ERRCOLOCBL$
END
```

Alcune considerazioni sulla rintracciabilità delle righe d'ordine collegate alla BL e in stato "L" lavorazione.

La variabile `ocrif()` corrisponde alla variabile `olrif()` e non è un riferimento al numero documento BL, poiché indica che la riga della BL è collegata alla riga OC che ha il numero documento indicato a quelle due variabili uguali. La variabile `ocrif()` è il riferimento al Prodotto Finito, come valore numerico. La `ocrif()` si valorizza quando l'OC è collegato alla BL. Ogni PF in BL è marcato con un numero. Se nella Bolla c'è un solo PF nella maggior parte dei casi equivale a 1.

Le variabili di rintracciabilità relative all'archivio magazzino (`octbl()`, `octfs()`, `octdt()`, ecc.), non riguardano un collegamento alle fasi di lavoro in senso stretto, poiché si valorizzano solo dopo che il Prodotto Finito (PF) conclude la fase Carico di Lavorazione (CL) nella Bolla di Lavoro, ovvero il prodotto è stato realizzato ed è tracciabile.

Le variabili di rintracciabilità dell'archivio di magazzino si valorizzano quando il prodotto finito ha concluso il ciclo di lavoro, in quanto di ciò che avviene "prima" della fase CL l'archivio ordini non conosce le informazioni dinamiche che avvengono in produzione e che sono di competenza dell'archivio "ol" (non si conoscerebbe nemmeno quale tipo di fase: stampa, impegno MP, ecc.).

Per conoscere quale è la fase di lavoro occorre interrogare l'archivio produzione sia la bolla (`ocndo()`) che l'ordine (`olord()`) si conoscono, si conosce se il prodotto finito è rintracciabile (es. `octfs()`) oppure si può conoscere la fase con GETBL oppure dizionario "ol" (`olfsc()`, `olfsi()`, `olfsd()`).

Esempio: collegamento OC con lotto a BL

```
AZZVARSYS 4
_MMSIG$="OC"
_MMSE$=1
_MMNUM=36
_MMDAT$="20210101"
_MMMAG=1
_MMCLI$="201.00001"
_MMNTE(1)=1
_MMTPR$(1)="R"
_MMART$(1)="MON-CTX17"
_MMQTA(1)=1
_MMORD$(1)="S"
_MMPRZ(1)=100
_MMALIS(1)="20"
_MMIDLOT(1,1)=3
_MMLNDQTA(1,1)=0
_MMLQTA(1,1)=1
_PUTMM 0
IF _ERRMM<>0 THEN VIMSG 4, _ERRMM$

oc 'A=ocnri() 'ciclo tabellare su "oc" con scelta ocnum()=36

_OBPIA$="N"
_OBNUM=19
_OBDAT$="20210101"
_OBMAG=1
_OBMAT=1
_OBNSP=-1
_OLTPR$(1)="R"
_OLLAV(1)=32000
_OLART$(1)="MON-CTX17"
_OLPFP$(1)="MON-CTX17"
_OLRIF(1)=16000001
_OLCLI$(1)="201.00001"
_OLQTF(1)=1
_OLSDO(1)=1
_OLORD(1)=36
_OLDAT$(1)="20210101"
_OLSCO$(1)="20210101"
_OLSCA$(1)="20210101"

PUTBL
IF _ERRBL<>0 THEN VIMSG 4, _ERRBL$

COLOCBL 19,1,A
IF _ERRCOLOCBL<>0 THEN VIMSG 4, _ERRCOLOCBL$

GETBL 19,0
IF _ERRBL<>0 THEN VIMSG 4, _ERRBL$
FOR I=1 TO 1
  VIMSG 4, STR$( _OLLOT(I),0,0)
NEXT
END
```

In presenza di semilavorati non c'è l'automatismo che permette di lanciare le sottobolle automatiche (le bolle risultano collegate, ma non viene visualizzato in alto a destra il pulsante 'SML' e all'emissione delle sottobolle viene emesso il messaggio 'la bolla non contiene semilavorati'). Sprix non sempre attiva gli automatismi propri del gestionale, perché non raggiunge la sua stessa potenza. Per ottenere questi automatismi occorre replicarne le logiche: SVILDB degli articoli, identificare i semilavorati e stabilire `_OLTPR$()="S"` con `_OLLAV()=0` e associarvi il relativo `_OLRIF()` da 1 a 16.000.000 nella produzione per ordine cliente o > 16.000.000 nella produzione per magazzino.

Caso: metodi alternativi alla GETATTMM di documenti SL/CL/DL in fase di ON_PUTMM o ON_PUTBL dal menu BOLLA DI LAVORAZIONE

Esempi di metodi alternativi possono essere l'esecuzione con SHELLSPRIX di uno Sprix con GETMM sugli ultimi documenti (in base a quanto riportato dalla GETNUM) oppure con un ciclotabellare e relativa selezione: ON_PUTBL: GETATTBL : N=_OBNUM : SHELLSPRIX..."-vN" e ciclo 'mm' con selezione `mmrbo=VAL(_ARGCDM$)`.

Caso: identificazione univoca della riga = `_OBNUM()+_OBNSO()+OLRII()`

Caso: PUTBL con `_ERRBL$="Numero univoco riga(0), utilizzato più volte"` significa che il controllo sulla variabile `_OLRII` risulta impostato uguale su due righe (dichiarate due variabili `_OLRII()` uguali). Massimo `_OLRII()` è 65000

Caso: Le variabili `_OLUNI` e `_OLUNP` sono valorizzate se "gestione collegamento pf" degli ulteriori dati della DBP è impostato ad "S"

Num	36.0000	Data	08/01/15	MPF	1	NOSTRO	MP	1	NOSTRO	
Scad	Urgente	Note								
Piano di lav	N	Cod. conto					NoSML		SML	
Modulo	BL	Scar aut	N	Val scar			PF	N	Prz	
								N	Op	
								S	Fs	
Cat SML										
6	Codice e descrizione prodotto	Unr	Pronto	/	Lavoraz	Im	DL	St	SL	Fs
	DBPRIMA Articolo con distinta base	NR			13					
	DBPRIMA Articolo con distinta base	NR			22	1	1	1		
	DBPRIMA Articolo con distinta base	NR			23	2	2	2		
	DBPRIMA Articolo con distinta base	NR			1	3	3	3		
	DBPRIMA Articolo con distinta base	NR			1	4	4	4		

```
ON_PUTBL: AZZVARSYS 6 : GETATTBL
FOR X=1 TO _NBL ` (righe presenti nella BL)
  VIMSG 1, _OLART$(X) + " - " + STR$( _OLUNI(X),0,0) + " - " + STR$( _OLUNP(X),0,0)
NEXT
END
```

Il ciclo arriva a 7: 4 articoli trattati nella prima videata (gli ultimi quattro) e 3 (dettaglio della prima riga)

Caso: PUTBL non scrive semilavorati (SML) sospesi

La non corretta inizializzazione delle variabili `_OLTPR$` (tipo riga) `_OLLAV` (numero sottoBL -sottobolla -) genera una bolla di lavoro (BL) senza il flag SML sospesi nonostante nella distinta siano previsti (non aggancia correttamente i semilavorati). Nel piede dell'elenco delle variabili `_OL` ci sono due importanti note, che riportano le condizioni operative

Si ricorda che la PUTBL non ha automatismi (va ricostruita) e che l'utilizzo di una struttura ad un solo elemento di vettore non è sufficiente per caricare sia la BL che il SML. Per capire come costruire la struttura per la PUTBL è sufficiente eseguire la registrazione da gestionale e poi leggere (GETBL) tutta la struttura attraverso più vettori.

Caso: GETBL in stampa raddoppia le righe dei prodotti finiti

Per evitare una stampa doppia dei codici del PF occorre stabilire una condizione IF su `_OLLAV()`, poiché se esistono dei SML sospesi essi vengono considerati in stampa. Occorre la condizione `IF _OLLAV(J)=0 THEN SALTA`

Caso: Lettura Bolla di lavoro in MEXALDB

La tabella BLAV sono contenuti le seguenti colonne:

```
NGB_SBO_L_PADRE Numero sottobolla PADRE
NGL_BOL         Numero bolla
NGB_SBO_L       Numero sottobolla
```

La tabella BLSM invece ha le seguenti colonne:

```
NGL_BOL         Numero bolla
NGB_SBO_L       Numero sottobolla
NGB_SBO_L_LAV   Numero sottobolla lavorazione
```

La Bolla X.Y la cui bolla padre è la bolla X.Z è registrata in questo modo:

prodotti in lavorazione nella bolla:
X.Z.Y (Se Y uguale a 0 allora è scritto 32000)

Prodotti caricati
X.Z.-Y (Se Y uguale a 0 allora è scritto 32000)

Semilavorati sospesi
X.Y.0

Tradotto nelle colonne delle tabelle diventa:

- Bolla NGL_BOL . NGB_SBO_L_LAV la cui bolla padre è la bolla NGL_BOL . NGB_SBO_L è registrata in questo modo:
- prodotti in lavorazione nella bolla: NGL_BOL . NGB_SBO_L . NGB_SBO_L_LAV (Se NGB_SBO_L_LAV uguale a 0 allora è scritto 32000)
- prodotti caricati: NGL_BOL . NGB_SBO_L . -NGB_SBO_L_LAV (Se NGB_SBO_L_LAV uguale a 0 allora è scritto 32000)
- semilavorati sospesi: NGL_BOL . NGB_SBO_L_LAV . 0

Modifica articoli in documenti collegati (OC - BL) e mantenimento del collegamento

Può essere valutata la fattibilità e gli effetti delle seguenti operazioni:

- cancellazione OC e BL, riscrittura con modifiche e procedere con COLOCBL
- lettura GETMM, modifica codici, PUTMM, GETBL, modifica codici, PUTBL

Caso: correlazione Semilavorato e Sottobolla a Prodotto Finito e Bolla

In seguito sono indicati gli elementi della tabella "ol" che consentono di risalire alla correlazione fra le righe articolo e le bolle che li hanno generati, per capire il collegamento fra articolo prodotto finito della bolla principale e articolo semilavorato della sottobolla.

Ciò consente di rispondere a varie esigenze spesso simili fra loro, ma con punti di partenza diversi, per esempio, come individuare l'articolo semilavorato della sottobolla collegata all'articolo prodotto finito della bolla principale, oppure come rintracciare le sottobolle che derivano dalla riga della bolla principale, oppure, come rintracciare, a partire dall'articolo della bolla principale, le sottobolle a questo collegate e/o gli articoli delle sottobolle collegate a quell'articolo, oppure riferimenti delle sottobolle alla riga della bolla principale, ecc.. ecc..

In linea generale, spesso, le correlazioni non sono eseguibili "da monte a valle", bensì vanno ricostruite al contrario, ovvero, ciclando i risultati (componenti/sml delle sottobolle riferibili a bolle principali), esportarli e riordinarli, per poi relazionarli in modo da raggiungere chi li ha originati.

Indichiamo quali elementi consentono questa correlazione, ovvero "olrli" e "olunp". Data la bolla 10.0000 con due righe cliente, la prima con due prodotti finiti e la seconda con un prodotto finito, per un totale di 3 prodotti finiti...

Bolla di lavorazione 10 indice sottobolle			
Sottobolla note di lavorazione	Data	MMP	Distinta Base
Principale NOTEB	01/01/21	1	NOTEB
1 M/B PENTIUM PER NOTEBOOK	01/01/21	3	SL-MBPNB4343
2 SEZIONE AUDIO NOTEBOOK	01/01/21	3	SL-SCAUDIO

Numero 10.0000 Data 01/01/2021 Scadenza Urgente Piano di lavoro No
 Magazzino PF 1 SEDE CENTRA Magazzino MP 1 SEDE CENTRA
 Nota _____ Escludi SML _____
 Codice conto _____ Modulo stampa BL
 Scarico automatico No Valore scarico _____ Fissa Prezzi No
 Inf.Prod.Finiti Dettaglio MP+Rif. Operazione Scarico/Carico Fase operazione _____

1	Codice e descrizione prodotto	Um Pronto	/	Lavoraz	Im	DL	St	SL	Fs
NOTEB10004540MON	NOTEB 100 04 540 MON	pz		3					

...e relativo dettaglio delle righe...

Numero 10.0000 Data 01/01/2021 Scadenza Urgente Piano di lavoro No
 Magazzino PF 1 SEDE CENTRA Magazzino MP 1 SEDE CENTRA
 Nota _____ Escludi SML _____
 Codice conto _____ Modulo stampa BL
 Scarico automatico No Valore scarico _____ Fissa Prezzi No
 Inf.Prod.Finiti Dettaglio MP+Rif. Operazione Scarico/Carico Fase operazione _____

Dettaglio: NOTEB10004540MON NOTEB 100 04 540 MON											
Cliente	Numero	Data	Scad.Ordine	Scadenza	Quantita'	Im	DL	St	SL	S	Fs
				Urgente	2						
				Urgente	1						

1. 2 (2/ 0) Esist. 0 Dispon 0 Mag. 1

Numero 10.0000 Data 01/01/2021 Scadenza Urgente Piano di lavoro No
 Magazzino PF 1 SEDE CENTRA Magazzino MP 1 SEDE CENTRA
 Nota _____ Escludi SML _____
 Codice conto _____ Modulo stampa BL
 Scarico automatico No Valore scarico _____ Fissa Prezzi No
 Inf.Prod.Finiti Dettaglio MP+Rif. Operazione Scarico/Carico Fase operazione _____

Dettaglio: NOTEB10004540MON NOTEB 100 04 540 MON											
Cliente	Numero	Data	Scad.Ordine	Scadenza	Quantita'	Im	DL	St	SL	S	Fs
				Urgente	2						
				Urgente	1						

2. 2 (1/ 0) Esist. 0 Dispon 0 Mag. 1

Date le relative sottobolle dei semilavorati

Bolla di lavorazione 10 indice sottobolle			
Sottobolla note di lavorazione	Data	MMP	Distinta Base
Principale NOTEB	01/01/21	1	NOTEB
1 M/B PENTIUM PER NOTEBOOK	01/01/21	3	SL-MBPNB4343
2 SEZIONE AUDIO NOTEBOOK	01/01/21	3	SL-SCAUDIO

La prima sottobolla del primo semilavorato con due righe riferite ai quantitativi per ciascuna riga cliente. Spezzate in base all'impegno.

Numero	10.0001	Data	01/01/2021	Scadenza	Urgente	Piano di lavoro	No									
Magazzino PF	1 SEDE CENTRA.	Magazzino MP	3 DEPOSITO B													
Nota							Escludi SML									
Codice conto							Modulo stampa	BL								
Scarico automatico	No	Valore scarico						Fissa Prezzi	No							
Inf.Prod.Finiti	Dettaglio MP+Rif.	Operazione	Scarico/Carico					Fase operazione								
1 Codice e descrizione prodotto								Um Pronto	/	Lavoraz	Im	DL	St	SL	Fs	
SL-MBPNB4343		M/B PENTIUM PER NOTEBOOK						nr		2	99					
SL-MBPNB4343		M/B PENTIUM PER NOTEBOOK						nr		1						

In basso a sinistra del dettaglio sono visibili due serie di numeri fra esse relazionate e separate da "/".
La prima riga della prima sottobolla riporta la relazione (2/2) riferita alla quantità 2 semilavorati.

Numero	10.0001	Data	01/01/2021	Scadenza	Urgente	Piano di lavoro	No				
Magazzino PF	1 SEDE CENTRA.	Magazzino MP	3 DEPOSITO B								
Nota							Escludi SML				
Codice conto							Modulo stampa	BL			
Scarico automatico	No	Valore scarico						Fissa Prezzi	No		
Inf.Prod.Finiti	Dettaglio MP+Rif.	Operazione	Scarico/Carico					Fase operazione			
Dettaglio: SL-MBPNB4343 M/B PENTIUM PER NOTEBOOK											
Cliente	Numero	Data	Scad.Ordine	Scadenza	Quantita'	Im	DL	St	SL	S	Fs
				Urgente	2	99					
<div style="display: flex; justify-content: space-between; align-items: center;"> 1. 1 (2/ 2) Esist. 0 Dispon 0 </div>											
<small> <input type="checkbox"/> Varia data scadenza <input type="checkbox"/> Varia codice cliente <input type="checkbox"/> Rintrac.Doc. avanzamento PF </small>								<small> <input type="checkbox"/> Raggruppamento righe <input type="checkbox"/> Elimina riga <input type="checkbox"/> Ok <input type="checkbox"/> Annulla </small>			

Numero	10.0001	Data	01/01/2021	Scadenza	Urgente	Piano di lavoro	No									
Magazzino PF	1 SEDE CENTRA.	Magazzino MP	3 DEPOSITO B													
Nota							Escludi SML									
Codice conto							Modulo stampa	BL								
Scarico automatico	No	Valore scarico						Fissa Prezzi	No							
Inf.Prod.Finiti	Dettaglio MP+Rif.	Operazione	Scarico/Carico					Fase operazione								
2 Codice e descrizione prodotto								Um Pronto	/	Lavoraz	Im	DL	St	SL	Fs	
SL-MBPNB4343		M/B PENTIUM PER NOTEBOOK						nr		2	99					
SL-MBPNB4343		M/B PENTIUM PER NOTEBOOK						nr		1						

La seconda riga della prima sottobolla riporta la relazione (1/1) riferita alla quantità 1 semilavorato.

Numero	10.0001	Data	01/01/2021	Scadenza	Urgente	Piano di lavoro	No				
Magazzino PF	1 SEDE CENTRA	Magazzino MP	3 DEPOSITO B								
Nota							Escludi SML				
Codice conto							Modulo stampa	BL			
Scarico automatico	No	Valore scarico			Fissa Prezzi	No					
Inf.Prod.Finiti	Dettaglio MP+Rif.	Operazione	Scarico/Carico	Fase operazione							
Dettaglio: SL-MBPNB4343 M/B PENTIUM PER NOTEBOOK											
Cliente	Numero	Data	Scad.Ordine	Scadenza	Quantita'	Im	DL	St	SL	S	Fs
				Urgente	1						
<div style="display: flex; justify-content: space-between;"> 1. 1 (1/ 1) Esist. 0 Dispon 0 </div>											

La seconda sottobolla del secondo semilavorato e le relative due righe riferite ai quantitativi per ciascuna riga cliente.

Bolla di lavorazione 10 indice sottobolle				
Sottobolla note di lavorazione	Data	MMP	Distinta Base	CSL
Principale NOTEB	01/01/21	1	NOTEB	CS
1 M/B PENTIUM PER NOTEBOOK	01/01/21	3	SL-MBPNB4343	CS
2 SEZIONE AUDIO NOTEBOOK	01/01/21	3	SL-SCAUDIO	CS

E sono visualizzate non spezzate poiché senza impegni.

Numero	10.0002	Data	01/01/2021	Scadenza	Urgente	Piano di lavoro	No				
Magazzino PF	1 SEDE CENTRA	Magazzino MP	3 DEPOSITO B								
Nota							Escludi SML				
Codice conto							Modulo stampa	BL			
Scarico automatico	No	Valore scarico			Fissa Prezzi	No					
Inf.Prod.Finiti	Dettaglio MP+Rif.	Operazione	Scarico/Carico	Fase operazione							
Dettaglio: SL-SCAUDIO SEZIONE AUDIO NOTEBOOK											
1	Codice e descrizione prodotto	Um	Pronto	/	Lavoraz	Im	DL	St	SL	S	Fs
	SL-SCAUDIO SEZIONE AUDIO NOTEBOOK	NR			3,00						

La prima riga della seconda sottobolla riporta la relazione (1/1) riferita alla quantità 1 semilavorato.

Numero	10.0002	Data	01/01/2021	Scadenza	Urgente	Piano di lavoro	No				
Magazzino PF	1 SEDE CENTRA	Magazzino MP	3 DEPOSITO B								
Nota							Escludi SML				
Codice conto							Modulo stampa	BL			
Scarico automatico	No	Valore scarico			Fissa Prezzi	No					
Inf.Prod.Finiti	Dettaglio MP+Rif.	Operazione	Scarico/Carico	Fase operazione							
Dettaglio: SL-SCAUDIO SEZIONE AUDIO NOTEBOOK											
Cliente	Numero	Data	Scad.Ordine	Scadenza	Quantita'	Im	DL	St	SL	S	Fs
				Urgente	1,00						
				Urgente	2,00						
<div style="display: flex; justify-content: space-between;"> 1. 2 (1/ 1) Esist. 0,00 Dispon 0,00 </div>											

La seconda riga della seconda sottobolla riporta la relazione (2/2) riferita alla quantità 2 semilavorati.

Numero	10.0002	Data	01/01/2021	Scadenza	Urgente	Piano di lavoro	No
Magazzino PF	1	SEDE CENTRA		Magazzino MP	3	DEPOSITO B	
Nota							
Codice conto							
Scarico automatico	No	Valore scarico					
Inf.Prod.Finiti	Dettaglio MP+Rif.	Operazione	Scarico/Carico				
Dettaglio: SL-SCAUDIO SEZIONE AUDIO NOTEBOOK							
Cliente	Numero	Data	Scad.Ordine	Scadenza	Quantita'	Im	DL St SL S Fs
				Urgente	1,00		
				Urgente	2,00		
2. 2 (2/ 2) Esist. 0,00 Dispon 0,00							
Varia data scadenza	Modifica quantita'	Varia codice cliente	Rintrac.Doc. avanzamento PF	Raggruppamento righe	Elimina riga	Ok	Annulla

La correlazione fra gli articoli PF e SML nello sviluppo fra bolle e sottobolle è data dalla sequenza interna relativa ai valori di "olrii" e "olunp".

La ricerca di questa relazione non è però eseguibile partendo dall'alto, cioè dalle BOLLE e PRODOTTI FINITI, ma bensì partendo dal basso, cioè dalle SOTTOBOLLE e SEMILAVORATI.

Le letture tabellari, pertanto, devono processare i dati della tabella "ol" ed esportare le relazioni fra "olrii" e "olunp" e, infine, riordinarli per potere risalire alla relazione fra articoli e bolle che hanno generato queste righe.

Caso: quantità minima di variabili da azzerare per eliminare la riga (cancellazione SML sospeso dalla BL)

I semilavorati sospesi sono normali righe della bolla di lavorazione, ma con l'attributo di sospeso e non interessano i progressivi di magazzino. Per eliminarli, azzerare la seguente minima quantità di variabili:

```

_OLTPR$(D)=" " 'Tipo riga:
_OLLAV(D)=0 'Numero di sottoBL di lavorazione
_OLART$(D)=" " 'Codice PF
_OLPPP$(D)=" " 'Codice PF Padre
_OLRIF(D)=0 'Numero di riferimento
_OLRII(D)=0 'Numero riferimento impegni
_OLCBL(D)=0 'Numero sottoBL collegamento lavorante
_OLNRX(D)=0 'Numero record
    
```

IMPEGNI DI LAVORAZIONE (IL)

Le funzioni e variabili di Produzione, non sono utilizzabili dai livelli di prodotto che non sono di Produzione		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 5	Istruzione che azzerà variabili impegni di lavorazione _IL	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "IL"	Istruzione che azzerà variabili struttura _IL (come azzvarsys)	S/C/Cr/Cd Sm/Cm
PUTIL	Istruzione che scrive impegni di lavorazione	S/Cr/Cd/Sh
GETIL	Par: <ndoc>, <nsot>, <nropf>, <fase>, <mag>, <cod> Istruzione che legge impegni di lavorazione	S/Cr/Cd/Sh
GETATTIL	Istruzione che legge gli Impegni di Lavorazione a video (menu 'PB e 'PV') nel momento in cui l'istruzione viene eseguita. Esempio l'esecuzione con campo "Numero BL" vuoto, non ritorna alcun valore. Ritorna il valore delle variabili di struttura _IL caricate da Gestionale all'atto dell'utilizzo dell'istruzione; questa azzerà le variabili preesistenti ed inizializza la struttura con i valori esistenti. L'istruzione opera solamente su un singolo record (Impegno), nella revisione del Impegno la Bolla è bloccata e non modificabile. Essendo la GETATTIL un funzione mono record il valore dell'array è sempre 1. Non è realizzabile una istruzione in grado di leggere tutti gli impegni della bolla; paragonando ad emissione revisione documenti, gli impegni stanno alle anagrafiche degli articoli come la bolla di lavorazione sta al documento di magazzino; analogamente alla GETATTAR, consentita accedendo in revisione anagrafica articoli da emissione revisione documento, aprendo la revisione impegni dalla bolla di lavoro ed accedendo all'impegno è consentita la GETATTIL dell'impegno in revisione.	C
DELIL	Par: <ndoc>, <nsot>, <nropf>, <fase>, <mag>, <cod> Istruzione che cancella impegni di lavorazione	S/Cr/Cd/Sh
_ERRIL	Codice risultato operaz. Su impegni di lavorazione	S/Cd
_ERRIL\$	Messaggio operazioni su impegni di lavorazione	S/Cd
_NIL	Numero Componenti Materia Prima	S/Cd
_IL	Variabili struttura impegni di lavorazione	S/Cd
esempi e casi dizionario		
tabella "oi"	<p>Ciclo "ol" e ciclo "oi", lettura incrociata delle variabili righe di prodotti finiti e impegni di lavorazione olrii() → olrif().</p> <p>Il ciclo tabellare delle righe prodotti finiti ciclo "ol", e il ciclo tabellare sugli impegni "oi" richiedono come chiave di lettura: bolla, sottobolla, riga.</p> <p>Il numero di riga dell'archivio "ol" non è presente nel ciclo degli impegni, così come il numero di riga dell'archivio "oi" non è presente nel ciclo delle righe prodotti finiti. Questo determina che non si possono leggere in maniera diretta le variabili di un archivio ciclando sull'altro. Le soluzioni possibili sono le seguenti.</p> <p>Prima soluzione: eseguire un ciclo esempio su "ol" (righe prodotti finiti) salvando il risultato o in memoria o su disco mantenendo il riferimento degli impegni [olrii()], quindi ciclando sul file degli impegni interroghi la situazione che si è precedentemente elaborata usando come chiave di lettura il riferimento degli impegni[oirif()] presente in entrambi gli archivi. Quindi per determinare dagli impegni il pf di riferimento la chiave diventa: oinum()=olnum(), ← o variabile di appoggio oirif()=olrii() ← o variabile di appoggio se oisot() = 0 allora ollav()=32000; se oisot() >0 allora oisot()=ollav()</p> <p>Seconda soluzione: ciclando ad esempio sugli impegni tramite l'istruzione GETBL di cui si hanno i parametri, leggere la bolla interessata quindi utilizzare le variabili che interessano. Entrambe le soluzioni proposte possono essere utilizzate sugli archivi interessati nel caso della seconda soluzione il comando da utilizzare è GETIL</p>	S/C
<p>Esempio: legge gli Impegni di Lavorazione di una BL in base al numero di riferimento PF, visualizza i dati di una Materia Prima con ciclo di lettura in base _NIL e varia la quantità della MP al Rif. PF (n.1)</p> <pre> GETIL 19,0,1,1,0,"MP-ORE" : IF _ERRIL<>0 THEN GOTO ERRORE FOR I=1 TO _NIL LPRINT "NDOC.": ; : LPRINT _ILNDO(I) LPRINT "NSOT.": ; : LPRINT _ILNSD(I) LPRINT "NROFF.": ; : LPRINT _ILNRB(I) LPRINT "FASE.": ; : LPRINT _ILFAS(I) LPRINT "MAG.": ; : LPRINT _ILMAG(I) LPRINT "CODMP.": ; : LPRINT _ILAMP\$(I) NEXT IF _ILQTA(1)>2 THEN _ILQTA(1)=2.2 PUTIL : IF _ERRIL<>0 THEN GOTO ERRORE ERRORE: IF _ERRIL<>0 THEN VIMSG 4, _ERRIL\$ END </pre>		
<p>Esempio di cancellazione:</p> <p>Leggo tutti gli impegni della bolla di lavorazione.</p> <pre> GETIL 149,0,0,0,0,"" Ciclo da 1 a quanti impegni ho letto FOR I = 1 TO _NIL Cancello tutti gli impegni con la chiave che ho letto precedentemente. DELIL 149,0, _ILNRB(I), _ILFAS(I), _ILMAG(I), _ILAMP\$(I) IF _ERRIL <> 0 THEN VIMSG 5, _ERRIL\$+" DELIL" </pre>		

NEXT

Caso: le istruzioni GETIL / PUTIL ripetute in sequenza generano l'errore "Articolo MP: non trovato in archivio articoli"

L'errore si manifesta eseguendo due o più GETIL / PUTIL consecutive, durante la scrittura degli impegni, quando il numero degli impegni della scrittura attuale risulta essere numericamente inferiore a quelli dell'impegno precedente.

WorkAround: prima di eseguire la GETIL impostare la variabile _ILNDO(1) = 0

```
_ILNDO(1) = 0  
AZZVARSYS 5  
GETIL 2,0,0,0,0,""
```

PROGRESSIVI ARTICOLI (PAM)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 74	Istruzione che azzeri variabili struttura _PAM	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "PAM"	Istruzione che azzeri variabili struttura _PAM (come azzvarsys)	S/C/Cr/Cd Sm/Cm
PROGARTMAG	Par: <[codicearticolo\$],[numeromagazzino]> Istruzione che legge i progressivi articolo	S/Cr/Cd
_ERRPAM	Codice esito operazione	S/Cr/Cd
_ERRPAM\$	Descrizione esito operazione	S/Cr/Cd
_PAM	Variabili struttura Progressivi Articoli	S/Cr/Cd
esempi e casi dizionario		

Esempio: utilizzo dell'istruzione PROGARTMAG.

```

1 'la funzione puo' essere utilizzata in piu' modi:
2 ' 1) specificando un articolo ed un magazzino;
3 ' 2) specificando un articolo ed il magazzino 0 (tutti i magazzini)
4 ' 3) senza specificare un articolo preciso ("" ed indicando un magazzino
5 ' 4) senza specificare un articolo preciso ("" ed indicando il magazzino 0 (tutti i magazzini)
6 'nota bene:
7 ' 1) la funzione ritorna solo articoli movimentati con qualunque saldo
8 ' 2) il numero massimo di risultati ritornabili e' legato alle capacita' delle matrici numeriche di sprix (131040/32=4095)
9 ' 3) vengono ritornate piu' righe per lo stesso articolo per quanti magazzini e ubicazioni risultano movimentati
10 ARTICOLO$="MERCE-1" : MAGAZZINO=0
11 PROGARTMAG ARTICOLO$,MAGAZZINO
12 IF _ERRPAM<>0 THEN VIMSG 1, _ERRPAM$:END
13 IFF _NPAM<>0
14 INVENTARIO=0:CARICO=0:SCARICO=0
15 FOR A=1 TO _NPAM
16 N RIGA = ~A\
17 N _PAMCODART$ = ~_PAMCODART$(A)
18 N _PAMDESART$ = ~_PAMDESART$(A)
19 N _PAMMAG = ~_PAMMAG(A) \ \ \ \
20 N _PAMIDUBI = ~_PAMIDUBI(A) \ \
21 N _PAMUM1$ = ~_PAMUM1$(A) \ \ \
22 N _PAMUM2$ = ~_PAMUM2$(A) \ \ \
23 N _PAMINQ = ~_PAMINQ(A) \ \ \ \
24 INVENTARIO=INVENTARIO+_PAMINQ(A)
25 N _PAMINT = ~_PAM( 1) ~_PAM( 2) ~_PAM( 3) ~_PAM( 4) ~_PAM( 5) ~_PAM( 6) ~_PAM( 7) ~_PAM( 8) ~_PAM( 9) ~_PAM(10) ~_PAM(11) ~_PAM(12)
~_PAM(13) ~_PAM(14) ~_PAM(15) ~_PAM(16)
26 N ~_PAM(17) ~_PAM(18) ~_PAM(19) ~_PAM(20) ~_PAM(21) ~_PAM(22) ~_PAM(23) ~_PAM(24) ~_PAM(25) ~_PAM(26) ~_PAM(27) ~_PAM(28) ~_PAM(29)
~_PAM(30) ~_PAM(31) ~_PAM(32)
27 N _PAMTIN = ~_PAMTIN(A) \ \ \ \
28 N _PAMCIN = ~_PAMCIN(A) \ \ \ \
29 N _PAMCAR = ~_PAMCAR(A) \ \ \ \
30 CARICO=CARICO+_PAMCAR(A)
31 N _PAMCAT = ~_PAM( 1) ~_PAM( 2) ~_PAM( 3) ~_PAM( 4) ~_PAM( 5) ~_PAM( 6) ~_PAM( 7) ~_PAM( 8) ~_PAM( 9) ~_PAM(10) ~_PAM(11) ~_PAM(12)
~_PAM(13) ~_PAM(14) ~_PAM(15) ~_PAM(16)
32 N ~_PAM(17) ~_PAM(18) ~_PAM(19) ~_PAM(20) ~_PAM(21) ~_PAM(22) ~_PAM(23) ~_PAM(24) ~_PAM(25) ~_PAM(26) ~_PAM(27) ~_PAM(28) ~_PAM(29)
~_PAM(30) ~_PAM(31) ~_PAM(32)
33 N _PAMTCA = ~_PAMTCA(A) \ \ \ \
34 N _PAMCCA = ~_PAMCCA(A) \ \ \ \
35 N _PAMSCA = ~_PAMSCA(A) \ \ \ \
36 SCARICO=SCARICO+_PAMSCA(A)
37 N _PAMSCT = ~_PAM( 1) ~_PAM( 2) ~_PAM( 3) ~_PAM( 4) ~_PAM( 5) ~_PAM( 6) ~_PAM( 7) ~_PAM( 8) ~_PAM( 9) ~_PAM(10) ~_PAM(11) ~_PAM(12)
~_PAM(13) ~_PAM(14) ~_PAM(15) ~_PAM(16)
38 N ~_PAM(17) ~_PAM(18) ~_PAM(19) ~_PAM(20) ~_PAM(21) ~_PAM(22) ~_PAM(23) ~_PAM(24) ~_PAM(25) ~_PAM(26) ~_PAM(27) ~_PAM(28) ~_PAM(29)
~_PAM(30) ~_PAM(31) ~_PAM(32)
39 N _PAMTSC = ~_PAMTSC(A) \ \ \ \
40 N _PAMCSC = ~_PAMTSC(A) \ \ \ \
41 N _PAMOFOR = ~_PAMOFOR(A) \ \ \ \
42 N _PAMOFORT = ~_PAM( 1) ~_PAM( 2) ~_PAM( 3) ~_PAM( 4) ~_PAM( 5) ~_PAM( 6) ~_PAM( 7) ~_PAM( 8) ~_PAM( 9) ~_PAM(10) ~_PAM(11)
~_PAM(12) ~_PAM(13) ~_PAM(14) ~_PAM(15) ~_PAM(16)
43 N ~_PAM(17) ~_PAM(18) ~_PAM(19) ~_PAM(20) ~_PAM(21) ~_PAM(22) ~_PAM(23) ~_PAM(24) ~_PAM(25) ~_PAM(26) ~_PAM(27) ~_PAM(28) ~_PAM(29)
~_PAM(30) ~_PAM(31) ~_PAM(32)
44 N _PAMOCLIMP = ~_PAMOCLIMP(A) \
45 N _PAMOCLIMPT = ~_PAM( 1) ~_PAM( 2) ~_PAM( 3) ~_PAM( 4) ~_PAM( 5) ~_PAM( 6) ~_PAM( 7) ~_PAM( 8) ~_PAM( 9) ~_PAM(10) ~_PAM(11)
~_PAM(12) ~_PAM(13) ~_PAM(14) ~_PAM(15) ~_PAM(16)
46 N ~_PAM(17) ~_PAM(18) ~_PAM(19) ~_PAM(20) ~_PAM(21) ~_PAM(22) ~_PAM(23) ~_PAM(24) ~_PAM(25) ~_PAM(26) ~_PAM(27) ~_PAM(28) ~_PAM(29)
~_PAM(30) ~_PAM(31) ~_PAM(32)
47 N _PAMOCLDAIM = ~_PAMOCLDAIM(A)
48 N _PAMOCLDAIMT = ~_PAM( 1) ~_PAM( 2) ~_PAM( 3) ~_PAM( 4) ~_PAM( 5) ~_PAM( 6) ~_PAM( 7) ~_PAM( 8) ~_PAM( 9) ~_PAM(10) ~_PAM(11)
~_PAM(12) ~_PAM(13) ~_PAM(14) ~_PAM(15) ~_PAM(16)
49 N ~_PAM(17) ~_PAM(18) ~_PAM(19) ~_PAM(20) ~_PAM(21) ~_PAM(22) ~_PAM(23) ~_PAM(24) ~_PAM(25) ~_PAM(26) ~_PAM(27) ~_PAM(28) ~_PAM(29)
~_PAM(30) ~_PAM(31) ~_PAM(32)
50 N _PAMOCLAUTO = ~_PAMOCLAUTO(A)
51 N _PAMOCLAUTOT = ~_PAM( 1) ~_PAM( 2) ~_PAM( 3) ~_PAM( 4) ~_PAM( 5) ~_PAM( 6) ~_PAM( 7) ~_PAM( 8) ~_PAM( 9) ~_PAM(10) ~_PAM(11)
~_PAM(12) ~_PAM(13) ~_PAM(14) ~_PAM(15) ~_PAM(16)
52 N ~_PAM(17) ~_PAM(18) ~_PAM(19) ~_PAM(20) ~_PAM(21) ~_PAM(22) ~_PAM(23) ~_PAM(24) ~_PAM(25) ~_PAM(26) ~_PAM(27) ~_PAM(28) ~_PAM(29)
~_PAM(30) ~_PAM(31) ~_PAM(32)
53 NEXT
54 N INVENTARIO: ~INVENTARIO \ \ \ \ CARICO: ~CARICO \ \ \ \ \ \ \ \ \ \ SCARICO: ~SCARICO \ \ \ \ \ \ \ \ \ \
55 ELSEF
56 VIMSG 1,"NON SONO STATI TROVATI ARTICOLI CON I PARAMETRI SPECIFICATI"
57 ENDF

```

CONDIZIONI DOCUMENTO (PZD)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 58	Istruzione che azzerà le variabili della struttura _PZD	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "PZD"	Istruzione che azzerà le variabili della struttura _PZD (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETCONDDOC	Par: <tipo>, <codiceconto\$>, <codart\$>, <datadoc\$>, <sigladoc\$>, <qta>, <coeff>, <valuta>, <listino>, <prezzo>, <sconto\$>, <catsconto>, <catprov>, <extra\$> Istruzione che ritorna prezzo, sconto e provvigione in base ai parametri documento.	S/C/Cr/Cd/Sm
_ERRPZD	Codice esito operazione	S/C/Cr/Cd/Sm
_ERRPZD\$	Descrizione esito operazione	S/C/Cr/Cd/Sm
_PZD	Variabili struttura Condizione Documenti	S/C/Cr/Cd/Sm
esempi e casi dizionario		
<p>Esempio: utilizzo dell'istruzione GETCONDDOC.</p> <pre> TIPO=1 CLIENTE\$="501.20" ARTICOLO\$="ACQUA" DATADOC\$=sxdat DOCUMENTO\$="BC" QTA=100 LISTINO=0 GETCONDDOC TIPO,CLIENTE\$,ARTICOLO\$,DATADOC\$,DOCUMENTO\$,QTA,0,0,LISTINO,0,"",0,0,"" IF _ERRPZD<>0 THEN VIMSG 1,STR\$(_ERRPZD,0,0)+" "+_ERRPZD\$:END IF TIPO=1 THEN LPRINT "PREZZO: "; LPRINT _PZDPRZ' = prezzo IF TIPO<3 THEN LPRINT "SCONTO: "; LPRINT _PZDSCOS' = sconto LPRINT "----- NO MULTIAGENTE -----" LPRINT "PROVVIGIONE: "; LPRINT _PZDPRO' = Provvigione LPRINT "----- SI MULTIAGENTE -----" FOR X= 1 TO 5 LPRINT "RIGA "; LPRINT X LPRINT " _PZDPROTP\$("; LPRINT X; LPRINT ");"; LPRINT _PZDPROTP\$(X) ' = Tipo provvigione, %(percentuale) oppure T (totale) LPRINT " _PZDPROFORM\$("; LPRINT X; LPRINT ");"; LPRINT _PZDPROFORM\$(X) ' = Formula provvigione, valida solo per primo elemento LPRINT " _PZDPROAGES("; LPRINT X; LPRINT ");"; LPRINT _PZDPROAGES(X) ' = Codice agente LPRINT " _PZDPROQUOTA("; LPRINT X; LPRINT ");"; LPRINT _PZDPROQUOTA(X) ' = Quota provvigione ripartita, totale quote = 100% LPRINT " _PZDPROCOND("; LPRINT X; LPRINT ");"; LPRINT _PZDPROCOND(X) ' = Condizione agente, solo primo elemento LPRINT " _PZDPROMOD\$("; LPRINT X; LPRINT ");"; LPRINT _PZDPROMOD\$(X) ' = Mod. provvigione (R/P), solo primo elemento LPRINT " _PZDPROBIMP\$("; LPRINT X; LPRINT ");"; LPRINT _PZDPROBIMP\$(X) ' = Base imp. di calcolo provvigione: 1 lordo, 2 netto NEXT END </pre>		

LISTE DI PRELIEVO (LST)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 75	Istruzione che azzeri variabili struttura _LST	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "LST"	Istruzione che azzeri variabili struttura _LST (come azzvarsys)	S/C/Cr/Cd Sm/Cm
DELLSTDOC	Par: <idlista>, <sigladoc\$>, <sottoazienda>, <serie>, <numdoc>, <rigaordine>, <progressivopick> Istruzione che cancella i dati delle liste di prelievo	S/Cr/Cd
GETLSTDOC	Par: <idlista>, <sigladoc\$>, <sottoazienda>, <serie>, <numdoc>, <rigaordine>, <progressivopick>, [param\$] Istruzione che legge i dati delle liste di prelievo	S/Cr/Cd
PUTLSTDOC	Istruzione che scrive i dati delle liste di prelievo	S/Cr/Cd
PUTLSTDOC_EXT	Par: [param\$] Istruzione che scrive i dati delle liste di prelievo	S/Cr/Cd
_ERRLST	Codice esito operazione	S/Cr/Cd
_ERRLST\$	Descrizione esito operazione	S/Cr/Cd
_LST	Variabili struttura Liste di Prelievo	S/Cr/Cd
_NLSTR	Variabile contenente il numero di record inseriti in struttura _LST	S/Cr/Cd
esempi e casi dizionario		
Esempio: Creazione di una lista di prelievo		
<pre> AZZVARSYS 75 _LSTTDESC\$="Creata da sprix il "+STR\$(sxdag,0,0)+"/"+STR\$(sxdam,0,0)+"/"+STR\$(sxdaa,0,0)+" "+sxsor _LSTTIPO\$ = "V" 'VALORI AMMESSI: (V)endita _LSTTSTATO\$ = "C" 'VALORI AMMESSI: (C)ostruzione, (D)a elaborare, (L) in lavorazione, (E)laborata, (T)erminata, (A)nnullata PUTLSTDOC IF _ERRLST THEN VIMSG 1, _ERRLST\$ ELSE VIMSG 1, "CREATA LISTA N. "+STR\$(_LSTTID,0,0) END </pre>		
Esempio: Associazione di una riga ordine cliente ad una lista di prelievo		
<pre> LISTA=1 '0 IN FASE DI CREAZIONE LISTA, OPPURE 1-8000000 per operazioni di update su lista esistente SERIE=1 ORDINE=42 AZZVARSYS 75 _LSTTID = LISTA _LSTTDESC\$ = "Aggiornata da Sprix il "+STR\$(sxdag,0,0)+"/"+STR\$(sxdam,0,0)+"/"+STR\$(sxdaa,0,0)+" "+sxsor RIGA = 1 _LSTRSIG\$(RIGA) = "OC" 'SIGLADOC\$ _LSTRSAZ\$(RIGA) = 0 'SOTTOAZIENDA _LSTRSER\$(RIGA) = SERIE 'SERIE _LSTRNUM\$(RIGA) = ORDINE 'NUMERO DOCUMENTO _LSTRNRCRE\$(RIGA) = 1 'NUMERO DI CREAZIONE RIGA IN ORDINE - RECUPERABILE PER GLI OC DA ocrig() - _LSTRPROG\$(RIGA) = 0 '-1 PICK (DA NON USARE IN QUESTO CASO), 0 nuova RIGA ORDINE, OPPURE NUMERO>0 PER UPDATE RIGA PICK ESISTENTE PUTLSTDOC IF _ERRLST THEN VIMSG 1, _ERRLST\$ ELSE VIMSG 1, "AGGIUNTO RIGA ORDINE A LISTA "+STR\$(LISTA,0,0) END </pre>		
Esempio: Cancellazione righe ordine cliente da una lista di prelievo		
<pre> LISTA=1 '0 IN FASE DI CREAZIONE LISTA, OPPURE 1-8000000 per operazioni di update su lista esistente DIM VALORI(,4) 'SERIE, NUMERO, RIGA DI CREAZIONE ORDINE , PROGRESSIVO DI RIGA ORDINE (ERDOC) DIM TIPO_DOC\$() 'OC O OX AZZVARSYS 75 ' GETLSTDOC LISTA, SIGLADOC\$, SOTTOAZIENDA,SERIE,NUMERO,RIGAORDINE, PROGRESSIVO ' Con PROGRESSIVO = 0 Tutte le righe (ordine e pick) ' Con PROGRESSIVO = -1 solo righe ordine originali (quelle con prog = 0) ' Con PROGRESSIVO = -2 solo righe inserite (quelle con prog > 0) GETLSTDOC LISTA, "", 0,0,0,0, -1 IF _ERRLST THEN VIMSG 1, _ERRLST\$:END 'RECUPERO LE INFORMAZIONI ORDINE IFF _NLSTR>0 FOR RIGA=1 TO _NLSTR TIPO_DOC\$(RIGA) = _LSTRSIG\$(RIGA) VALORI(RIGA,1) = _LSTRSER\$(RIGA) VALORI(RIGA,2) = _LSTRNUM\$(RIGA) NEXT 'ATTENZIONE, LA CANCELLAZIONE OPERA SU RIGHE ORDINI E PICK. SE I PARAMETRI NON VENGONO IMPOSTATI 'IN MODO PUNTUALE LA CANCELLAZIONE OPERA SU TUTTE LE RIGHE RIENTRANTI NEI VALORI DEI PARAMETRI IMMESSI 'NE CONSEGUENZE CHE: ' DELLSTDOC LISTA, "", 0,0,0,0,0 CANCELLA TUTTE LE RIGHE ORDINI ASSOCIATI ALLA LISTA ' DELLSTDOC LISTA, TIPO_DOC\$(1),0,VALORI(1,1),VALORI(1,2),0,0 ESEGUE UNA CANCELLAZIONE PUNTUALE DELLSTDOC LISTA, TIPO_DOC\$(1),0,VALORI(1,1),VALORI(1,2),0,0 'CANCELO TUTTE LE RIGHE DEL PRIMO ORDINE ASSOCIATO ALLA LISTA IF _ERRLST THEN VIMSG 1, _ERRLST\$ ELSE VIMSG 1,"CANCELLAZIONE ESEGUITA" ELSEIF VIMSG 1,"LA VARIABILE _NLSTR VALE "+STR\$(_NLSTR,0,0) ENDF END </pre>		
Esempio: Scrittura righe di pick di tutte le righe ordini cliente inserite in una lista di prelievo		
<pre> LISTA=1 '0 IN FASE DI CREAZIONE LISTA, OPPURE 1-8000000 per operazioni di update su lista esistente AZZCATVARSYS "LST" GETLSTDOC LISTA, "OC", 0,0,0,0, -1 'LEGGO TUTTE LE RIGHE ORDINI PRESENTI IN LISTA IF _ERRLST THEN VIMSG 1, _ERRLST\$:END </pre>		

AUTORIZZAZIONI UTENTE (AU)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 46	Istruzione che azzera variabili struttura _AU	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "AU"	Istruzione che azzera variabili struttura _AU (come azzvarsys)	S/C/Cr/Cd Sm/Cm
GETAUTORIZUTE	Par: <[utente\$],[idutente]>,<filtro\$>,[filtro\$] Istruzione che legge autorizzazioni sui dati dell'azienda	S/Cr/Cd/Sm/Cm
_ERRAU	Codice esito operazione	S/Cd
_ERRAU\$	Descrizione esito operazione	S/Cd
_AU	Variabili struttura Autorizzazioni Utente	S/Cd
esempi e casi dizionario		
<p>Esempi GETAUTORIZUTE:</p> <pre>'Lettura del filtro cliente impostato sul campo agente AZZVARSYS 46 GETAUTORIZUTE "UTENTE",0,"_PCAGE" IF _ERRAU <> 0 THEN VIMSG 4, "Errore: " + _ERRAU\$: END VIMSG 4, "Filtro su Agente: " + _AUVAL\$_AUNUMVAL) + " anche vuoti: " + _AUNONOIMPS 'Lettura dei filtri serie documenti AZZVARSYS 46 GETAUTORIZUTE "UTENTE",0,"_MMSIG" IF _ERRAU <> 0 THEN VIMSG 4, "Errore: " + _ERRAU\$: END FOR I = 1 TO _AUNUMVAL VIMSG 4, "Filtro documento: " + _AUVAL\$(I) NEXT</pre>		

IMPORT MODULI (IMOD)

Complesso di istruzioni e variabili Sprix per importare dati da file ASCII mediante definizioni di aree testo oggetto di import; l'insieme delle definizioni di aree compone il formato del modulo per l'esecuzione dell'import. La definizione guidata delle aree testo per generare il formato è consentita tramite la funzione video [SH+F6] dell'editor Sprix (apre due voci fra cui scegliere quella dal nome "EDITOR MODULISTICA"). In Sprix si devono indicare le caratteristiche del documento e le aree da cui prelevare i dati ed i campi d'assegnazione; al documento va indicato il percorso (anche standard) ed il nome del file da importare.

Dal menu SERVIZI \ PERSONALIZZAZIONI tramite "Modelli Importazione Moduli" si possono generare programmi SPRIX che si aggiungono automaticamente a quelli già installati dall'utente (per chi non volesse definire e strutturare il modulo e la generazione del sorgente Sprix relativo attraverso l'editor Sprix). La funzione di Importazione Moduli prevede la voce Tipi sostituzioni per stabilire una tabella di correlazione fra valori dove fosse necessario; anche in Sprix è prevista questa funzione, passando dall'EDITOR MODULISTICA [SH+F6] e premendo [SH+F3] sull'area selezionata. Per approfondimenti si rimanda alla relativa sezione del Manuale Generale dell'utente.

Per istruire in un sorgente Sprix un modulo di importazione sono disponibili delle istruzioni specifiche. Alcune istruzioni sono la MODOPEN, la MODCLOSE, la MODGET e la MODFIELD, queste istruzioni sono riconosciute e interpretate dall'editor di importazione moduli in modo da permettere anche una personalizzazione, entro certi limiti, del sorgente Sprix senza dover riscrivere o reimpostare le istruzioni del modulo di importazione. Per la descrizione dettagliata di queste nuove istruzioni si rimanda al manuale SPRIX/COLLAGE.

Note alla scrittura del codice Sprix

L'editor moduli di importazione riconosce e gestisce un solo modulo all'interno di un sorgente Sprix. La modifica del codice generato deve essere fatta da tecnici esperti e con cognizione di causa sapendo che l'editor potrebbe in alcuni casi non considerare delle istruzioni ed escluderle dal codice Sprix quando lo rigenera. Istruzioni personalizzate inserite nelle stesse righe delle istruzioni sopra indicate non saranno riscritte dall'editor e andranno perse.

Se nelle istruzioni MOD?? sono passate delle variabili o espressioni come argomenti, l'editor si rifiuterà di interpretare il modulo. E' escluso da questo limite il nome del file nell'istruzione MODOPEN alla quale si può passare una variabile, ma non un'espressione.

La variabile _MODSCRIVI regola la scrittura dei dati, quando vale 1 indica che il record dati può essere scritto. Normalmente la _MODSCRIVI vale 1 ad ogni fine pagina oppure ad ogni fine riga logica corpo. Nel caso in cui siano stati definiti dei campi di rottura il valore della _MODSCRIVI sarà legato al cambiamento dei valori di questi campi, se si definiscono più campi di rottura basta che uno solo di essi cambi per far impostare a 1 la _MODSCRIVI e quindi fare scrivere il record. Se i campi di rottura sono su righe diverse l'eventuale cambiamento di uno dei valori verrà segnalato con la lettura della riga che precede il primo campo definito. La variabile _MODSCRIVI e di lettura e scrittura e quindi il programmatore SPRIX può personalizzare il suo funzionamento secondo le proprie esigenze.

Non consentita nel Livello di sviluppo (F2)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
MODOPEN	Par: <tipo\$>, <nomefile\$>, <archivi\$>, <opzioni\$>, <righe>, <colonne>. Istruzione che apre il file test relativo al modulo Non consentita nel Livello di sviluppo (F2)	S/Cd
MODCLOSE	Istruzione che chiude il file testo relativo al modulo Non consentita nel Livello di sviluppo (F2)	S/Cd
MODFIELD	Par: <opz\$>, <FROM row TO...>, <FROM col TO... AS varbuf [tipoiimp]>, ... Istruzione che definizione variabili e aree Non consentita nel Livello di sviluppo (F2)	S/Cd
MODBFIELD	Par: <opz\$>, <FROM row TO...>, <FROM col TO... AS varbuf [tipoiimp]>, ... Istruzione che definizione variabili e aree (MODBODY) Non consentita nel Livello di sviluppo (F2)	S/Cd
MODBODY	Par: <opzioni\$>, <FROM rowstart TO rowend>, <rows> Istruzione che gestione corpo pagina Non consentita nel Livello di sviluppo (F2)	S/Cd
MODGET	Istruzione che legge dati e assegna variabili Non consentita nel Livello di sviluppo (F2)	S/Cd
MODPUT	Par: <archivi\$>. Istruzione che scrive variabili negli archivi Non consentita nel Livello di sviluppo (F2)	S/Cd
MODSFIELD	Par: <opz\$>, <varbufori [tipoiimp]>, <nomefile\$>, <obbligo>, <varbufdesori\$> Istruzione che associa a campo la tabella sostituzioni Non consentita nel Livello di sviluppo (F2)	S/Cd
_MSGMOD\$()	Par: (). Vettore di messaggi durante l'elaborazione (v. MODVIMSG)	S/Cd
MODVIMSG	Istruzione visualizza messaggi fine elaborazione modulo (v. _MSGMOD\$) Non consentita nel Livello di sviluppo (F2)	S/Cd
_MOD	Variabili di Import Moduli	S/Cd
_MSC	Variabili selezione campi (Saldi Contabili)	S/Cd
PUTSC	Istruzione che scrive/cancella Primanota Non consentita nel Livello di sviluppo (F2)	S/Cr/Cd
_SC	Variabili di struttura per PUTSC	S/Cd
_ERRSC	Codice errore istruzione PUTSC	S/Cd
_ERRSC\$	Descrizione dell'errore	S/Cd
_ERRMOD	Codice esito dell'operazione di elaborazione modulo	S/Cd
_ERRMOD\$	Descrizione del codice esito (in fase MODCLOSE)	S/Cd

esempi e casi

Esempio IMOD-SPRIX per Saldi Contabili

```
'IMOD-SPRIX: apertura modulo MODOPEN "S", "<c:\tmp\pdcscl", "sc", "", 1, 131
'IMOD-SPRIX: definizione campi pagina
MODFIELD "", FROM 1 TO 1, FROM 2 TO 10 AS MSCCOD
```

```
MODFIELD "", FROM 1 TO 1, FROM 16 TO 51 AS _MSCDESC$
MODFIELD "", FROM 1 TO 1, FROM 54 TO 64 AS _MSCINIZ NUM2
MODFIELD "", FROM 1 TO 1, FROM 67 TO 77 AS _MSCDARE NUM2
MODFIELD "", FROM 1 TO 1, FROM 80 TO 90 AS _MSCAVER NUM2

`IMOD-SPRIX: definizione sostituzioni
MODSFIELD "", _MSCCOD$, "CONTI", "S"

`IMOD-SPRIX: inizio ciclo di import dati
IMOD_CICLORIGHE:

  `IMOD-SPRIX: lettura file
  MODGET `IMOD-SPRIX

  `IMOD-SPRIX: chiamata MODPUT per scrittura dati
  IF _ERRMOD = 0 AND _MODSCRIVI = 1 THEN MODPUT "sc" `IMOD-SPRIX

  REM --- IMOD:controllo errore e uscita
  IF _ERRMOD <> 0 THEN _MODTOTRIG = 0 `IMOD-SPRIX
  IF _MODNUMRIG < _MODTOTRIG THEN GOTO IMOD_CICLORIGHE `IMOD-SPRIX

`IMOD-SPRIX: chiusura modulo
MODCLOSE
`IMOD-SPRIX: messaggi di elaborazione
MODVIMSG `IMOD-SPRIX
`IMOD-SPRIX: messaggio errore
IF _ERRMOD <> 0 THEN VIMSG 5, _ERRMOD$ `IMOD-SPRIX
```

ELENCO STATO APP(APPSTS)

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 76	Istruzione che azzerava le variabili della struttura _APPSTS	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "APPSTS"	Istruzione che azzerava le variabili della struttura _APPSTS	S/C/Cr/Cd Sm/Cm/Sh
ELENCSTATOAPP	Funzione che legge l'elenco delle APP per l'azienda aperta	S/C/Cr/Cd
_APPSTS	Variabili struttura Elenco APP	S/C/Cr/Cd
_NAPPSTS	Variabile contenente il numero di record inseriti in struttura _APPSTS	S/C/Cr/Cd

Esempi e casi dizionario

```

AZZVARSYS 76
ELENCSTATOAPP
IFF _NAPPSTS>0
FOR I = 1 TO _NAPPSTS
  LPRINT _APPSTSCOD$(I) " " + _APPSTSNOMES$(I) " " + _APPSTSAZIES$(I)
NEXT
ENDF
END

```

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
PARAM_REDD	Par: <sog\$>,[mod\$]. Istruzione e ciclotabellare per Contratti Locazione	S/C/Cr/Cd
_ERRREDSPX	Codice errore istruzione PARAM_REDD	S
_ERRREDSPXS\$	Descrizione dell'errore PARAM_REDD	S
ON_ERRMSG:	Evento all'uscita della finestra dei messaggi differiti	C/Cd
Esempi e casi dizionario		
<pre> PARAM_REDD "D", "MOD=730" IF _ERRREDSPX OR _ERRREDSPXS\$<>" THEN VIMSG 4, STR\$(_ERRREDSPX, 0, 0) + " " + _ERRREDSPXS\$: END PRA\$="ALE": INT\$="1" +-----+ Cod.tab.:rd REDDITI - Modello 730 QUADRO GE +-----+ rd N Cognome:~rd730_ge_cognd() Comune Nascita:~rd730_an_ccnas() Accesso Diretto:~PRA\$--INT\$:Cognome:~rd730_ge_cognd(PRA\$,INT\$)\\\\ </pre>		

		S/C/Cr/Cd/Sm Cm/Ca/Sh/Cc
AZZVARSYS 35	Istruzione che azzerà variabili _MYDB (non azzerà _MYDBK\$())	S/C/Cr/Cd Sm/Cm/Sh
AZZCATVARSYS "MYDB"	Istruzione che azzerà variabili struttura _MYDB (come azzvarsys)	S/C/Cr/Cd Sm/Cm
_ERRMYDB	Valore numerico errore istruzione MyDB	S/C/Cd
_ERRMYDB\$	Messaggio errore istruzione MyDB	S/C/Cd
MYDBGET	Par: <Archivio\$>,<NumeroKey>,<Numero ID>,[Parametri\$] Istruzione che legge un'anagrafica	S/C/Cr/Cd
MYDBPUT	Par: <Archivio\$>,[Parametri\$]. Istruzione che scrive un'anagrafica	S/C/Cr/Cd
MYDBDEL	Par: <Archivio\$>,[Parametri\$]. Istruzione che cancella un'anagrafica	S/C/Cr/Cd
MYDBPREV	Par: <Archivio\$>,<NumeroKey>,[Parametri\$] Istruzione che posiziona sull'anagrafica precedente	S/C/Cr/Cd
MYDBNEXT	Par: <Archivio\$>,<NumeroKey>,[Parametri\$] Istruzione che posiziona sull'anagrafica successiva	S/C/Cr/Cd
MYDBVIEW	Par: <Archivio\$>,<NumeroKey>,<NumeroID>,[Parametri\$]. Istruzione di finestra di ricerca di un archivio	S/C/Cr/Cd
MYDBINPUT	Par: <Archivio\$>,<NumeroID>,[Parametri\$] Istruzione di input a finestra di un'anagrafica	S/C/Cd
MYDBGETATT	Istruzione di lettura dell'anagrafica corrente (solo Collage)	C/Cd
MYDBPUTATT	Par: [Parametri\$] Istruzione di scrittura a video dell'anagrafica corrente (solo Collage)	C/Cd
MYDBRIORG	Par: <Archivio\$>,[Parametri\$]. Istruzione di riorganizzazione di un archivio	S/Cd
MYDBLOF()	Par: (<Archivio\$>,[Parametri\$]). Funzione che ritorna il numero di record inseriti nell'anagrafica	S/C/Cr/Cd
_MYDB	Variabili struttura archivi e tabelle MyDB	S/C/Cd
PRODUTTORE_ARCHIVIO	Nomi archivi MyDB	S/C/Cd
ARCHIVIO	Nomi archivi MyDB in PassBuilder	S/C/Cd
CODICEAPP_ARCHIVIO	Nomi archivi MyDB in PassBuilder richiamati dall'esterno	S/C/Cd
DM	Prefisso Sezioni Documenti	S/C/Cd
PA	Prefisso Sezioni Parcelle	S/C/Cd
ON_STARTMYDB	All'ingresso anagrafica (prima di apertura input anagrafico o tabella)	C/Cd
ON_STARTMYDBRIC	All'ingresso della lista dei record visualizzati	C/Cd
ON_STARTFILTRMYDB	All'ingresso del filtro di un'anagrafica	C/Cd
ON_CTRLMYDB	All'uscita di un'anagrafica	C/Cd
ON_CTRLFILTRMYDB	All'uscita del filtro di un'anagrafica dopo essere stati fatti i controlli e prima della ON_ENDFILTRMYDB	C/Cd
ON_ENDMYDB	All'uscita di un'anagrafica	C/Cd
ON_ENDFILTRMYDB	All'uscita del filtro di un'anagrafica	C/Cd
ON_PUTMYDB	Dopo la scrittura di un'anagrafica e prima della ON_ENDMYDB. Scatta in tutti i punti dove effettivamente il dato viene scritto, tranne nei documenti (in cui avviene tutto in memoria fino alla conferma del documento).	C/Cd
ON_DELMYDB	Dopo la cancellazione di un'anagrafica e prima della ON_ENDMYDB	C/Cd
ON_IN_YMYD001	All'ingresso del campo con codice 001 (Rif. colonna codice della definizione)	C/Cd
ON_OUT_YMYD001	All'uscita dal campo con codice 001 (Rif. colonna codice della definizione)	C/Cd
ON_IN_YMYDZZZ	All'ingresso del campo estensione (Rif. colonna codice della definizione)	C/Cd
ON_OUT_YMYDZZZ	All'uscita dal campo estensione (Rif. colonna codice della definizione)	C/Cd
ON_OUT_MDSELTEST	Evento su finestra [Shift+F8] di interrogazione dati MyDB di testata di un documento trasformato (esempio una bolla nata da più ordini)	C/Cd
<p>Note: ON_CTRLFILTRMYDB e ON_CTRLMYDB sono richiamate dopo i controlli del programma, ma prima della ON_ENDMYDB; sono attivi sia MYDBGETATT che MYDBPUTATT e la variabile _TF che contiene il codice del tasto di uscita. Se si modifica _TF o si esegue la MYDBPUTATT, il programma rientra in anagrafica, per visualizzare le modifiche e rieseguire i controlli sull'anagrafica. Se si imposta _TF con un tasto valido questo viene elaborato.</p>		
MYDBTAB	Par: <Archivio\$>,[Parametri\$]. Istruzione da impostare prima del ciclotabellare per dichiarare quale tabella elaborare	S/Cr/Cd

La gestione MyDB è autonoma e collegata in chiave. Non dispone degli stessi automatismi di altri moduli standard (Primanota, Magazzino, Produzione, Controllo di Gestione, ecc.). Le singole App di PassBuilder sono reciprocamente indipendenti (non si influenzano fra di loro), pertanto, non è consentito accedere agli archivi MyDB di altre App.

Notare bene:

- MyDB non è gestito dal modulo Produzione, eventuali archivi possono essere riportati nei documenti CL, SL, DL solo attraverso la procedura standard dal modulo Magazzino.
- Se si vuole modificare un record MyDB già esistente, deve essere valorizzato in modo corretto il campo _MYDBID (consigliata una preventiva MYDBGET)
- la scrittura di un MyDB CONTROLLA e BLOCCA l'inserimento di eventuali chiavi doppie, sia la chiave primaria ID che la chiave 1 se espressamente richiesto nella definizione (a meno che non ci siano chiavi rovinare, con necessità di riorganizzazione una tantum).
La gestione di Mydb incrociata con emissione/revisione documenti ha delle particolarità che possono essere riassunte in alcune seguenti regole:
 - tutte le manipolazioni dei Mydb devono essere fatte solo dopo la PUTMM

- verificare sempre gli errori che ritornano le funzioni MyDB perché capita spesso di pensare di manipolare un record ed invece ci si è posizionati su un altro.
- assicurarsi che la chiave che si riferisce al documento che si sta scrivendo sia corretta, perché se si fa riferimento ad un documento diverso ma esistente il programma non lo controlla.
- Nella funzione 'Emissione/revisione documenti' all'apertura del documento tutti i mydb legati a quel documento vengono caricati in memoria; al salvataggio del documento vengono fisicamente cancellati tutti i record e riscritti, tutto quello che avviene su disco nel mentre (MYDBGGET e MYDBPUT) non viene tenuto in considerazione. Alla luce di questo tutto quello che deve essere mantenuto con collage deve essere fatto all'evento ON_PUTMM. Cosa diversa, invece, se si è nella videata del dato aggiuntivo: in questo caso si può agire sui campi presenti in videata con le istruzioni MYDBGGETATT e MYDBPUTATT

MYDBVIEW: se si seleziona un record nella MYDBVIEW vengono automaticamente impostate le variabili come se si fosse eseguita una MYDBGGET quindi la struttura _MYDB\$(). Per mantenere il record deve utilizzare la MYDBINPUT.

Dimensioni:

All'interno di ogni singola anagrafica MyDB il numero massimo di record gestibili, lunghi 1byte, è di 2.147.483.647. Questo numero si può ridurre, però, in base ai limiti dettati dal sistema operativo. Esempio, se il sistema operativo è a 32bit, incontra il limite della dimensione del file che non può superare i 2Gb.

Il numero di record gestibili è inversamente proporzionale alla dimensione del record, più è grande il record e meno sono i record gestiti. Non ci sono limiti sul numero dei campi che può inserire.

NOTARE BENE: MyDB è un'eccezione rispetto alla regola generale secondo cui Collage può solamente leggere a video (tramite l'istruzione GETATT) e non può eseguire le istruzioni GET e PUT (salvo farlo tramite Sprix eseguito attraverso SHELLSPRIX). In Collage, infatti, MyDB consente di eseguire direttamente le istruzioni GET e PUT.

L'archivio MyDB può essere solamente gestito o disabilitato, pertanto non consente la possibilità in "sola lettura". Collage consente comunque in controllo dell'utente in fase di apertura archivio e, se necessario, forza il tasto Esc (505) oppure intercetta la conferma (610) e la converte in Esc (505), oppure sono disponibili altri metodi per bloccare l'utente (es. _TF=0).

MYDBGGET potrebbe rilasciare il messaggio "**record non trovato**" quando si stanno eseguendo le prime prove Sprix su un'archivio MyDB non ancora avviato, ma soprattutto quando viene sbagliata la chiave. Per esempio, documenti di tipo Ordine non prevedono la presenza del codice cliente e la cui presenza fa rilasciare l'errore.

Il messaggio può essere rilasciato anche in circostanze apparentemente normali. Esempio: la tabella MYDB associata a righe documenti FT, BC, BF può rilasciare il messaggio _MYDBERR\$ "Record Non Trovato" per una MYDBGGET su campi _MYDB\$(...) comunque correttamente popolati. Potrebbe sembrare una situazione "occasionale", ma in realtà può essere normale ed è per questo che va sempre essere testata la variabile _ERRMYDB\$, in quanto potrebbe non trovare dati MyDB quando in realtà ci sono e sono visualizzati! Se la chiave fosse sbagliata si avrebbe lo stesso problema su tutte le righe e non solo su alcune, perché? Quale è il motivo di questo comportamento?

Il principio generale è che viene eseguita la lettura di un record passando una determinata chiave: se il record non viene trovato (con quella chiave) viene dato l'errore di 'Record non trovato'; **tutte le variabili, però, non risulteranno vuote, ma verranno valorizzate con il primo record successivo trovato, a meno che il file non sia vuoto.** Ne consegue che è vero che trova dei dati nelle varie variabili, ma non sono quelli che avrebbe voluto leggere! È fondamentale, quindi, non solamente costruire una chiave _MYDBK\$() valida e coerente, ma anche considerare che il 'progressivo riga' non necessariamente corrisponde al numero di riga che viene vista a video, ma equivale alla variabile **_MMNCRERIGA(N)**. Le due numerazioni possono differire, ad esempio, se viene cancellata una riga oppure vengono aggiunte righe in mezzo ad altre già esistenti o ancora se vengono accorpate più documenti. Il documento di riferimento è sempre l'ultimo trasformato letto con GETMM. La numerazione delle righe e delle testate invece è più articolata. Es.

BC/1	BC/2	FT/1
ARTICOLO A - TESTATA 1 - RIGA 1 ARTICOLO B - TESTATA 1 - RIGA 2	ARTICOLO C - TESTATA 1 - RIGA 1 ARTICOLO D - TESTATA 1 - RIGA 2	ARTICOLO A - TESTATA 1 - RIGA 1 ARTICOLO B - TESTATA 1 - RIGA 2 ARTICOLO C - TESTATA 2 - RIGA 1 ARTICOLO D - TESTATA 2 - RIGA 2

Per la composizione della chiave _MYDBK\$(1):

```
FOR R=1 TO _NMM
KEY$="000"
KEY$=KEY$+_MMSIG$           'SIGLA DELLA FATTURA
KEY$=KEY$+_MMSER           'SERIE DELLA FATTURA
KEY$=KEY$+" "
KEY$=KEY$+RIGHT$("000000"+STR$(_MMNUM,0,0),6) 'NUMERO FATTURA
KEY$=KEY$+_MMCLI$ 'CODICE CLIENTE
```

'notare: richiede l'intestataro del documento come riportato nelle tabelle di chiavi elencate in calce alla MYDBGGET, poiché per altri tipi di documento che non sono di tipo "movimento" (ad esempio gli ordini) l'intestataro non è richiesto;

```
KEY$=KEY$+RIGHT$("00000000"+STR$(_MMNCRERIGA(R),0,0),8) 'NUMERO RIGA
KEY$=KEY$+RIGHT$("000"+STR$(_MMNTE(R),0,0),3) 'Riferimento a testate per documento trasformati
```

'Lettura con MYDBGGET:

```
AZZCATVARSYS "MYDB"
_MYDBK$(1)=KEY$
MYDBGGET "128554_MYDB3", 99,0,"DOC=FT"
IF _ERRMYDB$="" THEN VIMSG 4, _MYDB$(2) ELSE VIMSG 4, _ERRMYDB$
NEXT
```

MyDB condivisi su App diverse.

All'interno di una App è abilitata la possibilità di accedere ad anagrafiche/tabelle MyDB di altre App presenti sull'installazione. Nell'App di origine, dove sono contenute le definizioni delle anagrafiche/tabelle MyDB, occorre indicare se la tabella/anagrafica può essere gestita da altre App e in quale forma (nessuna, lettura o lettura/scrittura).

Notare: nell'App in cui si desidera utilizzare i dati MyDB, la sintassi per accedere dalle funzioni Sprix a tali archivi è:

CodiceApp@CodiceMyDB
 (esempio: 923875CUSCAREPRO@mydb1)
(esempio: 010999FATTURAPA@ANAPADOCDMTC) se legato alla testata documento magazzino
(esempio: 010999FATTURAPA@ANAPADOCDMRX) se legato alle righe documento magazzino

Attenzione, non si possono fare stampe grafiche dirette su archivi di altre App e non è possibile, all'interno di una anagrafica dell'App stessa, relazionare o estendere archivi di altre App.

Emissione Revisione Documenti

Nella funzione 'Emissione/revisione documenti' all'apertura del documento tutti i MYDB legati a quel documento vengono caricati in memoria; al salvataggio del documento vengono fisicamente cancellati tutti i record e riscritti, tutto quello che avviene su disco nel mentre (MYDBGET e MYDBPUT) non viene tenuto in considerazione. Detto questo, tutto quello che deve essere mantenuto con collage deve essere fatto all'evento ON_PUTMM. Cosa diversa se, invece, ci si trova nella videata del dato aggiuntivo; in questo caso si può agire sui campi presenti in videata con le istruzioni MYDBGETATT e MYDBPUTATT.

Chiarimento su _MYDBDIN\$().

E' un array di _MYDBELE elementi (come _MYDB\$()). Contiene il flag di riportabilità sui documenti. Relativo alle anagrafiche DINAMICHE. Con la MYDBGET l'array viene impostato a "S" per quei campi che sono stati selezionati come riportabili, mentre per gli altri campi viene riportato una stringa vuota ("")

Con la MYDBPUT si possono variare con la seguente regola:

- " " = Inalterato, non viene ne acceso ne spento, mantiene il vecchio valore
- "N" = Spento, NON viene riportato negli archivi dinamici (Accetto anche "n")
- "S" = Acceso, viene riportato negli archivi dinamici (Accetto anche "s")

Esempio:

Prendiamo un archivio della FatturaPA per la parcella, dove abbiamo abilitato 2 campi:

Fattura XML (Parcella) [V 2]	Posizione nella definizione dell'archivio
Cliente	1 (Estensione)
2.1.1.11 - Causale	2
2.1.(2/3/4).4 - Numero item	7
2.1.(2/3/4).5 - CodCommessaConvenz	8
*2.2.1.7 - Data inizio periodo	3
*2.2.1.8 - Data fine periodo	4
2.2.1.15 - Rif. amministrazione	5
2.4.2.1 - Beneficiario	6

La MYDBGET ritorna questi valori:

```
_MYDBDIN$(1)=""
_MYDBDIN$(2)=""
_MYDBDIN$(3)="S"
_MYDBDIN$(4)="S"
_MYDBDIN$(5)=""
_MYDBDIN$(6)=""
_MYDBDIN$(7)=""
_MYDBDIN$(8)=""
```

Se si impostano in questo modo:

```
_MYDBDIN$(3)=""
_MYDBDIN$(4)="N"
_MYDBDIN$(5)="S"
```

Se si esegue una MYDBPUT otterremo questo:

Fattura XML (Parcella) [V 2]

Cliente

2.1.1.11 - Causale

2.1.(2/3/4).4 - Numero item

2.1.(2/3/4).5 - CodCommessaConvenz

*2.2.1.7 - Data inizio periodo

2.2.1.8 - Data fine periodo

*2.2.1.15 - Rif. amministrazione

2.4.2.1 - Beneficiario

Il 3° elemento mantiene il dato, il 4° viene spento, ed il 5° viene acceso. Ovviamente impostare a "N" un elemento già disabilitato o a "S" uno già abilitato non comporta modifiche.

Esempi e casi dizionario

Esempio MYDBTAB per eseguire il CICLOTABELLARE in MYDB

(dato il Codice App "907692CORSOSFXM0" e l'Archivio "mydbtecn")

Caso: Sprix interno all'App contenente MyDB. Lettura ciclica di un archivio MyDB Standard o una Estensione diretta dei documenti: "ciclo tabellare yd" (archivio standard): MYDBTAB "mydbtecn"

(dato l'Archivio "mydbabcd")

Se Riportabile/Dinamico sui movimenti: "ciclo tabellare yy" (archivio su documento)
 Archivio + 4 caratteri della sezione documento
 MYDBTAB "mydbabcdDMTC", "DOC=documento"
 "DOC" vuole una sigla documento di magazzino, che serve esclusivamente ad indirizzare la lettura per lo specifico archivio.
 (nota: se richiamata da un'altra App MYDBTAB "CODICEAPP@mydbabcdDMTC", "DOC=documento")

(dato l'Archivio "mydbabcd")

Se Riportabile/Dinamico sulle parcella: "ciclo tabellare yd" (archivio su parcella)
 Archivio + 3 caratteri della sezione parcella (PAT per Testate e PAR per righe)
 MYDBTAB "mydbabcdPAT", "DOC=PA"

Se Riportabile/Dinamico sui movimenti: "ciclo tabellare yy" (archivio su documento)
 Archivio + 4 caratteri della sezione documento
 "DOC" vuole una sigla documento di magazzino, che serve esclusivamente ad indirizzare la lettura per lo specifico archivio.

Esempi di lettura ciclata e diretta dei valori anagrafici MyDB.

Realizzo due estensioni anagrafiche, ovvero una STANDARD e una RIPORTABILE, per aggiungere un campo ai Clienti (estensione PDCC). La MyDB standard si chiama MYDBNRIP e quella riportabile si chiama MYDBCLI.

La MYDBNRIP (standard) mi genera il seguente dizionario nella tabella "yd"

```
ydbnrip000 (ID MyDB)
ydbnripz01 (ANNULLATO)
ydbnripz02 (BLOCCATO)
ydbnripzzz (Clienti)
ydbnrip001 (ETICHETTA STD)
```

La MYDBCLI (riportabile) mi genera il seguente dizionario nella tabella "yd"

```
ydbcli000 (ID MyDB)
ydbcliz01 (ANNULLATO)
ydbcliz02 (BLOCCATO)
ydbclizzz (Clienti)
ydbcli001 (ETICHETTA RIP)
```

Scrivo i dati aggiuntivi nell'anagrafica del Conto Cliente "201.00030" nella "ETICHETTA STD" di MYDBNRIP inserisco il valore "StdStdStd" nella "ETICHETTA RIP" di MYDBCLI inserisco il valore "RipRipRip"

Scrivo anche un documento ordine OC n.122 per il cliente "201.00030", in cui cambio l'etichetta propositiva, ovvero cambio il valore "RipRipRip" nel valore "Riportabile".

Ho quindi inserito tutti i dati che mi servono per eseguire delle letture, per esempio, dei ciclitabellari sugli archivi anagrafici MyDB "MYDBNRIP" e "MYDBCLI".

Ciclotabellare su tabella "yd" "MYDBNRIP"

```
MYDBTAB "mydbnrip"
nel campo ydbnrip000() trova valore = 1 (record n.1, cioè lo "ID MyDB")
nel campo ydbnrip001() trova valore = "StdStdStd" (l'etichetta standard)
```

Ciclotabellare su tabella "yd" "MYDBCLI"

```
MYDBTAB "mydbncli"
nel campo ydbcli000() trova valore = 1 (record n.1, cioè lo "ID MyDB")
nel campo ydbcli001() trova valore = "RipRipRip" (l'etichetta riportabile)
```

Notare: in entrambe le tabelle, rilevo che lo ID MyDB equivale a 1, ed è la chiave che mi serve per accedere al dato di anagrafica MyDB in modo diretto, ma ciclando da altre tabelle.

Ciclotabellare su tabella Rubrica "pa" o Piano Conti "pc"

```
nel campo ydbnrip001(1) trova valore "StdStdStd"
nel campo ydbcli001(1) trova valore "RipRipRip"
```

Potrei volere fare un ciclotabellare su tabella ordini "oc" per vedere l'etichetta riportabile modificata (cioè dove ho modificato l'etichetta dalla proposizione "RipRipRip" in "Riportabile"), ma nel caso delle anagrafiche MyDB riportabili, la tabella dizionario

non è più "yd", bensì quella relativa ai documenti, ovvero "yy".

Per ciclare la tabella "yy" occorre una determinata sintassi relativa al tipo di documento, cioè testata dell' Ordine Cliente, pertanto mi serve l'estensione "DMTC". **NOTA BENE: la logica di costruzione delle chiavi è in calce alla tabella delle variabili _MYDB**

```
MYDBTAB "mydbncliDMTC", "DOC=OC"
nel campo ymydbclidmtcz00() trova valore = OC cioè il documento
nel campo ymydbclidmtcz000() trova valore = 1 (record n.1) cioè lo ID MyDB
nel campo ymydbclidmtczzzz() trova valore "0000C01.00012200000000", chiave di testata documento Cliente
nel campo ymydbclidmtcz001() trova il valore dell'etichetta modificata in "Riportabile"
```

Infine.

Adesso conosco la chiave (Sigla\$,ID) per trovare il medesimo dato anagrafico riportabile e modificato, ma in modo diretto e ciclando, invece, dalla tabella degli ordini "oc", attraverso il campo ymydbclidmtcz001("OC",1) passandogli i parametri chiave necessari, ovvero la Sigla\$ del documento e lo ID dell'anagrafica MyDB.

Esempio MYDBGET

(dato il produttore n. "1900074000" e l'archivio "MYDB2")

```
ARCHIVIO$ = "1900074000_MYDB2"
`** Lettura di un'anagrafica dell'archivio
` valorizza il vettore _MYDB$( )
` valorizza _MYDBID
` valorizza _MYDBANN$

MYDBGET "1900074000_MYDB2", 0, 7
IF _ERRMYDB$ <> "" THEN VIMSG 5,_ERRMYDB$
`** Legge per l'archivio dichiarato il record con ID = 7

_MYDBK$(1) = "501.00001"
MYDBGET "1900074000_MYDB2", 99, 0
IF _ERRMYDB$ <> "" THEN VIMSG 5,_ERRMYDB$
`** Legge per l'archivio dichiarato il primo record dove la chiave 99 è = 501.00001
```

Esempi Lettura Estensione

(dato il produttore 1901536000 e l'anagrafica/tabella "EST")

```
Lettura di una Estensione diretta delle testate sulla bolla BC1/116 del cliente 501.00001:
`Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
_MYDBK$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001" `(Riga 0 - testata 1)
MYDBGET "1901536000_ESTDOCR", 99, 0, "DOC=BC"
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$

Lettura di una Estensione diretta delle testate sulla bolla BC1/116 del cliente 501.00001 su azienda INFRANNUALE:
`Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
_MYDBK$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001"+"20160115" `(Riga 0 - testata 1)
MYDBGET "1901536000_ESTDOCR", 99, 0, "DOC=BC"
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$

Lettura di una Estensione diretta delle righe sulla bolla BC1/116 del cliente 501.00001:
`Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
_MYDBK$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000001"+"001" `(Riga 1 - testata 1)
MYDBGET "1901536000_ESTDOCR", 99, 0, "DOC=BC" `(Produttore_Anagrafica, Doc = Mov.)
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$

Lettura di una Estensione dei clienti (standard, riportabile o dinamica) sul CLIENTE 501.00001:
`Imposto il campo estensione per agganciare il cliente
_MYDBK$(1) = "501.00001"
MYDBGET "1901536000_ESTCLR", 99, 0 `(CodiceProduttore_CodiceTabella/Anagrafica)
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$

Lettura di una Estensione dei clienti (riportabile o dinamica) sulla TESTATA CLIENTI della bolla BC1/116 del cliente 501.00001:
`Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
_MYDBK$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001" `(Riga 0 - testata 1)
MYDBGET "1901536000_ESTCLRDMTC", 99, 0, "DOC=BC"
` Produttore_Anagrafica???+Sezione documento (*1 - Vedere la tabella di identificativi MyDB di testata e di riga elencati in fondo
alla tabella struttura archivi_MYDB), Doc = Movimenti
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$

Lettura di una Estensione dei clienti (riportabile o dinamica) sulla RIGA degli ARTICOLI VALORIZZATI della bolla BC1/116 del cliente
501.00001:
`Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
_MYDBK$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000001"+"001" `(Riga 1 - testata 1)
MYDBGET "1901536000_ESTCLRDMRX", 99, 0, "DOC=BC"
` Produttore_Anagrafica???+Sezione documento (*1 - Vedere la tabella di identificativi MyDB di testata e di riga elencati in fondo
alla tabella struttura archivi_MYDB), Doc = Movimenti
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$

Lettura di una Estensione degli articoli (standard o riportabile) sull'ARTICOLO 001:
`Imposto il campo estensione per agganciare l'articolo
_MYDBK$(1) = "001"
MYDBGET "1901536000_ESTARTR", 99, 0 `(CodiceProduttore_CodiceTabella/Anagrafica)
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$

Lettura di una Estensione degli articoli (riportabile) sulla RIGA della bolla BC1/116 del cliente 501.00001:
`Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
_MYDBK$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000001"+"001" `(Riga 1 - testata 1)
MYDBGET "1901536000_ESTARTRDMRX", 99, 0, "DOC=BC"
` Produttore_Anagrafica???+Sezione documento (*1 - Vedere la tabella di identificativi MyDB di testata e di riga elencati in fondo
alla tabella struttura archivi_MYDB), Doc = Movimenti
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
```

Esempio MYDBPUT

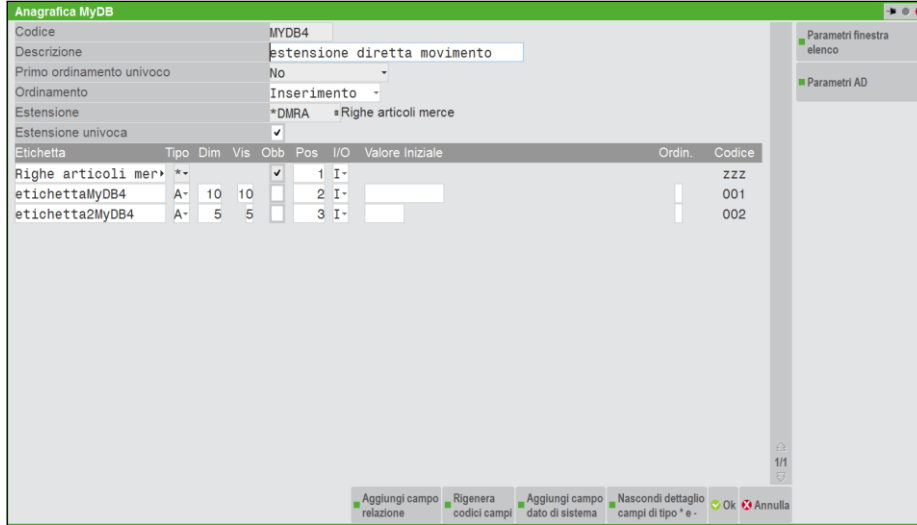
```
Esempio scrittura di una anagrafica:
Letto un record con l'istruzione MYDBGET (specificato in _MYDBID)
` Modifico il valore del campo 10
_MYDB$(10) = "Modificata da sprix"
MYDBPUT ARCHIVIO$
```

```
Esempio scrittura di tabella:
` Modifico il valore del campo descrizione del record 4
```

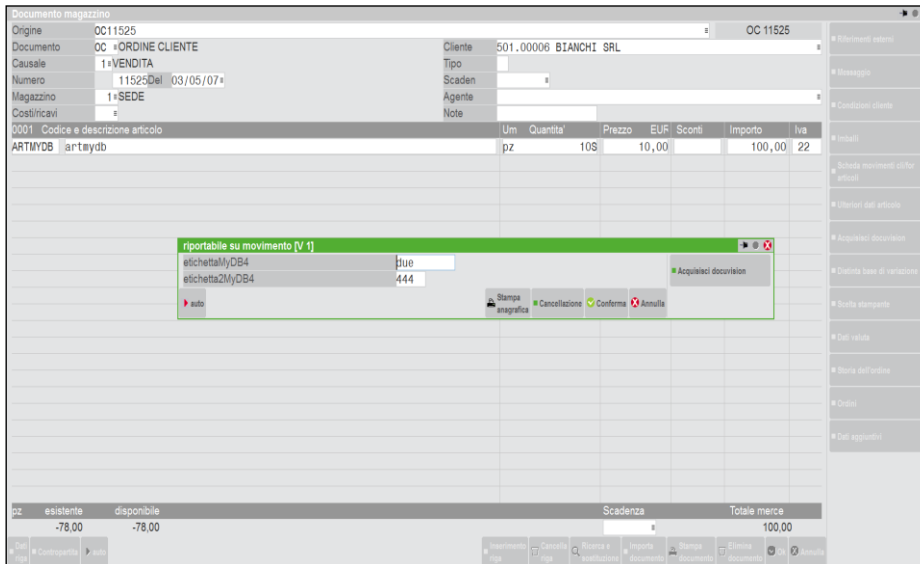
```
AZZVARSYS 35
_MYDBID = 4
_MYDB$(2) = "Descrizione inserita da sprix"
MYDBPUT TABELLA$
```

Esempio MYDBPUT

Data un'anagrafica MyDB come estensione diretta delle righe movimento, leggo un Ordine Cliente a cui sono stati attribuiti dei valori di riga come estensione diretta MyDB, e ne modifico il contenuto con Sprix.



La riga movimento ha due campi MyDB valorizzati che voglio modificare con MYDBPUT



Prima leggo il documento, poi leggo MYDB e infine modifico i campi.
PRESTARE ATTENZIONE ALLA SINTASSI MYDBPUT

```
GETMM 'OC11525' ' leggo il documento
IF_ERRMM <> 0 THEN VMSG 4,_ERRMM$
AZZVARSYS 35
MYDBGET '1928900000_MYDB4',0,1,'DOC=OC' ' leggo MyDB
IF_ERRMYDBS <> '' THEN VMSG 4,_ERRMYDBS
VMSG 1, STR$( _MYDBID,0,0)+' / '+_MYDB$(1)+' / '+_MYDB$(2)+' / '+_MYDB$(3) ' visualizzo contenuti

_MYDB$(2)="tre" ' attribuisco nuovo valore al campo 2
_MYDB$(3)="666" ' attribuisco nuovo valore al campo 3
MYDBPUT 'MYDB4',"DOC=OC" ' sintassi utile DENTRO LA STESSA APP

MYDBPUT '1928900000_MYDB4DMRT',"DOC=OC" sintassi equivalente alla precedente (ma con la specifica del codice produttore)
MYDBPUT '123456app@MYDB4',"DOC=OC" ' SINTASSI UTILE PER ESECUZIONI IN UN'ALTRA APP (e relativo codice App)
MYDBPUT '1928900000_MYDB4DMRT',"DOC=OC" (forma NON corretta perche' NON sono di tipo riportabile, bensì estensione diretta)

IF_ERRMYDBS <> '' THEN VMSG 4,_ERRMYDBS
END
```

Risultato sul documento

Documento magazzino

Origine OC11525 OC 11525

Documento OC + ORDINE CLIENTE Cliente 501.00006 BIANCHI SRL

Causale 1+VENDITA Tipo

Numero 11525Del 03/05/07 Scaden

Magazzino 1+SEDE Agente

Costitricavi Note

0001	Codice e descrizione articolo	Um	Quantità	Prezzo	EUR	Sconti	Importo	Iva
ARTMYDB	artmydb	pz	10S	10,00			100,00	22

Stampa anagrafica Cancellazione Conferma Annulla

riportabile su movimento [V 2]

etichettaMyDB4 tre

etichetta2MyDB4 666

Acquisisci documenti

Stampa anagrafica Cancellazione Conferma Annulla

pz	esistente	disponibile	Scadenza	Totale merce
	-78,00	-78,00		100,00

Insediamento Cancellazione Ricerca e sostituzione Importa Annulla Stampa Annulla

Esempi Scrittura Estensione:

(dato il produttore 1901536000 e l'anagrafica/tabella "EST")

Scrittura di una **Estensione diretta** delle **testate** sulla bolla BC1/116 del cliente 501.00001:
 `Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
 _MYDB\$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001" `(Riga 0 - testata 1)
 _MYDB\$(2) = "1133"
 _MYDB\$(3) = "Scritto da sprix"
 MYDBPUT "1901536000 **ESTDOCR**", "DOC=BC" `(Produttore_Anagrafica, Doc = Movimenti)
 IF _ERRMYDB\$ <> "" THEN VIMSG 4,_ERRMYDB\$

Scrittura di una **Estensione diretta** delle **testate** sulla bolla BC1/116 del cliente 501.00001 su azienda **INFRANNUALE**:
 `Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
 _MYDB\$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001"+"20160115" `(Riga 0 - testata 1)
 _MYDB\$(2) = "1133"
 _MYDB\$(3) = "Scritto da sprix"
 MYDBPUT "1901536000 **ESTDOCR**", "DOC=BC" `(Produttore_Anagrafica, Doc = Movimenti)
 IF _ERRMYDB\$ <> "" THEN VIMSG 4,_ERRMYDB\$

Scrittura di una **Estensione diretta** delle **righe** sulla bolla BC1/116 del cliente 501.00001:
 `Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
 _MYDB\$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001" `(Riga 1 - testata 1)
 _MYDB\$(2) = "1133"
 _MYDB\$(3) = "Scritto da sprix"
 MYDBPUT "1901536000 **ESTDOCR**", "DOC=BC" `(Produttore_Anagrafica, Doc = Movimenti)
 IF _ERRMYDB\$ <> "" THEN VIMSG 4,_ERRMYDB\$

Scrittura di una **Estensione dei clienti (standard, riportabile o dinamica)** sul CLIENTE 501.00001:
 `Imposto il campo estensione per agganciare il cliente
 _MYDB\$(1) = "501.00001"
 _MYDB\$(2) = "1133"
 _MYDB\$(3) = "Scritto da sprix"
 MYDBPUT "1901536000 **ESTCLIR**", `(CodiceProduttore_CodiceTabella/Anagrafica)
 IF _ERRMYDB\$ <> "" THEN VIMSG 4,_ERRMYDB\$

Scrittura di una **Estensione dei clienti (riportabile o dinamica)** sulla TESTATA CLIENTI della bolla BC1/116 del cliente 501.00001:
 `Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
 _MYDB\$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001" `(Riga 0 - testata 1)
 _MYDB\$(2) = "1133"
 _MYDB\$(3) = "Scritto da sprix"
 MYDBPUT "1901536000 **ESTCLIRDMTC**", "DOC=BC"
 ` Produttore_Anagrafica??#+Sezione documento (*1 - Vedere la tabella di identificativi MyDB di testata e di riga elencati in fondo alla tabella struttura archivi_MYDB), Doc = Movimenti
 IF _ERRMYDB\$ <> "" THEN VIMSG 4,_ERRMYDB\$

Scrittura di una **Estensione dei clienti (riportabile o dinamica)** sulla RIGA degli ARTICOLI VALORIZZATI della bolla BC1/116 del cliente 501.00001:
 `Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
 _MYDB\$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001" `(Riga 1 - testata 1)
 _MYDB\$(4) = "1133"
 _MYDB\$(5) = "Scritto da sprix"
 MYDBPUT "1901536000 **ESTCLIRDMRX**", "DOC=BC"
 ` Produttore_Anagrafica??#+Sezione documento (*1 - Vedere la tabella di identificativi MyDB di testata e di riga elencati in fondo alla tabella struttura archivi_MYDB), Doc = Movimenti
 IF _ERRMYDB\$ <> "" THEN VIMSG 4,_ERRMYDB\$

Scrittura di una **Estensione degli articoli (standard o riportabile)** sull'ARTICOLO 001:
 `Imposto il campo estensione per agganciare l'articolo
 _MYDB\$(1) = "001"
 _MYDB\$(2) = "Valore 1"
 _MYDB\$(3) = "Scritto da sprix"
 MYDBPUT "1901536000 **ESTARTR**", `(CodiceProduttore_CodiceTabella/Anagrafica)
 IF _ERRMYDB\$ <> "" THEN VIMSG 4,_ERRMYDB\$

Scrittura di una **Estensione degli articoli (riportabile)** sulla RIGA della bolla BC1/116 del cliente 501.00001:
 `Imposto il campo estensione per agganciare il documento (vedere le tabelle chiavi elencate nella voce MYDBGET)
 _MYDB\$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001" `(Riga 1 - testata 1)
 _MYDB\$(2) = "Valore 1"
 _MYDB\$(3) = "Scritto da sprix"
 MYDBPUT "1901536000 **ESTARTRDMRX**", "DOC=BC"
 ` Produttore_Anagrafica??#+Sezione documento (*1 - Vedere la tabella di identificativi MyDB di testata e di riga elencati in fondo alla tabella struttura archivi_MYDB), Doc = Movimenti
 IF _ERRMYDB\$ <> "" THEN VIMSG 4,_ERRMYDB\$

Esempi Scrittura Estensione diretta delle Righe su BOLLA nuova del Cliente 501.00007:

Configuro un'anagrafica di estensione riportabile di riga. Importante è considerare il Codice Produttore e Codice Anagrafica, come evidenziato nell'immagine. La sintassi degli archivi può richiedere l'aggiunta di un ulteriore codice (sono elencati in fondo alla tabella delle variabili di struttura_MYDB), in questo esempio è DMRT, pertanto l'intero codice dell'archivio è "1928900000_CODXYZDMRT". Le due etichette mi inseriranno di default sulle righe del documento i valori "AAAAAAAAA" e "33333".

MODIFICA ANAGRAFICA MYDB

Codice: CODXYZ
 Descrizione: Estensione informazioni articoli
 Stato: T A Terminato
 Primo ordinamento univoco: S N E Per estensione
 Ordine: I E 1 2 3 Estensione
 Estensione: ARTT Tutti gli articoli
 Estensione univoca: S N Si Tipo estensione: S R Riportabile

Etichetta	3	Tipo	Dim	Vis	Obb	Pos	I/O	Valore Iniziale	Ordin.	Codice
Tutti gli articoli	*		8		S	1	I			ZZZ
Seconda etichetta	A		10	10	S	2	I	AAAAAAAAA	1	001
Terza etichetta	N		5		S	3	I	33333	2	002

Attenzione

- > Elaborazione archivio 1928900000_codxyz
- Elaborazione archivio 1928900000_codxyzdmrtm
- Elaborazione archivio 1928900000_codxyzdmrtp
- Elaborazione archivio 1928900000_codxyzdmrtc
- Elaborazione archivio 1928900000_codxyzdmrtf
- Elaborazione archivio 1928900000_codxyzdmrta
- Elaborazione archivio 1928900000_codxyzdmrtn

Stampa Chiudi

Da Emissione Documenti ho scritto una Bolla con due righe, ciascuna con le etichette di estensione di default

DOCUMENTO MAGAZZINO

Origine: BC22 BC 22
 Documento: BC BOLLAACCOMPAG Cliente: 501.00007 ALESSI
 Causale:
 Numero: 22 Del 03/05/07 ABANO TERME BAGNI PD
 Magazzino: 1 SEDE Agente:
 Costi/ricavi: Note:

0002	Codice e descrizione articolo	Um	Quantita'	Prezzo	EUR	Sconti	Importo	Iva
ANTAREX	Antarex blister 10 pz	nr	10	22,95	3,00	222,62	22	
SAPONE	descrizione del sapone	pz	10	0,00		0,00	22	

Estensione informazioni articoli (Tutti gli Art.) [V 4]

Seconda etichetta: AAAAAAAAAA
 Terza etichetta: 33333

Cancellazione Conferma Annulla

pz	esistente	disponibile	Totale merce
	-100,00	-100,00	222,62

Valore iva Contropartita Inserimento riga Cancella riga Ricerca e sostituzione Importa documento Stampa documento Elimina documento Ok Annulla

Eseguo uno Sprix che legge questo documento, lo controlla e lo duplica aggiungendo sia una riga in più (l'articolo "ART") sia una modifica ai campi delle estensioni di tutte le righe (per farle diventare "BBBBBBBBB" e "55555"). Notare la sintassi del nome dell'archivio (v. tabella variabili struttura_MYDB): "1928900000_CODXYZDMRT".

Dopo della lettura potrei fare qualche ciclo per verificare eventuali dati MYDB di testata e poi di riga, successivamente eseguo le modifiche che desidero.

```

GETMM "BC22"
' controllo la presenza di eventuali dati MYDB nella testata e nelle righe
FOR T=1 TO 3 : VIMSG 1, STR$( _MMYDBT(T),0,0) : NEXT
FOR R=1 TO _NMM : VIMSG 1, STR$( _MMYDBR(R),0,0) : NEXT
' duplico il documento e aggiungo una 3^ riga (per brevit a infatti, considero gi a che ci sono solo 2 righe)
_MMNUM=0 : _MMTPR$(3)="R" : _MMART$(3)="ART" : _MMQTA(3)=10 : _MMALI$(3)="22" : PUTMM 0 : VIMSG 1, _ERRMMS
'gestisco in stringa sia la serie che il numero della chiave del documento BC
SER$="01" : NUM=nunum("BC",sxann,1) : A$="000000" : BS=STR$(NUM,0,0) : C$=A$+B$ : NUM$=RIGHT$(C$,6)
' rileggo il documento (perche' ho aggiunto una riga e quindi _NMM e' cambiato)
AZZVARSYS 4 : GETMM "BC"+STR$(NUM,0,0)
' modifico i dati anagrafici MYDB delle _NMM righe (sono 3) scrivendo l'estensione diretta delle righe
'imposto il campo estensione per agganciare il documento
' ciclo le righe sulla testata 1 (perche' il documento non deriva dalla una fusione di documenti)
FOR RR=1 TO _NMM
  AZZCATVARSYS "MYDB"
  K1$="00000000" : K2$=STR$(RR,0,0) : K3$=K1$+K2$ : NUM_RIGAS$=RIGHT$(K3$,8)
  _MYDB$(1)= "000"+_MMSIG$+SER$+"."+NUM$+_MMCLIS$+NUM_RIGAS$+"001"
  _MYDB$(2)= "BBBBBBBBBB" : _MYDB$(3)= "55555"
  MYDBPUT "1928900000_CODXYZDMRT", "DOC=BC" ' (CodiceProduttore_CodiceAnagrafica, Doc = Movimenti)
  IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
NEXT
  
```

6]1.....1.....2.....3.....4.....5.....6.....7.....8.....9.....10.....

Il nuovo documento copiato (BC30) presenta tutte le estensioni modificate.

DOCUMENTO MAGAZZINO

Origine **BC30** BC 30

Documento **BC** BOLLAACCOMPAG Cliente **501.00007 ALESSI**

Causale **30Del** **03/05/07** ABANO TERME BAGNI PD

Magazzino **1** SEDE Agente

Costi/ricavi Note

0002	Codice e descrizione articolo	Um	Quantita'	Prezzo	EUR	Sconti	Importo	Iva
ANTAREX	Antarex blister 10 pz	nr	10	22,95		3,00	222,62	22
SAPONE	descrizione del sapone	pz	10	0,00			0,00	22
ART	articolo	nr	10	0,00			0,00	22

Estensione informazioni articoli (Tutti gli Art.) [V 17]

Seconda etichetta **BBBBBBBBBB**

Terza etichetta **55555**

pz	esistente	disponibile	Totale merce
	-110,00	-110,00	222,62

Valore Iva Contropartita
 Inserimento riga Cancella riga
 Ricerca e sostituzione Importa documento
 Stampa documento Elimina documento

Con un ciclotabellare posso verificare le modifiche avvenute alla nuova copia della bolla

```

yy N MYDBTAB "1928900000_codxyzdmrt"
~yy1928900000_codxyzdmrtzzz() ~yy1928900000_codxyzdmrt001() ~yy1928900000_codxyzdmrt002()
  
```

NOTARE: si consideri che il documento   NUOVO seppure sia proveniente dalla copia di uno gi a esistente. In caso di modifica di uno gi a esistente, invece, occorrono altri accorgimenti e controlli sulle variabili, per esempio il vettore _MYDB\$()

Esempio MYDBVIEW:

(dato il produttore 1900074000 e l'archivio "MYDB3")

```

*** Visualizzazione elenco dei record
MYDBVIEW "1900074000 MYDB3", 0, 0
  
```

La finestra mostra tutti I record elencati per ID

```
_MYDBK$(1) = "501.00002"
MYDBVIEW "1900074000_MYDB3", 1, 0
La finestra mostra tutti I record elencati a partire dal codice 501.00002
```

```
_MYDBF$(1)="501.00002"
_MYDBF$(2)="Domenica"
MYDBVIEW "1900074000_MYDB3", 0, 0, "FILTRO=1"
```

La finestra mostra tutti i record filtrati per il primo elemento dell'archivio = a 501.00002 e secondo elemento dell'archivio = a "Domenica"

```
MYDBVIEW "1901536000_ESTCLIRDMTC", 99, 0, "FILTRO=1, DOC=FT"
```

La finestra mostra, in base ad un filtro, tutti I record dell'archivio estensione dei clienti riportabile sui documenti, nella sezione delle testate documenti clienti (DMTC), identificati come movimenti (DOC=FT). "DOC" vuole una sigla documento di magazzino. La sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio.

Esempio _MYDBF\$()

Per eseguire la MYDBVIEW filtrata da data a data, supponendo un archivio MyDB nel quale la data che si desidera filtrare sia il 4 elemento, se volesse ottenere tutti i record con tale data compresa tra 01/01/2019 e 31/12/2019, dovrebbe utilizzare il codice di seguito:

```
AZZVARSYS 35:
_MYDBF$(4)="20190101": _MYDBF_DTA$(4)="20191231":
MYDBVIEW "archivio",0,0,"FILTRO=1":
```

Se invece volesse ottenere solo i record della data 01/06/2019:

```
AZZVARSYS 35:
_MYDBF$(4)="20190601": _MYDBF_DTA$(4)="20190601":
MYDBVIEW "archivio",0,0,"FILTRO=1":
```

Esempio di utilizzo MYDBVIEW con parametro NOPUT.

Si è voluto permettere l'inserimento controllato di una anagrafica MYDB (TMYDBVIE) che è composta da due soli campi di tipo data: Data inizio e Data fine. Si vuole una gestione in grado di controllare al momento dell'inserimento di un nuovo record o alla modifica di uno esistente di effettuare i controlli opportuni sul corretto inserimento delle due date, dando un opportuno messaggio in caso di inserimento errato e riproponendo i dati appena inseriti con la possibilità di modificarli e confermarli nuovamente.

```
VISUALIZZA_MYDBVIEW:
_W_ARC$="TMYDBVIE":_MYDBID=0:O_VALID_ID=0:
LOOP_V_MYDBVIEW:
MYDBVIEW _W_ARC$,0,_MYDBID,"INSER,NOPUT"
IF _MYDBTF=0 THEN GOTO END_LOOP_V_MYDBVIEW:

'QUI ASSEGNO L'ID DELL'ULTIMO RECORD SELEZIONATO
IF _MYDBID<>-1 THEN O_VALID_ID=_MYDBID
'QUI ARRIVO DOPO L'INSERIMENTO F4 O INVIO PER MODIFICA
_W_CHECK=0: _W_OK=0: _W_RECID=_MYDBID

LOOP_V_MYDBINPUT:
IF _W_CHECK OR _MYDBID=-1 THEN GOSUB CONTROLLA_RECORD
IF _W_OK THEN GOSUB SALVA_RECORD: GOTO END_LOOP_V_MYDBINPUT

MYDBINPUT _W_ARC$, _MYDBID, "NOPUT, NOCAN"
IF _MYDBTF=505 THEN GOTO END_LOOP_V_MYDBINPUT

_W_CHECK=1: _MYDBID=_W_RECID: GOTO LOOP_V_MYDBINPUT
END_LOOP_V_MYDBINPUT:

IF _MYDBID=-1 THEN _MYDBID=O_VALID_ID
GOTO LOOP_V_MYDBVIEW
END_LOOP_V_MYDBVIEW:
END

CONTROLLA_RECORD:
_W_OK=1
IF _MYDB$(1)>_MYDB$(2) THEN VIMSG 4,"INTERVALLO DI DATE ERRATO!":_W_OK=0
RETURN

SALVA_RECORD:
IF _MYDBID=-1 THEN _MYDBID=0
MYDBPUT _W_ARC$
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
RETURN
```

Esempio MYDBINPUT

(dato il produttore 1900074000 e l'archivio "MYDB4")

Apri a video il record 13 ma non esegue la scrittura del record

```
ARCHIVIO$="1900074000_MYDB4"
MYDBINPUT ARCHIVIO$, 13, "NOPUT" \
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
```

Apri a video il record 13 ed esegue la registrazione dello stesso

```
MYDBINPUT "1900074000_MYDB4", 13
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
```

Esempio MYDBLOF:

```
NumeroChiave = MYDBLOF ("1900074000_MYDB4", " ")
Ritorna il numero di record presenti nell'archivio indicato.
E' utile per trovare l'ultimo record
```

Esempio MYDBDEL:

(dato il produttore 1901536000, l'archivio "MYDB5" e l'anagrafica/tabella "EST")

Esempio di cancellazione del record con _MYDBID = 10:

```
ARCHIVIO$ = "1901536000_MYDB5"
MYDBGET ARCHIVIO$, 0, 10
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
MYDBDEL ARCHIVIO$
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
```

Esempio di cancellazione di un record di una estensione testata documento:

```
ARCHIVIO$ = "1901536000_ESTDOCT"
_MYDBK$(1) = "000"+"BC"+"01"+"."+"000116"+"501.00001"+"00000000"+"001"+"20170115"
MYDBGET ARCHIVIO$, 99, 0, "DOC=BC"
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
MYDBDEL ARCHIVIO$, "DOC=BC"
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
```

Esempio di cancellazione dell'intera anagrafica:

```
ARCHIVIO$ = "1901536000_MYDB5"
MYDBDEL ARCHIVIO$, "TOTALE"
IF _ERRMYDB$ <> "" THEN VIMSG 4,_ERRMYDB$
```

Esempio MYDBGETATT:

```
ON_OUT_YMYD002:
`Leggo le info dalla videata corrente e valorizzo l'array _MYDB$()
MYDBGETATT
`Modifico le informazioni lette dentro l'array _MYDB$()
_MYDB$(2) = "501.00001" : _SINP$ = "501.00001" ` Cliente
_MYDB$(3) = "2" ` Tecnico
_MYDB$(4) = "1" ` Tipo intervento
_MYDB$(5) = "1" ` Stato intervento
_MYDB$(6) = sxdat ` Data
_MYDB$(7) = "5" ` Durata
_MYDB$(8) = "250,50" ` Prezzo
_MYDB$(9) = "N" ` Pagato
_MYDB$(10) = "Creato da Collage" ` Note
` Imposto nella videata i valori impostati nell'array
` Mi posiziono sul campo Cliente
MYDBPUTATT "GOTO=002"
IF _ERRMYDB$ <> "" THEN VIMSG 5,_ERRMYDB$
END
```

Esempio: Collage su emmissione documenti che richiama il singolo record di anagrafica MyDB con il valore univoco di ordinamento corrispondente al Gruppo Merceologico. Data la seguente configurazione: un insieme di gruppi merceologici (prima immagine), dato l'articolo collegato al gruppo "SECONDO" (seconda immagine), data un'anagrafica mydb con ordinamento univoco ed elemento 5 relativa al gruppo merceologico (terza e quarta immagine), realizzo il Collage che richiama l'anagrafica MyDB (quinta e sesta immagine)

Merceologici			
Cod.	Descrizione	Liv Padre	A
PRIMO	primo gruppo	0	
SECONDO	secondo gruppo	0	
TERZO	terzo gruppo	0	

Anagrafica articoli		ALTRI DATI ANAGRAFICI	
Codice	SAPONE	Decimale	2
Tipologia	Merci	Confezione	
Descrizione	descrizione d	Codice Alternativo	
Unità di Misura	pz	Costo standard	
Altra U.M.		Categoria statistica	00
Aliquota Iva	22	Gruppo merceologico	SECONDO=secondo gruppo
Costo ultimo		Natura	
Prezzo		Magazzino	TOT MAGAZZINI
Note		Data progressivi	01/01/2
		Data Creazione	01/01/2007
		Data Ultima Modifica	01/01/2007
		PRODUZIONE	
		Consumo per taglia	S N No
		Forma	S N No

GESTIONE MYDB

Descrizione	Produttore	Codice	Tp	St	Est
dati per calcolo	1928900000	MAGGIOR	A	T	

MODIFICA ANAGRAFICA MYDB

Codice: MAGGIOR
 Descrizione: dati per calcolo
 Esecuzione da menu': Magazzino
 Stato: Terminato
 Primo ordinamento univoco: S N E Si
 Ordinarmento: I E 1 2 3 Ordinarmento 1

Etichetta	Tipo	Dim	Vis	Obb	Pos	I/O	Valore Iniziale	Ordin.	Codice
% standard	2	6		N	2	I			001
% extra	2	6		N	4	I			003
fisso	2	6		N	5	I			004
ricarico	2	6		N	6	I			005
gruppo merceologico	M	7	4	S	1	I		1	006
senza sconto	M	1		N	7	I	N		007

PARAMETRI FINESTRA ELENCO

Etichetta	Pos	Titolo	Dim
% standard	2		
% extra	3		
fisso	4		
ricarico	5		
gruppo merceologico	1		
senza sconto	6		

gruppo	% stan	% extr	fisso	ricari s
PRIMO	1,00	3,00	4,00	5,00 N
SECONDO	1,20	2,30	4,40	7,00 S
TERZO	0,20	1,00	3,00	10,00 N

dati per calcolo [V 5]

gruppo merceologico	SEC
% standard	1,20
% extra	2,30
fisso	4,40
ricarico	7,00
senza sconto	S

```
ON_IN_MDQTA:
GETATTMM : GRUPMERC$=argme(gmmer(_MMART$(1)))
_MYDBF$(5)=GRUPMERC$
'_MYDBF$(5)="SECONDO"
MYDBVIEW "1928900000_maggior",1,0, "FILTRO=1"
END
```

DOCUMENTO MAGAZZINO

Documento: OC ORDINE CLIENTE
 Causale: 1 VENDITA
 Numero: AUTO Del 01/01/07
 Magazzino: 1 SEDE
 Clienti: 501.00057 Francesca
 Tipo:
 Scadenza:
 Agente:
 Note:
 Costi/ricavi:
 Note:

0001	Codice e descrizione articolo	Um	Quantita'	Prezzo	EUR	Sconti	Importo	Iva
SAPONE	descrizione del sapone	pz		0,00			0,00	22

dati per calcolo (gruppo merceologico) [Filtro attivo]

gruppo	% stan	% extr	fisso	ricari s
SECONDO	1,20	2,30	4,40	7,00 S

Stampa Ordinarmento utente Selezione Uscita

pz	esistente	disponibile	Scadenza	Totale merce
	0,00	0,00		0,00

Dati riga Contropartita Inserimento riga Cancella riga Ricerca e sostituzioni Importa documento Stampa documento Elimina documento Ok Annulla

Messaggio: Errore non correggibile MyDB Chiave duplicata a seguito di una scrittura da sprix

Eseguendo una MYDBPUT su un'estensione diretta del documento, possono non essere valorizzate correttamente le variabili _MMYDBT(T) (per la testata) e _MMYDBR(R) per le righe, e quindi, entrando nel documento si verifica l'errore: Errore non correggibile MyDB Chiave duplicata: Riferimenti: mx23.exe erdoc() pxind.c EsegInse()

Codice Sprix:

.....

```
GETMM "OC1/1"
IF ERRMM$ <> "" THEN VIMSG 2, ERRMM$
```

```

FOR X=1 TO _NMM
  FOR H=1 TO _MMQTA(X)
    _MYDB$(1)="000"+_MMSG$+RIGHT$("00"+STR$( _MMSER,0,0),2)+"."+RIGHT$("000000"+STR$( _MMNUM,0,0),6)+RIGHT$("00000000"+STR$( _MNCRERIGA(X),0,0),8)
  )
    _MYDB$(2)="TEST"+STR$(H,0,0)
    _MYDB$(3)=sxdat
    MYDBPUT "1901536000_ESTDOCR"+SEZIONE$, "DOC=OC"

    '1901536000 = Produttore
    'ESTDOCR = Anagrafica
    'SEZIONE$ = Sezione documento (vedere tabella di identificativi MyDB testata/riga in calce alla tabella struttura archivi _MYDB)
    'DOC = Documento

    IF _ERRMYDB$ <> "" THEN VIMSG 2, _ERRMYDB$
  AZZVARSYS 35
NEXT
NEXT
PUTMM 0
  IF _ERRMM$ <> "" THEN VIMSG 2, _ERRMM$

.....

```

Spiegazione:

Quando vengono scritti da sprix dei dati MyDB legati ai documenti occorre valorizzare a '1' sul documento stesso, le variabili _MMYDBT(T) (per la testata) e _MMYDBR(R) per le righe. Questo serve in emissione documenti perché il programma capisca se sono presenti o meno dei MyDB.

Tali variabili vengono valorizzate anche automaticamente dalla scrittura del MyDB legati ai documenti. Nello sprix, a seguito della scrittura del MyDB (che valorizza tali variabili), viene effettuata una PUTMM con la struttura che è in memoria in quel momento e quindi tali variabili vengono nuovamente azzerate.

Per risolvere il problema, occorre togliere la PUTMM finale, oppure impostare preventivamente le variabili di cui sopra.

Esempio: crea una struttura MyDB agganciata all'Anagrafica Contatti, se il contatto diventa un Cliente/Fornitore i dati MyDB vanno trasmessi ad una equivalente struttura MyDB agganciata all'Anagrafica Clienti/Fornitori

Premessa la creazione di due strutture MyDB "gemelle" rispettivamente collegate alle due anagrafiche (dei contatti e dei clienti / fornitori), l'operazione può essere eseguita da un Collage (oppure anche uno Sprix se si volesse piuttosto un'operazione di Servizio).

Per quanto riguarda Collage, la scelta degli eventi dipende da come si opera (in particolare, l'inserimento dei dati MYDB e la "trasformazione" del contatto in conto). Nell'esempio che segue si è scelto di comandare l'evento manualmente con F12 (sia inizialmente in anagrafica contatti che successivamente in anagrafica Clienti/Fornitori).

Il Collage esegue un'iniziale MYDBGGET dell'archivio collegato al Contatto e poi esegue una successiva MYDBPUT per replicare il dato nell'ambito anagrafico del cliente/fornitore al momento della sua generazione. Le due strutture MyDB hanno due campi ciascuna: il primo campo è quello di aggancio all'anagrafica, il secondo campo è quello di cui voglio fare la trasmissione del dato.

```

'Relativamente all'Anagrafica Contatti
ON 612_ALL:
  GETATTCON
  _MYDBK$(1)=_DCCTO$
  MYDBGGET "1928900000_MYDB1",99,_MYDBID
  VIMSG 1, _ERRMYDB$
  VIMSG 1, STR$( _MYDBID,0,0)+" "+_MYDB$(1)+" "+_MYDB$(2)
  D$=_MYDB$(2)
END

'Relativamente all'Anagrafica Clienti/Fornitori (quando vi si entra a seguito della generazione del Cli/For da anagrafica Contatti)
ON 612_CANAZ:
  AZZVARSYS 35
  GETATTFC
  _MYDBK$(1)=_PCCOD$
  MYDBGGET "1928900000_MYDB2",99,_MYDBID
  VIMSG 1, _ERRMYDB$
  VIMSG 1, STR$( _MYDBID,0,0)+" "+_MYDB$(1)+" "+_MYDB$(2)
  _MYDB$(2)=D$
  MYDBPUT "1928900000_MYDB2"
  VIMSG 1, _ERRMYDB$
END

```

Esempio: scrittura dei DATI AGGIUNTIVI di una FATTURA ELETTRONICA

'DAL DIZIONARIO SI COMPRENDE LA SINTASSI DEGLI ARCHIVI DI ANAGRAFICA MYDB relativi ai DATI AGGIUNTIVI FatturaPA (R) (Tutti gli Art.)
 'yy010999fatturapa@anapadoadmrt001()
 'yy010999fatturapa@anapadoadmrt002()

'leggo il documento FATTURA ELETTRONICA (una FE con una riga articolo e due campi aggiuntivi FatturaPA (R) (Tutti gli Art.)
 GETMM "FT98" : IF _ERRMM <> 0 THEN VIMSG 4, _ERRMM\$

```

AZZVARSYS 35

'mi costruisco la CHIAVE (vedere la composizione nelle tabelle di MYDBGGET)
_MYDBK$(1)="000FT01.000098501.0008300000001001"

'mi costruisco la SINTASSI DEL NOME DELL'ANAGRAFICA
MYDBGGET "010999FATTURAPA@ANAPADOADMRT",99,0,"DOC=FT"
IF _ERRMYDB$ <> "" THEN VIMSG 4, _ERRMYDB$

'visualizzo i contenuti che sto leggendo
VIMSG 1, STR$( _MYDBID,0,0)+" / "+_MYDB$(1)+" / "+_MYDB$(2)+" / "+_MYDB$(3)

'inserisco nuovi valori nei campi
_MYDB$(2)="CCCCCCCCCCCC" ' attribuisco nuovo valore al campo 2
_MYDB$(3)="BBBBBBBBBBBB" ' attribuisco nuovo valore al campo 3

'registro i campi e verifico
MYDBPUT "010999FATTURAPA@ANAPADOADMRT","DOC=FT"
IF _ERRMYDB$ <> "" THEN VIMSG 4, _ERRMYDB$

```

END

LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI

_AB - Articoli Abbinati

_AB

Variabili Struttura Codici Articoli Abbinati (GETABBIN – PUTABBIN – DELABBIN)	
_ABPRO()	(Numero progressivo; Specifico = 0,1, 2, .. ; AUTO = -1)
_ABCOD\$()	(Codice abbinato)
_ABCAT\$()	(Categoria abbinamento); la PUTABBIN non scrive le categorie, devono essere già in tabella
_ABCATD\$()	(Descrizione categoria); la PUTABBIN non scrive le descrizioni, devono essere già in tabella
_ABART\$()	(Codice articolo)
_ABNO1\$()	(Nota 1)
_ABNO2\$()	(Nota 2)
_ABDTC\$()	(Data creazione – sola lettura)
_ABDTU\$()	(Data ultima modifica – sola lettura)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ACCFILEUTE

_ACCFILEUTE

Parametri	
= 1	Controllo attivato
= 0	Controllo non attivato

Determina la gestione degli accessi a file ed ai percorsi tramite comandi sprix da parte degli utenti definiti nel Servizio di Configurazione Amministrazione Utenti (v. manuale generale), in cui stabilire le directory consentite e/o vietate. Se = 1 avviene il controllo, se = 0 gli accessi sono liberi per tutti gli utenti come se nel Servizio non fosse definita alcuna directory (salvo permessi stabiliti a livello file system del Server). Il controllo avviene anche per le righe di stampa su file (di tipo 'F') dell'editor.

La gestione di file utente _ACCFILEUTE = 1 valorizza _ERRFILE\$ con il relativo errore di accesso non consentito; l'errore sulla directory è valorizzato dall'istruzione LEGGIDIR. Per regolamentare l'accesso ai record del file avvalersi della semaforica. **Notare:** in tutte le operazioni che richiedono il nome del file, se questo non è completo di percorso (directory) viene aggiunta automaticamente la eventuale directory specificata come parametro alla partenza del gestionale (esempio nella dati); la lunghezza del nome del file compresa l'eventuale directory è max 200 caratteri (compreso eventuale redirectione '<'); alcuni sistemi operativi distinguono fra caratteri maiuscoli e minuscoli.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ACQFILE

_ACQFILE\$

Variabile di struttura con riferimento all'oggetto acquisito (vedere ACQUISISCIFILE); la variabile usabile solo nell'etichetta dopo l'operazione di acquisizione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_AD - Associazione Gestionale - Docuvision

_AD

Variabili Struttura Associazione Gestionale – Docuvision (GETASSOCDV – PUTASSOCDV – DELASSOCDV)	
_ADDVIDID()	Numero documento; sola lettura; la PUTASSOCDV può registrare il medesimo ID per riassociare l'entità del gestionale
_ADTIPO()	Classi delle associazioni; sola lettura
_ADNUMASSOC	Numero totale associazioni gestionale-docuvision; solo lettura
PIANO DEI CONTI	
_ADPCCOD\$()	Codice conto
_ADPCTIPC\$()	Tipo conto C/F
_ADPCCAL\$()	Codice alternativo
_ADPCRIC\$()	Nome ricerca
_ADPCPRO\$()	Provincia
_ADPCONA()	Zona
_ADPCCST()	Categoria statistica
PRIMANOTA	
_ADPNSAZ()	Codice SottoAzienda
_ADPNDRES()	Data di registrazione
_ADPNPRN()	Progressivo di registrazione
_ADPNCAU\$()	Causale
_ADPNTDES()	Descrizione
_ADPNRPR\$()	Tipo Registro A/V/C
_ADPNSPR()	Serie protocollo
_ADPNNPR()	Numero protocollo
_ADPNND()	Numero documento

_ADPNDDO\$()	Data documento
ARTICOLI	
_ADARCOD\$()	Codice articolo
_ADARDES\$()	Descrizione articolo
_ADARALT\$()	Codice alternativo
_ADARSTA\$()	Categoria statistica
_ADARSCHTGL()	Scheda taglie (valorizzata in PUTDVDOC tramite parametro facoltativo allegati: "PRGSCH=(1..99), TPALL=S")
DOCUMENTI DI MAGAZZINO	
_ADMMSAZ\$()	Codice SottoAzienda
_ADMSTPD\$()	Documento stampato
_ADMMSIG\$()	Sigla documento
_ADMMSER\$()	Serie documento
_ADMNUM\$()	Numero documento
_ADMMDAT\$()	Data documento
_ADMMAFIS\$()	Anno fiscale; solo documento di movimento
_ADMCLI\$()	Codice conto
_ADMZON\$()	Zona conto; solo per documenti di movimento; nei documenti che non movimentano il magazzino (ordini, preventivi, matrici) questo dato non è memorizzato, pertanto non è gestito.
_ADMSTA\$()	Categoria statistica; solo per documenti di movimento; Nei documenti che non movimentano il magazzino (ordini, preventivi, matrici) questo dato non è memorizzato, pertanto non è gestito.
_ADMNOT\$()	Note
BOLLE DI LAVORAZIONE	
_ADOBTPD\$()	Documento stampato
_ADOBNUM\$()	Numero bolla
_ADOBNSO\$()	Numero sottobolla
_ADOBDAT\$()	Data bolla
_ADOBCTO\$()	Codice conto
_ADOBDST\$()	Nota
NOTULE / PARCELLE	
_ADNPSAZ\$()	Codice SottoAzienda
_ADNPSTPD\$()	Documento stampato
_ADNPSIG\$()	Sigla documento
_ADNPSER\$()	Serie documento
_ADNPNUM\$()	Numero documento
_ADNPDAT\$()	Data Documento
_ADNPCLI\$()	Codice conto
_ADNPNOT\$()	Note
CDG	
_ADCDGTANAG\$()	Tipo anagrafica "A", "C", "R".
_ADCDGCOD\$()	Codice dell'anagrafica specificata
_ADCDGPADRE\$()	Padre della Commessa/Area
_ADCDGTIPOS\$()	Tipo Commessa/Area o Tipo conto Analitico
_ADCDGRESPO\$()	Responsabile Commessa/Area
_ADCDGZONA\$()	Zona Commessa/Area
_ADCDGNAT\$()	Natura Analitico
_ADCDGQUAL\$()	Qualità Analitico
_ADCDGSTATO\$()	Stato Commessa
_ADCDGCLI\$()	Cliente Commessa
_ADCDGDMERC\$()	Destinazione Merce Commessa
_ADCDGAGEN\$()	Agente Commessa
TIPO ALLEGATO	
_ADARTPA\$()	Tipo Allegato, modificabile solo con parametro [alleg] di PUTASSOCDV
_ADARPRGA\$()	Progressivo Immagine, modificabile solo con parametro [alleg] di PUTASSOCDV
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_AE - Anagrafica Aree

_AE

Variabili Struttura Anagrafica Aree (GETAREA – PUTAREA)	
_AECOM\$	Codice Area
_AEDES\$	Descrizione
_AEPAD\$	Codice Area padre
_AESOS\$	Codice Area Sostitutiva
_AETPA\$	Tipo Area
_AETPZ\$	Zona
_AERES\$	Responsabile
_AENOT\$	Nota
_AEDTV\$	Data Validità
_AEDTC\$	Data creazione
_AEDTM\$	Data ultimo aggiornamento
_AEANN\$	Annullato
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_AF – Esistenza campi su archivi mobile

_AF

Variabili Struttura Archivi mobile (GETARCHFLD)	
_AFESISTE\$	Variabile flag che indica se il campo è presente nell'archivio mobile
_AFTIPVAL\$	Tipo valore (DATA, NUMERO, STRINGA)
_AFNCAR	Dimensione in caratteri del campo
_AFNDEC	Numero decimali campo

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_AGP - Agenda di Lavoro

_AGP

Variabili di struttura archivi AGENDA DI LAVORO (GETAGP – PUTAGP – DELAGP)	
_AGPNELE	Variabile che riporta il totale di elementi trovati con l'istruzione di lettura GETAGP
_AGPDTLAV\$()	Data lavori (campo chiave)
_AGPID()	Numero record (campo chiave, non visibile a video)
_AGPSTATO\$()	Stato del lavoro A/F/P
_AGPPRCOD\$()	Codice della prestazione
_AGPDATAPA\$()	Riporta data Prestazione su notula/Parcella S/N
_AGPNOTAPA\$()	Riporta Nota su No/PA N/V
_AGPSPESA()	Spesa anticipata
_AGPCODCLI()	Codice Cliente
_AGPNOTE\$()	Nota
_AGPAZIE\$()	Sigla azienda
_AGPPRZ()	Prezzo
_AGPQTA()	Quantità
_AGPTIPOR\$()	Origine (Stato=A/F/P/N/S)
_AGPAZZERA\$()	S/N
_AGPUTE()	Id Utente

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_AL - Codici Alias Articoli

_AL

Variabili di struttura archivi Codici Alias Articoli (GETALIAS – PUTALIAS – DELALIAS)	
_ALID()	Numero ID degli Alias;
_ALPRO()	Numero progressivo; Specifico = 0,1,2,...; AUTO=-1
_ALCOD\$()	Codice Alias
_ALART\$()	Codice articolo
_ALDESS\$()	Caratteristiche
_ALNO1\$()	Nota 1
_ALNO2\$()	Nota 2
_ALCOLLI()	Numero colli / numero taglia relativa a _ALQTA() per articoli a taglie
_ALQTA()	Quantità
_ALTPUM\$()	Unità di misura: ritorna "P" primaria o "S" secondaria; se l'articolo è a peso netto oppure è a taglie, torna sempre "P"
_ALDTC\$()	Data creazione – solo lettura
_ALDTU\$()	Data aggiornamento – solo lettura

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_APF – Avanzamento di Produzione

_APF

Variabili di struttura archivi Avanzamento di Produzione (AVANZAPF)	
_APFCONTO\$	Codice di conto al quale intestare il documento - solo input
_APFSER	Numero di serie che si vuole usare per il documento – input/output
_APFNUM()	Numeri documento creati – solo output
_APFMAG()	Numero del magazzino dei documenti (dalla versione 2024B è possibile utilizzare la variabile per specificare il magazzino di carico dei prodotti).
_APFIDLOT(E)	ID lotto da utilizzare per il documento CL – solo input, valori ammessi da 1 a _NLOTTIAPF
_APFLQTA(E)	Quantità del lotto – solo input, valori ammessi da 1 a _NLOTTIAPF
_APFLCOLLI(E)	Colli del lotto – solo input, valori ammessi da 1 a _NLOTTIAPF
_APFLQTATGL(E,32)	Quantità a taglie del lotto – solo input, valori ammessi da 1 a _NLOTTIAPF
_APFLNDQTA(E)	Numero di decimali quantità lotto – solo input, valori ammessi da 1 a _NLOTTIAPF
_APFQTATG(32)	Variabile di solo input che consente di specificare in fase di carico la sovrapproduzione per gli articoli a taglie. Nota bene: il totale della sovrapproduzione inserita nella variabile deve coincidere con il totale indicato in AVANZAPF; se non viene specificato nulla ma è presente una quantità specificata in AVANZAPF tutto il sovrapprodotta viene inserito in taglia 1.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_APPSTS – Elenco Stato APP

_APPSTS

Variabili di struttura Elenco APP (ELENCASTATOAPP)	
_APPSTSCOD\$()	Codice APP
_APPSTSNOME\$()	Nome APP
_APPSTSAZIE\$()	Satato dell'APP per l'azienda aperta
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_AR - Anagrafica Articoli

_AR

Variabili struttura archivi Anagrafica Articoli (GETAR – GETATTAR – PUTAR – DELAR) (le variabili evidenziate in grassetto sono quelle minime obbligatorie)	
ANAGRAFICA GENERALE	
_ARCOD\$	<p>Codice articolo; se EAN il "+" alla fine è check; può essere utile consultare la lunghezza corretta con il dizionario "amdco" per evitare eventuali troncamenti dell'eccedenza in fase di scrittura.</p> <p>Si sconsiglia di utilizzare il carattere speciale doppio apice (") nel codice articolo, in quanto avvia la ricerca della parola esatta nella ricerca estesa; questo può comportare che simili codifiche (ART"/1, ART"/2, ART"/3, ecc) la ricerca per ART"/ non ne trova alcuno (poichè il carattere speciale non viene considerato nei caratteri da ricercare) ed occorrerebbe, pertanto, ricercare per "ART"/.</p>
_ARNRA	Numero record articolo; se = 0 è articolo nuovo; se <> 0 è articolo in variazione; è impostato solo dall'Anagrafica ed è in sola lettura e solo con la GETATTAR (oppure dizionario arrec(_ARCOD\$)); se l'articolo viene cancellato e reinserito può avere un record diverso.
_ARTIP\$	<p>Tipo articolo [dizionario artip()]. Un carattere alfanumerico [visibile codificando da gestionale un Articolo in Anagrafica Articoli oppure una Prestazione in Parcellazione Studio, aprendo il menu [F2] della tipologia che un elenco con lettere sottolineate che identificano il Tipo).</p> <p>L'archivio è comune a tutti i record di qualsiasi Tipo. Esempio:</p> <p>I tipi di articolo del menu AR (Anagrafica Articoli) sono: A, L, P, S, M, C, D, T, V, Z I tipi di articolo del menu CLP (Prestazioni Clienti) sono: R, D, T, V Di conseguenza i tipi D, T, V sono gli stessi articoli in comune ai due menu "AR" e "CLP".</p> <p>Nota: il Tipo è univocamente legato ad _ARCOD\$, non possono coesistere due tipi diversi per il medesimo _ARCOD\$, pertanto _ARTIP\$ non è modificabile ed occorre un nuovo _ARCOD\$ (dopo la prima scrittura su nuovo _ARCOD\$ la _ARTIP\$ è in sola lettura). La PUTAR con variazione della sola _ARTIP\$ in seguito ad una GETAR, riscrive lo stesso _ARCOD\$ aggiornando solo la data di ultima modifica e senza alcuna eccezione da parte della _ERRAR\$..</p>
_ARSME	Tipo merce Fatture Emesse (campo per RSM, numerico 1 carattere); vedere _MMTIPM(R)
_ARSMR	Tipo merce Fatture Ricevute (campo per RSM, numerico 1 carattere); vedere _MMTIPM(R)
_ARSTR	Numero struttura
_ARDES\$	Descrizione articolo
_ARDES\$(1-5)	Descrizione dettagliata. Array di 5 elementi da 200 caratt. ciascuno per descrizioni estese articolo (1000 caratteri); nel dizionario ardse() gli elementi sono 5, cioè ards(5)(). Notare: il 5° elemento sarà da 123 a 147 caratteri a seconda della dimensione del codice articolo (amdco); a seconda di questa dimensione, la dimensione totale è: (1000 - (80 - amdco) - 5) e quindi, nel caso di un codice da 8 caratteri la dimensione sarà di (1000 - (80 - 8) - 5) = 923 caratteri, mentre nel caso di un codice a 32 caratteri la dimensione sarà di 947. <u>Si rimanda alla tabellina presente nei dati anagrafici di Anagrafica Articolo del Manuale Utente.</u>
_ARAGG\$	Descrizione aggiuntiva articolo;
_ARALT\$	Codice alternativo / ulteriore descrizione; attenzione: se nel codice alternativo dell'articolo è presente un codice alfanumerico con un carattere "E" oppure "D" in conversione a valore numerico è considerato come esponente 10^, con generazione di numero max pari a 18.446.774.073.709.552.000. Esempio, se il codice alternativo è "9E2" o "9D2", il VALORE=VAL(aralt("ART")) equivale a 900, ovvero 9*10^2.
_ARIVA\$	Aliquota iva
_ARUM1\$	Unità di misura primaria
_ARUM2\$	Unità di misura alternativa; 10 – 32 se T > 9; non modificabile, ma scrivibile da putmm solo se l'articolo è nuovo
_ARDEC	Numero decimali articolo
_ARTCO\$	Tipo confezione
_ARCON	Confezione articolo
_ARKOE	Coefficiente conversione / peso tara
_ARKES	Coefficiente conversione esteso – solo lettura
_ARPTA	Peso tara
_ARVUC	Valuta costo ultimo anno corrente
_ARCUC	Costo ultimo anno corrente – solo lettura
_ARDTCUL\$	Data costo ultimo anno corrente – solo lettura
_ARVST	Valuta costo standard
_ARSST	Costo standard
_ARMFT	Quantità minima fatturabile
_ARMIN	Importo minimo fatturabile

_ARMAX	Importo massimo fatturabile
_ARCPS	Controllo peso
_ARQPR	Pr / Pa (prendi / paghi) quantità presa
_ARQPA	Pr / Pa (prendi / paghi)quantità pagata
_ARIMB\$	Imballo
_ARSCO	Tabella Sconti
_ARSCQ	Tabella Sconti quantità
_ARTPR	Tabella Provvigioni
_ARLIS	Categoria Formazione prezzi
_ARSTAS\$	Categoria statistica; alfanumerico
_ARSTN	Categoria statistica; numerico
_ARDTC\$	Data creazione
_ARDTAS\$	Data ultimo aggiornamento anagrafica (la conferma)
_ARDTAS\$	Data ultima modifica; solo tecnologia BP (recenti versioni) data di sistema e non data terminale
_ARRIC\$	Contropartita Ricavo
_ARCOSS\$	Contropartita Costo
_ARRSO\$	Contropartita Ricavo Sospeso, v. parcellazione
_ARTSC\$	Tessera Sanitaria: Tipo Spesa sanitaria
_ARTST\$	Tessera Sanitaria: Particolarità
_ARANC\$	Analitico di Costo
_ARANR\$	Analitico di Ricavo
_ARPRE\$	Articolo precancellato S/N
_ARANN\$	Articolo annullato S/N
_ARPRU\$	Articolo usato in PR – S/N – per GET/PUT/DEL
_ARSOS\$	Codice Articolo sostitutivo
_ARDBP\$	Articolo con DBP – S/N
_ARDBV\$	Articolo con DBV – S/N
_ARDBA\$	Articolo con DBA – S/N – sola lettura. Impostata a "CANCELLA" prima dell'esecuzione dell'aggiornamento anagrafica con PUTAR, causa la cancellazione dell'eventuale DBA associata all'articolo.
_ARCAR\$	Articolo con CARTELLA – S/N
_ARTMO\$	Tipo modificatore articolo
_ARVMO	Valore modificatore
_ARSMO\$	Somma prezzo base modificatore
_ARAMO\$	Tipo arrotondamento
_ARRRO	Valore arrotondamento
_ARRST	Riferimento struttura
_ARTDE\$	Tipo descrizione
_ARTPM\$	Tipo Valore: N = al Netto, F = Fisso
COSTI ANNI PRECEDENTI	
_ARVUP	Valuta costo ultimo anni precedenti
_ARCUP	Costo ultimo anni precedenti
_ARDTCUP\$	Data costo ultimo anno precedente – solo lettura
_ARVUL	Valuta costo ultimo anno
_ARCUL	Costo ultimo ultimo anno
FORNITORI	
_ARFOR\$(3)	Codice fornitore
_ARFORE\$(4-9)	Codice fornitore (estensione). Ritorna i restanti sei fornitori; i primi tre fornitori sono di output e non modificabili
_ARGGR(3)	Giorni di riordino
_ARGGRE(4-9)	Giorni di riordino (estensione)
_ARLOT(3)	Lotto di riordino
_ARLOTE(4-9)	Lotto di riordino (estensione)
_ARVAL(3)	Valuta prezzo base
_ARVALE(4-9)	Valuta prezzo base (estensione)
_ARFPR(3)	Prezzo base
_ARFPRE(4-9)	Prezzo base (estensione)
_ARCOF\$(3)	Codice articolo del fornitore
_ARCOFE\$(4-9)	Codice articolo del fornitore (estensione)
_ARQTA(3,3)	Fino a Quantità
_ARQTAE(4-9,3)	Fino a Quantità (estensione)
_ARSCQ\$(3,4)	Sconto quantità
_ARSCQE\$(4-9,4)	Sconto quantità (estensione)
_ARSCQN\$(3,4)	Tabella Sconti quantità -cascata-
_ARSCQNE\$(4-9,4)	Tabella Sconti quantità -cascata- (estensione)
CAMPIONARIO	
_ARCMA\$(200)	Codice componente
_ARCMU\$(200)	Unità di misura
_ARCMQ\$(200)	Quantità
_ARCMT(200,16)	Quantità Taglia 1 – 16
_ARCMTXL(200,16)	Quantità Taglia 17 – 32
_ARCMP\$(200)	Tipo condizione
_ARCMR\$(200)	Condizione art/sml (articolo/semilavorato)
_ARCMC\$(200)	Condizione Cliente
TESTO	
_ARTXV\$(200)	Testo Variabile
_ARTXT\$(200)	Testo Fisso
NOTE	

_ARNDS\$(2)	Note
_ARNDE\$(2)	Destinazione Nota
_ARNDT\$(2)	Data scadenza Nota
LINGUE	
_ARDSL\$(9)	Descrizione in Lingua
LISTINI - SCONTI - PROVVISORI	
_ARPRZ(1-9)	Prezzo su listini dal 1 al 9
_ARPRZE(10-999)	Prezzo su listini dal 10 al 999 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output.</i> <i>Nota: rispetto alle variabili Sconti e Provvigioni su Listino, presenta un indice fino a 999 elementi (piuttosto che 18) in quanto nell'anagrafica articolo sono scorribili a video [analogamente al dizionario aprz(999)]. Nelle variabili sconti e provvigioni su Listino il limite è 18 elementi, perchè sono quelli statici a video (analogamente al dizionario aysco(aycod(),aylst())) e che, sui 999 disponibili ne sono configurabili 18 da menu [ZCN] + [Parametri articoli] + [Listini da visualizzare].</i>
_ARSCO\$(1-9)	Sconti su listino dal 1 al 9; i caratteri "*****" significano "non visibile"
_ARSCOES\$(10-18)	Sconti su listino dal 10 al 18 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
_ARCON\$(1-9)	Sconti su listino dal 1 al 9; per la tipologia "cascata"
_ARSCONES\$(10-18)	Sconti su listino dal 10 al 18 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
_ARPROTP\$(1-9)	Provvigioni su listino dal 1 al 9
_ARPROTPE\$(10-18)	Provvigioni su listino dal 10 al 18 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
_ARPROFORM\$(1-9)	Provvigione
_ARPROFORMES\$(10-18)	Provvigioni su listino dal 10 al 18 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
_ARPROAGENS\$(1-9)	Provvigioni su listino dal 1 al 9
_ARPROAGENES\$(10-18)	Provvigioni su listino dal 10 al 18 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
_ARPROCOND(1-9)	Provvigioni su listino dal 1 al 9
_ARPROCONDE(10-18)	Provvigioni su listino dal 10 al 18 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
_ARPRO(1-9)	Provvigione; solo GET in multiagente
_ARPROE(10-18)	Provvigioni su listino dal 10 al 18 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
_ARPER(1-9)	Percentuale (%) Formazione prezzi su listino
_ARPERE(10-18)	Variazione prezzo su listino dal 10 al 18 - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
_ARNUMLISCON(18)	Listini per condizioni commerciali - <i>Listini Estesi di 18 elementi a video fra quelli selezionabili (da 10 a 999). I primi 9 elementi sono solamente di output</i>
EXTRA	
_ARVEX	Codice valuta Extra
_AREXG	Extra fisso
_AREXA	Extra a
_AREXB	Extra b
_AREXC	Extra c
PRODUZIONE	
_ARCTG\$	Articolo consumo x taglia
_ARFRM\$	Articolo di tipo forma
_ARMAC\$	Valida per articoli di tipo "L" che sono macchinari accetta valori S/N
_ARRCM\$	Categorizzazione di un articolo come tipologia ricambio (vale solo per articoli di tipo "A" o "P"); trattasi di una variabile flag che accetta i valori S/N ed è attiva solo con modulo IFTECHNOLOGY presente nel codice contratto.
_ARPESWMS\$	Attiva pesata WMS, accetta valori S/N
GRUPPO MERCEOLOGICO	
_ARGRPMER\$	Codice Gruppo Merceologico
_ARCATWEB\$	Categoria Web
NATURA	
_ARNATURA\$	Codice Natura
LOTTE	
_ARRDT\$	Data inizio gestione rintracciabilità
_ARRTL\$	Tipo lotto; variabile se non movimento
_ARRRI\$	Richiesta lotto
_ARRSA\$	Selezione automatica variabile se non movimentato
ENASARCO	
_ARENAS\$	Gestione Enasarco; la gestione Ritenuta d'Acconto è riportata da _ARPRIT\$ indicata nell'elenco di _AR relative alla Parcellazione - Prestazioni Clienti
RATEI / RISCONTI	
_ARPRR	Periodo in mesi del rateo Risconto
_ARGRR\$	Gestione ratei risconti -S/N-
AGROALIMENTARE	
_ARAGR\$	Agroalimentare; valori "A", "D", ""
DATI MAGAZZINO E UBICAZIONI	
_ARUBI\$	Ubicazione
_ARSUB\$	Sotto ubicazione
_ARCAD\$	Ultima data di carico

_ARCA\$	Ultima data di scarico
_ARORD\$	Ultima data ordini/impegni
_ARDTM\$	Ultima data movimenti
_ARSMI	Scorta minima
_AR SMA	Scorta massima
_ARUID	Id Ubicazione; nota: in gestione Ubicazione Estesa prevale su _ARUBI e _AR SUB
_ARLAR	Larghezza; dizionario "arlar"
_ARALZ	Altezza; dizionario "aralz"
_ARPRF	Profondità; dizionario "arprf"
_ARPES	Peso; dizionario "arpes"
_ARTMI\$	Temperatura Minima (= "" se non gestito, altrimenti da "-100,0" a "+998,0"); diz. "artmi"; v. _PUTMIN\$
_ARTMA\$	Temperatura Massima (= "" se non gestito, altrimenti da "-100,0" a "+998,0"); diz. "artma"; v. _PUTMAX\$
PROGRESSIVI ANNO/MAGAZZINO	
variabili di sola lettura; le modifiche sono consentite con i documenti rettificativi o con gli interventi sui documenti di carico/scarico; esempio: cancellazione e reimmissione nella data di interesse e successiva riorganizzazione.	
Le variabili dipendenti dal numero di magazzino rilevano in base al magazzino considerato; in Collage, il magazzino totale (0) richiede l'impiego del dizionario, altrimenti la struttura risulterebbe non valorizzata.	
_ARINQ	Quantità inventario
_ARINT(16)	Quantità inventario Taglia 1 – 16
_ARINTXL(16)	Quantità inventario Taglia 17 – 32
_ARINV	Valore inventario
_ARTIN	Tara inventario
_ARCIN	Colli inventario
_ARCAR	Quantità carico
_ARCAT(16)	Quantità carico Taglia 1 – 16
_ARCATXL(16)	Quantità carico Taglia 17 – 32
_ARCAV	Valore carico
_ARTCA	Tara carico
_ARCCA	Colli carico
_ARSCA	Quantità scarico
_AR SCT(16)	Quantità scarico Taglia 1 – 16
_AR SCTXL(16)	Quantità scarico Taglia 17 – 32
_ARSCV	Valore scarico
_ARTSC	Tara scarico
_ARCSC	Colli scarico
ORDINI	
_ARORF	Ordini fornitori
_ARORT(16)	Ordini fornitori Taglia 1 – 16
_ARORTXL(16)	Ordini fornitori Taglia 17 – 32
_ARCLI	Clienti impegni
_ARCTI(16)	Clienti impegni Taglia 1 – 16
_ARCTIXL(16)	Clienti impegni Taglia 17 – 32
_ARCLS	Clienti sospesi
_ARCTS(16)	Clienti sospesi Taglia 1 – 16
_ARCTSXL(16)	Clienti sospesi Taglia 17 – 32
_ARCLA	Clienti automatici/da produrre
_ARCTA(16)	Clienti automatici/da produrre Taglia 1 – 16
_ARCTAXL(16)	Clienti automatici/da produrre Taglia 17 – 32
_ARPRD	In produzione
_ARCTP(16)	In produzione Taglia 1 – 16
_ARCTPXL(16)	In produzione Taglia 17 – 32
INTRACEE	
_ARINOS\$	Nomenclatura combinata
_ARIMAS\$	Massa netta
_ARIKM	Coefficiente massa netta
_ARIUS\$	Unità supplementare
_ARIDUS\$	Descrizione unità supplementare
_ARIKU	Coefficiente unità supplementare
_ARIPR\$	Provincia origine merce CESSIONI
_ARIPC\$	Paese origine merce CESSIONI
_ARIPA\$	Paese origine merce ACQUISTI
_ARICS	Codice servizio
_ARIVS	Valore statistico unitario
_ARSMN\$	Gestione RSM: Non rimborsabile TM1 e TM2
STATO ANAGRAFICA	
_ARMRA\$(12)	Stato anagrafica attiva; vedere mirroring (dbmirror); vedere Passweb; Vedere tabella in menu 'AES' e tasti funzione [F5] "Altri dati anagrafici" e [Shift+F11] "Stato anagrafica" in Anagrafica Articolo. La funzione potrebbe non essere disponibile in determinate configurazioni (es. Live).
_ARMRV\$(12)	Stato anagrafica variata; vedere mirroring (dbmirror); vedere Passweb; Vedere tabella in menu 'AES' e tasti funzione [F5] "Altri dati anagrafici" e [Shift+F11] "Stato anagrafica" in Anagrafica Articolo. La funzione potrebbe non essere disponibile in determinate configurazioni (es. Live).
IMMAGINI	
_ARIMM\$	Immagine articolo (*a)
_ARCAT\$	Immagine catalogo (*a)

_ARICO\$	Immagine Icona (*a)
_ARDTIMG\$	Data ultima modifica immagini articolo (viene aggiornata con la data di sistema)
	<p>(*a) variabili di una cartella del percorso dati aziendali (es...\\azi\imgart\4200000000000000_c.bmp); leggibili con GETAR; in scrittura con PUTAR, l'azzeramento della variabile cancella l'immagine associata, la non modifica della variabile non implica variazioni, la modifica dell'eventuale valore della variabile comporta la ricerca del file specificato (se trovato è convertito nel nome atteso dal Gestionale, se non trovato avvisa che l'immagine non viene trovata e blocca la PUTAR). Le variabili sono sempre valorizzate sia che il file esiste, sia che il file immagine non esiste.</p> <p>In ambiente Sprix Mobile l'associazione della variabile di struttura _ARCAT\$ con la variabile di lista di Sprix Mobile _WLROWICO\$ consente l'associazione dell'immagine dell'icona dell'articolo nel catalogo sul dispositivo Mobile. Es. _WLROWICO\$ = _ARCAT\$</p>
VIDEATE ARTICOLO	
_ARVU1\$(26)	26 videate anagrafica non riportabile;
_ARVU2\$(26)	26 videate anagrafica riportabile;
_ARWWW\$(26,24)	<p>26 videate <u>aggiuntive</u> con 24 campi; l'istruzione GET converte da num → str\$, la PUT viceversa; Videate _ARWWW\$(numero videate, numero di campo) è utilizzata anche per videate Passweb</p> <p>I dati delle videate sono gestibili e valorizzabili e sono impiegabili per gestire le logiche che si vogliono attribuire ai dati aggiuntivi di Passweb, anche in modalità di operazione pianificata. Per esempio, una procedura Sprix che imposti a "S" o "N" gli appositi campi delle videate in base a logiche per Passweb (data di creazione articoli per gestire le "novità" oppure la data di scadenza della particolarità per gestire le "Offerte").</p>
PARCELLA PRESTAZIONI CLIENTI	
_ARPPER\$	Periodicità
_ARPMES	Periodicità mese
_ARPRIT\$	Ritenuta d'acconto
_ARPCAS\$	Cassa previdenza
_ARGSIS\$	Gestione separata INPS\$
_ARENPA\$	Soggetto ENPAM (valori S=Si, N=No, R= Si - RDA Imponibile RDA al netto delle ritenute d'acconto)
_ARENPAL\$	Soggetto ENPALS (valori S=Si, N=No)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

ARCHVIEW - Tabelle ABI e CAP

ARCHVIEW

Variabili di struttura dell'istruzione su tabelle ABI e CAP (ARCHVIEW)	
_ARCHVIEWNUM	Numero di colonne gestite nella gestione dell'archivio: 8 per la Tabella Banche e 5 per la tabella CAP;
_ARCHVIEW\$(1)	Valore ritornato e/o di filtro della colonna ABI/LOCALITA' della riga selezionata e/o filtrata
_ARCHVIEW\$(2)	Valore ritornato e/o di filtro della colonna CAB/CAP della riga selezionata e/o filtrata
_ARCHVIEW\$(3)	Valore ritornato e/o di filtro della colonna DENOMINAZIONE BANCA/PROVINCIA della riga selezionata e/o filtrata
_ARCHVIEW\$(4)	Valore ritornato e/o di filtro della colonna SPORTELLO/COMUNE della riga selezionata e/o filtrata
_ARCHVIEW\$(5)	Valore ritornato e/o di filtro della colonna CAP/CODICE POSTALE della riga selezionata e/o filtrata
_ARCHVIEW\$(6)	Valore ritornato e/o di filtro della colonna LOCALITA' della riga selezionata e/o filtrata
_ARCHVIEW\$(7)	Valore ritornato e/o di filtro della colonna PROVINCIA della riga selezionata e/o filtrata
_ARCHVIEW\$(8)	Valore ritornato e/o di filtro della colonna INDIRIZZO della riga selezionata e/o filtrata
_ARCHVIEW\$(1)	Codice catastale del comune (v. parametro arch=3 di ARCHVIEW)
_ARCHVIEW\$(2)	Descrizione del comune (v. parametro arch=3 di ARCHVIEW)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

ARGCMD

ARGCMD\$

<p>Variabile che contiene l'argomento proveniente dal parametro "-v" del <comandospx\$>. E' utilizzato per esecuzioni di Sprix a cui passare una stringa contenuta dal parametro "-v". Lo scheduler del sistema operativo (Operazioni Pianificate, Cron, ecc.) consente le esecuzioni temporizzate (es. dal prompt dei comandi), di appositi eseguibili Passepartout (come ad esempio l'esecutore a riga di comando) in grado di lanciare SPRIX analogamente a quanto avviene con la SHELLSPRIX.</p> <p>Verificare sempre i riferimenti a percorsi esistenti. Notare: la corretta valorizzazione di _ARGCMD\$ dipende dalla costruzione della stringa passata dal parametro "-v" della SHELLSPRIX. Per approfondimenti consultare SHELLSPRIX e le operazioni batch documentate nelle Nozioni Generali (componente client/server) del Manuale Generale.</p> <p>Attenzione! si ricorda che dalla versione 2020A (v750) è disponibile lo <u>scheduler integrato</u> di Sprix nel menu Servizi - Configurazioni - Operazioni Pianificate.</p> <p style="text-align: center;">Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>

AU - Autorizzazione Utenti

AU

Variabili di struttura per la gestione permessi relativa all'Autorizzazione Utenti (GETAUTORIZUTE)	
_AUNUMVAL	Numero degli elementi caricati nel vettore _AUVAL\$()

_AUVAL\$()	<p>Vettore contenente i valori del filtro richiesto dal parametro 'filtro\$'. Nello specifico:</p> <ul style="list-style-type: none"> - "_PCAGE" Codice conto agente con la sintassi gmm.ccccc. (_AUNUMVAL ritorna elementi multipli) - "_PCCST" Codice categoria statistica. (_AUNUMVAL ritorna elementi multipli) - "_PCONA" Codice di zona. (_AUNUMVAL ritorna elementi multipli) - "_PCPAE" Sigla paese. (_AUNUMVAL ritorna un solo elemento) - "_PCCOD" Codice Cliente / Fornitore ammessi, oppure -1 se non è ammesso accedere al tipo di conto interessato. Ritorna l'eventuale codice cliente o fornitore a cui è limitato l'accesso (in base a quanto chiesto tramite la stringa <TIPOCONTO=C> nei parametri facoltativi). - "_ARTIP" Contiene nel solo primo elemento (_AUNUMVAL), l'elenco dei tipi articolo ammessi, o valore vuoto per indicare che sono ammessi tutti i tipi articolo - "_ARSTA" Categoria statistica articolo. (_AUNUMVAL ritorna elementi multipli) - "_ARGRPMER" Gruppo merceologico articolo (_AUNUMVAL ritorna elementi multipli) - "_ARNATURA" Natura articolo (_AUNUMVAL ritorna elementi multipli) - "_MMAGE" Codice conto agente con la sintassi gmm.ccccc (_AUNUMVAL ritorna elementi multipli) - "_MMSIG" ogni elemento del vettore contiene una stringa "XX,N" che corrisponde alla coppia tipo documento XX e serie N (i tipi documento non elencati non sono vincolati ad alcun valore) (_AUNUMVAL ritorna 'n' elementi, tanti quanti sono i filtri impostati) - "_MMMAG" ritorna una stringa "NN,X" che corrisponde alla coppia numero magazzino (NN) ed esclusivo (X, vale 'S' se l'uso del magazzino è esclusivo oppure 'N' in caso contrario). (_AUNUMVAL ritorna un solo elemento) - "_MMCLI" Filtro sul conto. Ritorna "S" se l'utente può visualizzare solo i documenti il cui cliente o fornitore soddisfa i propri filtri, altrimenti "N" (vedere campo Accesso filtrato intestatari).
_AUNONOIMP\$	Da testare solo se presenti i valori nell'array precedente. Vale "S" se per la variabile richiesta sono accessibili anche i dati con valore non impostato, vale "N" in caso contrario. Ritorna "N" se non può accedere ai documenti senza cliente o fornitore, altrimenti "S".
_AUNUMLIS\$	Variabile che per il filtro\$=_MMMAG ritorna la stringa dei listini da visualizzare. La stringa viene tornata nella forma per come è scritta, e cioè, i listini sono separati dalla virgola.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_BUFCOM

_BUFCOM\$

<p>Variabile contenente il buffer di memorizzazione dei byte letti da GETCOM. Se mancano dei caratteri da leggere, la GETCOM attende il <timeout> di OPENCOM.</p>

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_CDF - Codice Fiscale

_CDF

Variabili di struttura archivi di Creazione o Scorporo del Codice Fiscale (CREACF)	
_CDFCOGN\$	Cognome
_CDFNOME\$	Nome
_CDFDATN\$	(formato data "GGMMAAAA")
_CDFSESSO\$	Sesso
_CDFDCOM\$	Codice Comune
_CDFDSCOM\$	Descrizione Comune
_CDFPROV\$	Provincia
_CDFISCALE\$	Codice Fiscale
_CDFERR	Codice errore. Valori: Valore = 0 (esito positivo dell'operazione, nessun messaggio in _CDFERR\$, che risulta vuota); Valore <> 0 (esito negativo dell'operazione, l'errore è descritto in _CDFERR\$);
_CDFERR\$	Descrizione del Codice errore

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_CDG - Contabilità Analitica

_CDG

Variabili di struttura archivi di Contabilità Analitica (GETCDG - PUTCDG - DELCDG)	
_CDGNR	Numero di elementi dell'array dinamico (numero righe) della struttura _CDG. Sola lettura GETCDG.
_CDGTIPREG\$(N)	Tipo Registrazione: A = MATRICI, P = PREVENTIVI, E = EXTRA, R = PRIMANOTA, M = MOVIMENTI, MAGAZZINO, F = ORDINI FORNITORE, C = ORDINI CLIENTI; (*)
_CDGSN\$(N)	Considera movimento su CDG S/N; (*)
_CDGDDTCOMP\$(N)	Data Competenza; (*)
_CDGSCODANL\$(N)	Codice Analitico; (*)
_CDGSCOMM\$(I)	Codice Commessa; (*)
_CDGSAREA\$(I)	Codice Area; (*)
_CDGCTOCG\$(I)	Codice Contropartita righe dei movimenti; (*)
_CDGCODARMO\$(N)	Codice Articolo Movimento
_CDGSIGDOC\$(N)	Sigla Documento di magazzino (*)
_CDGISERDOC\$(N)	Serie Documento di magazzino (*)

_CDGLNUMDOC(N)	Numero Documento di magazzino (*)
_CDGDDTDOC\$(N)	Data Documento di magazzino (*)
_CDGCTOT\$(N)	Conto Cliente/Fornitore Documento
_CDGNTEST(N)	Numero testata documento di origine di riga; se diverso da uno richiede _CDGPROGRIG(N) equivalente a _MMNCRERIGA(N); (*)
_CDGCAUPRN\$(N)	Causale movimento di primanota
_CDGCPROT\$(N)	Protocollo –Registro-
_CDGIPROTS(N)	Serie protocollo
_CDGLPROTN(N)	Numero protocollo
_CDGDDTREG\$(N)	Data registrazione di primanota
_CDGNDOCNP(N)	Numero di documento di primanota
_CDGNOTA\$(N)	Nota riga controllo di gestione
_CDGFVALG(N)	Importo valuta di gestione
_CDGIDEC(N)	Numero di decimali gestiti in variabile quantità
_CDGFQTA(N)	Quantità
_CDGSTRIPR\$(N)	Stato riga preventivo: T (trasformata) oppure ' ' –spazio- (sospesa)
_CDGCATSTA\$(N)	Tipo movimento CDG presente sul movimento (' ' –spazio- è default come tabella automatismi causali contabili/documenti magazzino) altrimenti valore inserito dall'utente
_CDGIVAL(N)	Valuta Importo di sola lettura
_CDGFVAL(N)	Importo in valuta di sola lettura
_CDGPROGANA(N)	numero progressivo -numero riga- relativo all'analitico
_CDGPROGRIG(N)	numero riga documento: in caso di primanota è il numero di riga visibile a video; per i movimenti di magazzino, invece, è il valore rilasciato dalla _MMNCRERIGA() o dalla mmrig() in caso di trasformazione dei documenti, altresì è il numero di riga del documento; (*)
_CDGPROGEXT(N)	progressivo per extra, da gestire solo per movimenti extra e viene incrementato dalla procedura per la gestione di movimenti fra essi identici.
_CDGRDOCSOS(N)	numero riga di documenti sospesi è il valore rilasciato da _MMNCRERIGA o dalla mmrig()
_CDGRNPRPN(N)	numero primo record assegnato –da 1 a 16777214-
_CDGTIPEXT\$(N)	tipo di Extra P/M/G; (*)
_CDGCESCDS\$(N)	Codice Cespite
_CDGCESANNO(N)	Anno Cespite
_CDGCESCONT\$(N)	Conto di contabilità relativo al Cespite
	(*) variabili minime per la scrittura di un movimento di analitica collegata ad un movimento di magazzino; per gli importi e le quantità queste sono prese direttamente dalla riga del movimento a cui fanno riferimento. Il movimento extra deve essere tale, quindi non è agganciabile ai reali documenti che derivassero dalla gestione contabile o di magazzino. Questo lo si vede operando con le funzioni standard: la registrazione di un movimento extra può chiedere il tipo di movimento, ma è solo un dato statistico e quindi, non un aggancio a un reale documento.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_CFR - Referenti

_CFR

Variabili di struttura archivi dei Referenti (GETCFR – PUTCFR – DELCFR)	
_CFRNELE	Numero di elementi dell'array della struttura _CFR, referenti presenti. Sola lettura GETCFR;
_CFRCOD\$(1-30)	Codice del Cliente/Fornitore; si può inserire anche un codice contatto;
_CFRPRG(1-30)	Codice Referente; è il collegamento fra referente registrato e array;
_CFRPOS(1-30)	Cod.Posizione Referente in tabella posizioni;
_CFRNOME\$(1-30)	Nome Referente;
_CFRNOTA\$(1-30)	Nota Referente;
_CFRTEL\$(1-30)	Telefono Referente;
_CFRCELL\$(1-30)	Cellulare Referente; per l'azzeramento occorre impostare con uno spazio (= " ");
_CFRFAX\$(1-30)	Fax Referente;
_CFRMAIL\$(1-30)	Indirizzo e-mail Referente;
_CFRWEB\$(1-30)	Indirizzo Internet Referente;
_CFRTP\$(1-30)	Tipo Referente R= Cli/For, O=Contatti;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_CFRERR

_CFRERR

_CFRERR\$

Variabili che ritornano codice e descrizione dell'operazione sui Referenti; ai messaggi è anteposto "Rigo: nn" (nn = elemento array)	
0	Esito positivo dell'operazione (nessun messaggio, il campo è vuoto)
1	Referente assente (in GET)
1	Codice cli/for [codice] intestatario non trovato in anagrafica cioè non codificato, in PUT
1	Codice cli/for intestatario non specificato non valorizzato su array 1, in PUT
2	Codice cli/for intestatario diverso da precedente (array con _CFRCOD\$ diversi, in PUT)

3	Codice referente [codice] errato, (ammessi da 1 a 9) (in PUT)
4	Codice referente [codice] già presente su rigo [rigo] (in PUT)
5	Dati referente non valorizzati (in PUT)
6	Posizione referente già assegnata ad altro referente (in PUT)
7	Tipo referente errato (accettati 'R' per Cli/For, 'O' per Contatti) (in PUT)
10	Referenti assenti (in DEL con parametro <tipo> uguale a zero)
11	Codice referente non esistente (in DEL con param. <tipo> <> da zero)
12	Codice referente errato (in DEL con parametro <tipo> diverso da zero)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_CM - Calendario Agenda Attività Utente

_CM

Variabili di struttura archivi di Calendario dell'Agenda Attività Utente (GETCM – PUTCM – DELCM)	
_CMDATA\$()	Data obbligatoria in formato "AAAAMMGG". La lettura e la cancellazione consentono il 4° parametro [adata\$] che è una stringa facoltativa in formato "AAAAMMGG" il cui inserimento determina l'elaborazione di tutti i calendari dell'utente compresi fra le date impostate.
_CMNCRE()	Id Agenda (numero record creazione); formato numerico (da passare in formato stringa per lettura o cancellazione)
_CMUTE\$()	Id Nome Utente;
_CMSORA\$()	Orario Inizio, "HH:MM";
_CMEORA\$()	Orario Fine, formato "HH:MM";
_CMDUR\$()	Tempo netto di durata effettiva, formato "HH:MM"; se lasciato vuoto viene calcolato;
_CMFDUR\$()	Forzatura durata, formato "" oppure "S"; se lasciato vuoto viene calcolato;
_CMNOTA\$()	Nota;
_CMPROM\$()	Promemoria, formato "M/O/G/S"
_CMTPROM()	Promemoria formato numerico relazionato al precedente;
_CMDATT\$()	Descrizione attività;
_CMSCAT\$()	Colore (categoria statistica): "R/G/Y/B/V/W/L/P/O";

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_CO - Anagrafica Commesse

_CO

Variabili di struttura archivi di Anagrafiche Commesse (GETCOMM – PUTCOMM)	
ANAGRAFICA GENERALE	
_COCOM\$	Codice
_CODAL\$	Codice Alternativo
_CODESS	Descrizione
_COPAD\$	Codice Padre
_CODEAS	Descrizione Aggiuntiva
_CONOT\$	Note
_COCLI\$	Cliente
_COMER\$	Destinatario Merce
_CORESS	Responsabile
_COPAG	Codice pagamento
_COSTA\$	Stato Commessa
_COANN\$	Commessa Sostitutiva
_CODTP\$	Data Preventivo
_CODTA\$	Data Apertura
_CODTO\$	Data Ordine
_CODTS\$	Data Spedizione
_CODTG\$	Data Chiusa in Garanzia
_CODTD\$	Data Chiusura
_CODTANN\$	Data Annullamento
_CODTSOS\$	Data Sostituzione
_CODTC\$	Data Creazione
_CODTU\$	Data Ultimo Aggiornamento
PAGAMENTI	
_COTPS\$(250)	Tipo Pagamento
_COTPR\$(250)	% o valore
_CORAP(250)	%
_CORAV(250)	Valore
_COSCA\$(250)	Scadenza
_COTPP\$(250)	Stato Pagamento
_CODEP\$(250)	Descrizione
CONDIZIONI COMMERCIALI	
_COAGE\$	Agente
_COPPR	% Provvigione
_COTPCOM\$	Tipo
_COZONA\$	Zona
CONTRATTO	
_COCDT\$	Data contratto

_CO1DT\$	Data 1^collaudo
_CO2DT\$	Data 2^collaudo
_COLDT\$	Data consegna lavori
_COANT	% Anticipo
_CORIC	% Ricarico Prezzo
_COADT\$(3)	Data Anticipo
_COAIM(3)	Anticipo
_COAPE(3)	Penale
_COADE\$(3)	Descrizione
ALLEGATI	
_COALL\$(4)	Allegati
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_COD - codici in GETPC2 - GETCODART - GETARVAR

_COD

_CODPICONR	Numero di elementi ritornati da GETPC2. (struttura PC)
_CODPICO\$()	Variabile Array valorizzata in base al numero di elementi ritornati da GETPC2 (struttura PC)
_CODARTNUM	Variabile numerica che ritorna quanti Codici Anagrafica Articoli trovati da GETCODART. es. (dalla v711) (struttura AR)
_CODART\$()	Variabile Array stringa con Codici di Anagrafica Articoli trovati da GETCODART. es. (dalla v711) (struttura AR)
_CODARVAR	Variabile che ritorna il prossimo articolo variato dopo la data di partenza che viene impostata nell'istruzione GETARVAR, e prima della data di termine se viene impostata. Se nella GETARVAR non viene impostato un codice articolo di partenza (cioè parametro codice articolo è = ""), la variabile ritorna il primo che soddisfa la data di partenza.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento	

_CT - contenuto di CREATELIST

_CT

Variabili di struttura archivi dell'istruzione CREATELIST	
_CTVTOTRIG	Numero di righe totali da vedere a video se impostato a zero la lista viene aperta a tutto schermo;
_CTVTOTCOL	Numero di colonne totali da vedere a video minimo 1 massimo 20 la variabile è obbligatoria;
_CTTYPECOL\$()	Tipo di colonna accetta i valori I=icona, S=Stringa, N=numero, D=Data. L'istruzione SETTF elenca i tipi di icone
_CTTITCOL\$()	Intestazione della colonna;
_CTINDRIGSEL()	Numero dell'indice selezionato (quale riga è stata selezionata); porla = 0 (azzerarla) per impedire l'eventuale elaborazione stabilita da _CTAZIONE\$. Se prima dell'apertura è valorizzata insieme a _CTTOTRIGSEL le relative righe risulteranno già selezionate; nel caso di modolista di tipo 2 la _CTTOTRIGSEL va valorizzata con il totale delle selezioni presenti (fino a 50 preselezioni).
_CTTOTRIGSEL	Quantità totali degli indici selezionati (quante righe sono state selezionate). Nota: premendo F10 o Invio viene confermata, almeno, la riga su cui si è posizionati; se non si vogliono effettuare selezioni, occorre utilizzare il tasto ESC.
_CTLENCOL()	Lunghezza della colonna da considerare per la ricerca del dato;
_CTVWLENCOL()	Lunghezza della colonna da vedere a video;
_CTNUMCOLORD	Numero colonna di ordinamento iniziale; se zero imposta la prima altrimenti il numero della colonna; se maggiore delle colonne restituisce errore; se impostato negativo esegue ordinamento decrescente;
_CTORDAUTO\$	se impostare ordinamento iniziale automatico, accetta valore S o N
_CTPOSCURFI	se impostato con un valore diverso da zero imposta il cursore nell'ultima riga dell'elenco disponibile
_CTFILTEXT\$	Imposta il valore come filtro di ricerca. Se presenti più valori questi vengono mostrati. Se il filtro identifica un valore univoco, la finestra non viene mostrata ed è ritornato il numero dell'elemento filtrato.
_CTVALCOL1\$()	Valori della colonna 1
_CTVALCOL2\$()	Valori della colonna 2
_CTVALCOL3\$()	Valori della colonna 3
_CTVALCOL4\$()	Valori della colonna 4
_CTVALCOL5\$()	Valori della colonna 5
_CTVALCOL6\$()	Valori della colonna 6
_CTVALCOL7\$()	Valori della colonna 7
_CTVALCOL8\$()	Valori della colonna 8
_CTVALCOL9\$()	Valori della colonna 9
_CTVALCOL10\$()	Valori della colonna 10
_CTVALCOL11\$()	Valori della colonna 11
_CTVALCOL12\$()	Valori della colonna 12
_CTVALCOL13\$()	Valori della colonna 13
_CTVALCOL14\$()	Valori della colonna 14
_CTVALCOL15\$()	Valori della colonna 15
_CTVALCOL16\$()	Valori della colonna 16
_CTVALCOL17\$()	Valori della colonna 17
_CTVALCOL18\$()	Valori della colonna 18
_CTVALCOL19\$()	Valori della colonna 19
_CTVALCOL20\$()	Valori della colonna 20
Variabili per tasti funzione personalizzate	
_CTEVETIC\$	Parte variabile delle etichette per la gestione dei tasti funzione della successiva CREATELIST a cui si riferiscono . Esempio se _CTEVETIC\$= "PROVA", le etichette cercate da CREATELIST per gestire i tasti funzione si dovranno chiamare ON_604_PROVA: ON_611_PROVA: ecc.
_CTTASTFUNES\$	Variabile necessaria per gestire tasti funzione Invio "501" e/o Esc "505" nella CREATELIST "0"

	La variabile va impostata prima della array <code>_CTTASTFUN\$()</code> relativa ad altri tasti funzione; Esempi: <code>_CTTASTFUN\$="501"</code> , <code>_CTTASTFUN\$="501,505"</code> Nel caso di multiselezione con la conferma fatta con F10 il tasto 501 non è disponibile in quanto riservato
<code>_CTTASTFUN\$()</code>	Array nei cui elementi sono descritti i Tasti Funzione da gestire in CREATELIST dove ogni elemento e' un tasto con la sintassi che consente di specificare anche il tooltip [A S]<nnn>[<Descrizione breve> [[[<Descrizione> [[[<ToolTip>]]]]]]
<code>_CTAZIONE\$</code>	Azione che verra eseguita sui dati. I valori (costanti) gestiti per variabile <code>_CTAZIONE\$</code> per indicare a CREATELIST quale azione intraprendere dopo il termine del codice scritto in SPRIX per la gestione del tasto funzione premuto, sono: <code>"SCF_OPERAZIONE_ANNULLATA"</code> - Nessuna azione verra intrapresa dalla CREATELIST <code>"SCF_RIDISEGNA_PPRIMA_POS"</code> - Reimposta la riga corrente alla prima riga <code>"SCF_RIDISEGNA_CANCEL_POS"</code> - Cancella la riga corrente <code>"SCF_RIDISEGNA_INSER_POS"</code> - Aggiungendo delle righe dopo l'attuale ultima negli array dei valori, questi vengono usati per creare nuove righe da inserire <code>"SCF_RIDISEGNA_CHIUDI"</code> - Causa la chiusura della FinestraElenco <code>"SCF_RIDISEGNA_VER_MODIFICA"</code> - Consente di ridisegnare una riga con dei valori che nel codice di gestione del tasto sono stati modificati al momento modifica solo la riga corrente (e non quella selezionata in caso di gestione con multiselezione) non e stata gestita la possibilita di selezionare piu righe ed effettuare la variazione di tutte <code>"SCF_RIDISEGNA_CANCEL_SELEZ"</code> - Consente di cancellare tutte le righe selezionate
<code>_CTCURRRIGA</code>	Riga corrente
<code>_CTNUMRIGHE</code>	Numero delle righe attualmente presenti nell'elenco
<code>_CTINSRIGHE</code>	Numero delle righe aggiunte dopo quelle correntemente presenti (<code>_CTNUMRIGHE</code>)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

DA - Anagrafica Indirizzi Spedizione

DA

Variabili struttura archivi di Anagrafica Indirizzi Spedizione (GETIND – GETATTIND – PUTIND)	
<code>_DACTO\$</code>	Codice conto Cliente/Fornitore
<code>_DACIN\$</code>	Codice Indirizzo o AUTO
<code>_DADES\$</code>	Descrizione
<code>_DAIND\$</code>	Indirizzo
<code>_DACAP\$</code>	C.A.P.
<code>_DALOC\$</code>	Località
<code>_DAPRO\$</code>	Provincia
<code>_DANAZ\$</code>	Nazionalità
<code>_DAPAE\$</code>	Paese ISO2
<code>_DATEL\$</code>	Telefono
<code>_DAFAX\$</code>	Fax
<code>_DAMAI\$</code>	Indirizzo e-Mail
<code>_DAINT\$</code>	Indirizzo Internet
<code>_DACHI\$</code>	Chius. Sett. oppure stringa libera
<code>_DACON\$</code>	Riferimento
<code>_DACEL\$</code>	Cellulare Riferimento
<code>_DANOT\$</code>	Note
<code>_DAALL\$</code>	Nome file allegato
<code>_DADVA\$</code>	Data validità
<code>_DADCR\$</code>	Data creazione
<code>_DADAG\$</code>	Data aggiornamento
<code>_DACTS\$</code>	Codice conto sostitutivo
<code>_DACIS\$</code>	Codice indirizzo sostitutivo
<code>_DAANN\$</code>	Annullato
<code>_DARAG\$</code>	Raggruppa documenti S/N
<code>_DACIG\$</code>	Codice CIG
<code>_DACUP\$</code>	Codice CUP
<code>_DATPI\$</code>	Se PassWeb attivo vale "v" = vendita , "r" = ritiro
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

DC - Anagrafica Contatti

DC

Variabili struttura archivi Anagrafica Contatti (GETCON – PUTCON – DELCON)	
<code>_DCCTO\$</code>	Codice contatto
<code>_DCINT\$</code>	Indirizzo Internet
<code>_DCDES\$</code>	Descrizione
<code>_DCNOT\$</code>	Note
<code>_DCCOG\$</code>	Cognome
<code>_DCALL\$</code>	Nome file allegato
<code>_DCIND\$</code>	Indirizzo
<code>_DCDVA\$</code>	Data validità

_DCCAP\$	C.A.P.
_DCDCR\$	Data creazione
_DCLOC\$	Località
_DCDAG\$	Data aggiornamento anagrafica
_DCPRO\$	Provincia
_DCPIV\$	Partita iva
_DCNAZ\$	Nazionalità
_DCCFI\$	Codice fiscale
_DCPAE\$	Paese Iso
_DCPFI\$	Persona fisica S/N
_DCTEL\$	Telefono
_DCRIC\$	Nome di ricerca
_DCFAX\$	Fax
_DCPRI\$	Privato S/N - Nota bene: la corrispondente variabile dizionario non esiste, ma il valore puo' essere desunto consultando dcpiv() che vale "P" quando il flag è impostato a "S"
_DCMAI\$	Indirizzo e-mail
_DCCODPC\$	codice piano dei conti se = 0 ritorna nelle variabili _DCPCCLI\$ _DCPCFOR\$ i codici a cui l'anagrafia fa riferimento se = 1 anagrafica non collegata altrimenti riporta il codice del piano dei conti a cui è collegata
_DCPCCLI\$	se _DCCODPC\$ = 0 riporta il codice conto clientesolo OUTPUT
_DCPCFOR\$	se _DCCODPC\$ = 0 riporta il codice conto fornitore dolo OUTPUT
_DCPAG	Codice pagamento
_DCLIS	Numero listino
_DCSCO	Categoria sconto
_DCPRV	Categoria provvigione
_DCASI\$	Codice assoggettamento IVA
_DCLIN	Lingua estera
_DCAGE\$	Agente
_DCCAG	Condizioni agente
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_DES612

_DES612\$

<p>Variabile con cui stabilire la descrizione del tasto [F12] negli eventi ON_612_ALL, ON_612_CODICECAMPO, oppure anche al momento opportuno (esempio al ON_IN_CODICE: del medesimo campo). Se più Collage sono caricati contemporaneamente in memoria dalla stessa etichetta [F12], la variabile _DES612\$ visualizza le descrizioni [Altre Funzioni] per scegliere quale routine eseguire. esempio</p> <p style="text-align: center;">Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>
--

_DIRFILE

_DIRFILE\$()

<p>Variabile array stringa valorizzata dall'istruzione LEGGIDIR e che ritorna i nomi dei file contenuti nel percorso della directory.</p> <p style="text-align: center;">Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>

_DK - Collage DataTrek

_DK

Variabili di struttura di Collage Datatrek	
_DKSTOPSTATE	Impostata ad 1 blocca l'esecuzione della procedura tornando sul primo campo d'input
_DKSETTF\$	Imposta i tasti funzioni personali
_DKERRCAMPO\$()	Errore che si vuole visualizzare a video (DTK_GETVALUE)
_DKERRNMCMP\$()	Contiene il nome del campo della tabella sul quale mostrare l'errore (DTK_GETVALUE)
_DKNOMCAMPO\$()	Contiene il nome del campo della tabella (DTK_GETVALUE)
_DKVALCAMPO\$()	Valore del campo (DTK_GETVALUE)
_DKRECORDID\$	Ritorna l'ultimo recordId che è stato inserito o variato
_DKVLCMPOLD\$()	Ritorna il valore precedente alla modifica effettuata (DTK_GETVALUE)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_DV - Documenti Docuvision

_DV

Variabili di struttura archivi Documenti di Docuvision (GETDVDOC – PUTDVDOC – DELDVDOC)	
DOCUMENTO ELETTRONICO	
_DVTIPELETT	Tipo documento elettronico (TDnn, esempio: TD01). Vedere anche _MMTIPELETT\$
DATI DOCUMENTO	
_DVDID	Numero documento (0 = AUTO); dopo la registrazione AUTO contiene il numero attribuito (da leggere); specificare la _DVDID equivale ad aggiornare il documento (se _DVDID non esiste rilascia errore) altrimenti, se non è specificata equivale a inserire un documento nuovo
_DVDCLASS	Classe documento

_DVDTIT\$	Titolo documento
_DVDDATA\$	Data documento
_DVDPROTO\$	Numero protocollo
_DVDDATPRO\$	Data protocollo
_DVDCNT\$	Cliente/Fornitore
_DVDCOGN\$	Ragione sociale o Cognome
_DVDNOME\$	Nome
_DVDPIVA\$	Partita Iva
_DVDCFIS\$	Codice Fiscale
_DVDNOTA\$	Nota
_DVDAZIE\$	Sigla azienda; solo GET
_DVDALLCL\$	Classi a cui il documento è allegato; solo GET
_DVDUFFPA\$	Codice ufficio PA
_DVDSTDOC\$	Stato trasmissione documento in conservazione (accetta: I = Inviato, A = Accettato, R = Rifiutato, C = Conservato con entaksi, F = forzato ad andare in conservazione, D= pacchetto rifiutato per firma non valida o non apposta, N = forza che la fattura NON vada in conservazione, = blank da gestire.)
_DVDIDPACCH	Id invio pacchetto
_DVDTPCONTR\$	Tipo contratto
_DVDDTCOMIN\$	Data competenza contratto
_DVDTPCCLASS\$	Tipo documento della classe
_DVNUMVER	Numero totale revisioni documenti docuvision (solo GET, in scrittura è calcolata); (=1 con gestione pagine attiva);
_DVDPAGXVER()	Numero di pagine per revisione (solo GET); (=1 con gestione pagine non attiva);
DATI REVISIONE	
_DVVNUM()	Numero revisione. Con la gestione delle pagine attiva è inutile impostarla (l'elaborazione è sempre sull'ultima revisione); se attiva la gestione delle revisioni, specificandola equivale ad aggiornare una revisione del documento
_DVVESTFIL\$()	Estensione file. Array in sola lettura con le estensioni dei file di presa visione allegati al documento. L'indice dell'array è il numero di versione del documento.
_DVVNPAG()	Numero di pagina del documento [nota per vecchie installazioni: se è attiva la gestione delle pagine, l'indicazione in scrittura sostituisce l'allegato (alla pagina indicata e già allegata al documento) con il file indicato in _DVVFILEPUT\$; specificare _DVVNPAG=0 equivale ad aggiungere una pagina, altrimenti non scrive il documento e genera errore]
_DVVDATA\$()	Data revisione. Solo GET (e per l'autore se gestite le chiavi accesso); in scrittura sono ignorate e impostate automaticamente (aggiornate sul database in caso di sostituzione di tutte le pagine allegate a documento); con la modifica parziale delle pagine allegate, viene verificata la compatibilità con quelle già esistenti (la tipologia di file), altrimenti rilascia errore e non avviene la scrittura. Dalla versione 2014H la variabile _DVVAUTOR\$() non è più gestita ed è sostituita dalla variabile _DVVIDAUT.
_DVVAUTOR\$()	Autore (considerare la stessa nota di _DVVDATA\$)
_DVVNOTA1\$()	Campo nota 1. Richiesta in scrittura, ma non gestite con la gestione pagine attiva; [nota per vecchie installazioni: richiesta in scrittura, ma non gestite con la gestione pagine attiva]
_DVVNOTA2\$()	Campo nota 2. Riporta anche il nomefile della Fattura, equivalente alla colonna CDS_NOTA2 della tabella DV_VERSIONE contenuta nel file "dati\datiaz\sig\sig_dv.mdb". Richiesta in scrittura, ma non gestite con la gestione pagine attiva; [nota per vecchie installazioni: richiesta in scrittura, ma non gestite con la gestione pagine attiva]
_DVVVOL()	Numero volume; solo GET
_DVVFILEGET\$()	Nome file; solo GET
_DVVFILEPUT\$()	Nome file; solo PUT. Non è allegabile un nome file nuovo a una revisione già esistente (consentito invece inserendo una nuova revisione); se non è attiva la gestione delle revisioni, allegare un file senza specificare _DVVNUM() equivale a sostituire l'allegato all'ultima revisione presente nel database. (*)
_DVVIDAUT()	ID autore revisione ID=utente passepapout
_DVVFIRMA\$()	Firma revisione
_DVVMARCA\$()	Marca revisione
_DVVDTRAS\$()	Data trasmissione/ricezione
_DVVTPRIC\$()	Tipo ricevuta. Riporta lo stato equivalente a DS_TIPO_RICEVUTA (relativo al NKY_NUM_RELEASE più elevato) della tabella DV_VERSIONE contenuta nel file "dati\datiaz\sig\sig_dv.mdb".
_DVVOTRAS\$()	Ora data trasmissione nella forma hhmmss accetta AUTO
_DVVORA\$()	Ora data revisione nella forma hhmmss accetta AUTO
_DVVORIGINE\$	Tipo di generazione del file. Valori riportati: "S" (Scanner), "F" (acquisizione file o clipboard), "P" (Stampa); (attenzione: variabile di sola lettura, se modificata viene azzerata);
_DVVRELEASE\$()	Versione creazione/modifica di file gestito a documento oppure a pagine; le stampe su stampante Docuvision comprendono le stampe di fatture elettroniche, parcelle e le archiviazioni dei Redditi); integrative fatture elettroniche (TD16, ecc.) da primanota. Se si acquisisce come copia di un documento DV pre-esistente, la Release resta impostata = documento originale. (attenzione: variabile di sola lettura, se modificata viene azzerata);
_DVVPRODOTT\$	Prodotto; (attenzione: variabile di sola lettura, se modificata viene azzerata);
DATI CAMPI AGGIUNTIVI	
_DVAVAL\$()	Valori campi
_DVANOMECOL\$()	Nome colonna; per la PUT di un documento impostare il nome del campo dichiarato nella tabella dati aggiuntivi. In revisione può anche essere omissso.
_DVANCAMPI	Numero di campi
_DVVIDEST	Id documento docuvision AutoOn. Solo per documenti residenti su database di installazione 'AutoOn'. Per le operazioni GETDVDOC e PUTDVDOC (PUT può anche inserire documenti esterni e vale solo in installazione commercialista). La variabile contiene l'Id documento del documento docuvision, per trasferire la primanota con documenti allegati in docuvision, senza che siano trasmessi gli allegati, ma solo un collegamento ad essi e che quindi sono richiamabili pur risiedendo solo sull'installazione AutoOn. È un collegamento tra un elemento di docuvision del commercialista e il documento reale dell'azienda, per i soli documenti indicati come da inviare. Esempio: l'azienda AutoOn spedisce al commercialista una registrazione di primanota a cui è allegato il documento con Id 25 sul database locale alla AutoOn. Quando il commercialista carica la registrazione, crea un documento fittizio con un suo Id univoco sul database del commercialista (ad esempio 47) il cui record contiene il valore 25 nel campo leggibile attraverso la _DVVIDEST.

(*)

Nota su ALLEGATI: Sprix gestisce gli allegati relativi al Documento di Docuvision (fig. 1), non gestisce gli allegati relativi alla Fattura PA / B2B (fig. 2); il modo per intervenire sul file XML (della fattura elettronica) è consentito all'evento di stampa ON_STP_XML, ma non sono disponibili apposite variabili per codificare l'allegato dentro il file XML poiché occorre un'apposita codifica separata (fig. 2). Si ricorda che altri generici file XML sono gestiti in Sprix tramite apposite istruzioni XML. Per ottenere l'handle del documento in considerazione al fine di eseguirne le opportune modifiche con le istruzioni generali XML, avvalersi della variabile _FTE_DOCXML che consente l'accesso da collage al file _XML della FTE.

Fig. 1 (Allegati DOCUVISION)

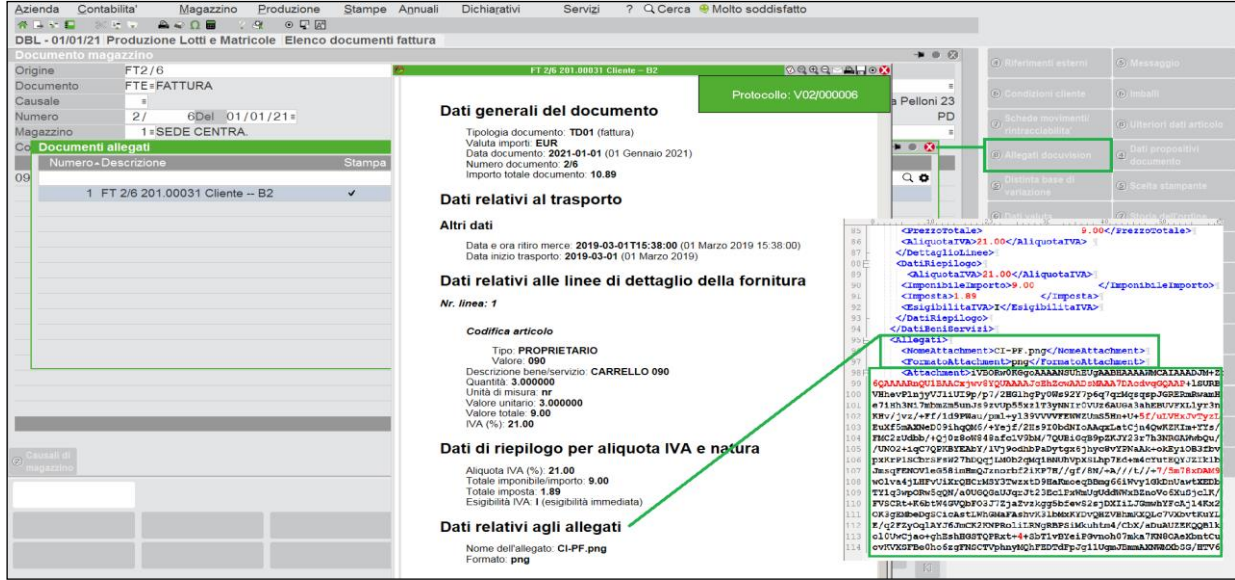
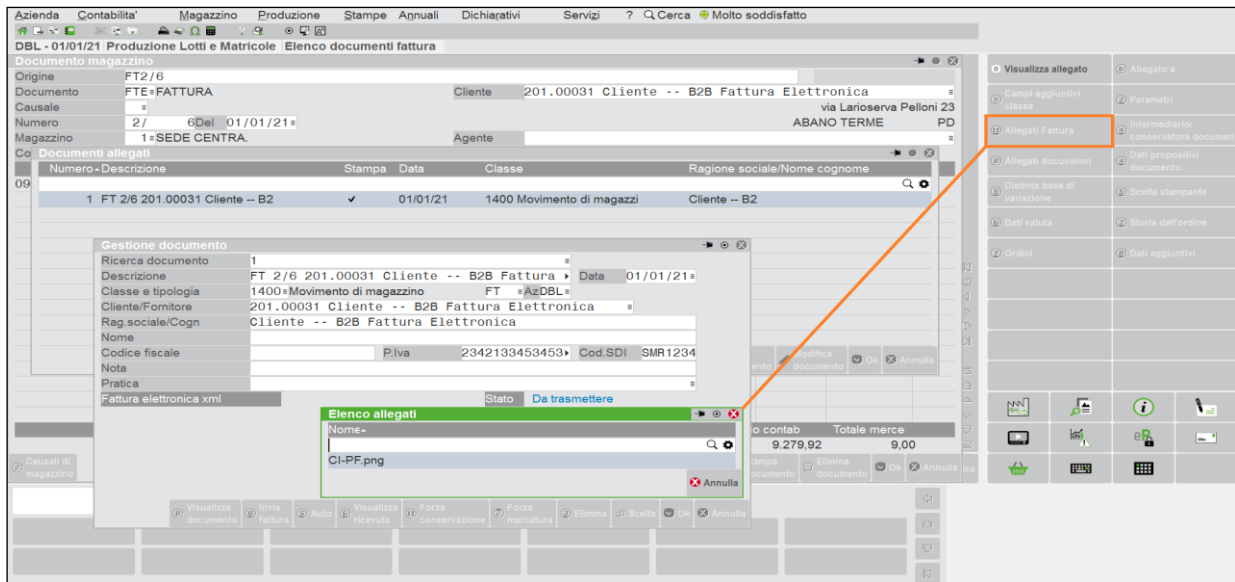


Fig. 2 (Allegati FATTURA)



Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_EMAIL - Stampante Email SMTP

_EMAIL

Variabili di struttura delle istruzioni Collage per il dispositivo stampante Email SMTP (EMAILGETATT - EMAILPUTATT)	
_EMAILMITT\$	Mittente (solo per email SMTP);
_EMAILNOMIT\$	Nome mittente (solo per email SMTP);
_EMAILRIA\$	Rispondi a (solo per email SMTP);

_EMAILNORIA\$	Nome rispondi a (solo per email SMTP);
_EMAILDEST\$	Destinatario;
_EMAILCC\$	Cc;
_EMAILCCN\$	Ccn;
_EMAILOGG\$	Oggetto;
_EMAILTESTO\$	Testo;
_EMAILFIRMA\$	Firma;
_EMAILFHTML\$	File html testo (solo per email SMTP);
_EMAILNOALL\$	Nome file allegato (per il file contenente il risultato della stampa) e allegato alla Email;
_EMAILFALL\$	Array di nomi file allegati;
_EMAILMICCN\$	Mittente in Ccn (Vale "S" o "N") (solo per email SMTP);
_EMAILCOLET\$	Conferma lettura (Vale "S" o "N");
_EMAILPRIOR\$	Priorità SMTP (Vale "A", "B" o "N") (solo per email SMTP);
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRAB

_ERRAB

_ERRAB\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) su Codici Abbinati (struttura _AB)

0	Esito positivo dell'operazione (nessun messaggio, _ERRAB\$ è vuota);
se <>0	Esito negativo dell'operazione (l'errore è descritto in _ERRAB\$);
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRAD

_ERRAD

_ERRAD\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) di Associazione Docuvison (struttura _AD)

0	Esito positivo dell'operazione (nessun messaggio, _ERRAD\$ è vuota);
1	Errore generico con relativa descrizione;
2	Funzione non eseguibile;
3	Associazioni non trovate;
4	Entità in uso da altri terminali;
7	Parametri errati;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRADCOM

_ERRADCOM

_ERRADCOM\$

Variabili che ritornano il codice e la descrizione dell'istruzione ADCOMMERCIALE

0	Esito positivo dell'operazione, nessun messaggio, il campo è vuoto;
1	Parametro 'Modalità aggiornamento' errato o assente. Valori accettati 'C'- completa o 'V'- solo variati;
2	Parametro 'Movimenti di magazzino' errato o assente. Valori accettati 'S' o 'N';
3	Azienda non apribile in uno o più anni indicati come parametro. Export non effettuato;
4	Parametro 'Calcolo margine' errato o assente. Valori accettati 'M'- movimento o 'A'- anagrafica articoli;
5	Parametro 'Ordine cliente' errato o assente. Valori accettati 'S' o 'N';
6	Parametro 'Ordine fornitore' errato o assente. Valori accettati 'S' o 'N';
7	Parametro 'Matrici' errato o assente. Valori accettati 'S' o 'N';
8	Trasferimento o analisi dati in uso dall'utente ???;
9	Modulo ADCOMMERCIALE non disponibile in questa installazione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRAE

_ERRAE

_ERRAE\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT) su Anagrafica Aree (struttura _AE)

0	Esito positivo dell'operazione (nessun messaggio, _ERRAE\$ è vuota);
1	Errore specificato da _ERRAE\$ (PUTAREA);
2	Area non trovata (GETAREA);
3	Parametro area non valorizzato (GETAREA);
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRAGP

_ERRAGP

_ERRAGP\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) su Agenda di Lavoro (struttura _AGP)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRAGP\$ è vuota);
1	Errore specificato a _ERRAGP\$;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRAL

_ERRAL

_ERRAL\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) su Codici Alias (struttura _AL)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRAL\$ è vuota);
se <>0	Errore descritto in _ERRAL\$;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRAPF

_ERRAPF

_ERRAPF\$

Variabili che ritornano il codice e la descrizione dell'operazione AVANZAPF (struttura _APF)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRAPF\$ è vuota);
se <>0	Errore descritto in _ERRAPF\$;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRAR

_ERRAR

_ERRAR\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) su Anagrafica Articoli (struttura _AR)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRAR\$ è vuota);
1	Errore generico, descritto in _ERRAR\$ (consultarla);
2	Codice articolo errato;
3	Numero magazzino errato;
4	Codice articolo non trovato;
5	Articolo non modificabile (archivio rovinato);
6	Impossibile inserire ulteriori articoli;
7	Articolo non cancellabile (archivio rovinato);
8	Articolo non cancellabile (tipo non ammesso o con progressivi);
9	Articolo non cancellabile (non esistente);
10	Articolo in uso dall'utente...; ingnorata in GETAR
11	Articolo: "arcod" ... <ul style="list-style-type: none"> - non è possibile togliere la struttura; - non è possibile inserire la struttura; - non può essere trasformato in 'A CORPO'; - non può essere trasformato in non 'A CORPO'; - non può essere trasformato in a taglie; - deve essere obbligatoriamente una taglia; - non può essere trasformato in a peso netto; - deve essere obbligatoriamente a peso netto;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRARVAR

_ERRARVAR

_ERRARVAR\$

Variabili che ritornano il codice e la descrizione dell'operazione GETARVAR su Anagrafica Articoli (struttura _AR)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRARVAR\$ è vuota);
1	Errore generico, descritto in _ERRARVAR\$ (consultarla);
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

ERRASCMIR

ERRASCMIR

Variabile che ritorna il codice dell'istruzione ASCMIRROR	
0	Esito positivo dell'operazione (nessun messaggio);
5	Tabella non gestita;
6	Tabella non ancora implementata;
7	Archivio sprix non trovato;
8	Tipo dato DB diverso da tipo dato sprix;
9	Campo sprix non trovato;
10	Dimensione del campo sprix superiore alla dimensione del campo DB;
11	Errore interno sul file di risorsa;
12	Directory di destinazione del file di output non esistente;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

ERRAU

ERRAU

ERRAU\$

Variabili che ritornano il codice e la descrizione dell'operazione GETAUTORIZUTE relativa all'Autorizzazione Utenti	
0	Esito positivo dell'operazione (nessun messaggio, _ERRAU\$ è vuota);
1	Errore specificato in _ERRAU\$;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

ERRAZIE

ERRAZIE

ERRAZIE\$

Variabili che ritornano il codice e la descrizione dell'istruzione APRIAZIE o APRIAZIE_EXT	
0	Operazione eseguita con successo nessun messaggio, _ERRAZIE\$ è vuota;
1	Alcuni file dell'azienda sono mancanti;
2	L'azienda è di livello superiore al dispositivo di protezione;
3	La sigla e/o la data non sono validi;
4	L'azienda non esiste;
5	La data non è gestita: non esistono archivi in linea per questa data;
6	Gli anni aziendali sono errati: il file AZIE non è valido;
7	L'accesso all'azienda non è consentito;
8	Troppe aziende installate;
9	Azienda non gestibile;
30	Azienda non in linea (rediretta, cioè fuori dal percorso dataiaz
99	Azienda non apribile (errore generico);
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

ERRBASE64

ERRBASE64

ERRBASE64\$

Variabili che ritornano il codice e la descrizione dell'operazione CODIF_BASE64	
0	Esito positivo dell'operazione (nessun messaggio, _ERRBASE64\$ è vuota);
<>0	Errore con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

ERRBL

ERRBL

ERRBL\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) in Bolla di Lavorazione (struttura _OB e _OL)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRBL\$ è vuota);
1	Errore generico con relativa descrizione; <ul style="list-style-type: none"> - riga OC non valida; - riga già collegata a XX xx/xxxxxx; - l'articolo ???? è senza DBP; - codici incompatibili: '????' e '????';

	<ul style="list-style-type: none"> - incompatibilità tipo articolo/quantità; - documento in uso da altro terminale riga non modificabile; - riga già variata da altro terminale; - riga con distinta base automatica riga non modificabile; - riga bloccata riga non modificabile; - riga ordine cliente annullata riga non modificabile; - errore generico riga non modificabile;
2	Numero bolla errato;
3	Memoria insufficiente;
4	Bolla padre non trovata;
5	Bolla non trovata;
6	Errore generico;
7	Numero BL padre diverso da quello effettivo;
8	Non è possibile inserire una sottobolla;
9	Bolla Padre non esistente;
10	Bolla non totalmente cancellata; alcune righe OC non scollegabili;
11	Documento già in uso dal terminale ??;
12	Art.???? documento OC collegato già in uso;
13	Art.???? collegamento con riga OC non valido;
14	Art.???? Non è possibile collegare righe OC;
15	Art.???? Riferimento OC non ammesso, su produzione per il magazzino;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ERRCDG

ERRCDG

ERRCDG\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) in Controllo di Gestione (struttura _CDG)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRCDG\$ è vuota);
1	Parametri istruzione errati;
2	Anno errato;
3	Errore elaborazione istruzione, documento non trovato;
4	Errore in GETCDG: codifica errata parametro chiave KeyDoc;
5	Errore in PUTCDG: documento in uso da un altro terminale;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ERRCHSTAMP

ERRCHSTAMP

ERRCHSTAMP\$

Variabili che ritornano il codice e la descrizione dell'operazione CHIUDISTAMP_EXT	
-1	Errore preliminare di configurazione tra l'ingresso nel caso Email e l'avvio del rispettivo motore di stampa, SMTP o MAPI. Esempio: invio senza testo o allegati configurati (non valido per stampanti di tipo "Testo"; in questo caso otteniamo un codice di errore 0 e la mail viene comunque inviata)
0	Esito positivo dell'operazione di invio email (nessun messaggio, _ERRCHSTAMP\$ è vuota)
1	Errore ritornato dal server SMTP (es.: utente non valido, password sbagliata, autenticazione non supportata, ecc...)
2	Errore dallo strato socket o TLS nell'invio SMTP (es.: errori di connettività come errore indirizzo server, porta sbagliata, firewall che bloccano la connessione, ecc...)
3	Errore di "altra" natura, specialmente gli abort utente sui warning TLS e gli allegati utente inaccessibili nell'invio SMTP (solo per client iDesk poichè mxrsbatch/spxremoto ignorano i primi ed emettono syserr sui secondi, ovvero syserr nel caso SMTP di file Certification Authority pubbliche sul client con spxremoto/mxrsbatch, mentre l'allegato utente su client e terminale mxrsbatch/spxremoto ora genera una syslog.
4	errore invio MAPI
6	errore autenticazione OAuth2

L'errore si tronca a 255 caratteri, ma l'originale resta conservato nel registro invio (a meno che non sia un errore preliminare o MAPI). Non emettendo più vimsg per i Desk, viene meno la possibilità del consenti riprova. Per averla in sprix, si dovrà continuare a invocare la consueta CHIUDISTAMP. Le variabili hanno i seguenti valori _ERRCHSTAMP=0 e _ERRCHSTAMP\$="" in caso di utilizzo di CHIUDISTAMP ed in caso di CHIUDISTAMP_EXT a seguito dell'uso di altri dispositivi di stampa non e-mail aperti tramite APRISTAMP.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ERRCO

ERRCO

ERRCO\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT) in Anagrafica Commessa (struttura _CO)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRCO\$ è vuota);
1	Errore specificato in _ERRCO\$ (PUTCOMM);
2	Commessa non trovata (GETCOMM);
3	Parametro commessa non valorizzato (GETCOMM);

_ERRCOD

_ERRCOD

_ERRCODPICO\$	Variabile stringa con l'eventuale messaggio di errore rilasciato da GETPC2 se la variabile _CODPICONR=0; (struttura PC)
_ERRCODART\$	Variabile stringa con l'eventuale messaggio di errore (se <>"") rilasciato da GETCODART. es. (dalla v711) (struttura AR)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento.

_ERRCOLOCBL

_ERRCOLOCBL

_ERRCOLOCBL\$

Variabili che ritornano il codice e la descrizione dell'operazione di collegamento ordine a bolla di lavorazione (COLOCBL)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRCOLOCBL\$ è vuota);
1	Generico con relativa descrizione; <ul style="list-style-type: none"> • riga OC non valida; • riga già collegata a XX xx/xxxxxx; • l'articolo ???? è senza DBP; • codici incompatibili: '????' e '????'; • incompatibilità tipo articolo/quantità; • documento in uso da altro terminale riga non modificabile; • riga già variata da altro terminale; • riga con distinta base automatica riga non modificabile; • riga bloccata riga non modificabile; • riga ordine cliente annullata riga non modificabile; • errore generico riga non modificabile;
2	Numero bolla errato;
3	Memoria insufficiente;
4	Bolla padre non trovata;
5	Bolla non trovata;
6	Errore generico;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRCM

_ERRCM

_ERRCM\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT) in Calendario dell'Agenda Attività Utente (struttura CM)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRCM\$ è vuota);
<>0	Errore specificato in _ERRCM\$;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRCOM

_ERRCOM\$

Variabile che ritorna la descrizione dell'operazione di gestione della porta seriale (COM)	
	"" (messaggio vuoto, assenza di errore);
	"Errore di apertura porta" (errore open porta, consigliato il reset macchina);
	"Errore lettura settaggio porta" (errore di porta "sporca", tipico da precedente chiusura anomala, consigliato il reset macchina);
	"Errore settaggio porta" (errore nella fase open indipendente dai parametri, consigliato il reset macchina);
	"EscapeCommFunction" (errore di inizializzazione della seriale in fase di apertura, consigliato reset macchina);
	"Errore lettura parametri dalla porta" (errore lettura dei parametri preimpostati sulla seriale, consigliato il reset macchina);
	"Byte size errato: valori possibili: 5,6,7,8" (è stata specificata una dimensione della parola errata, passare esclusivamente uno dei valori indicati);
	"Parità sbagliata: valori possibili: Nessuna, Pari, Dispari" (è stato indicato un valore di parità diverso da quelli ammessi: 0,1,2);
	"Baud Rate errato: valori possibili 110,300,600,1200,2400,4800,9600,19200,38400" (è stato indicato un valore di Baud Rate errato, passare solo uno dei valori indicati);
	"Errore settaggio parametri della porta" (errore dovuto alle caratteristiche dell'hardware: i parametri passati non sono compatibili con il tipo di porta seriale);
	"Errore chiusura porta" (errore che si può ricevere su una successiva fase di lettura e che potrebbe dipendere anche dall'utilizzo della seriale da parte di altre applicazioni);

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRCSV**_ERRCSV****_ERRCSV\$**

Variabili che ritornano il codice e la descrizione delle funzioni di lettura di file formato CSV (OPENCSV, CSVITERSTRT, CSVITERNXT, CSVROWVAL\$, CLOSECSV)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRCSV è vuota);
1	<ul style="list-style-type: none"> • OPENCSV() - i parametri non sono stati settati correttamente. I parametri sono mancanti o con valore non valido (Es: separatore lungo più di un carattere) • CSVITERSTRT() - I parametri non sono stati settati correttamente. Il parametro <HND> non è valido. • CSVITERNXT() - i parametri non sono stati settati correttamente. Il parametro <HND> non è valido. • CSVROWVAL() - i parametri non sono stati settati correttamente. Il parametro <HND> non è valido. • CLOSECSV() - i parametri non sono stati settati correttamente. Il parametro <HND> non è valido.
3	OPENCSV() - Non è possibile aprire un ulteriore file CSV, occorre chiudere uno di quelli aperti. Tentativo avanzare lo stato di un iteratore non avviato
4	OPENCSV() - L'apertura del file CSV non è riuscita
6	<ul style="list-style-type: none"> • CSVITERSTRT() - Occorre Aprire il file CSV prima di poter avviare un iteratore. Tentativo avanzare lo stato di un iteratore non avviato • CSVITERNXT() - Occorre avviare l'iteratore prima di poterlo avanzare sull'elemento successivo. Tentativo avanzare lo stato di un iteratore non avviato • CSVROWVAL() - Occorre avviare l'iteratore prima di poter acquisire il valore di una colonna. Tentativo di leggere il valore di una colonna ma il file CSV non è aperto • CLOSECSV() - Occorre almeno aprire il file CSV prima di poterlo chiudere. Tentativo di chiudere un file CSV non aperto
7	<ul style="list-style-type: none"> • CSVITERNXT() - E' stata raggiunta la fine del file. L'ultima riga del file CSV è già stata letta
8	OPENCSV() - Non è possibile aprire un ulteriori file, occorre chiudere uno di quelli aperti. Troppi file SPRIX sono stati aperti, occorre chiuderne almeno uno
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui + esempio in comandi per argomento	

_ERRCT**_ERRCT****_ERRCT\$**

Variabili che ritornano il codice e la descrizione dell'istruzione CREATELIST (struttura _CT)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRCT\$ è vuota);
<>0	Errore specificato in _ERRCT\$;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRDA**_ERRDA****_ERRDA\$**

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT) in Indirizzi di Spedizione (struttura _DA)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRDA\$ è vuota);
1	Dati immessi in modo errato (PUTIND);
2	Codice non trovato (GETIND);
3	Ultima anagrafica già utilizzata (PUTIND);
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRDB**_ERRDB****_ERRDB\$**

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) in Distinta Base Primaria (struttura _TP)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRDB\$ è vuota);
1	Errore generico descritto in _ERRDB\$;
2	Distinta base non trovata;
3	Codice distinta base non valido;
4	Numero fase non valido, il numero fase che si vuole leggere/scrivere è > 99;
5	Numero fase non valido;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRDBMIRR**_ERRDBMIRROR**

_ERRDBMIRR\$

Variabili che ritornano il codice e la descrizione delle ASCMIRROR_EXT, ASCMIRROR_EXT2, DBMIRROR, DBMIRROR_EXT	
-1	Elaborazione conclusa correttamente, ma con presenza di uno o più errori non bloccanti (warning); per definire la causa dell'errore occorrono precisi riferimenti in merito;
0	Esito positivo dell'operazione (nessun messaggio, _ERRDBMIRR\$ è vuota);
1	Errore durante la connessione al database, verificare il profilo DSN;
2	Database non supportato;
3	Errore ODBC sulla funzione GetSQLInfo; per DBMIRROR_EXT, non sono disponibili le informazioni inerenti lo RDBMS;
4	Errore generico su ODBC;
5	Tabella non gestita; per DBMIRROR_EXT, tabella <referimento> non trovata nel file risorse;
6	Tabella non ancora implementata; per DBMIRROR_EXT, tabella <referimento> non gestita dalla DBMIRROR;
7	Archivio sprix non trovato; per DBMIRROR_EXT, file <referimento> non disponibile;
8	Tipo dato DB <nome colonna RDBMS> differente da tipo dato sprix <voce dizionario>;
9	Campo sprix <voce dizionario> non trovato; verificare l'integrità del file risorse;
10	Dimensione del campo DB <nome colonna> inferiore alla dimensione del campo sprix <voce dizionario>;
11	Errore sul file di risorsa "dbmirror.dat"; per DBMIRROR_EXT, non trovato<nome file risorse>;
12	Directory <nome directory> di export file ASCII non trovata;
13	Elaborazione interrotta dall'utente;
14	File di personalizzazione <nome file> non trovato;
15	Colonna <nome colonna> indicata nel file di personalizzazione non valida;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRDC
_ERRDC
_ERRDC\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) in Anagrafica Contatti (struttura _DC)	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRDC\$ è vuota);
<> 0	Errore, con relativa descrizione;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRDIR
_ERRDIR\$

Variabile valorizzata dall'istruzione LEGGIDIR e che ritorna l'eventuale errore sul percorso della directory. Verificare le modalità di accesso _ACCFILEUTE.
--

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRDV
_ERRDV
_ERRDV\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) nei Documenti Docuvision (struttura _DV)	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRDV\$ è vuota);
1	Errore generico con relativa descrizione;
2	Funzione non eseguibile;
3	Documento non trovato;
7	Parametri errati;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRFILE
_ERRFILE\$

Variabile che ritorna l'errore dell'accesso non consentito nella gestione di file di tipo utente _ACCFILEUTE = 1; per il controllo di errori sul percorso della directory utilizzare l'istruzione LEGGIDIR.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRFIRMAF
_ERRFIRMAF\$

Variabile che ritorna l'errore della funzione FIRMAFILE di esecuzione della firma digitale di un file mediante SmartCard. La variabile torna valore vuoto ("") se la firma è stata eseguita correttamente, altrimenti un valore diverso da vuoto se la funzione non ha firmato il file e ne

riporta la descrizione dell'errore

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRGETINT

_ERRGETINT\$

Variabile che ritorna la descrizione dell'eventuale errore dell'istruzione GETINTERNET per la copia di un file .html. In caso di messaggio molto lungo (fino a 512 caratteri che può provenire dalle funzioni socket) il limite resta comunque 255 caratteri, quindi può risultare troncato.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRGETREM

_ERRGETREM\$

Variabile che ritorna l'errore di una qualsiasi GETREM... di Sprix Mobile. Vedere l'istruzione SRVCALL.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRGM

_ERRGM

_ERRGM\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT) nei Gruppi Merceologici (struttura _GM)

0	Esito positivo dell'operazione (nessun messaggio, la _ERRGM\$ è vuota);
1	Codice Gruppo merceologico non trovato (GETGM);
2	Codice Gruppo Merceologico Obbligatorio;
3	Descrizione Gruppo Merceologico Obbligatorio;
4	Codice padre non presente nell'anagrafica;
5	Il codice padre deve essere diverso dal codice gruppo merceologico;
6	Codice selezionato già presente all'interno della struttura di gruppo merceologico dell'anagrafica;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRGPS

_ERRGPS\$

Variabile di errore nell'operazione di recupero dati GPS all'evento Sprix Mobile ON_GETGPSDATA: (vedere WCALL "GETGPSDATA")

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRIA

_ERRIA

_ERRIA\$

Variabili che ritornano il codice e la descrizione dell'operazione ITERARC di SprixMobile (variabili struttura _IA)

0	Esito positivo dell'operazione (nessun messaggio, la _ERRIA\$ è vuota);
<>0	Errore, con relativa descrizione;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRIL

_ERRIL

_ERRIL\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) negli Impegni di Lavorazione (struttura _IL)

0	Esito positivo dell'operazione (nessun messaggio, la _ERRIL\$ è vuota);
1	Errore generico (solo PUTIL);
1	Codice articolo materie prime non trovato nel file articoli (solo GETIL);
2	Impegni non trovati in archivio (solo GETIL);

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRIMAP

_ERRIMAP

_ERRIMAP\$

Variabili che ritornano il codice e la descrizione dell'operazione CHIUDISTAMP_EXT	
0	nessun errore;
1	errore di transazione IMAP (es. autenticazione, mailbox oltre i limiti di spazio);
2	errore di rete (es. connessione fallita o chiusa in corso d'opera);
3	altro (es. uscita su decisione dell'utente in caso di avvisi sicurezza TLS);
<p>Variabili di errore di memorizzazione in posta inviata. Come le variabili _ERRCHSTAMP e _ERRCHSTAMP\$, tali variabili IMAP rimangono rispettivamente 0 e vuota quando non eseguita una memorizzazione messaggio.</p> <p>Assumono invece valori diversi ad ogni errore di memorizzazione, sia per la posta inviata di un'email regolarmente spedita ai destinatari come account SMTP, sia per la posta in arrivo per notificare un fallito invio email SMTP. Sono quattro le combinazioni possibili:</p> <ul style="list-style-type: none"> • se email SMTP spedita con successo e memorizzata nella posta inviata, allora le due stringhe saranno vuote e le due numeriche varranno 0; • se email SMTP spedita con successo ma fallita memorizzazione in posta inviata, allora _ERRIMAP varrà diverso da 0 e _ERRIMAP\$ non sarà vuota; • se fallita spedizione email SMTP e notifica memorizzata con successo nella posta in arrivo, allora _ERRCHSTAMP varrà diverso da 0 e _ERRCHSTAMP\$ non sarà vuota; • se fallita spedizione email SMTP e fallita memorizzazione notifica nella posta in arrivo, allora le due stringhe non saranno vuote e le due numeriche non varranno 0 <p>Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>	

_ERRIND

_ERRIND

Variabile che ritorna il codice in base all'esito positivo (= 0) o negativo (= 1) dell'operazione sul record del FILE ad INDICI	
GETKEY	0 = trovato; 1 = non trovato;
VARKEY	0 = variato; 1 = non variato, la chiave da variare non esiste o esiste già la nuova chiave;
INSKEY	0 = inserito; 1 = non inserito, esiste già;
DELKEY	0 = cancellato; 1 = non cancellato, non esiste;
NEXTKEY	0 = prossimo trovato; 1 = prossimo non trovato;
PREVKEY	0 = precedente trovato; 1 = precedente non trovato;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRJSON

_ERRJSON\$

Variabile che ritorna il messaggio di errore delle operazioni JSON	
""	Valore nullo (nessun messaggio) significa esito positivo dell'operazione;
<>"	Descrizione del messaggio di errore; gli errori sono principalmente generati dalle funzioni di Load / Dump di un file JSON oppure in seguito a parametri non corretti oppure dalla funzione di avanzamento sull'elemento successivo di un iteratore.
0	JSONGETSTRF() - operazione terminata correttamente
1	JSONGETSTRF() - Non e' stato possibile recuperare il valore stringa dell'oggetto JSON
2	JSONGETSTRF() - Non e' stato possibile scrivere il file di output
3	JSONGETSTRF() - L'oggetto JSON di cui e' stato indicato il riferimento non è del tipo corretto (JSON_STRING)
4	JSONGETSTRF() - Non e' stato indicato il file su cui salvare il valore stringa dell'oggetto JSON
5	JSONGETSTRF() - Il riferimento all'oggetto indicato, non contiene e' un riferimento valido
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRLOT

_ERRLOT

_ERRLOT\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) nella Anagrafica Lotti (struttura _LOT)	
0	Esito positivo dell'operazione (nessuna descrizione, la _ERRLOT\$ è vuota);
1	Errore generico di congruità dati; - "Tipo dato personalizzazione errato. Cod.<TpLotto><CodDatoUtente>(lotto)" - "Valore dato personalizzazione errato. Cod.<TpLotto><CodDatoUtente>(lotto)" - "Valore dato personalizzazione fuori limiti. Cod.<TpLotto><CodDatoUtente>(lot)" - "Dato personalizzato contat. Non modificabile. Cod.<TpLotto><CodDatoUtente>(lot)" - "Dato personalizzato File nome errato/inesistente. Cod.<TpLotto><CodDatoUtente>(lot)" - "Dati personalizzati non gestibili (lotto)"
2	Errore dato utente obbligatorio (in fase di scrittura); - "Dato personalizzato obbligatorio Cod. <TpLotto><CodDatoUtente> (lotto)"
3	Errore lettura lotto - "Codice lotto non trovato (lotto)" - "Codice lotto non valido o non trovato lotto ()"
4	Errore lotto in uso da altri terminale "Lotto in uso da utente()(lotto)"
5	Errore cancellaz. Lotto "Codice lotto movimentato non cancellabile(lot)"
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRLST

_ERRLST

_ERRLST\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) nella Lista di Prelievo (struttura _LST)	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRLST\$ è vuota);
1	Errore bloccante di elaborazione

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRMEXALDB

_ERRMEXALDB

_ERRMEXALDB\$

Variabili che ritornano il codice e la descrizione dell'istruzione MEXALDB	
Per ulteriori approfondimenti consultare il capitolo MexalDB del Manuale Utente	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRMEXALDB\$ è vuota);
1	Terminale non bloccabile;
2	Da "STD" alcune aziende non esportate;
3	Azienda non esportata;
4	Parametri errati;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRMOD

_ERRMOD

_ERRMOD\$

Variabili che ritornano il codice e la descrizione delle operazioni Import Moduli (il controllo è alla MODCLOSE)	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRMOD\$ è vuota);
<> 0	Errore bloccante di elaborazione modulo, con relativa descrizione (controllo in fase MODCLOSE);

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRMYDB

_ERRMYDB

_ERRMYDB\$

Variabili che ritornano il codice e la descrizione in base all'esito positivo (= 0) o negativo (= 1) dell'operazione in MYDB	
MYDBGGET	0 = trovato; 1 = non trovato;
MYDBPUT	0 = inserito/variato; 1 = non inserito/variato;
MYDBDEL	0 = cancellato; 1 = non cancellato, non esiste;
MYDBNEXT	0 = prossimo trovato; 1 = prossimo non trovato;
MYDBPREV	0 = precedente trovato; 1 = precedente non trovato;

101	Errore generico _ERRMYDB\$ contiene il dettaglio dell'errore;
102	Archivio MyDB non definito;
103	Numero Key non valida per archivio MyDB;
104	Istruzione ***** non eseguita;
105	ID (###) archivio MyDB in uso da un altro terminale;
106	_MYDBID non impostato per l'archivio MyDB;
107	Numero finestra errato;
108	Dato non modificabile (Ritornato in caso di tentativo di modifica di un record Bloccato);
109	ID (###) archivio MyDB già in uso dal terminale corrente;
110	Archivio MyDB non trovato;
111	Record non inserito (Riorganizzare archivio);
112	Record non variato (Riorganizzare archivio);
113	Record non cancellato (Riorganizzare archivio ...);
114	Archivio MyDB: Tipo archivio mancante o errato (MPCFAN);
115	Archivio MyDB: Riferimenti a documento non presente;
116	Archivio MyDB: Inserimento dati manuale non ammesso;
117	Archivio MyDB: Estensione non trovata;
118	Archivio MyDB: Disabilitato;
119	Archivio MyDB: Data obbligatoria nell'estensione documento per le aziende infrannuali;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRMM

_ERRMM

_ERRMM\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL e Totali) nei Documenti Magazzino (struttura _MM)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRMM\$ è vuota);
1	Errore generico, la descrizione è in _ERRMM\$ (esempio: fuori fido);
2	Errore parametri aziendali (conti automatici, fuori fido, ecc...);
3	Errore caricamento documento (non trovato in fase di lettura, ecc...);
4	Documento in uso da altri terminali (in fase di scrittura documento);
5	Documento da cancellare non trovato;
6	La cancellazione dell'operazione di primanota non è possibile;
7	Parametro non valido;
10	Uguale a 0 ma relativo a beni usati;
11	Uguale a 1 ma relativo a beni usati;
12	Uguale a 2 ma relativo a beni usati;
13	Uguale a 3 ma relativo a beni usati;
20	Errore MxRT con descrizione in _ERRMM\$
Msg. variabile _MMNRECRIGA	DBA con troppi componenti
	DBA con riferimenti incrociati o memoria insufficiente
	La lista dei componenti potrebbe essere incompleta (riga <Numero riga>)
	Uno o più componenti hanno superato la quantità massima (riga <Numero riga>)
Msg. variabile (tipo merce) _MMTIPM(R)	Con tipo merce nr.... è obbligatorio specificare almeno un DDT
	Tipo merce nr.... non valido quando non ci sono ddt in riga (riga nr...)
	Con tipo merce nr.... non è consentito applicare iva (riga nr....)
	Codice di esenzione con Natura diversa da nr.... (riga nr....)
	Sono ammesse le seguenti combinazioni: TM nr. oppure solo TM nr. oppure solo TM nr. (riga nr....)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRNUM

_ERRNUM\$

Variabile che ritorna il messaggio delle operazioni sui numeratori GETNUM e SETNUM.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRP7M

_ERRP7M\$

Variabile che ritorna la descrizione dell'eventuale errore dell'operazione DECODP7M

""	La variabile è vuota se il risultato di DECODP7M = 0
"descrizione"	Descrizione dell'errore se il risultato di DECODP7M <> 0

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRPA

_ERRPA

_ERRPA\$

Variabili che ritornano il codice e la descrizione delle operazioni (GET/PUT/DEL) in Tabella Pagamenti

0	Esito positivo dell'operazione in _ERRPA (nessun messaggio e la _ERRPA\$ è vuota);
<> 0	Errore rilevato in _ERRPA con relativa descrizione in _ERRPA\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRPAM

_ERRPAM

_ERRPAM\$

Variabili che ritornano il codice e la descrizione dell'istruzione PROGARTMAG (Progressivi Articoli)

0	Esito positivo dell'operazione (nessun messaggio, _ERRPAM\$ è vuota);
1	Articolo non trovato;
2	Articolo/i senza movimenti;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRPART**_ERRPART****_ERRPART\$**

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) nelle Particolarità (struttura _PART)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRPART\$ è vuota);
1	Errore generico descritto in _ERRPART\$;
2	Errore tipo particolarità non valido (valori ammessi P/S/V/A);
3	Occorre indicare almeno uno tra la categoria ed il codice cliente;
4	Si deve indicare un solo campo tra la categoria ed il codice cliente;
5	Occorre indicare almeno uno tra la categoria ed il codice articolo;
6	Si deve indicare un solo campo tra la categoria ed il codice cliente;
7	Data inizio validità errata,

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRPC**_ERRPC****_ERRPC\$**

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) nell'Anagrafica Piano dei Conti e Rubrica Clienti Fornitori (struttura _PC)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRPC\$ è vuota);
1	Errore generico, descritto in _ERRPC\$;
2	Codice non trovato nel piano dei conti;
3	La Partita Iva esiste già con un altro codice nello stesso gruppo/mastro;
4	Conto non cancellabile (ultima consegna diversa da zero);
5	Conto non cancellabile (valore ordini/bolle in corso diverso da 0);
6	Conto non cancellabile (progressivi diversi da 0);

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRPN**_ERRPN****_ERRPN\$**

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) nella Primanota (struttura _PN)	
0	Esito positivo dell'operazione (nessun messaggio, _ERRPN\$ è vuota);
1	<p>Errore generico.</p> <p>Nella cancellazione i seguenti messaggi:</p> <ul style="list-style-type: none"> - libro giornale del mese già stampato in maniera definitiva - registro IVA acquisti del mese già stampato in maniera definitiva - registro IVA vendite del mese già stampato in maniera definitiva - registro corrispettivi del mese già stampato in maniera definitiva - liquidazione del mese già stampata in maniera definitiva - registrazione con immobilizzazione già stampata sul registro - registrazione con immobilizzazione in parte o totalmente venduta - registrazione con costo pluriennale già stampato - documento in uso da altro terminale - archivio immobilizzazioni già in uso da un altro terminale - operazione inviata al commercialista - documento senza record scadenziario, documento non cancellabile (la delpn può cancellare registrazioni con clienti/fornitori solo se questi hanno la relativa rata in scadenziario) - nello scadenziario del conto il documento è già in pagamento - il documento può essere cancellato solo entrando nel programma con la sua data di registrazione - la cancellazione comporta variazione di una registrazione con ritenuta d'acconto già versata <p>Nella registrazione può essere rilasciato il messaggio "REGISTRAZIONE TROPPO GRANDE". In tal caso si rimanda alle caratteristiche richieste generali del modulo standard per la registrazione o la modifica di una primanota. Riportiamo la regola generale: una registrazione di primanota può contenere al massimo 200 righe, avendo presente che se contengono delle descrizioni, queste contano doppio, perciò se tutte le righe ne contengono una, il numero massimo si riduce a 125. Il medesimo conteggio deve essere effettuato per ogni riga con aliquota Iva o codice in esenzione Iva e per ogni codice analitico inserito nella registrazione. In ogni caso il numero massimo di conti ammessi è 200. Anche la descrizione nel campo sottostante la causale del documento primanota, se compilata, vale come una riga di primanota; se si tratta di registrazioni intracee, una riga viene occupata anche dai dati intracee.</p>

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRPU

_ERRPU

_ERRPU\$

Variabili che ritornano il codice e la descrizione delle istruzioni relative all'Anagrafica Ubicazioni	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRPU\$ è vuota); L'articolo o il lotto può essere inserita nell'ubicazione (CTRARTTOUBI)
1	L'articolo specificato non è stato trovato (CTRARTTOUBI)
2	L'articolo non può essere inserito nell'ubicazione specificata; vedere _ERRPU\$ per conoscerne il motivo (CTRARTTOUBI)
3	Id ubicazione non valido o non trovato (GETPU, DELPU) Non è stato specificato alcun codice articolo (è obbligatorio) (CTRARTTOUBI)
4	Anagrafica ubicazione in uso dall'utente (PUTPU con _PUID > 0 oppure DELPU)
5	Id ubicazione utilizzata non cancellabile (DELPU)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRPUTREM

_ERRPUTREM\$

Variabile che ritorna l'errore di una qualsiasi PUTREM... di Sprix Mobile. Vedere l'istruzione SRVCALL. Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

_ERRPZD

_ERRPZD

_ERRPZD\$

Variabili che ritornano il codice e la descrizione dell'istruzione GETCONDDOC (Condizioni Documento)	
0	Esito positivo dell'operazione (nessun messaggio, la variabile _ERRPZD\$ è vuota);
1	Sigla documento non valida;
2	Data Obbligatoria;
3	Codice articolo Obbligatorio;
4	Codice articolo errato;
5	"Articolo non revisionabile, utente non autorizzato" o "Codice non disponibile nel piano dei conti";
6	"Codice articolo non trovato" o "Codice conto errato o non trovato nel piano dei conti";
7	Valore listino errato
8	Tipo obbligatorio o errato. Valori accettati 1,2,3;
9	Sconto non valido;
10	Valore Extra non valido accettati A,B,C;
11	Valore sconto Cli/For errato da 0 a ...;
12	Valore provvigione Cli/For errato da 0 a
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRREDSPX

_ERRREDSPX

_ERRREDSPXS\$

Variabili che ritornano il codice e la descrizione dell'istruzione PARAM_REDD (Dichiarazioni dei Redditi)	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRREDSPXS\$ è vuota);
1	Errore "Tipo Coniuge/Dichiarante errato (?) [CD]" (il simbolo ? è il carattere passato all'istruzione)
2	Errore "Modello redditi non esistente (?) [730/RPF/RSP/RSC/RNC]" (il simbolo ? rappresenta i 3 caratteri immessi dopo MOD=)
3	"Errore di sintassi, non presente 'MOD=' (il secondo parametro, se immesso, deve contenere la parola MOD=)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRRT

_ERRRT\$

Variabile che ritorna il messaggio delle operazioni RT? sul Registratore Telematico. Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

_ERRSC

_ERRSC

_ERRSC\$

Variabili che ritornano il codice e la descrizione dell'operazione PUTSC di primanota nel Import Moduli (struttura _SC)	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRSC\$ è vuota);
<> 0	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRSEM**_ERRSEM**

Variabile che ritorna il codice dell'operazione BLOCCASEM e indica lo stato del semaforo	
0	È stato bloccato (anche con sistema composto da un solo terminale);
1	Non è stato bloccato (è già bloccato da un altro terminale);
2	Non è stato bloccato (<nomesem\$> errato);
3	Non è stato bloccato (sono in uso troppi semafori);
4	Non è stato bloccato (sono in uso troppi semafori globali);
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRSN**_ERRSN\$**

Variabile che ritorna l'errore dell'operazione GETSIGN (gestione della firma di presa visione in Sprix Mobile)	
= ""	Esito positivo dell'operazione (nessun messaggio in _ERRSN\$);
<> ""	Descrizione dell'errore; per approfondimenti consultare il Manuale Sprix Mobile;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRSPRIGBL**_ERRSPRIGBL****_ERRSPRIGBL\$**

Variabili che ritornano il codice e la descrizione dell'operazione SPRIGBL di spezza riga Bolla di Lavorazione	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRSPRIGBL\$ è vuota);
1	Errore generico, con relativa descrizione; <ul style="list-style-type: none"> - riga OC non valida; - riga già collegata a XX xx/xxxxxx; - l'articolo '???' è senza DBP; - codici incompatibili: '????' e '????'; - incompatibilità tipo articolo/quantità; - documento in uso da altro terminale riga non modificabile; - riga già variata da altro terminale; - riga con distinta base automatica riga non modificabile; - riga bloccata riga non modificabile; - riga ordine cliente annullata riga non modificabile; - errore generico riga non modificabile; - quantità da avanzare maggiore della qta di riga sul PF; - quantità da avanzare pari a qta di riga sul PF; - quantità da avanzare pari a zero; - impossibile spezzare la riga; - riga non trovata;
2	Numero Bolla errato
3	Memoria insufficiente
4	Bolla padre non trovata
5	Bolla non trovata
6	Errore generico
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERWSPRIGBL**_ERWSPRIGBL\$**

Variabile di sola lettura valorizzata dall'esecuzione di SPRIGBL nei soli terminali privi di interfaccia utente, quali sprix in operazione pianificata e procedura shaker, quando vengono azzerati i lotti presenti negli impegni delle materie prime.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

_ERRSQL_N**_ERRSQL_N**

Variabile che ritorna errori nativi generati dal data base di riferimento nelle istruzioni SQL, diversamente è vuota (nessun errore). Non differenzia tra un errore e un warning (esempio all'atto della connect a SQLServer). La visualizzazione di messaggi prima del test sulla variabile potrebbe duplicare record, si consiglia un'apposita label da richiamare.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRSQL_NDF

_ERRSQL_NDF

Variabile che indica il fine dati nel cursore (termine del recordset). Ha valore 0 (false = righe non terminate) o 1 (true = lettura terminata).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRSQL_M

_ERRSQL_M\$

Variabile che ritorna errori di istruzioni SQL_ e la lista di errori dipende dal tipo di database (presenti se _ERRSQL_M\$<>"") e la sequenza non corretta delle operazioni rilascia errore "function sequence error", diversamente è vuota (nessun errore). Non differenzia tra un errore e un warning (esempio all'atto della connect a SQLServer). La visualizzazione di messaggi prima del test sulla variabile potrebbe duplicare record, si consiglia un'apposita label da richiamare.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRSQL_S

_ERRSQL_S\$

Variabile che ritorna errori interni di ODBC (connessione, ecc.) e rappresenta il valore di SQL State (ulteriori informazioni sul manuale ODBC), diversamente è vuota (nessun errore). Non differenzia tra un errore e un warning (esempio all'atto della connect a SQLServer). La visualizzazione di messaggi prima del test sulla variabile potrebbe duplicare record, si consiglia un'apposita label da richiamare.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRSRV

_ERRSRV

_ERRSRV\$

Variabili che ritornano il codice e la descrizione dell'istruzione SRVCALL

0	Esito positivo dell'operazione (nessun messaggio, _ERRSRV\$ è vuota);
<> 0	Errore di comunicazione con il Server, il contenuto del messaggio di errore è nella _ERRSRV\$
	In caso di errore nell'esecuzione ON_START_SPRIX l'istruzione SRVCALL ha sollevato un'eccezione "chiamata remota già in esecuzione" occorre verificare l'integrità dell'applicazione e relative librerie: disinstallare e reinstallare l'app da mobile sul terminale oggetto dell'errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRSSD

_ERRSSD

_ERRSSD\$

Variabili che ritornano il codice e la descrizione dell'istruzione SRVSYNCD0C

0	Esito positivo dell'operazione (nessun messaggio, _ERRSSD\$ è vuota);
<> 0	Errore di comunicazione con il Server, il contenuto del messaggio di errore è nella _ERRSSD\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRSTORD

_ERRSTORD

_ERRSTORD\$

Variabili che ritornano il codice e la descrizione dell'istruzione STORD

0	Esito positivo dell'operazione (nessun messaggio, _ERRSTORD\$ è vuota);
1	Stampa interrotta (è stato premuto il tasto ESC durante la stampa);
2	Non trovati dati con la selezione specificata (non esistono documenti con i parametri indicati);
3	Numero documento non valido (numero dell'ordine <=0 o > al numero massimo gestito nei numeratori dell'azienda);
4	Numero serie documento non valido (num di serie = 0 o > al num. Massimo di serie gestite per l'ordine nei

	numeratori dell'azienda);
5	Anno non valido (anno di inizio ricerca minore o maggiore degli anni in linea gestiti dall'azienda; Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

_ERRSTPEMAIL

_ERRSTPEMAIL

Variabile valorizzata dalla istruzione APRISTAMP_EMAIL se la stampante non è di tipo Email.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRSVILDB

_ERRSVILDB\$

Variabile che ritorna gli errori generati dall'istruzione SVILDB (sviluppo della Distina Base Primaria); contiene l'errore in caso di _NSVILDB=0

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_ERRTPLOT

_ERRTPLOT

_ERRTPLOT\$

Variabili che ritornano il codice e la descrizione dell'operazione (GET/PUT/DEL) in Anagrafica Tipi Lotto (struttura _TPLOT)	
0	Esito positivo dell'operazione (nessuna descrizione, _ERRTPLOT\$ è vuota);
1	Errore generico di congruità dati: "Codice tipo lotto obbligatorio.()";
2	Errore dato utente non cancellabile (in scrittura): - "Cancellazione dato personalizzato non ammessa. Codice mancante o errato. ()";
3	Errore lettura tipo lotto: - "Codice tipo lotto non valido o non trovato (<cod tipo lotto>"); - "Codice tipo lotto non valido o non trovato ()"; - "Codice tipo lotto non immesso (<cod tipo lotto>"); - "Tipo ordinamento non valido. Valori ammessi 1 2 3 (<cod tipo lotto>");
4	Errore tipo lotto in uso da terminali: "Tipo lotto in uso da utente()";
5	Errore cancellazione tipo lotto: - "Codice tipo lotto non cancellabile. Stato diverso da 'C'()";
Ulteriori controlli:	<ul style="list-style-type: none"> - Controlli sui valori ammessi riportati in tabella; - Codice tipo lotto su dati utente, tutti uguali; - Codice dato utente obbligatorio se impostata la parte del codice tipo lotto; - Se tipo dato utente è: <ul style="list-style-type: none"> o A – controllo valore default con dimensione impostata (da 1 a 80); o N – controllo val.def. con dimensione impostata (da 1 a 10); o I – controllo dim.impost.(da 1 a 9); val.def. non ammesso; ctrl. Singolo dato I; o V – controllo val. default con dim. Impostata (da 1 a 15); o D – controllo val.def. con data valida o dicitura SISTEMA o TERMINALE; o F – controllo val.def. con dim. Impost. (da 1 a 255) e ctrl. Esist. File immesso; - Controllo univocità Ordine tra i dati utente; - Ctrl. Univocità elemento cod. utente (chiave) tra i dati pers.ti e se chiave il dato deve essere obbligatorio;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRUBUNI

_ERRUBUNI

_ERRUBUNI\$

Variabili che ritornano il codice e la descrizione dell'istruzione GETRUBUNICA	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRUBUNI\$ è vuota);
<> 0	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRVIEW

_ERRVIEW

_ERRVIEW\$\$

Variabili che ritornano il codice e la descrizione dell'Istruzione ARCHVIEW	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRVIEW\$ è vuota);
1	Errore specificato in _ERRVIEW\$ (La ricerca non ha fornito risultati);

2	Uscita dalla gestione con il tasto Annulla (nessuna scelta eseguita);
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRWD

_ERRWD

_ERRWD\$

Variabili che ritornano il codice e la descrizione dell'operazione Sprix mobile WCREATEDIALOG	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRWD\$ è vuota);
<> 0	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRWF

_ERRWF

_ERRWF\$

Variabili che ritornano il codice e la descrizione delle operazioni Sprix mobile WCREATEFORM, WSHOWFORM	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRWF\$ è vuota);
<> 0	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRWI

_ERRWI

_ERRWI\$

Variabili che ritornano il codice e la descrizione dell'operazione Sprix mobile WCREATEINPUT	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRWI\$ è vuota);
<> 0	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRWP

_ERRWP

_ERRWP\$

Variabili che ritornano il codice e la descrizione dell'operazione Sprix mobile WPRTFORMOPT	
0	Esito positivo dell'operazione (nessun messaggio, la _ERRWP\$ è vuota);
<> 0	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRXML

_ERRXML\$

Variabile che ritorna la descrizione delle Istruzioni e Funzioni di gestione Xml	
= ""	Esito positivo dell'operazione (nessun messaggio, la _ERRXML\$ è vuota);
<> ""	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRZA

_ERRZA

_ERRZA\$

Variabile che ritorna il codice e la descrizione delle operazioni GET/PUT/DEL sull'anagrafica unica	
= 0	Esito positivo dell'operazione (nessun messaggio, la _ERRZA\$ è vuota);
<> 0	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ERRZIP

_ERRZIP\$

Variabile che ritorna la descrizione dei comandi di gestione file ZIP	
= ""	Esito positivo dell'operazione (nessun messaggio, la _ERRZIP\$ è vuota);
<> ""	Errore, con relativa descrizione;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui.	

_ESPOSIZERR

_ESPOSIZERR\$

Variabile che ritorna la descrizione della funzione GETESPOS di lettura esposizione alla data	
= ""	Esito positivo dell'operazione (nessun messaggio, la _ESPOSIZERR\$ è vuota);
<> ""	Errore, con relativa descrizione; -“codice di tipo diverso da C/F” -“codice non trovato in pico” (piano dei conti) -“codice non trovato in rudt” (rubrica) -“data per calcolo esposizione non gestita/errata”
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_FTE - del evento ON_STP_XML

_FTE

Variabili di utilizzo per l'evento ON_STP_XML	
_FTE_DOCXML	Handle del documento in considerazione. Non essendo possibile l'intervento sullo stesso Tag che riporta più informazioni (esempio: lotti spezzati sulla stessa riga articolo), occorre gestire l' intero documento con le istruzioni generali XML. La variabile _FTE_DOCXML consente l'accesso da collage al file _XML della FTE in quel momento in considerazione al fine di eseguirne le opportune modifiche con le istruzioni generali XML, es.
_FTENODO\$(R,100)	Codice nodo della fattura elettronica espresso nella forma "2.1.1.1", elemento bidimensionale il secondo valore massimo 100 elementi è utilizzato per racchiudere tutti gli elementi del nodo.
_FTEVALNODO\$(R,100)	Valore che deve essere inserito/modificato nel nodo specificato; elemento bidimensionale il secondo valore massimo 100 elementi è utilizzato per racchiudere tutti gli elementi del nodo.
_FTEOPNODO\$(R,100)	Imposta l'operazione per il nodo indicato. Accetta i valori: <ul style="list-style-type: none"> • "I=inserisci", se il nodo esiste non avverrà alcuna operazione, se non esiste sarà inserito; • "V=varia", se il nodo esiste sarà variato, se non esiste non avverrà alcuna operazione; • "U=Inserisci/Varia", se il nodo esiste sarà variato, se non esiste sarà inserito; • "C=Cancella", se il nodo esiste sarà cancellato, se non esiste non avverrà alcuna operazione vedi https://easypass.passepartout.sm/Soluzione/67608 • " " = default (blank) <ul style="list-style-type: none"> ○ se il nodo è presente nel file viene: <ul style="list-style-type: none"> ▪ se sono ammessi campi multipli, vengono accodati; ▪ se non sono ammessi campi multipli, viene sovrascritto con l'ultimo elaborato; ○ se il nodo non è presente nel file, viene aggiunto: <ul style="list-style-type: none"> ▪ se ripetuto più volte e sono ammessi campi multipli vengono accodati (se il nodo esiste ed è multiplo, sarà inserito); ▪ se ripetuto più volte e non sono ammessi campi multipli, viene sovrascritto con l'ultimo elaborato (se il nodo esiste ed è univoco, sarà sostituito)
_FTERIGMM(R)	Per i nodi legati alle righe dei movimenti di magazzino specifica il numero della riga del movimento di magazzino, per riga si intendono le righe articolo del documento.
_FTENUMELE	Indica quanti elementi sono stati caricati nelle variabili. Se tale valore è zero non verrà processata nessuna variabile. Per numero di elemento si identifica quanti NODI padre devono essere inseriti.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_FTP - del comando FTPSECCMD

_FTP

Variabili struttura FTP utilizzate da FTPSECCMD per avviare la comunicazione ed eseguire i comandi richiesti su srv FTP/FTPS	
_FTPCMD\$	Comando, valori ammessi ("PUT" "GET" "DIR" "DEL")
_FTPURL\$	URL della cartella remota <protocollo>://<Host>[:<porta>]/<percorso>[/<percorso>[...]] <protocollo> può assumere i valori "ftp", "ftps" oppure "sftp"; Vedere anche _FTPREMOTEFILE\$
_FTPFRXST\$	Se presente un proxy è il Nome Host del proxy
_FTPFRXPORT	Se presente un proxy è la Porta del Proxy
_FTPFRXUSPW\$	Se presente un proxy sono le Credenziali di accesso al proxy (Utente:password)
_FTPFRXTUNN	Se presente un proxy è il Tunneling On/Off
_FTPUSER\$	Utente per l'autenticazione sul server FTP
_FTPPASS\$	Password per l'autenticazione sul server FTP
_FTPHEADERS\$()	Array di Header (opzionali) da aggiungere alla comunicazione

_FTPSSLTLS\$	(SOLO protocollo FTP/FTPS) Modalità di utilizzo di SSL/TLS nella sessione FTP <ul style="list-style-type: none"> • "NONE" = SSL/TLS non usato • "TRY" = Tenta uso SSL/TLS, se fallisce continua comunque • "CONTROLCH" = SSL/TLS SOLO per ControlChannel • "ALL" = SSL/TLS per TUTTA la comunicazione
_FTPSSLVER	Versione del protocollo SSL/TLS (default quello della libreria). Questo valore ha significato SOLO quando _FTPSSLTLS\$!= "NONE". Valori ammessi: <ul style="list-style-type: none"> • 1 = SSLv2 • 2 = SSLv3 • 3 = TLSv1 • 4 = TLSv1_0 • 5 = TLSv1_1 • 6 = TLSv1_2 • 99 = DEFAULT
_FTPCRTFILE\$	(SOLO protocollo FTP/FTPS e solo quando ho un certificato Client) Questo valore ha significato SOLO quando _FTPSSLTLS\$!= "NONE" Se si dispone di un certificato client da presentare al server va indicato qua (formato PEM)
_FTPCRTPASS\$	(SOLO protocollo FTP/FTPS e solo quando ho un certificato Client) Se il certificato prevede l'utilizzo di una Passphrase, settare questa variabile con il suo valore
_FTPCRTKEY\$	Chiave privata associata al certificato. Questo valore ha significato SOLO quando _FTPSSLTLS\$!= "NONE" oppure il protocollo indicato è sftp nel qual caso questa variabile dovrà contenere il percorso ed il nome del file contenente la chiave privata
_FTPAUTHOR\$	(SOLO protocollo FTP/FTPS) (Questo valore ha significato SOLO quando _FTPSSLTLS\$!= "NONE") File con le Autorità di certificazione (CA) in cui verificare la catena di certificazione del certificato presentato dal server
_FTPNOCRTCHK	(SOLO protocollo FTP/FTPS) (Questo valore ha significato SOLO quando _FTPSSLTLS\$!= "NONE") Se settata (< 0), significa chiedere di bypassare i controlli del certificato ricevuto dal Server; è la variabile con cui si stabilisce di fare (se = 0; default) oppure di non fare (se = 1), i controlli relativi alla coppia di valori: Nome Host + chiave pubblica inviata dal Server SFTP. Se si vogliono ottenere i controlli, occorre che questi dati siano registrati affinché i controlli, effettuati in occasione dei contatti successivi, possano avere un esito positivo. La variabile _FTPNOCRTCHK, pertanto, si avvale di altre due variabili _FTPWRHOST e _FTPKNWNHST\$, con le quali è si può fare in modo che la chiave pubblica del Server contattato, se ritenuto affidabile, venga salvata; l'operazione è da fare al primo contatto del Server e se c'è la certezza che sia veramente il Server da contattare (per esempio, confrontando la chiave ricevuta con quella preventivamente comunicata da <u>qualcuno</u> della parte remota). Il procedimento è simile a quanto avviene, ad esempio, nell'utilizzo dell'applicazione PuTTY per effettuare una connessione SSH: al primo contatto del Server, questi mostra la sua chiave pubblica e se ritenuta affidabile la si salva; quando successivamente si ricontatta il medesimo host, se questi presenta la stessa chiave pubblica, la coppia di valori viene trovata nel file in cui si registrano gli host affidabili e non viene ripetuta alcuna domanda.
_FTPWRHOST	Variabile che consente di salvare o meno la chiave pubblica di un server SFTP. Se il valore = 0 non viene salvato alcunchè (default); Se il valore = 1 la chiave pubblica + Nome Host viene salvata; il file in cui le chiavi dei Server ritenute affidabili vengono salvate - e comunicate - con il valore della variabile _FTPKNWNHST\$;
_FTPKNWNHST\$	SFTP - Variabile nome file in cui vengono registrati/verificati HOST+PubKEY (SSH Format);
_FTPLOGFILE\$	Nome del file di Log (viene creato nella cartella canonica per i log di Passepartout)
_FTPLOGLEV	Livello del log <ul style="list-style-type: none"> • 0 = Nessun log prodotto • 1 = viene stampato tutto il traffico, caratteri non stampabili rappresentati con '.' • 2 = Trace Log, vengono stampate sul file di log 2 colonne, nella prima tutti i byte con rappresentazione esadecimale nella seconda colonna la rappresentazione testo degli stessi byte della colonna 1
_FTPCNNTOUT	Timeout per la connessione al server FTP, in secondi; default = 60 secondi
_FTPTIMEOUT	Timeout (secondi) per tutto il trasferimento (se settato questo non devono esserlo _FTPMINBSEC e _FTPMAXBSLOW); default = 0 secondi
_FTPMINBSEC	Traffico minimo accettato per massimo _FTPMAXBSLOW secondi consecutivi prima di abbattere la connessione Bytes al secondo; default = 100 Bytes/secondo
_FTPMAXBSLOW	Numero di secondi dopo i quali, se il traffico si mantiene inferiore o uguale a _FTPMINBSEC (Bytes/sec), la connessione viene abbattuta; default 60 sec
_FTPLOCFILE\$	Percorso/Nome del file locale (da inviare se _FTPCMD\$="PUT" in cui salvare quello ricevuto se _FTPCMD\$="GET"). Il file può essere aperto SOLO nelle directory dirtemp '~', dirdatiutente ']' oppure sul file system del client '<' (per i client in grado di gestire il file system).
_FTPREMOTEFILE\$	Percorso/Nome del file remoto (su cui salvare se _FTPCMD\$="PUT", da prelevare se _FTPCMD\$="GET" o da cancellare se _FTPCMD\$="DEL"); il percorso eventualmente indicato nel valore di questa variabile viene accodato a quello indicato con il valore di _FTPURL\$; in caso di "PUT" e "DEL" comprende le sottocartelle di _FTPURL\$ (es. _FTPURL\$= "ftp://ftp.passepartout.sm/" : _FTPREMOTEFILE\$="sottocartella/file.txt"). Se il protocollo è SFTP si consiglia di indicare tutto il percorso in questa variabile, perchè la libreria potrebbe non riuscire a gestirlo (questione work in progress)
_FTPNUMTENT	Numero di tentativi da effettuare (oltre al primo) quando la connessione dovesse interrompersi; default = 0
_FTPRESUME	Se il file da trasferire con una GET dovesse essere stato parzialmente trasferito da precedenti tentativi, settando

	questa variabile con un valore <> 0, il nuovo tentativo di trasferimento riparte dal Byte successivo all'ultimo ricevuto, se invece il valore di questa variabile è = 0, il trasferimento riparte sempre dal primo Byte. (Nessun ChkSum)
_FTPCRREMDIR	In occasione di una PUT, se _FTPREFILE\$ contiene oltre ad un nome file anche un percorso, settando questa variabile con un valore <> 0 si chiede al server di creare le cartelle non presenti (se si dispone delle autorizzazioni ed il server "vuole" farlo)
_FTPCLRCMDC\$	(SOLO protocollo FTP/FTPS) (Questo valore ha significato SOLO quando _FTPSSLTLS\$!= "NONE") Settando questa variabile si chiede alla libreria sottostante (libcurl) di usare il ClearCommandChannel e di come gestire la fase di shutdown del layer SSL/TLS, Il ClearCommandChannel viene settato generalmente per aiutare le NAT a seguire la sessione FTP e viene quindi protetta la sola fase di autenticazione. I Valori ammessi sono <ul style="list-style-type: none"> • "NONE" = Tutta la comunicazione del CommandChannel è sotto SSL/TLS • "ACTIVE" = lo shutdown del layer SSL/TLS viene iniziato dal client • "PASSIVE" = lo shutdown del layer SSL/TLS viene iniziato dal server
_FTPRES\$()	Poiché un singolo comando prevede l'esecuzione di diversi step ed ognuno di questi comunica un proprio response, in questo array vengono riportati tutti in ordine cronologico, in maniera da poter determinare a fronte di errori quale operazione è fallita
_FTPDIRLST\$()	Se l'operazione richiesta è "DIR", cioè l'elenco dei file di una cartella, in questo array vengono riportate tutte le righe prodotte (Numero massimo di nomi riportabile 128000)
_FTPDIRNOMI	Settando questa variabile con valore <> 0, si chiede a fronte dell'esecuzione del comando "DIR" di salvare nell'array _FTPDIRLST\$() solo l'elenco dei nomi di file senza gli altri attributi (permission, ownership, size,...)
_FTPCODERR	Codice d'errore, determinato in base all'ultimo response, vale 0 se l'ultimo response è stato 226 (se questo dovesse risultare troppo semplice come criterio per determinare la buona riuscita di un'operazione, si possono analizzare tutti i response contenuti in _FTPRES\$())
_FTPDESERR\$	Messaggio d'errore, sempre in base all'ultimo response, vuoto quando l'ultimo response vale 226
_FTPUBKEY\$	Percorso/nome del file contenente la chiave pubblica, da utilizzare quando si specifica come protocollo sftp, la chiave privata va invece indicata in _FTPCRTKEY\$
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

GLOBAL

_GLOBAL()

_GLOBAL\$()

Variabili array globali (numerico e stringa) di massimo 32767 elementi ciascuno dove memorizzare dati generati da un Collage per renderli disponibili contemporaneamente a tutti i Collage caricati in memoria (in modo AUTO su evento o con CALL o STARTCOLLAGE), con visibilità/comunicabilità reciproca fra valori delle variabili e le routine su etichetta di Collage diversi.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#) o [Clicca qui](#)

GM - Anagrafica Gruppi Merceologici

_GM

Variabili di struttura archivi di Anagrafica Gruppi Merceologici (GETGM – PUTGM)	
_GMCOD\$	Codice gruppo merceologico;
_GMANN\$	Flag annullamento logico;
_GMDESS\$	Descrizione;
_GMDTC\$	Data creazione;
_GMDTM\$	Data ultimo aggiornamento;
_GMUM1\$	Unità di misura;
_GMRIC\$	Centro di ricavo;
_GMCOS\$	Centro di costo;
_GMCDP\$	Gruppo merceologico Padre;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

GPS - del evento ON_GETGPSDATA

_GPS

Variabili di struttura recupero dati GPS in evento Sprix Mobile ON_GETGPSDATA: (vedere istruzioni WCALL e GETGPSDATA)	
_GPSNUM	Numero di dati nell'array; viene resettato ad ogni chiamata
_GPSDATA\$()	Data Invio
_GPSORA\$()	Ora Invio
_GPSAZIENDA\$()	Sigla Azienda
_GPSUTENTE\$()	Utente
_GPSIDDISP\$()	Iddisp
_GPSNMDISP\$()	Nomedis
_GPSDISP\$()	Dispositivo
_GPSLAT\$()	Latitudine
_GPSLONG\$()	Longitudine
_GPSIND\$()	Indirizzo
_GPSLOC\$()	Località

_GPSIDU\$()	ID Utente
_GPSAPP\$()	Nome App
_GPSIDAPP\$()	ID App
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

IA - Iteratore ITERARC

IA

Variabili di struttura archivi del Iteratore ITERARC (SprixMobile)	
_IAORDNM\$(R)	opzionale , indica i campi su cui ordinare (es. = _ARCOD\$)
_IAORDDR\$(R)	opzionale, indica le direzioni di ordinamento dei campi su cui ordinare (es. = "D")
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

IL - Impegni di Lavorazione

IL

Variabili di struttura archivi degli Impegni di Lavorazione (GETIL – GETATTIL – PUTIL – DELIL) (le variabili evidenziate in grassetto sono quelle minime obbligatorie)	
_ILNDO(_NIL)	Numero documento;
_ILNSD(_NIL)	Numero sottodocumento;
_ILNRB(_NIL)	Numero riferimento prodotto finito: -1="RES_SL", -2="RES_DL";
_ILFAS(_NIL)	Numero fase prodotto finito;
_ILPOSDBP(_NIL)	Posizione articolo Materia Prima in DBP;
_ILCCFAS(_NIL)	Codice ciclo fase (sola lettura);
_ILMAG(_NIL)	Magazzino materie prime;
_ILDAD\$(_NIL)	Data scadenza riga bolla;
_ILAMP\$(_NIL)	Codice articolo Materia Prima;
_ILTQT(_NIL)	Misura: 1-99 taglie; 101 tipo articolo unità misura (um1); 102 tipo articolo unità misura (um2);
_ILCOL(_NIL)	Numero colli o quantità variabile;
_ILQTA(_NIL)	Quantità o quantità fissa;
_ILQTG(_NIL,16)	Quantità per taglia; T=1 to 16
_ILQTGXL(_NIL,16)	Quantità per taglia; T=17 to 32;
_ILPRZ(_NIL)	Prezzo;
_ILEVIS\$(_NIL)	Evidenzia; ' ', 'E', 'L', 'T';
_ILIND\$(_NIL)	Indice;
_ILDEP(_NIL)	Depositato (1 depositato, 0 non depositato);
_ILEDL(_NIL)	Edl (1 = si, 0 = no);
_ILES(_NIL)	Esl (1 = si, 0 = no);
_ILLAV(_NIL)	Lav (1 = si, 0 = no);
_ILFOR\$(_NIL)	Codice fornitore;
_ILTQVIN(_NIL)	Tipo di vincolo. Accetta valori da 0 a 8 <ul style="list-style-type: none"> 0 = Nessun Vincolo 1 = Il più presto possibile 2 = Il più tardi possibile 3 = Iniziare non prima del 4 = Iniziare non oltre il 5 = Finire non prima del 6 = Finire non oltre il 7 = Deve iniziare il 8 = Deve finire il
_ILORAVIN(_NIL)	Orario vincolo accetta valori da 0 a 2359; per indicare le 08:05 il valore è 805
_ILIDUBI(_NIL)	Id ubicazione impegni di produzione; viene tornata solo con la gestione della logistica;
_ILLAVDARIL\$(_NIL)	Lavorazione da rilevare (se non valorizzata in fase di PUT assume automaticamente il valore "N");
_ILLAVRILEV\$(_NIL)	Lavorazione rilevata (se non valorizzata in fase di PUT assume automaticamente il valore "N");
LOTTI E MATRICOLE (analoghe a quelle di struttura MM)	
_IL_INIZLOT(_NIL)	Consente di conoscere (GET) o stabilire (PUT) lo start dell'array che contiene l'elemento del lotto
_IL_MXQTLOT(_NIL)	Massima quantità di matricole; array monodimensionale
_ILL_IDLOT(L)	Id del lotto
_ILL_NDQTA(L)	Numero decimali quantità; array monodimensionale
_ILL_LQTA(L)	Quantità; array monodimensionale
_ILL_LCOLLI(L)	Colli; array monodimensionale
_ILL_LQTATGL(L,32)	Quantità per taglia; array bidimensionale
	Per leggere i lotti negli impegni: GETIL 64,0,1,1,1,1,"01" : per RIGAIMPEGNO = 1 A _NIL Si esegue il FOR per sapere quanti impegni ho letto _IL_MXQTLOT(RIGAIMPEGNO) per ELELOTTO = 1 TO _IL_MXQTLOT(RIGAIMPEGNO) Per ogni impegno letto ritrovo nella _IL_MXQTLOT(RIGAIMPEGNO) quante saranno le righe dei lotti associate alla riga di impegno. _ILL_IDLOT(ELELOTTO) : _ILL_NDQTA(ELELOTTO) : _ILL_LQTA(ELELOTTO) : eccetera ... PROSSIMO PROSSIMO

	In modo identifico si assegnano i lotti
	<p>IF(_IL_INIZLOT(R) > 0)THEN ULTIMOLOTTO = _IL_INIZLOT(R)+_IL_MXQTLOT(R); tramite questa IF si conosce l'ultimo elemento dell'array dei lotti; se _IL_MXQTLOT(R)=1 si sconsiglia l'uso di ciclo FOR (altrimenti ripeterebbe).</p> <p>Per impostare i lotti per una riga impegno già presente</p> <p>_IL_INIZLOT(NIL+1)=ULTIMOLOTTO _IL_MXQTLOT(NIL+1)=2</p> <p>_ILL_IDLOT(ULTIMOLOTTO) = 26 _ILL_NDQTA(ULTIMOLOTTO) =2 _ILL_LQTA(ULTIMOLOTTO) =2 _ILL_LCOLLI(ULTIMOLOTTO) = 0</p> <p>_ILL_IDLOT(ULTIMOLOTTO+1) = 97 _ILL_NDQTA(ULTIMOLOTTO+1) =2 _ILL_LQTA(ULTIMOLOTTO+1) =8 _ILL_LCOLLI(ULTIMOLOTTO+1) = 0</p>
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_INTER

_INTER

Variabile che attiva o disattiva la richiesta di interruzione forzata del programma con tasto Esc/Fine (equivalente alla combinazione [Ctrl+C]): se è = 1 attiva (default), se è = 0 disattiva (il tasto Esc/Fine viene attivato se impostato con istruzione SETTF).
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

_INTEST

_INTEST

Variabile che abilita (se =1, default) oppure disabilita (se =0 e ripristinabile a = 1) la stampa delle intestazioni. Elimina la formattazione della stampante con intestazioni (es. righe 'H'); non ha effetto nelle stampe senza la specifica di etichette di intestazione.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

_JSON

_JSON

Variabili di struttura relative alle operazioni JSON	
_JSONINPA\$(I)	Array monodimensionale su cui viene splittato dal programmatore un valore stringa da fornire alla funzione per l'inserimento di un dato di tipo JSON_STRING la cui dimensione sia superiore a 255 caratteri. Il primo elemento vuoto ("") viene considerato come terminazione del valore. La libreria assemblerà il valore per essere assegnato ad un nodo.
_JSONSTRLEN	Riporta la lunghezza totale del dato letto dalla funzione JSONGETSTR()
_JSONOUTA\$(I)	Array monodimensionale su cui viene splittato dalla libreria un valore di tipo JSON_STRING la cui dimensione sia superiore a 255 caratteri. Il primo elemento vuoto ("") viene considerato come terminazione del valore. Il programmatore dovrà considerare l'insieme degli elementi di questo array come un unico valore.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_LA - Lista Lotti

_LA

Variabili di sistema per la Lista Lotti nell'evento Collage di magazzino ON_LISTALOTTI	
_LAIDL()	Id lotto caricati in riga documento;
_LARIG	Numero riga che si sta processando;
_LANUM	Numero lotti gestiti nella riga;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_LISTADOCMM

_LISTADOCMM\$()

Variabile vettore contiene i 999 caratteri presenti a video all'evento lista documenti ON_LISTADOCMM: Tali informazioni, sono suddivise nei 4 slot dell'array (250+250+250+249 = 999)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

LOT - Anagrafica Lotti

LOT

Variabili di struttura archivi di Anagrafica Lotti (GETLOT – PUTLOT – DELLOT)	
_LOTID	ID lotto;
_LOTSTAT\$	Stato lotto;
_LOTMSG\$	Messaggio stato bloccato/annullato;
_LOTARM\$	Carico effettuato da movimento magazzino S/N;
_LOTTCCA\$	Tipo Controllo carico lotto;
_LOTTCCANEW\$	Controllo carico memorizzato in anagrafica lotto: se impostata modifica il tipo controllo: da M può andare a F o D o A; da D o F può andare ad A; se impostato un nuovo metodo di gestione dell'anagrafica lotto non è possibile tornare alla precedente;
_LOTART\$	Codice articolo;
_LOTCTPL\$	Codice tipo lotto;
_LOTFOR\$	Codice fornitore;
_LOTDTCA\$	Data di carico;
_LOTDTVA\$	Data validità;
_LOTDTCR\$	Data creazione (solo GET);
_LOTDTUM\$	Data ultima modifica (solo GET);
_LOTCUTE\$	Codice utente lotto;
_LOTDCD\$(999)	Codice campo dato utente personalizzato;
_LOTDTP\$(999)	Tipo dato utente personalizzato;
_LOTDUVL\$(999)	Valore dato utente personalizzato;
_LOTSTOR\$	Lotto storicizzato (flag S/N, sola lettura);

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LOT_M - Anagrafica Lotti (lettura mirata)

LOT_M

Variabili di struttura archivi di Anagrafica Lotti nella lettura mirata (GETMULTILOT)	
_LOTID_M()	ID lotto;
_LOTSTAT_M\$()	Stato lotto;
_LOTMSG_M\$()	Messaggio stato bloccato/annullato;
_LOTARM_M\$()	Carico effettuato da movimento magazzino S/N;
_LOTTCCA_M\$()	Tipo Controllo carico lotto;
_LOTART_M\$()	Codice articolo;
_LOTCTPL_M\$()	Codice tipo lotto;
_LOTFOR_M\$()	Codice fornitore;
_LOTDTCA_M\$()	Data di carico;
_LOTDTVA_M\$()	Data validità;
_LOTDTCR_M\$()	Data creazione;
_LOTDTUM_M\$()	Data ultima modifica;
_LOTCUTE_M\$()	Codice utente lotto;
_LOTSTOR_M\$()	Lotto storicizzato (flag S/N, sola lettura);

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LRECSORT

LRECSORT

Parametri	
= N°	Lunghezza in quantità di caratteri da ordinare;

Variabile che definisce la lunghezza in caratteri delle stringhe <str\$> da ordinare (lunghezza max della <str\$> da ordinare); la velocità di ordinamento è inversamente proporzionale alla lunghezza; se è > viene troncata, se è < sono aggiunti spazi bianchi.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LST – Liste di Prelievo

LST

Variabili di struttura archivi di Liste di Prelievo (GETLSTDOC– PUTLSTDOC– DELLSTDOC)	
_LSTID	ID lista;
_LSTDTCREA\$	Data creazione lista;
_LSTSTATO\$	Stato lista: valori ammessi (C)ostruzione, (D)a elaborare, (L) in lavorazione, (E)laborata, (T)erminata, (A)nnullata; quando la lista va in stato L, T o A non è possibile tornare a C e D; quando la lista è in stato T o A può solo assumere uno di questi 2 valori;
_LSTDESC\$	Descrizione lista;
_LSTTIPO\$	Tipo lista: valore ammesso (V)endita;
_LSTRINMAG\$()	Variabile di sola lettura valorizzata sulle righe tipo ordine da ZERODO: valori ammessi ' ' e 'M' (quando vale M l'ordine risulterà essere non modificabile);
_LSTRSIG\$()	Sigla documento;

_LSTRSAZ()	Sotto azienda;
_LSTRSER()	Serie documento;
_LSTRNUM()	Numero documento;
_LSTRNRCRE()	Progressivo di riga univoca nel documento di origine;
_LSTRPROG()	Progressivo di riga;
_LSTRART\$()	Codice articolo (solo lettura);
_LSTRARDESS\$()	Descrizione articolo (solo lettura);
_LSTRNRRART()	Record articolo (solo lettura);
_LSTTDADOC\$()	Data documento (solo lettura);
_LSTRMAG()	Magazzino (solo lettura nelle versioni precedenti alla 2022I);
_LSTRIDUBI()	ID ubicazione (solo lettura nelle versioni precedenti alla 2022I);
_LSTRSCA\$()	Data scadenza documento (solo lettura);
_LSTRTPUM\$()	Tipo unità di misura (solo lettura, 1(Um1/2),3(TG),4(Peso));
_LSTRUM\$()	Unità di misura (solo lettura);
_LSTRCOL()	Colli;
_LSTRQTA()	Quantità;
_LSTRNRDEC()	Numero decimali (solo lettura);
_LSTRKTA()	Coefficiente (solo lettura);
_LSTRIDLLOT()	ID lotto (dalla versione 2022H la variabile e' disponibile in scrittura con gestione lotti attiva e articoli gestiti a lotti);
_LSTRTAGTAB()	Tabella taglia (solo lettura);
_LSTRNUMTAG()	Numero taglia: questo campo viene valorizzato con il numero della taglia solo quando l'articolo nella riga di pick contiene quantità per un'unica taglia (solo lettura);
_LSTRTAG(,32)	Quantità per taglia (la somma delle quantità per taglia deve coincidere con il valore contenuto in _LSTRQTA)
_LSTRDALIAS()	ID alias (solo lettura);
_LSTRCLI\$()	Codice cliente (solo lettura);
_LSTRAGE\$()	Codice agente (solo lettura);
_LSTRVET\$()	Codice vettore (solo lettura);
_LSTSPED\$()	Codice conto di spedizione (solo lettura);
_LSTINDSPED\$()	Indirizzo di spedizione (solo lettura);
_LSTRINIZLOT()	Posizione iniziale in array lotti (solo lettura);
_LSTRMXQTLOT()	Numero elementi da posizione iniziale (solo lettura);
_LSTLIDLOT()	ID lotto (solo lettura);
_LSTLNDQTA()	Numero decimali in quantità lotto (solo lettura);
_LSTLQTA()	Quantità lotto (solo lettura);
_LSTLCOLLI()	Colli lotto (solo lettura);
_LSTLQTATGL(,32)	Quantità taglie per lotto (solo lettura);
_LSTPRIOR()	Priorità;
_LSTRRESSTR\$()	(R)esiduo/(S)tralcio valida solo in righe ordini. Per compatibilità con la prima versione rilasciata: vuota="R". Nelle righe di picking deve essere vuota;
Metadati (solo lettura)	
_LSTTDATCRE\$	Data di creazione (testata);
_LSTTORACRE\$	Ora di creazione (testata);
_LSTTUTECRE\$	Utente di creazione (testata);
_LSTTMENCRE\$	Menu' di creazione (testata);
_LSTTDATMOD\$	Ultima data di modifica (testata);
_LSTTORAMOD\$	Ultima ora di modifica (testata);
_LSTTUTEMOD\$	Ultimo utente di modifica (testata);
_LSTTMENMOD\$	Ultimo menu' di modifica (testata);
_LSTRDATCRE\$()	Data di creazione (righe ordine e pick);
_LSTRORACRE\$()	Ora di creazione (righe ordine e pick);
_LSTRUTECRE\$()	Utente di creazione (righe ordine e pick);
_LSTRMENCRE\$()	Menu' di creazione (righe ordine e pick);
_LSTRDATMOD\$()	Ultima data di modifica (righe ordine e pick);
_LSTRORAMOD\$()	Ultima ora di modifica (righe ordine e pick);
_LSTRUTEMOD\$()	Ultimo utente di modifica (righe ordine e pick);
_LSTRMENMOD\$()	Ultimo menu' di modifica (righe ordine e pick);
_LSTRIDUMOD\$()	ID utente ultima modifica (righe ordine e pick);
_LSTRDTOMOD\$()	Data e ora ultima modifica (righe ordine e pick).
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_MYDB

_MYDB

Variabili struttura archivi MyDB (MYDBGET – MYDBPUT – MYDBDEL – etc..)	
_MYDB\$()	Array che contiene i dati dell'archivio; ogni campo di definizione si riferisce ad un elemento del vettore. Tutti i valori sono trasformati in stringa. Le date sono in formato "AAAAMMGG". E' impostato dalle istruzioni MYDBGET, MYDBGETATT e MYDBVIEW; usato dalle funzioni MYDBPUT, MYDBPUTATT e MYDBINPUT; azzerato da AZZVARSYS 35
_MYDBDIN\$()	Array di _MYDBELE elementi come _MYDB\$(). Contiene il flag di riportabilità sui documenti. Relativo alle anagrafiche DINAMICHE.
_MYDBELE	Ritorna il numero degli elementi MYDB\$() inseriti nella definizione dell'anagrafica/tabella. Viene impostata dall'istruzione MYDBGET
_MYDBETIC\$()	Array che ritorna le etichette dei campi MYDB\$() impostati nella definizione. E' impostato dall'istruzione MYDBGET

	<p>Esempio di stampa dei campi di un record con la loro etichetta: MYDBGGET "1900074000_interven", 0, 7 APRISTAMP_EXT 127,"Prova",80 FOR I = 1 TO _MYDBELE ' _MYDBELE Num elementi vettore _MYDB\$(' _MYDBETIC\$ etichetta del campo + _MYDB\$(I) valore del campo LPRINT LEFT\$(_MYDBETIC\$(I)+STRING\$(30,""),25)+": "+_MYDB\$(I) NEXT</p>
_MYDBK\$()	<p>Variabile array di 5 elementi per la chiave MyDB (utilizzata da MYDBGGET, la composizione della chiave è indicata in seguito in queste tabelle). Attualmente nell'array è disponibile solo il primo elemento (1), pertanto può essere utilizzata solo nella seguente forma _MYDBK\$(1). I restanti quattro elementi dell'array (2, 3, 4, 5) non sono attivi. La variabile array è utilizzata dalle istruzioni MYDBGGET e MYDBVIEW per indicare i vari elementi della chiave.</p> <p>Esempio: Indica che il valore da ricercare, nelle relative istruzioni, è quello del campo impostato come ordinamento 1 _MYDBK\$(1) = "501.00001" MYDBGGET "1900074000_interven", 1, 0 IF _ERRMYDB\$ <> "" THEN VIMSG 5,_ERRMYDB\$</p> <p>****Ricerca tramite l'estensione****</p> <p>Esempio: Indica che il valore da ricercare, nelle relative istruzioni, è quello del campo impostato come estensione (99) _MYDBK\$(1) = "501.00001" MYDBGGET "1900074000_interven", 99, 0 IF _ERRMYDB\$ <> "" THEN VIMSG 5,_ERRMYDB\$</p>
_MYDBID	<p>Numero ID anagrafica corrente. Impostata/usata dalle funzioni MYDBGGET, MYDBPUT, MYDBPREV, MYDBNEXT, MYDBVIEW, MYDBINPUT, MYDBGGETATT e MYDBPUTATT Non è disponibile una funzione che ritorni l'ultimo ID usato (o il prossimo ID), perchè prima del salvataggio non è disponibile questa informazione; occorre gestire in altri modi, alcuni esempi: tramite MYDBGGETATT che all'ON_PUTMYDB ritorna _MYDBID appena salvato, oppure tramite la gestione di una tabella numeratore, oppure tramite MYDBLOF</p>
_MYDBNR	<p>Numero di record dell'anagrafica corrente. Impostata da MYDBGGET, MYDBPUT, MYDBDEL, MYDBPREV, MYDBNEXT, MYDBVIEW e MYDBINPUT</p>
_MYDBARC\$	<p>Variabile che ritorna il nome dell'archivio MyDB in cui l'input è posizionato (codice archivio corrente). Valido solo per MYDBGGETATT, MYDBPUTATT</p>
_MYDBANN\$	<p>Variabile che ritorna "S" se il record è annullato logicamente</p>
_MYDBCOD\$()	<p>Vettore che ritorna i codici di campo definiti nell'anagrafica. Corrispondente ai vari elementi _MYDB\$(I). E' impostato dalle istruzioni MYDBGGET e MYDBGGETATT</p>
_MYDBBLOC\$	<p>"S" se l'anagrafica è bloccata da Passepartout (non modificabile)</p>
_MYDBTF	<p>Ritorna il codice del tasto funzione utilizzato nei comandi MYDB (sostituisce la classica funzione _TF); valorizzata solo in MYDBVIEW e MYDBINPUT</p>
_MYDBF\$()	<p>Array utilizzato dall'istruzione MYDBVIEW per l'impostazione dei filtri di selezione; viene azzerato da AZZVARSYS 35. L'array funziona come _MYDB\$(I) della tabella che si vuole filtrare. Ogni campo di definizione della tabella si riferisce ad un elemento del vettore. Tutti i valori sono stringa.</p> <p>Se l'elemento del vettore è definito come campo alfanumerico o fa riferimento ad un'anagrafica MyDB è presente anche il campo _MYDBF_TP\$(I) che attiva la ricerca avanzata, con la possibilità di specificare il tipo di controllo da effettuare sul campo:</p> <ul style="list-style-type: none"> • (*) Ricerca (NON controlla maiuscolo/minuscolo) • (\$) Ricerca (Controlla maiuscolo/minuscolo) • (=) Uguale (Controlla maiuscolo/minuscolo) • () Uguale (Ricerca classica – NON controlla maiuscolo/minuscolo) • (.) Inizia per (Controlla maiuscolo/minuscolo) • (,) Inizia per (NON controlla maiuscolo/minuscolo) <p>Se l'elemento del vettore è una data, è presente anche il campo _MYDBF_DTA\$(I) per poter fare delle selezioni per periodo (Da data → A data). Se si vuole selezionare una data sola è comunque necessario inserire la stessa data su entrambi i campi.</p> <p>La MYDBVIEW visualizza la tabella filtrata in base ad ogni elemento del vettore valorizzato.</p> <p>Nota Bene: Nel caso in cui l'elemento dell'array _MYDBF\$(I) sia un campo dell'archivio stesso, si imposta il valore del campo. Se l'elemento dell'array _MYDBF\$(I) è una relazione ad un altro archivio o tabella MYDB, il valore da impostare nell'elemento _MYDBF\$(X) è il _MYDBID dell'archivio relazionato.</p>

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

Dizionario	
sxmyd	<p>Nome archivio MyDB principale in revisione.</p> <p>E' relazionabile al dizionario 'sxarc'. Esempio: IIF sxarc<>"123456esempi_ANMYDB" ("CodConNomApp_ANMYDB")</p> <p>Note:</p> <p>Il nome dell'App "CodConNomApp" è indispensabile, anche se è contenuta nella medesima App.</p>

	Il nome dell'App "CodConNomApp" è separato dal nome dell'anagrafica MyDB "ANMYDB" dal carattere "_"; (in passato, alcuni casi richiedevano il carattere "@"). ATTENZIONE: nella stringa "case sensitive" (maiuscole e minuscole) avvalersi delle funzioni CVMAIU(<str\$>) e CVMIN (<str\$>)
sxarc	Nome archivio MyDB corrente (cioè usato da <sxmyd>). Revisione di un'anagrafica all'interno di un'altra anagrafica.

Sintassi dei nomi degli ARCHIVI MyDB (modalità MyDB precedente a PassBuilder)	
Generali o Estensioni dirette dei doc.	PRODUTTORE_ARCHIVIO (10 car. Produttore + _ + 8 car. Codice archivio)
Parcelle:	PRODUTTORE_ARCHIVIO??? (10 car. Produttore + _ + 8 car. Codice archivio+3 caratteri sezione parcella ('PAT' testata / 'PAR' riga))
Riportabili e Dinamiche sui Documenti:	PRODUTTORE_ARCHIVIO???? (10 car. Del produttore + _ + 8 car. Del nome archivio+ 4 car. Sezione documento (*1))

PassBuilder: sintassi dei nomi degli ARCHIVI MyDB presenti all'interno di una App:	
Generali o Estensioni dirette dei doc.	ARCHIVIO (8 car. Codice archivio)
Parcelle	ARCHIVIO??? (8 car. Codice archivio+3 caratteri sezione parcella ('PAT' testata / 'PAR' riga))
Riportabili e Dinamiche sui Documenti	ARCHIVIO???? (8 car. Del nome archivio+ 4 car. Sezione documento (*1))

PassBuilder: sintassi dei nomi degli ARCHIVI MyDB presenti all'interno di un'altra App in cui tali archivi siano stati condivisi:	
Generali o Estensioni dirette dei doc.	ydCODICEAPP@ARCHIVIOCam() (yd + 6 car. Installazione + 10 car. Applicazione + @ + 8 car. Codice archivio + 3 car. Codice campo).
Parcelle	ydCODICEAPP@ARCHIVIO???Cam() (yd + 6 car. Installazione + 10 car. Applicazione + @ + 8 car. Codice archivio+ 3 caratteri sezione parcella ('PAT' testata / 'PAR' riga) + 3 car. Codice campo)
Riportabili e Dinamiche sui Documenti	yyCODICEAPP@ARCHIVIO????Cam() (yy + 6 car. Installazione + 10 car. Applicazione + @ + 8 car. Del nome archivio+ 4 car. Sezione documento (*1) + 3 car. Codice campo)

Sintassi dei nomi degli ARCHIVI MyDB presenti in PassBuilder, ma richiamati dall'esterno:	
Generali o Estensioni dirette dei doc.	CODICEAPP_ARCHIVIO (6 car. Installazione + 10 car. Applicazione + _ + 8 car. Codice archivio)
Parcelle	CODICEAPP_ARCHIVIO??? (6 car. Installazione + 10 car. Applicazione + 8 car. Codice archivio+3 caratteri sezione parcella ('PAT' testata / 'PAR' riga))
Riportabili e Dinamiche sui Documenti	CODICEAPP_ARCHIVIO???? (6 car. Installazione + 10 car. Applicazione + _ + 8 car. Del nome archivio+ 4 car. Sezione documento (*1))

Sintassi dei nomi delle VARIABILI di dizionario degli archivi MyDB:	
Generali o Estensioni dirette dei doc.	ydPRODUTTORE_ARCHIVIOCam() (yd + 10 car. Produttore + _ + 8 car. Codice archivio + 3 car. Codice campo)
Parcelle:	ydPRODUTTORE_ARCHIVIO???Cam() (yd + 10 car. Produttore + _ + 8 car. Codice archivio+3 caratteri sezione parcella ('PAT' testata / 'PAR' riga) + 3 car. Codice campo)
Riportabili e Dinamiche sui Documenti:	yyPRODUTTORE_ARCHIVIO????Cam() (yy + 10 car. Del produttore + _ + 8 car. Del nome archivio+ 4 car. Sezione documento (*1) + 3 car. Codice campo)

Sintassi dei nomi delle VARIABILI di dizionario degli archivi MyDB all'interno di PassBuilder:	
Generali o Estensioni dirette dei doc.	ydARCHIVIOCam()(yd + 8 car. Codice archivio + 3 car. Codice campo)
Parcelle	ydARCHIVIO???Cam() (yd + 8 car. Codice archivio+ 3 caratteri sezione parcella ('PAT' testata / 'PAR' riga) + 3 car. Codice campo)
Riportabili e Dinamiche sui Documenti	yyARCHIVIO????Cam() (yy + 8 car. Del nome archivio+ 4 car. Sezione documento (*1) + 3 car. Codice campo)

Sintassi nomi VARIABILI dizionario degli archivi MyDB presenti in PassBuilder, ma richiamate dall'esterno:	
Generali o Estensioni dirette dei doc.	ydCODICEAPP_ARCHIVIOCam() (yd + 6 car. Installazione + 10 car. Applicazione + _ + 8 car. Codice archivio + 3 car. Codice campo)
Parcelle	ydCODICEAPP_ARCHIVIO???Cam() (yd + 6 car. Installazione + 10 car. Applicazione + 8 car. Codice archivio+ 3 caratteri sezione parcella ('PAT' testata / 'PAR' riga) + 3 car. Codice campo)
Riportabili e Dinamiche sui Documenti	yyCODICEAPP_ARCHIVIO????Cam() (yy + 6 car. Installazione + 10 car. Applicazione + _ + 8 car. Del nome archivio+ 4 car. Sezione documento (*1) + 3 car. Codice campo)

(*1) – SEZIONI dei documenti utilizzate nei nomi delle anagrafiche **riportabili e dinamiche**

(nota: non riguardano le anagrafiche che sono **estensione diretta** di testate/righe)

Identificativo MyDB	Sezione
DMTC	Testate Documenti Clienti
DMTF	Testate Documenti Fornitore

DMTX	Testate Documenti Clienti e Fornitori
DMTN	Testate Documenti senza conto
DMTT	Testate Tutti i documenti
DMRX	Righe Articoli valorizzati
DMRA	Righe Articoli di tipo merce
DMRL	Righe Articoli di tipo lavorazione
DMRS	Righe Articoli di tipo spese
DMRZ	Righe Articoli di tipo prestazioni
DMRP	Righe Articoli di tipo conto/deposito
DMRT	Righe Tutti i tipi di articoli
PAT	Testate Parcelle
PAR	Righe Prestazioni parcelle
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

MM - Documenti di magazzino

MM

Variabili di struttura archivi dei Documenti di Magazzino (GETMM – GETMM_EXT – GETATTMM – PUTMM – RIGMM – TOTMM – PUTPNX – DELMM – DELMM_EXT). Elementi variabili array: (R) = righe documento; (T) = riferimenti testate (fusione documenti); (L) = riga lotto (max 200). Le variabili evidenziate in grassetto sono quelle minime obbligatorie.	
TESTATA DOCUMENTO	
_MMTIPELETT\$	Tipo documento elettronico. Variabile testata con precedenza su tutte le altre. Stringa di 4 caratteri di tipo "TDnn" (dove "nn" è un "numero" che va da 01 al tipo documento gestito). Es. doc FT/FS/FD: "TD01", "TD02", "TD03", "TD05", "TD06", "TD24", "TD25", "TD26", "TD27" doc NC/NF: "TD04" doc FF: "TD01", "TD02", "TD03", "TD05", "TD06", "TD20", "TD21", "TD22", "TD23", "TD24", "TD25", "TD26" Il tipo "TD06" è ammesso solo per l'azienda professionista Nota: se la sigla non viene trovata (non compresa nel range da TD01 a TD20) ritorna il messaggio "Sigla tipo documento elettronico non valida". Se la variabile _MMTIPELETT\$ vale "" (vuota), verrà impostato il valore di default per il tipo documento che si sta creando. In caso di documento FFE deve essere un fornitore self billing per potere procedere, altrimenti se FFE con fornitore elettronico NON self billing occorre specificare il TD20 per potere procedere. Se il documento non utilizza un modulo di stampa elettronico, allora l'informazione viene azzerata e si procede; se invece il modulo di stampa è elettronico allora viene controllato che il TDxx che si sta impostando sia ammesso per il tipo documento che si sta gestendo; se il TDxx non è ammesso, compare il messaggio di errore "Valore TDxx non ammesso" e la procedura esce con errore. In caso di documento elettronico TD20 con fornitore dove non è specificato il regime fiscale XML, compare il messaggio bloccante: "L'emissione di questo documento necessita del 'Regime fiscale FatturaXML'. Controllare nella scheda anagrafica del fornitore."
MMSIG\$	Sigla Documento;
MMSAZ	Codice sottoazienda documento attuale, v. APRIAZIE_EXT;
_MMMODSTP\$	Tipo modulo di stampa (variabile di 1 carattere rappresentante il terzo carattere della sigla documento); Nota: nel caso di gestione del formato documento con o senza il carattere alfanumerico, occorre considerare che la gestione di <u>più formati dello stesso tipo documento</u> presuppone l'opportuna inizializzazione della variabile _MMMODSTP\$. In presenza di più formati per un medesimo tipo di documento (es. "OCW" e "OC"), in caso di utilizzo del formato senza il carattere alfanumerico associato al tipo documento (es. "OC") il campo _MMMODSTP\$ ha comunque un valore diverso da "null", bensì è pari a "blank". L'inizializzazione di MMSIG\$="OC" , pertanto, dev'essere accompagnata da _MMMODSTP\$=" " . Quindi, se in Sprix si desidera attivare la gestione <u>multipla di formati dello stesso tipo di documento</u> (es. "OCA", "OC", "OCB", ecc) e se si desidera gestire correttamente il formato che non riporta ulteriori indicazioni (es. il formato n.2 "OC") è necessario inizializzare _MMMODSTP\$=" " , poiché se non lo si facesse, _MMMODSTP\$ risulterebbe "null" (_MMMODSTP\$="") e di conseguenza il programma non saprebbe quale modulo utilizzare, eccetto il primo che incontra ("OCA").
MMCMO(T)	Causale movimento; per movimentazioni solo a valore consultare _MMTIP\$(R) ;
_MMSER	Serie documento; se il destinatario del documento gestisce la fattura elettronica o la fattura PA (Pubblica Amministrazione), il numero di serie è prelevato in automatico;
_MMNUM	Numero documento. Se = 0 significa numerazione AUTO, dopo la PUT il documento nuovo contiene il numero. Vedere anche nota (*b) dei 'Riferimenti Documenti';
MMDAT\$	Data documento; valorizzata al ON_DELMM;
MxEcr e MxRT: documento telematico	
Riferimenti documento per Sigle documento "AN" (annullamento) - "RE" (reso) Nota: i campi per definire il documento di riferimento sono gestiti manualmente	
_MMNCGDOCRIF	Numero di chiusura giornaliera del documento di riferimento (è il primo numero) - dizionario 'mmnrg'; variabile da indicare per la PUTMM di AN/RE del documento RT da annullare/rendere.
_MMNDOCRIF	Numero del documento di riferimento (è il secondo numero) - dizionario 'mmndr'; variabile da indicare per la PUTMM di AN/RE del documento RT da annullare/rendere.
_MMDDOCRIF\$	Data del documento di riferimento - dizionario 'mmdrr'; variabile da indicare per la PUTMM di AN/RE del documento RT da annullare/rendere.
_MMNMATRRT\$	Matricola Registratore Telematico del documento di riferimento (deve essere lunga 11) - dizionario 'mnmtr'; variabile da indicare per la PUTMM di AN/RE del documento RT da annullare/rendere.
_MMSIGDRIF\$	Tipo documento a cui AN si riferisce (da v691 anche per RE); (v. nota)
_MMSAZDRIF	Codice sottoazienda del documento a cui AN si riferisce (da v691 anche per RE); (v. nota)
_MMSERDRIF	Serie documento (da 1 a 99) del documento a cui AN si riferisce (da v691 anche per RE); (v. nota)
_MMNUMDRIF	Numero documento (da 1 a 999.999) del documento a cui AN si riferisce (da v691 anche per RE); (v. nota)
_MMDATDRIF\$	Data documento del documento a cui AN si riferisce (da v691 anche per RE); (v. nota)

		nota)
		Nota: Va indicato il CO al quale si fa riferimento. Attualmente la PUTMM non controlla che il CO ed i relativi dati RT esistano effettivamente.
	_MMCALTRIF\$	Codice in lettura/scrittura, che identifica quale 'Altri Riferimenti' e' stato impiegato per eseguire un reso (valido solo per RE); valori: "P"=POS, "N"=ND,"V"=VR; (dizionario "mmiar")
	_MMDALTRIF\$	Data in lettura/scrittura, che indica in quale momento e' stato effettuato un reso tramite il codice 'Altri riferimenti' (valido solo per RE); (dizionario "mmdar")
		Se è configurato MxRT e la data del documento è < alla data di gestione della versione XML 7.0 oppure la data è assente o il documento è <> RE, la PUTMM azzerà le variabili _MMCALTRIF\$ _MMDALTRIF\$.
		Se il documento è = RE, la PUTMM valorizza _MMCALTRIF\$ ma non valorizza _MMDALTRIF\$, di conseguenza è rilasciato l'errore "La data degli 'Altri Riferimenti' per effettuare il Reso non può essere vuota"; la data degli 'Altri Riferimenti' non può essere superiore alla data documento.
		Se _MMDALTRIF\$ è valorizzata e la variabile _MMCALTRIF\$ non è valorizzata, è rilasciato l'errore "Il codice degli Altri Riferimenti per effettuare il Reso non può essere vuoto". Se la variabile _MMCALTRIF\$ è valorizzata, vengono azzerate le variabili _MMNUMDRIF, _MMDATDRIF\$, _MMNCGDOCRIF, _MMNDOCRIF, _MMDDOCRIF\$, _MMNMATRRT\$.
	_MMRTDOCTEL\$	Matricola registratore telematico documento stampato; (v. nota)
	_MMDTDOCTEL\$	Data documento stampato; (v. nota)
	_MMNCGDOCTEL	Numero chiusura documento stampato; (v. nota)
	_MMNDOCTEL	Numero documento stampato; (v. nota)
		Nota: dopo la GETMM la variabile contiene i riferimenti del documento telematico se già stampato; se la PUTMM esegue la stampa del documento telematico la variabile è impostata con i dati del documento stampato.
	_MMNODOCCOM\$	Disabilita stampa documento commerciale (scontrino fiscale) per CO/RE/AN vale "S" o "N" (default). (v. nota)
	_MMDOCGES\$	Abilita stampa documento gestionale (non fiscale) per CO vale "N" = No (default) , "S" = scontrino non fiscale,"C" = scontrino di cortesia. (v. nota)
		Nota: Dopo la GETMM contengono i default. Prima di una PUTMM occorre valorizzarle se si vuole disabilitare la stampa fiscale (_MMNODOCCOM\$="S") oppure se si vuole abilitare la stampa non fiscale (_MMDOCGES\$="S" oppure _MMDOCGES\$ = "C").
		Come il Gestionale, in caso di registrazione di documento non fiscale questo viene salvato con il default _MMNODOCCOM\$="N" (la spunta del campo MD_STPFISC) per agevolare la ristampa di tipo "fiscale".
		Notare: per stampare un documento telematico (CO, RE, AN) è sufficiente utilizzare la funzione PUTMM con il parametro impostato con uno dei seguenti valori: 2, 3, 5 (esegue solo la stampa sul RT), 12, 13. Anche la PUTMM come le altre funzioni di gestione registratore telematico, se è stata chiamata prima la funzione RTAPRI usa il registratore telematico aperto altrimenti utilizza quello default impostato sul Client.
	MMMAG	Magazzino; vedere anche Magazzino di riga;
	_MMAMA	A Magazzino; se 0 solo scarico _MMMAG; vedere anche A Magazzino di riga;
	_MMCCR	Centro di costo/ricavo;
	_MMCIMMORIS\$	Centro costo ricavo (CO/RI) contabilizzato primanota S/N - solo revisione-;
	MMCLI\$	Codice cliente/fornitore;
	MMCODCONT	Codice contatto; numerico 999999999
	_MMSTA(T)	Categoria statistica cliente/fornitore;
	_MMZON(T)	Zona cliente/fornitore;
	_MMIEV\$	Spedizione documento interamente evadibile S/N
	_MMALISPTRA\$	Aliquota IVA ripartizione spese trasporto. Nota: gli aggiornamenti di archivi provenienti dalla versione 2016A alla 2016I, in fase di lettura/scrittura del documento di magazzino possono provocare un messaggio di aliquota iva spese trasporto errata, poiché viene scritta nel formato " 00,0" anziché "". Occorre aggiungere il seguente test prima della scrittura (dalla versione 2016J non è più necessario): IF _MMALISPTRA\$= " 00,0" THEN _MMALISPTRA\$="" oppure utilizzare il pulsante "Ricarica spese" sul piede del documento.
	_MMEVD\$	Nuova riga evadibile di default S/N;
	_MMPICKINGS\$	Valori ammessi S/N; negli ordini cliente imposta stato Picking (valorizzata a S imposta a N le variabili _MMIEV\$ e _MMEVD\$); negli ordini fornitore imposta il contenuto del campo Stato;
	_MMINMAGAZ\$	Variabile flag di sola lettura che indica se le modifiche all'ordine sono state bloccate da procedura ZERODO: nel caso di valore pari a S il documento non sarà ne modificabile ne trasformabile ne cancellabile;
	_MMAGE\$	Codice agente documento o di default – Tipo F/P;
	_MMAGEDEF\$	Se=1 → _MMAGE\$ è default – non del documento – solo GET;
	_MMAGECOND	Numero condizione agente;
	_MMNOT\$(T)	Nota;
	_MMSCMERCE(T)	Sconto merce documento da 0 a 99,99. Se valorizzata il numero di sconti di riga si riduce da 9 a 8; La valorizzazione di questa variabile è incompatibile con forme di pagamento che presentano sconti di tipo A o M. Anche nel caso di forme di pagamento con sconto di tipo D, è necessario valorizzare questa variabile leggendo lo sconto dalla tabella pagamenti (cpsco(mmpag))
	_MMCLOTTRT\$	Codice Lotteria per documenti CO, RE, AN (MxRT); 8 caratteri alfanumerico proposto da anagrafica Conto se il documento è intestato; è modificabile, ma comunque univoco in caso di fusione documenti;
	_MMDOCOLD	Se=1 → documento precedente a versione 12.3 –arrotondamento difetto- ; se=0 posteriore alla medesima versione –arrotondamento medio-

_MMMRAS(T,12)	Stato documento attivo S/N;
_MMMRV\$(T,12)	Stato documento variato S/N; solo GET;
_MMIDPASSWEB	Numero sito Passweb;
_MMFFDP\$	Indica se FF IntraCee a doppio protocollo; -solo GET;
_MMEST	Valuta documento;
_MMVAL	Valuta – Cambio Valuta Documento → Valuta Gestione; (v. nota)
_MMCVF\$	Valuta – Cambio; 'V'=variabile, 'F'=fisso; La variabile _MMCVF\$="F" imposta il cambio fisso per inserire un cambio arbitrario da memorizzare sul documento e consultabile tramite le rimanenti variabili dei cambi (v. nota "a della struttura _MM)
_MMVVL\$	Valuta – Fissa prezzi; 'C'=Valuta Contabile, 'D'=Valuta Documento;
_MMEVE+ _MMDIE	Cambio valuta Documento → Euro + Flag cambio 0=No/1=Si; (v. nota)
_MMEVC+ _MMDIC	Cambio Euro → valuta Gest + Flag cambio 0=No/1=Si; (v. nota)
	Nota: solo GETATTMM con valuta documento <> da quella di gestione contabile: se il documento è emesso nella stessa valuta di gestione dell'azienda il Cambio = 1 e il Flag = 0. Dato un importo in valuta di partenza il Flag stabilisce come operare per calcolare l'importo nella valuta di arrivo: se Flag= 0 moltiplicare oppure se = 1 dividere. Es. triangolazione con area Euro (100\$->Eu-Lir) dati i cambi \$: Eu = 1,1565:1 e Eu : Lir = 1:1936,27 : (se _MMDIE=1 e _MMEVE=1,1565 -> 100:1,1565 = 86,47 Eu; (se _MMDIE=0 e _MMEVE=0,86467 -> 100*0,86467 = 86,47 Eu -cambio inverso-; Eu -> Lir (_MMDIC=0 e _MMEVC=1936,27) 87,47 Eu*1936,27 = 167,429 Lir
_MMTPSPTRO\$	Operazione riferita a contratto S/N;
_MMCROSPTRO\$	Codice Contratto;
_MMCIC(\$)	Cointestatori spesometro – conto;
_MMCIP()	Cointestatori spesometro – percentuale;
_MMTANNOLET(T)	Anno di ricevimento della Dichiarazione d'Intento; (solo doc. BC-FT-RC-NC-FC); numero formato dalle due cifre finali dell'anno, oppure -1 per modalità automatica; variabile di testata analoga a quella di riga _MMANNOLETT(R). v. nota
_MMTPROGLET(T)	Progressivo di ricevimento della Dichiarazione d'Intento; variabile di testata analoga a quella di riga _MMPROGLETT(R). v. nota
	Nota: _MMTANNOLET(T) e _MMTPROGLET(T) sono variabili di testata documento che sono utilizzate per assoggettare le spese accessorie (es. trasporto) al regime IVA delle dichiarazioni d'intento.
_MMMYPDB(T)	Dati MyDB collegati alla testata del documento. Valori accettati: <ul style="list-style-type: none"> (-1) → scrive automaticamente un record mydb su tutte le anagrafiche MyDB collegate alla testata, se e' stato specificato un valore iniziale per le estensioni documento o se l'anagrafica e' un'estensione riportabile e sono presenti dati in anagrafica; (0) → non sono presenti dati Mydb; (1) → sono presenti dati Mydb;
_MMNOADDBV\$	Escludi addebito importo bollo;
_MMNOBV\$	Escludi applicazione bollo;
CONTROLLO DI GESTIONE	
_MMDTANL\$	Data propositiva competenza analitica (se blank viene assunta data documento);
_MMCOMMANL\$	Commessa propositiva competenza analitica (massimo 8 caratteri);
_MMAREAANL\$	Area propositiva competenza analitica (massimo 4 caratteri);
	Nota bene: perché la scrittura avvenga, oltre alla valorizzazione delle presenti variabili, è indispensabile che nell'anagrafica degli articoli inseriti nel documento, siano state precedentemente valorizzate le voci di costo e di ricavo dei codici analitici presenti nelle condizioni contabili.
PROPRIETÀ DOCUMENTO (solo lettura)	
_MMDATCRE\$	Data creazione del documento;
_MMORACRE\$	Ora creazione del documento;
_MMUTECRE\$	Utente creazione del documento;
_MMMENCRES\$	Menu creazione del documento;
_MMDATVAR\$	Data ultima variazione documento; solo tecnologia BP;
_MMORAMOD\$	Ora ultima modifica del documento;
_MMUTEMOD\$	Utente ultima modifica del documento;
_MMMENMOD\$	Menu ultima modifica del documento;
RIFERIMENTI DOCUMENTI	
_MMRBO	Riferimenti Documento – Numero riferimento bolla di lavorazione; (v. nota)
_MMRSB	Riferimenti Documento – Numero riferimento sottobolla di lavorazione; (v. nota)
_MMRFB	Riferimenti Documento – Numero riferimento fase di lavorazione; (v. nota)
_MMMSAZ(T)	Riferimenti Documento – codice sottoazienda documento movimento magazzino, vedere APRIAZIE; (v. nota)
_MMOSAZ(T)	Riferimenti Documento – codice sottoazienda documento ordine origine, vedere APRIAZIE; (v. nota)
_MMOSI\$(T)	Riferimenti Documento – sigla documento ordine; (v. nota)
_MMOSE(T)	Riferimenti Documento – serie documento ordine; (v. nota)
_MMONU(T)	Riferimenti Documento – numero documento ordine; (v. nota)
_MMODA\$(T)	Riferimenti Documento – data documento ordine; (v. nota)
_MMMSI\$(T)	Riferimenti Documento – sigla documento origine movimento magazzino; (v. nota)
_MMMSE(T)	Riferimenti Documento – serie documento origine movimento magazzino; (v. nota)
_MMMNU(T)	Riferimenti Documento – numero documento origine movimento magazzino; (v. nota)
_MMMDSA\$(T)	Riferimenti Documento – data documento origine movimento magazzino; (v. nota)
_MMSIGRE\$	Riferimenti Documento Esterno – Sigla documento attuale; (v. nota)
_MMNUMRE\$	Riferimenti Documento Esterno – Numero documento attuale; (v. nota)
_MMDATRE\$	Riferimenti Documento Esterno – Data documento attuale; (v. nota)
_MMMSIRE\$(T)	Riferimenti Documento Esterno – Sigla documento di carico/scarico; (v. nota)
_MMMNURE\$(T)	Riferimenti Documento Esterno – Num documento di carico/scarico; (v. nota)
_MMMDSA\$(T)	Riferimenti Documento Esterno – Data documento di carico/scarico; (v. nota)
_MMOSIRE\$(T)	Riferimenti Documento Esterno – Sigla documento ordine origine; (v. nota)
_MMONURE\$(T)	Riferimenti Documento Esterno – Num documento ordine origine; (v. nota)
_MMODARE\$(T)	Riferimenti Documento Esterno – Data documento ordine origine; (v. nota)

	<p>Nota: Caratteri alfanumerici di numerazione e Riferimenti Documento</p> <p>In primanota e magazzino è possibile utilizzare i caratteri alfanumerici nella numerazione dei documenti che utilizzano il registro acquisti. Sia in primanota che in magazzino, il dato in input del numero di documento può essere numerico o alfanumerico, ed è gestito a scorrimento: se alfanumerico, in uscita dal campo questo viene elaborato in numerico (c'è una estrapolazione dei soli numeri a partire da destra, per la formazione della chiave numerica del documento), rientrando nel campo viene ripristinata la visualizzazione dell'alfanumerico (in magazzino, il dato è gestito per i nuovi documenti FF e NF ed è gestito anche nel campo "Numero" dei Riferimenti Esterni [F4] e non è svincolato dal numero documento; per i documenti vecchi o emessi senza questa modalità, la gestione rimane inalterata). In magazzino l'input alfanumerico è con <code>_MMNUMRE\$</code>, in primanota è con la variabile <code>_PNNUMRE\$</code>; non sono obbligatorie, se impostate devono contenere almeno un carattere numerico; l'input da gestionale imposta il numerico e in Sprix, ciò avviene quando la variabile numerica (cioè <code>_PNND0</code> o <code>_MMNUM</code>) è = 0; diversamente, se quest'ultime sono <> 0 avviene un controllo di congruità fra numerico e alfanumerico in fase di PUT. Se il dato alfanumerico corrisponde a un numero di max 6 caratteri numerici, il numero è riportato nella variabile numeriche e la variabile alfanumerica è azzerata.</p> <p>Riferimenti Documenti e Riferimenti Documenti Esterni</p> <p>Nella trasformazione di più documenti sospesi in uno di ordine superiore occorre azzerare la variabile <code>_MMMSI\$(T)=""</code> per <code>_NTMM</code>(righe). Sulle variabili dei riferimenti c'è un controllo nell'inserimento del documento magazzino: se il tipo documento di scarico è uguale al tipo documento, vengono forzati la data/numero/serie documento di scarico e riferimenti esterni documento di scarico; se il tipo documento di scarico di tipo movimento magazzino è diverso dal tipo documento, viene controllata la data di scarico che sia entro i limiti di gestione dell'anno fiscale. Notare: il semplice azzeramento delle variabili <code>_MMO?</code> e <code>_MMM?</code> rende comunque nuova la composizione del documento, seppur inizializzata da GET.</p> <p>Le variabili dei riferimenti esterni sono relazionate alla trasformazione o meno del documento. Se il documento non viene trasformato, questo prende i riferimenti originari dello stesso documento (se stesso) e quindi, in caso di aggiornamento dei dati dei riferimenti esterni, l'utilizzo di variabili inizializzate in trasformazione documento (quelle con array, es. <code>_MMMDARE\$(T)</code>) al posto di quelle senza array (es. <code>_MMDATRE\$</code>), provoca il mancato aggiornamento del dato.</p> <p>GETATTMM non valorizza variabili <code>_MMM?</code> in caso di documento che non sia di tipo movimento.</p>
PUBBLICA AMMINISTRAZIONE	
<code>_MMRIFAMMPA\$</code>	Riferimento amministrazione PA; Valido solo in revisione di un documento o in creazione nuovo documento per le seguenti sigle: "FT", "FD", "NC", "ND", "PA", "NA" (e "BC" dalla v 65.0)
<code>_MMRIFFTPA\$</code>	Riferimento numero fattura collegata PA; (disponibile su documenti FF e NF dalla v 82.1)
<code>_MMRIFDATA\$</code>	Riferimento data fattura collegata PA; (disponibile su documenti FF e NF dalla v 82.1)
<code>_MMTPRIFRE\$</code>	Tipo Identificativo ai riferimenti esterni (FPA): " "=Documento (default), C=Contratto, V=Convenzione;
<code>_MMCTCVRE\$</code>	Codice del documento per tipo identificativo "C" o "V" (FPA);
<code>_MMTPRIFRE\$(T)</code>	Tipo Identificativo ai riferimenti esterni (FPA) derivante da documento origine;
<code>_MMMCTCVRE\$(T)</code>	Identificativo documento origine (FPA) derivante da documento origine;
<code>_MMOTPRIFRE\$(T)</code>	Tipo Identificativo ai riferimenti esterni (FPA) derivante da ordine cliente;
<code>_MMOCTCVRE\$(T)</code>	Identificativo documento origine(FPA) derivante da ordine cliente;
SPESE FUNEBRI	
<code>_MMECSPFU\$</code>	Esclude dalla comunicazione delle spese funebri (S/N); (v. nota)
<code>_MMCFDEF\$</code>	Codice Fiscale del defunto; (v. nota)
	Nota: parametri documenti BD,DL,SL,CL: per documento nuovo sono presi i valori da tabella Parametri Documenti, per il documento già esistente in archivio la PUTMM utilizza le variabili memorizzate in fase di creazione del documento;
<code>_MMPRGMAG\$</code>	Progressivo Magazzino;
<code>_MMPRGAMAG\$</code>	Progressivo A Magazzino;
<code>_MMAGDTMAG\$</code>	Aggiornamento data Magazzino; solo lettura
<code>_MMAGDTAMAG\$</code>	Aggiornamento data A Magazzino; solo lettura
<code>_MMAGGCU\$</code>	Aggiornamento CU -documento CL-;
<code>_MMGCONDOC\$</code>	Clienti/Fornitori -NCF-;
TOTALI DI TESTATA (SOLA LETTURA)	
<code>_MMTOTVALID\$(T)</code>	Variabile di 1 carattere contenente i valori S/N, indica se per la testata sono presenti valori nelle variabili di totale (blank=N);
<code>_MMTOTIVA(T)</code>	Totale IVA;
<code>_MMTOTDOC(T)</code>	Totale documento;
<code>_MMTDAPAGARE(T)</code>	Totale da pagare;
	Nota: non vengono valorizzate da GETATTMM; queste variabili esistono per ogni sottotestata ma viene valorizzata sempre e solo la prima;
RIGHE DEL DOCUMENTO	
<code>_MMTPR\$(R)</code>	Tipo riga articolo D,T,R,r; (= "" cancella riga); (= 'r' è in trasformazione documento letto con GETATTMM);
<code>_MMNTE\$(R)</code>	Riferimenti a testate per documento trasformati;
<code>_MMART\$(R)</code>	Codice articolo -già codificato e leggibile-;
<code>_MMARM\$(R)</code>	Codice articolo proprietario dei progressivi. E' il codice articolo reale in caso di struttura con autocodifica ad elemento precedente.
<code>_MMIDALIAS(R)</code>	Codice numerico dell'Alias adoperato;
<code>_MMTSD\$(R)</code>	Tipo riga di descrizione (.) o (,);
<code>_MMDEE\$(R)</code>	Descrizione riga;
<code>_MMTST\$(R)</code>	Tipo scadenza articolo testo (.) o (,);
<code>_MMARR\$(R)</code>	Codice articolo testo;
<code>_MMDES\$(R)</code>	Descrizione articolo; "" = AUTO; fino a 32 caratteri (in base alla lunghezza del codice articolo) oppure fino a 72 (se viene modificata nel documento);
<code>_MMDESAGG\$(R,1-5)</code>	Array di riga di 5 elementi per gestire descrizioni dettagliate fino a 1000 caratteri (5 elementi da 200 car.) per tutti i documenti.

	<p>La corrispondente descrizione dettagliata di anagrafica articolo è l'array <code>_ARDSE\$(5)</code> di 5 elementi da 200 caratteri (per un totale di 1000 caratteri circa*). Nel Dizionario corrisponde al campo <code>ardse(5)()</code> da 5 elementi da 200 caratteri.</p> <p>Se nel documento la descrizione dettagliata <u>non viene modificata</u>, il dizionario del documento risulta <u>vuoto</u> e corrisponde all'anagrafica [<code>ardse(5)()</code>] (e si deve utilizzare quella anagrafica); se invece viene modificata, il dizionario <u>si valorizza con la modifica</u>: nel documento di tipo non movimento (es. ordine) la modifica è riportata in 4 elementi da 250 caratteri, nel documento di movimento (es. bolla) la modifica è ripartita in 7 elementi da 145 caratteri, cioè <code>mmdes(7)()</code>.</p> <p>ESEMPIO CON DOCUMENTO DI MOVIMENTO:</p> <p>Nota: nella GET gli elementi sono sempre valorizzati, per capire se modificati occorre confrontare con <code>_ARDES\$(1-5)</code></p> <pre>GETMM "BCnum" FOR R=1 TO _NMM LPRINT "_MMDESAGG\$ parte 1:" + _MMDESAGG\$(R,1) LPRINT "_MMDESAGG\$ parte 2:" + _MMDESAGG\$(R,2) LPRINT "_MMDESAGG\$ parte 3:" + _MMDESAGG\$(R,3) LPRINT "_MMDESAGG\$ parte 4:" + _MMDESAGG\$(R,4) LPRINT "_MMDESAGG\$ parte 5:" + _MMDESAGG\$(R,5) NEXT</pre> <p>Nota: nel ciclotabellare gli elementi non modificati non sono valorizzati (si prendono quelli di anagrafica); se gli elementi vengono modificati allora si valorizzano.</p> <p>Nel caso di modifica o nessuna modifica, questa e' sempre valorizzata (ed e' di ANAGRAFICA)</p> <pre>FOR X=1 TO 5 DESORIG\$=ardse(X)(mmart()) : LPRINT DESORIG\$ NEXT</pre> <p>Nel caso di nessuna modifica, questa e' sempre VUOTA Nel caso di modifica, questa e' valorizzata (con la MODIFICA)</p> <pre>FOR Y = 1 TO 7 DESMOD\$=mmdse(Y)() : LPRINT DESMOD\$ NEXT</pre> <p>(*) Attenzione: nel dizionario <code>ardse()</code> gli elementi sono 5, cioè <code>ardse(5)()</code>, ma il 5° elemento, però, sarà da 123 a 147 caratteri a seconda della dimensione del codice articolo (<code>amdco</code>); a seconda di questa dimensione, infatti, la dimensione totale disponibile risulta pari a $(1000 - (80 - amdco) - 5)$, e quindi:</p> <ul style="list-style-type: none"> nel caso di un codice da 8 caratteri la dimensione sarà di $(1000 - (80 - 8) - 5) = 923$ caratteri nel caso di un codice a 32 caratteri la dimensione sarà di $(1000 - (80 - 32) - 5) = 947$ caratteri <p><u>Si rimanda alla tabellina presente nei dati anagrafici di Anagrafica Articolo del Manuale Utente.</u></p>
<code>_MMDESAGGVU\$(R)</code>	Valori ammessi S/N. Quando la variabile vale 'S', allora la descrizione aggiuntiva, viene svuotata.
<code>_MMIDLISTA(R)</code>	Variabile di sola lettura, contiene l'ID della lista di prelievo agganciata alla riga.
<code>_MMCMR(R)</code>	Causale di riga/contropartita -vedere <code>_MMCMO-/Contropartita -max 32 x Documento</code> ; Nota: in caso di omaggio con rivalsa di iva viene rilasciato messaggio "contropartita di riga documento fissa non modificabile"
<code>_MMCAURIGA(R)</code>	Causale di riga.
<code>_MMTIPM(R)</code>	<p>Tipo merce. Variabile numerica per documenti FT, NC, OC, OF, BF, FF, NF, PA (parcellazione da magazzino), MF, MA, PC, PF. I documenti che in base al tipo merce influenzano la scrittura di prima nota, con la creazione di righe diverse in base al tipo merce sono: FT, NC, PA (parcellazione da magazzino), FF, NF.</p> <p>I valori gestiti, per i documenti con contropartita articolo di tipo ricavo sono: 1, 2, 3, 4, 7; per tutti gli altri documenti i numeri ammessi sono: 1, 2, 3, 4, 5, 6, 7, 8.</p> <p>Se 'Tipo merce' è diverso da 0, sono presenti tre casi: se l'azienda non è di provincia RSM il dato inserito viene azzerato senza alcun messaggio a video; se l'azienda è di provincia RSM ed è indicato "Data inizio nuova gestione San Marino", ed il valore non è ammesso per il tipo documento in esame, viene rilasciato errore "Tipo merce xxx non valido in riga (riga yyy)"; se l'azienda è di provincia RSM ed è indicato "Data inizio nuova gestione San Marino", ed il valore non è gestito per il tipo documento in esame, il dato viene azzerato senza alcun messaggio a video.</p> <p>Dizionario: <code>mmtpm, octpm, oftpm, omtpm, optpm, ortpm</code></p>
	DATI CDG (sola lettura con GETATTMM)
<code>_MMDTCANL\$(R)</code>	Data di competenza CDG
<code>_MMCODANL\$(R)</code>	Codice Analitico CDG
<code>_MMCOM\$(R)</code>	Codice Commessa CDG
<code>_MMAREACOM\$(R)</code>	Codice Area CDG
<code>_MMRMAG(R)</code>	Da Magazzino; (v. nota)
<code>_MMRAMA(R)</code>	A Magazzino; (v. nota)
	Nota: se tutti i 'Magaz' di riga sono uguali, sono valorizzati di default dal 'Magaz' di testata; se fra loro differenti sono invece prioritari rispetto a <code>_MMMAG</code> (valorizzato con ultimo <code>_MMRMAG()</code> di riga). Gli 'A Magaz' di riga non seguono questa logica (perché può essere = 0). Anche una sola valorizzazione di riga <code><></code> da 0 obbliga l'utilizzo e la valorizzazione di tutte le <code>_MMRAMA(R)</code> .
<code>_MMTIP\$(R)</code>	Unità misura: "1" o "" = Um1, "2" = Um2, "V" = V.Fisso. La valorizzazione di <code>_MMTIP\$(R)</code> in caso di riga movimento solo a valore (cfr. impostazione di variabile <code>_MMCMO(T)</code>) è pari a "V".
<code>_MMCOL(R)</code>	Numero colli;
<code>MMQTA(R)</code>	Quantità/Peso Lordo;
<code>MMALI\$(R)</code>	Aliquota Iva;
<code>_MMRUBI(R)</code>	Id Ubicazione (magazzino "DA") Il valore numerico dello "id ubicazione" può essere compreso fra valore 0 e 2147483647, diversamente

	viene rilasciato messaggio di errore. Utilizzare -1 per impostare nel movimento il default presente in anagrafica. Se viene impostato un "id ubicazione" non codificato per il magazzino di riga viene rilasciato messaggio di errore "id ubicazione ... non valido per il magazzino ..."
_MMRAUBI(R)	Id Ubicazione (magazzino "A") Il valore numerico dello "id ubicazione" può essere compreso fra valore 0 e 2147483647, diversamente viene rilasciato messaggio di errore. Utilizzare -1 per impostare nel movimento il default presente in anagrafica. Se viene impostato un "id ubicazione" non codificato per il magazzino di riga viene rilasciato messaggio di errore "id ubicazione ... non valido per il magazzino ..." Se il documento non prevede due magazzini, l'informazione di riga di ubicazione del secondo magazzino viene azzerata.
_MMTAGTAB(R)	Tabella Taglie;
_MMTAG(R,16)	Quantità per taglia dalla 1 alla 16;
_MMTAGXL(R,16)	Quantità per taglia dalla 17 alla 32;
_MMPRZ(R)	Prezzo;
_MMKTA(R)	Koeff/Tara sul movimento; solo nella PUTMM se è impostata a -1 = nel movimento si avrà l'azzeramento valore (0), se è impostata 0 = nel movimento si avrà il valore di anagrafica. La GETMM valorizza _MMKTA() con il dato attualmente scritto (es. valore = zero), pertanto, un'eventuale successiva PUTMM valorizzerebbe _MMKTA() con il medesimo dato (cioè zero), salvo che si intervenga impostando la variabile = -1 prima di eseguire la PUTMM. Nella PUTMM il funzionamento della _MMKTA consiste nel prelevare il valore di anagrafica articolo se valorizzata a 0, mentre invece, se si vuole impostare tale valore con 0, deve essere impostata a -1. Questo comporta che la scrittura di un documento dopo la GETMM abbia nel codice il controllo del valore di tale variabile e pertanto, se è 0, va impostata a -1. Questa logica facilita la scrittura diretta di nuovi documenti senza precedenti GETMM, evitando di impostare a -1 tale variabile. La GETMM legge ciò che è scritto su record e quindi, se una riga non ha tara il valore è zero. L'eventuale successivo utilizzo della PUTMM implica che se il valore è zero, il compito di PUTMM è quello di prelevare il valore da anagrafica; se la GETMM non fosse utilizzata per fare una successiva PUTMM, ma bensì servisse solo per estrapolare valori al fine di effettuare statistiche, il valore -1 non sarebbe significativo (quanto uno zero). Teoricamente un articolo a peso netto dovrebbe avere almeno una tara valorizzata e maggiore di zero, ma se questo valore non fosse inserito dall'utente (oppure non reso obbligatorio da un Collage), un modo per conoscere in fase di GETMM se la tara è stata dall'utente azzerata o modificata è quello di verificare prima della PUTMM con un FOR R=1 TO _NMM un IF _MMKTA(R)<>arkoe/arkes(_MMART\$(R)) [arkoe e arkos sono identici], cioè la variazione di condizione rispetto all'anagrafica THEN VAR(R)=1 o VAR(R)=0 e sulla base di questa risultanza, decidere se per la riga documento è opportuno prevedere un ulteriore IF _MMKTA(R)=0 THEN _MMKTA(R)=-1 ELSE _MMKTA(R)=_MMKTA(R), cioè la condizione rispetto a quello che invece deve risultare nel documento. Questi controlli in GET consentono di evitare l'eventuale esigenza di gestione di una nuova variabile (es. che mantenesse in PUT il valore di GET), che renderebbe necessaria la gestione di un'ulteriore variabile per indicare, in fase di PUTMM, quale fra le due sarebbe valida; ciò aumenterebbe la complessità nella gestione dello sprix, cioè, che per gestire l'informazione del coefficiente, richiederebbe almeno tre variabili. Le variabili arkos() e arkoe() possono ritornare o il peso della tara se la seconda unità di misura dell'articolo è "PN" (a peso netto) oppure il coeff. di conversione negli altri casi ritornano attualmente gli stessi valori. In versioni precedenti del gestionale si differenziavano solo per il numero di decimali gestiti. Le istruzioni RIGMM e TOTMM sono sottoposte alle stesse regole della PUTMM, pertanto si può affermare la seguente metodologia per la valorizzazione delle variabili _MMR_??? ed _MMT_??? in presenza di tara ed articoli a PN e doppia unità misura. Le variabili _MMR_QTA, _MMR_TARA valorizzabili tramite RIGMM, e le variabili _MMT_TIMPON, _MMT_TIMPOS valorizzabili tramite TOTMM utilizzano la variabile _MMKTA(R) (array riferito al coefficiente di conversione peso o tara.) con la seguente logica: <ul style="list-style-type: none"> • se questa variabile è letta a zero prima della RIGMM e TOTMM considera quanto impostato in anagrafica articolo; • se questa variabile è impostata a -1 prima della RIGMM e TOTMM considera quanto nella riga del movimento; Questo vale sia per articoli a peso netto e tara che per quelli a doppia unità di misura. Nella riga con articolo a Peso Netto e Tara a zero, prima di RIGMM occorre leggere _MMKTA(N): se = zero va impostata a -1 (per eseguire il calcolo con tara = 0), altrimenti viene preso il dato di anagrafica. Esempio IF _MMKTA(1) = 0 THEN _MMKTA(1) = -1 : RIGMM 1,2,4 :Se viene lasciato invariato il valore, il programma trova zero e lo modifica con il valore presente in anagrafica.
_MMNCAUZ\$(R)	Imballo variabile in emissione;
_MMORD\$(R)	Stato riga d'ordine E,N,S,B. Se non contiene uno di questi valori la scrittura rilascia il messaggio di errore "Tipo riga preventivo/ordine errato (riga numero). In documenti fornitori che non derivano dalla trasformazione, _MMORD\$(R)=" e articoli considerati Evasi; per questi documenti _MMORD\$ è <u>gestita solo in trasformazione</u> , in cui definire quali/quantità articoli devono essere evasi. La variabile, infatti, risulta vuota nei documenti di tipo "BF" creati da zero perché gli articoli sono sempre "evasi" e, analogamente per i documenti di tipo "OF", in quanto non si conosce lo stato in capo al fornitore. La variabile viene, quindi, gestita solo durante la trasformazione dei documenti OF → BF allorché bisogna definire quali articoli (e quanti) devono essere evasi e quali no. Dalla versione 2023B nella trasformazione nativa documento tramite variabile MMSIGTRAS\$ da documento OF a superiore, se lo stato di riga non viene indicato la riga verrà evasa o rimarrà sospesa in base al parametro "proposta evadibilità in trasformazione ordini fornitori" presente nei parametri di magazzino dell'anagrafica azienda. N.B.: nel caso di parametro impostato a "R" la trasformazione tratta tutte le righe come evadibili.
_MMCAN(R)	Categoria annullamento; _MMORD\$="n" / "N";
_MMSCO\$(R)	Sconto/maggiorazione. In Sprix Mobile l'interrogazione dell'archivio Omaggi/Abbuoni/Garanzie avviene con il dizionario gscod(), gsdes(), gsded() e gsdee() o con una tabella basata sull'archivio DESSCO. La variabile _MMSCO\$(R) di riga è gestita diversamente: tra Passmobile e Passepartout è <u>trasferito sempre e solo il valore numerico</u> . A partire dalla versione 2023G è stata introdotta la possibilità di specificare una maggiorazione % aggiungendo il carattere M prima del valore numerico (ad esempio: "M10" applicherà una maggiorazione del 10% sulla riga e non uno sconto). Attenzione: <ol style="list-style-type: none"> 1) la finestra degli sconti di riga, quando la gestione delle maggiorazioni è attiva, contiene un nuovo campo di cui va tenuto conto in eventuali personalizzazioni collage che popolano gli sconti tramite PUTTAST; 2) la variabile dizionario ??sco() ritorna degli asterischi anche in presenza di sconto/maggiorazione singolo/a quando viene specificato uno sconto documento, per questa ragione utilizzare preferibilmente le variabili sconti estese in combinazione alla variabile sconto merce documento.
_MMANNOLETT(R)	Anno di ricevimento della Dichiarazione d'Intento da associare alla riga; numero da 1 (anno 2001) a 99 (anno 2099); 0 = nessuna associazione; - 1 = associazione automatica; (v. nota)
_MMPROGLETT(R)	Progressivo di ricevimento della Dichiarazione d'Intento da associare alla riga; numero che da 1 a 999999; 0 = nessuna associazione; 0 = associazione automatica; (v. nota) Nota: Esempi: associazione della prima riga del documento alla dichiarazione ricevuta nel 2017 numero 57 (vedere data e progressivo della dichiarazione d'intento esportatore abituale contenuta nell'anagrafica

	<p>del conto) dichiarando le seguenti variabili: _MMANNOLETT(1)=17 : _MMPROGLETT(1)=57; nessuna associazione alla prima riga: _MMANNOLETT(1)=0 : _MMPROGLETT(1)=0; associazione automatica alla prima riga: _MMANNOLETT(1)=-1 : _MMPROGLETT(1)=0. Nota: _MMPROGLETT(R) non associa automaticamente al progressivo, poiché è sufficiente (necessario) impostare la sola _MMANNOLETT(R) = -1 per automatizzare l'associazione fra la riga del documento e la prima dichiarazione capiente.</p> <p>Da emissione/revisione documenti, se non si hanno lettere di intento capienti, la procedura si interrompe e viene invitato l'utente (con dei messaggi a video) a decidere cosa fare.</p> <p>Da sprix non c'è interazione, quindi nel caso in cui la lettera non è capiente il documento non viene scritto. Se il programmatore sprix vuole fare un documento senza dichiarazione d'intento deve mettere</p> <p>_MMANNOLETT(r)=0 _MMPROGLETT(r)=0</p> <p>Nota: la gestione della dichiarazione d'intento emessa è di competenza fiscale/F24. Generalmente Sprix non gestisce dati fiscali</p>
_MMMYDBR(R)	<p>Dati MyDB collegati alla riga del documento. Valori accettati:</p> <ul style="list-style-type: none"> -1 → scrive automaticamente un record mydb su tutte le anagrafiche MyDB collegate alla riga, se e' stato specificato un valore iniziale per le estensioni documento o se l'anagrafica e' un'estensione riportabile e sono presenti dati in anagrafica; 0 → non sono presenti dati Mydb; 1 → sono presenti dati Mydb;
LOTTE	
Attenzione: queste variabili si relazionano al livello di gestione dei Lotti (es. base/esteso), valutate eventualmente anche quelle successivamente elencate (_MML).	
_MMGESTLOT\$(R)	<p>Tipo di gestione lotto. Variabile di lettura, con valori strutturati impostati in automatico dalla procedura. Se azzerata, i dati sono nuovamente letti e memorizzati (salvo eventuali errori).</p> <p>1° carattere : Finestra gestione Lotti con richiesta (S), senza richiesta (N), antecedente data inizio (P) 2°/3° carattere : Tipo lotto (artrl, latip, lutpl); se modificati possono generare errori 4° carattere: Tipo selezione (F) FIFO data carico (L) LIFO data carico (C) FIFO data validità (D) LIFO data validità 5° carattere: Gestione matricole (S)/(N)</p>
_MMIDLOT(R,L)	Lotti- Idlotto; -1 autocodifica da carico, -2 autoprelievo da scarico;
_MMLNDQTA(R,L)	Lotti- Numero decimali quantità;
_MMLQTA(R,L)	Lotti- Quantità;
_MMLCOLLI(R,L)	Lotti- Colli;
_MMLQTATGL(R,L,Tg)	Lotti – Quantità per taglia;
LOTTE / MATRICOLE	
Attenzione: queste variabili si relazionano al livello di gestione Lotti (es. matricole), valutate eventualmente anche quelle precedentemente elencate.	
_MML_INIZLOT(R)	Inizio posizione matricole; (v. nota)
_MML_MXQTLOT(R)	Massima quantità di matricole; (v. nota)
_MML_IDLOT(L)	Idlotto; -1 autocodifica da carico, -2 autoprelievo da scarico; (v. nota)
_MML_NDQTA(L)	Numero decimali quantità; (v. nota)
_MML_LQTA(L)	Quantità; (v. nota)
_MML_LCOLLI(L)	Colli; (v. nota)
_MML_LQTATGL(L,32)	Quantità per taglia; (v. nota)
	<p>Nota: Con la gestione dei lotti di tipo 'M' (matricole) e conseguente superamento del limite dei 200 lotti, qualora si adoperino le vecchie variabili _MMIDLOT ecc. comparirà un messaggio di errore in fase di PUTMM del tipo "occorre rivedere la verticalizzazione". Con gestione dei lotti di tipo 'M' occorre adoperare le variabili di tipo _MML_ xxx al posto di _MMLxxx'.</p> <p>Con la gestione dei lotti di tipo 'B/E' (base/estesa), qualora si adoperino le nuove variabili MML_INIZLOT ecc. comparirà un messaggio di errore in fase di PUTMM del tipo "occorre rivedere la verticalizzazione". Con gestione lotti di tipo 'B/E' occorre adoperare le variabili di tipo _MMLxxx al posto di _MML_ xxx'.</p> <p>Superando il numero max 200, in sprix i lotti sono visti come elementi del documento, per esempio, se la prima riga contiene 3 lotti e la seconda riga 4 lotti, sono in totale 7 lotti. Le variabili _MML_INIZLOT e _MML_MXQTLOT sono indicizzate e sono riferite alla riga del documento. Le variabili _MML_IDLOT(L), _MML_NDQTA(L), _MML_LQTA(L), _MML_LCOLLI(L), _MML_LQTATGL(L) sono indicizzate, ma non hanno per indice la riga bensì un indice un progressivo (L) che va da uno (1) alla somma di tutti i lotti inseriti nel documento. Per recuperare i lotti per la riga del documento si utilizza _MML_INIZLOT (posizione di partenza) e _MML_MXQTLOT (quanti lotti sono gestiti per la riga del documento), pertanto, per recuperare i lotti della prima riga si usa l'indice da 1 a 3 (_MML_INIZLOT=1 e _MML_MXQTLOT=3) e della seconda riga l'indice da 4 a 7 (_MML_INIZLOT=4 e _MML_MXQTLOT=4).</p> <p>Le variabili dei lotti non sono gestite in caso di Distinta Base Automatica (DBA) con componenti a lotto. Per conoscere l'id lotto del componente occorre interrogare MEXALDB oppure _SINP\$.</p>
IMBALLI	
_MMCAU(16)	Quantità cauzione resa; quattro caratteri numerico; limite valore 'Diff' fra -32767 e +32767
PROVVIGIONI AGENTI	
_MMPRO(R)	Percentuale provvigione agente – se multiagente è di sola lettura (in tal caso per modificare occorre agire su _MMPROFORM\$()).
_MMPROAGE\$(R,5)	Codice agente; Tipo F/P; se 1° elemento = "-1" azzerata tutto; (variabile multiagenti)
_MMPROCOND(R,5)	Condizione agente; solo 1° elemento; (variabile multiagenti)
_MMPROTP\$(R,5)	Tipo provvigione; "%" (percentuale) oppure "T" (totale); (variabile multiagenti)
_MMPROFORM\$(R,5)	Formula provvigione; valida solo per 1° elemento; (variabile multiagenti)
_MMPROSA(R,5)	Calcolo formula provvigione – sola lettura e se multiagenti -; (variabile multiagenti)
_MMPROQUOTA(R,5)	Quota provvigione ripartita; totale quote = 100%; (variabile multiagenti)
_MMPROMOD\$(R,5)	Modalità provvigione "R" ripartita "P" percentuale; solo 1° elemento; (variabile multiagenti)

_MMPROBIMP\$(R,5)	Base imponibile di calcolo provvigione; "1" lordo, "2" netto; (variabile multiagenti)
ULTERIORI DATI ARTICOLO	
_MMCS2(R)	Categoria Numerica su movimento; in PUT se -1 → 0, se 0 → anagrafica;
_MMCS1\$(R)	Categoria Alfanumerica su movimento; in PUT se " " → "", se "" → anagrafica;
_MMULT(R)	Costo ultimo allo scarico; (v. nota)
_MMSTD(R)	Costo standard allo scarico; (v. nota)
_MMPON(R)	Costo medio ponderato allo scarico; (v. nota)
	Nota: Se in fase di scrittura sono impostate -1 sono assunti come costi (ultimo, standard, medio ponderato) quelli di anagrafica al momento della scrittura, altrimenti viene assunto il valore che viene impostato.
_MMINIRARI\$(R)	Data inizio Rateo Risconto;
_MMFINRARI\$(R)	Data fine Rateo Risconto;
_MMSCA\$(R)	Data scadenza;
_MMSOR\$(R)	Scadenza ordine origine;
_MMCONS\$(R)	Data prevista consegna – documenti sospesi – solo produzione;
_MMCNR\$(R)	Data prevista consegna – documenti sospesi – solo produzione;
_MMPROD\$(R)	Data prevista produzione – documenti di movimento – solo produzione;
_MMPROR\$(R)	Data prevista produzione – documenti di movimento – solo produzione;
_MMSAZOF(R)	Sottoazienda documento ordine fornitore OF collegato, vedi .APRIAZIE;
_MMSOF(R)	Numero serie ordine collegato;
_MMNOF(R)	Numero ordine fornitore collegato; (v. nota)
_MMNCRIGAOF(R)	Numero creazione riga ordine fornitore OF collegato –vedi _MMNCRERIGA(N)-;
_MMTDOCP\$(R)	Sigla documento OF precedentemente collegato alla riga;
_MMSOFP(R)	Serie documento OF precedentemente collegato alla riga;
_MMNOFP(R)	Numero documento OF precedentemente collegato alla riga;
_MMNBL(R)	Numero bolla lavorazione collegata; (v. nota)
_MMRDC(R)	Numero Riferimenti documento collegato; (v. nota)
	Nota: non va modificata; se _MMNOF o _MMNBL è <> da zero, nessuna variabile di quella riga può essere modificata
_MMSTRIGAPV\$(R)	Stato riga preventivo;
_MMPSAZ(R)	Sottoazienda documento preventivo origine, vedi APRIAZIE;
_MMPSI\$(R)	Riferimento sigla preventivo;
_MMPSE(R)	Riferimento serie preventivo;
_MMPNU(R)	Riferimento numero preventivo;
_MMPDA\$(R)	Riferimento data preventivo;
_MMNRECRIGA(R)	Numero record riga (6 caratteri). Mantiene DBA/DBVO (se articolo è con DBP) nelle righe documenti sospesi/movimento; utilizzabile in getmm/putmm (che la valorizza con i relativi numeri record). Non è gestita in trasmissione ASCII-CSV (come DBA/DBVO) e nel Mirroring; nel ciclotabellare si legge con la ??nr(i) dei documenti sospesi. La variabile assumere valori > < di zero (0). _MMNRECRIGA>0 (default): mantiene DBA/DBVO gestite su documenti precedenti; se modificata si perde il collegamento alla stessa riga e non può risalire alla DBA/DBVO da mantenere nel documento registrato con PUTMM. Effetti in base all'operatività: <ul style="list-style-type: none">• GETMM e PUTMM dello stesso documento sospeso: se lasciato invariato il valore ogni riga articolo che gestisce DBA/DBVO manterrà quella del documento originario.• Impostazione variabili _MM e PUTMM di documento sospeso: se forzata la variabile a > 0 è mantenuta quella originaria dell'eventuale documento in archivio relativo alle variabili _MM; se non esiste NON viene generata.• GETMM di documento sospeso e PUTMM di documento di movimento: se lasciato invariato il valore e se il documento di movimento gestisce la DBA, quest'ultimo genera i documenti di scarico e carico relativi alla DBA presenti nel documento sospeso.• Impostazione variabili _MM e PUTMM di documento di movimento: se impostate le variabili relative al documento origine di tipo sospeso e se il documento di movimento gestisce la DBA, quest'ultimo genera i documenti di scarico e carico relativi alla DBA presenti nel documento sospeso _MMNRECRIGA=0 (obsoleto default): non sono gestite DBA/DBVO sui documenti, le righe perdono le DBA/DBVO. Effetti in base all'operatività: <ul style="list-style-type: none">• GETMM e PUTMM dello stesso documento sospeso: se forzata la variabile a 0, ogni riga articolo che gestisce la DBA/DBVO la perde, cancellando quella originaria del documento.• Impostazione variabili _MM e PUTMM di documento sospeso: se forzata la variabile a 0, non viene gestita la DBV/DBVO.• GETMM di documento sospeso e PUTMM di documento di movimento: se impostato il valore della nuova variabile a 0, se il documento di movimento gestisce la DBA, NON genera i documenti di scarico e carico relativi alla DBA.• Impostazione variabili _MM e PUTMM di documento di movimento: se impostate le variabili relative al documento origine di tipo sospeso e se il documento di movimento gestisce la DBA quest'ultimo NON genera i documenti di scarico e carico relativi alla DBA. Questo vale anche se il documento di movimento gestisce la DBA e NON presenta le variabili del documento origine. _MMNRECRIGA<0 (-1): viene letta da anagrafica articoli la relativa DBA. Effetti in base all'operatività: <ul style="list-style-type: none">• GETMM e PUTMM dello stesso documento sospeso: se forzata la variabile a -1, ogni riga articolo che gestisce la DBA/DBVO caricherà quella di anagrafica e cancellerà quella originaria

	<p>del documento.</p> <ul style="list-style-type: none"> • Impostazione variabili _MM e PUTMM di documento sospeso: se forzata la variabile a -1, viene caricata la DBV/DBVO dell'anagrafica. • GETMM di documento sospeso e PUTMM di documento di movimento: se impostato il valore della nuova variabile a -1, se il documento di movimento gestisce la DBA, genera i documenti di scarico e carico relativi alla DBA definita in anagrafica e perde quella del documento sospeso. • Impostazione variabili _MM e PUTMM di documento di movimento: se impostate le variabili relative al documento origine di tipo sospeso e se il documento di movimento gestisce la DBA, quest'ultimo genera i documenti di scarico e carico relativi alla DBA definita in anagrafica, perdendo quella del documento sospeso. Questo vale anche se il documento di movimento gestisce la DBA e NON presenta le variabili del documento origine.
_MMNCRERIGA(R)	<p>Numero di riga univoco nei documenti (dizionario ??rig()) Numero assegnato in fase di creazione riga (di tipo articolo, testo, descrizione). Se la riga viene cancellata il numero viene perso. Se nella riga viene modificato il codice articolo (o il tipo di riga) la procedura assegna un nuovo numero univoco. Le logiche di gestione del dato quando non è modificato (diversamente la procedura controlla sia univoco e lo assegna alla riga se non diversamente specificato) sono le seguenti: Con la generazione di nuovo documento non derivante da sottodocumenti (es. BC 1) la procedura assegna il numero univoco di riga progressivo a partire da 1 (1, 2, 3,...), anche se è imposta la variabile.</p> <p>Es.: → OC di 3 righe (ID=_MMNCRERIGA) → ID riga 1, ID riga 2, ID riga 3 La variazione codice articolo/tipo alla riga 2 provoca rinumerazione: → ID riga 1, ID riga 4, ID riga 3 (la riga 2 è rinumerata)</p> <p>Se il documento è nuovo, ma deriva da sottodocumenti (es. OC1 e OC2 nella BC1) e con aggiunta di nuove righe, la procedura assegna il numero solo alle righe che hanno la _MMNCRERIGA = 0, lasciando quello origine alle restanti. Con la trasformazione di un documento sospeso in un altro sospeso, la riga articolo mantiene lo stesso numero univoco. Con la trasformazione/fusione di più documenti sospesi in altro sospeso, la procedura assegna alle righe articolo un numero univoco progressivo che parte da 1.</p> <p>Esempio fusione 2 documenti (OC 1 + OC2) in OC 3 OC1 + OC2 → OC3 (N.B. le righe vengono rinumerate) ID riga 3 ID riga 1 ID riga 1 ID riga 6 ID riga 4 ID riga 2 ID riga 7 ID riga 6 ID riga 3 ID riga 4 ID riga 5 ID riga 6</p> <p>Con la trasformazione di uno/più documenti sospesi in uno di movimento, il numero di riga univoco delle righe del movimento è mantenuto uguale a quello di riga del documento sospeso origine.</p> <p>Esempio OC 1 → BC1 ID riga 3 ID riga 3 questo vale anche per le trasformazioni ID riga 4 ID riga 4 multiple (es.OC1+OC2 in BC1) ID riga 7 ID riga 7</p> <p>Con evasione parziale di OC (gestione da Emissione-Revisione documenti) ID riga 3: 10 pezzi art A e ID riga 5: 10 pezzi art B l'evasione di 2 pezzi della 1ª riga in BC ha ID 3 (che resta anche per gli 8 pezzi dell'art A sull'OC)</p> <p>Con documenti di movimento derivati da più documenti: il numero di riga è univoco all'interno della sottotestata dei documenti: BC 1 Articolo: A numero di riga univoco: 1 Articolo: B numero di riga univoco: 2 BC 2 Articolo: C numero di riga univoco: 1 Articolo: D numero di riga univoco: 2 FT1 derivata da BC1 + BC2 Articolo: A numero di riga univoco: 1 rif BC1 Articolo: B numero di riga univoco: 2 rif BC1 Articolo: C numero di riga univoco: 1 rif BC2 Articolo: D numero di riga univoco: 2 rif BC2</p>
MMNCRERORIG(R)	Numero riga di creazione d'origine, equivalente ad ??rio()
RINTRACCIABILITÀ'	
MMRINTRCRER(R)	Riferimento prodotto finito PF della bolla di lavorazione BL; solo produzione;
MMRINTRDAT\$(R)	Data documento della bolla di lavorazione BL; solo produzione;
MMRINTRBNU(R)	Numero bolla di lavorazione BL; solo produzione;
MMRINTRBSBL(R)	Sotto bolla di lavorazione BL; solo produzione;
MMRINTRBFS(R)	Fase bolla di lavorazione BL; solo produzione;
MMRINTRBPDR(R)	Numero sottobolla del prodotto finito PF che ha generato semilavorato SML; solo produzione;
MMRINTRPPFD(R)	Numero riferimento prodotto finito PF che ha generato semilavorato SML; solo produzione;
MMRINTRDTP\$(R)	Data sottobolla prodotto finito PF che ha generato semilavorato SML; solo produzione;
MMRINTRSIG\$(R)	Sigla documento da ricercare; solo distinta base automatica DBA + produzione; se "" è bolla di lavorazione BL; (v. note)
MMRINTRSER(R)	Serie documento; solo articolo con distinta base automatica DBA + produzione;
MMRINTRNUM(R)	Numero documento; solo articolo con distinta base automatica DBA + produzione;

_MMRINTRCLI\$(R)	Cliente/fornitore; solo articolo con distinta base automatica DBA + produzione;
_MMRINTRNUMT(R)	Progressivo testata valido se _MMRINTROSIS\$() vuoto Righe riferimento ordine origine del documento da ricercare; Righe solo per articolo con distinta base automatica DBA + produzione;
_MMRINTROSIS\$(R)	Sigla documento ordine origine; articolo distinta base automatica DBA + produzione;
_MMRINTROSE(R)	Serie ordine origine; solo articolo con distinta base automatica DBA + produzione;
_MMRINTRONU(R)	Numero ordine origine; solo articolo con distinta base automatica DBA + produzione;
_MMRINTRODA\$(R)	Data ordine origine bolla di lavorazione BL in ricerca; solo articolo distinta base automatica DBA + produzione;
	<p>Note: per la PUTMM in Rintracciabilità</p> <p>Se la variabile _MMRINTRSIG\$ è vuota vengono azzerate tutte le seguenti variabili: _MMRINTRSER, _MMRINTRNUM, _MMRINTRCLI\$, _MMRINTRNUMT, _MMRINTROSIS\$, _MMRINTROSE, _MMRINTRONU, _MMRINTRODA\$; viene inoltre controllato il contenuto delle seguenti variabili:</p> <ul style="list-style-type: none"> • se (_MMRINTRCRER < 0 OR _MMRINTRCRER > 16777215L) "Rintracciabilità' numero riga articolo errato (deve essere compreso tra 0 e 16777215)" • se _MMRINTRDAT\$ non valida "Data documento BL errata" • se _MMRINTRBNU == 0, _MMRINTRBSBL = 0 e _MMRINTRBFS = 0 • se _MMRINTRBNU < 0 OR _MMRINTRBNU > 65.000 "Numero riferimento bolla di lavorazione errato (deve essere compreso tra 0 e 65.000)" • se _MMRINTRBSBL < 0 OR _MMRINTRBSBL > 19999 "Numero riferimento sottobolla di lavorazione errato (deve essere compreso tra 0 e 19.999)" • se _MMRINTRBFS < 0 OR _MMRINTRBFS > 100 "Numero fase bolla di lavorazione errato (deve essere compreso tra 0 e 100)" • se _MMRINTRBPDR < 0 OR _MMRINTRBPDR > 19999 "Rintracciabilità' numero sottobolla padre errato (deve essere compreso tra 0 e 19999)" • se _MMRINTRPPFD < 0 OR _MMRINTRPPFD > 16777215 "Rintracciabilità' riga PF errato (deve essere compreso tra 0 e 16777215)" • se _MMRINTRDTP\$ non valida "Data documento BL padre errata" <p>Se la variabile _MMRINTRSIG\$ è ammessa vengono azzerate tutte le seguenti variabili: _MMRINTRBNU, _MMRINTRBSBL, _MMRINTRBFS, _MMRINTRBPDR, _MMRINTRPPFD, _MMRINTRDTP\$; viene inoltre controllato il contenuto delle seguenti variabili:</p> <ul style="list-style-type: none"> • se _MMRINTRDAT\$ non valida "Data documento movimento da rintracciare errata" • se _MMRINTRSER == 0, _MMRINTRSER = 1 • se _MMRINTRSER < 0 OR _MMRINTRSER > 999 (nel caso di serie max gestite 999) "Numero di serie documento da rintracciare errato (deve essere compreso tra 1 e 999)" • se _MMRINTRSER < 0 OR _MMRINTRSER > 99 (nel caso di serie max gestite 99) "Numero di serie documento da rintracciare errato (deve essere compreso tra 1 e 99)" • se _MMRINTRNUM < 1 OR _MMRINTRNUM 999999 "Numero documento da rintracciare errato (deve essere compreso tra 1 e 999.999)" • se _MMRINTRCLI\$ non è valido, vengono rilasciati i seguenti possibili messaggi: "riferimento al conto cliente/fornitore da rintracciare errato"; "riferimento al conto cliente/fornitore da rintracciare con gruppo/mastro non definito"; "riferimento al conto cliente/fornitore da rintracciare di tipo non valido"; "riferimento al conto cliente/fornitore da rintracciare non trovato nel piano dei conti" • se _MMRINTRNUMT < 0 OR _MMRINTRNUMT > 250 "Numero di testata documento da rintracciare errato (deve essere compreso tra 1 e 250)" • se _MMRINTROSIS\$ vuoto, _MMRINTROSE = 0, _MMRINTRONU = 0, _MMRINTRODA\$ = 0 • se _MMRINTROSIS\$ non è vuoto: <ul style="list-style-type: none"> ○ se _MMRINTROSE < 0 OR _MMRINTROSE > 999 (nel caso di serie max gestite 999) "Numero di serie ordine da rintracciare errato (deve essere compreso tra 1 e 999)" ○ se _MMRINTROSE < 0 OR _MMRINTROSE > 99 (nel caso di serie max gestite 99) "Numero di serie ordine da rintracciare errato (deve essere compreso tra 1 e 99)" ○ se _MMRINTRNUM < 1 OR _MMRINTRNUM 999999 "Numero ordine da rintracciare errato (deve essere compreso tra 1 e 999.999)" • se _MMRINTRODA\$ non valida "Data ordine da rintracciare errata"
VARIABILI RIGMM	
<p>Notare: le totalizzazioni di riga devono rispettare le regole previste per la scrittura PUTMM e il controllo va comunque eseguito con _ERRMM e _ERRMM\$, poiché per congruità dell'operazione si dovrebbero considerare anche eventuali archivi coinvolti (lotti, documentale, cdg, ecc..).</p>	
MMR_QTA	Quantità/peso lordo; vedere ulteriori spiegazioni in variabile _MMKTA(R);
MMR_IMPON	Imponibile (è l'equivalente di quello che si vede nella colonna importo di riga di emissione/revisione)

	documenti);
_MMR_PREZZO	Prezzo;
_MMR_COLLI	Colli;
_MMR_TARA	Tara; vedere ulteriori spiegazioni in variabile _MMKTA(R);
_MMR_SCONTO	Valore sconto;
_MMR_OMAnnn	Merce omaggio tipo nnn=100, 110, 120, 130, 200;
_MMR_PROVV	Importo provvigione;
_MMR_PROVSA(15)	Totale provvigione per singolo agente;
_MMR_PROAGE\$(15)	Codice agente provvigione;
_MMR_IMPNT	Importo netto di riga (è l'importo reale di riga, se non è presente lo sconto merce sul pagamento _MMR_IMPON è uguale a _MMR_IMPNT, altrimenti _MMR_IMPNT contiene l'importo di _MMR_IMPON decurtato dello sconto merce sul pagamento);
_MMR_PSMER(9)	Percentuale sconto sulla riga (valido per gli sconti a cascata, a valore, a percentuale, con sconto >100 e < 109, con sconto >130 e < 139, con sconto >200 e < 209);
_MMR_VSMER(9)	Valore dello sconto;
_MMR_PSMER\$(9)	Variabile di 1 carattere: vale "M" in caso di maggiorazione, "" in caso di sconto;
_MMR_PSMEP	Percentuale sconto pagamento merce;
_MMR_VSMEP	Valore dello sconto pagamento merce;
PIEDE DEL DOCUMENTO	
RIFERIMENTI AL TRASPORTO	
_MMSPE\$(T)	Codice indirizzo spedizione;
_MMCDMYDB\$(T)	Gestione MYDB da codice indirizzo di spedizione. Accetta valore "S" o "N". Se _MMCDMYDB\$="S" indica che i dati MYDB sono quelli del destinatario (e non dell'intestatario). Se _MMCDMYDB\$="S" e _MMSPE\$="" (vuota) la PUTMM rilascia l'errore "non e' possibile gestire i dati aggiuntivi MyDB del destinatario se il destinatario non e' indicato.
_MMANSPE\$(T)	Codice da anagrafica indirizzo spedizione;
_MMTSR\$(T)	Trasporto a mezzo; solo per documenti di movimento; nelle versioni obsolete del gestionale il dato non veniva memorizzato nei documenti non movimentanti il magazzino (ordini, preventivi, matrici). Per ripristinare questo comportamento nelle versioni più aggiornate occorre modificare il parametro fgGest nella sezione installazione del pxconf.ini
_MMTRACKNUM\$(T)	Track number (tracciabilità o tracking); codice alfanumerico identificativo della spedizione (con vettore); 36 caratteri memorizzati in documenti di movimento; la variabile non è elemento di rottura; non disponibile nella modalità compatibilità Collage.
_MMVET\$(T)	Codice vettore;
_MMCOMM\$(T)	Committente;
_MMCAR\$(T)	Caricatore;
_MMPROP\$(T)	Proprietario;
_MMPOR\$(T)	Tipo porto;
_MMINCOT\$(T)	Codice Incoterm;
_MMDSP\$(T)	Ulteriore descrizione porto;
_MMTPS\$(T)	Tipo spese di spedizione;
_MMVAS(T)	Valore spese di spedizione;
_MMACOS	Calcolo automatico colli S/N;
_MMTCO(T)	Totale colli;
_MMAPE\$	Calcolo automatico peso S/N;
_MMTPE(T)	Totale peso;
_MMASP(T)	Tabella aspetto esteriore beni;
_MMDTINTR\$(T)	Data inizio trasporto;
_MMORINTR\$(T)	Ora inizio trasporto;
_MMSPTRASP(T)	Totale spese trasporto; sola lettura; (v. nota)
_MMTCAUVEN(T)	Totale cauzioni a vendere; sola lettura; (v. nota)
	nota: le variabili sono valorizzate solamente nella prima testata del documento di magazzino, ad esempio nell'unione OC1+OC2 in BC1, i totali sono indicati solamente nella prima testata della BC, ovvero nella prima testata del documento di scarico; di conseguenza, le ulteriori unioni riportano i riferimenti nelle prime rispettive testate, ad esempio unione OC1+OC2->BC1 e l'unione OC3+OC4->BC2 e poi unione BC1+BC2->FT nelle 4 testate della FT i totali sono nella testata 1 e 3, poichè separati per documento di scarico, cioè nella prima testata i totali della BC1 e nella terza testata i totali della BC2.
APPALTI PUBBLICI	
_MMCIG\$	Codice CIG;
_MMCUP\$	Codice CUP;
_MMNCTRIMPR\$	Iva MOSS Nota credito riferita ad operazioni trimestri precedenti;
_MMSPLITPAY\$	Split Payment, assume i seguenti valori: = 'S' Effettua sempre Split Payment. Significa che in creazione del documento il cliente ha le seguenti condizioni ENTE PUBBLICO = S e ENTE ESCLUDI SPLIT= N; = 's' Effettua sempre Split Payment. Significa che in creazione del documento il cliente ha le seguenti condizioni ENTE PUBBLICO = S e ENTE ESCLUDI SPLIT= S; = 'N' Non effettua mai Split Payment. Significa che in creazione del documento il cliente ha le seguenti condizioni ENTE PUBBLICO = S e ENTE ESCLUDI SPLIT= N; = 'n' Non effettua mai Split Payment. Significa che in creazione del documento il cliente ha le seguenti condizioni ENTE PUBBLICO = S e ENTE ESCLUDI SPLIT= S; = 'E' Effettua Split Payment in automatico e significa che in creazione del documento il cliente ha le seguenti condizioni ENTE PUBBLICO = S e ENTE ESCLUDI SPLIT= N; = 'e' Non effettua Split Payment e significa che in creazione del documento il cliente ha le seguenti condizioni ENTE PUBBLICO = S e ENTE ESCLUDI SPLIT= S; = '' Se si sta scrivendo un nuovo documento, verifica le condizioni del cliente ed applica se è possibile Split Payment. Se si sta revisionando un documento, non esegue nessuna operazione; = 'L' Valido solo in revisione di un documento esegue la lettura dei dati del cliente e se possibile viene

	<p>applicato lo Split Payment; ='A' esegui calcolo con condizione attuale documento e Ente Pubblico='N' e 'Altri soggetti Split Payment='S'. ='a' esegui sempre calcolo con cliente Ente Pubblico='N' e 'Altri soggetti split payment='S' ='Z' non eseguire il calcolo con cliente Ente Pubblico= N e 'Altri soggetti split payment='S'</p> <p>Esempio: per il Cliente che non sia ente pubblico e con flag = "S", nelle "Condizioni e automatismi" della sua anagrafica si possono impostare 3 intervalli di tempo; nel caso della emissione di un documento di magazzino con data compresa in uno dei suddetti intervalli viene eseguito lo Split Payment; se, invece, la data del documento non rientra e avviene una forzatura nel piede del documento, il valore che viene assunto è = "a" e viene eseguito comunque lo Split Payment; nei restanti casi il valore è "Z". Per questi tre casi la lettura della variabile eseguita da Sprix riporta il valore "A" o "a" o "Z".</p>
PAGAMENTI	
_MMPAG	Codice pagamento; Nota: la configurazione nel tipo di pagamento di una contropartita contabile diversa da quella del conto automatico, nella contabilizzazione dei corrispettivi può essere considerato quanto impostato nelle variabili di Acconto e di Abbuono (_MMCTRACC\$ e _MMCTRABB) (v. nota)
_MMDECOR\$	Data decorrenza pagamento; Nota: fino a che le tipologie di pagamento non consentono di stabilire l'emissione di documento con scadenza a una data fissa a fine mese , un modo per ottenere questa funzionalità consiste nell'operare con Collage controllando/modificando il valore di questo campo.
_MMBAPP	Numero banca di appoggio; -F5-
_MMCBZPAG\$	Codice Conto "Banca Azienda" del Piano dei Conti; -F5- (dalla v711)
	<p>Variabile per codificare/ricchiama una "Banca azienda" in documento note, parcelle, fatture, note di credito, ricevute fiscali, ddt da "Magazzino/Parcella - Emissione/revisione documenti".</p> <p>La variabile ha il corrispondente dizionario "opcbz", "omcbz", "occbz", "mmcbz". La variabile non è gestita nei documenti passivi "OF". La variabile è gestita in GETMM, GETATTMM, PUTMM; in caso di codice conto non valido, sono ritornati tre casi di messaggio bloccante:</p> <ol style="list-style-type: none"> 1. caso Conto Banca che non esiste nel Piano dei Conti. Messaggio: "Conto Banca non presente in Rubrica"; 2. caso Conto Banca che esiste nel Piano dei Conti ma non è di tipo classe conto Banca. Messaggio: "Conto Banca non classificato Banca"; 3. caso Conto Banca che esiste nel Piano dei Conti, è di tipo classe conto Banca, ma non collegato a Banca. Messaggio: "Conto Banca non collegato a Banca";
_MMCODRID	Codice mandato RID SEPA;
_MMACC	Acconto/pagato;
	Attenzione: non gestibile in presenza di codici pagamento (_MMPAG) con automatismi (es. "con arrotondamento automatico", "paga totale documento = S / A", ecc.); in caso si volessero gestire con altri importi, occorre utilizzare codici pagamento senza automatismi.
_MMABB	Abbuono;
	Attenzione: non gestibile in presenza di codici pagamento (_MMPAG) con automatismi (es. "con arrotondamento automatico", "paga totale documento = S / A", ecc.); in caso si volessero gestire con altri importi, occorre utilizzare codici pagamento senza automatismi.
_MMCTRACC\$	Contropartita Acconto; (v. nota)
_MMCTRABB\$	Contropartita Abbuono; (v. nota)
	<p>Nota: Per definire contropartite diverse da quelle dei conti automatici per ogni documento; se impostate a "-1" vengono prelevate le contropartite relative al codice pagamento del documento (se è impostato, altrimenti, se impostate a "-1" ma manca il codice pagamento sono prese quelle dei conti automatici).</p> <p>Nonostante _MMCTRABB\$="-1" la fattura emessa FT prende comunque l'abbuono passivo, perché l'abbuono attivo è considerato solo in caso di conto fornitore (e quindi di fattura fornitore). Per ottenere l'abbuono passivo è necessario forzare _MMCTRABB\$ col conto letto dalla tabella pagamenti.</p>
_MMALISPBAN\$	Aliquota Iva spese banca. Nota: gli aggiornamenti di archivi provenienti dalla versione 2016A alla 2016I, in fase di lettura/scrittura del documento di magazzino possono provocare un messaggio di aliquota iva spese banca errata, poiché viene scritta nel formato " 00,0" anziché "". Occorre aggiungere il seguente test prima della scrittura (dalla versione 2016J non è più necessario): IF _MMALISPBAN\$= " 00,0" THEN _MMALISPBAN\$="" oppure utilizzare il pulsante "Ricalcola spese" sul piede del documento.
_MMSPTRASP()	Totale spese trasporto; sola lettura
_MMTCAUVEN()	Totale imballi a vendere; sola lettura; se impostato correttamente la dichiarazione d'intenti nel cliente e la corretta aliquota Iva marcata Plafond nella tabella degli imballi
_MMSPBANCA	Valore spese banca; (v. nota)
_MMSPBOLLI	Valore spese bolli; (v. nota)
	<p>Nota: La PUTMM di documenti nuovi calcola le spese banca e bolli in base a quanto stabilito dal codice pagamento impostato e valori di tabella. Nella revisione l'aliquota e le spese sono modificabili; dalla versione v50.1 nella variazione di documento già presente, immesso un valore questo viene aggiornato, ma occorre aggiornare la primanota: se _MMSPBANCA viene impostato -1 allora sia l'aliquota che il valore delle spese banca sono impostate con i valori presenti in tabella; se a variabile _MMSPBOLLI viene impostato -1 allora il valore delle spese bolli viene impostato con il valore presente in tabella. Se le variabili sono impostate con valore -2 la relativa spesa (banca/bollo) è azzerata (sia in caso di documento nuovo che trasformato)</p>
BONUS EDILIZIO (gestione attivata da parametri di magazzino)	
_MMPERCBOEDI	variabile numerica di testata relativa alla % per il bonus edilizi; la percentuale deve essere compresa tra 0 < % < 100; dizionario "mmpbe"; se <> da 0, viene utilizzata la % per il calcolo abbuono, piuttosto che il valore passato in _MMABB; (v. nota)
_MMSUPBOEDI\$	variabile stringa di testata per indicare l'utilizzo del superbonus 110%; valori "S" o "N"; dizionario "mmsbe"; (v. nota)
_MMUTOBOEDI\$	variabile stringa di testata per indicare se l'abbuono nel documento è di tipo bonus edilizio; valori "S" o "N"; dizionario "mmtbe"; (v. nota)
_MMTIPBOEDI\$	Tipo bonus edilizio. Se la variabile non viene valorizzata e viene gestito il bonus, viene assegnato il valore di default "Ecobonus".
	<p>Nota:</p> <ul style="list-style-type: none"> • solo documenti FT/FD/FS/NC/ND/NS, altrimenti le variabili vengono azzerate • i documenti con l'indicazione del bonus edilizio non calcolano la RDA (ritenuta d'acconto), pertanto vengono azzerati aliquota e % imponibile RDA

	<ul style="list-style-type: none"> • se il documento di magazzino non esiste ed il parametro di magazzino "gestione bonus edilizio" è spento, le 3 variabili vengono azzerate dalla PUTMM • se l'abbuono = 0 e la % di bonus = 0 e si è indicato che è un abbuono di tipo edilizio, viene rilasciato l'errore "non si può indicare un abbuono di tipo bonus edilizio se abbuono uguale a 0 e percentuale bonus edilizio uguale a 0"; • se il pagamento prevede abbuono con arrotondamento automatico oppure il pagamento è con sconto di tipo abbuono, vengono rilasciati gli errori: "Pagamento con arrotondamento automatico non ammesso con Bonus edilizi"; "Pagamento con abbuono non ammesso con Bonus edilizi"; • se l'importo dell'abbuono inserito nel Bonus Edilizio è > l'importo totale del documento da pagare, viene rilasciato l'errore "In caso di bonus edilizi, non ammesso abbuono maggiore del totale documento da pagare"
DATI INTRACEE	
_MMNELINT\$	Escludi elaborazione Intracee;
_MMNTR\$	Natura transazione Intracee (tabella A);
_MMNBTR\$	Natura transazione Intracee (tabella B);
_MMRGM\$	Condizioni consegna Intracee;
_MMMTR\$	Modalità trasporto Intracee;
_MMVAOV	Ammontare valuta; (v. nota)
_MMCAOV	Cambio ammontare in valuta; (v. nota)
	Nota: Se _MMCAOV = 0 e _MMVAOV < > 0 è prelevato il cambio relativo alla data ingresso terminale.
_MMSTPIFORN\$	Stato Partita Iva Fornitore autofattura, documento -FF/NF-;
_MMCDPIFORN\$	Codice Partita Iva Fornitore autofattura, doc. -FF/NF-;
_MMRIFSERI	Serie documento fattura riferimento Intracee -NF/NC-;
_MMRIFNUMI	Numero documento fattura riferimento Intracee -NF/NC-;
_MMRIFNUI\$	Numero Alfanumerico documento fattura riferimenti Intracee -NF/NC-;
_MMRIFDTI\$	Data documento fattura riferimento Intracee -NF/NC-;
_MMPROMES\$	Provincia merce Intracee;
_MMPAEMES\$	Paese origine merce Intracee;
_MMTOCOS\$	Tipo operazione Call of stock
ENASARCO	
_MMCENAS	Contributo enasarco con spese banca o bolli; se=-2 viene ricalcolato; N.B.: in Mexal su aziende con gestione enasarco attiva, il dato viene sempre sostituito dal calcolo effettuato dal gestionale.
_MMDCCECA\$	Data di competenza provvigioni Enasarco;
RITENUTA D'ACCONTO - INPS GESTIONE SEPARATA - CASSA PREVIDENZA	
(Notare: non agiscono retroattivamente su documenti pregressi registrati prima della versione v680 -2019C) (R) = 1 TO _NMM (indica se la Riga è soggetta al calcolo della Rda, della Gestione Separata Inps, della/delle Cassa/Casse Previdenza)	
_MMSTUDASS\$	Flag di Studio Associato, valori ammessi: Si = S , No = qualsiasi altro carattere;
_MMRIT\$(R)	Flag di Riga soggetta a Ritenuta d'Acconto (RDA), valore "S" o "N";
_MMALQRDA	Aliquota Rda. Valori ammessi 0,00 - 99,99
_MMQIMPRDA	Percentuale Imponibile Rda. Valore 0-100
_MMSPBANRDA\$	Flag Assoggettamento Spese Banca Rda; valori ammessi: Si = S , No = qualsiasi altro carattere;
_MMCAUPRDA	Causale Pagamento Rda. Valori ammessi 0-99
_MMINPSS\$(R)	Flag di Riga soggetta a Gestione Separata Inps, valore "S" o "N";
_MMALIINPSS\$	Aliquota IVA Inps;
_MMPERINPS	% Gestione separata Inps;
_MMTPCAINPSS\$	Tipo Cassa Inps Xml - valori TC01-TC22;
_MMCAPRO\$(R)	Flag di Riga soggetta a Cassa Previdenza, valore "S" o "N";
_MMALICAPRO\$	Aliquota IVA Cassa Previdenza - Azienda professionista - (aliquota in GET/PUT; salvata nel documento di ultimo livello i documenti emessi precedentemente usano aliquota di tabella, quelli con data < 15-9-2011 usano il 20%; in revisione se è impostata = 0 legge la tabella);
_MMPERCAPRO	Percentuale Cassa Previdenza - Azienda professionista - (variabile di testata esistente prima della v680);
_MMPERCAPRO2	Percentuale Cassa Previdenza; valore 0,00-100,00
_MMPERCAPRO3	Percentuale Cassa Previdenza; valore 0,00-100,00
_MMQUCAPRO1	Percentuale Cassa Previdenza; valore 0,00-100,00
_MMQUCAPRO2	Percentuale Cassa Previdenza; valore 0,00-100,00
_MMQUCAPRO3	Percentuale Cassa Previdenza; valore 0,00-100,00
_MMTPCAPRO\$	Tipo Cassa Previdenza - valori TC01-TC22;
_MMTPCAPRO2\$	Tipo Cassa Previdenza - valori TC01-TC22;
_MMTPCAPRO3\$	Tipo Cassa Previdenza - valori TC01-TC22;
_MMCALCRDA	Metodo di calcolo (RDA, INPS, CASSA). Variabili di sola scrittura, accettano valori 0, -1, -2, cioè metodi diversi a seconda che il documento sia nuovo oppure in revisione :
_MMCALCINPS	
_MMCALCCAPRO	
	<ul style="list-style-type: none"> • (0) calcoli con valori letti da parametri tabella (nuovo) o da disco/variabili sprix (revisione); • (-1) calcoli con valori letti da parametri tabella (nuovo o revisione); • (-2) calcoli con valori letti da variabili sprix (nuovo o revisione);
ENPAM / ENPALS	
_MMENPAM\$	Articolo soggetto a gestione ENPAM; valori N,R,S; (getmm, getattmm, putmm);
_MMPENPAM	Percentuale contributo ENPAM; valori >=0 e <=99.999; (getmm, getattmm, putmm);
_MMCENPAM	Importo contributo ENPAM; valore >=0; (getmm, getattmm, putmm);
_MMRENAPAM	Ritenute sindacali ENPAM; valore >=0; (getmm, getattmm, putmm);
_MMMENPAM\$	Imp. ENPAM modificato manualmente; flag S/N; (getmm, getattmm, putmm);
_MMCAENPAM	Modo calcolo ENPAM; valori -1, -2; (putmm); (v. nota)
_MMENPALS\$	Articolo soggetto a gestione ENPALS; valori ammessi S e !=S; (getmm, getattmm, putmm);
_MMPENPALS	Percentuale contributo ENPALS; valori >=0 e <=99.999; (getmm, getattmm, putmm);
_MMCENPALS	Importo contributo ENPALS; valore >=0; (getmm, getattmm, putmm);

_MMSENALS	Importo solidarieta' ENPALS; valore >=0; (getmm, getattmm, putmm);
_MMIENALS	Importo integrativo ENPALS; valore >=0; (getmm, getattmm, putmm);
_MMMENALS\$	Imp. ENPALS modificato manualmente; flag S/N; (getmm, getattmm, putmm);
_MMCAENALS	Modo calcolo ENPALS; valori -1, -2; (putmm); (v. nota)
	<p>Nota: MMCAENPAM e MMCAENPALS Sono variabili di tipo numerico, di sola scrittura e non memorizzate su disco. Se il documento esiste e non vengono valorizzate, queste variabili vengono impostate di default a -2 (calcolo manuale). Se, invece, il documento NON esiste, il default sarà a -1 (calcolo automatico). Il loro significato è quello di:</p> <ul style="list-style-type: none"> = -1 viene forzato il calcolo in automatico. Da impostare se dopo una GETMM si vuole fare una PUTMM e forzare il calcolo in modalità AUTO, leggendo cioè i parametri di calcolo prelevati da parametri aziendali ed ignorando quello che si era letto durante la GETMM; = -2 viene forzato il calcolo in manuale. Da utilizzare se si vogliono forzare manualmente dei valori. Se _MMMENPAM\$ / _MMMENPALS ha valore: <ul style="list-style-type: none"> 'N' viene rifatto il calcolo del tributo utilizzando la percentuale di _MMPENPAL / _MMPENPAM e ignorando quello di _MMPENPAL / _MMCENPAM; 'S' vengono salvati i valori di _MMPENPAL / _MMPENPAM e di _MMPENPAL / _MMCENPAM senza effettuare il calcolo;
VARIABILI TOTMM	
<p>Notare: le totalizzazioni documento devono rispettare le regole previste per la scrittura PUTMM e il controllo va comunque eseguito con _ERRMM e _ERRMM\$, poichè per congruità dell'operazione si dovrebbero considerare anche eventuali archivi coinvolti (lotti, documentale, cdg, ecc..).</p>	
_MMT_IMPENAS	Imponibile soggetto a enasarco;
_MMT_CENAS	Contributo enasarco;
_MMT_PENAS	Percentuale enasarco a carico azienda; (obsoleta e non funzionante)
_MMT_TIMPON	Totale imponibile; vedere ulteriori spiegazioni in variabile _MMKTA(R);
_MMT_TIMPOS	Totale imposta; vedere ulteriori spiegazioni in variabile _MMKTA(R);
_MMT_PCALIQ\$(6)	Aliquota iva;
_MMT_PNALIQ(6)	Numero iva; solo aliquota numero;
_MMT_PIMPON(6)	Imponibile; somma per aliquota Iva;
_MMT_PIMPOS(6)	Imposta; somma per aliquota Iva;
_MMT_SPBANCA	Spese banca;
_MMT_SPBOLLI	Spese bolli;
_MMT_SPTRASP	Spese trasporto;
_MMT_SPESE	Spese; articoli tipo S;
_MMT_CAUVEN	Cauzioni vendute;
_MMT_MERCE	Valore merce;
_MMT_SCONTI	Valore sconti;
_MMT_OMAnn	Merce omaggio tipo nnn=100, 110, 120, 130, 200;
_MMT_PROVVI	Importo provvigione;
_MMT_PROVSA(15)	Totale importo provvigione singolo agente;
_MMT_PROAGE\$(15)	Codice agente provvigione;
_MMT_COLLII	Colli; (v. nota)
_MMT_PESO	Peso; (v. nota)
	<p>Nota: Se all'interno del documento si indicano colli e pesi in modalità 'A' (automatica) le variabili _MMT_COLLII e _MMT_PESO vengono valorizzate correttamente; se nel tipo di documento vengono inserite manualmente colli e pesi occorre adoperare al loro posto le variabili _MMTCO(T) e _MMTPE(T).</p> <p>Le variabili sono degli array, perciò se viene realizzato un raggruppamento di documenti è necessario ciclare all'interno di tutti i documenti (la variabile che indica il "Numero testate DOCUMENTI DI MAGAZZINO" è _NTMM) al fine di trovare i vari totali e di sommarli tutti assieme per trovare il valore complessivo, altrimenti se il documento è unico si può leggere il primo valore dell'array.</p>
_MMT_CAUVCON(32)	Cauzioni consegnate;
_MMT_CAUVVAL	Valore cauzioni; consegnate-rese;
_MMT_ACCONTO	Acconto;
_MMT_PSCVAL	Percentuale sconto in valuta;
_MMT_VSCVAL	Valore sconto in valuta;
_MMT_PSCMER	Percentuale sconto merce. Nota bene in seguito all'introduzione della variabile _MMSCMERCE: TOTMM riporta come in precedenza la percentuale dello sconto nel caso di codice pagamento 'M'; non è stato gestito il caso di tipo 'D' in quanto la percentuale sconto si trova nella testata del documento.
_MMT_VSCMER	Valore sconto merce (contiene il valore risultante dallo sconto merce riferito al pagamento);
_MMT_IMPRIT	Imponibile ritenuta d'acconto;
_MMT_RITAC	Importo ritenuta d'acconto;
_MMT_IMPCPR	Imponibile cassa professionisti;
_MMT_CAPRO	Importo cassa professionisti;
_MMT_TSPPAY	Imposta Split Payment;
_MMT_ENPAM	Importo totale ENPAM;
_MMT_IENPAM	Imponibile ENPAM;
_MMT_ENPALS	Importo totale ENPALS;
_MMT_IENPALS	Imponibile ENPALS;
_MMT_IMPCPR1	Imponibile cassa previdenza 1
_MMT_IMPCPR2	Imponibile cassa previdenza 2
_MMT_IMPCPR3	Imponibile cassa previdenza 3
_MMT_PBOEDI	Variabile numerica che indica la percentuale del bonus edilizio
_MMT_SUPABB	Variabile numerica che indica l'abbuono reale in caso di bonus 110% (es. 50.000x100%x110%=55.000)
_MMT_SUPBOE\$	Variabile stringa S/N che indica se il tipo di bonus è 110%

_MMT_UBOEDIS\$	Variabile stringa S/N che indica se il tipo di abbuono è di tipo bonus edilizio
_MMT_TBOEDIS\$	Tipo bonus edilizio
VIDEATE UTENTE	
_MMVDT\$(T,26)	26 campi videata di testata; nella fusione di documenti (es. OC1+2 in un unico OC) la videata dei dati aggiuntivi di testata del secondo documento viene persa (non è mantenuta dalla procedura standard, poiché la semplice fusione di due documenti accorpa i dati aggiuntivi, conserva quelli del primo documento _MMVDT\$(1,1) e perde quelli del secondo _MMVDT\$(2,1). La variabile è, però, relativa alle trasformazioni per fusione, cioè per il documento originato (es. BC) mantiene le videate poiché la sua testata punta a testate di documenti (gli OC), che restano concettualmente separate, in quanto "fuse" allo solo scopo di essere trasformate in un altro (diverso) documento.
_MMVDR\$(R,26)	26 campi videata di riga (max 627 articoli); R = riga del corpo, 1-26 numero campo videate; se si superano i limiti dell'indice dell'array occorre spezzare il documento; la GET converte da numero → stringa, la PUT viceversa;
ARRAY PER STAMPA DEL MODULO DOCUMENTO	
(nota: per approfondimenti consultare il capitolo STAMPE nella sezione Comandi per Argomento). Clicca qui	
_MMSXN(1-20)	Variabile Numerica di testata/piede;
_MMSXA\$(1-20)	Variabile Alfanumeri di testata/piede;
_MMSWN(R,1-10)	Variabile Numerica di riga; Array bidimensionale dove l'ultima riga utile R è pari alla variabile _NMM
_MMSWA\$(R,1-10)	Variabile Alfanumerica di riga; Array bidimensionale dove l'ultima riga utile R è pari alla variabile _NMM
_MMSWF(R,1-20)	Variabile Numerica di riga; Array bidimensionale dove l'ultima riga utile R è pari alla variabile _NMM
_MMSWS\$(R,1-20)	Variabile Alfanumerica di riga; Array bidimensionale dove l'ultima riga utile R è pari alla variabile _NMM
RISTAMPA DOCUMENTO	
_MMRICSTP	Variabile per la ristampa. Consente di attivare la richiesta di selezione stampante (se = 1), non attivare (se = 0, default) o stampare direttamente (se = -1); non è consentito indicare il numero di stampante (come da configurazione) e non è consentita in Shaker.
TRASFORMAZIONE NATIVA DOCUMENTO	
_MMSIGTRASS\$	Sigla documento trasformato.
_MMGESIMP\$	Nel caso di trasformazione di documenti che gestiscono in modo diverso l'iva sulle righe (ad esempio OX - > BC), la variabile determina la modalità di calcolo dell'importo di riga del documento trasformato. Accetta valori S/N e quando impostata a S effettua il calcolo con gli stessi automatismi di emissione/revisione documenti. Nota bene: il valore di default della variabile quando non espressamente indicato è N.
_MMCOLR(R)	Colli di riga.
_MMQTAR(R)	Quantità di riga.
_MMTAGR(R,T)	Quantità taglie di riga.
_MMLCOLLIR(R,L)	Colli di riga lotto (valida per gestione lotti base/estesa).
_MMLQTAR(R,L)	Quantità lotto (valida per gestione lotti base/estesa).
_MMLQTATGR(R,L,T)	Quantità taglie lotto (valida per gestione lotti base/estesa).
_MML_LCOLLIR(L)	Colli di riga lotto (valida per gestione lotti a matricole).
_MML_LQTAR(L)	Quantità lotto (valida per gestione lotti a matricole).
_MML_LQTATGR(L,T)	Quantità taglie lotto (valida per gestione lotti a matricole).
_MMUBISURES\$(R)	Aggiorna ubicazione articolo su documento d'origine: variabile flag S/N che specifica se in caso di modifica di ubicazione per l'articolo trasformato deve venire aggiornato anche l'ubicazione sul documento d'origine.
GESTIONE di 999 documenti (solo GETMM e GETMM_EXT)	
_MMRICDOC\$()	Variabile vettore che consente la lettura fino a 999 documenti alle funzioni GETMM e GETMM_EXT.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

MOD - Import Moduli

MOD

Variabili delle Istruzioni di Import Moduli (MOD)	
_MODTOTRIG	Totale righe file modulo di import;
_MODNUMRIG	Numero della riga letta con la MODGET, dopo la MODOPEN;
_MODFINEP	Flag di fine pagina, 1 = la riga letta è l'ultima della pagina;
_MODFINEC	Flag di fine corpo pagina, se gestito, 1 = riga letta è l'ultima riga della pagina;
_MODNUMRIGP	Numero della riga di pagina letta con la MODGET;
_MODNUMRIGC	Numero della riga del corpo pagina letta con la MODGET;
_MODNUMRIGLC	Numero della riga logica del corpo pagina letta con la MODGET;
_MODSCRIVI	Variabile che indica, quando impostato a 1, la necessità di scrittura dei dati, dovuta a fine pagina o fine corpo, oppure, se definito, per rottura dei valori dei campi selezionati;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

MODGRA - Moduli Stampa Sprix

MODGRA

Variabili dell'Istruzione di elenco moduli stampa Sprix (GETMODSTPGRAF)	
_MODGRA_NUM	Numero di modulistiche grafiche trovate
_MODGRA_COD\$()	I codici di tutte le modulistiche trovate
_MODGRA_DES\$()	Le descrizioni di tutte le modulistiche trovate
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_MSC - Saldi Contabili di Import Moduli

_MSC

Variabili di selezione campi (Saldi Contabili) di Import Moduli	
_MSCCOD\$	Codice conto;
_MSCDESC\$	Descrizione conto;
_MSCINIZ	Saldo iniziale;
_MSCSEZI\$	Sezione saldo iniziale (_MSCINIZ);
_MSCDARE	Saldo dare;
_MSCAVER	Saldo avere;
_MSCSALD	Saldo dare/avere;
_MSCSEZS\$	Sezione per Saldo dare/avere (_MSCSALD);

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_MSGMOD

_MSGMOD\$()

Variabile array con gli eventuali messaggi di avviso raccolti durante l'elaborazione nelle operazioni di Import Moduli, visualizzabili mediante la MODVIMSG.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_MSGSVILDB

_MSGSVILDB

_MSGSVILDB\$()

Variabile che ritornano la quantità (_MSGSVILDB numerica) e la descrizione (_MSGSVILDB\$() array alfanumerica) dei messaggi generati dall'istruzione SVILDB.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NAB - Codici Abbinati

_NAB

Variabile che ritorna il numero di elementi (codici abbinati) dell'array dinamico della struttura _AB nella lettura eseguita con GETABBIN.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NAL - Codici Alias

_NAL

Variabile che ritorna il numero di elementi (codici alias) dell'array dinamico della struttura _AL nella lettura eseguita con GETALIAS.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NAPF – Numero Documenti AVANZAPF

_NAPF

Variabile che ritorna il numero di elementi dell'array dinamico ritornati dalla funzione AVANZAPF nella variabile _APFNUM().

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NAPPSTS – Numero Documenti ELENCASTATOAPP

_NAPPSTS

Variabile che ritorna il numero di elementi dell'array dinamico ritornati dalla funzione ELENCASTATOAPP.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NBL - Righe Bolla di Lavorazione

_NBL

Variabile che ritorna il numero delle righe in Bolla Lavorazione al momento della lettura.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NCOMP - Componenti Fase

_NCOMP

Variabile che contiene il numero dei componenti di una fase (max 5000 per fase) in relazione alle operazioni sulla DBP.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NCAM - Codice Campo

_NCAM\$

Variabile che contiene il codice del campo del Gestionale in cui si trova il cursore in quel momento. Il codice del campo serve per creare le etichette personalizzabili in Collage ed è visibile anche selezionando "Informazioni su" dal menu del Gestionale, nelle quali sono visualizzati i possibili eventi: ON_IN_CODICE: (all'ingresso), ON_OUT_CODICE: (all'uscita), ON_ERR_CODICE: (alla visualizzazione di un errore) e ON_612_CODICE: (alla pressione tasto [F12]). Un esempio di etichetta personalizzata sul tasto funzione [F12] è stata dimostrata nell'esempio introduttivo di questo manuale.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NCM - Elementi Calendario Agenda Attività Utente

_NCM

Variabile che contiene il numero di elementi del Calendario dell'Agenda Attività Utente letti con GETCM.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NDUTPLOT - Dati utente nei Tipi Lotto

_NDUTPLOT

Variabile che all'esecuzione della GETTTPLOT riporta il numero di dati utente gestiti nei Tipi Lotto

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NFASI - Numero Fasi DBP

_NFASI

Variabile che contiene il numero delle fasi in relazione alle operazioni sulla DBP.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NFIN - Numero finestra attiva

_NFIN

Variabile che consente di controllare il numero di finestra Collage attiva in quel momento.

Se `_NFIN` è > 0 significa che è relativa alla personalizzazione (es. APRIFIN), se invece `_NFIN` è < 0 significa che è quella relativa al Gestionale.

Nella personalizzazione Sprix il numero di finestra è gestito, invece, con lo stesso codice del programma Sprix (es. AUTOFIN).

Vedere anche le funzioni CERCACODFIN() e GETCODFIN\$()

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NIL - Numero Righe Componenti Impegni Lavorazione

_NIL

Variabile che ritorna il numero di righe di componenti Materia Prima nell'operazione di lettura GETIL

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NLOTTIAPF – Numero Lotti AVANZAPF

_NLOTTIAPF

Variabile di input che negli avanzamenti di produzione, operazione di carico, specifica il numero di elementi presenti nelle variabili vettore relative ai lotti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NLSTR – Numero Righe Liste di Prelievo

_NLSTR

Variabile che nella lettura delle Liste di prelievo riporta il numero di Righe (in 'R' elementi di struttura) che lo compongono. Se non è azzerata da AZZVARSYS o da AZZCATVARSYS procedere direttamente con `_NLSTR=0`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NMM - Numero Righe Documento Magazzino

_NMM

Variabile che nella lettura del documento di magazzino riporta il numero di Righe (in 'R' elementi di struttura) che lo compongono. Se non è azzerata da AZZVARSYS o da AZZCATVARSYS procedere direttamente con `_NMM=0`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NPAM - Numero Righe Progressivi Articoli

_NPAM

Variabile che nella lettura dei progressivi articoli riporta il numero di Righe (in 'R' elementi di struttura) che lo compongono. Se non è azzerata da AZZVARSYS o da AZZCATVARSYS procedere direttamente con `_NPAM=0`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NPN - Numero Righe Registrazione Primanota

_NPN

Variabile che ritorna il numero di righe lette del documento di primanota.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NRDATI - Numero Record File Indici

_NRDATI

Variabile che riporta il numero di record del file ad indici a seguito di operazioni GETKEY, NEXTKEY, PREVKEY, VARKEY, INSKEY, DELKEY.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NSVILDB - Numero righe componenti SVILDB

_NSVILDB

Variabile che riporta il numero di righe componenti sviluppate dalla SVILDB. Se = 0 l'errore è contenuto nella variabile `_ERRSVILDB$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NTMM - Numero Testate Documenti Magazzino

_NTMM

Variabile che nella lettura del documento di magazzino riporta il numero di testate dei documenti origine/trasformati (in 'T' elementi di struttura) che lo compongono. Se non è azzerata da AZZVARSYS o da AZZCATVARSYS procedere direttamente con `_NTMM=0`. Notare: nella composizione del documento, la valorizzazione delle variabili di testata (quelle riferite alla `_NTMM`) devono essere valorizzate in modo uniforme e omogeneo, altrimenti non vengono stampate. (vedere approfondimenti sulla operatività come da Manuale Generale dell'utente)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NUMDIRFILE - Numero di file nella directory

_NUMDIRFILE

Variabile valorizzata dall'istruzione LEGGIDIR e che ritorna il numero di file nel percorso della directory.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_NUMSTAMP - Numero Stampante Aperta

_NUMSTAMP

Variabile che identifica la stampante attualmente aperta: Valori: da 1 a 126 (numero di stampante installata e attualmente aperta), 0 (stampa soppressa), 127 (stampa a video), -1 (stampa annullata incondizionatamente da utente con [Esc/Fine] al messaggio d'inserimento modulo o con CHIUDISTAMP); _NUMSTAMP non identifica (e non può modificare) il numero di stampante selezionata dall'utente nella procedura standard di stampa del gestionale.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_OB - Testata Bolle di Lavorazione

_OB

Variabili di struttura archivi di Testata della Bolla di Lavorazione (GETBL – GETATTBL – PUTBL – DELBL) (le variabili di Riga sono nella successiva struttura _OL)

_OBNUM	numero BL – valorizzata in PUT -
_OBNSO	numero sottoBL – valorizzata in PUT -
_OBNSP	num. sottoBL padre, come _OBNSO, -1 per le BL x.0000
_OBDAT\$	data documento BL
_OBMAG	magazzino PF – prodotto finito -
_OBMAT	magazzino MP – materia prima -
_OBSCA\$	data scadenza BL
_OBDST\$	nota BL
_OBPIA\$	piano di lavoro S/N
_OBCTO\$	codice conto
_OBTIP\$	tipo modulo di stampa
_OBSTA\$	stampa della BL, se="S" elimina la dicitura "BL" dal campo
_OBVAL	totale valore scarichi
_OBINF\$	informazioni del PF
_OBPRZ\$	fissa prezzi S/N
_OBASS	fase assegnata
_OBDDBP\$	codice DBP –se unico-
_OBOPE\$	tipo operazione impostata –solo GETATTBL-
_OBSML\$	Cat.SML da sviluppare [F7], valori A-Z
_OBCRS\$	Non creare sml sospeso
_OBDTC\$	Data creazione

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_OL - Riga Bolla di Lavorazione

_OL

Variabili di struttura archivi di Riga della Bolla di Lavorazione (GETBL – GETATTBL – PUTBL – DELBL) (le variabili di Testata sono nella precedente struttura _OB)

_OLTPR\$()	Tipo riga: "R"=normale; "r"=normale bloccata –BL lavorante-; "P"=PF sospeso –solo con sottoBL-; "S"=SML semilavorato sospeso;
_OLLAV()	numero sottoBL –stottobolla- : =0 SML sospeso; >0 in lavorazione (32000 per le BL x.0000); <0 già caricato (32000 per le BL x.0000);
_OLART\$()	codice PF
_OLPFP\$()	codice PF Padre
_OLRIF()	numero di riferimento: 1-16000000 collegata a OC; >16000000 produzione per il magazzino
_OLRII()	numero riferimento impegni
_OLCST()	totale valore scarichi
_OLCLI\$()	cod. cliente
_OLSPE\$()	cod. indirizzo di spedizione
_OLDBP\$()	cod. DBP
_OLSDO()	serie OC
_OLORD()	numero OC
_OLDAT\$()	data documento OC
_OLSCO\$()	data scadenza OC
_OLSCA\$()	data scadenza riga
_OLNTG()	tipo riga quantità: (0=no taglie, 1-99 taglie);
_OLDEC()	num. Decimali
_OLUM\$()	descrizione unità di misura (variabile in sola lettura)
_OLCOL()	Colli

_OLQTF()	Quantità
_OLTAG()	quantità per taglia
_OLTAGXL()	quantità per taglia 17-32
_OLCOE()	coeff. Conversione um1/um2
_OLFSC()	ultima fase scaricata
_OLFSI()	ultima fase impegnata
_OLFSD()	ultima fase depositata
_OLFST()	ultima fase stampata
_OLCBL()	num. sottoBL o sottoBL colleg lavorante
_OLCFS()	precedente fase scaricata nella BL lavorante
_OLMAG()	magazzino caricato
_OLNRX()	numero record. Sola lettura
_OLVAL()	valore corrente scarico
_OLUNI()	Numero univoco di riga. Sola lettura (è gestita dal programma); è valorizzata in GET se presenti righe PF nella BL. Nota: la variabile è attiva se "gestione collegamento pf" degli ulteriori dati della DBP è impostato "S" (collegamento PF-SML)
_OLUNP()	numero univoco di riga PF. Nota: Riferito al PF padre valido solo per SML (fa riferimento al prodotto finito padre, quindi per i PF ritorna zero, per i SML ritorna il PF a cui appartengono). Nota: la variabile è attiva se "gestione collegamento pf" degli ulteriori dati della DBP è impostato "S" (collegamento PF-SML)
_OLFAS()	num. Fase impostata su riga PF; solo GETATTBL
_OLLOT()	identificativo del lotto
_OLARE\$()	Codice Area
_OLCOM\$()	Codice Commessa
_OLTDOS\$()	Tipo Documento collegato
_OLSTR\$()	Stato di riga
_OLNOT\$()	Nota di riga
_OLSMLPF\$()	Categoria sviluppo SML per PF
_OLRIC()	Numero riferimento creazione

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PA - Tabella Pagamenti

PA

Variabili di struttura per Istruzioni Tabella Pagamenti (GETPAG - PUTPAG - DELPAG)	
	Pagamenti (255)
_PACOD	Codice pagamento (1-255)
_PADES\$	Descrizione pagamento (stringa 40 caratteri)
_PABOLFATT\$	B = data bolla F = data fattura
_PASPEBAN\$	Spese banca S/N
_PASPEBOL\$	Spese bolli S/N
_PAMERCABB\$	M = merce A = abbuono
_PARATEPER\$	Rate a percentuali S/N
_PASCADFIS	Giorni scadenza fissa (1-31)
_PAGIORNI	Più giorni da aggiungere a fine mese; attivo solo se Giorni scadenza fissa è uguale a 31
_PASCONTI	Sconto (99,99)
_PARATIVA	Indica in quale rata c'è l'iva. Da utilizzare come indice nella struttura Rate
_PADESPAPN\$	Descrizione pagamento in testata Primanota
_PARATACON\$	Paga rata contante
_PACOSRICA\$	Centro di costo e ricavo in Pagamento/Abbuono
_PACAUSPG\$	Causale PG
_PACAUSAB\$	Causale AB
_PACONTPG\$	Conto per PG
_PACONTABP\$	Conto per AB passivo
_PACONTABA\$	Conto per AB attivo
_PAPAGELET\$	Pagato elettronicamente
_PAMODOIVA\$	"+" = importo più iva " " = rata di sola iva
	Tipi rate (max 250 elementi)
_PATIPRATA\$()	Tipo rata
_PAGIORSCA()	Giorni scadenza
_PASOLOIVAS\$()	Stringa che può contenere 'l' o '+l'
_PAPERCENT()	Percentuale per trattare la rata a percentuale
_PATPRATFSS\$()	Tipo rata fattura self (v660)
	Tipi pagamenti (dati generali) (max 13 elementi)
_PAGETIPOR\$()	Tipo rata (esempio "C")
_PAGEDESTIS\$()	Descrizione (esempio "Contanti")
_PAGEVALUT()	Valuta
_PAGERAMIM()	Rata minima
_PAGEPAALT()	Pagamento alternativo: da prendere solo se tutte le rate sono dello stesso tipo e non sono a percentuale
_PAGETIALT\$()	Tipo rata alternativo: da prendere solo se il pagamento ha rate miste e/o a percentuale
_PAGETIRASS\$()	Tipo rata fattura self (Valore di default) Es "MP01" (v660)
	Tipi scadenze (dati scadenario dati generali) (4 elementi)
_PAGETISCA\$()	Tipo scadenza
_PAGEDESCA\$()	Descrizione
	Tipi pagamento per fattura elettronica (22 elementi)
_PAGEDESBFE\$()	Tipo rata fattura self es: "MP01"

_PAGEDESLFE\$()	Descrizione es: "Contanti"
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_PAM - Progressivi Articoli

PAM

Variabili di struttura per lettura Progressivi Articoli (PROGARTMAG)	
_PAMCODART\$()	Codice primario articolo
_PAMDESART\$()	Descrizione articolo
_PAMMAG()	Codice Magazzino
_PAMIDUBI()	Id Ubicazione
_PAMIDLOTTO	Id Lotto
_PAMUM1\$()	Unità di misura primaria
_PAMUM2\$()	Unità di misura secondaria
_PAMINQ()	Quantità inventario
_PAMINT(,32)	Quantità inventario articolo a taglie
_PAMTIN()	Tara inventario
_PAMCIN()	Colli inventario
_PAMCAR()	Quantità di carico
_PAMCAT(,32)	Quantità di carico articolo a taglie
_PAMTCA()	Tara carico
_PAMCCA()	Colli carico
_PAMSCA()	Quantità di scarico
_PAMSCT(,32)	Quantità di scarico articolo a taglie
_PAMTSC()	Tara scarico
_PAMCSC()	Colli scarico
_PAMOFOR()	Quantità ordini a fornitore
_PAMOFORT(,32)	Quantità ordini a fornitore articoli a taglie
_PAMOCLIMP()	Quantità ordini di clienti impegnati
_PAMOCLIMPT(,32)	Quantità ordini di clienti impegnati articoli a taglie
_PAMOCLDAIM()	Quantità ordini di clienti da impegnare
_PAMOCLDAIMT(,32)	Quantità ordini di clienti da impegnare articoli a taglie
_PAMOCLAUTO()	Quantità ordini di clienti automatici
_PAMOCLAUTOT(,32)	Quantità ordini di clienti automatici articoli a taglie
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_PART - Particolarità

PART

Variabili di struttura archivi delle Particolarità (GETPART – GETPARTICOL – GETATTPART – PUTPART – DELPART)	
_PARTTIP\$	Tipo Particolarità. Un carattere (P/V/S/A/I) oppure fino a tre caratteri. Questa variabile è il parametro principale che stabilisce il tipo di particolarità in oggetto e tradizionalmente è composta da un carattere (P/V/S/A/I); la variabile prevede altresì la possibilità di aggiungere anche un secondo e un terzo carattere che condizionano i restanti parametri della chiave al fine di considerare la combinazione di ulteriori elementi della particolarità (*). Consultare la voce PARTICOLARITA' del capitolo Comandi Raggruppati per Argomento. Clicca qui
_PARTCLI\$	Codice conto
_PARTCAC	Categoria conto
_PARTART\$	Codice articolo
_PARTCAA	Categoria articolo
_PARTDES\$	Descrizione articolo
_PARTDDT\$	Data di validità da...
_PARTADT\$	Data di validità a...
_PARTIVA\$	Aliquota / esenzione
_PARTVAL	Numero valuta
_PARTCONDS\$	Condizione (dizionario tzcon)
_PARTAPPLI\$	Applicazione (dizionario tzapl)
_PARTTPF\$	Tipo fino a F/Q/P/S/N
_PARTZONA	Zona Cliente (*)
_PARTSTATCLI	Categoria Statistica Cliente (*)
_PARTGRPMER\$	Gruppo Merceologico (*)
_PARTSTART\$	Categoria Statistica Articolo alfanumerica (*)
_PARTSTART	Categoria Statistica Articolo numerica (*)
_PARTNATURA\$	Natura Articolo (*)
_PARTPROMO	Codice Promo
_PARTAPPSU\$	Validità della Particolarità. Valori ammessi: "G" sul Gestionale; "S" sul Sito; "E" su Entrambi
_PARTFIN(1-3)	Valore fino a scaglione Q,P,S – solo 3 elementi -
_PARTPRZ\$(1-4)	Particolarità dello scaglione _PARTFIN; Indice = 4 se è Fisso, che rappresenta l'elemento 9 del video; vale "*****" se non è rappresentabile. Nota: sostituita da _PARTPRZN\$ dopo l'introduzione degli sconti in cascata). La retrocompatibilità con _PARTPRZ\$ è consentita solo per vecchi eseguibili non ricompilati. In presenza di GET di particolarità

	la ricompilazione inizializza _PARTPRZN\$ (sostituire _PARTPRZ\$).
_PARTTP\$(1-4)	Campo "%" o "T" – solo multiagente -
_PARTPROV\$(1-4)	Risultato formula _PARTPRZN\$ - solo multiag. e GETATTPART
_PARTAGEN\$(1-4)	Cod. agente – solo multiagente -
_PARTCOND(1-4)	Codice condizione – solo multiagente -
_PARTPRZN\$(1-4)	Particolarità, a partire dall'introduzione degli sconti in cascata
_PARTCOD\$	Codice articolo del Cli/For se <tpart\$>="A"
_PART_FIN(4-8)	Valore fino a scaglione Q,P,S
_PART_PRZN\$(4-8)	Particolarità dello scaglione
_PART_TP\$(4-8)	Campo "%" o "T" – solo multiagente -
_PART_PROV\$(4-8)	Risultato formula _PARTPRZN\$ - solo multiag. e GETATTPART
_PART_AGEN\$(4-8)	Cod. agente – solo multiagente -
_PART_COND(4-8)	Codice condizione – solo multiagente -
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_PC - Piano dei Conti (Anagrafica e Rubrica)

_PC

Variabili di Struttura della Anagrafica del Piano dei Conti (GETRUBUNICA – GETPC – GETATTPC – PUTPC – DELPC)					
Le variabili evidenziate in grassetto sono quelle minime obbligatorie, ma possono aumentare in base a:					
<ul style="list-style-type: none"> • tipo di conto (es. clienti / fornitori) • configurazione contabile (es. presenza di Rubrica Unica) • condizioni fiscali (es. Fatturazione Elettronica) • ... 					
		P	E	C	F
_PCCOD\$	Codice del conto. Può assumere codifica automatica "GMM.AUTO", esempio _PCCOD\$="201.AUTO". Se il parametro contabile "Rubrica unica cli/for = C" considerare il parametro contabile "All.aut.cli/for rub.unica": se = N non e' consentito l'allineamento, se = S e' consentito ma solo in forma "GMM.AUTO" e non e' consentita l'assegnazione diretta del numero di conto.	p1	p1	p1	p1
_PCCAL\$	Codice Alternativo	p1	p1	p2	p2
_PCRIFSTD\$	Conto di collegamento al PDC standard 80/81 (righe redditi) - solo lettura -	p1	p1	-	-
_PCIDRUBUNI	ID di RUBRICA UNICA, di sola lettura. La PUTPC di un nuovo conto riporta l'identificativo assegnato. Nota: la replica MexalDB è un concetto completamente aziendale; le tabelle generali (raggruppamenti, rubrica unica rudt, banche, ecc) sono database aziendali, pertanto trattati per ciascuna singola azienda. Es: nelle aziende AAA e BBB con rudt e MexalDB attivo automatico la replica iniziale di entrambe le aziende crea le tabelle AAA_RUDT in "aaa_rp" e BBB_RUDT in "bbb_rp", ma l'aggiornamento del dato di rubrica unica nell'azienda AAA aperta allinea solamente la tabella AAA_RUDT (non la tabella BBB_RUDT) e viceversa; anche in caso di aggiornamento senza alcuna azienda aperta non allinea alcunchè, perchè MexalDB non è attivo. Non è possibile aggiornare una tabella del database bbb_rp tramite l'azienda AAA aperta/chiusa e viceversa. Per allineare le tabelle delle altre aziende, basta aprire l'azienda ed in Servizi - Riorganizzazione archivi aziendali selezionare RUBRICA PC/CLIENTI/FORNITORI e premere F8 (Elabora solo MexalDB), oppure aprire la tabella di raggruppamento e riconfermare un valore.			p1	p1
_PCNAZ\$	Nazionalita'	-	-	p1	p1
_PCPAE\$	Paese	-	-	p1	p1
_PCPRI\$	Soggetto privato	-	-	p1	p1
_PCPFS\$	Persona fisica	-	-	p1	p1
_PCCFI\$	Codice Fiscale	-	-	p1	p1
_PCNPI\$	Partita Iva 15 caratteri senza identificativo del paese	-	-	p1	p1
_PCNPIEXT\$	Partita Iva estera dal 16 al 24 carattere	-	-	p1	p1
_PCPI\$	Partita Iva Secondaria	-	-	p1	p1
_PCNOPIVA\$	Senza Partita Iva S/N; Italiani, noprivati e cf=solonomico	-	-	p1	p1
_PCDENOM\$	Denominazione ditta individuale (persona fisica non privata)			p1	p1
_PCDESS\$	Descrizione conto / Ragione Sociale	p1	p1	p1	p1
_PCDSA\$	Desc. aggiuntiva AZIENDA (*a)	p1	p1	-	-
_PCNOLEAS\$	Nolo/leasing	p1	p1	-	-
_PCGCA\$	Giornale di cassa	p1	-	-	-
_PCASS\$	Assogettamento Iva -file archivio RUDT-	p1	p1	-	-
_PCIDT	Perc. Indetraibilita'	p1	-	-	-
_PCPAM(26)	Ammortamento standard	p1	-	-	-
_PCMIM\$	Mater/Immateriali/mater No calcolo manutenzioni	p1	-	-	-
_PCTPCESP\$	Automezzo trasportatore	p1	-	-	-
_PCTPMANUT\$	Tipo manutenzione	p1	-	-	-

_PCALMANUT	Percentuale manutenzione	p1	-	-	-
_PCCSK\$	Conto di costo -per K-	p1	-	-	-
_PCCSZ\$	Conto effettivo -per Z-	p1	-	-	-
_PCIMM\$	Conto immobilizzazione -per V-	p1	-	-	-
_PCECASTEL\$	Escludi da apertura automatica castelletto iva su conti patrimoniali	p1	-	-	-
_PCCLASS\$	Classe Conto (Banca / Cassa) - v. Ulteriori dati banca [archivio 'BANC']	p1	-	-	-
_PCDAD\$	Ragione Sociale fino alla data - formato data	-	-	F2	F2
_PCNAD\$	Ragione Sociale alla data - formato stringa	-	-	F2	F2
_PCCOG\$	Cognome	-	-	p1	p1
_PCNOM\$	Nome	-	-	p1	p1
_PCDTN\$	Data nascita/costituzione	-	-	p1	p1
_PCLON\$	Localita' nascita/costituzione - Comune/Stato estero	-	-	p1	p1
_PCPRN\$	Provincia nascita/costituzione - Pr	-	-	p1	p1
_PCIND\$	Indirizzo	-	-	p1	p1
_PCCAP\$	Cap	-	-	p1	p1
_PCLOC\$	Localita'	-	-	p1	p1
_PCPRO\$	Provincia	-	-	p1	p1
_PCTEL\$	Telefono	-	-	p1	p1
_PCFAX\$	Fax	-	-	p1	p1
_PCINT\$	EMail indirizzo Internet	-	-	p1	p1
_PCINT1\$	Sito web indirizzo Internet	-	-	p1	p1
_PCPEC\$	Indirizzo PEC	-	-	p1	p1
_PCSPORTIVO\$	Collaboratore sportivo S/N	-	-		
_PCSPORTAZI\$	Collaboratore esclusivo S/N	-	-		
_PCSPORTALR	Aliquota addizionale regionale collaboratore sportivo	-	-		
_PCSPORTALC	Aliquota addizionale comunale collaboratore sportivo	-	-		
_PCPAGA	Pag.art. alimentari non deperibili				
_PCPAGD	Pag.art. alimentari deperibili				
_PCALC	Aliq. Di Compensazione				
_PCRAT\$	Rateo/Risconto				
_PCTPMERCE	Tipo merce aziende RSM				
_PCIRPE	<i>Obsoleta, non utilizzabile</i>	-	-	-	-
_PCIRAP	<i>Obsoleta, non utilizzabile</i>	-	-	-	-
_PCRIC\$	Nome di ricerca	-	-	p2	p2
_PCTPSOC\$	Tipo societa' N,I,C,P	-	-	p2	p2
_PCLOTTE\$	Codice Lotteria per documenti CO, RE, AN (MxRT); 8 car. alfanumerico	-	-	p2	p2
_PCENTEP\$	Ente Pubblico	-	-	p2	p2
_PCESPLITPG\$	Escludi da gestione Split Payment	-	-	p2	p2
_PCFATELEPA\$	Fattura elettronica P.A. "S"/"N"; Nota: obsoleta dalla versione 2016L, campo di sola lettura	-	-	p2	p2
_PCTPFATTE\$	Fattura elettronica : F=Fattura PA, E=Eni e-Fatt; campo alternativo a _PCFATTELE\$ (viene eseguito un controllo in fase di scrittura PUTPC nella versione 65.4b)	-	-	p2	p2
_PCFATTELE\$	Fattura elettronica soggetti privati; campo anche in scrittura, accetta valori: P = PDF, S = SDI; campo alternativo a _PCTPFATTE\$ (viene eseguito un controllo in fase di scrittura PUTPC nella versione 65.4b)	-	-	p2	p2
_PCDFATTEL\$	Data inizio gestione Fattura Elettronica ; se omessa sempre attiva; il confronto in gestione è con la data di sistema; è una data che se immessa consente di posticipare la data inizio fattura SDI privati (B2B); corrisponde al dizionario pcdfe()	-	-	p2	p2
_PCSERIELE	Sezionale iva (serie Fattura Elettronica)	-	-	p2	p2
_PCALLXML\$	Includi formato allegato HTML accetta i valori S/N	-	-	p2	p2
_PCRALLFRM\$	Codice formato modulistica da allegare al file XML	-	-	p2	p2
_PCXMLSTIMP\$	Stato import xml; può assumere i valori "" (vuoto [Si/No] come da parametro aziendale -opzione predefinita-, "D" Da verificare, "V" Verificata)				
_PCXMLTPNR\$	Per costruire il numero documento dal Tag <Numero> Xml; Assume valori: N = non gestita; P = Passepartout; A = Altri (si integra con gli offset)				
_PCXMLPOSNR(P)	Array per costruire del numero documento dal Tag <Numero> Xml; L'indice numerico dell'Array può avere significato: - "DALLA POSIZIONE" (se P=1 oppure P=3); - "ALLA POSIZIONE" (se P=2 oppure P=4); - Se _PCXMLTPNR\$=A l'indice significa altri "DALLA 1 ALLA 20"; Nota: la seconda coppia DALLA 2 ALLA 4 (cioè quella utile per il secondo range di posizioni) è opzionale; L'array costruisce il numero del documento dal contenuto <Numero> del Tag della fattura elettronica, in quanto il numero potrebbe dovere essere "sistemato" in fase di import. Del numero lungo 20 caratteri, quindi, si può decidere di prendere una parte oppure due parti (opzionale), attraverso due range stabiliti attraverso l'elemento del indice numerico "P" della Array _PCXMLPOSNR(P). Le posizioni "DALLA" sono sull'elemento 1 e sul 3, le posizioni "ALLA" sono sull'elemento 2 e sul 4. Esempi: range primi 10 caratteri _PCXMLPOSNR(1)=1; _PCXMLPOSNR(2)=10; range tutti i caratteri _PCXMLPOSNR(1)=1; _PCXMLPOSNR(2)=20; ----- due range primo range _PCXMLPOSNR(1)= 3 ; _PCXMLPOSNR(2)=7 (inizia DALLA posizione 3 e termina ALLA posizione 7); secondo range (opzionale) _PCXMLPOSNR(3)=15; _PCXMLPOSNR(4)=19 (inizia DALLA posizione 15 e termina ALLA posizione 19); ----- due range sul numero 1234567890ABCDEFHIL				

	primo _PCXMLPOSNR(1)=5; _PCXMLPOSNR(2)=8; secondo _PCXMLPOSNR(3)=18; _PCXMLPOSNR(4)=20; il numero risultante è = 5678HIL				
	Variabili _PC per la costruzione del numero documento dal tag <Numero> Xml: _PCXMLTPNR\$ =(N=non gestita; P=Passepartout; A=Altri (si integra con gli offset)) _PCXMLPOSNR[1/2/3/4] (dalla posizione[1/3] alla posizione[2/4]; caso A=altri da 1 a 20)				
_PCRIFAMMPA\$	Riferimento amministrazione PA	-	-	p2	p2
_PCCODUPA\$	Codice Univoco Ufficio PA (Pubblica Amministrazione) - Fattura Elettronica	-	-	p2	p2
_PCOBRFAMPA\$	Obbligo Riferimento documenti	-	-	p2	p2
_PCCODSDI\$	Identificativo privati SDI	-	-	p2	p2
_PCNOSPESOM\$	Escludi spesometro "S"/"N"	-	-	p2	p2
_PCOPTURIS\$	Spesometro OP.legate al Turismo				
_PCCONTRSPS\$	Tipo contratto spesometro: N/C/P				
_PCDOCRIEP\$	Considera conto come "Documento Riepilogativo" nello Spesometro "S"/"N"	-	-	p2	p2
_PCESTEROM\$	Trattamento soggetto in Esterometro. "I" = Includi, "E" = Escludi, "" = Automatico; dizionario 'pctes'	-	-	p2	p2
_PCAGRI34C6\$	Soggetto Agricolo Art. 34 comma 6 "S"/"N"	-	-	p2	p2
_PCDOMU\$	Condominio	-	-	p2	p2
_PCDTNASC\$	Data di nascita – black list – [STATN]	-	-		
_PCPAENASC\$	Paese di nascita – black list – [STATN]	-	-		
_PCSELFBILL\$	Self billing. Solo conti di tipo "F", accetta i valori S/N	-	-	-	-
_PCRGFISC\$	Regime fiscale Fattura XML, solo conti "F", self billing attivo. Tipo di regime fiscale.	-	-	-	-
_PCIVADIFF\$	Gestione iva vendite esigibilità differita, solo conti "F" self billing attivo. Accetta i valori S/N	-	-	-	-
_PCALFORF	Classificazione - % forfetizzazione (conti Economici reg.speciale)				
_PCRGCLASSE\$	Classificazione - Conto per registro (regime iva 3)				
Codici Bilancio					
_PCUEDA\$	Codice 34 DIRETTIVA UE Dare	F3	F3	-	-
_PCUEAV\$	Codice 34 DIRETTIVA UE Avere	F3	F3	-	-
_PCCEDA\$	Codice IV DIRETTIVA CEE Dare Pat/Eco	F3	F3	-	-
_PCCEAV\$	Codice IV DIRETTIVA CEE Avere Patr.	F3	F3	-	-
_PCRSMDA\$	Codice Dare (Bilancio RSM in formato XBRL)	F3	F3	-	-
_PCRSMAV\$	Codice Avere (Bilancio RSM in formato XBRL)	F3	F3	-	-
Conti Automatici (per conti di tipo "I")					
_PCCOR\$	Conto ammortamento ordinario	ShF4	-	-	-
_PCCAN\$	Conto ammortamento anticipato	ShF4	-	-	-
_PCCND\$	Conto ammortamento non fiscalmente deducibile	ShF4	-	-	-
_PCAOR\$	Conto fondo ordinario	ShF4	-	-	-
_PCAAN\$	Conto fondo anticipato	ShF4	-	-	-
_PCAND\$	Conto fondo non fiscalmente deducibile	ShF4	-	-	-
_PCPLU\$	Conto plusvalenze	ShF4	-	-	-
_PCPND\$	Conto plusvalenze non fiscalmente tassabili	ShF4	-	-	-
_PCMIN\$	Conto minusvalenze	ShF4	-	-	-
_PCMND\$	Conto minusvalenze non fiscalmente deducibili	ShF4	-	-	-
Conti Automatici (per conti di tipo "R")					
_PCRIMIOS\$	Conto rimanente iniziali per Ordinaria (PiCo 80, mastri 204, 205, 206)	ShF4			
_PCRIMIS\$	Conto rimanente iniziali per Semplificata (PiCo 80, mastri 204, 205, 206)	ShF4			
Riconciliazione bancaria, Riconciliazione, CSV-Parametri import banca					
_PCBARICO	Configuratore CSV/TXT				
Archivio 'BANC' - modulo Banche (J) - [Ulteriori dati banca]					
_PCNBA	Numero Banca su Archivio 'BANC' (*a)				
_PCABI	ABI			F6	F6
_PCCAB	CAB			F6	F6
	<i>Azi, Fil, Ind, Loc, Cap, Pro, T, F, Mail, Int</i>				
_PCCUC\$	Codice univoco CBI (CUC)				
_PCBASIS\$	Cod. S.I.A (*a)				
_PCRIDIDE\$	Id. impresa (cod. ident. RID arricchito conti classificati banca)				
_PCBAAZ\$	Firma Azienda (*a)				
Coordinate Bancarie Italiane (BBAN) - Dati Piano Conti (aggancio _PCCOD\$ e _PCDES\$)					
_PCBACC\$	Conto Corrente c/c numero (*a)	F6 p.2 banche	-	-	-
_PCCCB\$	Conto Corrente c/c numero (*a) - Cli/For	-	-	F6	F6
_PCBATC\$	Tipo C/C (*a)	F6 p.2 banche	-	-	-
_PCBBCIN\$	Cin (controllo BBAN) (*a)	F6 p.2 banche	-	F6	F6
	<i>Banca abituale per F24</i>	F6 p.2 banche	-	-	-
Coordinate Bancarie Internazionali					
_PCIBPAE\$	Paese - codice per codifica IBAN (*a)			F6	
_PCIBCIN	Cin - codice per codifica IBAN (*a)			F6	
_PCIBCOD\$	BBAN - codice libero su codifica IBAN (*a)			F6	
_PCBBAN\$	BBAN- cod.composto: [Cin+ABI+CAB+c/c]			F6	

		(nota: verificare se usare_PCIBCOD\$				
_PCIBAN\$		IBAN[_PCIBPAE\$+_PCIBCIN+_PCIBCOD\$]			F6	
_PCBIC\$		Codice BIC/SWIFT, Condizioni (*a)			F6	
<i>Bollo virtuale Riba</i>						
		Prov. intendenza Finanza ... Autorizz., Numero ... Data ...				
Ulteriori Dati - Banche -						
_PCBADIS\$		Data iniz.rapporto (*a)	F5	-	-	-
_PCBADFS\$		Data fine rapporto (*a)	F5	-	-	-
_PCBAF1		Fido (scoperto C/C)	F5	-	-	-
_PCBAF2		Fido castelletto effetti	F5	-	-	-
_PCBAAD\$(6)		Data tasso attivo (*a)	F5	-	-	-
_PCBAAL(6)		Aliquota tasso attivo (*a)	F5	-	-	-
_PCBAPD\$(6)		Data tasso passivo (*a)	F5	-	-	-
_PCBAPL(6)		Aliquota tasso passivo (*a)	F5	-	-	-
_PCBARI\$		Riferimento Signor (*a)	F5	-	-	-
_PCBASE\$		Riferimento Settore (*a)	F5	-	-	-
_PCBATES\$		Riferimento Telefono (*a)	F5	-	-	-
_PCBANOS\$		Riferimento Note (*a)	F5	-	-	-
Condizioni e automatismi						
Condizioni generali del soggetto						
_PCASIS\$		Assogettamento IVA particolare - file PICO -	-	-	F6	F6
_PCASIXML\$		Esenzione operazioni passive con natura non compatibile				
_PCVAL		Valuta	-	-	F6	F6
_PCOPCHARG		Tipo operazione reverse charge	-	-	-	F6
_PCCST		Categoria statistica	-	-	F6	F6
_PCONA		Zona	-	-	F6	F6
_PCBES\$		Bollo esportatore abituale	-	-	F6	-
Condizioni di pagamento						
_PCPAG		Pagamento abituale	-	-	F6	F6
_PCSCF		Scadenza fissa	-	-	F6	F6
_PCSPC		+ giorni proroga	-	-	F6	F6
_PCMES(2)		Mese escluso	-	-	F6	F6
_PCMAP(2)		Anticipo/posticipo	-	-	F6	F6
_PCREF\$		Raggruppamento effetti: S/N/P/U/Y U=S+spese 1^ fatt.; Y=P+spese banca su 1^ fatt.	-	-	F6	-
_PCCOB\$		Banca presentazione effetti - (conto archivio 'PICO')	-	-	F6	F6
_PCCIGCUP\$		Gestione CIG-CUP Legge 136/2010 abilitato S/N	-	-	F6	F6
Banche appoggio cliente/fornitore (coordinate bancarie estese) (Coordinate Bancarie IBAN e BBAN vedere archivio 'BANC')						
_PCBAP\$		Banca d'appoggio Abituale	-	-	F6	F6
_PCSDDMND		Mandato RID - numero addebito diretto su Banca appoggio Abituale	-	-	F6	F6
_PCCDRID\$		Codice mandato cliente su RID (se _PCSDDMND=0)	-	-	F6	F6
_PCRIDTPI\$		Tipo mandato incasso Rid "" o "V" sdd B2B	-	-	F6	F6
_PCRIDDTM\$		Data mandato Rid clienti banca abituale	-	-	F6	F6
_PCRIDSTS\$		Stato sequenza Rid clienti banca abituale	-	-	F6	F6
Ulteriore banca (3)						
_PCUBBAP\$(3)		Descrizione			F6	F6
_PCUBSDDMND(3)		Numero Mandato addebito diretto RID su Banca appoggio Ulteriore				
_PCUBBIC\$(3)		Codice BIC della banca				
_PCUBPAE\$(3)		Sigla paese IBAN a				
_PCUBIBCIN(3)		Cin IBAN IBAN b				
_PCUBIBCOD\$(3)		Codice IBAN IBAN c = BBAN				
_PCUBBACIN\$(3)		Cin BBAN BBAN 1				
_PCUBABI(3)		Codice ABI BBAN 2				
_PCUBCAB(3)		Codice CAB BBAN 3				
_PCUBCCB\$(3)		Conto corrente BBAN 4				
_PCUBIBAN\$(3)		Codice IBAN solo GET a+b+c				
_PCUBBBAN\$(3)		Codice BBAN solo GET 1+2+3+4				
_PCUBCODRID\$(3)		Codice Cliente x RID				
Mandati addebito diretto (RID - SEPA)						
_PCSDDCODI\$(5)		Codice mandato			F5	-
_PCUBRIDDTM\$(3)		Data mandato Rid clienti				
_PCSDDDATMA\$(5)		Data mandato			F5	-
_PCUBRIDSTS\$(3)		Stato sequenza Rid clienti				
_PCSDDSTAT\$(5)		Stato sequenza			F5	-
_PCUBRIDTPI\$(3)		Tipo incasso "" o "V"				
_PCSDDTIPO\$(5)		Tipo incasso "" o "V"			F5	-
Dati Parcellazione						
_PCRSP\$		Responsabile			F6	-
_PCSPS\$		Spese anticipate studio			F6	-
_PCAIZ\$		Sigla azienda del cliente			F6	-

_PCCPF\$	Contabilizza parcelle -def.N- (*b)			F6	-
_PCSTU\$	Codice fornitore Studio			F6	-
_PCOST\$	Costo prestazione cliente			F6	-
_PCCPG\$	Registra PG di FR -def.N- (*b)			F6	-
_PCPPG\$	Conto da utilizzare per il PG (*b)			F6	-
_PCUTEID	Dati parcella - Utente parcella/contratti				
_PCTPRA\$	Dati parcella - Tipo pratica redditi				
_PCCPRA\$	Dati parcella - Pratica redditi				
	<i>modello / pratica</i>				
_PCPRE\$(12)	Cod.-Descriz. Prestaz. 'R'				
_PCPQT(12)	Quantita'				
_PCPMS(12)	Mese				
_PCPPR(12)	Prezzo				
Condizioni documenti di magazzino					
_PCLIN	Stampa descrizione articolo in lingua			F6	F6
_PCLIS	Listino di vendita			F6	-
_PCLIE\$	Extra su listino			F6	-
_PCCSC	Sconti - categoria			F6	-
_PCSCO\$	Sconto incondizionato			F6	-
_PCAGE\$	Agente - codice conto			F6	F6
_PCAGECOND	Condizioni agente			F6	-
_PCCPR	Provvigioni - categoria			F6	-
_PCRBL\$	Raggruppamento Documenti			F6	-
_PCIEV\$	Proposta evadibilita' ordini (S/N/E/T/P)			F6	-
	<i>Addebito bollo</i>				
_PCFTA\$	Fatturare a - codice conto			F6	F6
_PCSPEDIZ\$	Indirizzo spedizione merce - codice anagrafica			F6	F6
Vettore/Dati Trasporto					
_PCTRM\$	Trasporto a mezzo; valore [D/M/V]. Dizionario pctrm()			F7	F7
_PCVET\$	Cod. vettore abituale - codice conto			F7	F7
_PCVPO\$	Porto			F7	F7
_PCVDE\$	Descrizione porto			F7	F7
_PCMOC\$	Calcolo colli automatico. Determinazione automatica colli. La PUTPC richiede valore null ("") che significa automatico (check box accesa, ovvero "calcolo colli automatico") oppure valore "M" che significa manuale (check box spenta)			F7	F7
_PCMOP\$	Calcolo peso automatico. Determinazione automatica peso. La PUTPC richiede valore null ("") che significa automatico (check box accesa, ovvero "calcolo peso automatico") oppure valore "M" che significa manuale (check box spenta).			F7	F7
_PCVSP\$	Tipo spese spedizione			F7	F7
_PCVVA	Valore spese spedizione			F7	F7
_PCAEB	Aspetto esteriore dei beni; valore [0 - 30]. Dizionario pcaeb()			F7	F7
_PCINCOT\$	Codice Incoterm			F7	F7
_PCNISALBO\$	Num. Iscrizione all'albo			-	F7
_PCDISALBO\$	In data iscrizione all'albo			-	F7
_PCPARDEST\$(10)	Condizioni destinatario; Attenzione: questa array è statica (non ampliabile), pertanto le ulteriori condizioni sono rappresentate dall'array _PCUPARDEST\$. Vedere sezione DATI TRASPORTO in questa tabella.			F7	-
_PCUPARDEST\$(1-N)	Ulteriori condizioni destinatario; Array dinamico con tanti indici quanti sono gli elementi, per esempio: [1]Trasporto a mezzo, [2]Modalita' colli, [3]Modalita' peso, [4]Aspetto esteriore beni, [5]Dati aggiuntivi,....[N]. Per le altre precedenti dieci condizioni consultare l'array statico _PCPARDEST\$(1-10).			F7	-
Imballi					
_PCVAC\$	Valorizzazione cauzioni			F8	F8
_PCIMB(16)	Quantita' cauzioni			F8	F8
<i>Condizioni particolari (vedere variabili particolarità _PART)</i>				-	-
<i>Distinta Base Variazione (vedere variabili distinta base _TP)</i>				ShF5	ShF5
Contropartite automatiche - Clienti/Fornitori					
<i>Contropartite contabilità</i>					
_PCTRPRC\$	Ricavo stand. raggruppamento	-	-	F6	F6
_PCTRPPC\$	Ricavo personale	-	-	F6	F6
_PCTRCON\$(4)	Ulteriore ricavo C/F	-	-	F6	F6
<i>Contropartite magazzino</i>					
_PCTRPMA\$	Articoli Merce (A)	-	-	F6	F6
_PCTRPML\$	Articoli Lavorazione (L)	-	-	F6	F6
_PCTRPMP\$	Articoli C/deposito (P)	-	-	F6	F6
_PCTRPMS\$	Articoli Spesa (S)	-	-	F6	F6

_PCTRPMZ\$	Articoli Prestazione (Z)	-	-	F6	F6
Ritenuta acconto cliente (*c)					
_PCSRRA	Soggetto ritenuta d'acconto (*c)	-	-	p2 F6	p2
_PCALQRDA	Aliquota ritenuta d'acconto (valore massimo 99,99) (*c)	-	-	p2 F6	p2
_PCQIMPRDA	% imponibile ritenuta d'acconto (0 - 100) (*b)	-	-	p2 F6	p2
_PCCAUPRDA	Causale pagamento modello 770S (riferimento a tabella causali pagamento 770 presenti nei parametri prf-studio o in parametri magazzino e nei fornitori nella finestra dei dati percipiente) (*c)	-	-	p2 F6	p2
Rappresentante/comunicazione polivalente					
Persona Rappresentante della Società					
_PCRLCOGNO\$	Cognome Rapp. legale	-	-	F6	F6
_PCRLNOME\$	Nome Rapp. legale	-	-	F6	F6
_PCRLCDF\$	Codice fiscale Rapp. legale	-	-	F6	F6
Dati di Nascita Rappresentante legale					
_PCRLDTN\$	Data di nascita Rapp. legale	-	-	F6	F6
_PCRLLN\$	Comune o stato estero Rapp. legale	-	-	F6	F6
_PCRLPRN\$	Provincia Rapp. legale	-	-	F6	F6
Domicilio Fiscale Rappresentante legale					
_PCRLPAE\$	Codice Stato estero Rapp. legale	-	-	F6	F6
Stabile organizzazione (Emissione fatture elettroniche a soggetti esteri con stabile organizzazione)					
_PCRCAP\$	Codice di avviamento postale				
_PCRIND\$	Indirizzo				
_PCRCVI\$	Numero civico				
_PCRLOC\$	Località				
_PCRPRO\$	Provincia				
_PCRNAZ\$	Nazione				
Rappresentante fiscale (Emissione fatture elettroniche a soggetti esteri con rappresentante fiscale italiano e estero)					
_PCRPAE\$	Paese				
_PCRPI\$	Partita Iva				
_PCRDEN\$	Denominazione				
_PCRCOG\$	Cognome (alternativo a Denominazione)				
_PCRNOM\$	Nome (alternativo a Denominazione)				
Contabilità analitica					
_PCANL\$	Codice analitico per conti diversi da C/F	ShF11	ShF11	-	-
_PCCRIRIP	Criterio di suddivisione per conti diversi da C/F	ShF11	ShF11	-	-
_PCANLCF\$	Codice analitico per Clienti/Fornitori	-	-	F6	F6
_PCCRIRIPCF	Criterio di suddivisione per Clienti/Fornitori	-	-	F6	F6
Spese sanitarie					
_PCTSAOUT\$	Escludi trasmissione tessera sanitaria	-	-	F6	-
_PCTSATSR\$	Tipo spesa raggruppamento	-	-		
_PCTSAPSR\$	Particolarità spesa raggruppamento				
_PCTSATSA\$	Tipo spesa aziendale	-	-		
_PCTSAPSA\$	Particolarità spesa aziendale	-	-		
Split payment soggetti non Ente pubblico					
_PCCATSPILT\$	Categoria Esclusione Split Payment (S/N); variabile corrispondente ad "Altri soggetti Split Payment".	-	-	F6	F6
_PCDTSPIL\$(1-6)	Variabile Array di elementi di tipo data. (*d)	-	-	F6	F6
Dati percipiente/Regime agevolato					
_PCCTR\$	Codice tributo	-	-	-	ShF5
_PCTPRITEN\$	Tipo ritenuta. "A" = Acconto, "I" = Imposta	-	-	-	ShF5
_PCMANI\$	Regime agevolato. Attività Marginali /Nuove Iniziative	-	-	-	ShF5
_PCDMANI\$	Fino a data validita' di _PCMANI\$	-	-	-	ShF5
_PCGSAGE\$	Agente (dati percipiente)	-	-	-	ShF5
_PCFDP	% Fondo professionista	-	-	-	ShF5
_PCTPRE	Tipo prestazione	-	-	-	ShF5
_PCCATASTD\$	Codice catastale domicilio percipiente, 'F'cod.trib./n.iniziat.	-	-	-	ShF5
Gla/d					
_PCGLAD\$	GLA/D	-	-	-	F6
_PCALCOL	% Contributo Inps Collab.coord./Ven.porta porta	-	-	-	F6
_PCALDFT	% Deduzione forfetaria	-	-	-	F6
_PCATCOL	Codice attivita'	-	-	-	F6
Dati Enasarco					
_PCENAINIRA\$	Data inizio rapporto, se e' presente significa che il Soggetto Enasarco e' = S	-	-	-	ShF8
_PCENAFINRA\$	Data fine rapporto	-	-	-	ShF8
_PCENAVARRA\$	Data variazione rapporto	-	-	-	ShF8
_PCENATPATT\$	Tipo mandato attuale	-	-	-	ShF8
_PCENAIMPTR(3,4)	Progressivo imponibile, sola lettura. Nota per la dimensione a 3 elementi: 3=ultimo anno gestito; 2=penultimo anno; 1=terzultimo anno	-	-	-	ShF8
_PCENAIMPRT(3,4)	Rettifica imponibile. Nota per la dimensione a 3 elementi: 3=ultimo anno gestito; 2=penultimo	-	-	-	ShF8

		anno; 1=terzultimo anno				
	_PCENAMDFIR(3,4)	Rettifica manuale imponibile FIRR. Nota per la dimensione a 3 elementi: 3=ultimo anno gestito; 2=penultimo anno; 1=terzultimo anno	-	-	-	ShF8
	_PCENATPMAN\$(4)	Tipo mandato corrispondente, sola lettura	-	-	-	ShF8
	_PCENACONTR(4)	Contrib. Calc. Da documenti, sola lettura	-	-	-	ShF8
	_PCENADOVUT(4)	Contrib. Dovuto con compens.,sola lettura	-	-	-	ShF8
	_PCENAINTEG(4)	Integrazione dovuta, solo lettura.	-	-	-	ShF8
Recapito posta/Sede amministrativa-legale						
	_PCTPRECAP2\$	Tipologia recapito P,A,L	-	-	F3	F3
	_PCNOME2\$	Nominativo	-	-	F3	F3
	_PCIND2\$	Indirizzo	-	-	F3	F3
	_PCCAP2\$	CAP	-	-	F3	F3
	_PCLOC2\$	Localita'	-	-	F3	F3
	_PCPROV2\$	Provincia	-	-	F3	F3
	_PCINDEFF\$	Riporta su effetti N,S,R,E	-	-	F3	F3
Progressivi/Saldi						
	_PCRATING\$	Rating assegnato; ""(default); "0"(Non definito); "1"(Affidabile); "2"(Certo); "3"(Incerto); "4"(Inaffidabile). Il campo va gestito manualmente.	-	-	F5	F5
	_PCINS	Numero insoluti	-	-	F5	F5
	_PCVIN	Valore insoluti	-	-	F5	F5
	_PCFIDO	Fido in valuta di gestione; se negativo (<0) significa che è in % sul fatturato anno precedente; l'originaria variabile _PCFID prevale su _PCFIDO, pertanto azzerare _PCFID prima di utilizzare _PCFIDO.	-	-	F5	F5
	_PCFID	Fido espresso milioni di Lire o migliaia di Euro. Variabile a valori unitari e multipli di mille (se in Lire). La variabile è obsoleta, utilizzare la variabile _PCFIDO.	-	-	F5	F5
	_PCFIDTPMSG\$	Messaggio fuori fido doc. magaz. N/A/B	-	-	F5	F5
	_PCGGRIT(2-13)	Tempo medio ritardo incasso pagamento giorni; campo raggiungibile dalla voce "Ritardi medi" [F4]. Rappresenta la differenza espressa in giorni tra la data di pagamento reale e quella prevista per ciascun tipo pagamento.	-	-	F5	-
Note						
	_PCNDSS\$(2)	Descrizione nota	-	-	ShF6	ShF6
	_PCNDE\$(2)	Destinazione nota	-	-	ShF6	ShF6
	_PCNDT\$(2)	Scadenza data nota	-	-	ShF6	ShF6
Dati Aggiuntivi						
	_PCVDR\$(26)	Videata anagrafica riportabile - 26 campi (*a)	ShF8	ShF8	ShF8	ShF8
	_PCVDF\$(26)	Videata anagrafica non riportabile - 26 campi (*a)	ShF8	ShF8	ShF8	ShF8
	_PCWWW\$(26,24)	Videate aggiuntive - 26 videate con 24 campi; l'istruzione GET converte da num → str\$, la PUT viceversa; _PCWWW\$(numero videata, numero campo) è utilizzata anche per videate Passweb	ShF8	ShF8	ShF8	ShF8
Altri dati conto - Stato anagrafica/Altro						
	_PCDCR\$	Data creazione conto	ShF11	ShF11	ShF11	ShF11
	_PCDUM\$	Data modifica conto	ShF11	ShF11	ShF11	ShF11
	_PCFMO\$	Conto modificato S/N	ShF11	ShF11	ShF11	ShF11
	_PCMRA\$(12)	Stato anagrafica attiva; v. mirroring (DBmirror); v. PassWeb; (*a)	ShF11	ShF11	ShF11	ShF11
	_PCMRV\$(12)	Stato anagrafica variata; v. mirroring (DBmirror); v. PassWeb; (*a)	ShF11	ShF11	ShF11	ShF11
Dati Sistema						
	_PCDTSIST\$	Data sistema ultima scrittura fisica	-	-	-	-
	_PCORSIST\$	Ora sistema ultima scrittura fisica	-	-	-	-
	_PCANN\$	Annulla/Ripristina - Conto Precancellato	ShF7	ShF7	ShF7	ShF7
<p>(*a) Vedere tabella in menu 'AES' e tasto funzione [Shift+F11] "Stato anagrafica" in Anagrafica conto. La funzione potrebbe non essere disponibile in determinate configurazioni (es. Live). Queste variabili sono condizionate dalla inizializzazione di _PCSOLOPICO\$="S/N". Vedere le indicazioni riportate nella sezione della PUTPC. Notare: in presenza di raggruppamento, la possibilità di identificare conti come di tipo banca, non passa piu' solo dall'attribuzione di _PCCLAS\$="B" (che distingueva rispetto a conti di tipo Cassa, per esigenze di flussi finanziari), ma passa anche dal tipo di Mastro, per il quale e' attiva l'attribuzione come "Depositi Bancari". Se infatti nel raggruppamento, un Mastro patrimoniale appartenente a un gruppo visibile in Anagrafica Piano dei Conti viene marcato come "Ban" (cioe' come Deposito Bancario), questo Mastro (e relativi conti) sara' visualizzato in Rubrica clienti/fornitori/banche, perche' appartenenti alla specifica azienda in questione. Se il raggruppamento è su piano dei conti riservato (80/81), i conti non riservati (quindi personalizzabili) sono quelli con mastro superiore al 69.</p>						
<p>(*b) contabilizzazione parcelle per aziende Studio con Collage</p>						
<p>(*c) nei conti dei Clienti la variabile si attiva nei parametri di magazzino tramite "Gestione ritenuta d'acconto nei documenti emessi".</p>						
<p>(*d) In Sprix è dizionario "padsp" (per rudt) e "pcdsp" (per pico). Identifica il periodo di inclusione nella gestione dello Split Payment per soggetti di tipo non ente pubblico. I primi due elementi sono il primo periodo gestione (la data di inizio periodo è obbligatoria e non inferiore alla data 01/07/2017), il 3° e 4° corrispondono all'eventuale secondo, 5° e 6° all'eventuale terzo. La data iniziale di un periodo è obbligatoria, la data finale di un periodo, se omessa, significa che questo è attivo. La gestione delle date avviene nelle "Condizioni e automatismi" con la voce "Split payment soggetti non Ente pubblico". Lo Split Payment è attivo per il solo Cliente (se lo stesso soggetto è anche Fornitore, per quest'ultimo c'è la sola visualizzazione).</p>						
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui						

_PN - Primanota

_PN

Variabili di struttura archivi di Primanota (GETPN – GETATPN – PUTPN – PUTPNX – DELPN)	
_PNOPE	Numero operatore, 0 = AUTO;
_PNSAZ	Codice sottoazienda – solo GET -, vedere APRIAZIE;
_PNPRN	Numero progressivo – solo GET – vedere <prinr>;
_PNDRES\$	Data registrazione;
_PNCAUS\$	Causale contabile;
_PNTDES\$	Descrizione di testata;
_PNCCR(1)	Centro costo/ricavo di testata (vedere anche _PNCCR di riga); variabile disponibile solo in forma di array (righe corpo documento), in quanto è il dato propositivo di default di riga e nella GETATTMM è valorizzato al ON_IN del campo successivo a CICCCR. In assenza di centro specifico di riga, quello di testata è il default di riga (per averlo in testata va ripetuto il centro di testata in ciascuna riga).
_PNRPRS\$	Registro protocollo Iva;
_PNSPR	Numero serie protocollo
_PNNPR	Numero protocollo, 0 = AUTO
_PNNDO	Numero documento, 0 = AUTO
_PNNUMRES\$	Alfanumerico documento; vedere le note della variabile _MMNUMRES\$;
_PNDDOS\$	Data documento;
_PNVRP\$	Versione Documento se prima della 2010G torna "S";
_PNPVEN	Punto Vendita;
_PNDST\$	Data statistica;
_PNDTD\$	Data decorrenza pagamento;
_PNDFF\$	Data pagamento Fattura esigibilità differita;
_PNRIT\$	Flag registrazione Iva Art. 27;
_PNDRV\$	Data riferimento vidimazione ufficio tributario Repubblica San Marino;
_PNPAG	Codice pagamento; solo PUT per generare rate scadenziario (non è leggibile in quanto non è memorizzata); non è disponibile in scrittura per la causale "NE" (nota di accredito);
_PNPCAS\$	Causale documento in pagamento;
_PNPSD	Serie documento in pagamento;
_PNPND	Numero documento in pagamento;
_PNPDD\$	Data documento in pagamento;
_PNTDER\$	Nota registro [F4] Testata;
_PNCOMI\$	Competenza iva periodo precedente accetta valore "S"/"N"
_PNDPDS\$	Documento da escludere dalla presentazione della distinta RSM. valori ammessi S/N; variabile di testata, solo per aziende RSM, dizionario pndpd
PROPRIETÀ DOCUMENTO (solo lettura)	
_PNDATCRE\$	Data creazione del documento;
_PNORACRE\$	Ora creazione del documento;
_PNUTECRE\$	Utente creazione del documento;
_PNMENCRES\$	Menu creazione del documento;
_PNDATMOD\$	Data ultima variazione documento;
_PNORAMOD\$	Ora ultima modifica del documento;
_PNUTEMOD\$	Utente ultima modifica del documento;
_PNMENMOD\$	Menu ultima modifica del documento;
PROTOCOLLO VENDITE	
_PNRPS\$	Registro;
_PNSPS	Numero serie;
_PNNPS	Numero documento;
_PNINT\$	Intra CEE (S/N); considerare la marcatura della tipologia di Cliente (CEE oppure no)
_PNVAL	Codice valuta estera;
_PNDSV\$	Descrizione valuta;
_PNEST	Importo valuta;
_PNCMR	Codice mandato RID SEPA; codice RID associato alla registrazione e di sola lettura, in quanto l'impostazione del codice RID avviene esclusivamente dalla funzione di emissione effetti
VARIAZIONE NOTA	
_PNNVA\$	Nota variazione anni precedenti [F6];
CATEGORIA BILANCIO	
_PNBIL\$	Categoria bilancio specializzato [SHF6];
VALUTA DOCUMENTO	
_PNVDO	Valuta documento; (*a)
_PNEVE + _PNDIE	Cambio Valuta Documento → Euro + (Flag cambio 0=No/1=Si); (*a)
_PNEVC + _PNDIC	Cambio Euro → Valuta Gestione + (Flag cambio 0=No/1=Si); (*a)
BLACK LIST	
_PNDCOBL\$	Periodo di competenza;
_PNIOPBL\$	Includi operazioni nel elenco;
_PNNVABL\$	Nota di variazione operazioni dell'anno;
_PNNVPBL\$	Nota di variazione operazioni anni precedenti;
_PNCTRBL\$	Codice cliente blacklist;
_PNIEPBL\$	Escludi operazioni blacklist;
SPLIT PAYMENT	
_PNILSP\$	Includi in liquidazione operazione Split Payment;
IVA MOSS	
_PNNPPM\$	Nota credito riferita a trimestri precedenti (S/N);

SPESE SANITARIE	
_PNPTT\$	Pagamento Tracciato (elaborazione spese sanitarie) Tessera sanitaria (S/N): dizionario pnppt()
_PNEDT\$	Escludi operaz. da comunicazione delle spese sanitarie (S/N);
IVA OSS	
_PNPAEO\$	Codice paese gestione IVA OSS; se il paese destinazione beni se diverso dal cliente intestatario della fattura
_PNESO\$	Codice iva per liquidazione/dichiarazione – calcolo Plafond
SPESE FUNEBRI	
_PNEDSF\$	Escludi dalla comunicazione delle spese funebri (S/N);
_PNCFSF\$	Codice Fiscale del defunto;
RIGHE DOCUMENTO	
_PNCTO\$(200)	Codice conto;
_PNCCR(200)	Centro costo/ricavo di riga (vedere anche _PNCCR di testata);
_PNCTF\$(200)	Riferimento al conto di costo/ricavo per dichiarazioni iva;
_PNDES\$(200)	Descrizione;
_PNIMD(200)	Importo contabile;(*a)
_PNMDO(200,4)	Imponibile iva;(*a)
_PNIVD(200,4)	Imposta; (*a)
_PNIMP(200)	Importo contabile;
_PNTPM(200)	Numero tipo merce, variabile di riga conto, solo per aziende RSM con data inizio nuova gestione RSM (parametri contabili). Il dato viene gestito solo per i documenti FE/NE/FR/NR aventi data registrazione dell'anno data inizio nuova gestione RSM o successiva. Dizionario pntpm
CASTELLETTO IVA	
_PNIMB(200,4)	Imponibile iva;
_PNIVA(200,4)	Imposta iva;
_PNALI\$(200,4)	Aliquota iva;
_PNALD\$(200,4)	Aliquota iva ventilazione differita;
_PNCND\$(200,4)	Castelletto no doppio protocollo (S/N);
SPESOMETRO	
_PNEDPE\$	Escludi documento per pagamento elettronico (S/N); Equivalente a flag "Escludi doc. da comunicaz. dati fatture transfrontaliere-Esterometro", dizionario pnepe()
_PNFCS\$	Forza comunicazione se documento inferiore al limite (F/P/N);
_PNDIMU\$	Documento importo unitario (A/S/N); (*d)
_PNIMTOSP	Importo totale dell'operazione; (*d)
_PNIITOSP	Imposta su totale operazione; (*d)
_PNCIC\$()	Cointestatari spesometro – conto;
_PNCIP()	Cointestatari spesometro – percentuale;
_PNNVR\$(200,4)	Nota di variazione spesometro (S/N);
_PNRSVA(200)	Riferimento documento serie per nota di variazione;
_PNRNVA(200)	Riferimento documento numero per nota di variazione; (*d)
_PNRDVA\$(200)	Riferimento documento data per nota di variazione;
_PNPSE\$(200)	Prestazione di servizio; (*d)
_PNORCO\$(200)	Operazione riferita a contratto (S/N);
_PNECAC\$(200)	Escludi castelletto da comunicazione (S/N);
_PNCFPI\$(200)	Codice fiscale / Partita Iva;
_PNCCS\$(200)	Codice cliente;
_PNCCTR\$(200)	Codice contratto; se non sono previste istruzioni sull'archivio contratti, non è controllata l'esistenza contratto;
_PNROPE\$(200)	Riferimento operazione;
_PNFRIMP(200)	Importo fattura da rettificare imponibile; (*d)
_PNFRIMPO(200)	Importo fattura da rettificare imposta; (*d)
_PNRDN\$(200)	Riferimento documento numero esteso nota di variazione;
_PNTPC\$(200)	Tipo contratto;
_PNTNL\$(200)	Tipo noleggio leasing;
_PNOTU\$(200)	Operazione legata al turismo;
_PNTRC(200)	Tipo operazione reverse charge;
_PNDCP\$(200)	Data competenza plafond;
RIFERIMENTI PAGAMENTI PS	
_PNEDC\$(200)	Codice conto;
_PNEDS(200)	Serie documento in pagamento;
_PNEDN(200)	Numero documento in pagamento;
_PNEDD\$(200)	Data documento in pagamento;
CONTABILITÀ ANALITICA	
_PNSCO\$(200,50)	Codice analitico;
_PNSCM\$(200,50)	Commessa;
_PNSAN(200,50)	Anno;
_PNSIM(200,50)	Importo;
_PNSIV(200,50)	Importo in valuta estera;
CONTABILITÀ ANALITICA (CDG)	
_PNGCO	Collegamento Analitica (0=NO, 1=SI); (*b)
_PNSCO\$(200,15)	Codice analitico; (*b)
_PNSCM\$(200,15)	Commessa; (*b)
_PNARE\$(200,15)	Codice area; (*b)
_PNDRI\$(200,15)	Data riferimento; (*b)
_PNQTA(200,15)	Quantità; (*b)
_PNSIM(200,15)	Importo; (*b)
DATI INTRACEE	

_PNAOP(50)	Ammontare;
_PNNTR\$(50)	Natura transazione;
_PNNTB\$(50)	Natura transazione colonna B; gestione 2022 per i documenti con data registrazione 2022 o superiore e per le aziende con periodicità intra mensile (parametri contabili)
_PNNCM(50)	Nomenclatura combinata;
_PNAOV(50)	Ammontare operazioni in valuta;
_PNMNE(50)	Massa netta;
_PNUSP(50)	Unita' suppletiva;
_PNVST(50)	Valore statistico;
_PNRGM\$(50)	Condizioni consegna –ex regime statis-;
_PNMTR\$(50)	Modalità trasporto;
_PNPDE\$(50)	Paese destinazione/provenienza;
_PNPPO\$(50)	Provincia/paese origine;
_PNPOR\$(50)	Paese origine (cessioni beni); gestione 2022 per i documenti con data registrazione 2022 o superiore e per le aziende con periodicità intra mensile (parametri contabili)
_PNPDS\$(50)	Provincia destinazione;
_PNIEU\$	Importi Intra in Lire(N) o Euro(S) solo GET; automatica da PUT, se anno < al 2001 o valuta di gestione in Lire è impostata N, altrimenti S;
DATI INTRA Sez.2	
_PNMES(50)	Mese di riferimento 1-12;
_PNTRI(50)	Trimestre di riferimento;
_PNANN(50)	Anno di riferimento;
_PNSTA\$(50)	Codice Iso paese;
_PNCIV\$(50)	Codice Iva del conto;
_PNSGN\$(50)	Segno degli importi;
_PNNAT\$(50)	Natura transitoria;
_PNAOE(50)	Ammontare operazione in Euro;
_PNAVA(50)	Ammontare operazione in valuta;
_PNVAS(50)	Valore statistico;
_PNNOC(50)	Nomenclatura combinata;
PROVVIGIONE	
_PNAGES\$	Codice Agente;
_PNPRA\$(15)	Codice Agente;
_PNPRO	Importo provvigione complessivo;
_PNPRI(15)	Importo provvigione complessivo;
_PNPRP	Importo provvigione pagato;
_PNPRG(15)	Importo provvigione pagato;
_PNPRD	Importo provvigione complessivo in valuta estera (gestione mono agente, non valorizzata in GETATTPN);
_PNMPD(15)	Importo provvigione per agente in valuta estera (gestione multi agente, non valorizzata in GETATTPN);
RITENUTA D'ACCONTO	
_PNCTR	Codice tributo Rda [n]NNNN, se di 5 cifre Ritenuta d'Acconto [n] è indice tabella aliq. Iva associabile al trib.;
_PNDPT\$	Documento totalmente pagato; valori S/N; Ritenuta d'Acconto se = N l'assenza dello specifico conto automatico "Erario c/rit. Provvisorio" blocca la registrazione PUTPN;
_PNIRA	Importo Iva Rda;
_PNNS1	Imponibile non soggetto a Rda;
_PNNS2	Spese non soggette a Rda;
_PNFNS	Fondo professionisti non soggetto a Rda;
_PNSOG	Imponibile Rda;
_PNARA	Aliquota Rda;
_PNRIT	Importo Rda;
_PNDSC\$	Descrizione Rda;
_PNTIR\$	Tipo ritenuta (A=Acconto, I=Imposta);
ENASARCO	
_PNENA	Importo Enasarco;
_PNENZ	Importo Enasarco a carico dell'azienda;
_PNDCE\$	Data competenza Enasarco;
_PNENASN\$	Enasarco (S/N);
INPS CO.CO.CO.	
_PNRPR	Ritenuta previdenziale 1/3;
_PNIMA	Imponibile soggetto a contributo al netto deduzione;
_PNPED	Percentuale aliquota contributo;
_PNPDF	Percentuale deduzione forfetaria;
_PNIMC	Importo contributo;
_PNIAC	Importo contributo a carico committente 2/3;
_PNPAD\$	Periodo attività dal;
_PNPAA\$	Periodo attività al;
_PNCOA	Codice attività collaboratore;
_PNDPC\$	Data pagamento compenso;
_PNIDV	Importo contributo da versare;
_PNDVE\$	Data versamento;
_PNIMS	Imponibile soggetto contributo;
_PNPRS	Tipo prestazione –tabella TPRES-;
SCADENZARIO A PARTITE	
_PNPAR\$	Partita Generale; (*c)
_PNSCT\$(249)	Cli/For; (*c)
_PNSPA\$(249)	Partita; (*c)

_PNSDS\$(249)	Data Scadenza; (*c)
_PNSID(249)	ID Scadenza; (*c)
_PNSTP\$(249)	Tipo pagamento rata; (*c)
_PNSCA\$(249)	Causale documento; (*c)
_PNSSE(249)	Serie docum. Iva; (*c)
_PNSNU(249)	Numero documento; (*c)
_PNSDD\$(249)	Data documento; (*c)
_PNSIP(249)	Importo pagamento; (*c)

Note

(*a) Per le versioni precedenti la 2023D solo GETATTPN con valuta documento <> da quella di gestione contabile: se il documento è emesso nella stessa valuta di gestione dell'azienda il Cambio = 1 e Flag = 0. Dato un importo in valuta di partenza il Flag stabilisce come operare per calcolare l'importo nella valuta di arrivo: se Flag = 0 moltiplicare oppure se = 1 dividere. Esempio: triangolazione con area Euro (100\$ → Euro - Lire) dati i cambi \$:Euro = 1,1565:1 ed Euro:Lire = 1:1936,27 : (se _PNDIE=1 e _PNEVE=1,1565 → 100:1,1565 = 86,47 Euro; (se _PNDIE=0 e _PNEVE=0,86467 → 100*0,86467 = 86,47 Euro -cambio inverso-; Euro → Lire (_PNDIC=0 e _PNEVC=1936,27) 87,47 Euro*1936,27 = 167,429 Lire

(*b) Obsolete e sostituite da specifiche variabili di struttura _CDG del Controllo di Gestione, caricabili comunque da GETATTPN. Gli archivi di _CDG /_PN /_MM sono separati ma collegati (esempio: conto di riga, progressivi di documento/riga). GETATTPN carica tutte le possibili _CDG ma questo array monodimensionale è distribuibile in _PN (array bidimensionale) con ciclo sul progressivo di riga: Esempio:
ON_612_ALL: GETATTPN : FOR X=1 TO 4 : PRINT _CDGSCODAN\$(X)+"|"+_CDGCTOCG\$(X)+"|"+STR\$(_CDGPROGRIG(X),0,0)+"|"+_PNCTO\$(_CDGPROGRIG(X)) : NEXT

(*c) Solo scadenziario a partite: _PNPAR\$ è la PARTITA (variabile generale di 13 caratteri alfanumerici -14 nella gestione di sezionali estesi - non leggibili con GET, esempio "04FE01/000077") da assegnare a tutte le rate all'inserimento in scadenziario di nuovi documenti; se impostata, sostituisce quella automatica generata dal programma secondo i criteri di primanota (sia per importi positivi, che negativi); la selezione delle 1 - 249 rate obbliga l'indicazione del conto _PNSCT\$ e può effettuarsi in base ai seguenti **schemi di pagamento** delle rate per:

- per Partita: _PNSCT\$+_PNSPA\$+_PNSDS\$+_PNSID+_PNSTP\$+_PNSIP
- per Riferimenti Documento: _PNSCT\$+_PNSCA\$+_PNSSE+_PNSNU+_PNSDD\$+_PNSDS\$+_PNSID+_PNSTP\$+_PNSIP
- per Data Scadenza e/o Tipo Pagamento: _PNSCT\$+_PNSDS\$+_PNSID+_PNSTP\$

Il _PNSTP\$="M" è Rimessa, "=" è tutti; _PNSID è un progressivo da 0 a 255 (se > 0 obbliga l'indicazione di _PNSDS\$ altrimenti viene ignorato) e va impostato solo se nella stessa partita di un cli/for ci sono più scadenze nello stesso giorno. L'array precedentemente indicato (1-249) infatti non significa che l'indice rappresenta la singola rata, ma serve a identificare ciò che si vuole pagare (più conti di uno stesso documento, più documenti ecc. fino a un massimo di 249 operazioni di pagamento, ovvero il limite fisicamente raggiungibile). L'uso dei RIFERIMENTI esclude la ricerca per partita (se impostata vengono ignorati). Sull'operazione dei pagamenti vengono riportati in automatico i vari dati (Partita, Scadenza + ID,...) delle rate pagate (ereditati). I pagamenti si possono eseguire impostando i campi di selezione: se questi non vengono valorizzati viene generata una operazione di scadenziario per ogni cli/for presente nell'operazione di primanota, pari al relativo saldo contabile (come partita è assunta quella automatica generata dal programma). ATTENZIONE: questo importo è considerato solo se nel criterio di pagamento si specifica la Partita oppure i Riferimenti Documento, altrimenti viene ignorato. L'importo che guida l'operazione è sempre quello di primanota: un pagamento non può essere > al valore della rata pagata, e la somma degli importi immessi per ogni cli/for deve sempre quadrare col relativo saldo di primanota (somma degli importi _PNIMP di _PNCTO\$); in ogni caso, se la somma degli importi immessi è > al saldo di primanota viene eseguito il pagamento solo per l'importo corrispondente al saldo di primanota del cli/for (la parte eccedente viene ignorata); se la somma è < viene aggiunta nello scadenziario del cli/for una rata = alla differenza (la partita assunta è la variabile generale e se assente, quella automatica generata dal programma). Se l'importo pagamento non è impostato sono pagate tutte le rate del cli/for immesso in base ai campi di selezione Partita, Scadenza, Tipo Pagamento, anche in combinazione fra loro (AND). Se non è indicata la data scadenza i pagamenti vengono effettuati ad iniziare dalla rata più vecchia fino a raggiungere il saldo contabile di primanota del cli/for (fino a 249 rate per ogni cli/for).

Le rate dei documenti con Ritenuta D'Acconto (max 24) vanno interamente pagate (se pagata parzialmente la rata viene spezzata e per il residuo è creata automaticamente una specifica rata). Lo Scadenziario a Partite consente 250 rate per ogni cliente/fornitore esistente nelle singole operazioni di primanota, 24 rate per documenti con RDA, 255 rate nella stessa data scadenza per ogni partita dei singoli clienti/fornitori. L'array è limitata a 249 elementi poiché l'ultimo elemento è riservato al programma per gestire la quadratura del saldo contabile del cli/for presente nell'operazione di primanota con il saldo degli importi pagati dallo stesso cliente/fornitore.

Ricordare: gli array di struttura della gestione dello scadenziario a partite (_PNSCT\$(),_PNSPA\$(),_PNSDS\$()) sono elementi di selezione relativi ad una determinata partita, utilizzabili per dichiarare quale elemento "chiudere"; queste variabili, pertanto, non sono valorizzate in GETPN e non vanno dichiarate nei movimenti contabili che generano rate (es. FT).

Dalla v806 la PUTPN può compensare rate negative di scadenziario **non abbinare ad alcuna riga positiva** (es. la nota di credito). Valorizzando le variabili per i pagamenti _PNSCT\$(n), _PNSPA\$(n), _PNSDS\$(n), _PNSID(n), _PNSIP(n) si attribuiscono i riferimenti che consentono al programma di compensare la nota di credito. La compensazione di rate negative va fatta con importi negativi, per i pagamenti invece deve essere sempre positivo. La compensazione deve essere uguale/superiore all'importo della rata negativa da compensare (se inferiore non avviene la compensazione). E' possibile compensare più righe negative attraverso l'indice degli elementi delle suddette variabili array _PNSCT\$(n), _PNSPA\$(n), _PNSDS\$(n), _PNSID(n), _PNSIP(n).

Esempio: una 'NE' che in scadenziario risulta di -300,00 può essere compensata in scadenziario con un 'PG' di 300,00

```
_PNSCT$(1)="502.00001" 'conto
_PNSPA$(1)="20NE01/000014" 'partita
_PNSDS$(1)="20200104" 'data scadenza rata
_PNSID(1)=0 'id data scadenza rata
_PNSIP(1)=-300 'importo da compensare
```

E' possibile compensare righe negative e pagare righe positive nello stesso documento.

Esempio: compensazione di una nota di credito e pagamento di una fattura dello stesso importo in valore assoluto

Il cliente risulterà con importo a zero in primanota.

```
'compensazione
_PNSCT$(1)="502.00001" 'conto
_PNSPA$(1)="20NE01/000014" 'partita
```

_PNSDS\$(1)="20200104"	'data scadenza rata
_PNSID(1)=0	'id data scadenza rata
_PNSIP(1)=-300	'importo da compensare
'pagamento	
_PNSCT\$(1)="502.00001"	'conto
_PNSPA\$(1)="20FE01/000055"	'partita
_PNSDS\$(1)="20200404"	'data scadenza rata
_PNSID(1)=0	'id data scadenza rata
_PNSIP(1)= 300	'importo da pagare

A partire dalla versione 2023D la variabile _PNSIP() importo da pagare/compensare, deve essere valorizzata esclusivamente nella valuta del documento. Le compensazioni di rate negative e pagamenti, relative allo stesso conto Cli/For, non sono più ammessi all'interno dello stesso documento se questo e' in valuta diversa da quella di gestione (e' necessario creare piu' documenti distinti).

(*d) Variabile obsoleta e non più attiva dalla versione v550 (2011F)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_PU - Anagrafica Ubicazioni

_PU

Variabili di struttura archivi della Anagrafica Ubicazioni (GETPU – GETATTPU – PUTPU – DELPU)	
_PUID	Id ubicazione
_PUMAG	Codice Magazzino
_PUCOD\$	Codice Ubicazione
_PUCODALT\$	Codice alternativo ubicazione
_PUTIPOS\$	Tipo ubicazione ("S" = Standard, "E" = Servizio)
_PUDESC\$	Descrizione ubicazione
_PUSTATOS\$	Stato anagrafica (" " = Attiva, "A" = Annullata logicamente, "B" = Bloccata, "U" = Bloccata in uscita, "I" = Bloccata in ingresso)
_PUAREAS\$	Area ubicazione (" " = Stock, "C" = Carico, "P" = Picking, "A" = Scarico)
_PUPERCORSO	Percorso ubicazione
_PUQUANT	Quantità
_PULARG	Larghezza (6 decimali)
_PUALT	Altezza (6 decimali)
_PUPROF	Profondità (6 decimali)
_PUPESO	Peso in Kg (6 decimali)
_PUTMIN\$	Temperatura minima (1 decimale) (= " " se non gestito, altrimenti da "-100,0" a "+998,0"); v. _ARTMI\$
_PUTMAX\$	Temperatura massima (1 decimale) (= " " se non gestito, altrimenti da "-100,0" a "+998,0"); v. _ARTMA\$
_PULAT	Latitudine (6 decimali)
_PULNG	Longitudine (6 decimali)
_PUELEV	Elevazione (6 decimali)
_PUALTSUOLO	Altezza dal suolo (6 decimali)
_PUSTARES\$	Struttura articoli esclusiva
_PUARTES\$	Codice articolo esclusivo
_PUCSTES\$	Categoria Statistica esclusiva
_PUGMRES\$	Gruppo Merceologico esclusivo
_PUNATES\$	Natura esclusiva
_PUCNTE\$	Codice fornitore esclusivo
_PUPCODART\$	Codice primario articolo
_PUPDESART\$	Descrizione articolo
_PUPUM1\$	Unità di misura primaria
_PUPUM2\$	Unità di misura secondaria
_PUPINQ	Quantità inventario
_PUPINT	Quantità inventario articolo a taglie (Array di 32)
_PUPTIN	Tara inventario
_PUPCIN	Colli inventario
_PUPCAR	Quantità di carico
_PUPCAT	Quantità di carico articolo a taglie (Array di 32)
_PUPTCA	Tara carico
_PUPCCA	Colli carico
_PUPSCA	Quantità di scarico
_PUPSCT	Quantità di scarico articolo a taglie (Array di 32)
_PUPTSC	Tara scarico
_PUPCSC	Colli scarico
_PUPOFOR	Quantità ordini a fornitore
_PUPOFORT	Quantità ordini a fornitore articoli a taglie (Array di 32)
_PUPOCLIMP	Quantità ordini di clienti impegnati
_PUPOCLIMPT	Quantità ordini di clienti impegnati articoli a taglie (Array di 32)
_PUPOCLDAIM	Quantità ordini di clienti da impegnare
_PUPOCLDAIMT	Quantità ordini di clienti da impegnare articoli a taglie (Array di 32)
_PUPOCLAUTO	Quantità ordini di clienti automatici
_PUPOCLAUTOT	Quantità ordini di clienti automatici articoli a taglie (Array di 32)
_NPU	Numero di righe progressivi ubicazione

METADATI

_PUTDATCRE\$	Data di creazione
_PUTORACRE\$	Ora di creazione
_PUTUTECRE\$	Utente di creazione
_PUTMENCRE\$	Menu di creazione
_PUTDATMOD\$	Data ultima modifica
_PUTORMOD\$	Ora ultima modifica
_PUTUTEMOD\$	Utente ultima modifica
_PUTMENMOD\$	Menu ultima modifica
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_PZD - Condizioni Documento

_PZD

Variabili di struttura per funzione GETCONDDOC	
_PZDPRZ	Prezzo;
_PZDSCO\$	Sconto;
Provvigioni: gestione mono agente	
_PZDPRO	Provvigione;
Provvigioni: gestione multi agente (non implementata in Sprix Mobile)	
_PZDPROTP\$(5)	Tipo provvigione, "%" (percentuale) oppure "T" (totale);
_PZDPROFORM\$(5)	Formula provvigione, solo 1° elemento;
_PZDPROAGE\$(5)	Codice agente, Tipo F/P;
_PZDPROQUOTA(5)	Quota provvigione ripartita, totale quote = 100%;
_PZDPROCOND(5)	Condizione agente, solo 1° elemento;
_PZDPROMOD\$(5)	Modalità provvigione "R" ripartita "P" percentuale, solo 1° elemento;
_PZDPROBIMP\$(5)	Base imponibile di calcolo provvigione: "1" lordo, "2" netto
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_RT - Registratore Telematico (MXRT)

_RT

Variabili di struttura per funzioni MXRT (Registratore Telematico)	
_RTNUMCHGIO	Numero chiusura giornaliera. Valorizzata da RTCHIUSURAGIO.
_RTDATA\$	Data chiusura giornaliera. Valorizzata da RTCHIUSURAGIO.
_RTMATIC\$	Matricola registratore telematico. Valorizzata da RTCHIUSURAGIO.
_RTREPMAX	Numero reparti configurati per il registratore telematico. Valorizzata da RTGETREPARTI.
_RTREPNUM()	Numero reparto. Valorizzata da RTGETREPARTI.
_RTREPDES\$()	Descrizione reparto. Valorizzata da RTGETREPARTI.
_RTREPALQ()	Aliquota reparto. Valorizzata da RTGETREPARTI.
_RTREP NAT\$()	Natura reparto. Valorizzata da RTGETREPARTI.
_RTREPESE\$()	Codice esenzione reparto. Valorizzata da RTGETREPARTI.
_RTPAGMAX	Numero pagamenti configurati per il registratore telematico. Valorizzata da RTGETPAGAMENTI.
_RTPAGNUM()	Numero pagamento. Valorizzata da RTGETPAGAMENTI.
_RTPAGDES\$()	Descrizione pagamento. Valorizzata da RTGETPAGAMENTI.
_RTPAGCOD()	Codice pagamento Passepartout. Valorizzata da RTGETPAGAMENTI.
_RTIDDEF	Identificatore registratore telematico di default impostato per il client (0 = non impostato). Valorizzata da RTELENCODISP.
_RTNUMDISP	Numero di registratori telematici configurati. Valorizzata da RTELENCODISP.
_RTID(..)	Identificatore registratore telematico (numero). Valorizzata da RTELENCODISP.
_RTNOME\$(..)	Nome registratore telematico. Valorizzata da RTELENCODISP.
_RTDISP\$(..)	Tipo dispositivo registratore telematico. Valorizzata da RTELENCODISP.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_SINP - Stringa in Input

_SINP\$

Variabile che alla chiamata della routine contiene la stringa del campo di Input , per esempio, al verificarsi di un messaggio di errore sul campo di Input la variabile contiene il medesimo messaggio (ON_ERR_CAMPO: _SINP\$="messaggio"). Modificando la variabile, questa contiene e inserisce nel campo il valore eventualmente modificato. Esempio all'ingresso del campo nota del documento di magazzino (ON_IN_MDNOTE: _SINP\$="inserisco" : END) oppure all'uscita dello stesso campo (ON_OUT_MDNOTE: _SINP\$="" : END).	
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_SC - Primanota da Import Moduli

_SC

Variabili di struttura per istruzione PUTSC (operazione su Primanota di Import Moduli)	
_SCCOD\$()	Codice conto;
_SCINIZ()	Saldo iniziale;
_SCDARE()	Saldo dare;

_SCAVER()	Saldo avere;
_SCSALD()	Saldo dare/avere;
_SCDATAAP\$	Data apertura;
_SCDATAACH\$	Data chiusura;
_SCOPER	Operatività funzione PUTSC: (0 = Scrittura record primanota, default); (1 = Cancellazione record in data apertura _SCDATAAP\$);
_SCNUMREG	Numero registrazioni PN eseguite;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_SN - Firma in Sprix Mobile

_SN

Variabili per istruzione GETSIGN (gestione firma di presa visione in Sprix Mobile)	
_SN???	<p>Variabili per la gestione della firma di presa visione relativa ad un dato documento. La firma verrà allegata al documento e, se necessario, sincronizzata lato gestionale con Docuvision.</p> <p>Riportiamo sinteticamente le variabili:</p> <ul style="list-style-type: none"> • _SNFILESIGN\$: documento collegato alla firma; se presente non verrà creato il documento dal form e ignorata la variabile _SNFORMID, ma verrà utilizzato il documento in locale su passmobile (es:pdf documento dopo srvsyncdoc o dopo una ACQUISISCIFILE). Se il file non esiste verrà restituito errore di file non presente nella variabile _ERRSN\$ • _SNVIEWDOC\$: default "S" ; se "N" non verrà visualizzata una preview il documento prima di acquisire la firma; il documento verrà comunque creato o tramite form id o acquisito da file locale. • _SNTYPEOP\$: tipo dell'operazione. Può assumere i seguenti valori <ul style="list-style-type: none"> ○ "DEL": per eliminare un documento salvato in locale e non ancora sincronizzato ○ "SYNC": è il default, ed indica che la firma e il documento dovranno essere sincronizzati con il server ○ "NOSYNC": indica che il documento e la firma rimarranno solo sul dispositivo mobile e non saranno mai sincronizzati con il server ○ "GET": il file della firma viene restituito nella variabile _SNFILESIGN\$ gestito nell'etichetta ON_GETSIGN solo se il documento non è già stato sincronizzato • _SNFILEDOC\$: nome del documento collegato alla firma • _SNCLASS: classe del file documento in docuvision <p>Per approfondimenti consultare il Manuale di Sprix Mobile.</p>

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_SCORRI - Condiziona la INPUT come una PRINT

_SCORRI

<p>Variabile che condiziona le successive INPUT; se _SCORRI=0 sono interpretate come delle INPUT, altrimenti (se _SCORRI=1) come delle PRINT con: azzeramento di _TF, la sola visualizzazione di <str\$> della INPUT, senza richiesta di immissione, senza variazioni di posizione della funzione o cambiamenti di colore del cursore.</p>
--

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_SORT_STR_CI - Ordinamento Caratteri Estesi

_SORT_STR_CI

<p>Variabile che impostata all'inizio del programma stabilisce il tipo di ordinamento in caso di caratteri estesi (es. lettere accentate): è case sensitive se = 0 (default) oppure case insensitive se = 1.</p>
--

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_SSD - Sincronizzazione Documenti Mobile-Server

_SSD

Variabili per istruzione SRVSYNCDOC (sincronizzazione documenti fra Mobile e Server)	
_SSDTPSTP\$	Se impostata = "E" esegue la stampa con invio email dalla stampante email opportunamente scelta nel modulo PassMobile.
_SSDNUMDOC	Nuovo numero del documento;
_SSDFILESTP\$	Percorso file risultato della stampa (_SSDFILESTP\$) da poter visualizzare con la VISUALIZZAFIILE;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_SST - Riga in stampa

_SST\$

Variabile con la riga in stampa.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

_STP - Contenuto di Stampa

_STP

Variabili di stampa valorizzate dopo le istruzioni comandi APRISTAMP o APRISTAMP_EXT e azzerate all'istruzione CHIUDISTAMP (nota: per approfondimenti consultare il capitolo STAMPE nella sezione Comandi per Argomento) Clicca qui	
_STPLUNGRIG	Lunghezza della pagina in caratteri di stampa (numero delle righe)
_STPLARGCOL	Larghezza della pagina in caratteri di stampa (numero delle colonne)
_STPLUNGRMM	Lunghezza della pagina in Millimetri
_STPLARGCMM	Larghezza della pagina in Millimetri
Variabili di stampa per le Stampe Personalizzate di Magazzino (evento ON_STP_DOC) (nota: per approfondimenti consultare il capitolo STAMPE nella sezione Comandi per Argomento) Clicca qui	
_STPMMRIG	Variabile utilizzata per le stampe personalizzate "Documenti di Magazzino" e "Movimenti di Magazzino". Riporta il numero di riga elaborata del documento elaborato.
_STPNO	Variabile relativa alle stampe personalizzate "Documenti di Magazzino" e "Movimenti di Magazzino". Impostata ad "1" esclude la riga dalla successiva stampa. Saranno escluse dalla stampa tutte le righe finché la variabile non viene nuovamente impostata a 0 (zero).
_STPGXN(1-20)	Variabile array numerica generale, sempre valorizzabile e modificabile; definisce un valore che sarà riportato alla fine dell'elaborazione nella relativa variabile di stampa;
_STPGXA\$(1-20)	Variabile array alfanumerica generale, sempre valorizzabile e modificabile; definisce un valore che sarà riportato alla fine dell'elaborazione nella relativa variabile di stampa;
_STPTXN(1-20)	Variabile array numerica di documento, attiva per ogni riga elaborata dalla stampa;
_STPTXA\$(1-20)	Variabile array alfanumerica di documento, attiva per ogni riga elaborata dalla stampa;
_STPRXN(R,1-10)	Variabile obsoleta oppure non più utilizzabile . Non considerarla. Variabile array numerica bidimensionale, attiva per le righe del documento (R = riga documento, elemento vettore); il funzionamento è identico alle variabili Collage_MM per la stampa del Modulo Documento Magazzino;
_STPRXA\$(R,1-10)	Variabile obsoleta oppure non più utilizzabile , Non considerarla. Variabile array alfanumerica bidimensionale, attiva per le righe del documento (R = riga documento, elemento vettore); il funzionamento è identico alle variabili Collage_MM per la stampa del Modulo Documento Magazzino;
_STPRWF(R,1-20)	Variabile array numerica bidimensionale, attiva per le righe del documento (R = riga documento, elemento vettore); il funzionamento è identico alle variabili Collage_MM per la stampa del Modulo Documento Magazzino;
_STPRW\$(R,1-20)	Variabile array alfanumerica bidimensionale, attiva per le righe del documento (R = riga documento, elemento vettore); il funzionamento è identico alle variabili Collage_MM per la stampa del Modulo Documento Magazzino;
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_TF - Codice Tasto

_TF

<p>Variabile che ritorna il codice numerico (dichiarato con SETTF prima della INPUT) del tasto utilizzato per uscire dal campo di INPUT o con l'evento ON_OUT_CAMPO.</p> <p>Assume valore 1=Invio (anziché 501), 1=Freccia su, -1=SH+Tab e -1=Freccia giù in assenza di SETTF e nel evento ON_OUT_CAMPO (in cui vale 1000 se cliccato il mouse). Codici numerici 6000 (+ num. riga) sono attinenti alla zona mouse.</p> <p>_TF è globale e modificabile: se cambiato il contenuto, viene considerato come tasto utilizzato per l'uscita dal campo in sostituzione di quello premuto dall'utente (esempio, annullamento pressione del tasto funzione F8: ON_OUT_MDQTA: IF _TF=608 THEN _TF=0 : END).</p> <p>Alla chiamata della routine collage _TF è automaticamente settata, se non è attivata la relativa etichetta _TF non è valorizzata.</p> <p>Nelle istruzioni MYDBVIEW e MYDBINPUT questa variabile è sostituita dalla _MYDBTF.</p> <p style="text-align: right;">Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>

_TMC - Colonna Finestra Mouse

_TMC

<p>Variabile che ritorna il numero di colonna della finestra mouse (coordinata colonna).</p> <p>Viene attivata nelle zone finestra mouse (v. SETFSM) che non siano interessate da SETZM e viene attivata in INPUT e INPUT_EXT con parametro <numzonamouse> = 0, ovvero _TMZ=0. La INPUT_EXT deve prevedere <numzonamouse> = 0 in partenza senza forzare a zero la _TMZ successivamente (non funzionerebbe).</p> <p style="text-align: right;">Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>
--

_TMR - Riga Finestra Mouse

_TMR

Variabile che ritorna il numero di riga della finestra mouse (coordinata riga).

Viene attivata nelle zone finestra mouse (v. SETFSM) che non siano interessate da SETZM e viene attivata in INPUT e INPUT_EXT con parametro <numzonamouse> = 0 (ovvero _TMZ=0). La INPUT_EXT deve prevedere <numzonamouse> = 0 in partenza senza forzare a zero la _TMZ successivamente (non funzionerebbe).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_TMZ - Numero Finestra Mouse

_TMZ

Variabile che ritorna il numero della zona mouse. La variabile è globale per tutte le finestre aperte in Sprix.

Vedere istruzioni SETFSM, SETZM e INPUT_EXT.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_TP - Distinta Base Primaria

_TP

Variabili di struttura archivi della Distinta Base Primaria (GETDB – PUTDB – DELDB) (le variabili evidenziate in grassetto sono quelle minime obbligatorie)

TESTATA	
_TPCOD\$	cod.articolo con distinta base
_TPSVI\$	articolo per calcolo costo
_TPCTR\$	controllo esistenza/disponibilità
_TPFPF\$	prodotto finito padre di default
_TPAGG\$	tabella colori di aggancio
_TPSML\$	categoria semilavorato
_TPUSA\$	usa DBVC/DBVO
_TPCOL\$	gestione collegamento PF->SML
_TPUMP\$	tipo unità di misura utilizzato (P o S)
_TPMMP	magazzino materie prime di default
_TPDEC	numero di decimali
_TPCLI\$	cod. cliente di default
_TPOAP	ore di approntamento
_TPORE	ore per lotto
_TPPNR	lotto di produzione
_TPKOE	coefficiente
_TPKOC(...,16)	consumi per taglia personali
_TPKOCXL(...,16)	consumi per taglia personali (Tg 17-32)
FASE	
_TPFAS	numero fase
_TPDES\$	descrizione della fase (se la descrizione non e' presente nella tabella "Fasi di lavorazione" avviene l'autocodifica; si veda anche la variabile _TPCICLO)
_TPFSC\$	tipo condizione della fase
_TPULT\$	ulteriore descrizione della fase
_TPNQT\$	quantità per riferimento
_TPDOC\$	documenti in stampa
_TPNOT\$(1-25)	gruppo di 10 note della fase da 40 caratteri (per versioni < v816) oppure gruppo di 25 note della fase da 40 caratteri (per versioni >= v816)
_TPLAG	minuti di ritardo/anticipo (valori positivi/negativi) - gestita in fase di scrittura dell'App Planning
_TPYTP\$	tipo relazione; stringa che accetta valori da "0" a "3" - gestita in fase di scrittura dell'App Planning
_TPCTL\$	gestione conto lavoro S/N
_TPFOL\$	fornitore conto lavoro
_TPMAL	magazzino conto lavoro
_TPCICLO	ciclo lavorazione: equivale al campo codice della tabella "Fasi di lavorazione"; questo nuovo campo ha priorità su _TPDES\$ che rimane comunque obbligatoria per il mantenimento della retrocompatibilità
_TPCOLLEGA\$	variabile flag S/N che indica se la fase è eseguita su macchina
COMPONENTE	
_TPIND\$()	indice
_TPART\$()	cod. componente
_TPQTA\$()	quantità
_TPCOE()	coefficiente di conversione
_TPEVI\$()	evidenza o fase di disimpegno rapido
_TPNEC\$()	negazione cond. Cliente
_TPNED\$()	negazione cond. Data
_TPNET\$()	negazione cond. Taglie
_TPSS\$(,n)	array art. sostitutivi - tutti e 4 -
_TPMAG()	magazzino componente - in standby -

_TPNEG\$()	negazione condizioni
_TPCONS\$()	tipo condizione "A" o "S"; Nota: accettava anche valori "C", "T", "D" e pertanto, se tali valori sono utilizzati nel sorgente in essere, questi dati sono ignorati dalla istruzione (es. la PUTDB non va in errore ma ignora); se presente una IF che testa queste valorizzazioni occorre modificare la IF altrimenti la condizione viene ignorata.
_TPCAR\$()	condizione (cod.articolo)
_TPUNM\$()	unità misura utilizzata: 1 -prim- o 2 -second-
_TPCCL\$()	cod. cliente condizione
_TPCDD\$()	da data
_TPCDA\$()	a data
_TPCNOOn\$()	Note del componente; l'elemento 10 non è disponibile ed è sostituito dallo zero (0): <ul style="list-style-type: none"> (versioni >= v816): nota componente, 'n' assume valore da 0 a 25 ("n=0" equivale a "n=10") (versioni < v816): nota componente, 'n' assume valore da 0 a 9 ("n=0" equivale a "n=10")
_TPTAG\$()	condizione per taglia
_TPCT\$()	tipo sostituzione componente
_TPSOS\$()	articolo in sostituzione
_TPDIR\$()	direzione P=precedente S=successiva
_TPlot\$()	spezza lotto; se non impostato (cioè viene lasciato pari a "") allora viene azzerato (errato) e rilascia messaggio "tipo spezza lotto errato azzerato"
_TPEDE\$()	escludi da deposito S/N (GET/PUT)
_TPESC\$()	escludi da scarico S/N (GET/PUT)
_TPEST\$()	lavorazione esterna S/N (GET/PUT)
_TPCST\$()	tipo di costo - U,S,M,V,P - (GET/PUT)
_TPCOS\$()	costo lavorazione (GET/PUT)
_TPEFO\$()	codice Fornitore (GET/PUT)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_TPlot - Anagrafica Tipo Lotto

_TPlot

Variabili di struttura archivi di Anagrafica Tipo Lotto (GETTPlot – PUTTPlot – DELTPlot)	
_TPlotCOD\$	(Codice tipo lotto; "" se nuovo inserimento)
_TPlotDESC\$	(Descrizione tipo lotto)
_TPlotSTAT\$	(Stato tipo lotto)
_TPlotCUNI\$	(Chiave utente univoca(S/N))
_TPlotMATR\$	(Tipo lotto matricola: '' = Lotto, 'M' = Matricola)
PARAMETRI LOTTO	
_TPlotAPGS\$	(Gestione: E/D/e/d)
_TPlotAtp\$	(Tipo F/L/C/D)
_TPlotAPCO\$	(Mantenere colli quantità S/N)
_TPlotAPCL\$	(Mantenere colli peso lordo S/N)
_TPlotAPSL\$	(Spezza quantità lotto S/N)
_TPlotAPQR\$	(Quantità residua lotto N/A/C)
_TPlotSCAQ\$	(Avviso quantità insuff.: N/E/D)
_TPlotCATP\$	(Tipo controllo M/F/D/A)
CAMPI PERSONALIZZATI	
_TPlotDUCD\$(999)	(Codice tipo lotto – se nuovo, i primi 2 car. Sono "" -
_TPlotDUET\$(999)	(Etichetta)
_TPlotDUTP\$(999)	(Tipo)
_TPlotDUDI(999)	(Dimensione)
_TPlotDUOB\$(999)	(Obbligatorio S/N)
_TPlotDUOR(999)	(Riga ordine visualizzazione)
_TPlotDUVL\$(999)	(Valore iniziale default –anche nome file-)
_TPlotDUCH(999)	(num elem. Cod. utente (chiave) (max 63)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_TSST - Array di Stampa

_TSST\$()

Variabile Array con la riga in stampa.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

_VERSAPP - Versione Applicazione

_VERSAPP

_VERSAPP_PRE

Variabili numeriche di sistema [Collage controllo App] che ritorano la versione dell'Applicazione.
_VERSAPP rileva Applicazione <u>nuova</u> ed è sempre valorizzata.

_VERSAPP_PRE rileva Applicazione precedente (solo in fase di controllo applicazione); può assumere valore 0 in caso di assenza di versioni precedenti.

Gli eventi per agire con Collage controllo App sugli stati dell'App sono i seguenti:

- ON_NUOVA_APP:
- ON_MODIFICA_APP:
- ON_ELIMINA_APP:

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_WD - Messaggio in Sprix Mobile

_WD

Variabili di struttura Sprix mobile per la WCREATEDIALOG	
_WDTITLE\$	Titolo della finestra
_WDMESSAGE	Testo messaggio
_WDOID	Numero univoco della finestra

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_WF - Form in Sprix Mobile

_WF

Variabili di struttura Sprix mobile per la WCREATEFORM	
_WFTITLE\$	Titolo della finestra
_WFOID	Numero univoco della finestra non viene azzerata dalla relativa AZZVARSYS
_WFHDRTXT\$	Testo destinato all'Header del form; se = "" non è mostrata la barra di visualizzazione
_WFFTRTXT\$	Testo destinato al Footer del form; se = "" non è mostrata la barra di visualizzazione
_WFPCFMSG\$	Testo da visualizzare nel messaggio di chiusura del form. Se settata vuota (= "") non apparirà alcun messaggio. La gestione è fra eventi ON_PRECLOSE_<NOMEFORM> (evento alla chiusura del form) e ON_CLOSE_<NOMEFORM> (evento che scatta sempre, anche se la _WFPCFMSG\$ è settata = "")
_WFPCFEVT	Variabile Back Form che viene inizializzata nell'evento PRECLOSE. <ol style="list-style-type: none"> 1. _WFPCFEVT = 100 : back effettuato tra i form (freccia a sinistra <) 2. _WFPCFEVT = 1 : back effettuato premendo Esci , Home oppure cambio Sprix.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

_WI - Input in Sprix Mobile

_WI

Variabili di struttura Sprix mobile per la WCREATEINPUT (definizione di una casella di input)																	
_WITYPE\$	Tipologia del campo. Obbligatoria. Accetta i seguenti valori: <ul style="list-style-type: none"> • "TEXT" per i campi di input; • "BUTTON" per i pulsanti; • "WEB", "PHONE", "MAIL", "DATE"; se utilizzati viene mostrata anche una icona rappresentativa del tipo di input (es. icona web, telefono, e-mail e date time picker). • "ARCOD"; casella di input per gli articoli strutturati; viene mostrato un input sul quale cliccare per accedere alla maschera di inserimento dei codici strutturati; occorre definire un'etichetta per gestire l'evento che scatta all'uscita della maschera d'inserimento del articolo strutturato: l'evento è ON_ARCODSTRU_<codice_oggetto>. La definizione di questa etichetta è necessaria in quanto è l'unico punto in cui si possono gestire le variabili di output per l'articolo strutturato. Le variabili sono le seguenti sotto riportate: _WIARTARTNUM, _WIARTART\$, _WIARQTA(), _WIARTAGTAB(), _WIARTAG(), _WIARTAGXL(), _WIARAUTOCS(), _WIARERRATCS() <table border="1" data-bbox="513 1680 1391 1966"> <tbody> <tr> <td>_WIARTARTNUM</td> <td>Numero di codici articolo restituiti</td> </tr> <tr> <td>_WIARTART\$()</td> <td>Array contenente i codici articolo</td> </tr> <tr> <td>_WIARQTA()</td> <td>Array contenente le quantità impostate</td> </tr> <tr> <td>_WIARTAGTAB()</td> <td>Array con i numeri della tabella taglie</td> </tr> <tr> <td>_WIARTAG()</td> <td>Array con le quantità delle taglie da 1 a 16</td> </tr> <tr> <td>_WIARTAGXL()</td> <td>Array con le quantità delle taglie da 17 a 32</td> </tr> <tr> <td>_WIARAUTOCS()</td> <td>Per ogni codice articolo contiene: <ul style="list-style-type: none"> • "N" se l'articolo non è da auto codificare • "S" se l'articolo è da auto codificare • "E" se c'è stato un errore seguita dalla stringa di errore </td> </tr> <tr> <td>_WIARERRATCS()</td> <td>Numero di codici articolo restituiti</td> </tr> </tbody> </table> <p>Nota: nel caso di articoli strutturati è necessario importare anche i seguenti archivi: CART,DBCO,TAGLIE,STRUAR. Nell'archivio ARTI è necessario che siano importati i seguenti campi: _ARSTR, _ARCAR\$, _ARALT\$, _ARUM2\$, _ARRST, _ARTCOS\$, _ARCON, _ARPTA</p> <ul style="list-style-type: none"> • "ARQTA". Input che apre in automatico la schermata di inserimento delle taglie come nel carrello. 	_WIARTARTNUM	Numero di codici articolo restituiti	_WIARTART\$()	Array contenente i codici articolo	_WIARQTA()	Array contenente le quantità impostate	_WIARTAGTAB()	Array con i numeri della tabella taglie	_WIARTAG()	Array con le quantità delle taglie da 1 a 16	_WIARTAGXL()	Array con le quantità delle taglie da 17 a 32	_WIARAUTOCS()	Per ogni codice articolo contiene: <ul style="list-style-type: none"> • "N" se l'articolo non è da auto codificare • "S" se l'articolo è da auto codificare • "E" se c'è stato un errore seguita dalla stringa di errore 	_WIARERRATCS()	Numero di codici articolo restituiti
_WIARTARTNUM	Numero di codici articolo restituiti																
_WIARTART\$()	Array contenente i codici articolo																
_WIARQTA()	Array contenente le quantità impostate																
_WIARTAGTAB()	Array con i numeri della tabella taglie																
_WIARTAG()	Array con le quantità delle taglie da 1 a 16																
_WIARTAGXL()	Array con le quantità delle taglie da 17 a 32																
_WIARAUTOCS()	Per ogni codice articolo contiene: <ul style="list-style-type: none"> • "N" se l'articolo non è da auto codificare • "S" se l'articolo è da auto codificare • "E" se c'è stato un errore seguita dalla stringa di errore 																
_WIARERRATCS()	Numero di codici articolo restituiti																

	<ol style="list-style-type: none"> 1. In creazione dell'input impostare : <code>_WITYPE = "ARQTA"</code> <ol style="list-style-type: none"> a. Se <code>_WIARQTATAB\$ = "S"</code> → campo readonly, immagine taglie, apertura taglie con eventi b. Se <code>_WIARQTATAB\$ = "N"</code> → campo readonly, immagine tastiera numerica c. E' possibile modificare dinamicamente la proprietà tramite WSET 2. Eventi: <ol style="list-style-type: none"> a. ON_ARQTA_NOMEINPUT: Evento richiamato all'apertura dalla finestra delle taglie. b. In ingresso: <ol style="list-style-type: none"> i. <code>_WIARART\$(1)</code> = Codice Articolo ii. <code>_WIARTAG(1,16)</code> = taglie da 1 - 16 iii. <code>_WIARTAGXL(1,16)</code> = taglie da 17 - 32 3. ON_ARQTAOK_NOMEINPUT: Evento richiamato in uscita dalla finestra taglie 4. In uscita: <ol style="list-style-type: none"> a. <code>_WIARQTA(1)</code> = quantità totale delle taglie b. <code>_WIARTAG(1,16)</code> = taglie da 1 - 16 c. <code>_WIARTAGXL(1,16)</code> = taglie da 17 - 32 <p>NOTA BENE: Se input numerico è possibile recuperare il contenuto del campo tramite la variabile numerica <code>_WIVALUE</code> (ha lo stesso comportamento della funzione VAL di sprix - converte in numerico la stringa fino al primo carattere non numerico ES: VAL(13,33EURO) -> 13,33)</p> <p>La formattazione del valore numerico viene impostata tramite la variabile : <code>_WIUSERFMT\$</code> nel solito formato <code>“,2”</code>, <code>“,.”</code>, <code>“,....”</code></p> <ul style="list-style-type: none"> • "NUMBER". Input numerico (che apre in automatico la tastiera numerica). La formattazione del valore numerico viene impostata tramite la variabile : <code>_WIUSERFMT\$</code> nel solito formato <code>“,2”</code>, <code>“,.”</code>, <code>“,....”</code> • "DIVIDER". Viene mostrato un elemento separatore con etichetta impostabile in <code>_WIVALUE\$</code>. Crea linea di separazione (facoltativa) in cui è possibile specificare il testo; esempio <code>_WIVALUE\$="titolo"</code>. • "BARCODE". Viene mostrata l'icona che attiva la fotocamera per la scansione di codici a barre o QRcode: una volta acquisito il codice viene inserito automaticamente nel campo di testo. Vedere evento <code>ON_BARCODE_NOMEINPUT</code>: Clicca qui • "TIME". Viene mostrato un time-picker per la selezione dell'orario. • "TEXTAREA". Viene mostrata una textarea della dimensione di 4 righe per un totale di 255 caratteri. Il numero massimo di caratteri ammessi può essere limitato agendo su <code>_WILENCAR\$</code> • "CHECKBOX". Con relativo evento <code>ON_CHECK</code> alla pressione del checkbox; <code>_WIVALUE\$</code> viene inizializzata prima che sia richiamato l'evento
<code>_WIPARENTID</code>	Numero del Form che conterrà il campo. Obbligatoria.
<code>_WIUSERFMT\$</code>	Formattazione valore numerico
<code>_WIVALUE</code>	Contenuto del campo di tipo numerico;
<code>_WIVALUE\$</code>	Stringa testo visualizzata = "S" oppure "N"; per valore diverso oppure vuoto equivale al default = "N"
<code>_WIHINT\$</code>	Testo visualizzato in grigio all'interno del campo per aiutare l'utente alla sua valorizzazione. Facoltativa.
<code>_WIDESS\$</code>	Etichetta descrittiva del campo. Facoltativa.
<code>_WIOUTONLY\$</code>	Opzione campo. Facoltativa. Se = "S" è campo di sola lettura, ma l'eventuale icona associata al campo sarà attiva (tasto cliccabile); Se = "N": il campo e l'eventuale icona sono attivi; Se = "T": sia il campo di testo che l'eventuale icona associata saranno in sola lettura (tasto non cliccabile);
<code>_WICLEARFLD\$</code>	Cancella il contenuto di una casella di input read-only collegata a lista. Valore "S" o "N". In caso di casella di tipo read-only e collegata a lista, se valorizzata a "S", viene visualizzato un pulsante per cancellare il contenuto e far scattare l'ON_OUT.
<code>_WINOTEMPTY\$</code>	Stringa S/N facoltativa. Se valorizzata con "S" il campo risulta obbligatorio (bordato di rosso): vedi gestione del pulsante per convalida con <code>_WICHKEMPTYS\$</code> .
<code>_WILENCAR</code>	Lunghezza input. Serve per impostare il layout di un Form. Facoltativa, se valorizzata definisce la dimensione della casella di testo espressa in numero di caratteri. Se il campo non ha altri campi affiancati occuperà tutto lo spazio disponibile a prescindere dal valore impostato in questa variabile
<code>_WILINE\$</code>	Stringa InLine. Serve per impostare il layout di un Form (per l'input). Facoltativa, se valorizzata con "INLINE" la casella di testo viene affiancata alla precedente se lo spazio lo permette (utilizzare in accoppiata con <code>_WILENCAR</code>).
<code>_WIMAXCAR</code>	Quantità massima di caratteri in input; limita (tronca) l'immissione del testo fino a <code>_WIMAXCAR</code> . Se modificata successivamente con WSET, non si modifica il testo contenuto nell'input.
<code>_WILISTID</code>	ID lista collegata all'archivio. Facoltativa: deve essere valorizzato con l'id dell'eventuale lista che si vuole agganciare (vedi sezione dedicata).
<code>_WILISTNFLD</code>	Numero campo lista da visualizzare nell'input dopo aver scelto una riga dalla lista associata. Deve essere valorizzato con il numero del campo da mostrare una volta che si è scelto il record della lista associata
<code>_WIARQTATAB\$</code>	Tipologia immagine (taglie o tastiera numerica); valore "S" o "N". In caso di input strutturato decide se abilitare ("S") o meno ("N") l'input di quantità a matrice
<code>_WIOID</code>	Numero univoco del campo rilasciato dal sistema operativo (non è azzerata da AZZVARSYS)
definizione di un pulsante	
<code>_WIPARENTZN\$</code>	Zona del form in cui verrà visualizzato l'input. Definisce la zona (ZN) in cui inserire il pulsante: "ACTION": il pulsante viene inserito nella barra del titolo del form qualsiasi altro valore: il pulsante viene inserito in fondo al form in una pulsantiera dedicata, con scroll attivo in caso i pulsanti eccedano l'aera visibile
<code>_WIICO\$</code>	Per l'inserimento delle Icona <tipoico\$>. Esempio <code>_WIICO\$ = <tipoico\$></code> . <tipoico\$> rappresenta il codice dell'icona e può assumere i seguenti valori: "1" → "Ok"; "2" → "Annulla"

	<p>“3” → “Modifica” “4” → “Salva” “5” → “Aggiungi” “6” → “Elimina” “7” → “Stampa” “8” → “Copia” “9” → “Cerca documento” “10” → “Cerca contatto” “11” → “Cerca articolo” “12” → “Carrello” “13” → “Immagine”</p> <p>Nella definizione dell’input di tipo action può essere specificato l’ID di un altro INPUT di tipo action anziché quello del form. In questo caso il risultato è la realizzazione di un menù contestuale.</p> <p>Con questa variabile è possibile impostare delle icone personalizzate nei pulsanti form e nei pulsanti action impostando la proprietà di <code>_WICO\$</code> con il percorso dell’immagine da utilizzare.</p>
<code>_WICKEMPTY\$</code>	Se valorizzata con “S”, alla pressione del pulsante verrà inibito l’evento di <code>ON_PRESS</code> se nel form è presente almeno una casella di testo non compilata per cui era stato impostato <code>_WINOTEMPTY\$=“S”</code> . Il fuoco verrà dato al primo campo obbligatorio non compilato.
<code>_WIINSRIFID</code>	Riferimento del campo per <code>_WIINSPOSS</code>
<code>_WIINSPOSS</code>	<p>Posizione pulsante (“AFTER”, “BEFORE”, “BEFOREFIRST”, “AFTERLAST”) Esempio: <code>_WIINSPOSS\$=“AFTER”</code></p> <p>Con AFTER e BEFORE il pulsante viene inserito dopo o prima dell’input specificato con <code>_WIINSRIFID</code>; con BEFOREFIRST e AFTERLAST, invece, il pulsante viene inserito prima del primo input che trova o dopo l’ultimo</p>
ON_BARCODE	
<code>_WIBARCODE\$</code>	Array restituita all’evento di lettura barcode <code>ON_BARCODE_NOMEINPUT</code> :
<code>_WIBARNUM</code>	Numero elementi della Array restituita all’evento di lettura barcode <code>ON_BARCODE_NOMEINPUT</code> :
<code>_WIBARTYPE\$</code>	Stringa contenente il tipo di barcode
ON_ARCODSTRU	
<code>_WIARTNUM</code>	Numero di codici articolo restituiti
<code>_WIART\$()</code>	Array contenente i codici articolo
<code>_WIARQTA()</code>	Array contenente le quantità impostate
<code>_WIARTAGTAB()</code>	Array con i numeri della tabella taglie
<code>_WIARTAG(16)</code>	Array con le quantità delle taglie da 1 a 16
<code>_WIARTAGXL(16)</code>	Array con le quantità delle taglie da 17 a 32
<code>_WIARAUTO\$()</code>	<p>Per ogni codice articolo contiene:</p> <p>“N” se l’articolo non è da auto codificare “S” se l’articolo è da auto codificare “E” se c’è stato un errore seguita dalla stringa di errore</p>
<code>_WIARERRAT\$()</code>	Array che contiene l’eventuale errore dell’articolo n-esimo, altrimenti è vuoto
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

WL - Lista in Sprix Mobile

WL

Variabili di struttura Sprix mobile per la WCREATELIST (definizione di una lista)	
<code>_WLPARENTID</code>	Numero del Form che conterrà la lista. Ogni lista dinamica deve avere un suo form dedicato che contiene solo la lista;
<code>_WLPARENTZN\$</code>	Nome lista collegata al campo;
<code>_WLARCNAME\$</code>	Nome dell’archivio da collegare alla lista: vale solo per lista collegata ad archivio;
<code>_WLTOTFLD</code>	Numero dei campi di una riga della lista;
<code>_WLTOTROW</code>	Numero di righe della lista; numero totale delle righe da mostrare. Nel caso di lista collegata ad archivio questo parametro non è necessario.
<code>_WLARCFLD\$()</code>	Dati dell’archivio da visualizzare; Assegnazione del campo n-esimo al particolare elemento dell’archivio: vale solo per lista collegata ad archivio
<code>_WLDESFLD\$()</code>	Nome del campo della lista; Assegnazione di una etichetta al campo n-esimo della lista
<code>_WLHIDEFLD\$(n)</code>	Nasconde i campi di una lista
<code>_WLVALFLD\$()</code>	Valore del campo della riga. Assegnazione del valore al campo n-esimo per la riga attuale. Per indicare l’unione di più campi si indica “ <code>_ARCOD\$+ +_ARDESS\$</code> ”; in precedenza, per inserire spazi o testo fra questi dati si inseriva il carattere speciale “^”;
<code>_WLORDFLD()</code>	Assegnazione del campo che comanda l’ordinamento n-esimo (al momento è gestito solo 1 ordinamento). Impostato ad 1 specifica l’elemento della lista che comanda l’ordinamento;
<code>_WLORDDR\$()</code>	Accetta i valori “A”, “D” per determinare l’ordinamento della lista ascendente / discendente;
<code>_WLROWOK\$</code>	Accetta valore “S” o “N” se si il valore letto viene espresso nella lista altrimenti viene scartato; se valorizzato a “N” la riga attuale viene invalidata;
<code>_WLNUMROW</code>	Indice della riga corrente sulla quale riga si sta operando. Relazionata all’etichetta evento <code>ON_ROW</code>
<code>_WLENCARFLD</code>	Per impostare il layout di un Form (per le liste); definisce la lunghezza in caratteri del campo n-esimo
<code>_WLLINEFLD\$</code>	Per impostare il layout di un Form (per le liste); se valorizzato con “INLINE”, definisce se il campo n-

	esimo deve essere affiancato al precedente
_WLROWICO\$	Definisce se utilizzare un'icona per la lista. Nome dell'immagine catalogo definita per l'articolo specificata nella variabile di struttura relativa al catalogo _ARCAT\$. Esempio _WLROWICO\$ = _ARCAT\$. Si ricorda che Sprix Mobile può anche avvalersi di gestione del file tramite l'istruzione generale VISUALIZZAFI. Per l'inserimento delle icone nella definizione delle liste dinamiche (come le liste archivio): valorizzare _WLROWICO\$ = "S" e nella corrispondente etichetta ON_ROW_<nomelista> valorizzare il campo icona quella dell'archivio di anagrafica articoli _WLVALICO\$ = _ARCAT\$;
_WLVALICO\$	Definisce l'icona da utilizzare (solo per le liste dinamiche). Variabile per valorizzare il campo icona, ad esempio con quella dell'archivio di anagrafica articoli (vedere _WLROWICO\$); Con questa variabile è possibile impostare delle icone personalizzate nelle liste dinamiche impostare la proprietà di _WLVALICO\$ con il percorso dell'immagine da utilizzare nella riga.
_WLPOSICO\$	Posizione Icona; valori: "L","R","U","D" (Left (default),Right,Up,Down); in caso di lista carrello non è possibile posizionarla a Destra a causa dello stepper di incremento; se nella lista c'è un tasto, l'icona a destra non si può mettere; (a*)
_WLDIMICO	Dimensione Icona; valori: 1-10; default=1; dimensioni 1x, 2x, 3x e personalizzata (4-10); (a*)
_WLNELRIGA	Numero di elementi per riga; valori: 1, 2; default=1; un blocco oppure due blocchi; (a*)
_WLTPFLD\$()	Definisce il tipo di dato del campo n-esimo: STRINGA - NUMERO,dec - DATA. Variabile array per gestire l'ordinamento e filtri nelle liste dinamiche (come avviene per le liste collegate ad archivio). Il tipo di dato dell'ennesimo elemento dell'array può assumere i seguenti valori: <ul style="list-style-type: none"> • "STRINGA" (testo) • "NUMERO,dec" (numerico con numero di decimali dopo la virgola) • "DATA" (data da passare nel formato AAAAMMGG)
_WLICOFLG\$()	Definizione icona tipo flag se = "S". Relazionata all'etichetta evento ON_ROW
_WLFMTFLD\$()	Variabile di struttura per la lista. Se viene usata la stringa "." [cioè _WLFMTFLD\$(n) = "."] verrà visualizzato il numero con il separatore delle migliaia. Nelle liste di estensione è possibile scegliere una formattazione per la visualizzazione dei numerici, pur mantenendo integro il dato di tipo (cioè numerico). In questo modo, un campo numerico può essere mostrato a video con i punti di separazione delle migliaia, ma senza la necessità di impostare il campo come stringa; in questo modo, anche l'ordinamento per il suddetto campo, seguirà un ordine basato sul valore numerico e non sul valore in stringa;
_WLUSERORD\$	Variabile per gestire l'ordinamento nelle liste dinamiche. Se = "S" abilita l'ordinamento sulla lista; nota: se è = "S" la variabile _WLTPFLD\$() deve essere valorizzata per tutti gli elementi della lista dinamica;
_WLUSERFLT\$	Variabile array per gestire i filtri nelle liste dinamiche. Se = "S" abilita il filtro sulla lista; nota: se è = "S" la variabile _WLTPFLD\$() deve essere valorizzata per tutti gli elementi della lista dinamica;
_WLOID	Numero univoco del campo della lista. Non è azzerato da AZZVARSYS;
lista estesa collegata ad archivio	
_WLTOTFLDEXT	Numerico. Numero totale dei campi personalizzati della lista estesa
_WLNMFLEXT\$()	Nome del campo della lista estesa;
_WLTPFLEXT\$()	Tipo del campo della lista estesa. Valori accettati "STRINGA" "NUMERO,DECIMALI"(NUMERO,0) "DATA";
_WLVLFLEXT\$()	Valore del campo della lista estesa;
ON CART : - ON CARTQTA : - ON CARTOK :	
_WLTYPE\$	Variabile di attivazione Carrello . Assume valore = "CART". <ol style="list-style-type: none"> 1. In creazione della lista inserire _WLTYPE\$="CART" per attivare il carrello. 2. Alla pressione del prezzo mostriamo una finestra con sconto e totale di riga. 3. Per riaprire il carrello con le quantità preimpostate negli articoli utilizzare l'evento ON_CART richiamato alla visualizzazione del carrello (evento SHOWFORM o dopo il back dal form successivo).
_WLARPRZNFLD()	Variabile in uscita all'evento ON_CART_NOMELISTA: Indice dei campi che si vogliono nascondere nella lista di partenza e che non verranno riportati nella lista carrello
_WLARPRZFM\$	Variabile in uscita all'evento ON_CART_NOMELISTA: Formato da visualizzare (vale anche nel formato delle liste non carrello) <ol style="list-style-type: none"> 1. es: .,3<{prefisso} oppure ,3>{suffisso} oppure . oppure .,<{prefisso} etc.. (. = punto migliaia). 2. < imposta un prefisso 3. > imposta un suffisso
_WLARIMPDE\$	Variabile in uscita all'evento ON_CART_NOMELISTA: Etichetta del totale della riga;
_WLARQTATAB\$	Variabile in uscita all'evento ON_CART_NOMELISTA: Abilita/disabilita l'apertura dell'input matrice (default N)
_WLARARTNUM	Variabile in uscita all'evento ON_CART_NOMELISTA: Numero articoli preimpostati; In ingresso all'evento ON_CARTQTA_NOMELISTA: è impostato a 1; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) è il numero codici articolo restituiti;
_WLARART\$()	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente i codici articolo; In ingresso all'evento ON_CARTQTA_NOMELISTA: il primo indice è codice articolo; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene i codici articolo
_WLARCOL()	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente il numero di colli; In ingresso all'evento ON_CARTQTA_NOMELISTA: il primo indice è i colli dell'articolo; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene il numero dei colli;
_WLARQTA()	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente le quantità impostate; In ingresso all'evento ON_CARTQTA_NOMELISTA: il primo indice è la quantità dell'articolo; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene le quantità impostate;
_WLARTAGTAB()	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente il numero di tabella taglie; In ingresso all'evento ON_CARTQTA_NOMELISTA: il primo indice è la taglia dell'articolo; In ingresso

	all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene il numero della tabella taglie;
_WLARTAG(n,16)	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente le quantità di taglie 1 - 16; In ingresso all'evento ON_CARTQTA_NOMELISTA: quantità taglie 1 - 16; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene le quantità delle taglie 1 - 16;
_WLARTAGXL(n,17-32)	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente le quantità di taglie 17 - 32; In ingresso all'evento ON_CARTQTA_NOMELISTA: quantità taglie 17 - 32; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene le quantità delle taglie 17 - 32;
_WLARPRZ()	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente i prezzi singoli degli articoli; In uscita all'evento ON_CARTQTA_NOMELISTA: il primo indice è il prezzo articolo; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene i prezzi singoli degli articoli;
_WLARSCO\$()	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente gli sconti degli articoli; In uscita all'evento ON_CARTQTA_NOMELISTA: il primo indice è lo sconto dell'articolo; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene gli sconti degli articoli;
_WLARIMPVAL()	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore totale della riga con cui aggiornare il totale del documento; In uscita all'evento ON_CARTQTA_NOMELISTA: il primo indice è primo il totale della riga con cui aggiornare il totale del documento; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene il totale della riga con cui aggiornare il totale del documento;
_WLARPRO()	Variabile in uscita all'evento evento ON_CART_NOMELISTA: Vettore contenente i valori della provvigione; In uscita all'evento ON_CARTQTA_NOMELISTA: il primo indice è la provvigione; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene i valori della provvigione;
_WLARARTIND()	Variabile in uscita all'evento ON_CART_NOMELISTA: Vettore contenente i valori di indice ; In ingresso all'evento ON_CARTOK_NOMELISTA: (tutti gli articoli nel carrello) il vettore contiene i valori di indice (0 se nuova riga, altrimenti il valore passato nell'evento CART);
_WLCARTADD\$	Blocca l'inserimento di articoli dal carrello nell'evento ON_CARTQTA
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_WOP - Proprietà oggetto in Sprix Mobile

_WOP

Variabili di struttura Sprix mobile per la proprietà di un oggetto (WSET, WGET, WGETOID)	
_WOPNAME\$	Nome della proprietà;
_WOPVAL\$	Valore della proprietà alfanumerica;
_WOPVAL	Valore della proprietà numerica. La parameterizzazione WGETOID "PMMENU" carica la variabile con l'ID del menu di PassMobile al fine di poter aggiungere etichette (vedere WGETOID).
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_WP - Dimensioni in Sprix Mobile

_WP

Variabili di struttura Sprix mobile per l'istruzione WPRTFORMOPT	
_WPHEIGHT\$	altezza (mm , senza virgole) es: _WPHEIGHT\$ = "200"
_WPWIDTH\$	larghezza (mm , senza virgole) es: _WPWIDTH\$ = "100"
_WPDMFONT\$	dimensione font (mm, senza virgole) "{titolo}:{header}:{testo}:{footer}" esempio: _WPDMFONT\$ = "10:10:8:10"
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_WS - Chiamata a Web Services

_WS

Variabili utilizzate per avviare la comunicazione Web Services (CALLWEBSVC)	
_WSURL\$	URL del servizio (senza il protocollo che va indicato con apposita variabile)
_WSQRYSTR\$	Query string (parametri del servizio, cioè la stringa separata dall'URL dal carattere '?' costituita da serie di nome=valore separati da '&'). Es: www.google.it/?gfe_rd=cr&ei=-CE7V_WpCMykOreWk-AJ
_WSQRYSTRA\$()	Variabile vettore alternativa a _WSQRYSTR\$ che consente di specificare query string che eccedono i 255 caratteri. N.B.: _WSQRYSTR\$ è prevalente rispetto a questa variabile, per cui non va valorizzata quando si intende utilizzare _WSQRYSTRA\$().
_WSPROTOCOL\$	Protocollo (valori ammessi: "http" e "https")
_WSPRXHST\$	Nome Host (o indirizzo) del Proxy
_WSPRXPRT	Porta del proxy
_WSPRXUSRPW\$	Dati per l'autenticazione sul proxy (user:password)
_WSPRXAUTH	Tipo autenticazione (0 = non spec., 1 = Any, 2 = Basic, 3 = NTLM, 4 = Negoziazione, 5 = Digest)
_WSPRXXTUNN	Tunneling (0 = Non Abilitato, <> 0 = Abilitato)

_WSAUTENT\$	Tipo Autenticazione. Valori ammessi: "basic", "bearer", "user", " digest ". La stringa consente la combinazione di caratteri maiuscoli e minuscoli.
_WSUSER\$	User (nome_utente); insieme alla password andrà a costituire una stringa codificata con base64 da passare ad esempio all'autenticazione "basic"). Nota: teoricamente _WSUSER\$ deve essere = "nome_utente", ma se si riscontrasse un errore di libreria (libcurl v. 7.40 utilizzata per i contatti HTTP/HTTPS ed FTP) occorre indicare anche l'informazione relativa " realm " (che è un valore ritornato dal Server e che il Client potrebbe non replicare in un eventuale nuovo contatto al Server); occorre pertanto fare in modo di poter conoscere questo dato "realm" dai log di dialogo Client-Server), pertanto, la forma eventualmente da utilizzare è quella che lo indica direttamente: _WSUSER\$="valore_realm/nome_utente"
_WSPASS\$	Password
_WSTKN\$()	Token (nel caso di autenticazione non "basic" il dato è già preparato). L'array _WSTKN\$() è utile quando il WS ha una specifica struttura di gestione Token; se è richiesta una gestione del Token nell'header è disponibile la _WSOHEADER\$(). Il dialogo con i WS che richiedono l'autenticazione tramite token si basa sui passi: <ol style="list-style-type: none"> 1. prima autenticazione al WS che assegna un token, da memorizzare e mantenere per le successive comunicazioni; 2. successive n-conessioni da effettuare passando il token ottenuto; E' il WS con cui si dialoga che definisce le regole di gestione del token, perchè potrebbe far scadere il token dopo un certo tempo oppure un certo numero di connessioni, e quindi potrebbe essere necessario ripetere la procedura di richiesta e assegnazione di token per singola connessione; inoltre, potrebbe richiedere all'utilizzatore di invalidare il token (tramite appositi metodi messi a disposizione) alla fine della comunicazione.
_WSCERTFILE\$	Eventuale percorso/nome_file del certificato OpenSSL del client
_WSCERTPASS\$	Eventuale Passphrase richiesta per l'uso del certificato
_WSKEYFILE\$	Eventuale percorso/nome_file del file contenente la chiave privata
_WSCAFILE\$	File contenente le autorità di certificazione
_WSVERIFICA	Se diverso da 0 richiede la verifica del peer
_WSUSRAGENT\$	Testo da comporre liberamente il cui contenuto viene passato come UserAgent
_WSMETHOD\$	Metodo HTTP (GET, "POST", "HEAD", "PUT", "DELETE", "TRACE", "OPTIONS", "CONNECT", "FORMMULTIPART", "FORMMULTIPARTTOM")
_WSIBDYFNAM\$	Percorso/nome del file contenente il "BODY" della richiesta già preparato. Alternativa alla variabile successiva _WSIBODY\$()
_WSIBODY\$()	Array di elementi contenente il body del messaggio. Alternativa alla variabile precedente _WSIBDYFNAM\$.
_WSICNTTYP\$	Valore del "Content-type" dell'input fornito
_WSHEADER\$()	Eventuali Header aggiuntivi. Nota: gli header HTTP lunghi più di 255 caratteri vengono spezzati su più righe dell'array di output _WSOHEADER\$ (analogamente per l'array di input _WSHEADER\$), delimitandoli con due appositi meta-tag. I valori dei meta-tag si ottengono chiamando la funzione HTTPHDRTAG\$() a cui si passa il valore stringa "BEGIN" per ottenere il tag di inizio ed il valore "END" per ottenere il tag di fine. Esempio
_WSCOOKIES\$	Eventuali cookie da passare al Server
_WSLOGNAME\$	Eventuale nome per il file di Log (creato nella cartella prevista per i log); es. _WSLOGNAME\$="SPRIX_WS.lo
_WSLOGVERB	Tipo Log e verbosità richiesti; valori ammessi (0 = Nessun Log, 1 = Log verboso, 2 = Trace (dump esadecimale dei byte trasferiti)
_WSSOAPACT\$	Variabile da utilizzare se la richiesta è di tipo SOAP; la stringa di questa variabile equivale alla SOAP Action, cioè a quanto definito dal WSDL del servizio e che va specificato in questa stessa variabile; alcuni servizi potrebbero richiedere che la stringa sia specificata, ma con valore null, ovvero _WSSOAPACT\$="" (e questo non significa che sia un'assenza di valore SOAP).
Gestione Multipart m = indice dell'array che indica la sezione del corpo multipart n = indice dell'array che indica il numero di segmento di testo (max 255) esempio	
_WSMELTYPE\$(m)	Variabile per definire come trattare ogni singola parte. Valori consentiti: <ul style="list-style-type: none"> • "FILE" per comunicare che il valore della parte andrà recuperato da un file (di cui si fornisce il nome con la variabile _WSMFNAME\$()); • "TESTO" per comunicare che il valore della parte viene ricavato ri assemblando tutti gli elementi che si trovano nella matrice _WSMPBUFF\$(<n>, <Max_TXT_CHUNK >). Per ogni singola parte si possono avere massimo (<n>) 255 segmenti.
_WSMCPNAME\$(m)	Nome della sezione di multipart. <ul style="list-style-type: none"> • Se non specificato e _WSMELTYPE\$()="FILE", il suo valore diventa il basename del file; • Se non specificato e _WSMELTYPE\$()="TESTO", il suo valore di default diventa una stringa con la forma tipo "data_<nnnn>" dove <nnnn> è la posizione della sezione all'interno del multipart. Questo valore contribuisce a formare il valore dell'header "Content-Disposition:" generando l'attributo "name".
_WSMFILNAME\$(m)	Valore che contribuisce a creare l'attributo "filename" dell'header "Content-Disposition:". Se nessun valore viene indicato, l'attributo "filename" non viene espresso.
_WSMFNAME\$(m)	Valore che comunica il [percorso]/nome del file da inviare come valore della sezione di multipart. (se _WSMELTYPE\$()="FILE")
_WSMPBUFF\$(n,m)	Matrice che fornisce i 255 (max) segmenti di testo che costituiranno il corpo della sezione (se _WSMELTYPE\$ = "TESTO").
_WSMCTYPE\$(m)	Indica il "mime type" della sezione del corpo multipart che si sta costruendo e contribuirà a formare l'header "Content-Type:" relativo alla sezione. Va indicato solo il valore, il nome dell'header viene inserito automaticamente.
_WSMPHEADER\$(m)	Header inseriti dal programmatore. In questa matrice si possono indicare massimo 32 Header per ogni sezione. Se si indicano anche gli header che vengono di default inseriti automaticamente (Content-Type: e Content-Disposition:), quelli indicati dal programmatore saranno quelli effettivamente inviati.
_WSFILUPLOAD	Parametro on/off con valore (0 1). Usare il contenuto del file _WSIBDYFNAM\$ per creare il BODY della richiesta (0), oppure, effettuare un file Upload (1)

_WSEXPECT100	Parametro on/off con valore (0 1). Istruisce la libreria libcurl su come gestire l'invio dell'header Expect: 100-continue <ul style="list-style-type: none"> • se 0 (default) l'header viene cancellato e quindi non inviato • se 1 l'header viene inviato ed il client non invierà il BODY della richiesta fino a quando il serve non invia come ack il response HTTP 100
_WSCHUNKED	Parametro on/off con valore (0 1). Istruisce la libreria se effettuare la trasmissione in un'unica soluzione oppure fare un invio a blocchi <ul style="list-style-type: none"> • 0 = invio in unica soluzione • 1 = invio a blocchi
_WSONNTMOUT	Timeout connessione (secondi), default = 60
_WSTRASFTOUT	Timeout transazione HTTP, (secondi), default 0 (zero = disattivato)
_WSMINBYTSEC	Traffico minimo accettato (bytes/secondo), default 100
_WSMAXSECLW	Tempo massimo di costante traffico inferiore o uguale a bytes al secondo prima di chiudere per timeout, (secondi) default 10. Indica per quanto tempo si accetta di sopportare continuamente un traffico minore/uguale in bytes/secondo indicati dalla variabile _WSMINBYTSEC, cioè, se per X secondi persiste una situazione in cui non passano più di Y bytes al secondo, si abbatte la connessione in quanto non soddisfacente.
_WSKEEPALIVE	Attivazione TCP KeepAlive, bool (0/1), default 0
Variabili utilizzate per la restituzione dei risultati:	
_WSRAWERR\$	Eventuali errori di comunicazione (anche errori di caricamento libreria, allocazione memoria, comunicazione, ...). Se vuoto nessun errore e se previsto dal servizio si riceverà un BODY.
_WSORESP	Codice del Response HTTP ricevuto dal server
_WSORESPD\$	Descrizione testuale del Response HTTP ricevuto dal server
_WSOCNTYPE\$	Content-Type del body del response ricevuto dal server
_WSOHEADER\$()	Header ricevuti dal server Gli header HTTP lunghi più di 255 caratteri vengono spezzati su più righe dell'array di output _WSOHEADER\$ (analogamente per l'array di input _WSHEADER\$), delimitandoli con due appositi meta-tag. I valori dei meta-tag si ottengono chiamando la funzione HTTPHDRTAG\$() a cui si passa il valore stringa "BEGIN" per ottenere il tag di inizio ed il valore "END" per ottenere il tag di fine.
_WSOFNAME\$	Percorso/nome del file contenente il "BODY" del response ricevuto dal server
_WSOBODY\$()	Risultato della comunicazione. L'array ha limite di 200 elementi. Se non fornito un file (_WSOFNAME\$) in cui depositare il body, questo viene restituito negli elementi di questo array
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento Clicca qui	

_XML - Stringa XML

_XML

Variabili di struttura Xml	
_XMLVALNR	Numero di elementi di suddivisione della stringa di lunghezza superiore ai 255 caratteri;
_XMLSTRVAL\$()	Variabile array per la suddivisione della stringa in _XMLVALNR elementi
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

_ZA – Anagrafica unica

_ZA

Variabili di struttura Anagrafica Unica (_ZA)	
_ZACOD	Codice anagrafico;
_ZADTV\$	Data fine validità;
_ZADIV\$	Data di inizio validità (valore desunto da calcolo e non presente su disco);
_ZADTC\$	Data creazione;
_ZADTM\$	Data ultima modifica;
_ZAANN\$	Annulato logicamente;
_ZACF\$	Codice fiscale;
_ZANPI\$	Partita iva;
_ZA_TPSOG\$	Tipo soggetto, valori ammessi M/F/C/P/E;
_ZAPFI\$	Persona fisica (obsoleta, rimpiazzata in 2023H da _ZA_TPSOG\$);
_ZASE\$	Tipo soggetto/sexo (obsoleta, rimpiazzata in 2023H da _ZA_TPSOG\$);
_ZARAG\$	Ragione sociale;
_ZACGN\$	Cognome;
_ZANOM\$	Nome;
_ZA_DESC\$	Descrizione;
_ZADEN\$	Denominazione;
_ZANOT\$(5)	Nota anagrafica;
_ZADTN\$	Dati costituzione: data;
_ZACAN\$	Dati costituzione: CAP;
_ZACON\$	Dati costituzione: comune;
_ZAPRN\$	Dati costituzione: provincia;
_ZAIRN\$	Dati costituzione: codice irpef;
_ZAINR\$	Sede legale: indirizzo completo;
_ZACAR\$	Sede legale: CAP;
_ZACOR\$	Sede legale: comune;
_ZAPRR\$	Sede legale: provincia;
_ZAIRR\$	Sede legale: codice irpef;

_ZALOR\$	Sede legale: località;
_ZANAR\$	Sede legale: nazione;
_ZASTR\$	Sede legale: domicilio fiscale coincide con la sede legale;
_ZATPD\$	Documento di identificazione: tipo;
_ZANRD\$	Documento di identificazione: numero;
_ZAARR\$	Documento di identificazione: autorità di rilascio;
_ZADTR\$	Documento di identificazione: data rilascio;
_ZADTS\$	Documento di identificazione: data scadenza;
_ZAPAE\$	Ulteriori dati persona fisica: nazionalità;
_ZAPRQ\$	Ulteriori dati persona fisica: professione/qualifica;
_ZASTC\$	Ulteriori dati persona fisica: stato civile;
_ZASTU\$	Ulteriori dati persona fisica: titolo di studio;
_ZATIT	Ulteriori dati persona fisica: titolo;
_ZANAT	Ulteriori dati persona giuridica: natura giuridica;
_ZASTS\$	Ulteriori dati persona giuridica: stato della società;
_ZASIS\$	Ulteriori dati persona giuridica: situazione società;
_ZAINF\$	Domicilio fiscale: indirizzo completo;
_ZACAF\$	Domicilio fiscale: CAP;
_ZACOF\$	Domicilio fiscale: comune;
_ZAPRF\$	Domicilio fiscale: provincia;
_ZAIRF\$	Domicilio fiscale: codice irpef;
_ZANAF\$	Domicilio fiscale: nazione;
_ZALOF\$	Domicilio fiscale: località;
_ZAINES\$	Residenza estera: indirizzo;
_ZALOE\$	Residenza estera: località
_ZASTE\$	Residenza estera: stato federato;
_ZACOE\$	Residenza estera: codice stato;
_ZACFE\$	Residenza estera: codice fiscale;
_ZANAE\$	Residenza estera: stato;
_ZALE_ID	Legale rappresentante: codice id;
_ZALE_CFSS\$	Legale rappresentante: codice fiscale;
_ZALE_SESS\$	Legale rappresentante: sesso;
_ZALE_CGN\$	Legale rappresentante: cognome;
_ZALE_NOM\$	Legale rappresentante: nome;
_ZALE_CCA\$	Legale rappresentante: codice carica;
_ZALE_DEC\$	Legale rappresentante: data decorrenza;
_ZALE_DTNS\$	Legale rappresentante dati di nascita: data;
_ZALE_COR\$	Legale rappresentante dati di nascita: comune;
_ZALE_PRR\$	Legale rappresentante dati di nascita: provincia;
_ZALE_INR\$	Legale rappresentante domicilio fiscale: indirizzo completo;
_ZALE_DCAR\$	Legale rappresentante domicilio fiscale: CAP;
_ZALE_DCOR\$	Legale rappresentante domicilio fiscale: comune;
_ZALE_DPRR\$	Legale rappresentante domicilio fiscale: provincia;
_ZALE_TEL\$	Legale rappresentante domicilio fiscale: telefono;
_ZALE_SID	Legale rappresentante società dichiarante: codice id;
_ZALE_SCFSS\$	Legale rappresentante società dichiarante: codice fiscale;
_ZALE_SRAG\$	Legale rappresentante società dichiarante: ragione sociale;
_ZATEL\$	Contatti: numero telefono;
_ZACEL\$	Contatti: numero cellulare;
_ZAFAX\$	Contatti: numero FAX;
_ZAEMA\$	Contatti: indirizzo email;
_ZATE2\$	Contatti: numero telefono 2;
_ZATE3\$	Contatti: numero telefono 3;
_ZAEM2\$	Contatti: indirizzo email 2;
_ZAPEC\$	Contatti: indirizzo email pec;
_ZAPE2\$	Contatti: indirizzo email pec 2;
_ZAINTS\$	Contatti: sito internet;
_ZA_NOTTEL\$	Contatti: nota telefono;
_ZA_NOTTEL2\$	Contatti: nota telefono 2;
_ZA_NOTTEL3\$	Contatti: nota telefono 3;
_ZA_NOTCELL\$	Contatti: nota cellulare;
_ZA_NOTFAX\$	Contatti: nota fax;
_ZA_NOTMAI\$	Contatti: nota email;
_ZA_NOTMAI2\$	Contatti: nota email 2;
Variabili volatili utilizzate per la storicizzazione (solo PUTZA):	
_ZA_ANOT\$	Nota storicizzazione: anagrafica;
_ZA_ADAT\$	Data storicizzazione: anagrafica (fine validità);
_ZA_ADAI\$	Data storicizzazione: anagrafica (inizio validità, usare con parametro "STOR=R");
_ZA_NNOT\$	Nota storicizzazione: nota;
_ZA_NDAT\$	Data storicizzazione: nota (fine validità);
_ZA_NDAI\$	Data storicizzazione: nota (inizio validità, usare con parametro "STOR=R");
_ZA_RNOT\$	Nota storicizzazione: residenza;
_ZA_RDAT\$	Data storicizzazione: residenza (fine validità);
_ZA_RDAI\$	Data storicizzazione: residenza (inizio validità, usare con parametro "STOR=R");
_ZA_TNOT\$	Nota storicizzazione: documento di identificazione;
_ZA_TDAT\$	Data storicizzazione: documento di identificazione (fine validità);
_ZA_TDAI\$	Data storicizzazione: documento di identificazione (inizio validità, usare con parametro "STOR=R");

_ZA_FNOT\$	Nota storicizzazione: ulteriori dati persona fisica;
_ZA_FDAT\$	Data storicizzazione: ulteriori dati persona fisica (fine validità);
_ZA_FDAI\$	Data storicizzazione: ulteriori dati persona fisica (inizio validità, usare con parametro "STOR=R");
_ZA_GNOT\$	Nota storicizzazione: ulteriori dati persona giuridica;
_ZA_GDAT\$	Data storicizzazione: ulteriori dati persona giuridica (fine validità);
_ZA_GDAI\$	Data storicizzazione: ulteriori dati persona giuridica (inizio validità, usare con parametro "STOR=R");
_ZA_INOT\$	Nota storicizzazione: domicilio fiscale;
_ZA_IDAT\$	Data storicizzazione: domicilio fiscale (fine validità);
_ZA_IDAI\$	Data storicizzazione: domicilio fiscale (inizio validità, usare con parametro "STOR=R");
_ZA_ENOT\$	Nota storicizzazione: residenza estera;
_ZA_EDAT\$	Data storicizzazione: residenza estera (fine validità);
_ZA_EDAI\$	Data storicizzazione: residenza estera (inizio validità, usare con parametro "STOR=R");
_ZA_LNOT\$	Nota storicizzazione: legale rappresentante;
_ZA_LDAT\$	Data storicizzazione: legale rappresentante (fine validità);
_ZA_LDAI\$	Data storicizzazione: legale rappresentante (inizio validità, usare con parametro "STOR=R");
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui	

ABILCOLLAUTO

ABILCOLLAUTO <nomefile\$>,<abil>

Parametri	
nomefile\$	Nome del file Collage automatico senza estensione. Notare bene: Per approfondimenti sulla sintassi necessaria per l'ambiente di sviluppo App in PassBuilder consultare il parametro "-p" dell'istruzione SHELLSPRIX.
Abil	1 = abilita (default); 0 = disabilita;

Istruzione che abilita o disabilita il caricamento del nome file Collage automatico senza estensione per impedirne l'eventuale ricaricamento quando fermato con STOPCOLLAGE.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ABS

ABS(<num>)

Parametri	
num	Numero di cui si vuole il valore assoluto;

Funzione che ritorna il valore assoluto di <num>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ACAPO

ACAPO\$()

Funzione che ritorna una stringa di caratteri da inviare alla stampante (con formato testo) attualmente aperta per effettuare un salto di riga (per andare a nuova riga). Se nessuna stampante è aperta, ritorna una stringa vuota. Es. LPRINT ACAPO\$();

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ACQUISISCIFILE

ACQUISISCIFILE <etichetta\$>

Parametri	
etichetta\$	Riferimento al Collage per il trattamento del file acquisito (vedere anche variabile _ACQFILE\$)

Istruzione per l'acquisizione file dal dispositivo mobile. La variabile di struttura con riferimento all'eventuale errore è _ERRFILE\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ADCOMMERCIALE

ADCOMMERCIALE <ModAgg\$>,<MM\$>,<AnnoInizElab>,<AnnoFineElab>,<PrzCalcMarg\$>,<OC\$>,<OF\$>,<MA\$>

Parametri	
ModAgg\$	Modalità aggiornamento. Valori accettati "C" = completa o "V" = solo variati;

MM\$	Movimenti di magazzino. "S" = si o "N" = no;
AnnoInizElab	Anno inizio elaborazione;
AnnoFineElab	Anno fine elaborazione,
PrzCalcMarg\$	Prezzo calcolo margine. Valori accettati "M" = movimento o "A" = anagrafica;
OC\$	Ordini clienti: "S" = si o "N" = no;
OF\$	Ordini fornitori: "S" = si o "N" = no;
MA\$	Matrici: "S" = si o "N" = no;

Istruzione che esporta i dati commerciali del gestionale nel database di AD. Nel caso in cui MM\$ = "N" le variabili degli anni possono essere vuote. Ritorna eventuali errori nelle variabili _ERRADCOM, _ERRADCOM\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ADPCOMMERCIALE

ADPCOMMERCIALE <annorif>,<annoprec>,<esplostrut>

ADPINDICI

Parametri	
Annorif	Anno di riferimento;
Annoprec	Anno precedente;
esplostrut	Esplosione articoli strutturati: 0 = non esplose le strutture; 1= esplose le strutture;

Istruzione che esporta i dati commerciali del gestionale nel database di ADP (funzione obsoleta già integrata nel Gestionale); gli anni da confrontare sono l'anno di riferimento e l'anno precedente dove il primo è maggiore del secondo; con eventuale esplosione degli articoli strutturati. In passato era prevista anche un'esportazione ADPINDICI per le analisi di bilancio.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APPENDISPX

#APPENDISPX <nomefile\$1> [<nomefile\$2> <nomefile\$3>...]

Parametri	
nomefile1\$	Nome del sorgente che si vuole includere, da indicare senza estensione e fra i simboli di minore e maggiore;

Istruzione che aggiunge il programma <nomefile\$1> in coda al sorgente compilato; <nomefile\$1> obbligatorio, da inserire senza estensione fra i simboli minore e maggiore <..>, gli altri parametri (quelli fra le parentesi quadre [...]) sono facoltativi.

All'interno degli sprix creati in PassBuilder è possibile aggiungere solo librerie presenti nella stessa PassApp, con la stessa sintassi appena indicata: #APPENDISPX <libsp\$1>, oppure librerie presenti in un'altra PassApp con la sintassi: <CodiceApp@libreria> (Es: #APPENDISPX <123456prova@libsp\$1>).

Serve per agganciare altri programmi in coda al codice in uso, in modo da renderlo più snello e non ripetitivo. La collocazione ottimale di questi programmi è nella voce di menu Libreria, che li rende disponibili per il riutilizzo di codice (es. routines) per diversi programmi Sprix/Collage. La Libreria non è compilabile e non è eseguibile autonomamente.

L'istruzione #APPENDISPX deve sempre iniziare dalla prima riga del programma. Sono consentite più istruzioni su più righe in sequenza ed a partire dalla prima riga.

Attenzione: la modifica alle librerie richiedono la ricompilazione dell'Applicazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRIAZIE

APRIAZIE <sigla\$>,<data\$>

Parametri	
sigla	Sigla dell'azienda che si vuole aprire;
data\$	Data di apertura dell'azienda nel formato "AAAAMMG^",

Istruzione che chiude l'azienda attualmente in uso e apre l'azienda nella data specificata (cambia l'archivio e ridisegna la videata uscendo della routine; è ciclabile solo nella tabella 'sa'). L'istruzione valorizza le variabili _ERRAZIE e _ERRAZIE\$. L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Esempio APRIAZIE "AZI", "20170314".

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRIAZIE_EXT

APRIAZIE_EXT <sigla\$>,<data\$>,<codSottAz\$>,<att\$>

Parametri	
sigla	Sigla dell'azienda che si vuole aprire;
data\$	Data di apertura dell'azienda nel formato "AAAAMMGG";
codSottAz\$	Codice sottoazienda: da 1 a 256 (30 se attività Iva) oppure 'G' (Generale) o 'P' (Prevalente, cioè default se parametro = ""). La 'G' (per convenzione è la numero -sxcsa- 651) può ciclare solo i suoi dati e non può leggere in modalità 'T' (primenote delle sotto attività come avviene in revisione primanota) salvo singoli APRIAZIE_EXT su ciascuna sotto azienda.
att\$	Parametro che accetta valore "FINATT" (apre una finestra che avvisa di attendere)

Istruzione equivalente alla APRIAZIE, ma per la gestione di sottoaziende (esempio: attività Iva). L'istruzione valorizza le variabili _ERRAZIE e _ERRAZIE\$. L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder. Esempi:

APRIAZIE_EXT "A36",sxdat,"1".

APRIAZIE_EXT azsig, sxdat, STR\$(sxcsa,0,0), "FINATT"

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRIFIN

APRIFIN <nfin>,<colore>,<rifnfin>,<rig>,<col>,<alt>,<larg>

Parametri	
nfin	Numero delle finestra. Accetta valori da 1 a 9 anche contemporaneamente + quella numero 0, cioè quella base del Gestionale e che è necessariamente aperta, anche in assenza di alcuna apertura);
colore	Numero di colore finestra: 1 normale; 2 normale; 3 aiuto; 4 messaggio; 5 errore; 9 base. Il parametro consente anche di eliminare il bordo di queste finestre ed i numeri sono rispettivamente: 11; 12; 13; 14; 15; 19. Il colore è condizionato dal tipo di interfaccia, in quanto è divenuto obsoleto nell'interfaccia grafica (il colore aveva la funzione di distinguere le finestre nell'interfaccia a carattere).
rifnfin	Riferimento di apertura della finestra rispetto alla precedente: dalla 0 (base del gestionale) fino a 9. Nel Collage le finestre in negativo sono quelle del gestionale; è utilizzabile _NFIN (numero di finestra attiva in quel momento);
rig	Riga riferita a <rifnfin> da cui partire per disegnare la finestra: da 1 al numero massimo di righe gestite dal desk; se < 0 si intende in alto rispetto <rifnfin>; se = 0 si intende la riga centrale;
col	Colonna riferita a <rifnfin> da cui partire per disegnare la finestra: da 1 al numero massimo di colonne gestite dal desk; se < 0 si intende a sinistra rispetto <rifnfin>; se = 0 si intende la colonna centrale e se anche <rig> = 0 la finestra è centrata nell'area di lavoro;
alt	Numero di righe che compongono la finestra. Da 1 a quanto consentito dal tipo di video: (per il 4/3 = 21 Righe per 94 Colonne); (per il 16/9 = 21 Righe per 116 Colonne);
larg	Numero di colonne che compongono la finestra. Da 1 a quanto consentito dal tipo di video: (per il 4/3 = 21 Righe per 94 Colonne); (per il 16/9 = 21 Righe per 116 Colonne);

Istruzione che apre una finestra. La finestra aperta viene resa attiva (tutte le altre finestre vengono disattivate) e la posizione del cursore viene impostata a riga 1, colonna 1 della finestra stessa. Il numero attribuito ad una finestra è puramente indicativo e non indica la sequenza. Viene considerata la sequenza effettiva con cui le finestre vengono aperte, anche con numeri a caso. **Nota:** indipendentemente dalla risoluzione del video la gestione delle finestre certificata è di 19 righe per 94 colonne. Le righe 1, 2, 3, 24, 25 sono riservate. Es APRIFIN 1,1,0,5,10,10,30

Attenzione: la SETTF posiziona i tasti funzione nella finestra base.

Per renderli visibili nell'interfaccia grafica successiva alla v2020J (v801) occorre avvalersi di APRIFIN_TIT o APRIFIN_EXT; in alternativa si rendono visibili spostando e fissando (con l'icona "spillo/puntina" presente nell'angolo in alto/destra della finestra del modulo che copre la finestra base in cui detti tasti funzione sono visualizzati).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRIFIN_EXT

APRIFIN_EXT <nfin>,<rifnfin>,<rig>,<col>,<alt>,<larg>,<tit\$>,<pulsanti\$>,<aspetto\$>

Parametri	
nfin	Come l'istruzione APRIFIN, ovvero: Numero delle finestra. Accetta valori da 1 a 9 anche contemporaneamente + quella numero 0, cioè quella base del Gestionale e che è necessariamente aperta, anche in assenza di alcuna apertura);
rifnfin	Come l'istruzione APRIFIN, ovvero: Riferimento di apertura della finestra rispetto alla precedente: dalla 0 (base del gestionale) fino a 9. Nel Collage le finestre in negativo sono quelle del gestionale; è utilizzabile _NFIN (numero di finestra attiva in quel momento);
rig	Come l'istruzione APRIFIN, ovvero: Riga riferita a <rifnfin> da cui partire per disegnare la finestra: da 1 al numero massimo di righe gestite dal desk; se < 0 si intende in alto rispetto <rifnfin>; se = 0 si intende la riga centrale;
col	Come l'istruzione APRIFIN, ovvero: Colonna riferita a <rifnfin> da cui partire per disegnare la finestra: da 1 al

	numero massimo di colonne gestite dal desk; se < 0 si intende a sinistra rispetto <rifnfin>; se = 0 si intende la colonna centrale e se anche <rig> = 0 la finestra è centrata nell'area di lavoro;																																
alt	Come l'istruzione APRIFIN, ovvero: Numero di righe che compongono la finestra. Da 1 a quanto consentito dal tipo di video: (per il 4/3 = 21 Righe per 94 Colonne); (per il 16/9 = 21 Righe per 116 Colonne);																																
larg	Come l'istruzione APRIFIN, ovvero: Numero di colonne che compongono la finestra. Da 1 a quanto consentito dal tipo di video: (per il 4/3 = 21 Righe per 94 Colonne); (per il 16/9 = 21 Righe per 116 Colonne);																																
tit\$	Come l'istruzione APRIFIN_TIT, ovvero: Titolo delle finestra																																
pulsanti\$	<p>Pulsanti da visualizzare (separati dal carattere ",").</p> <p>Sintassi: "<cod>[#<idpulsantenav>][@<idico>][<desc>], ... , <cod>[#<idpulsantenav>][@<idico>][<desc>]"</p> <table border="1"> <tr> <td><cod></td> <td colspan="3">Codice numerico del tasto (vedere la tabella dei codici tasto in SETTF)</td> </tr> <tr> <td>[idpulsantenav]</td> <td colspan="3">[parametro facoltativo]: ID del pulsante di navigazione Nota: se specificato uno o piu' <idpulsantenav> è attivata la barra di navigazione della finestra.</td> </tr> <tr> <td></td> <td>1 - riga precedente</td> <td>5 - prima riga</td> <td>9 - pagina sinistra</td> </tr> <tr> <td></td> <td>2 - riga successiva</td> <td>6 - ultima riga</td> <td>10 - pagina destra</td> </tr> <tr> <td></td> <td>3 - pagina precedente</td> <td>7 - colonna sinistra</td> <td>11 - inizio riga</td> </tr> <tr> <td></td> <td>4 - pagina successiva</td> <td>8 - colonna destra</td> <td>12 - fine riga</td> </tr> </table> <table border="1"> <tr> <td>[idico]</td> <td colspan="3">[parametro facoltativo] - Eventuale ID dell'icona relativa al pulsante, consultare la tabella con la lista di icone elencate in SETTF.</td> </tr> <tr> <td>[desc]</td> <td colspan="3">[parametro facoltativo] - Eventuale Descrizione personalizzata (max 20 caratteri) e Tooltip del pulsante (separati da ' '). (vedere SETTF)</td> </tr> </table>	<cod>	Codice numerico del tasto (vedere la tabella dei codici tasto in SETTF)			[idpulsantenav]	[parametro facoltativo]: ID del pulsante di navigazione Nota: se specificato uno o piu' <idpulsantenav> è attivata la barra di navigazione della finestra.				1 - riga precedente	5 - prima riga	9 - pagina sinistra		2 - riga successiva	6 - ultima riga	10 - pagina destra		3 - pagina precedente	7 - colonna sinistra	11 - inizio riga		4 - pagina successiva	8 - colonna destra	12 - fine riga	[idico]	[parametro facoltativo] - Eventuale ID dell'icona relativa al pulsante, consultare la tabella con la lista di icone elencate in SETTF.			[desc]	[parametro facoltativo] - Eventuale Descrizione personalizzata (max 20 caratteri) e Tooltip del pulsante (separati da ' '). (vedere SETTF)		
<cod>	Codice numerico del tasto (vedere la tabella dei codici tasto in SETTF)																																
[idpulsantenav]	[parametro facoltativo]: ID del pulsante di navigazione Nota: se specificato uno o piu' <idpulsantenav> è attivata la barra di navigazione della finestra.																																
	1 - riga precedente	5 - prima riga	9 - pagina sinistra																														
	2 - riga successiva	6 - ultima riga	10 - pagina destra																														
	3 - pagina precedente	7 - colonna sinistra	11 - inizio riga																														
	4 - pagina successiva	8 - colonna destra	12 - fine riga																														
[idico]	[parametro facoltativo] - Eventuale ID dell'icona relativa al pulsante, consultare la tabella con la lista di icone elencate in SETTF.																																
[desc]	[parametro facoltativo] - Eventuale Descrizione personalizzata (max 20 caratteri) e Tooltip del pulsante (separati da ' '). (vedere SETTF)																																
aspetto\$	<p>Parametri aspetto separati cal carattere ", "</p> <ul style="list-style-type: none"> • SIMFBASE: simula finestra base • SIMFBASECENTRO: simula finestra base con posizionamento al centro • CSFONDO: colore sfondo finestra dello stesso colore di <rifnfin>; parametro obsoleto ereditato dal secondo parametro di APRIFIN ed utilizzabile solo nelle vecchie interfacce. • NOBORDO: senza bordi • IDFIN = [8 caratteri maiuscoli che identificano il nome della finestra]; se è presente il parametro IDFIN si attiva la memorizzazione delle finestre; se tutte le finestre si chiamano con lo stesso nome e viene modificato il posizionamento di una si cambiano tutte • "NOVISEZIONE" per non mostrare la sezione aperta di default per l'intera area della finestra (v. esempio) • "SFONDOAPP" per visualizzare finestra con sfondo bianco anziché grigio (per finestre di messaggio che non devono mostrare campi di input); (v. esempio) 																																

Istruzione che è un'evoluzione estesa delle istruzioni APRIFIN e APRIFIN_TIT, con l'aggiunta di Icone, Tooltip, Pulsanti di Navigazione

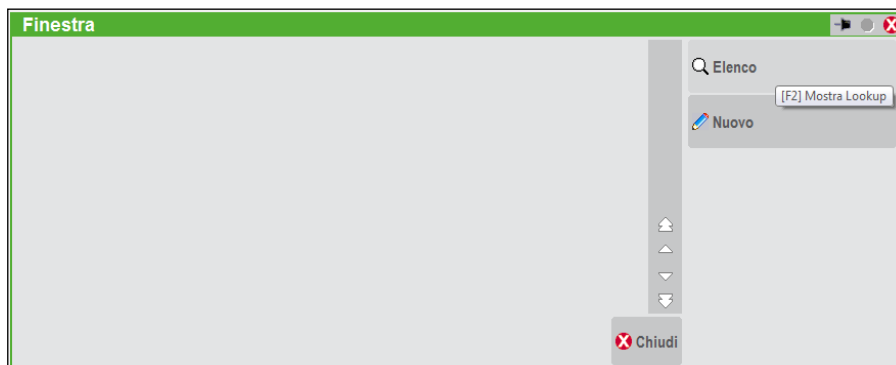
Attenzione: mentre l'istruzione APRIFIN_TIT aggiunge sempre una riga in più per opsitare i tasti funzione, l'istruzione APRIFIN_EXT, invece, aggiunge una riga solo se vi sono tasti funzione di tipo "azione" definiti per la finestra. Se si prevedono quindi tasti funzione per i singoli input presenti nella finestra l'aggiunta di una riga alle dimensioni deve essere prevista dal programmatore;

Consultare anche le Funzioni NRIGDISPFIN() e NCOLDISPFIN(),

Esempi

APRIFIN_EXT 1,0,1,1,RIG,COL,"Finestra","A505,512#1,513#2,510#3,511#4,602@11Elenco|Mostra Lookup,604Nuovo","NOBORDO"

APRIFIN_EXT (l'immagine che segue è uno fra i vari risultati possibili, non è relativa alla sintassi della APRIFIN_EXT precedente)



APRIFIN_EXT W,0,1,1,10,50,"Titolo","A505,A610", ""



APRIFIN_EXT W,0,1,1,10,50,"Titolo","A505,A610", "NOVISEZIONE" (nota: il colore grigio è lo sfondo senza bordi)



APRIFIN_EXT W,0,1,1,10,50,"Titolo","A505,A610", "SFONDOAPP"



Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRIFIN_TIT

APRIFIN_TIT <nfin>,<colore>,<rifnfin>,<rig>,<col>,<alt>,<larg>,<tit\$>,<tasti>

Parametri	
nfin	Numero delle finestra. Accetta valori da 1 a 9 anche contemporaneamente + quella numero 0, cioè quella base del Gestionale e che è necessariamente aperta, anche in assenza di alcuna apertura);
colore	Numero di colore finestra: 1 normale; 2 normale; 3 aiuto; 4 messaggio; 5 errore; 9 base. Per eliminare il bordo di queste finestre i numeri sono rispettivamente: 11; 12; 13; 14; 15; 19. Il colore è condizionato dal tipo di interfaccia.
rifnfin	Riferimento di apertura della finestra rispetto alla precedente: dalla 0 (base del gestionale) fino a 9. Nel Collage le finestre in negativo sono quelle del gestionale; è utilizzabile _NFIN (numero di finestra attiva in quel momento);
rig	Riga riferita a <rifnfin> da cui partire per disegnare la finestra: da 1 al numero massimo di righe gestite dal desk; se < 0 si intende in alto rispetto <rifnfin>; se = 0 si intende la riga centrale;
col	Colonna riferita a <rifnfin> da cui partire per disegnare la finestra: da 1 al numero massimo di colonne gestite dal desk; se < 0 si intende a sinistra rispetto <rifnfin>; se = 0 si intende la colonna centrale e se anche <rig> = 0 la finestra è centrata nell'area di lavoro;
alt	Numero di righe che compongono la finestra. Da 1 a quanto consentito dal tipo di video: (per il 4/3 = 21 Righe per 94 Colonne); (per il 16/9 = 21 Righe per 116 Colonne);
larg	Numero di colonne che compongono la finestra. Da 1 a quanto consentito dal tipo di video: (per il 4/3 = 21 Righe per 94 Colonne); (per il 16/9 = 21 Righe per 116 Colonne);
tit\$	Titolo delle finestra
tasti	Codice del tasto che si vuole visualizzare (vedere istruzione SETTF); in Sprix, antepoendo al codice del tasto la lettera 'A' si imposta il tasto come di tipo "Azione" (visualizzandolo in basso a destra nel piede della finestra aperta) altrimenti, senza la lettera 'A' si imposta il tasto come di tipo "Scheda" (visualizzato nella colonna dei pulsanti sulla destra del gestionale); Collage imposta solo tasti di tipo "Azione"

Istruzione equivalente ad APRIFIN, che consente di intitolare la finestra e di indicare i tasti attivi con la relativa descrizione, attivati da SETTF. Esempio: NF=AUTOFIN() : APRIFIN_TIT NF,1,0,3,3,10,10,"Titolo","502Descriz" (oppure "A502Descriz") : SETTF "502Descriz"

Nota: indipendentemente dalla risoluzione del video la gestione delle finestre certificata è di 19 righe per 94 colonne. Le righe 1, 2, 3, 24, 25 sono riservate.

Il parametro dei tasti attivi con descrizione è utilizzabile solo in Sprix e visualizza il pulsante con descrizione, nel elenco verticale dei tasti funzione visualizzato sulla destra del gestionale, comunque comandati dalla SETTF (diversamente non è garantita la compatibilità con la funzione chiamante); bisogna utilizzare la SETTF per attivare i tasti e quelli di APRIFIN_TIT si collocano nella parte destra anziché nell'ultima riga della finestra aperta (come farebbe SETTF). Nel caso ci siano più tasti funzioni che debbano essere utilizzati all'interno della finestra e visualizzati nella pulsantiera di destra questi devono essere tutti dichiarati nell'APRIFIN_TIT e successivamente gestiti dal SETTF

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRISTAMP

APRISTAMP <numstamp>,<nomemod\$>

Parametri	
numstamp	Numero della stampante che si vuole aprire (da 1 a 126 per quante stampanti sono state configurate all'interno del gestionale; 0=soppressa, 127=video; i <numstamp> sono memorizzati anche nel file di configurazione, interrogabile).
nomemod\$	Nome che si vuole assegnare al modulo da inserire, es. "SCHEDE CONTABILI" (tale descrizione serve per la richiesta di inserimento del cambio modulo, se "" non viene visualizzato nessun messaggio). Nome modulo è valido per le stampanti ad eccezione di quelle video.

Istruzione che apre la stampante numero e con richiesta nome modulo. **Non consente di richiamare moduli di stampe grafiche (vedere invece APRISTAMP_EXT e APRISTAMP_EMAIL)**. Dopo l'utilizzo della stampante, questa va chiusa e liberata per altri terminali con la CHIUDISTAMP (anche per stampa di modulo grafico Sprix in App PassBuilder) che chiude la stampante attualmente aperta (anche automaticamente da altre operazioni di stampa). APRISTAMP valorizza la variabile _NUMSTAMP e le variabili dimensionali _STP.

All'evento di stampa e/o termine di stampa, la stampante definita nella configurazione non è bypassabile dall'apertura di una stampante diversa (e con il messaggio "stampante occupata o già in uso"): attendere il termine dalla stampa (e/o richiamare APRISTAMP con Sprix) oppure intercettare l'evento successivo con la gestione di un flag di attivazione (per evitare la ripartenza della stampa). I formati di stampa (o automatismi) dipendono dalla configurazione della stampante, ad esempio i caratteri "!" per la stampante di tipo e-mail (come avveniva in passato prima di apposita istruzione APRISTAMP_EMAIL).

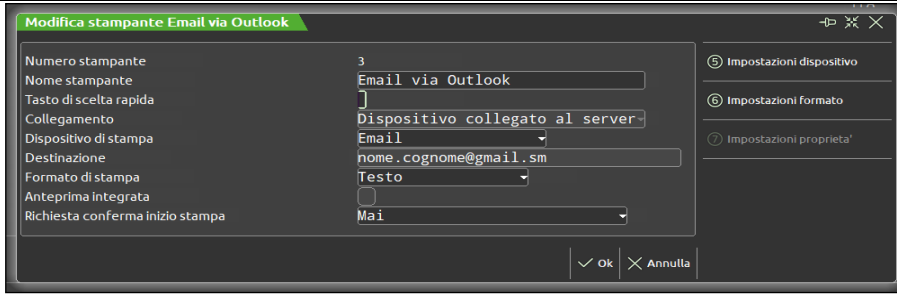
Esempio: APRISTAMP 127,"" : LPRINT "1234567890" : CHIUDISTAMP

Dalla versione 2020A (v750) l'istruzione è disponibile anche in **Collage Server Remoto** per stampe immediate su file e stampe fisiche, ma **esclusivamente in assenza di interazioni I/O e solo per stampe che non utilizzino la modulistica grafica**. La stampa con dispositivo collegato al Server è consentita con stampe immediate su file e fisiche configurate senza la richiesta conferma inizio stampa. Non sono consentite tutte le condizioni che richiedano un I/O o che rilascino messaggi di qualsiasi forma e contenuto. Esempio: stampa video, richiesta conferma stampante, scelta dispositivo/formato, invio mail in corso, mittente/destinatario, errori di configurazione mail SMTP. In mancanza dei suddetti requisiti, sono provocati simili errori (con/senza chiusura immediata del Client Mobile): "eccezione non gestita", "Impossibile accedere alla tastiera - Tastiera non disponibile in modalita senza interfaccia utente", "Sprix "CollageServerRemoto" - Stampa non disponibile - Routine collage interrotta", ecc... Le combinazioni per l'utilizzo dalla stampante variano in base al tipo di operazione ed il tipo di configurazione in uso, pertanto non sono elencabili in modo esaustivo e occorre considerarle empiricamente, come anche le ipotesi di ristampa su file pdf nel momento in cui il medesimo file pdf sia aperto in consultazione sul Server oppure l'uso di editor di stampa che richiedano esiti oppure conferme dell'elaborato (es. Adobe PDF) oppure l'inceppamento/esaurimento della carta su una stampante fisica, in quanto si tratta di un evento non intercettato dal Collage (salvo qualora sia possibile, per esempio, che la stampante sia successivamente in grado di comunicare il suddetto evento ad un altro Collage che stia "in ascolto" sulla porta), e così via... Per vederne un'applicazione in SRVCALL consultare gli esempi di Sprix Mobile.

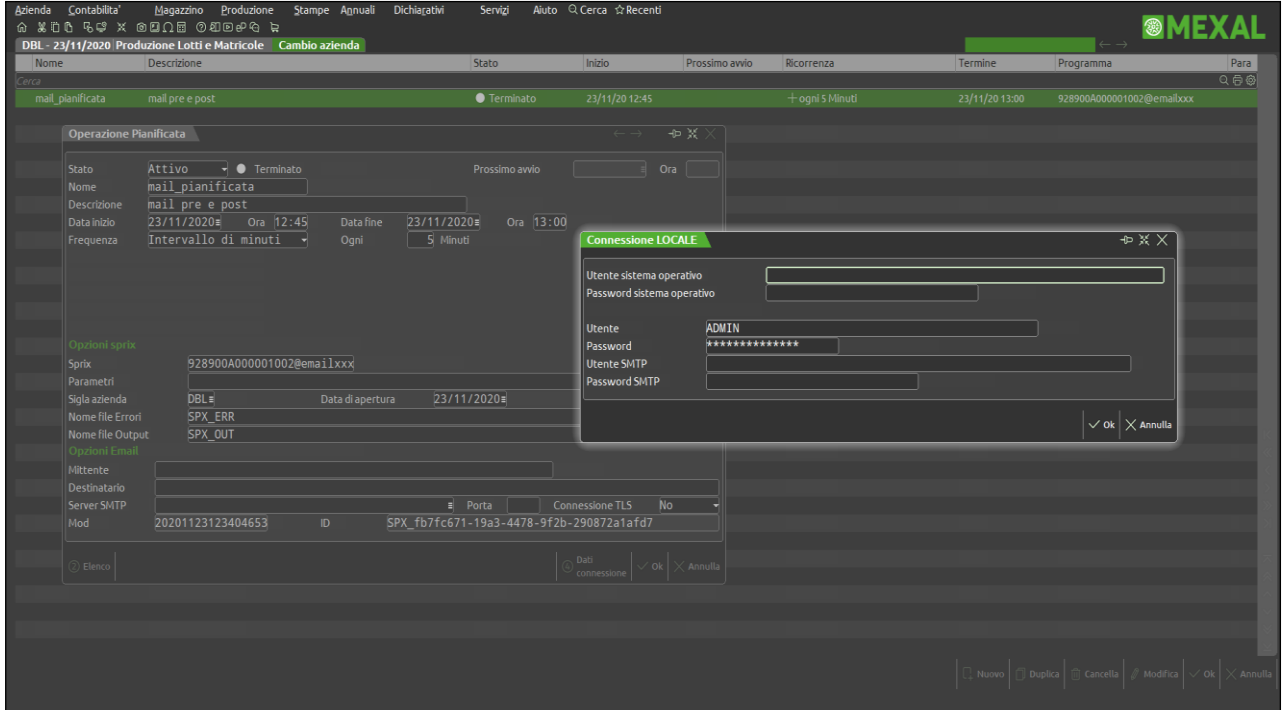
Le OPERAZIONI PIANIFICATE consentono l'uso di comandi di stampa solo se soddisfano i requisiti di assenza di interazioni I/O video previsti per le operazioni schedate di tipo batch (consultare "mxrsbatch.exe" nel manuale utente).

Esempio: esecuzione di APRISTAMP_EMAIL ogni 5 minuti, tramite le Operazioni Pianificate (menu ZCP)

1) Configurazione "silente" del dispositivo



2) Configurazione dell'operazione schedulata. Notare che le informazioni sulla posta elettronica non ci sono, perchè sono gestite dalla precedente configurazione del device di stampa di tipo Email, richiamata da APRISTAMP_EMAIL



3) Esecuzione dello Sprix

```

Sprix "EMAIL" emailxxx.ss
MAIL:
  APRISTAMP_EMAIL 3,"", "nome.cognome@gmail.com", "", "", "Oggetto", "Testo", "", "", "", "" : LPRINT "prova di stampa" : CHIUDISTAMP_EXT
  CHIUDISTAMP_EXT 'chiusura stampante aperta con APRISTAMP_EXT
  IF _ERRCHSTAMP<>0 THEN MIOERR_STP$ = _ERRCHSTAMP$ : 'test CHIUDISTAMP_EXT
END
    
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRISTAMP_EMAIL

APRISTAMP_EMAIL

<numstamp>,<mittente\$>,<destinatario\$>,<cc\$>,<ccn\$>,<oggetto\$>,<testo\$>,<firma\$>,<nomeallegato\$>,<allegato\$>,[modstsp\$]

Parametri	
numstamp	numero della stampante che deve essere obbligatoriamente di tipo "E"
mittente\$	Indirizzo email del mittente (vedere la nota (*))
destinatario\$	Indirizzo email del destinatario (vedere la nota (*))
cc\$	Indirizzo email del copia conoscenza
ccn\$	Indirizzo email del copia conoscenza nascosto
oggetto\$	Oggetto della email

testo\$	Testo da inserire come prima parte del corpo della email
firma\$	Firma della email
nomeallegato\$	Nome del file allegato
allegato\$	Percorso del file da allegare alla mail; consentiti più allegati concatenati dal simbolo “;”
[modstspx\$]	<p>Parametro opzionale relativo al codice del modulo Sprix per la stampa grafica Sprix nella App PassBuilder, utilizzabile in relazione alle istruzioni grafiche LPRINTGRAF_???; diversamente, se non viene indicato (omesso), APRISTAMP_EMAIL mantiene l'originaria funzionalità idonea alle tradizionali istruzioni LPRINT???</p> <p>Il parametro consente di gestire l'ultimo modulo utilizzato in apertura stampante senza fermarsi alla richiesta del modulo. Nel codice del modulo possono essere indicate le seguenti alternative:</p> <ul style="list-style-type: none"> • Un codice modulo. Se il modulo esiste all'interno della App viene utilizzato senza fermarsi alla richiesta di selezione del modulo; se il modulo non esiste viene segnalato modulo inesistente. Esempio "MODSTP1". • Un codice modulo seguito dal carattere '?'. Se il modulo esiste all'interno della App viene utilizzato come default e si ferma alla richiesta di selezione del modulo; se il modulo non esiste viene segnalato modulo inesistente. Esempio "MODSTP1?". • Il carattere "?". Se esiste almeno un modulo di stampa SPRIX viene utilizzato il primo come default e si ferma alla richiesta di selezione del modulo; se il modulo non esiste viene segnalato modulo inesistente. • Il carattere "*". Se è già stato selezionato un modulo tramite una precedente APRISTAMP_EXT oppure una APRISTAMP_EMAIL viene utilizzato senza fermarsi alla richiesta di selezione del modulo; diversamente, il parametro funziona come con il carattere "?". • I caratteri "**?". Se è già stato selezionato un modulo tramite una precedente APRISTAMP_EXT oppure una APRISTAMP_EMAIL viene utilizzato come default e si ferma alla richiesta di selezione del modulo; diversamente, il parametro funziona come con il carattere "?".

Istruzione equivalente all'APRISTAMP, ma per stampanti di tipo Email. I parametri dell'istruzione rispettano fedelmente quelli che sono i campi della stampante Email del gestionale. Per le specifiche di ogni singolo campo si rimanda al relativo manuale. **Nota: è necessario almeno una riga di stampa per inviare la Email.** Valorizza la variabile di errore _ERRSTPEMAIL se la stampante non è di tipo Email.

Esempio: APRISTAMP_EMAIL 69,"",Eccetera... : LPRINT "1234567890" : CHIUDISTAMP

L'istruzione è disponibile in Collage Server Remoto se la stampante è correttamente configurata per non essere interattiva (no I/O video), così come richiesto anche per **APRISTAMP_EXT**.

Nota per le OPERAZIONI PIANIFICATE consultare APRISTAMP.

L'istruzione APRISTAMP_EMAIL usa una stampante email attiva e funzionante nell'installazione. Tale comando è uguale ad APRISTAMP; se il programma che utilizza questo comando non effettua realmente una stampa, il comando APRISTAMP_EMAIL non viene eseguito. I valori passati al comando APRISTAMP_EMAIL si sostituiscono a quelli presenti nella configurazione della stampante. Esempio:

APRISTAMP_EMAIL 30,MITT\$,DEST\$,CC\$,CCN\$,OGG\$,TEST\$,FIRM\$,ALL\$,PER\$

LPRINT "";

Se la configurazione della stampante Email è di tipo pdf, il risultato della stampa viene generato come allegato della mail; se è presente anche un allegato, il ricevente avrà due file, uno vuoto ed uno ulteriore allegato dalla procedura.

Se la configurazione della stampante Email è di tipo testo, il risultato della stampa viene inserito nel corpo della mail; se è presente anche un allegato, il ricevente avrà un solo file, che è quello allegato dalla procedura.

APRISTAMP_EMAIL è un comando di gestione del dispositivo, non gestisce i contenuti come potrebbe fare Outlook o simili programmi di posta elettronica. L'istruzione può solamente allegare e/o scrivere testo nel corpo, l'unico modo per inserire immagini è con l'impostazione del dispositivo di stampa configurato Email su formato qualsiasi (esempio Jpg) ed impostandolo con un Testo su file html. Questo file (htm o html) può contenere immagini o altre forme grafiche.

Per approfondimenti sulla gestione dell'EMAIL SMTP consultare il [manuale utente](#) (Servizi - Configurazioni - Stampanti).

(*) La finestra di richiesta dei dati email si apre solo con invio SMTP (in MAPI esiste la visualizzazione interattiva, regolata dal relativo parametro). Se mittente e destinatario sono valorizzati in sprix, vengono utilizzati questi ultimi indipendentemente dalla stampante (non esistono livelli intermedii). Se non risultano valorizzati in sprix, vengono assunti quelli della stampante, dove:

- 1) se valorizzati staticamente, allora prosegue;
- 2) se da richiedere oppure con tag non risolto (es. [IntestDocStp Autom]), si ferma all'atto di stampa.

Risulta sufficiente che uno dei due campi ricada nel caso 2) per aprire la finestra all'atto di stampa. Per CC e CCN si mantiene invece un diverso comportamento. Se destinatario e mittente sono in qualche modo entrambi risolti, l'apertura della finestra di conferma avviene solo se uno tra CC o CCN è da richiedere. In assenza di tale condizione, la procedura non apre la finestra di richiesta dati email. Anche qualora almeno uno tra CC e CCN sia con uno degli altri tag non risolti, es. [IntestDocStp Autom].

Nota: in relazione al "formato" della stampante aperta, il prodotto concesso in "comodato d'uso" genera una e-mail con allegato un file contenente output "fac-simile".

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRISTAMP_EMAIL_EXT

APRISTAMP_EMAIL_EXT

<numstamp>,<mittente\$>,<nomemittente\$>,<rispondiun\$>,<nomerispondiun\$>,<destinatario\$>,<cc\$>,<ccn\$>,<oggetto\$>,<testo\$>,<firma\$>,<filetestohtml\$>,<nomeallegato\$>,<allegati\$>,<mittenteinCCN\$>,<confermalettura\$>,<prioritasmp\$>,[modstspx\$]

Parametri	
numstamp	numero della stampante che deve essere obbligatoriamente di tipo "E"
mittente\$	Indirizzo email del mittente
nomemittente\$	Nome del mittente
rispondiA\$	Indirizzo del rispondi a
nomerispondiA\$	Nome del rispondi a
destinatario\$	Indirizzo email del destinatario
cc\$	Indirizzo email del copia conoscenza
ccn\$	Indirizzo email del copia conoscenza nascosto
oggetto\$	Oggetto della email
testo\$	Testo da inserire come prima parte del corpo della email
firma\$	Firma della email
filetestohtml\$	File del testo html
nomeallegato\$	Nome del file allegato
allegati\$	Percorso del file da allegare alla mail; consentiti più allegati concatenati dal simbolo ","
mittenteinCCN\$	Flag "S" o "N"
confermalettura\$	Flag "S" o "N"
prioritasmt\$	(vale "A", "B", "N") se "" lascia quello impostato sulla stampante
[modstsp\$]	Parametro opzionale relativo al codice del modulo Sprix per la stampa grafica Sprix nella App PassBuilder

Istruzione equivalente all'APRISTAMP_EMAIL, ma con il testo su file HTML.

Istruzione per stampare su stampante email e tra i vari parametri gli si può passare il percorso di un file HTML che diventerà (quindi sostituirà) il corpo dell'email.

Per ogni parametro se passata la stringa vuota ("") lascia quello impostato sulla stampante.

Consultare APRISTAMP_EMAIL e la gestione di EMAIL SMTP in [manuale utente](#) (Servizi-Configurazioni-Stampanti).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRISTAMP_EXT

APRISTAMP_EXT <numstamp>,<nomemod\$>,<lar>,[modstsp\$]

Parametri	
numstamp	Numero della stampante che si vuole aprire
nomemod\$	Nome che si vuole assegnare al modulo; tale descrizione serve per la richiesta di inserimento del cambio modulo, se "" non viene visualizzato nessun messaggio;
lar	Larghezza massima in quantità di caratteri che la stampante deve gestire. Consente di stabilire la larghezza in quantità di caratteri della riga di stampa (numero di colonne): se lar = 0 è impostata a 220, se lar < 79 è impostata a 79, se lar > 220 è impostata a 220.
[modstsp\$]	<p>Parametro opzionale relativo al codice del modulo Sprix per la stampa grafica Sprix nella App PassBuilder, utilizzabile in relazione alle istruzioni grafiche LPRINTGRAF_???.; se il parametro è preceduto dal carattere ":" significa che il modulo deve essere cercato fra quelli personalizzati, altrimenti, se tale carattere non viene indicato cerca tra quelli presenti dentro la App.</p> <p>Se il parametro [modstsp\$] non viene indicato (omesso), APRISTAMP_EXT mantiene l'originaria funzionalità idonea alle tradizionali istruzioni LPRINT???</p> <p>Il parametro [modstsp\$] consente di gestire l'ultimo modulo utilizzato in apertura stampante senza fermarsi alla richiesta del modulo. Nel codice del modulo possono essere indicate le seguenti alternative:</p> <ul style="list-style-type: none"> • Un codice modulo. Se il modulo esiste all'interno della App viene utilizzato senza fermarsi alla richiesta di selezione del modulo; se il modulo non esiste viene segnalato modulo inesistente. Esempio "MODSTP1". • Un codice modulo seguito dal carattere '?'. Se il modulo esiste all'interno della App viene utilizzato come default e si ferma alla richiesta di selezione del modulo; se il modulo non esiste viene segnalato modulo inesistente. Esempio "MODSTP1?". • Il carattere "?". Se esiste almeno un modulo di stampa SPRIX viene utilizzato il primo come default e si ferma alla richiesta di selezione del modulo; se il modulo non esiste viene segnalato modulo inesistente. • Il carattere "***". Se è già stato selezionato un modulo tramite una precedente APRISTAMP_EXT oppure una APRISTAMP_EMAIL viene utilizzato senza fermarsi alla richiesta di selezione del modulo; diversamente, il parametro funziona come con il carattere "?". • I caratteri "**?". Se è già stato selezionato un modulo tramite una precedente APRISTAMP_EXT oppure una APRISTAMP_EMAIL viene utilizzato come default e si ferma alla richiesta di selezione modulo; diversamente, il parametro funziona come con il carattere "?".

Istruzione equivalente all'APRISTAMP, con la possibilità di stabilire la lunghezza della riga di stampa. Valorizza variabili dimensionali _STP
Esempio: APRISTAMP_EXT 127,"",79 : LPRINT "1234567890" : CHIUDISTAMP

Dalla versione 2020A (v750) l'istruzione è disponibile anche in Collage Server Remoto, per stampe immediate su file e stampe fisiche, ma solamente in assenza di interazioni I/O (per approfondimenti consultare APRISTAMP) e solo per stampe che non utilizzino la modulistica grafica.

Nota per le OPERAZIONI PIANIFICATE consultare APRISTAMP.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

APRISTAMP_FILE

APRISTAMP_FILE <tipofile\$>,<nomefile\$>,<lar>,[modstspx\$]

Parametri	
tipofile\$	Tipo di file da generare. Può assumere i seguenti valori: <ul style="list-style-type: none"> • "P" oppure "PDF" per generare la stampa su file PDF • "C" oppure "CSV" per generare stampa su file CSV (richiede che sia specificato un modulo della stampa grafica)
nomefile\$	Nome completo del percorso del file sul quale generare la stampa
lar	Larghezza del modulo (come in APRISTAMP_EXT). Questo parametro non è considerato se è presente il modulo, poiché quest'ultimo è prioritario e possiede degli specifici parametri di configurazione.
modstspx\$	Parametro opzionale per indicare il codice modulo stampa grafica. Se indicato apre la stampa in modalità grafica. Nota: nella stampa CSV del modulo Sprix vengono generati almeno 5 file: si genera il file principale con tutte le righe del corpo della stampa ed un file per ciascuna sezione di sottocorpo del modulo con dettaglio delle relative righe (tre sottocorpi "Det1", "Det2", "Det3"), infine un file per MyDB. Sebbene il parametro consenta la gestione del modulo come nella APRISTAMP_EXT e nella APRISTAMP_EMAIL con le loro relative istruzioni LPRINTGRAF_???, la finestra per la selezione del nome file viene comunque aperta nelle seguenti condizioni: <ul style="list-style-type: none"> • il nome file non è stato indicato e l'istruzione viene utilizzata in SPRIX, COLLAGE o COLLAGE SERVER REMOTO • il nome file contiene un carattere non valido e l'istruzione viene utilizzata in SPRIX, COLLAGE o COLLAGE SERVER REMOTO • il file contiene una directory non esistente e l'istruzione viene utilizzata in SPRIX, COLLAGE o COLLAGE SERVER REMOTO • nel nome del file è presente il carattere '?' e l'istruzione viene utilizzata in SPRIX, COLLAGE o COLLAGE SERVER REMOTO • il nome del file inizia con il carattere '<' (file del client) e l'istruzione viene utilizzata in SPRIX, COLLAGE o COLLAGE SERVER REMOTO

Istruzione che genera la stampa sul tipo di file indicato senza alcuna interazione con l'utente. L'istruzione sfrutta delle stampanti virtuali, pertanto non è necessario avere configurato le stampanti di tipo file PDF oppure file CSV.

Sebbene l'istruzione sia di stampa, riguarda anche la gestione del file, pertanto, se _ACCFILEUTE è stata impostata a 1 vengono ritornati degli errori in _ERRFILE\$ altrimenti il Server Passepartout rilascia un errore di sistema (che comunque viene ritornato al MOBILE).

Nota per le OPERAZIONI PIANIFICATE consultare APRISTAMP.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui per la gestione Stampe](#) oppure [Clicca qui per la gestione File](#)

ARCHVIEW

ARCHVIEW <arch>,<gest\$>

Parametri	
arch	Valore dell'archivio da visualizzare; accetta i valori 1, 2, 3. 1= Tabella Banche 2= Tabella CAP 3= Tabella Codici Catastali dei comuni
gest\$	Parametro di gestione, attualmente l'unico valore ammesso è "NOVIEW=1" che disabilita l'apertura della finestra di elenco nel caso in cui la ricerca ritorna un solo valore.

Istruzione che esegue automaticamente una finestra elenco sulle tabelle Banche e CAP. L'istruzione utilizza le variabili di struttura _ARCHVIEW per impostare i filtri di ricerca e per ritornare i valori selezionati. L'istruzione valorizza le variabili di errore _ERRVIEW e _ERRVIEW\$. La navigazione della lista è la stessa della funzionalità standard del gestionale per la richiesta di ABI e CAP.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ARRO

ARRO(<numero>,<tipo>,<arr>)

Parametri	
numero	numero intero da arrotondare
tipo	tipo di arrotondamento 1 = difetto, 2 = medio-difetto, 3 = medio-eccesso, 4 = eccesso
arr	valore dell'arrotondamento. Al centesimo di Euro è pari a 0.01.

Funzione che ritorna il numero intero arrotondato in base al valore intero di arrotondamento.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ARRON

ARRON(<numero>,<tipo>,<decimali>)

Parametri	
numero	numero da arrotondare
tipo	tipo di arrotondamento 1 = difetto, 2 = medio-difetto, 3 = medio-eccesso, 4 = eccesso
decimali	numero di decimali a cui arrotondare

Funzione che ritorna il numero arrotondato in base al numero di decimali.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ARTDBP

ARTDBP(<codart\$>)

Parametri	
codart\$	Codice principale dell'articolo

Funzione che controlla se l'articolo con codice principale codart\$ è di produzione (ritorna 1) oppure no (ritorna 0).
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ASC

ASC(<str\$>)

Parametri	
str\$	Carattere di cui si vuole il valore ASCII

Funzione che ritorna il valore numerico ASCII del primo carattere della stringa <str\$>.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ASCMIRROR

ASCMIRROR <tabella\$>,<dir\$>,<file\$>

Parametri	
tabella\$	Tabella da elaborare
dir\$	Directory dove salvare il file\$
file\$	Nome del file da generare

Istruzione che legge tutti i dati contenuti nei record del nome di <tabella\$> del dizionario esportando in sovrascrittura/precancellazione del supporto queste informazioni su file ASCII(ASCMIRROR).

Le informazioni relative all'import/export della struttura dati sono disponibili nel menu SERVIZI del Gestionale:

- Servizi - Trasferimento Archivi - Export/Import CSV - ASCII (è disponibile anche un PDF con i tracciati CSV pubblicato nel chiarimento n. 54452 del supporto on-line)
- Servizi - Personalizzazioni - Stampa Dizionario Sprix (indica le specifiche dei campi)
- Consultare con editor di testo il file presente nel percorso \programmi\binloc\dbmirror.dat" dell'installazione.

Richiede il nome <file\$> (minuscolo se Linux; se "" il default è il nome di<tabella\$.ASC>) e il relativo percorso <dir\$> (se "" il default è directory dati azienda); l'errore di accesso al file è rilasciato da _ERRFILE\$. Il file è in formato CSV se utilizzato il ';' come separatore fra le colonne. Le stringhe non sono delimitate da alcun carattere, se nell'informazione di un campo stringa ci fosse almeno un ';' potrebbe comportare discrepanza tra colonne e dati. La prima riga è riservata ai nomi delle voci del dizionario per il battesimo delle colonne. I campi data non valorizzati nel gestionale sono convertiti in NULL (ASCMIRROR)

Viene visualizzato lo stato di avanzamento progressivo dei record trattati e si può interrompere con tasto [Esc] (previa impostazione di

_INTER) con riempimento del file/database fino a quanto eseguito (l'interruzione dell'elaborazione della tabella corrente non interrompe gli eventuali mirroring successivi). Non è eseguibile nel ciclotabellare, di cui valgono alcune regole, come ad esempio l'istruzione LEGGIDESC per attivare la lettura delle righe descrittive nel documento e l'elaborazione annuale per anno di ingresso in azienda. L'istruzione valorizza la _ERRASC MIR che rilascia il codice dell'esito della operazione effettuata.

Attenzione: il mirroring (anche quello in tempo reale) dei documenti di anni diversi con chiave identica (ma che non comprenda l'anno, es. sigla+serie+numero+conto a numeratore unico e non annuale) si sovrascrivono.

Il mirroring di interi archivi impegna il sistema, se è già appesantito da altre condizioni (dbmirror in tempo reale, reti sovraccariche, computer non performanti) è preferibile l'export delle tabelle necessarie ai soli dati di interesse da esportare con procedure batch di fine lavoro oppure interrogazioni di database replicati in chiaro o infine mirroring ottimizzati.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ASCMIRROR_EXT

ASCMIRROR_EXT <tip>,<rif\$>,<dir\$>,<file\$>,<slot>,<att>,<var>,<azz>,[vis]

Parametri	
tip	= 0 si stabilisce che il riferimento <rif\$> è il nome di una tabella appartenente al set della DBMIRROR classica (comprendente anche le tabelle delle videate personalizzate). = 1 si stabilisce che <rif\$> è il nome di un file di personalizzazione in cui sono codificate per ciascuna tabella le colonne che devono essere trattate nell'esportazione
rif\$	Nome di un file di personalizzazione in cui sono codificate per ciascuna tabella le colonne che devono essere trattate nell'esportazione; tale valore dipende dal parametro "tip";
dir\$	Directory dove salvare il file\$
file\$	Nome del file da generare
slot	Numero da 1 a 8 slot di esportazione presente nella tabella Stato anagrafiche del Gestionale
att	Record attivi: 0 = tratta tutti i record; 1 = solo se sono attivi
var	Variazione: = 0 non gestisce ottimizzazioni di export; = 1 esporta i soli record variati dall'ultimo aggiornamento; = 2 elimina i record dal database non gestendo lo stato del variato dello <slot>
azz	Azzeramento: = 0 non azzerava lo stato variato; = 1 azzerava lo stato variato;
vis	Visualizzazione facoltativa dello stato di avanzamento progressivo dei record trattati (letti e scritti): se = -2 non visualizza, se = -1 visualizza al termine dell'elaborazione il totale dei record trattati, se = 1 o = 0 (o assente) visualizza la progressione dei record (step=1), se = N visualizza con step=N e al termine visualizza il totale di record trattati.

Istruzione analoga alla ASCMIRROR, ma ottimizzata sulle tabelle Anagrafiche Piano dei Conti e Articoli, riducendo il volume dei dati esportato dalle tabelle: PC_PP_PA_ e AR_AH_AF_AX_AY_AL_AB_AW_AQ_AO_AE_.

La riduzione avviene con i flag <slot>, <att>, <var>, <azz>. Nel menu Aziende -Parametri di base - tabella Stato anagrafiche del Gestionale, si possono stabilire fino a 8 tipi di slot di esportazione (da utilizzare in altrettante applicazioni Sprix, esempio E-COMMERCE) per le quali definire, con la pressione del tasto [SHF11] e la selezione di 'Stato anagrafica' sul conto o sul articolo nelle rispettive anagrafiche, lo stato ATTIVO oppure NON ATTIVO oppure NON ATTIVO dello <slot> sulla variazione del dato: il singolo record è quindi esportato o meno in base a quanto stabilito con lo slot> sul flag di variazione del singolo record (che viene alzato dal Gestionale alla modifica del record). Nelle operazioni PUT la scrittura default è N per 'ATTIVO' e S per il 'VARIATO'; gli elementi 9-12 sono invece di sola lettura.

Gli <slot> della DBMIRROR e ASCMIRROR sono in comune: una esportazione su database con DBMIRROR determina la non esportazione su file dell'ASCMIRROR sul medesimo slot.

Lo <slot> è quindi un numero da 1 a 8; se = 0 non gestisce questa ottimizzazione attraverso il flag numerico <var> per l'export dei soli record variati dall'ultimo aggiornamento (se <var>=1) o di tutti i record con cancellazione (se <var>=0) o drop (se <var>=2) della tabella nel database (rendendo quindi irrilevante il flag 'Variato' dello <slot>). Gli <slot> della DBMIRROR e ASCMIRROR sono in comune: una esportazione su database con DBMIRROR determina la non esportazione su file dell'ASCMIRROR sul medesimo slot.

Il <var> dipende comunque dal flag numerico <att> (se <att>=0 tratta tutti i record, se <att>=1 solo se sono attivi). Terminata l'esportazione, per evitare successivi ri-trattamenti del record, il <var> di tutti gli slot (relativi all'intero archivio anagrafico e non della singola tabella) viene azzerato se <azz>=1 (se <azz>=0 non azzerava <var>), poiché se il flag numerico <tip> è = 0 si stabilisce che il riferimento <rif\$> è il nome di una tabella appartenente al set della DBMIRROR classica (comprendente anche le tabelle delle videate personalizzate).

Diversamente, con <tip>=1 si stabilisce che <rif\$> è il nome di un file di personalizzazione in cui sono codificate per ciascuna tabella le colonne che devono essere trattate nell'esportazione (se il file contiene il riferimento a più tabelle l'esportazione avverrà per ognuna di esse). Per i parametri <dir\$> <file\$> vedere ASCMIRROR.

Il nome del file di personalizzazione richiede quello della tabella e della colonna separati da uno o più blank o tabulatore ASCII; le colonne nel file possono avere un ordine diverso da quello definito nella struttura della tabella nel database. L'ordine può essere casuale (tabelle intervallate da linee vuote) e sono inseribili dei commenti (marcatore con apice "" al inizio di una nuova riga).

Col parametro numerico [vis] facoltativo, si stabilisce il modo di visualizzazione dello stato di avanzamento progressivo dei record trattati (letti e scritti): se = -2 non visualizza, se = -1 visualizza al termine dell'elaborazione il totale dei record trattati, se = 1 o = 0 (o assente) visualizza la progressione dei record (step=1), se = N visualizza con step=N e al termine visualizza il totale di record trattati.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ASCMIRROR_EXT2

ASCMIRROR_EXT2 <tip>,<rif\$>,<dir\$>,<file\$>,<slot>,<att>,<var>,<azz>,[impo\$]

Parametri	
tip	= 0 si stabilisce che il riferimento <rif\$> è il nome di una tabella appartenente al set della DBMIRROR classica (comprendente anche le tabelle delle videate personalizzate).

	= 1 si stabilisce che <rif\$> è il nome di un file di personalizzazione in cui sono codificate per ciascuna tabella le colonne che devono essere trattate nell'esportazione
rif\$	Nome di un file di personalizzazione in cui sono codificate per ciascuna tabella le colonne che devono essere trattate nell'esportazione; tale valore dipende dal parametro "tip";
dir\$	Directory dove salvare il file\$
file\$	Nome del file da generare
slot	Numero da 1 a 8 slot di esportazione presente nella tabella Stato anagrafiche del Gestionale
att	Record attivi: 0 = tratta tutti i record; 1 = solo se sono attivi
var	Variazione: = 0 non gestisce ottimizzazioni di export; = 1 esporta i soli record variati dall'ultimo aggiornamento; = 2 elimina i record dal database non gestendo lo stato del variato dello <slot>
azz	Azzeramento: = 0 non azzerava lo stato variato; = 1 azzerava lo stato variato;
Impo\$	Imposta informazioni

Istruzione analoga alla ASCMIRROR_EXT, con la possibilità di inserire una stringa tramite il parametro [impo\$] per impostare le informazioni nel file ASCII da generare, tramite i seguenti prefissi:

- SV = step di visualizzazione (equivalente al parametro [vis])
- CI = carattere separatore item (es. ;)
- CT = carattere delimitatore delle informazioni di tipo testo (es. “)

Nell'ASCMIRROR_EXT2 se si esportano i dati per differenza indicando <slot> e settando <var>, il file ottenuto conterrà solo i record trattati dalla esportazione rimuovendo eventuali altri record già esportati.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ATTIVAFIN

ATTIVAFIN <nfin>

Parametri	
nfin	Numero di finestra da attivare

Istruzione che attiva la finestra <nfin> precedentemente aperta (numero da 0 – base del gestionale – a 9; in Collage assume valore negativo per indicare le finestre che appartengono al gestionale) e posiziona il cursore in riga 1 e colonna 1. Disattiva finestre precedenti, in quanto solo una <nfin> alla volta è attiva per le istruzioni a video.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

AUTOFILE

AUTOFILE()

Funzione che ritorna il primo numero di file disponibile; se nessun numero è disponibile, la funzione ritorna 0. Si possono gestire fino a 15 file di cui 5 possono essere con chiavi (le chiavi non vengono conteggiate nel numero di file gestibili).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

AUTOFIN

AUTOFIN()

Funzione che ritorna il primo numero di finestra disponibile (0=nessuno). Esempio: NUMFIN = AUTOFIN()

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

AVANZAPF

AVANZAPF <bl>,<sbl>,<rifpfn>,<operazione\$>,<fase>,<qtapiu>,<datadoc\$>

Parametri	
bl	Numero bolla di lavorazione
sbl	Numero della sotto bolla di lavorazione
rifpfn	Numero del riferimento del prodotto finito
operazione\$	Operazione - valori ammessi (S)carico e (C)arico
fase	Numero della fase dell'operazione
qtapiu	Quantità da aggiungere ai movimenti - solo carico
datadoc\$	Data documento – se blank allora viene impostata la data terminale

Funzione che esegue gli avanzamenti di produzione. L'istruzione richiede MDS.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

AZZARRAY

AZZARRAY [num/str\$]

Parametri	
num/str\$	Variabile array numerica o alfanumerica

Azzerare il dimensionamento dell'array effettuato con comando DIM.

L'array non è gestito in parametri esecuzione [F6] e non è azzerato in campo di stampa [F2].

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#) o [Clicca qui](#)

AZZCACHE

Istruzione per la gestione della CACHE relativa al Dizionario. Se non viene eseguita non avviene l'aggiornamento in tempo reale delle variabili Dizionario. Le variabili di Struttura, invece, sono azzerate da apposite istruzioni relative a ciascuna singola struttura (AZZVARSYS oppure AZZCATVARSYS).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

AZZCRONO

AZZCRONO <nc>

Parametri	
nc	Numero cronometro

Istruzione che resetta il cronometro numero <nc>

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

AZZTAST

Istruzione che cancella i caratteri in sospeso nel buffer di tastiera, compreso quelli inviati con PUTTAST. Tutti i caratteri precedentemente inviati con l'istruzione PUTTAST non avranno più alcun effetto.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

AZZVARSYS

AZZVARSYS <NumeroStruttura>

Parametri	
1	Anagrafica Piano dei Conti (_PC). Clicca qui
2	Registrazione di Primanota (_PN). Clicca qui
3	Anagrafica Articoli (_AR). Clicca qui
4	Documento di magazzino (_MM). Clicca qui
5	Impegni di Lavorazione (_IL). Clicca qui
6	Bolla di Lavorazione (_OB e _OL). Clicca qui
7	Particolarità Conti e Articoli (_PART). Clicca qui
8	Distinta Base Primaria (_TP). Clicca qui
12	Anagrafica Indirizzi di Spedizione (_DA). Clicca qui
13	Anagrafica Contatti (_DC). Clicca qui
15	Lista Lotti (_LA). Clicca qui
17	Anagrafica Lotti (_LOT). Clicca qui
18	Anagrafica Tipo Lotto (_TLOT). Clicca qui
19	Movimenti di Analitica del controllo di gestione (_CDG). Clicca qui
20	Documenti Docuvision (_DV). Clicca qui
21	Codici Alias (_AL). Clicca qui

22	Codici Abbinati (_AB). Clicca qui
23	Anagrafica Commesse (_CO). Clicca qui
24	Anagrafica Aree (_AE). Clicca qui
25	Associazioni di Documenti Gestionale-Docuvision (_AD). Clicca qui
33	Referenti della Rubrica Clienti / Fornitori (_CFR). Clicca qui
34	Agenda di Lavoro (_AGP). Clicca qui
35	MyDB (_MYDB); Non azzera _MYDBK\$(). Clicca qui
36	Anagrafica unica (_ZA). Clicca qui
37	Anagrafica Gruppi Merceologici (_GM). Clicca qui
38	Sprix mobile, classe variabili form grafico (_WF). Clicca qui
39	Datatrek, variabili struttura (_DK). Clicca qui
41	Sprix mobile, classe variabili input grafico (_WI). Clicca qui
42	Sprix mobile, classe variabili lista grafica (_WL). Clicca qui
43	Sprix mobile, classe variabili oggetto grafico (_WOP). Clicca qui
44	Sprix mobile, classe variabili verifica esistenza archivio (_AF). Clicca qui
46	Autorizzazioni utente (_AU). Clicca qui
47	Chiamate Web Service (_WS). Clicca qui
51	Creazione Lista (_CT). Clicca qui
40	Calendario Agenda Attività Utente (_CM). Clicca qui
54	Elenco moduli stampa Sprix (_MODGRA). Clicca qui
55	Comandi richiesti su srv FTP/FTPS FTPSECCMD (_FTP)
56	Tabelle ABI e CAP (_ARCHVIEW) Clicca qui
64	Struttura modifica file XML fattura elettronica (_FTE)
69	Tabella Pagamenti (_PA) Clicca qui
72	Anagrafica Ubicazioni (_PU) Clicca qui
74	Progressivi articoli (_PAM) Clicca qui
75	Liste di Prelievo (_LST) Clicca qui
58	Condizioni documento (_PZD) Clicca qui
73	Avanzamento di produzione (_APF) Clicca qui

Istruzione che azzera le variabili della struttura archivi indicata dal parametro NumeroStruttura. E' analoga alla AZZCATVARSYS
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento (vedere ciascun link)

AZZCATVARSYS

AZZCATVARSYS <Struttura\$>

Parametri	
"PC"	Anagrafica Piano dei Conti. Clicca qui
"PN"	Registrazione di Primanota. Clicca qui
"PAG"	Tabella Pagamenti. Clicca qui
"AR"	Anagrafica Articoli. Clicca qui
"MM"	Documento di magazzino. Clicca qui
"IL"	Impegni di Lavorazione. Clicca qui
"BL"	Bolla di Lavorazione. Clicca qui
"PART"	Particolarità Conti e Articoli. Clicca qui
"DB"	Distinta Base Primaria. Clicca qui
"IND"	Anagrafica Indirizzi di Spedizione. Clicca qui
"CON"	Anagrafica Contatti. Clicca qui
"LOT"	Anagrafica Lotti. Clicca qui
"TLOT"	Anagrafica Tipo Lotto. Clicca qui

"CDG"	Movimenti di Analitica del Controllo di Gestione. Clicca qui
"DVDOC"	Documenti Docuvision. Clicca qui
"ALIAS"	Codici Alias. Clicca qui
"ABBIN"	Codici Abbinati. Clicca qui
"COMM"	Anagrafica Commesse. Clicca qui
"AEREA"	Anagrafica Aree. Clicca qui
"ASSOCDV"	Associazioni di Documenti Gestionale-Docuvision. Clicca qui
"CFR"	Referenti della Rubrica Clienti / Fornitori. Clicca qui
"AGP"	Agenda di Lavoro. Clicca qui
"MYDB"	Variabili MyDB. Non azzera _MYDBK\$. Clicca qui
"GM"	Anagrafica Gruppi Merceologici. Clicca qui
"WF"	Sprix mobile, classe variabili form grafico. Clicca qui
"WI"	Sprix mobile, classe variabili input grafico. Clicca qui
"WL"	Sprix mobile, classe variabili lista grafica. Clicca qui
"WO"	Sprix mobile, classe variabili oggetto grafico. Clicca qui
"AF"	Sprix mobile, classe variabili verifica esistenza archivio. Clicca qui
"AU"	Autorizzazioni utente. Clicca qui
"WEBSVC"	Chiamate Web Services. Clicca qui
"XML"	Gestione File XML. Clicca qui
"CT"	Creazione Lista. Clicca qui
"CM"	Calendario Agenda Attività Utente. Clicca qui
"MODSTPGRAF"	Elenco moduli stampa Sprix. Clicca qui
"FTP"	Comandi richiesti su srv FTP/FTPS FTPESECCMD (_FTP)
"ARCHVIEW"	Tabelle ABI e CAP Clicca qui
"FTE"	Struttura variabili modifica file XML fattura elettronica (_FTE)
"DTK"	Datatrek, variabili struttura (_DK) Clicca qui
"PU"	Anagrafica Ubicazioni (_PU). Clicca qui
"PAM"	Progressivi articoli (_PAM) Clicca qui
"LST"	Liste di Prelievo (_LST) Clicca qui
"PZD"	Condizioni documento (_PZD) Clicca qui
"APF"	Avanzamento di produzione (_APF) Clicca qui
"ZA"	Anagrafica unica (_ZA) Clicca qui

Istruzione che azzera le variabili della struttura archivi indicata dal parametro Struttura\$. E' analoga alla AZZVARSYS
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento (vedere ciascun link)

BEEP

Istruzione che emette un suono per richiamare l'attenzione. Richiede settaggio di apposita frequenza nel file ".ini" di configurazione.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

BLOCCASEM

BLOCCASEM <nomesem\$>

Parametri	
nomesem\$	<p>Nome alfanumerico del semaforo. Fino a 8 caratteri numerici e/o lettere maiuscole.</p> <ul style="list-style-type: none"> Aziendale (validi solo per terminali che hanno la stessa azienda aperta) Globale se inizia con carattere asterisco (*) significa che è valido per tutti i terminali collegati al sistema (quindi per tutte le aziende e non solo relativamente alla medesima azienda aperta) <p>Ogni terminale può bloccare contemporaneamente massimo 10 semafori (di cui massimo 3 globali)</p>

Istruzione che blocca il semaforo con nome <nomesem\$>; l'istruzione regola la semaforica di accesso al medesimo supporto dati (es. file) per evitare interferenze fra utenti e le sovrapposizioni nell'aggiornamento dei record. La semaforica per l'accesso di elaborazioni Sprix ai dati del gestionale, invece, è regolata come gli altri Client (concorrenza sul dato, regole di gestione). L'esito dell'istruzione è valorizzato in _ERRSEM.

Sono Preferibili le elaborazioni in memoria per poi bloccare e scaricare su disco (per evitare blocchi/sblocchi continui e/o prolungati). La regola generale per l'utente è questa:

- 1) elaborazioni in memoria
- 2) blocco il semaforo
- 3) controllo l'errore di semaforica (anche controllo su record)
 - 3.a) se c'è errore → attendo
 - 3.b) se non c'è errore → registro dati
- 4) sblocco semaforo (all'evento/all'uscita del programma)
- 5) end/return...azzeramento memoria

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CALL

CALL <nomefile\$>,<nomeetic\$>

Parametri	
nomefile\$	Nome del file Collage (senza estensione). Notare bene: Per approfondimenti sulla sintassi necessaria per l'ambiente di sviluppo App in PassBuilder consultare il parametro “-p” dell'istruzione SHELLSPRIX.
nomeetic\$	Etichetta della routine Collage; può appartenere a un altro Collage;

Istruzione che richiama e attiva l'esecuzione della routine stabilita nell'etichetta <nomeetic\$> del nome file Collage senza estensione <nomefile\$> e che sia già caricato in memoria (con AUTO con STARTCOLLAGE), pena l'interruzione della routine con messaggio “nomefile\$.es non caricato in memoria”. CALL sospende l'esecuzione del Collage caricato in memoria finché la routine richiamata non termina con END per poi riprendere; è simile alla GOSUB, ma viene terminata con END anziché RETURN.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CALLWEBSVC

CALLWEBSVC

Istruzione che permette di comunicare attraverso il Web Service con le applicazioni di terzi. L'esecuzione del comando “http” è gestito tramite la valorizzazione delle variabili _WS per la gestione delle chiamate dirette a Web Service. Es: CALLWEBSVC : IF _WSRAWERR\$ <> "" THEN VIMSG 5,_WSRAWERR\$: END

Gli header HTTP lunghi più di 255 caratteri vengono spezzati su più righe dell'array di output _WSOHEADER\$ (analogamente per l'array di input _WSHEADER\$), delimitandoli con due appositi meta-tag. I valori dei meta-tag si ottengono chiamando la funzione HTTPHDRTAG\$() a cui si passa il valore stringa “BEGIN” per ottenere il tag di inizio ed il valore “END” per ottenere il tag di fine.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CAMBIOVAL

CAMBIOVAL(<valp>,<vala>,<impo>,<dtcambio>,<numdec>)

Parametri	
valp	Valuta dell'importo (da 1 a 30, altrimenti ritorna 1)
vala	Valuta di conversione dell'importo (da 1 a 30, altrimenti ritorna 1)
impo	Importo (accetta anche 0)
dtcambio	Data del cambio (se non è valida viene assunta l'odierna)
numdec	Numero di decimali (da 0 a 6; se <0 è assunto 0, se >6 è assunto 6)

Funzione che converte l'importo <impo> espresso in valuta <valp> nella valuta <vala> richiesta, alla data del cambio <dtcambio> definita e con numero di decimali <numdec> richiesto.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CARCOLLAGE

CARCOLLAGE(<nomefile\$>)

Parametri	
nomefile\$	Nome del file Collage (senza estensione).

Notare bene: Per approfondimenti sulla sintassi necessaria per l'ambiente di sviluppo App in PassBuilder consultare il parametro “-p” dell’istruzione SHELLSPRIX.

Funzione che ritorna il numero di ordine nell’esecuzione del Collage caricato in memoria con nome file <nomefile\$> senza estensione; torna 0 se il file non è ancora caricato.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CERCACODFIN

CERCACODFIN(<IDfin\$>)

Parametri

IDfin\$	Identificativo finestra presente nelle "Informazioni su Passepartout" (visibili anche premendo due volte [F1]). Per esempio la lista degli articoli è "VISSART".
---------	--

Funzione che restituisce il numero di finestra in base all'identificativo della finestra. Restituisce i seguenti valori

- <= 0 rappresenta il numero di finestra Passepartout;
- > 0 rappresenta il numero di finestra Sprix/Collage;
- -1000 codice finestra non trovato;

Vedere anche GETCODFIN() e _NFIN

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CHIUDIFIN

CHIUDIFIN <nfin>

Parametri

nfin	Numero di finestra <nfin> da 1 a 9
------	------------------------------------

Istruzione che chiude il numero di finestra <nfin> da 1 a 9 (esclusa la 0) precedentemente aperta e attiva quella aperta immediatamente prima con posizione del cursore in riga 1, colonna 1. Chiude le finestre aperte successivamente e indipendentemente dalla sequenza, anche a caso (nfin è indicativo e non indica la sequenza. La Apertura di 6, 3, 5, 2 e la Chiusura di 5: chiude 5, 2 e attiva 3.

Nota: la sequenza di APRIFIN/CHIUDIFIN deve rispettare la gerarchia fra le finestre nelle seguenti regole: la chiusura di una finestra comporta la chiusura delle figlie; nell'intervallo fra l'apertura e la chiusura della finestra da parte del gestionale (es. l'estensione della descrizione sulla riga articolo) non può essere chiusa quella Collage (quella collage può vivere solo al interno o al esterno di questo intervallo).

Esempio:

momento T1 Aprifin Collage
 momento T2 Aprifin Gestionale
 momento T3 Chiudifin Gestionale
 momento T4 Chiudifin Collage

Se la gerarchia non è rispettato la routine Collage si interrompe e rilascia errore "impossibile chiudere la finestra".

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CHIUDISTAMP

CHIUDISTAMP

Istruzione che chiude la stampante precedentemente aperta. Valorizza _NUMSTAMP. CHIUDISTAMP può rilasciare il messaggio "immettere uno dei seguenti valori (S/N), ripetere l'immissione", quando si esce con ESC dalla routine Collage che, all'ON_612_? Su campo non vuoto esegue APRISTAMP (ad esempio 127) con LPRINT seguita dalla CHIUDISTAMP: normalizzare azzerando con AZZTAST. Azzerare le variabili dimensionali _STP.

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CHIUDISTAMP_EXT

CHIUDISTAMP_EXT

Istruzione che chiude il dispositivo di stampa e-mail precedentemente aperto con istruzione APRISTAMP_EMAIL. Valorizza le variabili di

errore _ERRCHSTAMP e _ERRCHSTAMP\$, _ERRIMAP e _ERRIMAP\$.

L'istruzione non è preclusa per altri dispositivi di stampa non e-mail aperti con APRISTAMP e le suddette variabili di errore sono utilizzabili anche in presenza di CHIUDISTAMP, ma in entrambi i casi si hanno i seguenti valori: _ERRCHSTAMP=0 e _ERRCHSTAMP\$="". Consultare anche le variabili di errore di memorizzazione in posta inviata _ERRIMAP e _ERRIMAP\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CHR

CHR\$(<num>)

Parametri	
Num	Numero di cui si vuole il valore ASCII

Funzione che ritorna una stringa equivalente al valore ASCII di <num> (da 0 a 255).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CIFLET

CIFLET\$(<num>)

Parametri	
num	Numero da convertire in lettere

Funzione che converte il numero <num> in lettere senza trattare il segno e decimali.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CIFLETDEC

CIFLETDEC\$(<num>,<numdec>)

Parametri	
num	Numero da convertire in lettera
numdec	Decimali

Funzione che converte il numero <num> (con decimali separati da ".") in lettere con decimali separati da "/" stabiliti in <numdec> (da 0 a 6). Valori Euro si convertono nella parte intera e i due decimali (anche se =00).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CLEARLOC

CLEARLOC

Istruzione SprixMobile di cancellazione dei dati in aree di memoria.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CLOSE

CLOSE <numfile>

Parametri	
numfile	Numero file

Istruzione che chiude il file numero <numfile> dopo che sono terminate le operazioni della sua elaborazione successive all'apertura, in modo da renderlo disponibile. Il termine del programma chiude il file.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CLOSECOM

Istruzione che chiude la porta precedentemente aperta per la lettura/scrittura.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CLOSECSV

<coderr> = CLOSECSV(<HND>)

Valore di ritorno	
coderr	valore di ritorno con eventuale codice di errore, lo stesso assegnato alla avariabile di sistema _ERRCSV
Parametri	
HND	Handle ottenuto dalla funzione OPENCSV()

Funzione che chiude il file CSV e rilascia le risorse usate dall'iteratore. La funzione valorizza le variabili di codice e descrizione errore _ERRCSV e _ERRCSV\$. Esempio: HERR=CLOSECSV(HND)

[esempio in comandi per argomento](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CLS

Istruzione che cancella tutto il contenuto della finestra attiva (se è quella base – 0 – le righe riservate 1,2,3,24,25 dello schermo fisico sono escluse). Le coperture a video avvengono con finestre con gestione del posizionamento del cursore all'uscita dei campi. La pulizia di campi sporchi avviene con la chiusura e riapertura della finestra.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CLS_EXT

CLS_EXT <parametri\$>,[titolo\$]

Parametri	
parametri\$	Stringa parametri separati da virgola: <ul style="list-style-type: none"> • "CCBORDI" = cancella anche i bordi; • "VISEZIONE" = visualizza la sezione sui bordi;
titolo\$	Titolo della sezione (se VISEZIONE in <param>); il parametro è opzionale

Istruzione che cancella le sezioni personalizzate definite con l'istruzione SEZIONEFIN.

Esempio: CLS_EXT "CCBORDI,VISEZIONE", "TITOLO 1"

In questo modo è consentito, quindi, ottenere la pulizia della finestra ed il contestuale disegno della sezione di default per tutta la finestra (verosimilmente ad APRIFIN senza "NOVISEZIONE").

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CODIF_BASE64

CODIF_BASE64 <TipoOp\$>,<File_Input\$>,<File_Output\$>

Parametri	
TipoOp\$	Tipo operazione di codifica o decodifica. Può assumere due valori 'E' = encode e 'D' = decode
File_Input\$	Parametro che deve contenere il percorso + il nome del file di Input (contenuto base64 se <TipoOp\$> = 'D', altro se <TipoOp\$> = 'E')
File_Output\$	Parametro che deve contenere il percorso + il nome del file di Output, il file verrà sovrascritto. Al termine il contenuto sarà la codifica Base64 del contenuto del file di Input se <TipoOp\$> = 'E', oppure, il contenuto binario del file di Input codificato Base64 se <TipoOp\$> = 'D'.

Istruzione che consente la codifica del contenuto di un file in Base64 o la decodifica del contenuto di un file codificato Base64. Eventuali errori incontrati dall'istruzione vengono riportati nella variabile di sistema _ERRBASE64.

Base64 è un sistema di codifica che consente la traduzione di dati binari in stringhe di testo ASCII, rappresentando i dati sulla base di 64 caratteri ASCII diversi. Viene usato principalmente come codifica di dati binari nelle e-mail, per convertire i dati nel formato ASCII.

L'istruzione è utile in Collage Server Remoto per gestire la transizione di file (es. immagini) fra Server e WebApp tramite i file Json. Per inviare immagini da Collage Server Remoto verso la webapp occorre passare tramite un file Json che, nonostante consenta di passare solo testo, non esclude la possibilità di convertire l'immagine in BASE64, cioè una sequenza di caratteri che vengono scritti come un qualsiasi altro testo. Successivamente, lato webapp, questi caratteri vengono ricodificati per riavere l'immagine e le pagine web elaborano anche il BASE64.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

COLOCBL

COLOCBL <NumBL>,<NumPF>,<NumRecRigaOC>

Parametri	
NumBL	Numero di Bolla di Lavorazione;

NumPF	Numero di Prodotto Finito;
NumRecRigaOC	Riga Ordine Cliente relativa al record;

Istruzione che consente il collegamento di una riga OC ad una BL; se il numero di Prodotto Finito <NumPF> non viene indicato (=0), la riga OC relativa al record <NumRecRigaOC> viene aggiunta al numero BL <NumBL>; se <NumPF> è <> da 0, <NumRecRigaOC> viene agganciata al PF con il <NumPF> specificato relativo al numero BL <NumBL>; controlli effettuati: che <NumBL> esista; che <NumPF> esista in <NumBL>; che <NumRecRigaOC> sia un record valido, la riga sia sospesa e l'articolo di produzione; se indicato <NumPF>, che i 2 codici siano uguali. L'istruzione valorizza le variabili `_ERRCOLOCBL` e `_ERRCOLOCBL$`

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

COLOR

COLOR <colore>

Parametri		
colore	Numero colore	Descrizione
	5, 6 o 18	Titolo sezioni
	8	Titolo colonne
	0	Testo normale
	2	Testo alta intensita'
	14	Testo disabilitato
	1 o 7	Testo evidenziato
	3	Riga selezionata / tendina
	15	Riga selezionata disabilitata
	9	Pulsante
	16	Pulsante disabilitato
	10	Link
	17	Link disabilitato
	11	Descrizione campo input
	4	Campo input
12	Campo input disabilitato	
13	Placeholder campo input	

Istruzione che imposta il colore di cursore e caratteri da visualizzare a video in base al numero <colore>. Il colore effettivo è determinato in funzione al colore di sfondo della finestra video. Esempio: A=2: COLOR A: PRINT "alta intensità"

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CONVPDFA

CONVPDFA <pdf\$>,<pdfa\$>,<type> [,<tit\$>,<aut\$>,<obj\$><app\$>,<datetime\$>]

Parametri	
pdf\$	Nome del file PDF sorgente (completo di path) da convertire;
pdfa\$	Nome del file PDF/A di destinazione (completo di path) da generare;
type	Tipo di PDF/A; campo numerico: 1 = PDF/A tipo 1B, 2 = PDF/A tipo 2B, 3 = PDF/A tipo 3B;
tit\$	Titolo da inserire nelle proprietà del file PDF generato (parametro opzionale);
aut\$	Autore da inserire nelle proprietà del file PDF generato (parametro opzionale);
obj\$	Oggetto da inserire nelle proprietà del file PDF generato (parametro opzionale);
app\$	Se diverso da 0 nella proprietà "Applicazione" del file PDF generato viene inserito il nome del prodotto Passepartout (parametro opzionale);
datetime	Se diverso da 0 nella proprietà "Creato il" del file PDF generato viene inserita la data e ora di creazione (parametro opzionale);

Istruzione che converte il file PDF nel formato file PDF/A. L'istruzione è sottoposta a tutte le condizioni ed i controlli che riguardano la gestione dei file in Sprix (vedere la variabile di accesso al file `_ACCFILEUTE`, di controllo errori `_ERRFILE$`, ecc..).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

COPYFILE

COPYFILE <nomefile1\$>,<nomefile2\$>

Parametri	
nomefile1\$	File da copiare
nomefile2\$	File copiato

Istruzione che copia il nome file <nomefile1\$> nel nome file <nomefile2\$> in cui sostituire i nuovi dati al posto dei precedenti. I file devono esistere ed essere chiusi, possono essere preceduti dalle rispettive directory. Se <nomefile1\$> non esiste la copia non è effettuata, se <nomefile2\$> non esiste viene creato. Per ulteriori gestioni di file consultare `INPUT_EXT` (tipo F/P) e altri comandi raggruppati per

argomento "DATABASE – FILE – COM – SHELL"

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CREACF

CREACF [cdfcogn\$],[cdfnome\$],[cdfdatn\$],[cdfsesso\$],[cdfcdcom\$],[cdfdscom\$],[cdfprov\$],[cdfiscale\$]

Parametri	
cdfcogn\$	Cognome, es. "ROSSI"
cdfnome\$	Nome, es. "PAOLO"
cdfdatn\$	Data di nascita in forma GGMMAAAA, es. "05061986"
cdfsesso\$	Sesso, es. "M"
cdfcdcom\$	Codice Comune, es. "H237"
cdfdscom\$	Descrizione Comune, es. "RIMINI"
cdfprov\$	Provincia, es. "RN"
cdfiscale\$	Codice Fiscale, es. "RSSPLA86C31I304U"

Funzione di Creazione e Scorporo del Codice Fiscale. Restituisce il Codice fiscale, oppure, passandole un Codice Fiscale [cdfiscale\$], i dati che da questo si possono desumere [SHF8]. La funzione gestisce le variabili di struttura _CDF e dispone delle variabili di errore _CDFERR e _CDFERR\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CREATELIST

CREATELIST <nomelista\$>,<modolista>,<riga>,<colonna>,[idfin\$]

Parametri	
nomelista\$	Stringa che rappresenta anche il titolo della finestra
modolista	Modalità della lista; accetta i valori 0, 1, 2. 0 = lista con selezione semplice; 1 = lista con multi selezione dove il tasto per la multi selezione è F9 accettazione dei valori tasto INVIO; 2 = lista con multi selezione dove il tasto per la multi selezione è INVIO accettazione dei valori tasto F10
riga	Riga del video in cui aprire la lista, se zero vale centro;
colonna	Colonna del video in cui aprire la lista, se zero vale centro;
idfin\$	Id finestra per mantenere il salvataggio delle impostazioni effettuate; è opzionale, se non indicato non è visibile la rotellina delle impostazioni elenco. Es. CREATELIST "Utenti", 0, 05, 8 , "IDFIN=CODFIN"

Istruzione che esegue automaticamente una finestra elenco i cui valori sono stabiliti dall'utente. L'istruzione si basa su variabili di struttura _CT che impostano il comportamento della lista, l'impostazione della colonna di ordinamento, il tipo di ordinamento, il posizionamento di partenza nella lista e filtro iniziale full text. L'istruzione valorizza le variabili di errore _ERRCT e _ERRCT\$.

In caso di colonna Icona la lunghezza del video viene automaticamente impostata a 2 caratteri e per i valori da impostare consultare la [lista di icone](#) elencate in SETTF. La navigazione della lista è la stessa della funzionalità standard della procedura.

Note:

Se prima di lanciare l'istruzione non viene creata una finestra, la creazione avviene sulla finestra base, altrimenti nella finestra.

Premendo F10 o Invio viene confermata, almeno, la riga su cui si è posizionati; se non si vogliono effettuare selezioni, occorre utilizzare il tasto ESC.

Gestione di tasti funzione personalizzati nella CREATELIST. Nota bene: sono eventi utilizzabili anche in SPRIX

La sintassi delle etichette per la gestione dei tasti funzione è la seguente: ON_<nnn>_<AAAAAAA>
(<nnn> = codice del tasto, <AAAAAAA>= valore della variabile _CTEVETIC\$)

Tale sintassi consente di scrivere nello stesso sorgente, la gestione di un determinato tasto per invocazioni diverse e separate di CREATELIST.

Definendo l'etichetta **ON_CAMBIOELEM_<Nome etichetta>**, questa viene invocata OGNI volta che la riga corrente cambia perche la si è modificata utilizzando i tasti di navigazione o altri tasti che ne causano la variazione (es: cancellazione riga corrente), le variabili _CTCURRIGA e _CTNUMRIGHE vengono aggiornate con i nuovi valori prima di invocare l'etichetta predisposta

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CREAUID

CREAUID\$(<pref\$>)

Parametri	
pref\$	Stringa che diviene prefisso nel calcolo dell'identificativo univoco universale;

Funzione che ritorna una stringa secondo le specifiche rfc4122 (Identificativo Univoco Universale); ha una forma del tipo 6639b9df-2c3d-

11e8-a41d-331406746252; il valore ritornato sarà sempre diverso essendo basato, fra l'altro, anche sul tempo. Esempi:

UID1\$=CREAUID\$("") si ottiene UID1\$="6639b9df-2c3d-11e8-a41d-331406746252"
 UID2\$=CREAUID\$("SPRIX") si ottiene UID2\$="SPRIX6639b9df-2c3d-11e8-a41d-331406746252"

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CRONO

CRONO(<nc>)

Parametri	
nc	Numero di cronometro

Funzione che legge il cronometro numero <nc> e ritorna il tempo trascorso in forma "secondi.millesimi", dall'ultimo azzeramento con AZZCRONO <nc>. Sono consentiti fino a 3 cronometri contemporaneamente.

Esempio: AZZCRONO 1 : CICLO: PRINT CRONO(1); IF CRONO(1)<10 THEN GOTO CICLO
 Esempio: Nell'elenco di anagrafica articoli (menu MA) viene contato il tempo dal primo ingresso nel campo "MASCEART"

```
ON_IN_MASCEART:
IF TRIG=0 THEN END
IF SCREEN$( _NFIN,30,5,3)=" " THEN VIMSG 4,"secondi, millisecondi = "+STR$(CRONO(1),-1,0)
END

ON_OUT_MASCEART:
IF TRIG=1 THEN END
TRIG=1 : AZZCRONO 1
END
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CSVITERNXT

<nr_rec> = CSVITERNXT(<HND>)

Valore di ritorno	
nr_rec	è il valore di ritorno in numero della riga letta
Parametri	
HND	Handle ottenuto dalla funzione OPENCsv()

Funzione di avanzamento dell'iteratore su file CSV alla prossima/successiva riga. La funzione valorizza le variabili di codice e descrizione errore _ERRCSV e _ERRCSV\$.

Esempio: CSVITERNXT(HND) 'Skip Riga intestazioni

NRREC ottenuto da CSVITERNEXT è semplicemente il numero di riga testè letta (non ha grande utilità, si possono contare anche in autonomia; la prima riga la si ottiene nel momento in cui si invoca la CSVITERSTRT che inializza l'iteratore e si trova posizionato sulla prima riga.

[esempio in comandi per argomento](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CSVITERSTRT

<num_colonne> = CSVITERSTRT(<HND>)

Valore di ritorno	
num_colonne	è il valore di ritorno in numero delle colonne di dati nelle righe del file CSV, dovrebbe ovviamente essere lo stesso per tutte le righe del file
Parametri	
HND	Handle ottenuto dalla funzione OPENCsv()

Funzione di avvio dell'iteratore sul file CSV già aperto per la lettura (se termina correttamente acquisisce già la prima riga e ritorna il numero di campi presenti nella riga).

Quando la funzione termina con successo viene ritornato un valore numerico che è il numero di colonne presenti in una riga del file CSV separate dal carattere separatore comunicato alla funzione OPENCsv(). L'esecuzione della funzione CSVITERSTRT() acquisisce automaticamente la prima riga di dati del file, se il file prevede una riga con l'intestazione delle colonne, sarà questa ad essere letta. La funzione valorizza le variabili di codice e descrizione errore _ERRCSV e _ERRCSV\$.

Esempio: NUMFLD = CSVITERSTRT(HND) ' avvio iteratore

[esempio in comandi per argomento](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CSVROWVAL

<valore> = CSVROWVAL(<HND>,<NrColonna>,<Trim\$>)

Valore di ritorno	
valore	Valore di ritorno della colonna richiesta nella riga attualmente in memoria, letta dal file dalle funzioni CSVITERSTR() e CSVITERNXT()
Parametri	
HND	Handle ottenuto dalla funzione OPENCSV()
NrColonna	Numero della colonna di cui si desidera acquisire il valore
Trim\$	Carattere che indica come effettuare il "trim" (elimina spazi non significativi) attorno al valore letto. Assume valori "D", "S", "T" (qualunque altro valore vale come vuoto) ed ha il significato di eliminazione degli spazi (blank, TAB, CR, LF) D=destra, S=SINISTRA, T=TUTTI (sia destra che sinistra del valore): "D" significa eliminare gli spazi a destra, per cui per es: " AA001 " viene restituito come " AA001" "S" significa eliminare gli spazi a sinistra, per cui per es: " AA001 " viene restituito come "AA001" "T" significa eliminare tutti gli spazi non significativi, per cui per es: " AA001 " viene restituito come "AA001" Sono considerati spazi i caratteri ; blank -0x20, TAB-0x08, CR-0x0d e LF-0x0a. Se passato vuoto ("") oppure con valori diversi da quelli sopra elencati, il valore viene restituito così come letto dal file.

Funzione che chiude il file CSV e rilascia le risorse usate dall'iteratore. La funzione valorizza le variabili di codice e descrizione errore _ERRCSV e _ERRCSV\$. Esempio:

READ_NXT:

VAL_FLD2\$= CSVROWVAL(HND, 2) ' Lettura seconda colonna

VAL_FLD3\$= CSVROWVAL(HND, 3) ' Lettura terza colonna

VAL_FLD4\$= CSVROWVAL(HND, 4) ' Lettura quarta colonna

PRINT VAL_FLD2\$+ " - "+VAL_FLD3\$+ " - "+VAL_FLD4\$ ' Stampa valori

Il valore letto sarà il valore al netto della presenza di delimitatori di stringa "" (virgolette) utilizzate per includere il carattere separatore nel valore ed anche le sequenze di escape per inserire anche le virgolette nel valore della colonna.

[esempio in comandi per argomento](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CTRARTTOUBI

CTRARTTOUBI <codart\$>,<idubi>,<idlotto\$>,[qta]

Parametri	
codart\$	Codice Articolo
idubi	Id ubicazione
idlotto\$	Id Lotto
qta	Quantità

Istruzione per controlli inserimento ubicazione su articolo. I primi tre parametri sono obbligatori (se non c'è il lotto mettere 0), la quantità è facoltativa (può essere indicata o meno). L'esito del controllo viene indicato nelle variabili di errore _ERRPU e _ERRPU\$

Se _ERRPU = 0 l'esito è positivo e l'articolo può essere inserito nell'ubicazione controllata. In caso contrario si ottiene un _ERRPU = 2 (esito negativo) e in _ERRPU\$ il motivo dell'esito negativo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CTRCODFIS

CTRCODFIS(<codfis\$>)

Parametri	
codfis\$	Codice Fiscale

Funzione che controlla il Codice Fiscale del conto <codfis\$>; ritorna 0 se errato, 1 se corretto. Esempio: CTCODFIS(pacfi("201.00003"))

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CTRPIVA

CTRPIVA(<partiva\$>)

Parametri	
partiva\$	Partita Iva

Funzione che controlla la Partita Iva del conto <partiva\$>; ritorna 0 se errata, 1 se corretta. La P.Iva è numerica (al massimo preceduta dai caratteri "IT", altrimenti è considerata straniera – valore = 1 -)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CVA

CVA\$(<str\$>,<numbyte>)

Parametri	
str\$	Stringa da elaborare
numbyte	Lunghezza in byte

Funzione che toglie dalla stringa <str\$> lunga <numbyte> tutti i caratteri ASCII = 32 partendo dalla destra fermandosi al primo carattere diverso da ASCII = 32. Se sono presenti caratteri ASCII = 0 la stringa viene troncata.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CVMAIU

CVMAIU\$(<str\$>)

Parametri	
str\$	Stringa da elaborare

Funzione che converte la stringa <str\$> in maiuscolo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CVMINU

CVMINU\$(<str\$>)

Parametri	
str\$	Stringa da elaborare

Funzione che Converta la stringa <str\$> in minuscolo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

CVN

CVN(<str\$>,<numbyte>,<numdec>)

Parametri	
str\$	Stringa da elaborare
numbyte	Lunghezza in byte
numdec	Numero decimali

Funzione che converte in numero con <numdec> decimali la stringa <str\$> lunga <numbyte>, impaccata precedentemente con MKN\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DBMIRROR

DBMIRROR <tabella\$>

Parametri	
Tabella\$	Tabella da elaborare

Istruzione che legge tutti i dati contenuti nei record del nome di <tabella\$> del dizionario esportando in sovrascrittura/precancellazione del supporto queste informazioni su file RDBMS(DBMIRROR).

Richiede la configurazione ODBC DRIVER MANAGER da specificare nella funzione Configurazione Moduli [F4] dei dati Anagrafici Aziendali; I campi data non valorizzati nel gestionale sono convertiti al 1/1/1900.

Viene visualizzato lo stato di avanzamento progressivo dei record trattati e si può interrompere con tasto [Esc] (previa impostazione di _INTER) con riempimento del file/database fino a quanto eseguito (l'interruzione dell'elaborazione della tabella corrente non interrompe gli eventuali mirroring successivi). **Non è eseguibile in ciclotabellare**, di cui valgono alcune regole, come ad esempio l'istruzione LEGGIDESC per

attivare la lettura delle righe descrittive nel documento e l'elaborazione annuale per anno di ingresso in azienda.

Attenzione: il mirroring (anche quello in tempo reale) dei documenti di anni diversi con chiave identica (ma che non comprenda l'anno, es. sigla+serie+numero+conto a numeratore unico e non annuale) si sovrascrivono.

Il mirroring di interi archivi impegna il sistema, se è già appesantito da altre condizioni (dbmirror in tempo reale, reti sovraccariche, computer non performanti) è preferibile l'export delle tabelle necessarie ai soli dati di interesse da esportare con procedure batch di fine lavoro oppure interrogazioni di database replicati in chiaro o infine mirroring ottimizzati.

Si ricorda che la funzione MEXALDB (che contempla la gestione degli indici) è un'alternativa alla funzione DBMIRROR. La DBMIRROR è un funzione sempre più in disuso. In passato, per risalire alla chiave logica utilizzata dal programma veniva rilasciata della documentazione tecnica a questo riguardo (tabelle, tracciati, diagrammi), ma ormai non più aggiornata.

Si ricorda che la struttura dei dati del Gestionale è consultabile nei seguenti modi:

- Consultare con editor di testo il file presente nel percorso 'programmi\binloc\dbmirror.dat' dell'installazione
- Le Tabelle e Colonne sono definite nel documento "ElencoColonneMexalDB.xls" allegato al chiarimento soluzione n. 30663 del servizio di assistenza EasyPass accessibile dall'Area Riservata di www.Passepartout.net. I contenuti delle colonne nel file vanno aperte ed estese. **Per approfondimenti consultare il capitolo MexalDB del Manuale Utente**
- I Tracciati CSV sono pubblicati nel chiarimento n. 54452 del supporto on-line. I Tracciati CSV sono verificabili nel menu del Gestionale [Servizi - Trasferimento Archivi - Export/Import CSV - ASCII]
- Specifiche dei campi Dizionario nel menu del Gestionale [Servizi - Personalizzazioni - Stampa Dizionario Sprix]

I dati obbligatori sono indicati a video dal Gestionale e le relazioni fra i dati sono descritte nel Manuale Utente

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DBMIRROR_EXT

DBMIRROR_EXT <tip>,<rif\$>,<slot>,<att>,<var>,<azz>,[vis]

Parametri	
<tip>	= 0 si stabilisce che il riferimento <rif\$> è il nome di una tabella appartenente al set della DBMIRROR classica (comprendente anche le tabelle delle videate personalizzate). = 1 si stabilisce che <rif\$> è il nome di un file di personalizzazione in cui sono codificate per ciascuna tabella le colonne che devono essere trattate nell'esportazione
<rif\$>	Nome di un file di personalizzazione in cui sono codificate per ciascuna tabella le colonne che devono essere trattate nell'esportazione; tale valore dipende dal parametro "tip";
<slot>	Numero da 1 a 8, slot di esportazione presente nella tabella Stato anagrafiche del Gestionale
<att>	Record attivi: = 0 tratta tutti i record, = 1 solo se sono attivi;
<var>	Variazione: = 0 non gestisce ottimizzazioni di export; = 1 esporta i soli record variati dall'ultimo aggiornamento; = 2 elimina i record dal database non gestendo lo stato del variato dello <slot>
<azz>	Variato: = 0 non azzerà lo stato variato; = 1 azzerà lo stato variato;
[vis]	Visualizzazione facoltativa dello stato di avanzamento progressivo dei record trattati (letti e scritti): se = -2 non visualizza, se = -1 visualizza al termine dell'elaborazione il totale dei record trattati, se = 1 o = 0 (o assente) visualizza la progressione dei record (step=1), se = N visualizza con step=N e al termine visualizza il totale di record trattati.

Istruzione analoga alla DBMIRROR, ma ottimizzata sulle tabelle Anagrafiche Piano dei Conti e Articoli, riducendo il volume dei dati esportato dalle tabelle: PC_PP_PA_ e AR_AH_AF_AX_AY_AL_AB_AW_AQ_AO_AE_.

La riduzione avviene con i flag <slot>, <att>, <var>, <azz>. Nel menu Aziende - Parametri di base - tabella Stato anagrafiche del Gestionale, si possono stabilire fino a 8 tipi di slot di esportazione (da utilizzare in altrettante applicazioni Sprix, esempio E-COMMERCE) per le quali definire, con la pressione del tasto [SHF11] e la selezione di 'Stato anagrafica' sul conto o sul articolo nelle rispettive anagrafiche, lo stato ATTIVO oppure NON ATTIVO dello <slot> sulla variazione del dato: il singolo record è quindi esportato o meno in base a quanto stabilito con lo <slot> sul flag di variazione del singolo record (che viene alzato dal Gestionale alla modifica del record). Nelle operazioni PUT la scrittura default è N per 'ATTIVO' e S per il 'VARIATO'; gli elementi 9-12 sono invece di sola lettura.

Lo <slot> è quindi un numero da 1 a 8; se = 0 non gestisce questa ottimizzazione attraverso il flag numerico <var> per l'export dei soli record variati dall'ultimo aggiornamento (se <var>=1) o di tutti i record con cancellazione (se <var>=0) o drop (se <var>=2) della tabella nel database (rendendo quindi irrilevante il flag 'Variato' dello <slot>). La Configurazione Moduli ([F4] in Anagrafica Azienda) contiene un parametro 'RIMOZIONE DATI CANCELLATI' che DBMIRROR_EXT consente di tenere traccia in un file log dbmi.sig (sig = sigla azienda) dei record rimossi nei movimenti di magazzino per il successivo allineamento quando si opera per differenza. Gli <slot> della DBMIRROR e ASCMIRROR sono in comune: una esportazione su database con DBMIRROR determina la non esportazione su file dell'ASCMIRROR sul medesimo slot.

Il <var> dipende comunque dal flag numerico <att> (se <att>=0 tratta tutti i record, se <att>=1 solo se sono attivi). Terminata l'esportazione, per evitare successivi ri-trattamenti del record, il <var> di tutti gli slot (relativi al intero archivio anagrafico e non della singola tabella) viene azzerato se <azz>=1 (se <azz>=0 non azzerà <var>), poiché se il flag numerico <tip> è = 0 si stabilisce che il riferimento <rif\$> è il nome di una tabella appartenente al set della DBMIRROR classica (comprendente anche le tabelle delle videate personalizzate).

Diversamente, con <tip>=1 si stabilisce che <rif\$> è il nome di un file di personalizzazione in cui sono codificate per ciascuna tabella le colonne che devono essere trattate nell'esportazione (se il file contiene il riferimento a più tabelle l'esportazione avverrà per ognuna di esse) e in cui è la medesima DBMIRROR_EXT a preoccuparsi di alzare il flag quando si elabora l'ultima tabella dell'archivio.

Il nome del file di personalizzazione richiede quello della tabella e della colonna separati da uno o più blank o tabulatore ASCII; le colonne nel file possono avere un ordine diverso da quello definito nella struttura della tabella nel database. L'ordine può essere casuale (tabelle intervallate da linee vuote) e sono inseribili dei commenti (marcatura con apice "" al inizio di una nuova riga).

Se col file di personalizzazione è possibile indicare quali colonne trattare (selezione in verticale di un'ipotetica matrice), con gli slot è possibile filtrare i record da trattare con DBMIRROR_EXT (selezione in orizzontale)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELABBIN

DELABBIN <codart\$>,<codab\$>,<nprog>,<cat\$>

Parametri	
codart\$	Codice articolo dell'abbinato; (se "" = tutti gli articoli);
codab\$	Codice abbinato; (se "" = tutti gli abbinati);
nprog	Numero progressivo per i codici abbinati doppi (se -1 = tutti, altrimenti specificare il numero di chiave che si intende prelevare: 0, 1, 2, ...);
cat\$	Categoria di abbinamento (se "" = tutte le categorie di abbinati);

Istruzione che cancella i dati dei Codici Abbinati relativi alle variabili di struttura archivi _AB in base al codice articolo <codart\$> a cui sono abbinati, al codice abbinato <codab\$>, al numero progressivo per codici abbinati doppi <nprog> e alla categoria di abbinamento <cat\$>. Non è consentita la cancellazione di tutti gli alias per tutti gli articoli. L'istruzione valorizza le variabili _ERRAB e _ERRAB\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DECODP7M

DECODP7M (<FileInput_p7m\$>, <FileOutput_cont\$>)

Parametri	
FileInput_p7m\$	Percorso + nome file contenente un dato a cui è stata apposta una firma digitale CADES;
FileOutput_cont\$	Percorso + nome file in cui verrà scritto il dato ripulito dai dati di firma;

Funzione per decodificare i file firmati p7m (CADES, acronimo di Cryptographic Message Syntax Advanced Electronic). La funzione estrae ("sbusta") i dati contenuti in file firmato con estensione ".p7m" eliminando tutte le informazioni aggiunte dal processo di firma; se il file di input è codificato Base64, questa funzione esegue la decodifica Base64 prima della decodifica p7m).

La funzione restituisce un valore numerico in base all'esito dell'estrazione e la variabile _ERRP7M\$ contiene la descrizione dell'eventuale errore, altrimenti è vuota: ritorna valore 0 (zero) se l'operazione è terminata correttamente, altrimenti un valore diverso da 0 (zero).

Esempio: RET=DECODP7M ("fattura.p7m", "fattura.xml") : IF RET <> 0 THEN VIMSG 4, _ERRP7M\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELAGP

DELAGP <nrec>,<dal\$>

Parametri	
nrec	Numero record (consente la ricerca diretta o la simulazione del ciclo);
dal\$	Da data lavoro "AAAMMGG";

Istruzione che cancella i dati delle righe di Agenda di Lavoro, relativi alle variabili di struttura archivi _AGP, in base ai parametri identificativi obbligatori <nrec> e <dal\$>. L'istruzione valorizza le variabili _ERRAGP e _ERRAGP\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELALIAS

DELALIAS <codart\$>,<codalis\$>,<nprog>

Parametri	
codart\$	Codice articolo dell'alias; (se "" = tutti gli articoli);
codalis\$	Codice alias; (se "" = tutti gli abbinati);
nprog	Numero progressivo per i codici alias doppi (se -1 = tutti, altrimenti specificare il numero di chiave che si intende prelevare: 0, 1, 2, ...);

Istruzione che cancella i dati dei Codici Alias relativi alle variabili di struttura archivi _AL sulla base al codice articolo <codart\$> a cui sono associati, al codice alias <codalis\$> e al numero progressivo per i codici alias doppi <nprog>. Non è consentita la cancellazione di tutti gli alias per tutti gli articoli. L'istruzione valorizza le variabili _ERRAL e _ERRAL\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELAR

DELAR <codart\$>

Parametri	
codart\$	Codice articolo;

Istruzione che cancella l'anagrafica dell'articolo in base al codice principale <codart\$>. Valorizza le variabili _ERRAR e _ERRAR\$
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELAASSOCDV

DELAASSOCDV <tipo>,<keypass>,<iddoc>

Parametri	
tipo	Tipo di Entità del Gestionale;
keypass	<p>Chiave di ricerca sulle tabelle di associazione fra Gestionale e Docuvision, eccetto per le classi <tipo>=1 Stampa generica, =800 Dichiarativi/Versamenti, =900 Stampe Fiscali; =1000 Agenda di lavoro ed eccetto i casi in cui il livello di azienda o il codice contratto non consentono l'accesso ad alcune tabelle, come esempio le operazioni di classe 'Parcella/Notula' senza gestione parcellazione, classi di Magazzino con azienda Contabile, ecc.):</p> <p>Piano dei Conti: <tipo>=400 <keypass>="<codcon\$>" (codice conto del piano dei conti);</p> <p>Primanota: <tipo>=600 <keypass> = "<dreg\$>;<prinr>;<cau\$>;<protr\$>;<prots>;<protn>;<docn>;<ddoc\$>" <dreg\$> (data di registrazione); <prinr> (n. progressivo registrazione primanota); <cau\$> (causale del documento); <protr\$>(registro di protocollazione); <prots> (serie del protocollo); <protn> (numero di protocollazione); <docn> (numero del documento); <ddoc> (data documento);</p> <p>Anagrafica Articoli: <tipo>=1200 <keypass> = "<codart\$>" (codice articolo);</p> <p>Movimenti di Magazzino: <tipo>=1400 <keypass> = "<sig\$>;<ser>;<num>;<clifor\$>;<ddoc\$>" <sig\$> (sigla documento); <ser> (serie del documento); <num> (numero del documento); <clifor\$> (codice conto intestatario); <ddoc\$> (data documento);</p> <p>Ordine/Preventivi/Matrici: <tipo>=1500 <keypass>="<sig\$>;<ser>;<num>;<ddoc\$>" <sig\$> (sigla documento); <ser> (serie del documento); <num> (numero del documento); <ddoc\$> (data documento);</p> <p>Bolla di Lavoro: <tipo>=1600 <keypass> = "<numeroBL>;<numeroSottoBL>" <numeroBL> (numero di bolla di lavoro); <numeroSottoBL> (numero di sottobolla di lavoro); Inserire "-1" per avere tutte le sottobolle, inserire "0" (o non specificare) per intendere la BL principale. In operazione DEL è richiesta anche la data della bolla.</p> <p>MyDB: <tipo>=1700 <keypass> = "<nome MyDB>;<numero elemento>" <nome MyDB> ("codice" (per mydb su app corrente) o "codiceApp@codice" (per mydb su altra app)); <numero elemento> (cod. numerico univoco che identifica l'n-esimo elemento di anagrafica (_MYDBID));</p> <p>Notule/Parcelle: <tipo>=1100 <keypass>="<cau\$>;<ser>;<num>;<clifor\$>;<ddoc\$>"; <cau\$> (causale del documento); <ser> (serie del documento); <num> (numero del documento); <clifor\$> (codice conto); <ddoc> (data documento);</p>
iddoc	ID del documento Docuvision

Istruzione che cancella i dati delle associazioni fra il Documento Docuvision e l'Entità del Gestionale, relativi alle variabili di struttura archivi _AD, in base all'ID del documento Docuvision <idDoc> (se <idDoc>=0 le elimina tutte) e il <tipo> di Entità del Gestionale, determinata univocamente dalla <keypass> di ricerca sulle tabelle d'associazione fra Gestionale e Docuvision. Le Entità del Gestionale appartenenti ad archivi annuali devono essere relative all'anno aperto.

Attenzione: la funzionalità è attiva per il modulo "Gestione documento", ma non per il modulo "Fatture elettroniche PA-B2B".

DELBL

DELBL <numeroBL>,<numeroSottoBL>

Parametri	
numeroBL	Numero Bolla di Lavorazione;
numeroSottoBL	Numero Sotto Bolla di Lavorazione;

Istruzione che cancella i dati della la Bolla di Lavorazione relativi alle variabili _OB (testata) e _OL (righe), in base al numero della bolla <numeroBL> e sottobolla <numeroSottoBL> di lavorazione (per le sottoBL dei lavoranti aggiungere 10000, esempio la sottobolla 0002 → 10002). La cancellazione di una BL senza indicare le sottoBL apre una finestra per confermare la cancellazione delle sottoBL collegate alla BL. L'istruzione valorizza le variabili ERRBL e ERRBL\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELCDG

DELCDG <TipoDoc\$>,<KeyDoc\$>,<Anno>,<NumRigaDoc>,<NumProgAna>

Parametri	
TipoDoc\$	Tipo movimento: (R=Primanota; M=Magazzino; C=Ord.Cliente; F=Ord.Fornitore; P=Preventivi; A=Matrici; E=Extra);
KeyDoc\$	La Chiave del movimento <KeyDoc\$> è formata da campi obbligatori di ricerca in base a <TipoDoc\$> (separati da ;) <ul style="list-style-type: none"> R – “PNDRE=x;PNND0=x;” (PNND0 = num. Progressivo registrazione primanota); M – “MMSIG=x;MMNUM=x;MMSAZ=x;MMSER=x;MMCLI=x;NTMM=x;” (M è in relazione all'anno di ingresso; MMCLI è facoltativo per i documenti non intestati; se NTMM – Num. Riferimento testata documento non è definito è impostato = 0); C/F/P/A – “MMSIG=x;MMNUM=x;MMSAZ=X;MMSER=x;”; E – “PRGEXTRA=x;DTCOMP=x;AREA=x;COMM=x;CODANL=x;TIPMOV=x;” PRGEXTRA non è obbligatorio e rappresenta un elemento progressivo per ottimizzare la chiave di ricerca in caso di movimenti identici ed è consultabile con il dizionario bgepr().
Anno	Anno del movimento: (contabile 1960 -2048 se tipo C/F, fisso 2048 se tipo E/P, fisso 1 se tipo A);
NumRigaDoc	Numero della riga del documento <NumRigaDoc> a cui sono associate le registrazioni di analitica (se 0=tutte le righe);
NumProgAna	Numero progressivo di riga <NumProgAna> relativo all'analitico (se 0=tutte le righe; in magazzino è sempre 0 dato che per ciascuna riga di movimento è consentita una sola riga di analitica mentre in primanota ne sono consentite di più);

Istruzione che cancella i dati dei movimenti di Contabilità Analitica del Controllo di Gestione relativi alle variabili _CDG, in base al Tipo movimento <TipoDoc\$> nell'anno del movimento <Anno> al numero della riga del documento <NumRigaDoc> a cui sono associate le registrazioni di analitica e al numero progressivo di riga <NumProgAna> relativo all'analitico. Valorizza variabili _ERRCDG e _ERRCDG\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELCM

DELCM <ute\$>,<data\$>,[id\$],[adata\$]

Parametri	
ute\$	Id Nome Utente; obbligatorio
data\$	Data di ricerca nel calendario, formato “AAAAMMGG”; obbligatoria
id\$	Id Agenda (numero record creazione);
adata\$	Data di fine ricerca nel calendario; formato “AAAAMMGG”; se inserita determina l'elaborazione di tutti i calendari dell'utente compresi fra le date impostate.

Istruzione che cancella dati del Calendario dell'Agenda Attività Utente. Valorizza le variabili di struttura _ERRCM e _ERRCM\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELCFR

DELCFR <cod\$>,<tipo>,[arch]

Parametri	
cod\$	Codice Cliente / Fornitore / Contatto;
tipo	Referente: 0 = tutti i referenti; 1 a 30 = un referente specifico
arch	“TP=0” esegue la lettura sull'archivio dei contatti in base al codice impostato

Istruzione che cancella i Riferimenti dei Referenti presenti nell'anagrafica della rubrica Clienti Fornitori (tasto funzione F4) o Contatto relativi alle variabili di struttura _CFR in relazione ai valori del Cliente/Fornitore <cod\$> in base al <tipo> (se 0 = tutti i referenti, da 1 a 30 = un referente specifico). L'istruzione valorizza le variabili _CFRERR e _CFRERR\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELCON

DELCON <codice\$>

Parametri	
codice\$	Codice contatto

Istruzione che cancella i dati di Anagrafica Contatti relativi alle variabili _DC in base al codice contatto <codice\$>. L'istruzione valorizza le variabili _ERRDC e _ERRDC\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELDB

DELDB <codart\$>,<numfase>

Parametri	
codart\$	Codice articolo con Distinta Base Primaria;
numfase	Numero di fase della Distinta Base Primaria;

Istruzione che cancella i dati della Distinta Base Primaria relativi alle variabili di struttura archivi _TP in base al numero di fase <numfase> della DBP (1-99) e del codice articolo <codart\$> con DBP. Le operazioni sulla DBP valorizzano _NCOMP (numero dei componenti di una fase: max 5000 per fase). La cancellazione è per tutte le fasi se <numfase> = 0 oppure solo dell'ultima se <numfase> <> 0. L'istruzione valorizza le variabili _ERRDB e _ERRDB\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELDVDOC

DELDVDOC <numdoc>,<numrev>

Parametri	
numdoc	Numero Documento di Docuvision;
numrev	Numero Revisione documento (in lettura se -1=tutte le revisioni, 0=ultima revisione, >0=revisione specificata; in cancellazione se 0=cancella l'intero documento). Le operazioni verificano i Parametri Generali di Docuvision poiché la gestione delle pagine è alternativa a quella delle revisioni di documenti già con la gestione pagine attiva è gestita solo l'ultima <numrev> contenuta in <numdoc> e quindi viene ignorata sia in lettura (i dati sono riferiti all'ultima revisione) che in cancellazione (quindi va valorizzata = 0).

Istruzione che cancella i dati del Documento Docuvision relativi alle variabili struttura archivi _DV, in base al numero Documento di Docuvision <numdoc> e numero revisione <numrev>. L'istruzione valorizza le variabili _ERRDV e _ERRDV\$.

Attenzione: la funzionalità è attiva per il modulo "Gestione documento", ma non per il modulo "Fatture elettroniche PA-B2B".

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELIL

DELIL <ndoc>,<nsot>,<nropf>,<fase>,<mag>,<cod>

Parametri	
ndoc	Numero della bolla di lavorazione, corrisponde alla variabile _ILNDO
nsot	Numero della sottobolla di lavorazione, corrisponde alla variabile _ILNSD
nropf	Numero riferimento prodotto finito, corrisponde alla variabile _ILNRB
fase	Numero di fase del componente, corrisponde alla variabile _ILFAS; (se 0 = tutte le fasi);
mag	Magazzino materie prime, corrisponde alla variabile _ILMAG; (se 0 = tutti i magazzini);
cod	Codice del componente, corrisponde alla variabile _ILAMP\$; (se "" = tutte le MP);

Istruzione che cancella i dati degli Impegni di Lavorazione relativi alle variabili di struttura archivi _IL in base al numero Bolla Lavorazione <ndoc>, il numero della sottobolla <nsot> e il numero di riferimento del prodotto finito <nropf>; la specificazione del codice della materia prima <cod> implica l'indicazione del numero di magazzino <mag> che, se specificato a sua volta implica il numero fase <fase>. Per effettuare la cancellazione dei residui di scarico (-1=RES_SL) o residui di deposito (-2=RES_DL) il numero di riferimento prodotto finito <nropf> non potendo essere un numero negativo, indicare per RES_SL il valore 65534 al posto di -1 e per RES_DL 65533 al posto di -2. L'istruzione valorizza le variabili _ERRIL e _ERRIL\$. **Notare:** non è possibile cancellare tutto l'archivio impegni, o tutti gli impegni di una

bolla di lavorazione con una sola istruzione DELIL.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELKEY

DELKEY <numfile>,<key\$>

Parametri	
numfile	Numero File Indici
key\$	Chiave principale

Istruzione che nel file a indici aperto con numero <numfile> cancella il record in base alla chiave stringa principale <key\$>; se la chiave non esiste la cancellazione non esegue alcuna operazione; tutte le chiavi sono obbligatorie e i loro valori non possono essere tutti ASCII 0 o ASCII 255. La Cancellazione e riscrittura di chiavi diverse possono richiedere tempi differenti in base a dimensione/struttura del file, posizione logica della chiave. Valorizza _NRDATI e _ERRIND.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELLOC

DELLOC <tipo\$>,<chiave\$>

Parametri	
tipo\$	Tipo di cancellazione: "T" = temporanea, "P"=persistente;
chiave\$	Nome della chiave

Istruzione SprixMobile relativa alla cancellazione selettiva della chiave. La variabile di riferimento relativa a messaggi di errore è _ERRLOC\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELLOT

DELLOT <idlot>

Parametri	
idlot	Numero ID lotto

Istruzione che cancella i dati di Anagrafica Lotto relativi alle variabili struttura archivi _LOT in base al numero ID lotto <idlot>. La cancellazione avviene se non è in uso da utente e/o non è stato caricato da movimento magazzino (in pratica è cancellabile se in anagrafica lotti è attivo il tasto [ShF3] per attivare la cancellazione dal modulo di servizio). Valorizza le variabili _ERRLOT e _ERRLOT\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELLSTDOC

DELLSTDOC <idlista>,<sigladoc\$>,<sottoazienda>,<serie>,<numdoc>,<rigaordine>,<progressivopick>,[param\$]

Parametri	
idlista	ID lista di prelievo
sigladoc\$	Sigla documento
sottoazienda	Numero sotto azienda
serie	Serie documento
numdoc	Numero documento
rigaordine	Numero di creazione riga ordine
progressivopick	Numero di pick
param\$	Parametro facoltativo: accetta valore "NOCTLSTATO" che consente di bypassare i controlli di lista e rimuovere dati indipendentemente dallo stato della lista. N.B: il parametro è ininfluenza sulle liste inserite da procedura ZERODO.

Istruzione che cancella i dati della Lista di Prelievo relativi alla struttura archivi _LST. L'istruzione valorizza le variabili _ERRLST e _ERRLST\$. L'istruzione richiede MDS.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELM

DELM <doc\$>

Parametri	
doc\$	Documento di Magazzino da scrivere nella forma Sigla[Serie/]Numero (la serie è facoltativa se è l'unica); la sintassi identifica documenti dell'anno solare dell'azienda aperta (per l'infrannuale, il parametro di magazzino "Numerazione

solare mov.mag (S/N) = N" consente di inserire l'anno del documento "nn" prima del protocollo SiglaSerie/nn-Numero, esempio "FT5/08-3432" – per la infrannuale aperta nel 2009 – o "FT5/09-3432" – per quella aperta nel 2008 -) In Sprix mobile la sintassi da utilizzare sui <u>movimenti di magazzino</u> è SiglaSerie/nn-Numero (con la specifica dell'anno anche se l'azienda non è infrannuale), mentre rimane SiglaSerie/Numero per le altre tipologie di documento.
--

Istruzione che cancella i dati del documento di magazzino relativi alla struttura archivi _MM, in base al documento <doc\$>. L'identificazione del documento presente in archivio più di una volta ma di clienti/fornitori diversi non apre la finestra di selezione del documento, ma viene cancellato il primo documento trovato (v. DELMM_EXT). L'istruzione valorizza le variabili _ERRMM e _ERRMM\$ <small>Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</small>

DELMM_EXT

DELMM_EXT <doc\$>[,conto\$[,cancpn]]

Parametri	
doc\$	Documento di Magazzino da scrivere nella forma Sigla[Serie/]Numero (la serie è facoltativa se è l'unica); la sintassi identifica documenti dell'anno solare dell'azienda aperta (per l'infrannuale, il parametro di magazzino "Numerazione solare mov.mag (S/N) = N" consente di inserire l'anno del documento "nn" prima del protocollo SiglaSerie/nn-Numero, esempio "FT5/08-3432" – per la infrannuale aperta nel 2009 – o "FT5/09-3432" – per quella aperta nel 2008 -) In Sprix mobile la sintassi da utilizzare sui <u>movimenti di magazzino</u> è SiglaSerie/nn-Numero (con la specifica dell'anno anche se l'azienda non è infrannuale), mentre rimane SiglaSerie/Numero per le altre tipologie di documento.
conto\$	Conto Cliente / Fornitore relativo al documento di magazzino
cancpn	1 cancella la Primanota collegata al documento magazzino (se primanota non è cancellabile il documento non è cancellato); 10 cancella documento magazzino, eventuale mov. bene usato; 11 cancella documento di magazzino, eventuale registrazione di primanota, eventuale mov. bene usato; 14 cancella documento magazzino, eventuale CDG; 15 cancella documento di magazzino, eventuale registrazione di primanota, eventuale CDG; 16 cancella documento di magazzino, eventuale registrazione di primanota, eventuale mov. bene usato , eventuale CDG.

Istruzione che cancella i dati del documento di magazzino relativi alla struttura archivi _MM, in base al documento <doc\$>. In caso di documento presente in archivio più di una volta ma di clienti/fornitori diversi l'istruzione consente di indicare il conto cliente/fornitore e l'eventuale cancellazione della primanota. Il conto identifica univocamente il documento evitando, quindi, l'apertura della finestra di selezione del documento; se non è indicato il conto opera come la DELMM. L'istruzione valorizza le variabili _ERRMM e _ERRMM\$ <small>Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</small>
--

DELPAG

DELPAG <codpag>

Parametri	
codpag	codice numerico del pagamento;

Istruzione che cancella i dati del codice pagamento <codpag> presente nella tabella pagamenti; riporta l'esito dell'operazione nelle variabili di errore _ERRPA e _ERRPA\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELPART

DELPART <tpart\$>,<codcon\$>,<catcon>,<codart\$>,<catart>,<dtvalda\$>

Parametri	
tpart\$	Tipo Particolarità: "P", "S", "V", "A", "I"; nota: la variabile è il parametro principale che stabilisce il tipo di particolarità in oggetto ed è composta da un carattere (P/V/S/A/I); la variabile prevede la possibilità di aggiungere anche un secondo ed un terzo carattere che condizionano altri parametri al fine di considerare combinazioni di ulteriori elementi della particolarità. Per la valorizzazione estesa di questo parametro fare riferimento all'istruzione GETPARTICOL. Per approfondimenti la voce PARTICOLARITA' del capitolo Comandi Raggruppati per Argomento. Clicca qui
codcon\$	Codice Conto Cliente/Fornitore;
catcon	Numero di categoria Listino, Sconto, Provvigione; (alternativo a <codcon\$>);
codart\$	Codice Articolo;
catart	Numero di categoria formazione Prezzo, Sconto, Provvigione; (alternativo a <codart\$>);
dtvalda\$	Data d'inizio validità della particolarità;

Istruzione che cancella i dati delle Particolarità (Prezzi, Sconti, Provvigioni, Articolo) relativi alle variabili struttura archivi _PART, in base al tipo particolarità <tpart\$>, al codice conto <codcon\$> o numero di categoria <catcon>, al codice articolo <codart\$> o numero di categoria

<catart> e con data d'inizio validità <dtvalda\$> della particolarità. La combinazione di quattro parametri individua la specifica particolarità; consente quattro combinazioni: <codcon\$>,<codart\$>; (<codcon\$>,<catart\$>); (<catcon\$>,<codart\$>); (<catcon\$>,<catart\$>). L'istruzione valorizza le variabili _ERRPART e _ERRPART\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELPC

DELPC <codcon\$>

Parametri	
codcon\$	Codice Conto di Anagrafica Piano dei Conti o Rubrica Clienti fornitori

Istruzione che cancella i dati del Conto di Anagrafica Piano dei Conti o Rubrica Clienti Fornitori relativi alle variabili di struttura archivi _PC, in base al codice conto <codcon\$>. Valorizza _ERRPC, _ERRPC\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELPN

DELPN <dreg\$>,<prnr>,<cau\$>,<protr\$>,<prots>,<protn>,<docn>,<ddoc\$>

Parametri	
dreg\$	Data di registrazione;
prnr	Numero progressivo di registrazione di primanota (1 – 16.777.214); solo GET; vedere _PNPRN
cau\$	Causale del documento;
protr\$	Registro di protocollo;
prots	Serie di protocollo;
protn	Numero di protocollazione;
docn	Numero del documento
ddoc\$	Data del documento

Istruzione che cancella i dati della registrazione di Primanota relativi alle variabili di struttura archivi _PN, in relazione ai parametri riportati in tabella. Valorizza _ERRPN, _ERRPN\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELPU

DELPU <idubi>

Parametri	
idubi	ID ubicazione

Istruzione che cancella i dati della struttura _PU in base allo ID ubicazione. Valorizza le variabili di errore _ERRPU, _ERRPU\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELTLOT

DELTLOT <tplot\$>

Parametri	
tplot\$	Codice Tipo Lotto; (in caso di nuovo tipo lotto indicare "***");

Istruzione che cancella i dati di Anagrafica Tipo Lotto relativi alle variabili struttura archivi _TLOT in base al codice Tipo Lotto <tplot\$>; la cancellazione del Tipo Lotto (in stato 'C' creato) richiede il Tipo Lotto <tplot\$>. L'istruzione valorizza le variabili _ERRTLOT e _ERRTLOT\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DELZA

DELZA <id>,<data\$>

Parametri	
id	ID anagrafica unica;
data\$	Data

Istruzione che cancella i dati storicizzati dell'Anagrafica Unica relativi alle variabili struttura archivi _ZA in base al codice anagrafico <id> e data storicizzazione <data\$>. L'istruzione non cancella i dati dell'archivio corrente e l'operazione valorizza le variabili _ERRZA e _ERRZA\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DESART

DESART\$(<codart\$>)

Parametri	
codart\$	Codice principale dell'Articolo;

Funzione che ritorna la descrizione dell'articolo (comprensiva di quella aggiuntiva) con codice principale <codart\$>. Lettere minuscole e caratteri di tipo "?" nel <codart\$> sono ignorati e considerati blank.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DIFDATA

DIFDATA(<data1\$>,<data2\$>)

Parametri	
data1\$	Stringa prima data (in formato "AAAAMMGG"); compreso fra 19600101 e 20481231;
data2\$	Stringa seconda data (in formato "AAAAMMGG"); compreso fra 19600101 e 20481231;

Funzione che ritorna la differenza in giorni fra le stringhe <data1\$> e <data2\$>. Il valore viene ritornato negativo se le <data1\$> è inferiore alla <data2\$>. Esempio: OGGI\$="20090620": IERI\$="20090619": D=DIFDATA(IEI\$,OGGI\$) risultato D = -1

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DIM

DIM <num>([dim1],dim2,dim3)

DIM <str\$>([dim1],dim2,dim3)

Parametri	
num	Variabile numerica relativa all'array; (alternativa a <str\$>);
str\$	Variabile stringa relativa all'array; (alternativa a <num>);
dim1	Prima dimensione; facoltativa;
dim2	Seconda dimensione; obbligatoria se dichiarata;
dim3	Terza dimensione; obbligatoria se dichiarata;

Funzione che dichiara le dimensioni [dim1], <dim2> e <dim3> dell'array <num> (numerico) o <str\$> (alfanumerico). Le dimensioni <dim2> e <dim3> sono obbligatorie se dichiarate, mentre la prima [dim1] è facoltativa, perché non dichiarandola si attribuisce all'array una natura automatica che consente risparmio di memoria, grazie all'allocazione fino all'elemento più alto utilizzato (memoria richiesta in automatico per quantità di elementi utilizzati), ma a discapito della velocità.

L'array, quindi, consente di dichiarare fino a 3 dimensioni (matrice di vettori) da dichiarare con DIM seguito dal nome della variabile array e i simboli di funzione "()" in cui indicare numeri costanti interi non in forma di variabile, rappresentativi delle dimensioni e del numero di elementi di ciascuna dimensione, fino a un massimo di 262080 per l'array stringa e di 131040 per quello numerico, entrambi ricavati dal prodotto degli elementi di ciascuna dimensione. Se la dimensione non è automatica, ma costante, nella dichiarazione della dimensione non si possono utilizzare valori superiori a 32767 elementi, in quanto non esegue la compilazione e dichiara "costante numerica troppo grande per la dimensione".

Il messaggio di errore " Indice dell'array fuori dai limiti " significa che gli indici hanno valori fuori dalla dimensione dichiarata oppure elementi il cui prodotto eccede il consentito. Spesso è un'iniziale dimensionamento dell'array non corretto, specialmente quando nei multidimensionali la prima dimensione è automatica, ma correlata alle restanti seconda e/o terza dimensione, pertanto occorre ridefinire le dimensioni iniziali, in modo da rientrare nei suddetti limiti. Nel caso di articoli a lotti a taglie e di articoli a lotti senza taglie, l'errore si presenta quando si raggiunge il limite massimo di righe inseribili da Sprix in un singolo documento; questo può verificarsi anche in caso di lettura tramite GETMM se il documento letto supera certi valori.

Un numero di elementi non corretti o utilizzati prima del loro dimensionamento (anche quando richiamato in libreria), provocano errori 'dimensioni dell'array troppo grandi', 'array non dimensionato'. Per una grande quantità di elementi su array monodimensionali è consigliata una DIM automatica (es. DIM VAR() anziché DIM VAR(99999), in quanto l'ottenimento della massima dimensione è previsto dall'intera array e raggiungibile in automatico.

L'array non è gestito in parametri esecuzione [F6] e non è azzerato in campo di stampa [F2].

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#) o [Clicca qui](#)

DISTAST

DISTAST <dis>

Parametri	
dis	Parametro on/off: = 1 disabilita; = 0 riabilita;

Istruzione che disabilita o riabilita l'input da tastiera; qualsiasi istruzione di input cancella la disattivazione. In base al flag <dis> (se = 1 disabilita la tastiera; se =0 la riabilita) disabilita o riabilita l'input da tastiera; la disattivazione della tastiera avviene automaticamente da una qualsiasi istruzione di input da tastiera. Utile per evitare che l'utente intervenga interrompendo l'esecuzione di qualche routine. Vedere anche _TF e INKEY.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DISVID

DISVID <dis>

Parametri

dis	Parametro on/off: = 1 disabilita; = 0 riabilita;
-----	--

Istruzione che disabilita o riabilita il video. Evita all'utente il disturbo da apertura/chiusura di finestre, spostamenti cursore con PUTTASTI. La DISVID è cancellata automaticamente da istruzioni input.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DOCMODCONT

DOCMODCONT()

Funzione per conoscere lo stato di aggiornamento contabile in fase di revisione del documento di magazzino.

Ritorna i seguenti valori, in cui il primo indica se ha registrato la primanota, il secondo se tale primanota è stata modificata rispetto a quella generata dal documento di magazzino:

- "NN" (non ha generato primanota e questa non è stata modificata)
- "SN" (ha generato primanota e questa non è stata modificata)
- "SS" (ha generato primanota e questa è stata modificata)
- "" (null, se fuori dal documento)

Esempio: F\$=DOCMODCONT()

L'utilizzo richiede le seguenti condizioni:

1. che i documenti registrati siano quelli del ciclo attivo che possano aggiornare la primanota (cioè con flag "Aggiorna contabilità" nel piede del documento, esempio: Fatture, Ric. Fiscali,...); non è attiva per i documenti che non registrano la primanota, oppure per i documenti del ciclo passivo (fornitori) o produzione o quelli che non consentono l'aggiornamento da magazzino (esempio "NO", "CO", ...)
2. che il documento sia in revisione (non in emissione) e con esecuzione della funzione nel piede, in quanto, se viene eseguita in testata/corpo potrebbe ritornare un valore sporco (occorrerebbe trattare i primi due caratteri dalla stringa)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DTK_GETCOLONNA

DTK_GETCOLONNA <nc>,<nstmt>,<nomecampo>,<elementobindcol>

Parametri

nc	Numero connessione;
nstmt	Numero statement;
nomecampo	Nome campo;
elementobindcol	Elemento della bindcol;

Istruzione che ritorna il valore del campo nell'elemento della bindcol precedentemente dichiarato. Dopo che sono state eseguite le normali istruzioni SQL, per accedere alle informazioni occorre utilizzare il comando SQL_FETCH (che in questo caso si posiziona sull'elemento del recordset da leggere) e occorre utilizzare anche la DTK_GETCOLONNA.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

DTK_GETVALUE

DTK_GETVALUE <LCobjectTable>,<LCfield>

Parametri

LCobjectTable	Label Collage relativa all'oggetto tabella;
LCfield	Label Collage relativa all'elemento campo;

Istruzione per interagire nella tablepage "edit" di Datatrek. Questa istruzione di Collage Datatrek sostituisce l'analoga GETATT??? del

Collage tradizionale in quanto non consentita in Collage Datatrek. L'istruzione legge i dati presenti a video nel momento in cui viene eseguita e ritorna i valori nelle variabili di struttura `_DK`. La struttura viene azzerata tramite `AZZVARSYS 39` o `AZZCATVARSYS "DTK"`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ELENCASTATOAPP

ELENCASTATOAPP

Istruzione per leggere nelle variabili di struttura della categoria `_APPSTS` l'elenco delle APP e relativo stato per l'azienda aperta. L'istruzione richiede MDS.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

EMAILGETATT

EMAILGETATT

Istruzione Collage per leggere nelle variabili di struttura della categoria `_EMAIL???` le impostazioni correnti di invio Email SMTP; Vedere etichetta evento `ON_STARTSTPEMAIL`:

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

EMAILPUTATT

EMAILPUTATT

Istruzione Collage chiamata dopo l'impostazione oppure la modifica delle variabili della categoria `_EMAIL???` per modificare le impostazioni correnti di invio Email SMTP. Vedere etichetta evento `ON_STARTSTPEMAIL`:

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ENCFREE

ENCFREE(<EncodingREF>)

Parametri

EncodingREF	Encoding di input dell'ambiente
-------------	---------------------------------

Istruzione che per Encoding Xml. Rilascia l'ambiente di encoding inizializzato con `ENCINIT()`

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ENCINIT

ENCINIT(<EncodingIN\$>,<EncodingOUT\$>)

Parametri

EncodingIN\$	Encoding di input
EncodingOUT\$	Encoding di output

Funzione per Encoding Xml. Inizializza l'ambiente di encoding, con encoding di input `<EncodingIN$>` ed encoding di output `<EncodingOUT$>`. Ritorna il riferimento all'ambiente di encoding o zero in caso di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ENCODING

ENCODING\$(<EncodingREF>,<TextIn\$>)

Parametri

EncodingREF	Encoding di input dell'ambiente
TextIn\$	Testo

Funzione per Encoding Xml. Codifica il testo `<TextIn$>` espresso con l'encoding di input dell'ambiente `<EncodingREF>` e ritorna il testo codificato con l'encoding di output. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array `_XMLSTRVAL$()` ed il numero degli elementi viene riportato in `_XMLVALNR`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

END

END

Istruzione di interruzione incondizionata del programma, simile alla pressione di [Esc/Fine], con chiusura di ciò che è aperto, file, finestre, stampanti, semafori, ecc. Non equivale alla chiusura naturale attraverso l'esecuzione dell'ultima riga di istruzione che non sia di tipo GOTO, GOSUB – RETURN, o simili, che conservano le variabili impostate per una successiva esecuzione (ad esempio come nelle elaborazioni multiaziendali se l'azienda elaborata non è l'ultima).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ESECMENU

ESECMENU(<menu\$>)

Parametri

menu\$	Voce di menu da eseguire; (eccetto la voce 'AF' e nemmeno per lancio di Sprix, che deve avvenire con SHELLSPRIX)
--------	--

Istruzione che apre automaticamente la voce di menu <menu\$> indicato. In Sprix, invece, è consentito lanciare il terminale configurato con un menu sul profilo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ESISFILE

ESISFILE(<nomefile\$>)

Parametri

nomefile\$	Nome del file di cui controllare l'esistenza; Nomenclatura File
------------	---

Funzione che controlla l'esistenza del nome file <nomefile\$>. Ritorna valore 1 se esiste, altrimenti 0. Gli ambienti Client/Server possono richiedere determinate sintassi di percorsi per non riferire l'operazione di default alla macchina Server (es. "\\...", "<\\...", "<C:\..."). Dalla versione v64.3 può accedere alle cartelle dati di PassBuilder con la sintassi cartella\$nomefile.

Per accedere alle cartelle dati, utilizzare la sintassi NomeCartella\$NomeFile.est (Il percorso viene risolto con "\ " finale dopo il nome cartella) esempio fileapp\$immagine.jpg. Per ulteriori approfondimenti si rimanda al manuale PassBuilder.

Per ulteriori gestioni di file consultare INPUT_EXT (tipo F/P) e altri comandi raggruppati per argomento "DATABASE-FILE-COM-SHELL"

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

EXITTAB

Istruzione per eseguire l'uscita forzata dal ciclo tabellare di lettura del Dizionario, in modo da proseguire con il codice successivo al ciclo. Esempio: l'istruzione può essere eseguita quando si verifica una determinata condizione all'interno del ciclotabellare.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

FIELD

FIELD <numfile>,<lungbuf> AS <varbuf\$> [tipoiimp],

Parametri

numfile	Numero file;
lungbuf	Lunghezza del campo (massimo 255 byte);
varbuf\$	Campo del record;
tipoiimp	Tipo di impaccamento (facoltativo): <ul style="list-style-type: none"> - "Nessuno" per variabile stringa; se alla scrittura il contenuto del campo è < di <lungbuf> il campo è completato per la parte mancante con i caratteri ASCII 0; se invece è più lungo sono assunti i caratteri a partire da sinistra, con troncamento della parte eccedente; - "ALFA" per variabile stringa; se alla scrittura e lettura il campo è automaticamente impaccato o disimpaccato: in lettura sono tolti tutti i caratteri ASCII 0 a destra; equivale ad usare la MKA\$ prima della scrittura e la CVA\$ dopo la lettura; - "da NUM0 a NUM6" per variabile numerica; utilizzare da NUM0 a NUM6 in base ai decimali della variabile. Nella scrittura il campo viene trascritto in caratteri ASCII allineato a destra. L'eventuale virgola decimale viene scritta con il carattere punto "." La dimensione specificata in <lungbuf> determina il valore massimo del numero memorizzabile (ogni cifra occupa un carattere); - "da BIN0 a BIN6" per variabile numerica; Utilizzare da BIN0 a BIN6 in base ai decimali della variabile. Il numero è scritto in binario; equivale a usare la MKN\$ prima della scrittura e la CVN dopo la lettura; per i campi ad impaccamento binario "BIN" <lungbuf> deve essere un numero compreso tra 1 e 5 e determina il valore massimo del numero memorizzabile in base alla tabella specificata nella funzione MKN\$.

Istruzione che dichiara la struttura del buffer associato al file numero <numfile> aperto con OPEN; la struttura si estrinseca nel nome della variabile <varbuf\$> che identifica il campo nel record (se è array l'indice non deve essere una variabile, ma costante, esempio VA\$(1)) e nella lunghezza in caratteri del campo <lungbuf> fino a un totale massimo di 128 campi per ciascun file. Se la lunghezza record <lr> della OPEN è maggiore della sommatoria delle varie<lungbuf> la differenza è assunta come tanti ASCII 0, diversamente l'esecuzione produrrebbe un errore. Se più di una FIELD fa riferimento allo stesso <numfile> sono considerate come un'unica FIELD di cui le successive, sono considerate come continuazioni di quelle precedenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

FINE

FINECO\$()

FINEES\$()

FINEGR\$()

FINEIT\$()

FINESO\$()

Funzioni che ritornano una stringa di caratteri da inviare alla stampante per impostare la fine del modo di stampa in compresso FINECO\$(), espanso FINEES\$(), grassetto FINEGR\$(), corsivo FINEIT\$(), sottolineato FINESO\$(). Non tutti tipi di formato consentono interventi sullo stile del font (esempio il sottolineato in pdf, corsivo o espanso in rtf, ecc). Esempio LPRINT FINECO\$();

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

FIRMAFILE

FIRMAFILE (<nomefile\$>, <nomefirma\$>, <utente\$>, <modo>)

Parametri	
nomefile\$	Stringa percorso e nome file da firmare
nomefirma\$	Stringa percorso e nome file firmato; se vuota "", sarà nomefile\$ con l'aggiunta dell'estensione ".p7m".
utente\$	Stringa flag di accesso utente; valori "S", "N" oppure vuota ""; se assegnato "S" attiva i controlli di autorizzazione di accesso utente, quindi il file può essere firmato solo se l'utente ha l'accesso al file.
modo	Numerico per modalità di firma <ul style="list-style-type: none"> se = 0 firma del documento; non entra nel merito del contenuto del documento, la firma racchiude interamente il file, compreso l'eventuale precedente firma che in esso sia contenuta; se = 1 firma in parallelo del documento; se si firma un file già firmato la firma attuale sarà in parallelo alla firma precedente;

Funzione FIRMAFILE per eseguire la firma digitale di un file mediante SmartCard. La funzione è analoga a quella richiamata per firmare i documenti di Docuvision e le fatture elettroniche con servizio in autonomia. Per utilizzare la funzione è necessario che sia stato installato e configurato il lettore di SmartCard con il programma di configurazione previsto dall'applicativo (v. menu Servizi-Configurazioni-Firma Digitale). Per la compilazione dei parametri di configurazione fare riferimento al manuale dell'applicativo. Se la firma è configurata di tipo interno la funzione esegue l'installazione automatica del modulo di gestione "SmartCard bit4id" se non presente. L'esecuzione corretta della funzione valorizza la variabile locale numerica con valore =1 altrimenti ritorna eventuali errori nella variabile _ERRFIRMAF\$. Esempio:

```
RET = FIRMAFILE("C:\temp\file.txt","",",",",",0)
IFF RET = 1
  VIMSG 4, "Firma eseguita"
ELSEF
  IFF _ERRFIRMAF$ = ""
    VIMSG 4, "Firma annullata"
  ELSEF
    VIMSG 4, "Errore firma. _ERRFIRMAF$ = " + _ERRFIRMAF$
  ENDF
ENDF
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

FIX

FIX(<num>,<numdec>)

Parametri	
num	Numero da troncare;
numdec	Numero dei decimali al quale troncare;

Funzione che tronca i decimali del <num> al numero <numdec> di decimali (da 0 a 6) senza arrotondare. È automatica con <numdec> = 6 sul risultato ottenuto, pertanto, in caso esecuzione di operazioni algebriche su valori con decimali può rilasciare risultati disattesi; questo automatismo è condizionato con SETDECPIX. Una chiamata esplicita di FIX ha priorità sul FIX eseguito automaticamente o condizionato con SETDECPIX.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

FOR NEXT

FOR <varnum>=<x> TO <y> [STEP <z>] : '...' : NEXT

Parametri	
varnum	Variabile numerica contatore:
x	Valore iniziale contatore;
y	Valore finale contatore;
z	Valore incrementale del contatore;

Istruzione che esegue espressioni contenute in '...' un numero definito di volte stabilite dalla differenza fra il valore iniziale <x> e valore finale <y> del contatore numerico <varnum> (no array). Al termine di ogni ciclo, <x> si incrementa di <z> (default dell'incremento del contatore = 1 e può assumere anche valore < 0) e il ciclo riprende dall'istruzione che inizia con la specifica FOR (se <x> <= <y>) altrimenti il programma prosegue alla prima istruzione successiva al comando NEXT (non modificabile, in quanto riferita all'ultima FOR aperta). FOR-NEXT è nidificabile (una nell'altra fino ad 8) dove la NEXT della FOR interna precede la NEXT della FOR esterna.

Nota: se il ciclo non è soddisfatto (ciclo null), l'istruzione viene eseguita almeno una volta. Esempio FOR I=1 TO LOF(n) con LOF(n)=0. È comunque possibile controllare il valore finale <y> per eseguire o saltare il ciclo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

FTPESECCMD

FTPESECCMD

Istruzione FTPESECCMD per invocare comandi FTP. Si avvale delle variabili di struttura _FTP per interagire da Sprix e Collage con Server FTP / SFTP (consigliata la compilazione del [Gestionale 64bit](#), poiché in S.O. 32bit non recenti potrebbe non funzionare) / FTPS invocandone gli specifici comandi del protocollo, per i percorsi e per eseguire letture, scritture e cancellazioni.

La comunicazione SSL/TLS verrà stabilita sia che venga specificato come protocollo "ftps" sia che il protocollo sia "ftp" e venga settata la variabile _FTPSSLTLS\$ con uno dei valori ammessi diverso da "NONE". Diversamente, non avviene la stessa cosa, richiedendo di usare il protocollo "ftps", la comunicazione parte con l'handshake SSL/TLS e poi prosegue nella maniera che le opzioni richieste impongono (vedere il controllo del control channel effettuata con il valore di _FTPCLRCMDC\$), mentre invece, richiedendo il protocollo "ftp" e settando il valore di _FTPSSLTLS\$ = "ALL", la comunicazione inizia in chiaro e avviene più avanti (prima dell'invio delle credenziali) fatta transitare dentro una sessione SSL/TLS, una modalità simile alla STARTTLS di SMTP. Nel caso in cui _FTPSSLTLS\$ = "TRY", la comunicazione avverrà in ogni caso, sia che lo strato SSL/TLS sia disponibile e vada a buon fine l'handshake, sia che questo non sia possibile, ed in quest'ultimo caso la comunicazione avverrà in chiaro. Il valore della variabile _FTPCLRCMDC\$ andrà settato dovendosi "accordare" soprattutto con gli agenti che consentono di transitare in rete (NAT, PROXY, FIREWALL), per esempio, un firewall in cui è settato _FTPCLRCMDC\$ = "NONE" non consente di transitare e abbatte immediatamente la connessione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GEOMAPPA

GEOMAPPA [LINK\$],[IND\$],[LOC\$],[CAP\$],[PROV\$],[ISONAZ\$],[NAZIONE\$]

Parametri	
LINK\$	Url del collegamento alla mappa; se è indicato "Http" o "HttpS" l'istruzione forza il collegamento della query su Google Maps, diversamente attiva la personalizzazione dell'URL tramite i restanti parametri;
IND\$	Indirizzo Civico; IND\$="Via Davanzati, 4"
LOC\$	Località; es. LOC\$="Rimini"
CAP\$	Codice di Avviamento Postale; CAP\$="47923"
PROV\$	Provincia; PROV\$="RIMINI"
ISONAZ\$	Sigla Nazione; es. ISONAZ\$="IT"
NAZIONE\$	Nazione; es. NAZIONE\$="ITALIA"

Istruzione per aprire la mappa di Google. Il comando ha sette parametri non obbligatori. Se il primo parametro non è "Http" o "HttpS" si può richiamare un provider diverso da Google Maps, concatenando i restanti parametri al fine di ottenere la composizione dell'URL desiderato: URL disponibile = LINK\$+IND\$+LOC\$+CAP\$+PROV\$+ISONAZ\$+NAZIONE\$.

Esempio di forzatura:

LINK\$ = "https://www.google.it/maps/place/Via+Bernardo+Davanzati,+4,+47923+Rimin+RN/@44.0451324,12.5978283,17z...
.../data=!3m1!4b1!4m5!3m4!1s0x132cc3223bed7645:0x72aab17ada609a3e!8m2!3d44.0451324!4d12.600017"
IND\$="" : LOC\$="" : CAP\$="" : PROV\$="" : ISONAZ\$="" : NAZIONE\$=""
GEOMAPPA LINK\$,IND\$,LOC\$,CAP\$,PROV\$,ISONAZ\$,NAZIONE\$

Esempio senza forzatura:

LINK\$="" : IND\$="Via Davanzati, 4" : LOC\$="Rimini" : CAP\$="47923" : PROV\$="RIMINI" : ISONAZ\$="IT" : NAZIONE\$="ITALIA"
GEOMAPPA LINK\$, IND\$,LOC\$,CAP\$,PROV\$,ISONAZ\$,NAZIONE\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GET

GET <numfile>,<numrec>

Parametri	
numfile	Numero del file sequenziale (di tipo "R") aperto;
numrec	Numero di record del file sequenziale (di tipo "R") aperto;

Istruzione che legge il numero di record <numrec> nel file aperto di tipo "R" numero <numfile>, utilizzando le variabili del buffer associato: in lettura sono riempite con i relativi valori letti (oppure tutte impostate a "" o 0 se il <numrec> è inferiore a 1 o superiore al numero di record presenti nel file).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETABBIN

GETABBIN <codart\$>,<codab\$>,<nprog>,<cat\$>

Parametri	
codart\$	Codice articolo dell'abbinato; (se "" = tutti gli articoli);
codab\$	Codice abbinato; (se "" = tutti gli abbinati);
nprog	Numero progressivo per i codici abbinati doppi (se -1 = tutti, altrimenti specificare il numero di chiave che si intende prelevare: 0, 1, 2, ...);
cat\$	Categoria di abbinamento (se "" = tutte le categorie di abbinati);

Istruzione che legge i dati dei Codici Abbinati relativi alle variabili di struttura archivi _AB in base al codice articolo <codart\$> a cui sono abbinati, al codice abbinato <codab\$>, al numero progressivo per codici abbinati doppi <nprog> e alla categoria di abbinamento <cat\$>. La lettura rilascia _NAB cioè il numero di elementi dell'array dinamico della struttura _AB. L'istruzione valorizza le variabili _ERRAB e _ERRAB\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETAGP

GETAGP <nrec>,<dal\$>,<ala\$>,<cli\$>,<pre\$>,<ute\$>,<sta\$>,<ori\$>,<azi\$>

Parametri	
nrec	Numero record (consente la ricerca diretta o la simulazione del ciclo);
dal\$	Da data lavoro "AAAAMMGG";
ala\$	A data lavoro "AAAAMMGG" se vuota viene posta a "dal\$";
cli\$	Codice Cliente "GMM.CCCCC";
pre\$	Codice prestazione;
ute\$	Utente "Nome di ricerca";
sta\$	Stato Lavoro;
ori\$	Origine;
azi\$	Sigla azienda;

Istruzione legge i dati delle righe dell'Agenda di Lavoro relativi alle variabili di struttura archivi _AGP, in base ai parametri identificativi obbligatori. L'istruzione ritorna un array di elementi contenuti nella variabile _AGPNELE. L'istruzione valorizza le variabili _ERRAGP e _ERRAGP\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETALIAS

GETALIAS <codart\$>,<codalis\$>,<nprog>

Parametri	
codart\$	Codice articolo dell'alias; (se "" = tutti gli articoli);
codalis\$	Codice alias; (se "" = tutti gli abbinati);
nprog	Numero progressivo per i codici alias doppi (se -1 = tutti, altrimenti specificare il numero di chiave che si intende prelevare: 0, 1, 2, ...);

Istruzione che legge i dati dei Codici Alias relativi alle variabili di struttura archivi _AL in base al codice articolo <codart\$> a cui sono associati, al codice alias <codalis\$> e al numero progressivo per i codici alias doppi <nprog>. La lettura rilascia la _NAL per conoscere il numero di elementi dell'array dinamico della struttura _AL. L'istruzione valorizza le variabili _ERRAL e _ERRAL\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETAR

GETAR <codart\$>,<nummag>

Parametri	
codart\$	Codice articolo;
nummag	Numero magazzino; se = 0 indica il totale dei magazzini (dato di calcolo e non memorizzato);

Istruzione che legge i dati dell'Anagrafica Articolo relativi alle variabili di struttura _AR in base al codice principale <codart\$> e gli eventuali progressivi del numero magazzino <nummag>. I progressivi sono in sola lettura (eccetto l'inventario nel primo anno dell'articolo) e sono modificabili solo con documenti rettificativi. L'istruzione valorizza le variabili _ERRAR e _ERRAR\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETARCHFLD

GETARCHFLD <archivio\$>,<campo\$>

Parametri	
archivio\$	Nome dell'archivio per il quale verificare l'esistenza del campo;
campo\$	Nome del campo di cui verificare l'esistenza.

Istruzione che legge i dati dell'Anagrafica Esistenza campi su archivio mobile relativi alle variabili struttura archivi _AF. Il campo archivio va espresso in maiuscolo replicando il nome file senza estensione, ad esempio ARTI per l'anagrafica articoli, mentre il campo va espresso con l'equivalente della struttura sprix a cui è associato, ad esempio _ARCOD\$ per il campo codice articolo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETAREA

GETAREA <area>

Parametri	
area	Codice area;

Istruzione che legge i dati dell'Anagrafica Aree relativi alle variabili struttura archivi _AE, in base al codice <area>. L'istruzione valorizza le variabili _ERRAE e _ERRAE\$. La possibilità di cancellazione dell'area è attraverso il modulo standard dei Servizi di Controllo di Gestione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETARVAR

GETARVAR [codart\$],<dadata\$>[,adata\$]

Parametri	
codart\$	Codice articolo da cui partire per trovare il prossimo articolo variato; se = "" viene ritornato il primo variato;
dadata\$	Data di partenza da cui considerare la variazione;
adata\$	Data di termine entro cui considerare la variazione;

Istruzione che legge gli articoli e ne ritorna quelli variati a partire <dadata\$>. L'istruzione valorizza le variabili _ERRARVAR, _ERRARVAR\$ e _CODARVAR\$ ovvero la variabile che, a partire dal codice successivo a <codart\$> (se indicato) oppure dal primo (se <codart\$>=""), ritorna il prossimo articolo variato dopo <dadata\$> e prima di [adata\$] (se questa è impostata).

Nota: la conferma (con F10) della revisione di un articolo senza apportare alcuna modifica all'anagrafica, comporta comunque una riconferma e una variazione della data di ultimo aggiornamento. Questa variazione non avverrebbe invece con l'abbandono della revisione (con Esc).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETASSOCDV

GETASSOCDV [<tipo>,<keypass>,<iddoc>

Parametri	
tipo	Tipo di Entità del Gestionale; in lettura è obbligatorio se viene indicata la <keypass>;
keypass	Chiave di ricerca sulle tabelle di associazione fra Gestionale e Docuvision, eccetto per le classi <tipo>=1 Stampa generica, =800 Dichiarativi/Versamenti, =900 Stampe Fiscali; =1000 Agenda di lavoro ed eccetto i casi in cui il livello di azienda o il codice contratto non consentono l'accesso ad alcune tabelle, come esempio le operazioni di classe 'Parcella/Notula' senza gestione parcellazione, classi di Magazzino con azienda Contabile, ecc.): Piano dei Conti: <tipo>=400 <keypass>="<codcon\$>" (codice conto del piano dei conti); Primanota: <tipo>=600 <keypass> = "<dreg\$>;<prinr>;<cau\$>;<protr\$>;<prots>;<protn>;<docn>;<ddoc\$>" <dreg\$> (data di registrazione);

	<p><prin> (n. progressivo registrazione primanota); <cau\$> (causale del documento); <protr\$>(registro di protocollazione); <prots> (serie del protocollo); <protn> (numero di protocollazione); <docn> (numero del documento); <ddoc> (data documento);</p> <p>Anagrafica Articoli: <tipo>=1200 <keypass> = "<codart\$>" (codice articolo);</p> <p>Movimenti di Magazzino: <tipo>=1400 <keypass> = "<sig\$>;<ser>;<num>;<clifor\$>;<ddoc\$>" <sig\$> (sigla documento); <ser> (serie del documento); <num> (numero del documento); <clifor\$> (codice conto intestatario); <ddoc\$> (data documento);</p> <p>Ordine/Preventivi/Matrici: <tipo>=1500 <keypass>="<sig\$>;<ser>;<num>;<ddoc\$>" <sig\$> (sigla documento); <ser> (serie del documento); <num> (numero del documento); <ddoc\$> (data documento);</p> <p>Bolla di Lavoro: <tipo>=1600 <keypass> = "<numeroBL>;<numeroSottoBL>" <numeroBL> (numero di bolla di lavoro); <numeroSottoBL> (numero di sottobolla di lavoro); Inserire "-1" per avere tutte le sottobolle, inserire "0" (o non specificare) per intendere la BL principale. In operazione DEL è richiesta anche la data della bolla.</p> <p>MyDB: <tipo>=1700 <keypass> = "<nome MyDB>;<numero elemento>" <nome MyDB> ("codice" (per mydb su app corrente) o "codiceApp@codice" (per mydb su altra app)); <numero elemento> (cod. numerico univoco che identifica l'n-esimo elemento di anagrafica (_MYDBID));</p> <p>Notule/Parcelle: <tipo>=1100 <keypass>="<cau\$>;<ser>;<num>;<clifor\$>;<ddoc\$>"; <cau\$> (causale del documento); <ser> (serie del documento); <num> (numero del documento); <clifor\$> (codice conto); <ddoc> (data documento);</p>
iddoc	ID del documento Docuvision

Istruzione che legge i dati delle Associazioni fra il Documento Docuvision e l'Entità del Gestionale, relativi alle variabili di struttura archivi _AD, in base all'ID del documento Docuvision <idDoc> e il <tipo> di Entità del Gestionale, determinata univocamente dalla <keypass> di ricerca sulle tabelle d'associazione fra Gestionale e Docuvision. Le Entità del Gestionale appartenenti ad archivi annuali devono essere relative all'anno aperto. GETASSOCDV valorizza _ADNUMASSOC che ritorna il numero globale d'associazioni.

Attenzione: per una migliore individuazione dell'associazione è importante concatenare i parametri riassumendoli in un'unica variabile stringa d'appoggio. Esempio: fattura emessa, sezionale 1, numero documento 10, cliente 501.00005, del primo marzo 2019.

```
VAR$="FT;1;10;501.00005;20190301" oppure
VAR$=_MMSIG$+";" + STR$(_MMSER,0,0)+";" +STR$(_MMNUM,0,0)+";" +_MMCLI$+";" +_MMDAT$
GETASSOCDV 1400,VAR$,0
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTAR

GETATTAR

Istruzione che legge i dati dell'Anagrafica Articolo presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _AR caricate dal Gestionale all'atto dell'utilizzo dell'istruzione; questa azzera le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTBL

GETATTBL

Istruzione che legge i dati della Bolla di Lavorazione presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _OB (testata) e _OL (righe) caricate dal Gestionale all'atto dell'utilizzo dell'istruzione; questa azzera le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTCFR

GETATTCFR

Istruzione che legge i dati dei Riferimenti dei Referenti presenti a video nell'Anagrafica della Rubrica Clienti Fornitori (tasto funzione F4) nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _CFR caricate dal Gestionale all'atto dell'utilizzo dell'istruzione; questa azzera le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTCOMM

GETATTCOMM

Istruzione che legge i dati relativi alla Anagrafica Commessa presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _CO caricate dal Gestionale all'atto dell'utilizzo dell'istruzione; questa azzera le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTCON

GETATTCON

Istruzione che legge i dati di Anagrafica Contatti presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _DC caricate dal Gestionale all'atto dell'utilizzo dell'istruzione; questa azzera le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTIL

GETATTIL

Istruzione che legge i dati degli Impegni di Lavorazioni presenti a video (menu 'PB e 'PV') nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _IL caricate da Gestionale all'atto dell'utilizzo dell'istruzione; questa azzera le variabili preesistenti ed inizializza la struttura con i valori esistenti.

L'istruzione opera solamente su un singolo record (Impegno), nella revisione del Impegno la Bolla è bloccata, non modificabile. Essendo la GETATTIL un funzione mono record il valore dell'array è sempre 1.

Non è realizzabile una istruzione in grado di leggere tutti gli impegni della bolla; paragonando ad emissione revisione documenti, gli impegni stanno alle anagrafiche degli articoli come la bolla di lavorazione sta al documento di magazzino; analogamente alla GETATTAR, consentita accedendo in revisione anagrafica articoli da emissione revisione documento, aprendo la revisione impegni dalla bolla di lavoro ed accedendo all'impegno è consentita la GETATTIL dell'impegno in revisione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTIND

GETATTIND

Istruzione che legge i dati anagrafici degli Indirizzi di Spedizione presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _DA caricate dal Gestionale all'atto dell'utilizzo dell'istruzione; questa azzera le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTMM

GETATTMM

Istruzione che legge i dati del Documento di Magazzino presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _MM caricate dal Gestionale all'atto dell'utilizzo dell'istruzione; questa azzerava le variabili preesistenti ed inizializza la struttura con i valori esistenti. L'istruzione valorizza _NMM e _NTMM

Nota: l'evento **START** si attiva dopo l'uscita dal primo campo che divide l'emissione dalla revisione, mentre in emissione differita documenti si attiva con trasformazione del documento e non con la ristampa! ovvero con "Emissione di prova = S" (vedi anche evento ON_PUTMM).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTPART

GETATTPART

Istruzione che legge i dati della Particolarità Prezzo, Sconto, Provvigione, Articolo presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _PART caricate dal Gestionale al momento dell'utilizzo dell'istruzione; questa azzerava le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTPC

GETATTPC

Istruzione che legge i dati del Conto di Anagrafica Piano dei Conti o Rubrica Clienti Fornitori presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _PC caricate dal Gestionale al momento dell'utilizzo dell'istruzione; questa azzerava le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTPN

GETATTPN

Istruzione che legge i dati della Primanota presente a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _PN caricate dal Gestionale al momento dell'utilizzo dell'istruzione; questa azzerava le variabili preesistenti ed inizializza la struttura con i valori esistenti (anche in caso di registrazione da magazzino). L'istruzione valorizza le variabili _NPN, _PNPRN.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTPU

GETATTPU

Istruzione che legge i dati della Anagrafica Ubicazioni presenti a video nel momento in cui l'istruzione viene eseguita. Ritorna il valore delle variabili di struttura _PU caricate dal Gestionale al momento dell'utilizzo dell'istruzione; questa azzerava le variabili preesistenti ed inizializza la struttura con i valori esistenti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTRIGMM

GETATTRIGMM <numriga>,<modo>,<numdecart>

Parametri	
numriga	Numero di riga valida del documento di magazzino;
modo	Modo di calcolo; se = 1 calcola per progressivi; se = 2 calcola per totale documento; la differenza di metodo è relazionata alle righe in um2 (unità di misura secondaria), ovvero, nel modo 1 si ottengono i valori (ad es. quantità e importo) di riga in unità di misura primaria e in valuta di gestione azienda, mentre invece, nel modo 2 gli stessi valori sono espressi in unità di misura secondaria (se gestita) ed in valuta del documento (se diversa da quella di valuta azienda).
numdecart	Numero di decimali articolo (obbligatorio se <modo>=1);

Istruzione utilizzabile solo in collage, equivalente della RIGMM di sprix, che calcola i totali di riga inserendo i risultati nelle variabili _MMR*. Non disponibile nel menù produzione e nelle stampe.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETATTTOTMM

GETATTTOTMM

Istruzione utilizzabile solo in collage, equivalente della TOTMM di sprix, che calcola i totali documento inserendo i risultati nelle variabili _MMT*. Non disponibile nel menù produzione e nelle stampe.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETAUTORIZUTE

GETAUTORIZUTE <[utente\$],[idutente]>,<filtro\$>,[filtro\$]

Parametri	
utente\$	Nome dell'utente di cui si vogliono conoscere le limitazioni (alternativo al successivo); parametro ignorato in Sprix Mobile poichè sono sempre restituiti solo i filtri relativi all'utente loggato in quel momento e i dati di filtro sono aggiornati sempre all'ultima sincronizzazione con il Server.
idutente	Id dell'utente di cui si vogliono conoscere le limitazioni (alternativo al precedente); parametro ignorato in Sprix Mobile poichè sono sempre restituiti solo i filtri relativi all'utente loggato in quel momento e i dati di filtro sono aggiornati sempre all'ultima sincronizzazione con il Server.
filtro\$	<p>Filtro di cui si vuole avere l'elenco dei valori accettati:</p> <p>AUTORIZZAZIONI CLIENTI/FORNITORI:</p> <ul style="list-style-type: none"> - "_PCAGE" Filtro sul codice agente. Tale filtro consente l'accesso ai soli clienti che hanno tale codice impostato nel campo agente, e, se richiesto, di quelli che non hanno il codice agente impostato. Disabilita l'accesso a tutti i fornitori tranne l'agente stesso. - "_PCCST" Filtro sulla categoria statistica cliente. - "_PCONA" Filtro sulla zona cliente. - "_PCPAE" Filtro sul paese cliente. - "_PCCOD" Filtro su conti cliente o fornitore ammessi; <p>AUTORIZZAZIONI ARTICOLI:</p> <ul style="list-style-type: none"> - "_ARTIP" Filtro sulla tipologia di articolo. - "_ARSTA" Filtro sulla categoria statistica articolo. - "_ARGRPMER" Filtro sul gruppo merceologico articolo. - "_ARNATURA" Filtro sulla natura articolo. <p>AUTORIZZAZIONE DOCUMENTI:</p> <ul style="list-style-type: none"> - "_MMAGE" Filtro sull'agente impostato nei documenti - "_MMSIG" Filtro sulla serie documenti di magazzino - "_MMMAG" Filtro sul numero di magazzino - "_MMCLI" Filtro sul conto
filtro\$	<p>Parametro facoltativo considerato solo se richiede le autorizzazioni "_PCCST", "_PCONA", "_PCPAE", "_PCCOD"; valori consentiti:</p> <ul style="list-style-type: none"> • "TIPOCONTO=C" per leggere le limitazioni sui conti cliente • "TIPOCONTO=F" per leggere le limitazioni sui conti Fornitore <p>Se non si passa nulla, restituisce quelle dei clienti (per compatibilità con le versioni precedenti, in cui le limitazioni sui fornitori non erano presenti).</p> <p>Specificando "_PCCOD" con "TIPOCONTO=C" si conseguono i seguenti possibili valori:</p> <ul style="list-style-type: none"> • un solo valore pari a -1 se è stato inibito l'accesso a TUTTI i clienti ("Accesso codici clienti" uguale a NO); • nessun valore, cioè, sono tutti ammessi purché valgano gli altri filtri eventualmente impostati; <p>Specificando "_PCCOD" con "TIPOCONTO=F" si conseguono i seguenti possibili valori:</p> <ul style="list-style-type: none"> • un solo valore pari a -1 se è stato inibito l'accesso a TUTTI i fornitori ("Accesso codici fornitore" uguale a NO) • lo stesso elenco fornito da "_PCAGE" se viene ammesso l'accesso solo agli agenti specificati; • nessun valore, che vuol dire che sono tutti ammessi purché valgano gli altri filtri eventualmente impostati;

Istruzione che legge le Autorizzazioni Utente sui dati impostate sull'azienda corrente e per un determinato utente, relativi alle variabili di struttura _AU. Nella maggior parte delle variabili l'istruzione ritorna valori multipli (elenchi di dieci valori) per tutti i filtri che li contengono. L'istruzione valorizza le variabili _ERRAU e _ERRAU\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETBL

GETBL <numeroBL>,<numeroSottoBL>

Parametri	
numeroBL	Numero Bolla di Lavorazione;
numeroSottoBL	Numero Sotto Bolla di Lavorazione;

Istruzione che legge i dati della Bolla di Lavorazione relativi alle variabili `_OB` (testata) e `_OL` (righe) in base al numero della bolla `<numeroBL>` e sottobolla `<numeroSottoBL>` di lavorazione (per le sottoBL dei lavoranti aggiungere 10000, esempio la sottobolla 0002 → 10002). La lettura valorizza la variabile `_NBL`, che ritorna il numero delle righe in bolla lavorazione. L'istruzione valorizza le variabili `_ERRBL` e `_ERRBL$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCDG

GETCDG `<TipoDoc$>`,`<KeyDoc$>`,`<Anno>`,`<NumRigaDoc>`,`<NumProgAna>`

Parametri	
TipoDoc\$	Tipo movimento: (R=Primanota; M=Magazzino; C=Ord.Cliente; F=Ord.Fornitore; P=Preventivi; A=Marici; E=Extra);
KeyDoc\$	La Chiave del movimento <code><KeyDoc\$></code> è formata da campi obbligatori di ricerca in base a <code><TipoDoc\$></code> (separati da ;) <ul style="list-style-type: none"> R – “PNDRE=x;PNND0=x;” (PNND0 = num. Progressivo registrazione primanota); M – “MMSIG=x;MMNUM=x;MMSAZ=x;MMSER=x;MMCLI=x;NTMM=x;” (M è in relazione all'anno di ingresso; MMCLI è facoltativo per i documenti non intestati; se NTMM – Num. Riferimento testata documento non è definito è impostato = 0); C/F/P/A – “MMSIG=x;MMNUM=x;MMSAZ=X;MMSER=x;”; E – “PRGEXTRA=x;DTCOMP=x;AREA=x;COMM=x;CODANL=x;TIPMOV=x;”: PRGEXTRA non è obbligatorio e rappresenta un elemento progressivo per ottimizzare la chiave di ricerca in caso di movimenti identici ed è consultabile con il dizionario bgepr().
Anno	Anno del movimento: (contabile 1960 -2048 se tipo C/F, fisso 2048 se tipo E/P, fisso 1 se tipo A);
NumRigaDoc	Numero della riga del documento <code><NumRigaDoc></code> a cui sono associate le registrazioni di analitica (se 0=tutte le righe);
NumProgAna	Numero progressivo di riga <code><NumProgAna></code> relativo all'analitico (se 0=tutte le righe; in magazzino è sempre 0 dato che per ciascuna riga di movimento è consentita una sola riga di analitica mentre in primanota ne sono consentite di più);

Istruzione che legge i dati dei movimenti di Contabilità Analitica del Controllo di Gestione relativi alle variabili `_CDG`, in base al Tipo movimento `<TipoDoc$>` nell'anno del movimento `<Anno>` al numero della riga del documento `<NumRigaDoc>` a cui sono associate le registrazioni di analitica e al numero progressivo di riga `<NumProgAna>` relativo all'analitico. La lettura rilascia la `_CDGnr` per conoscere il numero di elementi dell'array dinamico (numero righe) della struttura `_CDG`. L'istruzione valorizza `_ERRCDG` e `_ERRCDG$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCFR

GETCFR `<cod$>`,`<tipo>`,`[arch]`

Parametri	
cod\$	Codice Cliente / Fornitore / Contatto;
tipo	Referente: 0 = tutti i referenti; 1 a 30 = un referente specifico
arch	“TP=0” esegue la lettura sull'archivio dei contatti in base al codice impostato

Istruzione che legge i dati dei Referenti presenti o nell'anagrafica Clienti Fornitori (tasto funzione F4) o Contatti. Vengono restituiti i valori relativi nelle variabili di struttura `_CFR`. Escluso il valore “arch” dal comando viene automaticamente eseguita la lettura sull'archivio dei clienti fornitori. La lettura valorizza `_CFRNELE` (numero di referenti presenti); Il collegamento fra referente registrato e l'array è stabilito da `_CFRPRG()`. L'istruzione valorizza le variabili `_CFRERR` e `_CFRERR$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCM

GETCM `<ute$>`,`<data$>`,`[id$]`,`[adata$]`

Parametri	
ute\$	Id Nome Utente; obbligatorio
data\$	Data di ricerca nel calendario, formato “AAAAMMGG”; obbligatoria
id\$	Id Agenda (numero record creazione);
adata\$	Data di fine ricerca nel calendario; formato “AAAAMMGG”; se inserita determina l'elaborazione di tutti i calendari dell'utente compresi fra le date impostate.

Istruzione che legge dati del Calendario dell'Agenda Attività Utente. Valorizza le variabili di struttura `_CM` e le variabili `_NCM`, `_ERRCM` e `_ERRCM$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCODART

GETCODART\$ `<tipo$>`,`<codice$>`,`[parametri$]`

Parametri	
tipo\$	<p>Tipo di dato in input da trovare. Accetta i seguenti valori:</p> <ul style="list-style-type: none"> • "P" = Codice primario; • "S" = Codice alternativo; • "D" = Descrizione articolo; • "A" = Alias; • "B" = Abbinato; • "F" = Codice articolo del fornitore; • "L" = Chiave utente lotto (dati correnti); • "O" = Chiave utente lotto (dati storico) <p>La ricerca per 'P' è sempre attiva La ricerca per 'S'/D' è attiva solo se il parametro "Codice alternativo" è a SI La ricerca per 'A'/B' è attiva solo se il parametro "Ricerca articoli per codice Alias" è a SI La ricerca per 'F' è sempre attiva La ricerca per 'L'/O' è attiva solo se il parametro "Gestione lotti e rintracciabilità" è diverso da NO</p>
codice\$	<p>Valore in Input da ricercare.</p> <p>Nota bene: eccetto il caso <tipo\$>="D", è richiesto un codice esatto e PUNTUALE (non porzioni di codice, poichè non sarà trovato) e preferibilmente delle dimensioni giuste (es. ripulendo spazi in eccesso in base a LEN()).</p> <p>Per <tipo\$> = 'L' deve iniziare con il tipo lotto, se ha " " (due spazi) vuol dire che la ricerca su tutti</p>
parametri\$	<p>Parametro facoltativo per ulteriori controlli nella ricerca</p> <ul style="list-style-type: none"> • Per <tipo\$>="F" è consentito il parametro "FORN=GMM.CCCCC", cioè limita la ricerca del codice articolo dei fornitori solo al codice fornitore indicato in GMM.CCCCC. Es: GETCODART "F",COD\$,FORN=501.00123" • Per <tipo\$>="D" è consentito il parametro "CONT=". Può assumere i seguenti valori: <ul style="list-style-type: none"> ○ "CONT=*" Filtro "CONTIENE" <u>senza</u> controllo maiuscolo e minuscolo; ○ "CONT=\$" Filtro "CONTIENE" <u>con</u> controllo maiuscolo e minuscolo; ○ "CONT==" Filtro "UGUALE" <u>con</u> controllo maiuscolo e minuscolo; ○ "CONT=" Filtro "UGUALE" <u>senza</u> controllo maiuscolo e minuscolo; ○ "CONT." Filtro "INIZIA PER" <u>con</u> controllo maiuscolo e minuscolo; ○ "CONT," Filtro "INIZIA PER" <u>senza</u> controllo maiuscolo e minuscolo; <p>Esempio GETCODART "D",COD\$, "CONT="*</p>

Istruzione (dalla v711) che consente di trovare i codici di Anagrafica Articoli accomunati dal valore del parametro in Input <codice\$> relativamente al <tipo\$> di valore cercato.

L'istruzione valorizza le seguenti variabili _CODARTNUM (Numero di codici trovati), _CODART\$() (Array dei codici), _ERRCODART\$ (Eventuale messaggio d'errore).

Esempio, per trovare tutti i Codici Articoli che nella loro anagrafica hanno una determinata descrizione <codice\$> e che, quindi, sono relativi al <tipo\$> ="D". Altro esempio [es.](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCODFIN

GETCODFIN\$(<numfin>)

Parametri	
numfin	Numero relativo alla finestra in uso. Vedere anche la variabile _NFIN.

Funzione che restituisce l'identificativo della finestra in base al numero di finestra in uso. Esempio: CODFIN\$=GETCODFIN\$(_NFIN)
 Vedere anche CERCACODFIN() e _NFIN.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCOM

GETCOM <n>

Parametri	
n	Quantità di dati

Istruzione che legge dalla porta seriale aperta la quantità dei dati <n> (da 1 a 255, con default 1 se < 1 o 255 se > 255) memorizzando i byte nella variabile `_BUFCOM$` (buffer di memorizzazione dei byte letti da GETCOM). Se mancano caratteri da leggere, GETCOM attende il <timeout> di OPENCOM. La GETCOM valorizza la variabile `_ERRCOM$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCOMM

GETCOMM <commessa>

Parametri	
commessa	Codice commessa;

Istruzione che legge i dati di Anagrafica Commessa relativi alle variabili struttura archivi `_CO`, in base al codice <commessa>. L'istruzione valorizza le variabili `_ERRCO` e `_ERRCO$`. La possibilità di cancellazione è attraverso il modulo Servizi di Controllo di Gestione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCON

GETCON <codice\$>

Parametri	
codice\$	Codice contatto

Istruzione che legge i dati di Anagrafica Contatti relativi alle variabili `_DC` in base al codice contatto <codice\$>. L'istruzione valorizza le variabili `_ERRDC` e `_ERRDC$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETCONDDOC

GETCONDDOC <tipo>, <codiceconto\$>, <codart\$>, <datadoc\$>, <sigladoc\$>, <qta>, <coeff>, <valuta>, <listino>, <prezzo>, <sconto\$>, <catsconto>, <catprovv>, <extra\$>

Parametri	
tipo	Tipo calcolo, valori ammessi: 1 - calcola prezzo, sconto e provvigione; 2 - calcola sconto e provvigione; 3 - calcola la provvigione.
codiceconto\$	Codice conto.
codart\$	Codice articolo.
datadoc\$	Data per il calcolo.
sigladoc\$	Sigla documento (le particolarità non sono ricercate per documenti "SL", "CL", e "DL").
qta	Valore per lo scaglione degli sconti quantità e delle particolarità
coeff	Coefficiente: se zero lavora in UM primaria dell'articolo, se valorizzata in secondaria.
valuta	Valuta di elaborazione, se zero si assume la valuta dell'azienda (in sprix mobile ogni valore inserito equivale a 0 in quanto non vengono gestite le valute per i calcoli, pur essendo presente l'archivio).
listino	Listino, se 0 si utilizza il listino del cliente.
prezzo	Prezzo: se passato e si imposta tipo a 2 o 3 allora viene utilizzato come base per calcolo sconti e provvigioni.
sconto\$	Sconto: se passato e si imposta tipo 3 allora viene utilizzato come base per il calcolo delle provvigioni.
catsconto	Categoria sconto: se passata viene utilizzata questa al posto di quella del cliente.
catprovv	Categoria provvigioni: se passata viene utilizzata questa al posto di quella del cliente.
extra\$	Extra: se passata viene utilizzata per il calcolo.

Istruzione che ritorna le condizioni documento in base ai parametri specificati nelle variabili `_PZD`. L'istruzione valorizza le variabili `_ERRPZD` e `_ERRPZD$`. L'istruzione richiede MDS. Nota bene relativa sprix mobile: per poter calcolare correttamente le condizioni è indispensabile che gli archivi mobile contengano tutti i dati necessari al calcolo; se, ad esempio, l'azienda utilizza gli extra di listino e i campi relativi non sono stati inclusi negli archivi mobile, questi non potranno essere utilizzati nel calcolo che di conseguenza risulterà non corretto.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETDB

GETDB <codart\$>,<numfase>

Parametri	
codart\$	Codice articolo con Distinta Base Primaria;
numfase	Numero di fase della Distinta Base Primaria;

Istruzione che legge i dati della Distinta Base Primaria relativi alle variabili di struttura archivi _TP in base al numero di fase <numfase> della DBP (1-99) e del codice articolo <codart\$> con DBP. Le operazioni sulla DBP valorizzano _NCOMP (numero dei componenti di una fase: max 5000 per fase). La lettura valorizza il numero di fasi _NFAZI della DBP. Nella lettura di una grande DBP elaborare fase per fase (una alla volta, senza definire 49500 potenziali variabili). L'istruzione valorizza le variabili _ERRDB e _ERRDB\$.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETDVDOC

GETDVDOC <numdoc>,<numrev>,[DIRLOC];[GETPH]

Parametri	
numdoc	Numero Documento di Docuvision;
numrev	Numero Revisione documento (in lettura se -1=tutte le revisioni, 0=ultima revisione, >0=revisione specificata; in cancellazione se 0=cancella l'intero documento). Le operazioni verificano i Parametri Generali di Docuvision poiché la gestione delle pagine è alternativa a quella delle revisioni di documenti già con la gestione pagine attiva è gestita solo l'ultima <numrev> contenuta in <numdoc> e quindi viene ignorata sia in lettura (i dati sono riferiti all'ultima revisione) che in cancellazione (quindi va valorizzata = 0).
DIRLOC GETPH	Directory locale facoltativa dove potere copiare i documenti allegati. La sintassi specifica per il parametro è "DIRLOC=". Per esempio "DIRLOC=<C:\TEMP\>". Notare il simbolo "<" che indica la directory locale e la barra finale della directory scelta. Il parametro DIRLOC può essere usato anche per specificare percorsi locali al server tramite l'utilizzo dei caratteri speciali identificanti le cartelle del server consentite (ad esempio DIRLOC=).

Nota: **[DIRLOC];[GETPH]** consente la copia immediata del file allegato senza doverlo visualizzare da Docuvision.

Nella Fatturazione Elettronica la copia immediata del file Xml è consentita solo se il **Servizio** "Fattura XML - Sistema ricevente - Conservazione digitale" (vedere [Shift+F11] in Anagrafica Azienda) è configurato per essere eseguito in **autonomia**, altrimenti, occorrerebbe eseguire una preventiva visualizzazione del documento allegato da Docuvision in quanto, se il servizio di invio della fattura elettronica allo SDI è un servizio Passepartout (All Inclusive o Firma + Invio Ricezione), il file ".xml" corrispondente alla fattura inviata (es. IT04345180378_00001.xml) viene **cancellato** dal disco locale (e quindi non consentirebbe la copia [DIRLOC]) fino a che non si visualizza la fattura stessa dalla gestione documento e **dopo** che il servizio PassHub abbia firmato il file (es. IT04345180378_00001.xml.p7m). La visualizzazione del documento, infatti, **esegue la richiesta a PassHub** il quale ne restituisce una copia che viene resa disponibile **anche** per l'operazione di copia [DIRLOC].

Per eseguire in Sprix questa richiesta a PassHub al fine di scaricare il file Xml nella [DIRLOC], se disponibile e/o consentito da PassHub, dalla versione v711 è disponibile il parametro **[GETPH=SI]** ("GET file Xml da PassHub") da associare ad un percorso del parametro [DIRLOC=] **locale e accessibile** (es. "<C:\Temp"), in quanto se fosse nullo [DIRLOC=""] il download avviene nella "...dati\temp\fattxml\NNN\" che è ad uso esclusivo del Gestionale (NNN = numero di terminale che ha eseguito la GETDVDOC) e quindi, alla fine del processo Sprix (mx45.exe) verrebbe **ripulita** (è una cartella "temp") e senza consentire alcuna COPYFILE per salvare il file in locale, in quanto il suddetto percorso riservato non è accessibile. Il file che viene scaricato è in formato ".xml" oppure ".xml.p7m" (se la fattura transitata dallo SDI è quella firmata). In questo secondo caso è possibile decodificare in semplice Xml attraverso la funzione **DECODP7M()**.

Per conoscere altre informazioni relative alla processo di fatturazione elettronica, è possibile consultare la base dati contenuta nel percorso "...dati\data\sig\sig_dv.mdb", per esempio nella tabella "DV_VERSIONE", nella colonna CDS_NOTA2 (il nome del file, contenuta anche nella variabile _DVVNOTA2\$(I)) e la colonna CDS_TIPO_RICEVUTA (lo stato, contenuto anche nella variabile _DVVTPRIC\$(I)) relativa alla colonna NKY_NUM_RELEASE (quella con il valore più elevato). Impostando getdvdoc con -1 come numero di versione, le legge tutte, se impostata con 0 legge l'ultima, quella dove leggere lo stato della fattura in base alla ricevuta o, in caso di ricevuta vuota, in base al fatto che abbia o meno a data di trasmissione.

Nota Bene: se il documento è già presente e disponibile in Docuvision, la GETPH non preleva da PassHub, bensì preleva il documento disponibile in Docuvision; la ripetizione dell'operazione GETPH va in sovrascrittura in base al nome del file e senza alcun avvertimento. In ogni caso si consiglia di verificare i file attraverso LEGGIDIR e/o ESISFILE. L'operazione non è consentita per documenti del ciclo passivo (es. fatture fornitore), per interrogare PassHub occorrono specifiche funzioni (non disponibili); in alternativa usare i comandi Webservice per interrogare i servizi di interscambio (SDI) .

Esempio:
AZZVARSYS 20
FOR ID=1 TO 16
GETDVDOC ID,-1, "DIRLOC=<C:\tmpdv\;GETPH=SI" : IF _ERRDV<>0 THEN VIMSG 4, _ERRDV\$
VIMSG 1, "Num:"+STR\$(ID,0,0)+" Clas:"+STR\$(_DVCCLASS,0,0)+" Tit:"+_DVDTIT\$+ " Prot:"+_DVCPROTO\$+
Cont:"+_DVCNT\$+ " DatDoc:"+_DVDDATA\$+"DatPro:"+_DVDDATPRO\$
NEXT
END

Nota su ALLEGATI

Sprix gestisce gli allegati relativi al Documento di Docuvision (fig. 1), non gestisce gli allegati relativi alla Fattura PA / B2B (fig. 2); il modo per intervenire sul file XML (della fattura elettronica) è consentito all'evento di stampa ON_STP_XML, ma non sono disponibili apposite variabili per codificare l'allegato dentro il file XML poiché occorre un'apposita codifica separata (fig. 2). Si ricorda che altri generici file XML sono gestiti in Sprix tramite apposite istruzioni XML. Per ottenere l'handle del documento in considerazione al fine di eseguirne le opportune modifiche con le istruzioni generali XML, avvalersi della variabile `_FTE_DOCXML` che consente l'accesso da collage al file `_XML` della FTE.

Fig. 1 (Allegati DOCUVISION)

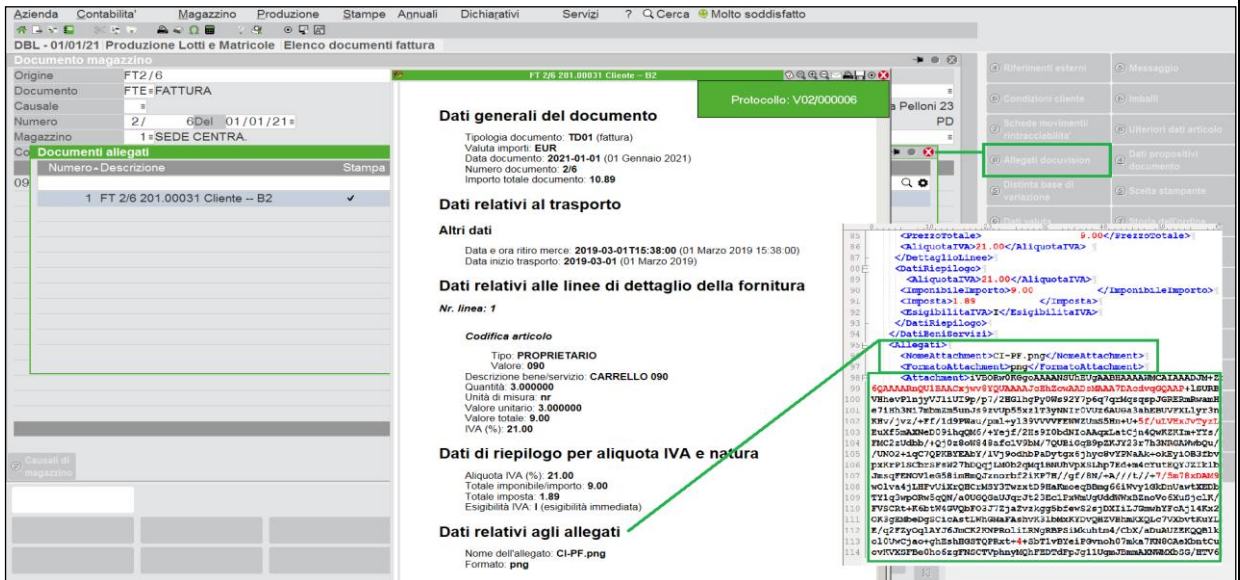
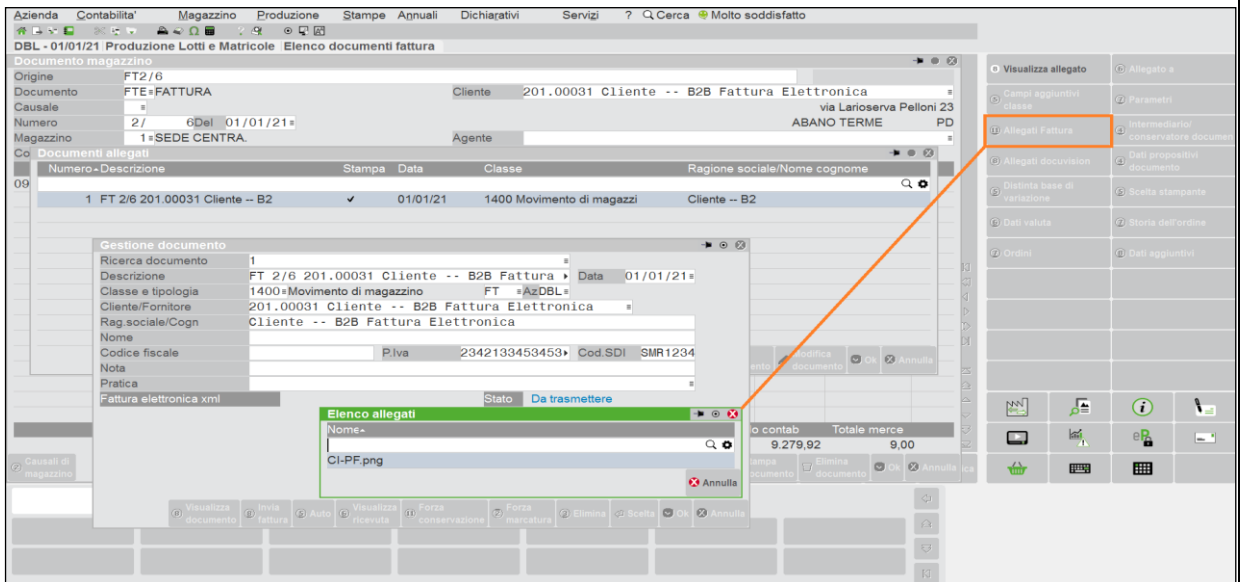


Fig. 2 (Allegati FATTURA)



GETPH Parametro facoltativo [GETPH=SI] che consente di interrogare PassHub per eseguire il download del file Xml (se disponibile e/o consentito).

Istruzione che legge i dati del Documento Docuvision relativi alle variabili struttura archivi `_DV`, in base al numero Documento di Docuvision `<numdoc>` e numero revisione `<numrev>`. La lettura valorizza `_DVDNUMVER` (in scrittura è calcolata) e `_DVDPAGXVER()`, che in Sprix ritornano il numero globale di revisione letta (=1 con gestione pagine attiva) ed il numero di pagine per revisione (=1 con gestione pagine non attiva) che permette di scorrere i vettori delle variabili contenenti i dati delle revisioni (tutti uguali eccetto il contatore di pagina) e il vettore dei nomi file `_DVVFILEGET`. L'istruzione valorizza le variabili `_ERRDV` e `_ERRDV$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETESPOS

GETESPOS <cod\$>,<data\$>

Parametri	
cod\$	Codice conto (formato "GMM.CCCCC");
data\$	Alla data (formato "AAAAMMGG");

Istruzione che legge l'esposizione del conto <cod\$> alla data <data\$>. L'istruzione lavora sull'anno aperto (modificabile con APRIAZIE) e valorizza la variabile _ESPOSIZER\$. L'istruzione rilascia informazioni nelle seguenti variabili:

_SALDOCLIFOR = saldo contabile del cliente/fornitore

_FIDOCLIFOR = valore fido (anche calcolato su fatturato anno precedente)

_VALORDBOLLE = valore ordini/bolle

_VALORDINI = valore ordini netto

_VALORDINII = valore ordini con IVA

_VALBOLLE = valore bolle netto

_VALBOLLEI = valore bolle con IVA

_ESPOSIZIONE = esposizione bancaria

_DATAESPOS\$ = data a cui e calcolata l'esposizione (DATA parametro)

_VALRISCHIO = valore di rischio, (formula composta come tabella progressivi = _SALDOCLIFOR + _VALORDBOLLE + _ESPOSIZIONE)

_FUORIFIDO = differenza fra FIDOCLIFOR e VALRISCHIO (solo se rischio>fido)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETFINTIT\$

GETFINTIT\$(<numfin>)

Parametri	
numfin	numero relativo alla finestra in uso. Vedere anche la variabile _NFIN.

Funzione che restituisce il titolo della finestra in base al numero di finestra in uso. Vedere anche CERCACODFIN() e _NFIN.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETGM

GETGM <gruppo\$>

Parametri	
gruppo\$	Codice del gruppo merceologico da leggere;

Istruzione che legge i dati anagrafici del Gruppo Merceologico relativo alle variabili struttura archivi _GM in base al codice del <gruppo\$>. L'istruzione valorizza le variabili _ERRGM e _ERRGM\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETGPSDATA

GETGPSDATA <AnnoMese>,[giorno],[azienda\$],[idutente\$],[idapp\$]

Parametri	
<AnnoMese>	Anno e mese
[giorno]	Giorno
[azienda\$]	Sigla azienda
[idutente\$]	ID Utente
[idapp\$]	ID App

Istruzione Collage Server Remoto per il recupero dei dati contenuti nelle variabili di struttura _GPS? (vedere WCALL "GETGPSDATA",0) restituiti all'evento ON_GETGPSDATA: Esempio GETGPSDATA "202001","", "" (nessun filtro azienda e utente). Per la visualizzazione della posizione, vedere l'istruzione GEOMAPPA.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETIL

GETIL <ndoc>,<nsot>,<nropf>,<fase>,<mag>,<cod>

Parametri	
ndoc	Numero della bolla di lavorazione, corrisponde alla variabile _ILNDO
nsot	Numero della sottobolla di lavorazione, corrisponde alla variabile _ILNSD
nropf	Numero riferimento prodotto finito, corrisponde alla variabile _ILNRB
fase	Numero di fase del componente, corrisponde alla variabile _ILFAS; (se 0 = tutte le fasi);
mag	Magazzino materie prime, corrisponde alla variabile _ILMAG; (se 0 = tutti i magazzini);
cod	Codice del componente, corrisponde alla variabile _ILAMP\$; (se "" = tutte le MP);

Istruzione che legge i dati degli Impegni di Lavorazione relativi alle variabili di struttura archivi _IL in base al numero Bolla di Lavorazione <ndoc>, il numero della sottobolla <nsot>; la specificazione del codice della materia prima <cod> implica l'indicazione del numero di magazzino <mag> che, se specificato a sua volta implica il numero fase <fase> che, se specificata, implica il numero di riferimento al prodotto finito <nropf> (se 0=tutte, se -1=RES_SL,se -2=RES_DL; attenzione, non essendo gestiti i numeri negativi, sostituire per RES_SL il valore 65534 al posto di -1 e per RES_DL 65533 al posto di -2.). La lettura valorizza la _NIL che ritorna il numero di righe di componenti Materia Prima. L'istruzione valorizza le variabili _ERRIL e _ERRIL\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETIND

GETIND <codice\$>,[<codice_conto\$>]

Parametri	
codice\$	Codice indirizzo di spedizione;
codice_conto\$	Codice conto associato all'indirizzo ("" se non utilizzato)

Istruzione che legge i dati anagrafici degli Indirizzi di Spedizione relativi alle variabili di struttura archivi _DA in base al codice indirizzo di spedizione <codice\$> e all'eventuale [<codice_conto\$>] cui è associato l'indirizzo. L'istruzione valorizza le variabili _ERRDA e _ERRDA\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETINTERNET

GETINTERNET <url\$>,<file\$>

Parametri	
url\$	Indirizzo del Sito Web Internet;
file\$	Percorso e nome del file;

Istruzione che nei sistemi Windows/Linux copia un file con il nome <file\$> dall'indirizzo <url\$> di un sito Internet nel percorso indicato nello stesso nome del <file\$>. L'istruzione è funzionale ai file ".HTML"; se non è utilizzabile per altre tipologie di file (es. .txt, .exe, .xml, .bmp, .csv, .ecc..) verificare se Web Service consente di utilizzare l'istruzione CALLWEBSVC.

I protocolli 'URL' consentiti sono Http e Https; la porte default sono '80' per Http e '443' (tramite TLS) per Https; se la porta del Server WEB è diversa occorre inserire il prefisso "[porta:nnn]" in <url\$>, dove 'nnn' è il numero della nuova porta. Se la porta è diversa dalla 80 non è consentita la forma [indirizzo:porta]. L'istruzione valorizza la variabile _ERRGETINT\$ con la descrizione dell'errore di ritorno della comunicazione ed annullamenti degli avvisi TLS.

Esempi:

GETINTERNET "https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=CIAO COME STAI", "<C:\TEMP\QRCODE.JPG"

GETINTERNET "https://www.rfc-editor.org/rfc/rfc5321.txt", "]rfc5321.txt" richiede la risorsa sulla porta 443 tramite TLS.

GETINTERNET "http://www.rfc-editor.org/rfc/rfc5321.txt", "]rfc5321.txt" richiede la risorsa sulla porta 80 in chiaro.

GETINTERNET "[porta:8080]http://portquiz.net", "]portquiz.net" richiede la risorsa sulla porta 8080 in chiaro.

[\(altri esempi\)](#)

Per i protocolli https, imap, smtp e altre comunicazioni sopra al TLS, dalla v810 la GETINTERNET cerca di impostare automaticamente il Server di destinazione come SNI, in modo da coprire la maggior parte dei casi. L'istruzione non consentiva, infatti, l'accesso a Server Web configurato con SNI (Server Name Indicator), ovvero l'elemento indispensabile per distinguere sulla medesima macchina (stesso indirizzo) differenti Siti Web. Occorreva che vi fosse univocità oppure la necessità di eseguire il download sul Client in locale "<C:\..." (il processo è diverso e minori sono i controlli), altrimenti la connessione viene interrotta forzatamente dall'host remoto. Esempio: "<C:\portquiz.net").

La modalità https negozia con il Server un livello di protocollo TLS v1.x analogamente all'invio mail SMTP. Di default, come file d'elenco CA pubbliche (utilizzato per convalidare l'identità del Server tramite certificato standard X509) si utilizza il binloc\cacertsptms.pem (lo stesso utilizzato per l'invio SMTP). Sempre di default, sono attivi gli avvisi di sicurezza TLS che consentono all'utente di rilevare interattivamente eventuali problemi nei controlli minimi che la negoziazione richiede come standard di sicurezza. In tal caso, come per l'invio SMTP, compaiono finestre di conferma che consentono di procedere oppure annullare l'operazione.

Riepilogando, i casi più frequenti di possibili avvisi sono:

- nome Server SMTP non presente nel soggetto del certificato TLS presentato dal Server stesso. Un caso tipico, ovvero quando

nell'URL si configura un indirizzo IP, in quanto nessuna autorità emittente di certificati è in grado di generare per IP – lo standard richiede un nome pienamente qualificato dal DNS;

- certificato presentato dal Server ha una data di inizio validità successiva alla data odierna;
- certificato presentato dal Server ha una data di scadenza precedente alla data odierna;
- catena di certificazione non risolta, ad esempio, il file con l'elenco delle CA pubbliche (default è "binloc\cacrtsmtps.pem") non contiene la chiave pubblica dell'autorità che ha emesso il certificato presentato dal Server e/o nessuna componente pubblica di tutta la catena fino alla radice. Un altro caso che provoca un avviso è il cosiddetto certificato auto firmato (self-signed). Alcuni antivirus hanno un dispositivo di sicurezza che s'inserisce proprio con un certificato auto firmato per controllare il traffico in uscita, generando un avviso in Passepartout.

Risulta possibile personalizzare il file d'elenco CA pubbliche, impostando nel file "dati\conf\pxconf.ini", sezione [Impostazioni], parametro: `HttpFileCAPubbliche=<percorso>`

Requisiti di personalizzazione di tale file:

- non può avere contenuto in output binario, anche se l'estensione non necessariamente deve essere ".pem";
- non può risiedere sul Client;
- deve esistere sul Server e risultare accessibile;

Non rispettando uno di tali requisiti, si genera un avviso di sicurezza.

A proposito degli avvisi stessi, risulta possibile disattivarli impostando nel file "dati\conf\pxconf.ini", sezione [Impostazioni], parametro: `AvvisiTLSGetInternet=0`

Fin quanto possibile occorre cercare la soluzione alla falla verificata che genera l'avviso piuttosto che disattivarlo, in quanto l'utilizzatore potrebbe perdere la percezione di eventuali abbassamenti di sicurezza.

NB: i parametri suddetti sono globali, valgono per tutte le chiamate con istruzione GETINTERNET di tutti gli Sprix nell'installazione. Se una GETINTERNET opera in chiaro, tali parametri non sono considerati.

Relativamente ai timeout, sono personalizzabili con parametri nella sezione [Impostazioni] del file "dati\conf\pxconf.ini".

`TimeoutConnessioneGetInternet = <secondi>` Timeout connessione al Server URL di download, corrispondente al tempo limite di eventuale negoziazione TLS (default 30).

`TimeoutLetturaGetInternet = <secondi>` Timeout lettura dal Server URL di download, parametro che vale sia per http sia per https (default 900, cioè 15 minuti).

`TimeoutScritturaGetInternet = <secondi>` Timeout scrittura al Server URL di download, parametro che vale sia per http sia per https (default 900, cioè 15 minuti).

Quindici minuti sono un tempo comunque sufficientemente lungo, considerando che l'attesa del programma è sempre "sincrona". I suddetti parametri agiscono sempre a livello globale, quindi valgono sia per tutti gli SPRIX che anche per tutto ciò che sfrutta la medesima funzione di libreria GetInternet.

Controllo di accesso Utente

Analogamente ad altre istruzioni che operano su file system (open, kill, copile, ecc.), in caso di scrittura su directory non autorizzate o di fallito accesso al file per restrizioni di sistema operativo (es. accesso negato), la procedura rilascia un errore non bloccante verificabile con variabile `_ERRGETINT$`.

In caso di prodotto Passepartout che non ha restrizione dati utente è consentito gestire sul Server gli accessi utente attraverso la variabile `_ACCFILEUTE`; se questa è attiva (=1) l'utente può eseguire il download scrivendo file solamente nelle directory consentite o non vietate. L'eventuale errore di violazione restrizione (cancellazione se il file esiste già, creazione, apertura) è ritornato in variabile `_ERRGETINT$`. Infine, nei prodotti senza restrizione dati utente, la procedura impedisce di eseguire il download in percorsi Server diversi da quelli ammessi, ovvero <DATI GENERALI> e <DATI AZIENDA>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETKEY

GETKEY <numfile>,<numkey>,<key\$>[,key1\$]

Parametri	
numfile	Numero File Indici
numkey	Numero chiave (da 1 a 4, cioè una fra quella primaria e le secondarie);
key\$	Chiave stringa
key1\$	Chiave principale (facoltativa), è utile in caso di ricerca per chiave secondaria di più record con la stessa chiave secondaria (individua il record cercato anziché leggere quello con la chiave principale più piccola)

Istruzione che cerca la chiave stringa <key\$> nel file a indici aperto con numero <numfile>, avvalendosi della chiave numero <numkey>. Le variabili del buffer associato vengono riempite con i corrispondenti valori della chiave trovata; se non è trovata, le variabili assumono i valori della chiave immediatamente successiva e se anche questa non esiste tutte le variabili del buffer sono impostate a "" oppure 0. Valorizza `_NRDATI` e `_ERRIND`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETLOC

GETLOC_STR <tipo>,<chiave>,<valore>

GETLOC_NUM <tipo>,<chiave>,<valore>

GETLOC_ARRAY <tipo>,<chiave>,<valore>

Parametri	
tipo	Tipo di memorizzazione: "T" = temporaneo, "P"=persistente;
chiave	Chiave dei dati;
valore	Variabile di interesse (stringa, numero, array)

Istruzione SprixMobile relativa al salvataggio dati in aree di memoria. L'istruzione legge stringa, numero e array. La variabile di riferimento relativa a messaggi di errore è _ERRLOC\$. Esempio:

```
PUTLOC_STR "T", "Cliente", CLI$      (Temporaneo)
GETLOC_STR "T", "Cliente", OLDCLI$
PUTLOC_NUM "P", "Operazioni", NUMOP (Persistente)
STORELOC "P", salva in modo persistente i dati in memoria.
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETLOT

GETLOT <idlot>

Parametri	
idlot	Numero ID lotto

Istruzione che legge i dati di Anagrafica Lotto relativi alle variabili struttura archivi _LOT in base al numero ID lotto <idlot>. La lettura valorizza _NDULOT (numero dati utente gestiti dal lotto). Valorizza le variabili _ERRLOT e _ERRLOT\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETLSTDOC

GETLSTDOC <idlista>,<sigladoc\$>,<sottoazienda>,<serie>,<numdoc>,<rigaordine>,<progressivopick>

Parametri	
idlista	ID lista di prelievo. Identificativo numerico univoco che identifica la lista per la quale recuperare i dati. Il range di valori per questa variabile va da 1 a 8000000.
sigladoc\$	Sigla documento. Se non viene specificata una sigla vengono considerate tutte le tipologie di ordini.
sottoazienda	Numero sotto azienda
serie	Serie documento. La serie dei documenti in lista di cui si vuole recuperare le informazioni, 0 = tutti.
numdoc	Numero documento. Il numero del documento in lista di cui si vuole recuperare le informazioni, 0 = tutti.
rigaordine	Numero di creazione riga ordine. Equivale a oorig().
progressivopick	Numero di pick. Numero progressivo di pick, può valere: <ul style="list-style-type: none"> • 0 ritorna tutte le righe, sia di tipo ordine che di tipo pick; • -1 ritorna tutte le righe ordine; • -2 ritorna tutte le righe di pick; • >0 ritorna la specifica riga di pick. Nota bene: le righe ordini si distinguono da quelle di pick perché hanno la variabile _LSTRPROG(R) pari a 0.

Istruzione che legge i dati della Lista di Prelievo relativi alla struttura archivi _LST e ritorna nella variabile _NLSTR il numero di record letti. L'istruzione valorizza le variabili _ERRLST e _ERRLST\$. L'istruzione richiede MDS.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETMM

GETMM <doc\$>

Parametri	
doc\$	Documento di magazzino dal leggere.

	<p>Sintassi per gestione di documenti dell'<u>anno solare</u> dell'azienda aperta:</p> <ul style="list-style-type: none"> • SIGLA[SERIE/]NUMERO (lettura di singolo documento); • SIGLA/NUMERO+SIGLA/NUMERO+ecc... (lettura multipla di documenti); <p>Sintassi per gestioni <u>infrannuali</u> (parametro magazzino "Numerazione solare movimenti magazzino (S/N): N") che consente di inserire l'anno del documento (AA) prima del protocollo:</p> <ul style="list-style-type: none"> • SIGLASERIE/AA-NUMERO (lettura di singolo documento); • SIGLA/AA-NUMERO+ SIGLA/AA-NUMERO+ecc... (lettura multipla di documenti); <p>Sprix mobile</p> <ul style="list-style-type: none"> • Non consente la lettura multipla di documenti Esempio: GETMM "BC1/1+BC1/2"; • SIGLA[SERIE/]NUMERO (sintassi per documenti che <u>non movimentano</u> il magazzino); • SIGLASERIE/AA-NUMERO (sintassi per documenti che <u>movimentano</u> il magazzino); i dati riguardano <u>tutti gli anni aperti</u>, pertanto, operando con archivi annuali (archivio mobile "MMAG") occorre <u>indicare l'anno solare</u> (Es: una azienda infra-annuale con anni fiscali aperti 2015 e 2016, trasferisce movimenti di magazzino con date documento che vanno dal 2015 al 2017); se sono rilevati più documenti, non è aperta alcuna finestra di selezione, poiché restituisce il "primo" documento trovato, in tal caso sfruttare la GETMM_EXT per indicare <u>anche il codice conto</u> del documento ricercato. <p>Esempio documenti azienda infra-annuale con data inizio anno contabile 01/05:</p> <p>Anno fiscale 2015 1 – BC 1/1 del 10/03/16 e cliente 501.00158</p> <p>Anno fiscale 2016 2 – BC 1/1 del 02/06/16 e cliente 501.00162 3– BC 1/1 del 24/10/16 e cliente 501.00158 4– BC 1/1 del 01/01/17 e cliente 501.00158</p> <p>A fronte di più record che soddisfano la stringa di ricerca (Es: Anno 2016 e GETMM "BC1/1"), lo <u>Sprix</u> restituirebbe una finestra di scelta tra i record 2 e 3. Se la stringa di ricerca restituisce due o più record in <u>Sprix</u> (Es: GETMM "BC1/16-1"), dato che lo <u>Sprix mobile</u> può lavorare contemporaneamente su due anni fiscali, ritorna ritorna il primo documento letto su database senza aprire una finestra di scelta.</p> <p>Nel caso particolare in cui i documenti tornati sono sia dell'anno fiscale in uso che dell'anno precedente, viene tornato il primo documento dell'anno fiscale in corso.</p> <ul style="list-style-type: none"> • In <u>Sprix mobile</u> l'anno fiscale in corso viene identificato dalla data di sistema e non dalla data di apertura azienda (come fa il gestionale); la data di sistema del dispositivo mobile condiziona l'individuazione dei documenti in lettura e in alcuni casi di aziende infra-annuali potrebbe non bastare la GETMM_EXT per garantire l'esatta individuazione di un documento. Esempio: <p>Lavorando in data (data sistema) 18/11/2016, nell'elenco dei documenti sopra riportato, non è possibile identificare univocamente i record 1 e 3. Entrambi i documenti rientrano tra i risultati della ricerca GETMM_EXT "BC1/16-1", "501.00158". Verrà sempre restituito il record 3 con data 24/10/2016 poiché è il primo documento trovato la cui data ricade nell'anno fiscale in corso. Attualmente, lavorando con la medesima data, in <u>Sprix mobile</u> non si può leggere il record 1, nemmeno attraverso la GETMM_EXT.</p>
--	---

Lettura il documento di magazzino <doc\$> e valorizza la struttura _MM; se rileva più documenti di clienti/fornitori diversi, apre una finestra di selezione documento (la pressione di [Esc] sulla finestra azzerà alcune variabili es. _ERRMM\$, _NMM); in alternativa utilizzare la GETMM_EXT. La lettura valorizza _NMM e _NTMM e le variabili _ERRMM e _ERRMM\$. Notare: le totalizzazioni di Riga e di Documento, avvengono tramite le istruzioni RIGMM e TOTMM.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETMM_EXT

GETMM_EXT <doc\$>[,conto\$]

Parametri	
doc\$	<p>Documento di magazzino dal leggere.</p> <p>Sintassi per gestione di documenti dell'<u>anno solare</u> dell'azienda aperta:</p> <ul style="list-style-type: none"> • SIGLA[SERIE/]NUMERO (lettura di singolo documento); • SIGLA/NUMERO+SIGLA/NUMERO+ecc... (lettura multipla di documenti); <p>Sintassi per gestioni <u>infrannuali</u> (parametro magazzino "Numerazione solare movimenti magazzino (S/N): N") che consente di inserire l'anno del documento (AA) prima del protocollo:</p> <ul style="list-style-type: none"> • SIGLASERIE/AA-NUMERO (lettura di singolo documento); • SIGLA/AA-NUMERO+ SIGLA/AA-NUMERO+ecc... (lettura multipla di documenti); <p>Sprix mobile</p> <ul style="list-style-type: none"> • Non consente la lettura multipla di documenti Esempio: GETMM "BC1/1+BC1/2"; • SIGLA[SERIE/]NUMERO (sintassi per documenti che <u>non movimentano</u> il magazzino); • SIGLASERIE/AA-NUMERO (sintassi per documenti che <u>movimentano</u> il magazzino); i dati riguardano <u>tutti gli anni aperti</u>, pertanto, operando con archivi annuali (archivio mobile "MMAG") occorre <u>indicare l'anno solare</u>

	<p>(Es: una azienda infra-annuale con anni fiscali aperti 2015 e 2016, trasferisce movimenti di magazzino con date documento che vanno dal 2015 al 2017); se sono rilevati più documenti, non è aperta alcuna finestra di selezione, poiché restituisce il "primo" documento trovato, in tal caso sfruttare la GETMM_EXT per indicare anche il codice conto del documento ricercato.</p> <p>Esempio documenti azienda infra-annuale con data inizio anno contabile 01/05:</p> <p>Anno fiscale 2015 1 – BC 1/1 del 10/03/16 e cliente 501.00158</p> <p>Anno fiscale 2016 2 – BC 1/1 del 02/06/16 e cliente 501.00162 3– BC 1/1 del 24/10/16 e cliente 501.00158 4– BC 1/1 del 01/01/17 e cliente 501.00158</p> <p>A fronte di più record che soddisfano la stringa di ricerca (Es: Anno 2016 e GETMM "BC1/1"), lo <u>Sprix</u> restituirebbe una finestra di scelta tra i record 2 e 3. Se la stringa di ricerca restituisce due o più record in <u>Sprix</u> (Es: GETMM "BC1/16-1"), dato che lo <u>Sprix mobile</u> può lavorare contemporaneamente su due anni fiscali, ritorna ritorna il primo documento letto su database senza aprire una finestra di scelta.</p> <p>Nel caso particolare in cui i documenti tornati sono sia dell'anno fiscale in uso che dell'anno precedente, viene tornato il primo documento dell'anno fiscale in corso.</p> <ul style="list-style-type: none"> In <u>Sprix mobile</u> l'anno fiscale in corso viene identificato dalla data di sistema e non dalla data di apertura azienda (come fa il gestionale); la data di sistema del dispositivo mobile condiziona l'individuazione dei documenti in lettura e in alcuni casi di aziende infra-annuali potrebbe non bastare la GETMM_EXT per garantire l'esatta individuazione di un documento. Esempio: <p>Lavorando in data (data sistema) 18/11/2016, nell'elenco dei documenti sopra riportato, non è possibile identificare univocamente i record 1 e 3. Entrambi i documenti rientrano tra i risultati della ricerca GETMM_EXT "BC1/16-1", "501.00158". Verrà sempre restituito il record 3 con data 24/10/2016 poiché è il primo documento trovato la cui data ricade nell'anno fiscale in corso. Attualmente, lavorando con la medesima data, in <u>Sprix mobile</u> non si può leggere il record 1, nemmeno attraverso la GETMM_EXT.</p>
Conto\$	Conto Cliente / Fornitore relativo al documento di magazzino

Letture il documento di magazzino <doc\$> e valorizza la struttura _MM; in caso di documento presente in archivio più di una volta ma di clienti/fornitori diversi l'istruzione consente di indicare il conto cliente/fornitore per identificare univocamente il documento evitando, quindi, l'apertura della finestra di selezione del documento; se non indicato il conto è come la GETMM. L'istruzione valorizza _NMM e _NTMM e le variabili _ERRMM e _ERRMM\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETMODSTPGRAF

GETMODSTPGRAF <tipo>

Parametri	
tipo	Tipo moduli richiesti; accetta i valori 0, 1 e 2 0= Predefiniti dentro la App e Personalizzati 1= Solo moduli Predefiniti dentro la App 2= Solo moduli Personalizzati

Istruzione che recupera l'elenco dei moduli di stampa Sprix predefiniti e personalizzati fuori dall'App. L'istruzione valorizza le variabili di struttura _MODGRA.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETMULTILOT

GETMULTILOT<[TipoLotto],[Codart],[TipoCodUt],[CodUtente],<Includi>,<Mag>

Parametri (in grassetto quelli obbligatori)	
TipoLotto	Tipo del lotto obbligatorio in alternativa al codice dell'articolo
Codart	Codice dell'articolo proprietario dei lotti, obbligatorio in alternativa al tipo lotto;
TipoCodUt	= 0 completo = 1 per la parte iniziale = 2 se parziale
CodUtente	Codice utente del lotto; se è impostato il codice utente [CodUtente] per ritornarne i suoi lotti il [TipoCodUt] stabilisce se questo codice utente sia completo (tipo=0), per la parte iniziale (tipo=1) o se parziale (tipo=2)
Includi	Inclusione obbligatoria 0 = include tutti i lotti, come F2, tranne quelli con stato B/N, con data validità inferiore alla data terminale e quantità 0 del solo archivio corrente; 1 = include tutti i lotti, come F3, del solo archivio corrente;

	2 = come 0 ma solo dei dati nello storico; 3 = come 1 ma solo dei dati nello storico; 4 = come 0 ma dei dati sia nel corrente che nello storico; 5 = come 1 ma dei dati sia nel corrente che nello storico;
Mag	Numero del magazzino; obbligatorio; (numero 1-255; se = 0 significa tutti i magazzini, non operativo se <Includi> = 1)

Istruzione che legge e carica l'elenco dei lotti relativi al codice di tipo lotto [TipoLotto] o al codice articolo [Codart], uno fra i due è obbligatorio, se inseriti entrambi in ricerca prevale Codart, nelle variabili array di struttura archivi _LOT?_M(), il cui numero di elementi è rappresentato dalla variabile _LOTNID_M e cioè il numero di ID letti e impostati da GETMULTILOT.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETNUM

GETNUM (<Tiposigla\$>,<SiglaDoc\$>,<Anno>,<Serie>)

Parametri	
Tiposigla\$	Tipo di area: "C" = contabile, "M" = magazzino;
SiglaDoc\$	Tipo di sezionale A/V/S/C/R/X/F se l'area è "C" o la sigla del documento di magazzino se l'area è "M";
Anno	Anno solare di riferimento;
Serie	Numero di serie (1 - 99);

Funzione che legge ritorna l'ultimo numeratore relativo all'area del <SiglaDoc\$> riferito all'anno solare <anno> e al numero di <Serie>. La funzione valorizza l'esito dell'operazione nella variabile _ERRNUM\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETNUMSTRUT

GETNUMSTRUT <codart\$>

Parametri	
codart\$	Codice principale dell'articolo;

Funzione che ritorna il valore numerico della struttura utilizzata dall'articolo con codice principale <codart\$>. I valori ritornati sono i seguenti:

- 1-18 se <codart\$> ha radice minima codificata e collegata a una struttura;
- 0 se <codart\$> ha radice minima codificata anche se non collegata ad alcuna struttura;
- -1 se <codart\$> non ha radice minima codificata;

La funzione è utilizzabile per sapere se la radice minima di <codart\$> può essere inserita oppure no. Se invece l'esigenza è quella di sapere se un codice esiste oppure no, basta utilizzare il dizionario ardes("<codart\$>") che, se ritorna valore significa che <codart\$> esiste, altrimenti, se non ritorna alcun valore significa che <codart\$> non esiste. Esempio (ipotesi: solo due codici nell'installazione), PF1 appartiene alla struttura 1 e PF2 non appartiene ad alcuna struttura:

- GETNUMSTRUT di qualunque <codart\$> che inizia con PF1 ritorna valore = 1;
- GETNUMSTRUT di qualunque <codart\$> che inizia con PF2 ritorna valore = 0 sia che <codart\$> esista oppure no;
- GETNUMSTRUT di qualunque <codart\$> che inizia con valori diversi da PF1 o PF2 ritorna valore = -1;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETNRI

GETNRIPRNVID()

GETNRIDOCVID()

GETNRIBLVID()

Funzioni Collage che rilasciano il progressivo di riga in cui l'utente è posizionato nel corpo della registrazione. Il numero progressivo di riga è quello che viene visualizzato a sinistra nella barra dei titolotoli delle colonne del corpo della registrazione.

GETNRIPRNVID() - lettura numero di riga del corpo di Primanota (PRN)
GETNRIDOCVID() - lettura numero di riga del corpo di Magazzino e avanzamento produzione SL/CL (DOC)
GETNRIBLVID() - lettura numero di riga del corpo di Produzione (BL)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento.

GETPAG

GETPAG <codpag>

Parametri	
codpag	codice numerico del pagamento; se codpag = 0 l'istruzione legge solo i dati generali

Istruzione che legge i dati del codice pagamento <codpag> presente nella tabella pagamenti; valorizza le variabili di struttura _PA? e riporta l'esito dell'operazione nelle variabili di errore _ERRPA e _ERRPA\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETPART

GETPART <tpart\$>,<codcon\$>,<catcon>,<codart\$>,<catart>,<dtvalda\$>

Parametri	
tpart\$	Tipo Particolarità: "P", "S", "V", "A", "I"; Attenzione: la variabile è il parametro principale che stabilisce il tipo di particolarità in oggetto ed è composta da un carattere (P/V/S/A/I); la variabile prevede, come nella GETPARTICOL, la possibilità di aggiungere anche un secondo ed un terzo carattere che condizionano altri parametri al fine di considerare anche la combinazioni di ulteriori elementi della particolarità. Per approfondimenti consultare GETPARTICOL.
codcon\$	Codice Conto Cliente/Fornitore;
catcon	Numero di categoria Listino, Sconto, Provvigione; (alternativo a <codcon>);
codart\$	Codice Articolo;
catart	Numero di categoria formazione Prezzo, Sconto, Provvigione; (alternativo a <codart\$>);
dtvalda\$	Data d'inizio validità della particolarità;

Istruzione che legge i dati delle Particolarità (Prezzi, Sconti, Provvigioni, Articolo) relativi alle variabili struttura archivi _PART, in base al tipo particolarità <tpart\$>, al codice conto <codcon\$> o numero di categoria <catcon>, al codice articolo <codart\$> o numero di categoria <catart> e con data d'inizio validità <dtvalda\$> della particolarità. La combinazione di quattro parametri individua la specifica particolarità; consente quattro combinazioni: (<codcon\$>,<codart\$>); (<codcon\$>,<catart>); (<catcon>,<codart\$>); (<catcon>,<catart>). L'istruzione valorizza le variabili _ERRPART e _ERRPART\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETPARTICOL

GETPARTICOL <tpart\$>,<codcon\$>,<catcon>,<codart\$>,<catart>,<dtvalda\$>,[mod]

Parametri	
tpart\$	Tipo Particolarità: "P", "S", "V", "A";
codcon\$	Codice Conto Cliente/Fornitore; se = "0" significa particolarità senza conto e senza categoria
catcon	Numero di categoria Listino, Sconto, Provvigione; (alternativo a <codcon>);
codart\$	Codice Articolo;
catart	Numero di categoria formazione Prezzo, Sconto, Provvigione; (alternativo a <codart\$>);
dtvalda\$	Data d'inizio validità della particolarità;
mod	Modalità di inizio ricerca delle particolarità;

Istruzione che legge i dati delle Particolarità (Prezzi, Sconti, Provvigioni, Articolo) relativi alle variabili struttura archivi _PART, in base al tipo particolarità <tpart\$>, al codice conto <codcon\$> o numero di categoria <catcon>, al codice articolo <codart\$> o numero di categoria <catart> e con data d'inizio validità <dtvalda\$> della particolarità. L'istruzione valorizza le variabili _ERRPART e _ERRPART\$.

La combinazione di quattro fra i suddetti parametri individua la specifica particolarità.

Sono consentite quattro combinazioni: (<codcon\$>,<codart\$>); (<codcon\$>,<catart>); (<catcon>,<codart\$>); (<catcon>,<catart>).

L'istruzione dispone del parametro [mod] che attiva la modalità di inizio ricerca delle particolarità che soddisfano i requisiti dichiarati nei parametri:

- se [mod] = 0 ricerca dalla 1^ particolarità congruente ai requisiti; non controlla la data d'inizio validità <dtvalda\$>;
- se [mod] <> 0 ricerca le successive congruenti; non controlla la data d'inizio validità <dtvalda\$>;
- se [mod] = 2 ritorna la prima occorrenza valida tra quelle presenti; se nessuna soddisfa i requisiti ritorna nulla; considera la data di d'inizio validità <dtvalda\$>;

Se cambiano i tipi di parametri, impostare nuovamente l'istruzione con dei nuovi parametri per leggere le ulteriori particolarità.

Attenzione: Al fine di conservare la retrocompatibilità degli Sprix, oltre alla consueta costruzione della chiave della particolarità (ovvero combinando i parametri <tpart\$>,<codcon\$>,<catcon>,<codart\$>,<catart>,<dtvalda\$>) è consentito anche un **utilizzo più esteso** del

parametro **<tpart\$>** in modo da definire ulteriori campi di selezione (oltre alle categorie), e cioè, una combinazione che consideri anche il seguente dizionario relativo a determinate tabelle:

"tzstc"	"CATEGORIA STAT.CONTO"
"tzzon"	"ZONA CONTO"
"tzgrm"	"GRUPPO MERCEOLOGICO"
"tznat"	"NATURA"
"tzsaa"	"CAT.STATISTICA ALF." + "tzsan" "CAT.STATISTICA NUM."

In aggiunta al carattere che definisce il tipo di particolarità (<tpart\$> = "P" oppure "V" oppure "S" oppure "A" oppure "I"), infatti, sono consentiti anche un **secondo** ed un **terzo** carattere che attivano alcune eccezioni.

Questi due caratteri sono collegati ai seguenti parametri, in modo **posizionale**:

<catcon> è collegato al secondo carattere;
<codart\$>, **<catart>** sono collegati al terzo carattere;

Oltre al tipo di particolarità <tpart\$> di un carattere, quindi, si possono aggiungere **due caratteri** per indicare la condizione. Qui di seguito sono indicati i valori che i due caratteri possono avere e sono indicati i relativi parametri in cui indicare la condizione:

carattere	valore	per che cosa	da indicare in
2°	niente (oppure N)	Categoria Conto (tzcac)	<catcon>
2°	Z	Zona (tzzon)	<catcon>
2°	S	Categoria Statistica (tzstc)	<catcon>
3°	niente (oppure T)	Categoria Articolo (tzcaa)	<catart>
3°	M	Gruppo Merceologico (tzgrm)	<codart\$>
3°	U	Natura (tznat)	<codart\$>
3°	E	Cat. St.Articolo (tzsaa+tzsan)	<codart\$> (lettera) <catart> (num) - eccetto nel dizionario -
note:			
a) la combinazione delle lettere consente la presenza di spazio, esempio nei prezzi: "P U", "P M", "PS ", "PN ", ecc..;			
b) nel dizionario la lettera "E" richiede la concatenazione in <codart\$> della stringa tzsaa+STR\$(tzsan,0,0);			

Esempio sulla differenza di dizionario con uno e con due caratteri del parametro <tpart\$>

- tzpr8("P", "", 2, "MERCE", 0, "20180101") significa particolarità di tipo **Prezzo** per la **Categoria Conto 2** e per l'**Articolo** "MERCE"
- tzpr8("PZ", "", 2, "MERCE", 0, "20180101") significa particolarità di tipo **Prezzo** per la **Zona 2** e per l'**Articolo** "MERCE"

Esempi di lettura

- GETPART "PZ", "", 1, "", 2, "20180130"
- GETPART "SZM", "", 1, "A11", 0, "20180130"
- GETPART "PZU", "", 5, "5", 0, "20180101"

[esempio](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETPC

GETPC <codcon\$>

Parametri	
<codcon\$>	Codice Conto di Anagrafica Piano dei Conti o Rubrica Clienti fornitori

Istruzione che legge i dati del Conto di Anagrafica Piano dei Conti o Rubrica Clienti Fornitori relativi al codice conto <codcon\$>. Valorizza ERRPC, ERRPC\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETPC2

GETPC2 <indice\$>, <datoricerca\$>, [tipoconto\$], [tiporicerca\$], [finestradiologo\$], [contobase\$]

Parametri	
<indice\$>	Ricerca per: "D" = Descrizione (case insensitive); "P" = Partita Iva (come codificata es. paese "IT" davanti); "F" = Codice Fiscale;
<datoricerca\$>	Stringa da cercare; esempio: "Abc"
[tipoconto\$]	Tipologia di conto Piano dei Conti; valori ammessi: "" = Solo Clienti e Fornitori; "#" = Solo conti <> da Clienti e Fornitori, solo se <indice\$> = "D"; "C" = Solo Clienti oppure un tipo conto valido se <indice\$> <> "D"; "F" = Solo Fornitori oppure un tipo conto valido se <indice\$> <> "D";
[tiporicerca\$]	Ricerca la stringa in modo:

	<p>"" = Puntuale, confronta interamente il <datoricerca\$> ed il dato su record (es. abc <> abcPPP) "P" = Parziale, confronta il <datoricerca\$> per la sua lunghezza (es. abc = abcPPP)</p>
[finestradiologo\$]	<p>Finestra di selezione; valori ammessi: "" se trova piu' elementi viene caricata _CODPICO\$() di dimensione _CODPICONR elementi; "S" se trova piu' elementi apre la finestra selezione (e _CODPICONR=1); la singola selezione del codice conto corrisponde a _CODPICO\$(1) di _CODPICONR=1;</p> <p>Parametro non disponibile in Shaker.</p>
[contobase\$]	<p>Codice conto. valori ammessi:</p> <p>se = "" la ricerca parte dal record con chiave ordinata crescente se <> "" (es. "GMM.CCCCC") viene usato insieme al <datoricerca\$> come start di ricerca che ha esito positivo sul record successivo (la ricerca torna il record successivo a quello identificato da <datoricerca\$> con <contobase\$> solo se <tiporicerca\$>="" e la key viene trovata);</p> <p>notare l'utilizzo del codice conto è quindi utile solo per le ricerche del successivo elemento con chiave secondaria <indice\$> stabilita; se si cerca una chiave secondaria indicando anche il giusto e preciso [contobase\$] il dato non viene trovato, ma è dichiarato assente oppure non presente in archivio (in tale caso utilizzare la GETPC), esempio "codice fiscale / descrizione / p.Iva successivo a codice origine [contobase\$] assente"; se invece le indicazioni sono sbagliate, in caso di insuccesso viene rilasciato il messaggio "codice fiscale / descrizione / p.Iva su codice origine [contobase\$] non trovato".</p>

Istruzione di ricerca conti nel Piano dei Conti (PICO) per Descrizione, Partita Iva e Codice Fiscale. Ritorna la prima occorrenza trovata oppure un elenco di selezione. **Nota: la ricerca parte dal primo carattere a sinistra, non è consentita una porzione interna di stringa.**

Le suddette tre chiavi sono secondarie rispetto al codice conto "GMM.CCCCC". Variabili di ritorno: l'array _CODPICO\$() viene dimensionata di elementi quando _CODPICONR=1 e viene caricata la relativa struttura _PC???

La variabile _CODPICONR vale 1 anche in caso di finestra dialogo con scelta; se _CODPICONR=0 la variabile _ERRCODPICO\$ contiene l'errore.

Dalla versione 2021A (v 805) GETPC2 è disponibile in Shaker, ma senza il parametro finestradiologo\$, perchè non è consentito in Shaker aprire finestre di dialogo. Il numero di risultati è ritornato nella CODPICONR. Nel caso il risultato sia uno solo verranno valorizzate le variabili _PC come nella GETPC. Nel caso di più risultati i codici saranno presenti nell'array CODPICO_S. Eventuali errori sono gestiti nella variabile ERRCODPICO_S

Esempio Sprix

A:
CLS

```
LOCATE 2,1 : PRINTD "Ricerca D=Des P=P.I. F=C.F. :"  
LOCATE 3,1 : PRINTD "Dato ricerca.....:"  
LOCATE 4,1 : PRINTD "Tipo conto C/F/.....:"  
LOCATE 5,1 : PRINTD "Ricerca "P"parziale,"uguale:"  
LOCATE 6,1 : PRINTD "Finestra dialogo S/N      :"  
LOCATE 7,1 : PRINTD "Conto base.....:"
```

```
LOCATE 2,30 : INPUT "AM",NKEY$, 1  
LOCATE 3,30 : INPUT "A",KEY$, 60  
LOCATE 4,30 : INPUT "AM",TPC$, 1  
LOCATE 5,30 : INPUT "AM",TPR$, 1  
LOCATE 6,30 : INPUT "AM",DIA$, 1  
LOCATE 7,30 : INPUT "C",CONTO$, 9
```

```
GETPC2 NKEY$, KEY$,TPC$,TPR$,DIA$,CONTO$  
IF _ERRCODPICO$<>"" THEN VIMSG 4, _ERRCODPICO$
```

```
IFF _CODPICONR > 1
```

```
  AZZVARSYS 51  
  _CTTITCOL$(1) = "Codice PICO " : _CTTYPECOL$(1) = "S" : _CTLENCOL(1) =14: _CTVWLENCOL(1) =14 ' WKD_ANACFI$ =  
  _CTVTOTCOL = 1  
  FOR I=1 TO _CODPICONR  
    _CTVALCOL1$(I) = LEFT$_CODPICO$(I)+"  ",14)  
  NEXT  
  _CTVTOTRIG = _CODPICONR  
  CREATELIST "Anagrafiche trovate "+ STR$_CODPICONR,0,0), 0, 0, 0  
  IF _ERRCT <> 0 THEN VIMSG 4, STR$_ERRCT,0,0) + " " + _ERRCT$  
  IF _CTTOTRIGSEL THEN VIMSG 4, "Selezionato " + _CTVALCOL1$( _CTINDRIGSEL(_CTTOTRIGSEL)) + " struttura _PC da caricare con  
GETPC"  
ELSEF  
  IF _CODPICONR = 1 THEN VIMSG 4, "Ritornato " + _CODPICO$(1) + " e struttura _PC valorizzata:" + _PCDESS "+"+_PCCOD$  
ENDIF
```

GOTO A

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETPN

GETPN <dreg\$>,<prinr>,<cau\$>,<protr\$>,<prots>,<protn>,<docn>,<ddoc\$>

Parametri	
<dreg\$>	Data di registrazione;
<prinr>	Numero progressivo di registrazione di primanota (1 – 16.777.214); solo GET; vedere _PNPRN;
<cau\$>	Causale del documento;
<protr\$>	Registro di protocollo;
<prots>	Serie di protocollo;
<protn>	Numero di protocollazione;
<docn>	Numero del documento
<ddoc\$>	Data del documento

Istruzione che legge i dati della registrazione di Primanota relativi alle variabili di struttura archivi _PN. I parametri dreg\$ e prinr sono chiavi di ricerca in caso di assenza dei restanti parametri identificativi della registrazione. Valorizza _NPN, _ERRPN, _ERRPN\$. Per approfondimenti consultare il paragrafo Primanota del capitolo Comandi Raggruppati per Argomento.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETPOCURSX

GETPOCURSX(<par>)

Parametri	
<par>	Parametro che assume valore 1 se la finestra è del programma Sprix/Collage oppure 2 se la finestra è del gestionale nella routine Collage, indipendentemente dall'interfaccia.

Funzione che ritorna la colonna in cui è posizionato il cursore in quel momento. L'utilizzo della ATTIVAFIN riposiziona alle coordinate 1,1 della finestra. Unica eccezione le finestre elenco articoli e Clienti/fornitori dove il valore ritorna impostato al massimo delle colonne gestite.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETPOCURSY

GETPOCURSY(<par>)

Parametri	
<par>	Parametro che assume valore 1 se la finestra è del programma Sprix/Collage oppure 2 se la finestra è del gestionale nella routine Collage, indipendentemente dall'interfaccia.

Funzione che ritorna la riga in cui è posizionato il cursore in quel momento. L'utilizzo della ATTIVAFIN riposiziona alle coordinate 1,1 della finestra. Unica eccezione le finestre elenco articoli e Clienti/fornitori dove il valore ritorna impostato al massimo delle colonne gestite. Non gestibile nei campi di input MALISTIN, utilizzare quelli precedenti (MAGESTLIST)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETPU

GETPU <idubi>,<prog\$>,[idart\$]

Parametri	
idubi	id ubicazione
prog\$	Flag "N" (non ritorna progressivi) oppure "S" (ritorna i progressivi per l'ubicazione indicata)
idart\$	id articolo (codice primario dell'articolo); parametro ignorato se prog\$="N"

Istruzione che legge i dati delle variabili di struttura _PU in base a "ID ubicazione".

Indicando i soli primi due parametri legge l'ubicazione per ID e ritorna un array contenente per ogni articolo i suoi progressivi.

Indicando tutti i parametri legge l'ubicazione per ID e ritorna un array con un solo elemento contenente i progressivi dell'articolo il cui codice primario è indicato in [idart], cioè, indicando prog\$="S" e idart\$="cod.art." la GETPU ritorna puntuale i soli progressivi dell'ubicazione indicata per il solo articolo specificato.

L'istruzione valorizza le variabili di errore _ERRPU e _ERRPU\$. La struttura viene azzerata con AZZVARSYS 72 oppure AZZCATVARSYS "PU".

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento [Clicca qui](#)

GETREM

GETREM_NUM <tag\$>,<numero>

GETREM_STR <tag\$>,<stringa\$>

GETREM_ARRAY <tag\$>,<array_di_stringhe\$ / array_di_numeri>

GETREM_FILE <tag\$>,<percorso_file\$>

Parametri	
tag\$	Tag assegnato;
numero	Variabile numerica;
stringa\$	Variabile alfanumerica;
array stringhe/numeri	Vettore alfanumerico / numerico;
percorso file\$	Directory del file;

Istruzione Sprix Mobile per leggere i valori che si vogliono inviare al Mobile o al Server. L'istruzione valorizza le variabili _ERRGETREM\$ per la gestione dell'errore ritornato da una qualsiasi GETREM...

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETROWVALFLD

GETROWVALFLD <"vararch\$">,<varloc>

Parametri	
"vararch\$"	Variabile archivio: nome della variabile che si vuole leggere per l'archivio interessato dalla lista;
varloc	Variabile locale: Nome della variabile personale in cui impostare il valore della variabile archivio;

Istruzione che imposta in una variabile locale il valore specificato nella variabile "vararch\$". Esempio: GETROWVALFLD "_MMCLI\$",TMP\$
Imposta nella variabile TMP\$ il valore letto nella variabile _MMCLI\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETRUBUNICA

GETRUBUNICA <id>

Parametri	
id	Identificativo numerico della Rubrica Unica.

Istruzione che legge i dati di Rubrica Unica per l'identificativo specificato. Valorizza le variabili della struttura del piano dei conti _PC relative ai dati di Rubrica. L'istruzione valorizza le variabili _ERRRUBUNI e _ERRRUBUNI\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETSIGN

GETSIGN <titolo_finestra_firma\$>

Parametri	
titolo_finestra_firma\$	titolo che comparirà sul form in cui sarà presente l'area in cui apporre la firma

Istruzione con cui è possibile fare una stampa PDF di un form o acquisire un file già esistente, associargli una firma e poi sincronizzare file e firma, lato server, su Docuvision. I documenti (pdf per il form e jpg per la firma) vengono salvati in locale e, se opportunamente specificato, vengono inviati a Docuvision dopo una sincronizzazione.

L'istruzione richiede l'inizializzazione di determinate variabili _MM? e _SN?

Per approfondimenti consultare il manuale Sprix Mobile.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETSORT

GETSORT <str\$>

Parametri	
str\$	Stringa da leggere

Istruzione che legge la prossima stringa <str\$> dalla lista di record ordinati da PUTSORT. Quando tutti record sono stati letti in <str\$> è impostato il valore "".

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETTLOT

GETTLOT <tplot\$>,[ord]

Parametri	
tplot\$	Codice Tipo Lotto; (in caso di nuovo tipo lotto indicare "***");
ord	Ordine di lettura dati utente personalizzati (1 = per codice dato; 2 = per come immessi a video –default se non viene indicato-; 3 = in base all'ordine impostato per il lotto);

Istruzione che legge i dati di Anagrafica Tipo Lotto relativi alle variabili struttura archivi _TPLOT, in base al codice Tipo Lotto <tplot\$> (in caso di nuovo tipo lotto indicare "***") e l'eventuale ordine di lettura dati utente personalizzati. L'istruzione valorizza _NDUTPLOT (numero dati utente gestiti). L'istruzione valorizza le variabili _ERRTPLOT e _ERRTPLOT\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GETZA

GETZA <id>,<data\$>

Parametri	
id	ID anagrafica unica;
data\$	Data (se impostata a "" viene recuperata l'anagrafica corrente);

Istruzione che legge i dati dell'Anagrafica Unica corrente e storica relativi alle variabili struttura archivi _ZA, in base al codice <id> e data <data\$>. L'istruzione valorizza le variabili _ERRZA e _ERRZA\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GIORSET

GIORSET(<data\$>)

Parametri	
data\$	Data in formato "AAAAMMGG" compreso fra 19600101 e 20481231

Funzione che ritorna il giorno della settimana della stringa <data\$>. I numeri relativi ai giorni sono: 1 (lun), 2 (mar), 3 (mer), 4 (gio), 5 (ven), 6 (sab), 7 (dom). Esempio: OGGI\$ = "20090618": N = GIORSET(OGGI\$) risultato N = 4 (giovedì).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GOTO

GOTO <nometic>

Parametri	
nometic	Etichetta a cui saltare;

Istruzione per eseguire il salto alla subroutine dell'etichetta <nometic>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GOSUB

GOSUB <nometic> ... RETURN

Parametri	
nometic	Etichetta a cui saltare;

Istruzione per eseguire il salto alla subroutine dell'etichetta <nometic> dello stesso programma, terminata la quale ne esce con RETURN per tornare al punto di partenza e successivo all'ultimo GOSUB eseguito. GOSUB è nidificabile (uno nell'altra fino a 30). In caso di ciclotabellare, si esce con EXITTAB e non con RETURN (da posizionare dopo e fuori il ciclotabellare).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

GRIGLIAFIN

GRIGLIAFIN <n1>,<n2>,<n3>,<n4>,<n5>,<n6>,[s\$]

Parametri	
n1	Numero finestra; corrispondente al n. finestra di riferimento aperta con aprifin_ext
n2	Flag attiva (=1) o disattiva (=0) la griglia;
n3	Numero riga iniziale;
n4	Numero colonna iniziale;
n5	Numero massimo righe;
n6	Numero massimo colonne;
s\$	Stringa non attiva (parametro predisposto);

Istruzione per attivare la lettura facilitata dei dati a video. Imposta la caratteristica del numero di finestra con lo sfondo a griglia di righe alternate "chiaro/scuro";

W=AUTOFIN():
 APRIFIN_EXT W,0,0,0,ALT,LARG,"Titolo","A505,A610","NOVISEZIONE"
 GRIGLIAFIN 1,1,1,1,10,20,""

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

HTTPHDRTAG

HTTPHDRTAG\$(<par\$>)

Parametri	
par\$	Parametro di delimitazione Tag dell'array stringa. Valori consentiti: "BEGIN" oppure "END"

Funzione per gestire gli header HTTP superiori a 255 caratteri (massimo valore variabili Sprix).

Queste stringhe sono divisibili in più righe dell'array di Output _WSOHEADER\$() [ed analogamente, per quello di Input _WSHEADER\$()], delimitandoli con due appositi Tag per marcare l'inizio e la fine del valore, e che sono richiamati dalla suddetta funzione.

I valori dei Tag si ottengono, quindi, richiamando questa funzione HTTPHDRTAG\$() a cui si passa il valore stringa "BEGIN" per ottenere il Tag di Inizio ed il valore "END" per ottenere il Tag di Fine.

Questi due Tag vanno inseriti in un elemento dello stesso array, ed in questo elemento, l'unico valore presente deve essere quello del Tag; tutti gli elementi che stanno fra quello con il Tag HTTPHDRTAG\$("BEGIN") ed il Tag HTTPHDRTAG\$("END"), saranno considerati come un unico valore. In altri termini, gli elementi che contengono i Tag che consentono di individuare gli elementi su cui un valore è stato suddiviso devono essere l'unica stringa contenuta in quell'elemento dell'ARRAY.

Il parametro <Par\$> può, quindi, assumere i valori "BEGIN" / "END", ma qualunque altro valore fa ritornare una stringa vuota (""), pertanto:

- HTTPHDRTAG\$("BEGIN") ritorna il Tag "__HEADER_BEGIN__"
- HTTPHDRTAG\$("END") ritorna il Tag "__HEADER_END__"

Si consiglia di usare sempre la funzione per ottenere i valori dei due Tag, in modo che, se questi dovessero variare, non occorre modificare i sorgenti degli SPRIX che li usano.

[Esempio](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

IF THEN ELSE

IF <espressione> THEN <comando1> [ELSE <comando2>]

Istruzione esegue <comando1> se <espressione> è una condizione vera altrimenti esegue <comando2>. THEN è obbligatorio ed ELSE è facoltativo, ma sempre tutto sulla stessa riga. La scorretta indentazione della IF dentro un ciclo (es. FOR) può rendere non operativa la condizione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

IFF ELSEF ENDF

IFF <espressione> <comandi1> [ELSEF <comandi2>] ENDF

Istruzione che esegue i <comandi1> se l'<espressione> è una condizione vera, altrimenti esegue i <comandi2> se presenti. ELSEF è facoltativo mentre ENDF è obbligatorio. I <comandi> sono istruzioni che possono anche essere messe su più righe. Le varie IFF possono essere annidate. La scorretta indentazione della IF dentro un ciclo (es. FOR) può rendere non operativa la condizione. Non è consentito GOTO.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INCAR

INCAR(opz\$)

Parametri	
opz\$	L'opzione può essere impostata con le seguenti: <ul style="list-style-type: none"> • "SOLOTFA" per ricevere i soli tasti attivati dalla SETTF; esempio T=INCAR("SOLOTFA") • "VISCURS" per visualizzare il cursore; esempio T=INCAR("VISCURS") • "VISCURSSOLOTFA" per gestire entrambe; es. T=INCAR("VISCURSSOLOTFA")

Funzione che consente la ricezione del tasto premuto; consente di selezionare un elemento dalla lista tramite mouse: se sono state impostate INPUT_EXT o SETZM, la funzione rileva anche se è premuto del mouse sulla zona attiva e ritorna le variabili _TF, _TMZ, _TMR, _TMC.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INIZ

INIZCO\$()

INIZES\$()

INIZGR\$()

INIZIT\$()

INIZSO\$()

Funzioni che ritornano una stringa di caratteri da inviare alla stampante per impostare l'inizio del modo di stampa in compresso INIZCO\$(), espanso INIZES\$(), grassetto INIZGR\$(), corsivo INIZIT\$(), sottolineato INIZSO\$(). Non tutti tipi di formato consentono interventi sullo stile del font (esempio il sottolineato in pdf, corsivo o espanso in rtf, ecc). Esempio LPRINT INIZCO\$();

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INPUT

INPUT <tipo\$>,<str\$>,<larg>

Parametri	
tipo\$	<ul style="list-style-type: none"> • "A" Alfanumerico; • "AM" Alfanumerico maiuscolo; • "AE" Alfanumerico caratteri estesi comprese le lettere accentate; • "D" Data con controllo di validità; accetta date dal 01/01/1960 al 31/12/2048; • "DL" Data libera accetta qualunque data; (data libera < 01/01/1960 o > 31-12-2048); • "C" Tutti i conti contabili del piano dei conti; • "C+" Solo i conti Clienti e Fornitori; • "C-" Tutti i conti, escluso i Clienti e Fornitori; • "C?" Conto in cui ? identifica il tipo di conto ammesso: M=Merci, S=Spese, V=Vendite beni ammortizzabili, L=Costi pluriennali, P=Patrimoniali generico, C=Clienti, F=Fornitori, I=Immobilizzazioni, K=Costi pluriennali, R=Rimanenze, O=Conti d'ordine, Z=Conti sospesi). Esempio CONT\$="5" : INPUT "CC",CONT\$,9; • "M" Articolo di magazzino; Notare: se non ancora implementato, l'accesso nella revisione dell'anagrafica articolo non consente di accedere ai moduli "SCHEDE ARTICOLI" e "SCHEDE MOVIMENTI DI MAGAZZINO" tramite il pulsante funzione [SHF7] "Visualizza righe documenti", in quanto è prerogativa della medesima funzione da menu Magazzino. • "N(1-6)" Numerico fino a un massimo di 6 decimali; es. N, N1, N2, N3, N4, N5, N6 (numero con 0,1,2,3,4,5,6 decimali); • "ST*" Monitor stampanti; NB: cambia la natura dei parametri <str\$> e <larg>; <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>(* può assumere i seguenti valori "tipo stampante": ""=Tutto il monitor stampanti "V"=Video "F"=File "E"=Email "R"=Word (rtf) "X"=Excel (csv) "D"=Docuvision "S"=Stampante testo (txt) "W"=Stampante "&"= Solo le disponibili "\$"= Solo le esistenti</p> <pre> _SCORRI=1 CC: LOCATE 2,2: PRINT "TUTTE" : LOCATE 2,20: INPUT "ST",NST1,32 LOCATE 3,2: PRINT "FILE" : LOCATE 3,20: INPUT "STF",NST2,32 IF _SCORRI = 1 THEN _SCORRI = 0: GOTO CC: END </pre> </div> <ul style="list-style-type: none"> • "ST2" Monitor stampanti; NB: cambia la natura dei parametri <str\$> e <larg> vedi [param\$]; • "I"=Internet, attiva il tasto "Apri il Browser" in basso a sx nella finestra; se premuto richiama il Browser predefinito passandogli l'URL inserito nel campo input. Viene controllata la validità formale dell'url passato; se il controllo fallisce viene rilasciato un messaggio sotto il campo di Input. Es: LOCATE 2,2: PRINT "Internet" : S\$="WWW.LIBERO.IT" : LOCATE 2,20: INPUT "I",S\$,32. • "E"=Email, attiva il tasto "Invia posta elettronica" in basso a sx nella finestra; se premuto richiama il client di posta predefinito passandogli come email di destinazione quella inserita nel campo di Input. Non viene controllata la validità

	<p>formale dell'email passata. Es: LOCATE 6,2: PRINT "Mail" : S\$="AAA@LIBERO.IT" : LOCATE 6,20: INPUT "E",S\$,32</p> <ul style="list-style-type: none"> • "T"=Telefono, attiva il tasto "Chiama il numero" in basso a sx nella finestra; se premuto, richiama tramite il protocollo TEL, il client predefinito per eseguire la telefonata (cisco, skype,..). Se per l'installazione non è stato impostato il codice d'uscita dal centralino, questo viene richiesto. Se memorizzato, non verrà più richiesto. Questo codice, viene passato come prefisso al numero di telefono inserito nel campo di input. Il valore può essere modificato successivamente da Impostazioni di configurazione (Codice di uscita centralino telefonico). Es: LOCATE 4,2: PRINT "Tel" : S\$="800414243" : LOCATE 4,20: INPUT "T",S\$,32 • "W" stringa di tipo Password maiuscola/minuscola criptata con caratteri asterisco (*) e possibilità di visualizzazione; la possibilità di attivare la gestione della conferma della "Nuova Password" è consentita nelle input che prevedono dei parametri (INPUT_EXT, INPUT_EXT_EXT, INPUT3) inserendo "CONFERMA" come parametro. (dalla v711) • "WP" stringa di tipo Password solo maiuscola criptata con caratteri asterisco (*) e possibilità di visualizzazione; la possibilità di attivare la gestione della conferma della "Nuova Password" è consentita nelle input che prevedono dei parametri (INPUT_EXT, INPUT_EXT_EXT, INPUT3) inserendo "CONFERMA" come parametro. (dalla v711) • "U" per l'utilizzo di codici ubicazione: deve essere seguito da un numero che va da 1 a 999 e rappresenta il numero del magazzino per il quale si vogliono gestire le ubicazioni. Se viene specificato un numero di magazzino superiore al numero dei magazzini gestiti viene rilasciato il messaggio di errore : "Il numero di magazzino fornito per l'input di tipo Ubicazione è superiore al numero di magazzini gestiti". Se viene indicato il tipo dato "U" senza numero di magazzino viene rilasciato il messaggio di "Tipo di input non valido". <p>Nota: questa differenziazione dell'istruzione non si estende anche ai Parametri di Esecuzione [F6].</p>
str\$	<p>Variabile dichiarata per l'input. Se <str\$> contiene già un valore questo è apposto come dato di ingresso modificabile (viene visualizzato il contenuto della variabile, il cursore si ferma sul primo carattere del campo, accetta e controlla l'immissione o la variazione del campo stesso; se è stringa e contiene un carattere ASCII 0, questo è considerato come la fine della stringa; se è array l'indice deve essere una costante, esempio INPUT "A",VAR\$(2),1.</p> <p>Se il parametro <tipo\$> = "ST*" il parametro <str\$> diventa una variabile numerica (con valori da 0 a 128) e attiva il tasto funzione [F2] monitor di scelta stampanti che sono configurate.</p> <p>Nota: INPUT non accetta array con indice sviluppato con ciclo FOR (o altro contatore), altrimenti rilascia messaggio "manca la costante numerica". Occorre sviluppare separatamente le costanti.</p>
larg	<p>Dimensione visibile del campo d'input.</p> <ul style="list-style-type: none"> • se <tipo\$> = "M" o "C" la <larg> superiore alla dimensione del codice visualizza la descrizione nella parte eccedente; la variabile viene impostata con il codice. • se <tipo\$> = "D", <larg> deve essere 10 o 8 o 5 se <str\$> di input è nella forma GG/MM/AAAA o GG/MM/AA o GG/MM (con AAAA assunto in data odierna); la variabile è sempre impostata nella forma "AAAAMMGG". • se <tipo\$> = "ST*" il parametro <larg> dimensiona la descrizione a video relativa alla stampante selezionata, dal carattere 2 fino a massimo 32 caratteri (se <larg> = 1 non visualizza la descrizione). Esempio: NST=3 : INPUT "ST",NST,20 <p>Nota: La larghezza del campo non può eccedere i limiti di colonne consentiti dalla finestra (vedere APRIFIN). Se è necessario inserire una stringa più lunga occorre spezzare il campo di input in più parti, oppure avvalersi dell'istruzione INPUT_EXT per lo scorrimento dei caratteri fino al massimo consentito dalla stringa.</p>

Istruzione che consente l'immissione di <str\$> (numerico o stringa) da tastiera nella finestra attiva, alla posizione attuale del cursore, in un campo di massimo <larg> caratteri in base al <tipo\$> di immissione. L'istruzione attiva le funzioni "monitor" standard del gestionale [F2] (ricerca, modifica, inserimento di un conto, di un articolo, di una data, ecc.); anche i Parametri di Esecuzione [F6] possono attivare questa funzione "monitor" della INPUT. L'istruzione è condizionata da variabile _SCORRI e valorizza la _TF (eccetto la pressione dei tasti funzione monitor quando <tipo\$>="C", "M" o "D" non c'è un'uscita dal campo e non ci sono eventi). L'istruzione non consente la scelta del font. Esempio selezione INPUT Stampante numero 3 : NST=3 : INPUT "ST",NST,20

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INPUT3

INPUT3 <numzonamouse>,<tipo\$>,<str\$>,<larg>,<maxlunstr>,<param\$>,<err\$>,[elencoscelte\$]

Parametri	
numzonamouse	Come l'istruzione INPUT_EXT_EXT, ovvero: vale da 1-127 (0 = disabilita la zona mouse; -1 = disabilita tutte le zone mouse della finestra); il parametro param\$ ha valore 'null', pertanto va scritto come "".
tipo\$	Come l'istruzione INPUT_EXT_EXT
str\$	Come l'istruzione INPUT_EXT_EXT
larg	Come l'istruzione INPUT_EXT_EXT, ovvero: Indica il numero dei caratteri visibili dell'input
maxlunstr	Come l'istruzione INPUT_EXT_EXT, ovvero: Dimensione massima del campo di input
param\$	Come istruzione INPUT_EXT_EXT:
	<p>Solo per INPUT3, indipendentemente dal tipo, sono disponibili i seguenti:</p> <ul style="list-style-type: none"> - Tipo = "ST2": <ul style="list-style-type: none"> o DISP = identifica i tipi di dispositivi che si vogliono utilizzare. <tipo\$> o FORM = Filtra la lista per i formati specificati o NODISP abilita la visualizzazione delle stampanti non disponibili. o NOESIST abilita la digitazione delle stampanti non esistenti.

err\$	Impostato un valore per la variabile, viene mostrato il messaggio di errore nel campo. La gestione della finestra è automatica.
elencoscelte\$	[parametro facoltativo]: elenco scelte per tipo di input "COMBO": array monodimensionale di stringhe alfanumeriche nella forma: <codice\$>=<descrizione\$>; Il carattere pipe ' ' non è ammesso e verrà eliminato). Il <codice\$> è un valore di <u>un solo carattere</u> ; la quantità di caratteri disponibili è sufficienti per il combobox (altrimenti è più potente usare la CREATELIST). Esempio: "7=Matita", "b=penna", "l=Lapis", "G=biro"

Istruzione estesa dell'istruzione INPUT_EXT. In uscita vengono valorizzate le seguenti variabili (stesso significato di INPUT_EXT):
_TF, _TMZ, _TMR, _TMC

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INPUT_EXT

INPUT_EXT <numzonamouse>,<tipo\$>,<str\$>,<larg>,<maxlunstr>,<param\$>

Parametri	
numzonamouse	vale da 1-127 (0 = disabilita la zona mouse; -1 = disabilita tutte le zone mouse della finestra); il parametro param\$ ha valore 'null', pertanto va scritto come "". Se = 0 sono attivate le variabili coordinate colonna (_TMC) e riga (_TMR).
tipo\$	Come l'istruzione INPUT. Solo per INPUT_EXT sono disponibili le due seguenti tipologie: <ul style="list-style-type: none"> • "F" tipo file abilita la navigazione del file system del Server oppure del Client; se viene indicato il simbolo di reindirizzamento "<", nella variabile <str\$> ritorna il percorso con il nome file. Nella componente Client WebDesk il tipo\$="F" abilita il parametro param\$ ad ulteriori due stringhe ("DOWNLOAD" e "UPLOAD") che attivano il tasto funzione [F3] per navigare il file system con possibilità di drag&drop del file; entrambe queste due operazioni vanno completate con l'istruzione COPYFILE del file "da" o "verso" la destinazione desiderata. • "P" tipo directory abilita la navigazione delle directory del file sistem del server o del client se indicato il simbolo di reindirizzamento "<", nella variabile <str\$> ritorna il percorso selezionato.
str\$	Come l'istruzione INPUT, ovvero: Variabile dichiarata per l'input. Se <str\$> contiene già un valore questo è proposto come dato di ingresso modificabile (viene visualizzato il contenuto della variabile, il cursore si ferma sul primo carattere del campo, accetta e controlla l'immissione o la variazione del campo stesso; se è stringa e contiene un carattere ASCII 0, questo è considerato come la fine della stringa; se è array l'indice deve essere una costante, esempio INPUT "A",VAR\$(2),1. Se il parametro <tipo\$> = "ST*" il parametro <str\$> diventa una variabile numerica (con valori da 0 a 128) e attiva il tasto funzione [F2] monitor di scelta stampanti che sono configurate.
larg	Come l'istruzione INPUT. Indica il numero dei caratteri visibili dell'input
maxlunstr	Dimensione massima del campo di input
param\$	Parametri della funzione, se contiene più parametri vanno separati dal carattere ",", (in passato era usato lo spazio " "): <ul style="list-style-type: none"> - Tipo = "F" (Input di tipo FILE - nome file): <ul style="list-style-type: none"> o NOCANC disabilita la cancellazione dei file; o NOESEC disabilita la possibilità di eseguire i file; o DOWNLOAD di file (solo per Client WebDesk) o UPLOAD file (solo per Client WebDesk) - Tipo = "P" (Input di tipo PATH - nome directory): <ul style="list-style-type: none"> o CREADIR abilita la creazione delle Directory. o SOLOCLI abilita l'accesso al solo Client - Tipo = "ST2": <ul style="list-style-type: none"> o DISP = identifica i tipi di dispositivi che si vogliono utilizzare. <tipo\$> o FORM = Filtra la lista per i formati specificati o NODISP abilita la visualizzazione delle stampanti non disponibili. o NOESIST abilita la digitazione delle stampanti non esistenti. - Tipo = "W" oppure "WP" (v. funzione INPUT, dalla v711): <ul style="list-style-type: none"> o CONFERMA abilita la possibilità di confermare la Nuova Password;

Istruzione che abilita la INPUT a scorrimento in una determinata zona mouse con rilascio di un determinato codice (numero zonamouse). Consultare anche istruzioni SETFSM e SETZM. Nella INPUT_EXT sono gestite le seguenti variabili:

- o _TMZ → ritorna il numero della zonamouse;
- o _TMR → ritorna il numero riga della finestra (coordinata di riga);

- o `_TMC` → ritorna il numero colonna della finestra (coordinata di colonna);

La `INPUT_EXT` consente anche la gestione di file e percorsi. Esempio di download file per un Client WebDesk:

```
INPUT_EXT 0,"F",VAL$,50,255,"DOWNLOAD"
```

```
COPYFILE VAL$,"<~>nome_file_local.ext" (apre la finestra con il link "nome_file_local.ext" per eseguire il download del file)
```

Nota: il carattere `~` (tilde) consente di indicare il nome file del Client (senza percorso).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INPUT_EXT_EXT

`INPUT_EXT_EXT <numzonamouse>,<tipo$>,<str$>,<larg>,<maxlunstr>,<param$>,[elencoscelte$]`

Parametri	
numzonamouse	Come l'istruzione <code>INPUT_EXT</code> , ovvero: vale da 1-127 (0 = disabilita la zona mouse; -1 = disabilita tutte le zone mouse della finestra); il parametro <code>param\$</code> ha valore 'null', pertanto va scritto come "".
tipo\$	Come l'istruzione <code>INPUT_EXT</code> Solo per <code>INPUT_EXT_EXT</code> sono disponibili le due seguenti tipologie: <ul style="list-style-type: none"> • "CHECK" tipo check box. • "COMBO" tipo combo box;
str\$	Come l'istruzione <code>INPUT_EXT</code>
larg	Come l'istruzione <code>INPUT_EXT</code> , ovvero: Indica il numero dei caratteri visibili dell'input
maxlunstr	Come l'istruzione <code>INPUT_EXT</code> , ovvero: Dimensione massima del campo di input
param\$	Come istruzione <code>INPUT_EXT</code> : Solo per <code>INPUT_EXT_EXT</code>, indipendentemente dal tipo, sono disponibili i seguenti: <ul style="list-style-type: none"> • SCORRI: assume il valore 1 per <code>_SCORRI</code> (<code>_SCORRI</code> non viene modificata) • NOSCORRI: assume il valore 0 per <code>_SCORRI</code> (<code>_SCORRI</code> non viene modificata) • OBBLIG: input obbligatorio (non fa uscire dal campo se è vuoto) • SEGNOBBLIG: input non obbligatorio ma visualizzato come obbligatorio (riquadro di rosso); ininfluente dalla nuova interfaccia a partire dalla versione v800, se utilizzato non produce effetti; • DISABIL: input disabilitato (l'esecuzione dell'istruzione termina immediatamente dopo aver visualizzato il campo) l'eventuale zona mouse viene disabilitata
elencoscelte\$	[parametro facoltativo]: elenco scelte per tipo di input "COMBO": array monodimensionale di stringhe alfanumeriche nella forma: <code><codice\$>=<descrizione\$></code> ; Il carattere pipe ' ' non è ammesso e verrà eliminato). Il <code><codice\$></code> è un valore di <u>un solo carattere</u> ; la quantità di caratteri disponibili è sufficiente per il combobox (altrimenti è più potente usare la <code>CREATELIST</code>). Esempio: "7=Matita", "b=penna", "!=Lapis", "G=biro" Per attivare la freccia elenco settare la finestra mouse (<code>SETFSM</code>), in caso di codici presenti nella <code>descrizione\$</code> (es. (I=Includi ed E=Escludi), dato che il parametro <code><larg></code> può essere inferiore/uguale/maggiore di <code><descrizione\$></code> , possono verificarsi dati in input/output non attesi (lo si nota quando si aggiungono, come <code>codice\$</code> , anche dei numeri o dei caratteri speciali non presenti nella riga descrittiva; nell'eventualità aggiungere un carattere blank " " ad inizio di <code><descrizione\$></code>).

Istruzione estesa dell'istruzione `INPUT_EXT`. In uscita vengono valorizzate le seguenti variabili (stesso significato di `INPUT_EXT`):
`_TF`, `_TMZ`, `_TMR`, `_TMC`

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INPUT_TXT

`INPUT_TXT <numzonamouse>,<txt$>,<numrigvid>,<numcolvid>,<maxrig>,<maxcol>,<param$>`

Parametri	
numzonamouse	Come l'istruzione <code>INPUT_EXT</code> , ovvero: vale da 1-127 (0 = disabilita la zona mouse; -1 = disabilita tutte le zone mouse della finestra); il parametro <code>param\$</code> ha valore 'null', pertanto va scritto come "".
txt\$	Righe di testo: array monodimensionale di stringhe alfanumeriche.
numrigvid	Numero di righe video (minimo 1 massimo larghezza finestra);
numcolvid	Numero di colonne video (minimo 3 massimo altezza finestra);
maxrig	Massimo numero di righe in input (minimo 1); se = 0 viene assunto il valore di <code><numrigvid></code> ; se l'array <code><txt\$></code> ha una dimensione inferiore viene assunto il valore della dimensione di <code><txt\$></code>
maxcol	Massimo numero di colonne in input (minimo 1 massimo 255); se = 0 viene assunto il valore di <code><numcolvid></code>
param\$	Parametri (separati da carattere ","):

	<ul style="list-style-type: none"> • SCORRI: assume il valore 1 per _SCORRI (_SCORRI non viene modificata) • NOSCORRI: assume il valore 0 per _SCORRI (_SCORRI non viene modificata) • OBBLIG: input obbligatorio • SEGNOBBLIG: input non obbligatorio ma visualizzato come obbligatorio; ininfluenza dalla nuova interfaccia a partire dalla versione v800, se utilizzato non produce effetti; • DISABIL: input disabilitato (l'esecuzione dell'istruzione termina immediatamente dopo aver visualizzato il campo) l'eventuale zona mouse viene disabilitata
--	--

Istruzione di Input multi-linee. In uscita vengono valorizzate le seguenti variabili (stesso significato di INPUT_EXT): _TF, _TMZ, _TMR, _TMC

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INSKEY

INSKEY <numfile>

Parametri	
numfile	Numero File Indici

Istruzione che nel file a indici aperto con numero <numfile> inserisce un record con i valori contenuti nelle variabili del buffer associato. In inserimento: se i valori delle variabili sono più lunghi dei buffer è scritta solo la parte non eccedente, diversamente, se inferiori, sono utilizzati ASCII 0 per l'eccedente; l'inserimento incrementa a multipli di 64 record e i record non utilizzati contengono tutti valore ASCII 0; se la chiave principale esiste già l'inserimento non è eseguito; tutte le chiavi sono obbligatorie e i loro valori non possono essere tutti ASCII 0 o ASCII 255. Se i campi contenenti chiavi sono variati i relativi file chiavi vengono riordinati. Più chiavi secondarie uguali del nuovo record vengono inserite in ordine di principale + secondaria. Fare attenzione a tutto il buffer (la dimensione della chiave deve mantenere la lunghezza). Valorizza _NRDATI e _ERRIND.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INSTR

INSTR(<str\$>,<ricstr\$>)

Parametri	
str\$	Stringa della ricerca;
ricstr\$	Stringa da cercare;

Funzione che cerca la stringa <ricercastr\$> nella stringa <str\$> ritornandone la prima posizione in cui la trova (se non c'è ritorna 0).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INSTR_CI

INSTR_CI(<str\$>,<ricstr\$>)

Parametri	
str\$	Stringa della ricerca;
ricstr\$	Stringa da cercare;

Funzione che cerca in modalità case insensitive (non distingue fra caratteri minuscoli e maiuscoli) la stringa <ricercastr\$> nella stringa <str\$> ritornandone la prima posizione in cui la trova (se non c'è ritorna 0).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

INKEY

INKEY()

Funzione che controlla la pressione del tasto da tastiera; ritorna zero in caso negativo oppure il valore ASCII (o del codice inserito in SETTF) del tasto premuto. L'utilizzo eccessivo della funzione impegna molto le risorse del sistema, soprattutto se accorpa altre istruzioni e funzioni impegnative, pertanto valutare l'opportunità di gestire SLEEP e _INTER=0 (_INTER=1 può disattivare la INKEY)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ITERARC

ITERARC <archivio\$>,<nomeiteratore\$>

Parametri	
archivio\$	Nome dell'archivio in iterazione;
nomeiteratore\$	Nome del iteratore;

Istruzione di richiamo dell'iteratore (SprixMobile). Dopo l'esecuzione dell'istruzione apparirà una progressbar non annullabile che indicherà il progresso dell'iteratore. La funzione gestisce le variabili di struttura _IA. E' possibile interromperne l'esecuzione lato utente, in questo caso arriveranno gli errori al programmatore nelle variabili _ERRIA , _ERRIA\$. Non è possibile eseguire contemporaneamente due o più iteratori.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONADDARR

JSONADDARR (<RifObj>,<Key\$>, <RifArrayDalns>)

Parametri	
RifObj	Riferimento ad un elemento del dato JSON
Key\$	Se <RifObj> è un JSON_OBJECT allora <Key\$> contiene il nome da dare all'elemento, altrimenti questo parametro sarà ignorato
RifArrayDalns	Riferimento ad un JSON_ARRAY già esistente, se NULL (zero) l'array viene inserito vuoto, altrimenti nell'array vengono inseriti tutti gli elementi presenti nell'array <RifArrayDalns>

Funzione per variare il valore degli elementi; aggiunge un array all'elemento <RifObj>. Ritorna:

- Il riferimento all'elemento aggiunto.
- NULL (zero) se l'operazione non riesce, la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONADDBOOL

JSONADDBOOL (<RifObj>,<Key\$>,<Bool>)

Parametri	
RifObj	Riferimento ad un elemento del dato JSON
Key\$	Se <RifObj> è un JSON_OBJECT allora <Key\$> contiene il nome da dare all'elemento, altrimenti questo parametro sarà ignorato
Bool	Valore Bool da inserire, se <Bool> è = zero viene inserito un valore JSON_FALSE mentre se <Bool> è <> da zero allora viene inserito un valore JSON_TRUE.

Funzione per variare il valore degli elementi; aggiunge un valore JSON_TRUE o JSON_FALSE all'elemento <RifObj>. Ritorna:

- Il riferimento all'elemento aggiunto.
- NULL (zero) se l'operazione non riesce, la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONADDFILE

JSONADDFILE (<RifObj>,<Key\$>,<NomeFileIn\$>)

Parametri	
RifObj	Riferimento ad un elemento del dato JSON
Key\$	Se <RifObj> è un JSON_OBJECT allora <Key\$> contiene il nome da dare all'elemento, altrimenti questo parametro sarà ignorato
NomeFileIn\$	Percorso/Nome del file da inserire come valore JSON_STRING nell'elemento <RifObj>

Funzione per variare il valore degli elementi. Aggiunge il contenuto di un file come JSON_STRING all'elemento <RifObj>, il file può anche essere binario in quanto prima dell'inserimento viene codificato come Base64 (in maniera speculare, quando verrà estratto sarà convertito da Base64 in forma binaria). Ritorna:

- Il riferimento all'elemento aggiunto.
- NULL (zero) se l'operazione non riesce, la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONADDNUM

JSONADDNUM (<RifObj>,<Key\$>,<ValNum>)

Parametri	
RifObj	Riferimento ad un elemento del dato JSON
Key\$	Se <RifObj> è un JSON_OBJECT allora <Key\$> contiene il nome da dare all'elemento, altrimenti questo parametro sarà ignorato
ValNum	Valore numerico intero da inserire nell'elemento <RifObj>

Funzione per variare il valore degli elementi. Aggiunge un valore JSON_NUMBER all'elemento <RiferimentoOggetto>. Ritorna:

- Il riferimento all'elemento aggiunto.
- NULL (zero) se l'operazione non riesce, la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONADDOBJ

JSONADDOBJ (<RifObj>,<Key\$>,<RifObjDals>)

Parametri	
RifObj	Riferimento ad un elemento del dato JSON
Key\$	Se <RifObj> è un JSON_OBJECT allora <Key\$> può contenere il nome da dare all'elemento, per ogni altro tipo di oggetto a cui fa riferimento <RifObj> questo parametro sarà ignorato
RifObjDals	Riferimento ad un JSON_OBJECT già esistente, se NULL (zero) l'Object viene inserito vuoto, altrimenti nell'Object vengono inseriti tutti gli elementi presenti nell'Object <RifObjDals>

<p>Funzione per variare il valore degli elementi. Aggiunge un valore JSON_OBJECT all'elemento <RifObj>. Ritorna:</p> <ul style="list-style-type: none"> Il riferimento all'elemento aggiunto. NULL (zero) se l'operazione non riesce, la variabile <code>_ERRJSON\$</code> conterrà il messaggio di errore. <p>Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>

JSONADDNULL

JSONADDNULL (<RifObj>,<Key\$>)

Parametri	
RifObj	Riferimento all'oggetto padre (numerico)
Key\$	Nome dell'oggetto che si desidera settare con NULL (stringa)

<p>Funzione consente di settare il valore NULL su un OBJECT JSON. Esempio:</p> <pre>JSON_ROOT = JSONLOAD("c:\temp\testjson.json") A = JSONADDNULL(JSON_ROOT, "ID") JSONDUMP JSON_ROOT, 3, "c:\temp\testjsonaddnull.json" JSONDESTROY(ROOTOBJ) END</pre> <p>Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>
--

JSONADDREAL

JSONADDREAL (<RifObj>,<Key\$>,<ValNumReal>)

Parametri	
RifObj	Riferimento ad un elemento del dato JSON
Key\$	Se <RifObj> è un JSON_OBJECT allora <Key\$> contiene il nome da dare all'elemento, altrimenti questo parametro sarà ignorato
ValNumReal	Valore numerico reale da inserire nell'elemento <RifObj>

<p>Funzione per variare il valore degli elementi. Aggiunge un valore JSON_REAL all'elemento <RifObj>. Ritorna:</p> <ul style="list-style-type: none"> Il riferimento all'elemento aggiunto. NULL (zero) se l'operazione non riesce, la variabile <code>_ERRJSON\$</code> conterrà il messaggio di errore. <p>Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>

JSONADDSTR

JSONADDSTR (<RifObj>,<Key\$>,<ValStr\$>)

Parametri	
RifObj	Riferimento ad un elemento del dato JSON
Key\$	Se <RifObj> è un JSON_OBJECT allora <Key\$> contiene il nome da dare all'elemento, altrimenti questo parametro sarà ignorato
ValStr\$	Valore stringa da inserire in <RifObj>, quando il valore dovesse superare la dimensione massima di una variabile SPRIX, si può passare questo parametro vuoto ("") e compilare invece gli elementi dell'array <code>SPRIX_JSONINPA\$()</code> , verranno tutti accodati in un unico valore, la scansione degli elementi si fermerà al primo elemento vuoto ("")

<p>Funzione per variare il valore degli elementi. Aggiunge un valore JSON_STRING all'elemento <RifObj>. Ritorna:</p> <ul style="list-style-type: none"> Il riferimento all'elemento aggiunto. NULL (zero) se l'operazione non riesce, la variabile <code>_ERRJSON\$</code> conterrà il messaggio di errore. <p>ATTENZIONE: è importante assegnare il valore vuoto almeno al primo elemento dell'array <code>_JSONINPA\$()</code> dopo averlo usato, altrimenti, la prima chiave che si intende scrivere vuota utilizzerà, invece, il contenuto dell'array; analogamente è importante, quando si utilizza l'array per fornire l'input alla funzione, azzerare l'elemento successivo all'ultimo utilizzato per evitare che venga trasferito anche parte del vecchio valore utilizzato.</p> <p>Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>
--

JSONARRSIZE

JSONARRSIZE <RifArray>

Parametri	
RifArray	Riferimento ad un elemento del dato JSON di tipo JSON_ARRAY

Funzione che ritorna il numero degli elementi inseriti nell'array <RifArray >, che deve necessariamente essere un riferimento ad un JSON_ARRAY. Ritorna:

- il numero di elementi di un array
- -1 se si verificano errori o il riferimento <RifArray> non si riferisce ad un JSON_ARRAY ed in questo caso la variabile `_ERRJSON$` conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONCREATE

JSONCREATE()

Funzione per creare elementi; crea un JSON_OBJECT vuoto, viene ritornato il riferimento ad esso. Ritorna:

- Il riferimento all'oggetto creato.
- NULL (zero) se l'operazione non riesce, la variabile `_ERRJSON$` conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONDELARR

JSONDELARR <RifObj>,<IdElem>

Parametri	
RifObj	Riferimento ad un oggetto di tipo JSON_ARRAY
IdElem	valori positivi 1 – n : cancella l'ennesimo elemento dell'array valore negativo -1 : cancella tutti gli elementi dell'array valore negativo -2 : cancella l'ultimo elemento dell'array

Istruzione che elimina uno o tutti gli elementi di un array

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONDELOBJ

JSONDELOBJ <RifObj>,<Key\$>

Parametri	
RifObj	Riferimento ad un oggetto di tipo JSON_OBJECT
Chiave\$	Valore della chiave da eliminare, se vuoto cancella tutte le chiavi

Istruzione che elimina l'elemento dell'oggetto con chiave = <Key\$>, oppure se <Key\$> è vuota ("") cancella tutte le chiavi dell'oggetto

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONDESTROY

JSONDESTROY <RifObj>

Parametri	
RifObj	Riferimento ad un elemento del dato JSON.

Istruzione che rilascia tutte le risorse (memoria) occupate da tutti gli elementi discendenti da quello di cui si fornisce il riferimento, se il riferimento è la radice di un dato JSON, allora tutto il dato viene rilasciato.

Va sempre chiamata, una volta terminato l'uso di un dato JSON, con il riferimento ottenuto dalla funzione JSONLOAD per rilasciare tutte le risorse (memoria) allocate per il documento. Va chiamata anche con il riferimento all'elemento radice di un dato JSON se il dato è stato creato direttamente in memoria.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONDUMP

JSONDUMP <RifObj>,<CharIdent>,<NomeFileOut\$>

Parametri	
RifObj	Riferimento ad un elemento del dato JSON

CharIdent	Se diverso da zero, vengono inseriti nel file prodotto un ritorno a capo fra il contenitore (array) ed i suoi elementi i quali vengono indentati di <CharIdent> caratteri
NomeFileOut\$	Percorso/nome del file su cui scrivere il dato JSON a partire dall'elemento individuato da <RifObj>.

Istruzione che scrive sul file <NomeFileOut\$> il dato JSON in memoria cui si riferisce <RifObj>.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONGETARR

JSONGETARR (<RifArray>,<IxElem>)

Parametri	
RifArray	Riferimento ad un elemento del dato JSON
IxElem	Indice dell'array di cui si vuole leggere il valore <IxElem> deve avere un valore da 0 (zero) a JSONARRSIZE-1 <RifArray>

Funzione di lettura valori. Acquisisce il riferimento all'elemento <IxElem> dell'array < RifArray >. Ritorna:

- Il riferimento all'elemento <IxElem> dell'array <RifArray>.
- NULL (zero) se l'operazione non riesce, la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONGETBOOL

JSONGETBOOL (<RifBoolObj>)

Parametri	
RifBoolObj	Riferimento ad un elemento del dato JSON, di tipo JSON_TRUE oppure JSON_FALSE

Funzione per lettura valori. Acquisisce il valore all'elemento del dato JSON <RifBoolObj>. Ritorna il valore dell'elemento <RifBoolObj>:

- 1 Se il valore è JSON_TRUE
- 0 Se il valore è JSON_FALSE
- -1 Se l'operazione non è riuscita e la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONGETFILE

JSONGETFILE <RifObj>,<Key\$>,<IxElem>,<NomeFileOut\$>

Parametri	
RifObj	Riferimento ad un elemento del dato JSON
Key\$	Se <RifObj> si riferisce ad un JSON_OBJECT, allora <Key\$> deve contenere la chiave da cui recuperare il valore, per qualunque altro tipo di elemento non viene considerato
IxElem	Se <RifObj> si riferisce ad un JSON_ARRAY, allora <IxElem> deve contenere l'indice dell'elemento da cui recuperare il Valore. Per qualunque altro tipo di elemento non viene considerato
NomeFileOut\$	Percorso/nome del file su cui scrivere il valore contenuto nell'elemento indicato.

Istruzione che legge un elemento del JSON (come fosse un file allegato) e quindi lo salva su disco codificato Base64; si ottiene su disco il contenuto di un file codificato in Base64 presente all'interno di un JSON. Nell'istruzione si indica quale elemento leggere e poi lo scrive sul file indicato. Restituisce eventuali errori in variabile _ERRJSON\$.

Il valore, codificato Base64 nell'elemento JSON, viene automaticamente convertito in binario al momento dell'estrazione; se, infatti, il valore è stato scritto codificato con Base64 (perchè in un file JSON non si possono scrivere valori binari, ma solo Testo con codifica UTF-8) di conseguenza lo decodifica automaticamente da Base64 alla sua forma originaria (per formato Testo si intende quello TXT, altri formati come Word, WordPad, altri Word Processor, sono file Binari), pertanto, nel JSON è visto codificato Base64 e conseguentemente aumentato di dimensioni.

La funzione richiede che il file allegato sia inserito all'interno di un oggetto matrice e che ogni singola stringa che compone la matrice non ecceda i 32765 caratteri (vedi esempio disponibile a questo [link](#)).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONGETNUM

JSONGETNUM (<RifNumObj>)

Parametri	
RifNumObj	Riferimento ad un elemento del dato JSON, di tipo JSON_NUMBER

Funzione per lettura valori. Acquisisce il valore all'elemento del dato JSON <RifNumObj>. Ritorna il valore dell'elemento <RifNumObj>, se l'operazione non riesce la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONGETOBJ

JSONGETOBJ (<RifObj>,<Key\$>,<IxElem>)

Parametri	
RifObj	Riferimento ad un elemento del dato JSON, di tipo JSON_OBJECT oppure JSON_ARRAY
Key\$	Nome dell'oggetto, da specificare SOLO se <RifObj> si riferisce ad un JSON_OBJECT, altrimenti vuoto ("")
IxElem	Indice dell'elemento, da specificare SOLO se <RifObj> si riferisce ad un JSON_ARRAY, altrimenti vuoto (0)

Funzione per lettura valori. Acquisisce il riferimento ad un elemento del dato JSON figlio di <RifObj> con chiave uguale <Key\$>. Ritorna:

- Il riferimento all'elemento cercato (<RifArray> e chiave = <Key\$>).
- NULL (zero) se l'operazione non riesce, la variabile _ERRJSON\$ conterrà il messaggio di errore

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONGETREAL

JSONGETREAL (<RifRealObj>)

Parametri	
RifRealObj	Riferimento ad un elemento del dato JSON, di tipo JSON_REAL

Funzione per lettura valori. Acquisisce il valore all'elemento del dato JSON <RifRealObj>. Ritorna: il valore dell'elemento <RifRealObj>, se l'operazione non riesce la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONGETSTR

JSONGETSTR (<RifStrObj>)

Parametri	
RifStrObj	Riferimento ad un elemento del dato JSON, di tipo JSON_STRING

Funzione per lettura valori. Acquisisce il valore riferito all'elemento del dato JSON <RifStrObj>.

Restituisce il valore letto se la lunghezza di questo è inferiore/uguale a 255 caratteri (MAX dimensione per variabile SPRIX), altrimenti, se la lunghezza è superiore a 255 caratteri il valore viene suddiviso sugli elementi dell'array _JSONOUTA\$, il primo elemento vuoto ("") identifica la fine del valore. Al ritorno della funzione, la variabile _JSONSTRLEN conterrà la lunghezza totale del valore. Per conoscere la lunghezza totale del dato occorre leggere tutti gli elementi dell'array e se il contenuto sia completamente contenuto nel valore ritornato dalla funzione. Se si verificano errori il valore della variabile _ERRJSON\$ non è vuoto (""), in questo caso sia il valore ritornato che il primo elemento dell'array _JSONOUTA\$ saranno vuoti.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONGETSTRF

JSONGETSTRF (<RifStrObj>,<nomefile>)

Parametri	
RifStrObj	Riferimento ad un elemento del dato JSON, di tipo JSON_STRING
nomefile	Percorso / Nome del file in cui viene scritto tutto il valore del nodo referenziato

Estensione della funzione JSONGETSTR per la lettura valori, ma con la scrittura diretta su file, ovvero acquisisce il valore riferito all'elemento del dato JSON <RifStrObj> e lo scrive direttamente sul file indicato <NomeFile>. La lettura con JSONGETSTR, invece, è restituita su array nel quale viene suddivisa a blocchi di 256 byte stante la dimensione massima di una variabile SPRIX, pertanto, la scrittura del valore su file risulta più lenta). Ritorna: 0 (zero) se l'operazione è terminata correttamente, altrimenti un valore diverso da 0 (zero). Se si verificano errori il valore della variabile _ERRJSON\$ ne conterrà una descrizione. Esempio CALL

```
JSON_OBJECT = 0
JSON_ARRAY = 1
JSON_STRING = 2
JSON_INTEGER = 3
JSON_REAL = 4
JSON_TRUE = 5
JSON_FALSE = 6
JSON_NULL = 7
```

```
JSON_ROOT = JSONLOAD("]Check_JSSTR2F.json")
IF JSON_ROOT <= 0 THEN VIMSG 4, _ERRJSON$: END
JSTR = JSONGETOBJ( JSON_ROOT, "access_token", 0 )
IF JSTR <=0 THEN VIMSG 4, "Acquisizione REF Object, non riuscita": END
```

```
IF JSONTYPE (JSTR ) <> JSON_STRING THEN VIMSG 4, "Acquisizione REF Object di [access_token], il tipo dell'oggetto non e' quello
atteso" : END
RES = JSONGETSTRF( JSTR, "JCheck_JSSTR2F_acc_tk.txt" )
IF RES <> 0 THEN VIMSG 4, _ERRJSON$ : END
.....
END
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONLOAD

JSONLOAD (<NomeFile\$>)

Parametri	
NomeFile\$	File da cui è caricato il dato JSON

Funzione che carica un dato JSON dal file <Nomefile\$>, Ritorna l'Handle all'elemento radice del documento. Se il caricamento del documento non riesce ritorna 0 e la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONITERKEY

JSONITERKEY <Hndlter>

Parametri	
Hndlter	Handle di un iteratore, ritornato da una call alla funzione JSONSTARTITR

Iteratore per JSON_OBJECT. Funzione che ritorna il valore della chiave dell'elemento su cui è attualmente posizionato l'iteratore con handle <Hndlter>. Ritorna la chiave dell'oggetto su cui l'iteratore <Hndlter> è attualmente posizionato. Non è gestita una chiave più lunga di 255 caratteri (Max dimensione per una variabile SPRIX). Se si verificano errori viene ritornato un valore vuoto ("") e la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONITERNXT

JSONITERNXT <Hndlter>

Parametri	
Hndlter	Handle di un iteratore, ritornato da una call alla funzione JSONSTARTITR

Iteratore per JSON_OBJECT. Funzione che sposta l'iteratore sull'elemento successivo. Ritorna valore:

- = 0 se l'operazione di avanzamento dell'iteratore sul prossimo elemento è stata eseguita correttamente.
- < 0 se l'operazione non è riuscita a causa di errori relativi a operazioni su dati JSON (il messaggio di errore è contenuto nella variabile _ERRJSON\$);
- > 0 se l'iteratore si trova dopo l'ultimo elemento (tutti gli elementi sono già stati letti ed è già raggiunto l'ultimo elemento)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONITERSET

JSON <Hndlter>,<RifObj>

Parametri	
Hndlter	Handle di un iteratore, ritornato da una call alla funzione JSONSTARTITR
RifObj	Oggetto di riferimento a cui punta l'iteratore

Iteratore per JSON_OBJECT. Funzione che cambia il valore dell'elemento attualmente puntato dall'iteratore <Hndlter>, con l'oggetto con handle <RifObj>. Ritorna:

- 0 se la funzione è stata eseguita correttamente
- -1 se i parametri non sono corretti
- -2 se l'oggetto a cui punta l'iteratore <Hndlter> punta attualmente non è di tipo JSON_OBJECT

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONINTERVAL

JSON <Hndlter>

Parametri	
Hndlter	Handle di un iteratore, ritornato da una call alla funzione JSONSTARTITR

Iteratore per JSON_OBJECT. Funzione che ritorna il riferimento all'oggetto JSON_OBJECT su cui è attualmente posizionato l'iteratore con handle <Hndlter>. Ritorna il valore dell'oggetto su cui l'iteratore <Hndlter> è attualmente posizionato. Se si verificano errori, viene ritornato un valore vuoto ("") e la variabile _ERRJSON\$ conterrà il messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONSTARTITR

JSONSTARTITR <RifObj>,<Key\$>

Parametri	
RifObj	Riferimento ad un elemento del dato JSON, di tipo JSON_OBJECT
Key\$	Chiave su cui si vuole posizionare l'iteratore, se vuota ("") si posiziona sul primo elemento; in ogni caso, proseguiranno le richieste di posizionarsi sul successivo elemento fino alla fine degli elementi, e senza che sia garantito alcun ordinamento.

Iteratore per JSON_OBJECT. Avvia un iteratore per l'elemento di tipo JSON_OBJECT <RifObj>, per ogni JSONSTARTITR dovrà essere chiamata, quando non sarà più utilizzato l'iteratore, la complementare JSONSTOPITR affinché le risorse usate dall'iteratore vengano rilasciate. Ritorna:

- L'handle dell'iteratore creato per <RifArray>
- NULL (zero) se l'operazione non riesce, la variabile _ERRJSON\$ conterrà in questo caso un messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONSTOPITR

JSONSTOPITR <Hndlter>

Parametri	
Hndlter	Handle di un iteratore, ritornato da una call alla funzione JSONSTARTITR

Iteratore per JSON_OBJECT. Termina un iteratore con Handle <Hndlter>, valore precedentemente ritornato dalla complementare funzione JSONSTARTITR. Questa funzione si occuperà di rilasciare tutte le risorse allocate dall'iteratore con l'handle specificato. Ritorna:

- 0 Se l'operazione è terminata correttamente
- -1 Se l'iteratore indicato appare non avviato in questo caso la variabile _ERRJSON\$ conterrà un messaggio di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

JSONTYPE

JSONTYPE (<RifObj>)

Parametri	
RifObj	Riferimento all'oggetto

Funzione che ritorna il tipo di oggetto a cui si riferisce <RifObj>. I tipi di oggetto sono :

```

JSON_OBJECT = 0
JSON_ARRAY  = 1
JSON_STRING = 2
JSON_INTEGER = 3
JSON_REAL   = 4
JSON_TRUE   = 5
JSON_FALSE  = 6
JSON_NULL   = 7

```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

KILL

KILL <nomefile\$>

Parametri	
nomefile\$	File da cancellare;

Istruzione che cancella il file dal nome <nomefile\$>. Il file deve esistere, essere chiuso e può essere eventualmente preceduto dalla directory. Per ulteriori gestioni di file consultare INPUT_EXT (tipo F/P) e altri comandi raggruppati per argomento "DATABASE – FILE – COM – SHELL"

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LEFT

LEFT\$(<str\$>,<numcar>)

Parametri	
str\$	Stringa da cui estrarre caratteri;
numcar	Quantità di caratteri da estrarre;

Funzione che estrae i primi caratteri <numcar> a partire da sinistra della stringa <str\$> (se vuota ritorna ""); se la lunghezza di <numcar> è superiore di <str\$> ritorna l'intera <str\$>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LEGGIDESC

LEGGIDESC <par>

Parametri	
par	3 = vengono lette le righe di tipo testo e descrittive del corpo dei documenti di magazzino nell'esecuzione del ciclo tabellare; 2 = vengono lette le righe di tipo testo del corpo dei documenti di magazzino nell'esecuzione del ciclo tabellare; 1 = vengono lette le righe descrittive del corpo dei documenti di magazzino nell'esecuzione del ciclo tabellare; 0 = disabilitato; non vengono lette le righe descrittive e testo del corpo dei documenti di magazzino nell'esecuzione del ciclo tabellare;

Istruzione che abilita o disabilita la lettura delle righe descrittive del corpo dei documenti di magazzino nell'esecuzione del ciclo tabellare. Il comando deve essere impostato prima della definizione del ciclo tabellare e nelle operazioni di mirroring (ASCMIRROR, DBMIRROR)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LEGGIDIR

LEGGIDIR <directory\$> o <nomecartella\$>,[maschera\$]

Parametri	
directory\$ - oppure - nomecartella\$	Percorso di ricerca accessibile: <ul style="list-style-type: none"> Esempio sul Server: LEGGIDIR "J", "*.txt" Esempio sul Client in Locale: LEGGIDIR "<c:\temp\","*.txt" oppure Nome della cartella dati (senza estensione) per l'elemento App PassBuilder di tipo "Cartella dati". (dalla v711). Esempio nome cartella dati "carte.bindati": LEGGIDIR "carte\$","". Per accedere alle cartelle dati, utilizzare la sintassi NomeCartella\$NomeFile.est (Il percorso viene risolto con "\ " finale dopo il nome cartella) esempio fileapp\$\immagine.jpg Per ulteriori approfondimenti si rimanda al manuale PassBuilder.
maschera\$	Filtro di ricerca del file. Il metacarattere asterisco (*) è carattere jolly. Esempi di ricerca solo su file di tipo "testo": "*.txt", "abc*.txt", "**abc123*.txt". Esempio di ricerca su tutti i tipi di file: "**_xyz_**" Non sono consentite sintassi combinate di estensioni (esempio file.est1.est2.est3). Il DOS consente forme combinate di file (es. DIR c:\tmp*.csv c:\tmp*.png) in modo da ottenere come risultato più cicli separati del comando DIR. La LEGGIDIR, invece, non consente questo, ma è replicabile con più cicli di leggidir per quante estensioni di file si cercano. Esempio, ricerca dei file "*.png" e "*.csv" DIM CSV\$() : DIM PNG\$() LEGGIDIR "C:\Tmpl\","*.csv" : FOR CSV=1 TO _NUMDIRFILE : CSV\$(CSV)=_DIRFILE\$(CSV) : NEXT LEGGIDIR "C:\Tmpl\","*.png" : FOR PNG=1 TO _NUMDIRFILE : PNG\$(PNG)=_DIRFILE\$(PNG) : NEXT In alternativa, fare una leggidir generale (e/o con un carattere particolare nel "nomefile"), poi interrogare i contenuti delle array (INSTR).

Istruzione che ritorna l'elenco di file contenuti nel percorso <directory\$> oppure nel <nomecartella\$> dati dell'App, tramite l'eventuale filtro di ricerca file [maschera\$]. Valorizza le variabili _ERRDIR\$ (eventuale errore su directory), _NUMDIRFILE (numero di file nella directory), _DIRFILE\$(I) (array stringa coi nomi file nella directory). Per ulteriori gestioni di file consultare INPUT_EXT (tipo F/P) e altri comandi raggruppati per argomento "DATABASE – FILE – COM – SHELL".

Esempio:

```
LEGGIDIR "J", "*.txt" oppure LEGGIDIR "<c:\temp\","*.txt"
IF _ERRDIR$ <> "" THEN VIMSG 4,_ERRDIR$
VIMSG 4, STR$( _NUMDIRFILE,0,0)
FOR I = 1 TO _NUMDIRFILE : VIMSG 4, _DIRFILE$(I)
```


NEXT

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LIVLICAPP

LIVLICAPP()

Funzione che ritorna il numero di livello (Codice) di licenza acquistato sul PassStore dell'App su cui tale funzione è eseguita.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LEN

LEN(<str\$>)

Parametri

str\$	Stringa di cui si vuole conoscere la lunghezza;
-------	---

Funzione che ritorna la lunghezza della stringa <str\$>; se è vuota oppure non esiste ritorna valore zero;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LOCATE

LOCATE <rig>,<col>

Parametri

rig	Coordinata di riga; nella finestra base (la numero 0) la <rig> va da 1 → 20 (riga fisica 4 → 23); nelle altre finestre (1-9) la <rig> va da 1 alla lunghezza della finestra -2.
Col	Coordinata di colonna; nella finestra base (la numero 0) la <col> va da 1 → 78/94 (colonna fisica 2 → 79); nelle altre finestre (1-9) la <col> va da 1 alla larghezza della finestra -2.

Istruzione che posiziona il cursore alle coordinate di riga <rig> e colonna <col> nella finestra attiva

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LOF

LOF(<numfile>)

Parametri

numfile	Numero file di cui si vuole conoscere la lunghezza;
---------	---

Funzione che ritorna la lunghezza (quantità caratteri) del file numero <numfile>. Ritorna 0 se il file è vuoto, ritorna -1 se non è stato aperto.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LOFKEY

LOFKEY(<numfile>)

Parametri

numfile	Numero file ad indici di cui si vuole conoscere la quantità di chiavi;
---------	--

Funzione che ritorna la quantità di chiavi (quindi di records) inserite nel file ad indici numero <numfile>. Ritorna 0 se il file è vuoto, ritorna -1 se non è stato aperto oppure il file non è ad indici.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINT

LPRINT <espr\$>[:;]

Parametri

espr\$	Espressione stringa da inviare alla stampante aperta;
;	Annulla il salto riga; se questo parametro è assente, dopo la stampa di <espr\$> effettua un salto riga;

Istruzione che invia l'espressione <espr\$> alla stampante aperta; le caratteristiche di stampa dipendono dalla configurazione, impostazione e caratteri del dispositivo hardware. Se utilizzata in concomitanza (prima o dopo) di altre istruzioni di stampa (es. barcode/immagine) dotate di parametri di posizionamento, si possono generare sovrapposizioni (poichè indipendenti). Esempio: A\$ = "X_1" : B\$ = "X_2" : C\$ =

"X_4" : LPRINT A\$ + B\$ + "X_3"; : LPRINT C\$

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINTBARCODE

LPRINTBARCODE <codice\$>,<tipo>,<prncod>,<um>,<x>,<y>,<cx>,<cy>

Parametri	
codice\$	Codice articolo relativo al codice a barre (con ultimo carattere 'check digit' corretto);
tipo	Tipo di codice (0 = CODE 39 pieno; 1 = EAN8/13 pieno; 2 = CODE 39 standard; 3 = EAN8/13standard; 4 = CODE128 pieno; 5 = CODE 128 compresso); per il codice 3 utilizzare la stringa codice con 8 o 13 caratteri numerici di lunghezza;
prncod	Parametro di stampa del codice: 1 = sì, 0 = no;
um	Unità di misura: 0 = millimetri, 1 = caratteri;
x	Distanza in <um> dal bordo sinistro del foglio;
y	Distanza in <um> dal bordo superiore del foglio;
cx	Larghezza in <um> del barcode, con proporzione ottimale con <cy> di 3 a 2
cy	Altezza in <um> del barcode, con proporzione ottimale con <cx> di 2 a 3

Istruzione che stampa su stampanti configurate di tipo 'W' o 'F' la stringa codice a barre <codice\$> di tipo <tipo> con stampa del codice articolo in base <prncod> e l'unità di misura <um> da utilizzare nei parametri numerici <x>/<y> (distanza dal bordo sinistro/superiore del foglio) e nei parametri num. <cx>/<cy>. Esempio: LPRINTBARCODE arcod(),1,0,0,15,80,30,20

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per la generazione di QR CODE utilizzare l'apposita istruzione LPRINTQRCODE.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINTGRAF_NUM

LPRINTGRAF_NUM <nomediz\$>,<indicediz>,<num>

Parametri	
<nomediz\$>	Nome vettore variabile di dizionario di stampa. Il dizionario di stampa prevede per ogni sezione un vettore di variabili di tipo numerico e un vettore di variabili di tipo stringa entrambe di 99 elementi: _inteststr vettore variabili stringa intestazione (disponibili in tutte le sezioni) _intestnum vettore variabili numeriche intestazione (disponibili in tutte le sezioni) _corpostr vettore variabili stringa corpo (disponibili nella sezione di corpo e sezioni dettaglio) _corponum vettore variabili numeriche corpo (disponibili nella sezione di corpo e sezioni dettaglio) _det1str vettore variabili stringa dettaglio 1 (disponibili nella sezione dettaglio 1) _det1num vettore variabili numeriche dettaglio 1 (disponibili nella sezione dettaglio 1) _det2str vettore variabili stringa dettaglio 2 (disponibili nella sezione dettaglio 2) _det2num vettore variabili numeriche dettaglio 2 (disponibili nella sezione dettaglio 2) _det3str vettore variabili stringa dettaglio 3 (disponibili nella sezione dettaglio 3) _det3num vettore variabili numeriche dettaglio 3 (disponibili nella sezione dettaglio 3) _piedestr vettore variabili stringa piede (disponibili nella sezione fine stampa) _piedenum vettore variabili numeriche piede (disponibili nella sezione fine stampa)
<indicediz>	Indice vettore variabile di dizionario di stampa
<num>	Valore da assegnare

Istruzione per valorizzare le variabili di dizionario numeriche nelle stampe modulo grafico Sprix in App PassBuilder utilizzabili tramite le istruzioni APRISTAMP_EXT e APRISTAMP_EMAIL. Esempi

LPRINTGRAF_NUM "_corponum", 1, 1
 LPRINTGRAF_NUM "_corponum", 1, 2

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINTGRAF_SEZ

LPRINTGRAF_SEZ <tiposez>,<dovedettaglio>

Parametri	
<tiposez>	Tipo di sezione. Valori accettati 0 = Corpo 1 = Dettaglio 1 2 = Dettaglio 2 3 = Dettaglio 3
<dovedettaglio>	Parametro per indicare dove collocare il dettaglio; parametro utilizzato solo per sezioni dettaglio (il primo parametro vale

	1, 2 o 3). Valori accettati: 1 = Stampa il dettaglio dopo l'ultimo corpo stampato 2 = Ordina il dettaglio con i corpi in caso di modulo con ordinamento 3 = Stampa il dettaglio prima di tutti i corpi 4 = Stampa il dettaglio dopo tutti i corpi
--	---

Istruzione per scegliere la sezione da stampare nel modulo grafico Sprix in App PassBuilder utilizzabile tramite le istruzioni APRISTAMP_EXT e APRISTAMP_EMAIL.

Si ricorda che non è necessario comandare la stampa delle sezioni Intestazione e Piede, in quanto stampate in automatico rispettivamente prima del primo corpo o dettaglio stampato e dopo l'ultimo corpo o dettaglio stampato.

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINTGRAF_STR

LPRINTGRAF_STR <nomediz\$>,<indicediz>,<str\$>

Parametri	
<nomediz\$>	Nome vettore variabile di dizionario di stampa. Il dizionario di stampa prevede per ogni sezione un vettore di variabili di tipo numerico e un vettore di variabili di tipo stringa entrambe di 99 elementi: _inteststr vettore variabili stringa intestazione (disponibili in tutte le sezioni) _intestnum vettore variabili numeriche intestazione (disponibili in tutte le sezioni) _corpostr vettore variabili stringa corpo (disponibili nella sezione di corpo e sezioni dettaglio) _corponum vettore variabili numeriche corpo (disponibili nella sezione di corpo e sezioni dettaglio) _det1str vettore variabili stringa dettaglio 1 (disponibili nella sezione dettaglio 1) _det1num vettore variabili numeriche dettaglio 1 (disponibili nella sezione dettaglio 1) _det2str vettore variabili stringa dettaglio 2 (disponibili nella sezione dettaglio 2) _det2num vettore variabili numeriche dettaglio 2 (disponibili nella sezione dettaglio 2) _det3str vettore variabili stringa dettaglio 3 (disponibili nella sezione dettaglio 3) _det3num vettore variabili numeriche dettaglio 3 (disponibili nella sezione dettaglio 3) _pedestr vettore variabili stringa piede (disponibili nella sezione fine stampa) _pedenum vettore variabili numeriche piede (disponibili nella sezione fine stampa)
<indicediz>	Indice vettore variabile di dizionario di stampa
<str\$>	Valore da assegnare

Istruzione per valorizzare le variabili di dizionario alfanumeriche nelle stampe modulo grafico Sprix in App PassBuilder utilizzabili tramite le istruzioni APRISTAMP_EXT e APRISTAMP_EMAIL. Esempi

LPRINTGRAF_STR "_corpostr", 1, "Stringa corpo 1"

LPRINTGRAF_STR "_corpostr", 2, "Stringa corpo 2"

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINTIMG

LPRINTIMG <nomefile\$>,<um>,<x>,<y>,<cx>,<cy>

Parametri	
nomefile\$	Nome del file immagine; formati ammessi: bmp, wmf, jpg, png;
um	Unità di misura (0 = millimetri, 1 = caratteri) da utilizzare nei parametri numerici <x>/<y>/<cx>/<cy>;
x	Distanza dal bordo sinistro del foglio;
y	Distanza dal bordo superiore del foglio;
cx	Larghezza dell'immagine; se cx e cy sono entrambi = 0 significa una grandezza reale;
cy	Altezza dell'immagine; se cx e cy sono entrambi = 0 significa una grandezza reale;

Istruzione che stampa su stampanti configurate di tipo W, F, E e PDF, l'immagine dal nome <nomefile\$> con unità di misura <um> da utilizzare nei parametri numerici <x>/<y> e <cx>/<cy>. Esempio: IMAGE\$="c:\temp\immagine.bmp": LPRINTIMG IMAGE\$,0,20,20,0,0.

Il parametro <nomefile\$> deve prevedere un percorso file accessibile e/o richiesto dalla configurazione del tipo di stampante (esempio in cartella ..\datiutente), altrimenti può essere rilasciato il messaggio: errore file immagine non trovato. Dalla versione v64.3 può accedere alle cartelle dati di PassBuilder con la sintassi cartella\$nomefile.

Per accedere alle cartelle dati, utilizzare la sintassi NomeCartella\$NomeFile.est (Il percorso viene risolto con "\" finale dopo il nome cartella) esempio fileapp\$immagine.jpg. Per ulteriori approfondimenti si rimanda al manuale PassBuilder.

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINTLINE

LPRINTLINE <um>,<x>,<y>,<lung>,<spessore>,[opz\$]

Parametri	
um	Unità di misura: 0 = millimetri, 1 = caratteri
x	Distanza dal bordo sinistro del foglio (in millimetri o caratteri)
y	Distanza dal bordo superiore del foglio (in millimetri o caratteri)
lung	Lunghezza linea (in millimetri o caratteri)
spessore	Spessore linea: 0 = sottile, 1 = medio, 2 = spesso
opz\$	Parametro opzionale per indicare il colore della linea nel formato "C000000000"; Esempio "C255000000" = rosso, "C255255255" = bianco, "C000000000" = nero (default);

Istruzione che stampa una linea personalizzabile.

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINTQRCODE

LPRINTQRCODE <stringa\$>,<tipo>,<lunghezza_codice>,<um>,<x>,<y>,<lunghezza_dato>,<livello_errore>

Parametri	
stringa\$	Stringa da stampare;
tipo	Tipo di codice (0 = automatico; 1 = alfanumerico (numeri e lettere solo maiuscole); 2 = numerico; 3 = binario (ogni tipo di carattere));
lunghezza_codice	Lunghezza di stringa\$ (obbligatorio quando tipo = 3, facoltativo negli altri casi);
um	Unità di misura: 0 = millimetri, 1 = caratteri;
x	Distanza in <um> dal bordo sinistro del foglio;
y	Distanza in <um> dal bordo superiore del foglio;
lunghezza_lato	Lunghezza in <um> di un lato del qrcode;
livello_errore	Livello di controllo errore (0 = basso, 1 = medio, 2 = alto, 3 = massimo);

Istruzione che stampa su stampanti configurate di tipo 'W' o 'F' la stringa qrcode in <stringa\$> di tipo <tipo> con unità di misura <um> da utilizzare nei parametri numerici <x>/<y> (distanza dal bordo sinistro/superiore del foglio) e lunghezza lato pari a <lunghezza_lato>. Per stampare buffer superiori ai 255 bytes occorre utilizzare il vettore _QRCODECOD(\$) e non valorizzare il parametro stringa\$ nella chiamata alla funzione. Le dimensioni massime di caratteri stampabili sono: tipo 1 = 4296, tipo 2 = 7089, tipo 3 = 2953.

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

L'istruzione richiede MDS attivo sull'installazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LPRINTTXT

LPRINTTXT <str\$>,<font\$>,<dimfont>,<attrifont\$>,<um>,<x>,<y>,[opz\$]

Parametri	
str\$	Stringa da stampare;
font\$	Font (disponibile e identico a quello nella cartella Windows);
dimfont	Dimensione in numero di punti;
attrifont\$	Attributi: "G" = Grassetto, "I" = Corsivo, "S" = Sottolineato, "" = Normale;
um	Unità di misura: 0 = millimetri, 1 = caratteri;
x	Distanza in <um> dal bordo sinistro del foglio;
y	Distanza in <um> dal bordo superiore del foglio;
opz\$	Stringa facoltativa con i seguenti parametri combinabili: <ul style="list-style-type: none"> Allineamento testo: AS=sinistra (default), AC=centrato, AD=destra; Stretch testo sia in altezza (SY) che larghezza (SX), entrambi seguiti dalla percentuale di stretch da 000 a 999 (se assente il default è 100), es: per raddoppiare l'altezza SY200 o dimezzare la larghezza SX050; Colore RGB testo: parametro C seguito da un numero di 3 cifre tra 000 e 255, es. bianco = C255255255, nero = C000000000 (è il default);

Istruzione che stampa su stampanti configurate di tipo W, F e PDF la stringa <str\$> con un font dal nome <font\$>, dimensione in numero di punti <dimfont>, gli attributi<attrifont\$> combinabili, l'unità di misura <um> da utilizzare nei parametri numerici <x>/<y>. Esempio: LPRINTTXT "Prova di stampa","Courier New",10,"GIS",0,15,15,"AC C255000000".

Non eseguibile in Linux spool, avvalersi di LPRINT e comandi della stampante (vedere il manuale stampante); in alternativa verificare le possibilità con gli stili di stampa (es. INIZCO\$(, ecc) e LPRINT CHR\$(...)+CHR\$(...)+..., con file ".dfs" per ciascuna stampante.

L'istruzione è abilitata nell'esecuzione dell'elemento COLLAGE SERVER REMOTO nella App PassBuilder.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LSTLOC

LSTLOC <tipo\$>

Parametri	
tipo\$	Restituzione di tutte le variabili di tipo: "T" = temporanea, "P"=persistente;

Istruzione SprixMobile relativa alla restituzione della lista dei parametri salvati. La variabile di riferimento relativa a messaggi di errore è _ERRLOC\$. Restituisce 2 array: _LSTLOC\$(): array con tutte le chiavi, _LSTLOCTIPO\$(): array con il tipo di dato ("S","N","A")

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

LUNGMOD

LUNGMOD\$(<numrighe>)

Parametri	
numrighe	Numero di righe lunghezza modulo;

Funzione che ritorna una stringa di caratteri da inviare alla stampante per impostare la lunghezza del modulo di stampa con un numero di righe pari a <numrighe>. La lunghezza del modulo di carta è in numero di righe, in 6' di pollice. Esempio: LPRINT LUNGMOD\$(66); (modulo impostato lungo 66 righe).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MAXCOLLAGE

MAXCOLLAGE()

Funzione che ritorna la quantità di Collage caricati in memoria nel momento.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MEXALDB

MEXALDB <p1\$>,<p2\$>[,d\$]

Parametri	
p1\$	Archivi Generali: "S" = si, "N" = no;
p2\$	Anno di riferimento: "C" = anno corrente, "T" tutti gli anni, "N" = nessuno;
d\$	Opzione facoltativa di esportazione per differenza, cioè "solo i variati": "SV=S", "SV=N"; riguarda solo i documenti di magazzino e primanota con attivo l'aggiornamento replica dati in chiaro "M" manuale.

Istruzione che esporta i dati sul database MEXALDB in chiaro relativi agli archivi Aziendali e Generali. Se eseguita dall'azienda 'STD' avviene per tutte le aziende. Esempio MEXALDB "S","T","SV=S"

L'istruzione valorizza le variabili _ERRMEXALDB e _ERRMEXALDB\$. Nell'esecuzione mxrsbatch.exe gli eventuali errori a video sono indicati nel file degli errori stabilito dall'utente (consultare le Nozioni Generali del Manuale Generale dell'utente).

Notare: il database MEXALDB deve essere attivato e configurato.

Si ricorda che la struttura dei dati del Gestionale è consultabile nei seguenti modi:

- Le Tabelle e Colonne sono definite nel documento "ElencoColonneMexalDB.Xls" allegato al chiarimento soluzione n. 30663 del servizio di assistenza EasyPass accessibile dall'Area Riservata di www.Passepartout.net. I contenuti delle colonne nel file vanno aperte ed estese. Per ulteriori approfondimenti consultare il capitolo MexalDB del Manuale Utente
- I Tracciati CSV sono pubblicati nel chiarimento n. 54452 del supporto on-line. I Tracciati CSV sono verificabili nel menu del Gestionale [Servizi - Trasferimento Archivi - Export/Import CSV - ASCII]
- Specifiche dei campi Dizionario nel menu del Gestionale [Servizi - Personalizzazioni - Stampa Dizionario Sprix]

I dati obbligatori sono indicati a video dal Gestionale e le relazioni fra i dati sono descritte nel Manuale Utente

Dalla versione 2020A (v750) la licenza MDS consente la replica su SQLite in chiaro.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MID

MID\$(<str\$>,<posiniz>,<numcar>)

Parametri	
-----------	--

str\$	Stringa da cui estrarre caratteri;
posiniz	Posizione iniziale sinistra dell'estrazione;
numcar	Quantità di caratteri da estrarre; non può essere 0 e se è superiore alla lunghezza <str\$> la parte eccedente è ignorata;

Funzione che estrae i primi caratteri <numcar> dalla stringa <str\$> partendo dalla posizione iniziale <posiniz> dalla sinistra di <str\$>; <posiniz> deve essere compreso fra 1 e la dimensione massima di <str\$>. La funzione è utile anche per ottenere ordinamenti non previsti dal ciclotabellare (ad esempio in presenza di valori numerici).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MKA

MKA\$(<str\$>,<numbyte>)

Parametri	
str\$	Stringa da trattare;
numbyte	Lunghezza totale da raggiungere;

Funzione che aggiunge tanti caratteri ASCII = 0 in fondo a stringa <str\$> fino a raggiungere la lunghezza totale di <numbyte>. Se <str\$> è più lunga di <numbyte> viene troncata.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MKN

MKN\$(<num>,<numbyte>,<numdec>)

Parametri																						
num	Numero compreso tra $2^{\text{numbyte} \cdot 8 \text{bit} / 2}$ e $2^{\text{numbyte} \cdot 8 \text{bit} / 2 - 1}$																					
numbyte	Lunghezza della stringa convertita compresa tra 1 e 5 e determina il limite massimo del valore da convertire:																					
<table border="1"> <thead> <tr> <th colspan="3">TABELLA sviluppo cifre binarie per Num. Bytes</th> </tr> <tr> <th>Num. Bytes</th> <th>da</th> <th>a</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>-128</td> <td>127</td> </tr> <tr> <td>2</td> <td>-32.768</td> <td>32.767</td> </tr> <tr> <td>3</td> <td>-8.388.608</td> <td>8.388.607</td> </tr> <tr> <td>4</td> <td>-2.147.483.648</td> <td>2.147.483.647</td> </tr> <tr> <td>5</td> <td>-549.755.813.888</td> <td>549.755.813.887</td> </tr> </tbody> </table>		TABELLA sviluppo cifre binarie per Num. Bytes			Num. Bytes	da	a	1	-128	127	2	-32.768	32.767	3	-8.388.608	8.388.607	4	-2.147.483.648	2.147.483.647	5	-549.755.813.888	549.755.813.887
TABELLA sviluppo cifre binarie per Num. Bytes																						
Num. Bytes	da	a																				
1	-128	127																				
2	-32.768	32.767																				
3	-8.388.608	8.388.607																				
4	-2.147.483.648	2.147.483.647																				
5	-549.755.813.888	549.755.813.887																				
numdec	Numero di decimali (tra 0 e 6 e per ogni decimale il limite va diviso per dieci)																					

Funzione che converte il numero <num> in una stringa attraverso la codifica binaria. Impacca il <num> in meno byte rispetto alla funzione STR\$; la lunghezza della stringa convertita <numbyte> con numero di decimali <numdec>; la stringa non è interpretabile direttamente e per riconvertirla in <num> occorre utilizzare la funzione CVN.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBDEL

MYDBDEL <Archivio\$>,[Parametri\$]

Parametri	
Archivio\$	Archivio su cui effettuare la cancellazione. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi _MYDB.
Parametri\$	"DOC=" Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (La sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio). "TOTALE" Permette la cancellazione dell'intero archivio.

Istruzione che esegue la cancellazione di un'anagrafica con ID uguale a _MYDBID. Valorizza le variabili _ERRMYDB e _ERRMYDB\$ che specificano il risultato dell'operazione.

La successiva riorganizzazione archivi, ottimizzerà la dimensione del file su disco.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBGET

MYDBGET <Archivio\$>,<NumeroKey>,<Numero ID>,[Parametri\$]

Parametri	
Archivio\$	Archivio su cui effettuare la lettura. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi _MYDB.
NumeroKey	= 0 esegue la lettura per ID >= 1 ordina ed esegue la lettura in base alle chiavi 1, 2 o 3 specificate sulla definizione. La lettura parte dal valore

	impostato nell'array <code>_MYDBK\$(1)</code> = 99 esegue ordinamento e lettura per estensione, se dichiarata, ricercando la chiave impostata nell'array <code>_MYDBK\$(1)</code>
Numero ID	Se NumeroKey = -1 legge l'ultimo record in archivio con il n. chiave specificato (non considera il valore in <code>MYDBK\$(1)</code>) Se NumeroKey = 0 oppure <> 0, allora ID è pari al ID impostato di cui leggere il record;
Parametri\$	valori ammessi: "DOC= ... ,NODEF" "DOC=" Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (la sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio). "NODEF" per non valorizzare con dei dati iniziali quando vengono recuperati dati che, nonostante non siano valorizzati, hanno un valore iniziale definito. Se omissso "NODEF", la MYDBGGET recupera i valori dei campi con il loro valore iniziale, anche quando risultano vuoti. Esempio: "DOC=OC,NODEF"

Istruzione che legge i dati dell'anagrafica dell'archivio impostato, per il valore della chiave o ID impostato. Ritorna i dati dell'anagrafica ricercata, o della successiva se non trovata. Imposta le variabili `_MYDBELE`, `_MYDB$()`, `_MYDBANN$`, `_MYDBETIC$()`, `_MYDBCOD$()`, `_MYDBID` e `_MYDBNR`. Valorizza le variabili `_ERRMYDB` e `_ERRMYDB$` che specificano il risultato dell'operazione.

Letture dati su anagrafiche MyDB legate ai documenti: la sintassi del nome dell'anagrafica MyDB e la chiave per i documenti cambiano in base alla tipologia di anagrafica che deve essere letta e se l'azienda è infrannuale oppure no. La **sintassi dei nomi degli archivi** MyDB è elencata in fondo alla tabella struttura archivi `_MYDB`.

Le varie **tipologie di chiavi** sono le seguenti:
(attenzione, le chiavi sono diverse a seconda del tipo di documento!)

Chiave per il collegamento con i Movimenti di magazzino:			
Da pos.	A pos.	Dim.	Campo
1	3	3	Numero sottoazienda; Il numero Sottoazienda sui movimenti di magazzino/ordini/matrici/preventivi/Annotazioni vale '000' se non sono presenti sottoaziende, mentre inizia da '001' in presenza di sottoaziende.
4	5	2	Sigla documento
6	7	2/3	Serie documento. NOTA BENE: Se sull'azienda sono stati attivati i sezionali estesi (variabile <code>amsez</code>) occorre indicare la serie lunga 3 anche se si stanno utilizzando sezionali sotto il 99. Le aziende che usano i sezionali estesi (fino a 999), devono impostare la chiave con 3 cifre di sezionale, per ottenere il collegamento con i documenti in tutte le istruzioni sprix (MYDBGGET, MYDBPUT, MYDBVIEW, ecc.)
8	8	1	'.' (un punto)
9	14	6	Numero documento
15	23	9	Codice conto intestatario documento in formato GMM.CCCCC (se l'intestatario non è presente passare 9 blank)
24	31	8	Progressivo riga (0 sulla testata, da 1 a 'n' sulle righe. Equivalente alla variabile <code>_MMNCRERIGA(N)</code>)
32	34	3	Numero di testata
35	42	8	Data documento (solo per le aziende infrannuali) nel formato AAAAMMGG

Chiave per il collegamento con i documenti Ordini/Matric/Preventivi/Annotazioni: ...notare che non c'è il codice conto intestatario....			
Da pos.	A pos.	Dim.	Campo
1	3	3	Numero sottoazienda; Il numero Sottoazienda sui movimenti di magazzino/ordini/matrici/preventivi/Annotazioni vale '000' se non sono presenti sottoaziende, mentre inizia da '001' in presenza di sottoaziende.
4	5	2	Sigla documento
6	7	2/3	Serie documento. NOTA BENE: Se sull'azienda sono stati attivati i sezionali estesi (variabile <code>amsez</code>) occorre indicare la serie lunga 3 anche se si stanno utilizzando sezionali sotto il 99. Le aziende che usano i sezionali estesi (fino a 999), devono impostare la chiave con 3 cifre di sezionale, per ottenere il collegamento con i documenti in tutte le istruzioni sprix (MYDBGGET, MYDBPUT, MYDBVIEW, ecc.)
8	8	1	'.' (un punto)
9	14	6	Numero documento
15	22	8	Progressivo riga (0 sulla testata, da 1 a 'n' sulle righe)

Chiave per il collegamento con le Parcelle:			
Da pos.	A pos.	Dim.	Campo
1	3	3	Numero sottoazienda; Il numero Sottoazienda sulle parcelle parte sempre da '001' indipendentemente dal fatto che siano presenti o meno sottoaziende.
4	5	2	Sigla documento
6	7	2/3	Serie documento. NOTA BENE: Se sull'azienda sono stati attivati i sezionali estesi (variabile <code>amsez</code>) occorre indicare la serie lunga 3 anche se si stanno utilizzando sezionali sotto il 99. Le aziende che usano i sezionali estesi (fino a 999), devono impostare la chiave con 3 cifre di sezionale, per ottenere il collegamento con i documenti in tutte le istruzioni sprix (MYDBGGET, MYDBPUT, MYDBVIEW, ecc.)
8	8	1	'.' (un punto)
9	14	6	Numero documento
15	23	9	Codice conto intestatario documento
24	33	10	Data documento (AAAAMMGG)
34	37	4	Numero riga (0 sulla testata, da 1 a 'n' sulle righe)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBGETATT

Istruzione che legge l'anagrafica MYDB corrente. Esegue in automatico l'istruzione AZZVARSYS 35 e AZZCATVARSYS "MYDB", ovvero azzerare le variabili preesistenti ed inizializza la struttura con i valori esistenti. L'istruzione non azzerare le variabili `_MYDBK$()`. Imposta le variabili `_MYDBID`, `_MYDBELE`, `_MYDB$()`, `_MYDBANN$`, `_MYDBETIC$()`, `_MYDBCOD$()` e `_MYDBARC$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBINPUT

MYDBINPUT <Archivio\$>,<NumeroID>,[Parametri\$]

Parametri	
Archivio\$	Archivio sul quale effettuare l'inserimento/modifica dei dati Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi <code>_MYDB</code> .
Numero ID	= 0 Apre la finestra dell'archivio vuota, tutti i campi vuoti, o con i valori di default (non attivabile su archivi collegati ai documenti); >=1 apre la finestra in Revisione dell'anagrafica con ID immesso; <0 apre la finestra dell'archivio mostrando a video i dati presenti nella struttura <code>_MYDB\$()</code> ;
Parametri\$	" NOPUT " = Non esegue la scrittura e/o Cancellazione del record all'uscita della finestra. Contestualmente a questo parametro è possibile attivare anche l'opzione " NOCTR " per disattivare i controlli che vengono fatti in automatico. " NOCAN " = Disabilita il pulsante per la cancellazione sia fisica che logica del record " DOC= " Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (La sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio) " RIGA ". Numero di riga di posizionamento della videata sulla finestra attiva; " COLONNA ". Numero di colonna di posizionamento della videata sulla finestra attiva; Se "RIGA=0, COLONNA=0 la videata viene aperta al centro. Esempio: MYDBINPUT ARC\$,_MYDBID,"NOPUT,NOCAN,RIGA=0,COLONNA=0"

Istruzione di Input a finestra di un'anagrafica. Alla conferma (tasto [F10]) viene eseguita una MYDBPUT, inoltre viene impostata la variabile `_MYDBTF`. **Notare:** Su archivi collegati a documenti e parcelle è ammessa solo la variazione, non è possibile inserire record nuovi.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBLOF

MYDBLOF (<Archivio\$>,[Parametri\$])

Parametri	
Archivio\$	Archivio su cui determinare il numero delle chiavi. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi <code>_MYDB</code> .
Parametri\$	" DOC= " Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (la sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio).

Funzione che ritorna il numero delle chiavi inserite nell'archivio specificato. Se ritorna -1 non trova l'anagrafica indicata nel parametro. **Notare:** la funzione non è utilizzabile all'interno di un ciclo di lettura del file (MYDBGET e MYDBNEXT / MYDBPREV) eseguire prima la MYDBLOF e poi il ciclo e le funzioni di lettura del file.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBNEXT

MYDBNEXT <Archivio\$>,<NumeroKey>,[Parametri\$]

Parametri	
Archivio\$	Archivio sul quale effettuare la lettura. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi <code>_MYDB</code> .
NumeroKey	= 0 Esegue la lettura per ID >= 1 Ordina ed esegue la lettura in base alle chiavi 1, 2 o 3 specificate sulla definizione = 99 Esegue l'ordinamento e la lettura per estensione, se dichiarata
Parametri\$	" DOC= " Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (la sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio).

Istruzione che si posiziona sull'anagrafica successiva all'ultima anagrafica letta o scritta, indipendentemente da eventuali impostazioni di filtri o di chiavi di ricerca. Valorizza le variabili `_ERRMYDB` e `_ERRMYDB$` che specificano il risultato dell'operazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBPREV

MYDBPREV <Archivio\$>,<NumeroKey>,[Parametri\$]

Parametri	
Archivio\$	Archivio sul quale effettuare la lettura. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi _MYDB.
NumeroKey	= 0 Esegue la lettura per ID >= 1 Ordina ed esegue la lettura in base alle chiavi 1, 2 o 3 specificate sulla definizione = 99 Esegue l'ordinamento e la lettura per estensione, se dichiarata
Parametri\$	"DOC=" Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (la sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio).

Istruzione che si posiziona sull'anagrafica precedente all'ultima anagrafica letta o scritta, indipendentemente da eventuali impostazioni di filtri o di chiavi di ricerca. Valorizza le variabili _ERRMYDB e _ERRMYDB\$ che specificano il risultato dell'operazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBPUT

MYDBPUT <Archivio\$>,[Parametri\$]

Parametri	
Archivio\$	Archivio su cui effettuare la scrittura. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi _MYDB.
Parametri\$	"DOC=" Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (La sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio).

Istruzione che esegue la scrittura/variazione di un'anagrafica/tabella. Se la scrittura viene fatta su un archivio estensione, viene controllato che l'estensione sia esistente. Prende i dati dalle variabili _MYDB\$(), _MYDBANN\$ e _MYDBID. Se _MYDBID = 0 esegue un inserimento. In caso di scrittura sulle tabelle occorre passare nella variabile _MYDBID il numero del record che si vuole scrivere. Imposta le variabili _MYDBID e _MYDBNR. Vengono ritornate le variabili _MYDBID, _MYDBNR e _ERRMYDB e _ERRMYD\$ che specificano il risultato dell'operazione.

Scrittura dati su anagrafiche MyDB legate ai documenti: La scrittura di una anagrafica legata ai documenti imposta in automatico a '1' anche le variabili _MMMYYDBT (per i dati di testata se presenti) e _MMMYYDBR (per i dati di riga se presenti). Per le specifiche si rimanda alla struttura archivi _MM. La **sintassi** del nome dell'anagrafica MyDB e la **chiave** per i documenti cambiano in base alla tipologia di anagrafica che deve essere scritta e se l'azienda è infrannuale oppure no. La **sintassi dei nomi degli archivi** MyDB è elencata in fondo alla tabella struttura archivi _MYDB, le varie **tipologie di chiavi** sono riportate nel paragrafo della MYDBGET).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBPUTATT

MYDBPUTATT [Parametri\$]

Parametri	
Parametri\$	"GOTO=CodiceCampo" . Si posiziona sul codice campo passato, o se non possibile, sul successivo disponibile. "NOCTR" . Non esegue i controlli di gestione sui vari campi

Istruzione di impostazione dell'anagrafica corrente. Legge il contenuto delle variabili _MYDB\$(), _MYDBANN\$ e sovrascrive (solo in memoria) l'anagrafica corrente. Poiché Collage è richiamato da un campo di input, i nuovi dati saranno visibili all'uscita del campo corrente.
Nota: Il comando MYDBPUTATT non esegue l'aggiornamento del campo in cui il cursore è posizionato, per tale valorizzazione utilizzare la _SINP\$; per azzerare il campo in cui ci si trova, quindi, occorre aggiungere _SINP\$="" altrimenti il campo viene reimpostato.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBRIORG

MYDBRIORG <Archivio\$>,[Parametri\$]

Parametri	
Archivio\$	Archivio sul quale effettuare la riorganizzazione. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi _MYDB.
Parametri\$	"SOLOKEY" Ricrea solo le chiavi, non controlla i dati "SILENT" Non visualizza la finestra di avanzamento della riorganizzazione "STAMP=numero stampante" Imposta la stampante sulla quale mandare l'esito della riorganizzazione. Se la stampante indicata non è disponibile, viene richiesta. "DOC=" Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (La sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio).

Istruzione di riorganizzazione di un archivio MyDB. Se lanciato senza parametri viene riorganizzato l'archivio passato (compreso il controllo dei dati) visualizzando la finestra di avanzamento. L'istruzione valorizza le variabili _ERRMYDB e _ERRMYD\$ che specificano il risultato dell'operazione, in particolare i codici 102, 110 114 relativi all'assenza o non definizione dell'archivio MyDB. **Poiché la riorganizzazione**

prevede la creazione delle chiavi, nessun terminale deve aver aperto tale archivio. Non viene controllato, quindi nel caso il programma chiude il terminale come fa con la OPEN "K*".

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBTAB

MYDBTAB <Archivio\$>,[Parametri\$]

Parametri	
Archivio\$	Archivio su cui eseguire il ciclo tabellare. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi _MYDB.
Parametri\$	"DOC=" Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle. (La sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio). Se il parametro non viene inserito il ciclo avviene su tutte le tipologie di documento.

Istruzione che imposta la tabella per il ciclo tabellare dell'archivio specificato. L'istruzione va dichiarata prima dell'esecuzione del ciclo tabellare, ma fuori dal ciclo stesso. L'istruzione valorizza le variabili _ERRMYDB e _ERRMYD\$ che specificano il risultato dell'operazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MYDBVIEW

MYDBVIEW <Archivio\$>,<NumeroKey>,<NumeroID>,[Parametri\$],

Parametri	
Archivio\$	Archivio su cui eseguire la lettura. Per la sintassi dei nomi degli archivi MyDB vedere in fondo alla tabella struttura archivi _MYDB.
NumeroKey	= 0 Esegue la lettura per ID >= 1 Ordina ed esegue la lettura in base alle chiavi 1, 2 o 3 specificate sulla definizione. La lettura parte dal valore impostato nell'array _MYDBK\$(1) = 99 Esegue l'ordinamento e la lettura per estensione, se dichiarata, ricercando la chiave impostata nell'array _MYDBK\$(1)
Numero ID	Se <NumeroKey> uguale a 0 si posiziona sull'ID immesso in questo parametro Se <NumeroKey> diverso da 0 si posiziona sulla chiave immessa in _MYDBK\$(1) ed a parità di chiavi eventualmente sul <Numero ID> immesso
Parametri\$	"INSER". Abilita l'inserimento di una nuova anagrafica. Non attivo su archivi documento o parcella. "FILTRO=A". Nessun filtro e visualizza anche gli annullati "FILTRO=1". Usa come filtro i campi specificati nell'array _MYDBF\$() (e NON considera gli annullati) "FILTRO=2". Usa come filtro i campi specificati nell'array _MYDBF\$() (e considera gli annullati) "FILTRO=3, Numero". Usa come filtro il numero di Selezione salvato precedentemente nella gestione dell'anagrafica (e NON considera gli annullati) "FILTRO=4, Numero". Usa come filtro il numero di Selezione salvato precedentemente nella gestione dell'anagrafica (e considera gli annullati) Nel caso di filtro 3 e 4 se il filtro non viene trovato, non viene ritornato nessun errore, e viene letto tutto l'archivio. "DOC=" Sigla documento. Richiesto solo per archivi collegati ai documenti e alle parcelle (La sigla impostata serve esclusivamente ad indirizzare la lettura per lo specifico archivio). "NOSTAM". Disabilita il pulsante di stampa. "NOPUT". Disattiva il salvataggio del record appena inserito, in modo da poter verificare il contenuto dei dati caricati prima del loro salvataggio. Se il parametro è attivo, il programma una volta effettuato F10, ritorna il controllo a Sprix/Collage che lo ha invocato per poter fare i controlli del caso e poi eventualmente rilanciare la MYDBVIEW o la MYDBINPUT in base al risultato delle verifiche. Contestualmente a questo parametro è possibile attivare anche l'opzione "NOCTR" per disattivare i controlli che vengono fatti in automatico. "NODUPL". Disabilita il pulsante di duplicazione. "RIGHE=??". Indica il numero di righe che si vogliono gestire nella finestra elenco: se < di 11 assume 11; se > delle righe disponibili a video assume il numero massimo delle righe disponibili, altrimenti il numero passato; se non indicato assume 11; "VISEST". Consente di ottenere in prima posizione la visualizzazione dell'estensione. "RIGA". Numero di riga di posizionamento della videata sulla finestra attiva; "COLONNA". Numero di colonna di posizionamento della videata sulla finestra attiva; Se "RIGA=0, COLONNA=0 la videata viene aperta al centro

Istruzione che apre la finestra di ricerca di un archivio. Se NumeroKey e NumeroID sono uguali a zero viene visualizzato tutto l'archivio. Se si seleziona un record vengono automaticamente impostate le variabili come se si fosse eseguita una MYDBGET, inoltre viene impostata la variabile _MYDBTF. Se viene aperta la finestra di ricerca di un archivio estensione dei documenti o delle parcelle non è attivabile il parametro "INSER".

Notare: nel caso in cui l'elemento dell'array _MYDBF\$() sia un campo dell'archivio stesso, si imposta il valore del campo; se l'elemento dell'array _MYDBF\$(X) è una relazione ad un altro archivio o tabella MYDB, il valore da impostare nell'elemento _MYDBF\$(X) è il _MYDBID dell'archivio relazionato.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODBFIELD

MODBFIELD <opz\$>,<FROM row TO...>,<FROM col TO... AS varbuf [tipoiimp]>, ...

Parametri	
opz\$	Opzioni del campo separate da punto e virgola “;” (obbl = se assegnato a 1 non fa attivare il flag _MODSCRIVI se il campo è vuoto; break = se assegnato a 1 imposta l’attivazione del flag _MODSCRIVI non più sui fine pagina ma sulla rottura del valore letto).
FROM row TO...	Riga di inizio e fine campo/i
FROM col TO...	Colonna d’inizio e fine campo
AS varbuf	Variabile associata
tipoimp	Tipo impaccamento (facoltativo)

Istruzione che definisce per la pagina, la variabile e l’area del file da cui assegnare i dati alla variabile. È simile all’istruzione FIELD con dimensione anche verticale <FROM rowstart TO rowend>, <FROM colstart TO colend AS varbuf [tipoimp]>. L’istruzione è utilizzabile se è invocata l’istruzione MODBODY.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODGET

Istruzione che legge i dati dal file definito in MODOPEN e li assegna alle variabili definite con le MODFIELD.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODBODY

MODBODY <opzioni\$>, <FROM rowstart TO rowend>,<rows>

Parametri	
opzioni\$	Opzioni del corpo pagina (parametro non ancora attivo);
FROM rowstart TO rowend	Riga di inizio e fine corpo nell’ambito della pagina;
rows	Righe fisiche della riga logica del corpo pagina;

Istruzione che definisce la gestione di un eventuale corpo pagina, con <opzioni\$> del corpo pagina (parametro non ancora attivo) e con riga di inizio e fine corpo nell’ambito della pagina <FROM rowstart TO rowend> e righe fisiche della riga logica del corpo pagina <rows>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODCLOSE

Istruzione che chiude il file testo del modulo e tutta la gestione modulo. Valorizza le variabili _ERRMOD e _ERRMOD\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODFIELD

MODFIELD <opz\$>,<FROM row TO...>,<FROM col TO... AS varbuf [tipoimp]>, ...

Parametri	
opz\$	Opzioni del campo separate da punto e virgola “;” (obbl = se assegnato a 1 non fa attivare il flag _MODSCRIVI se il campo è vuoto; break = se assegnato a 1 imposta l’attivazione del flag _MODSCRIVI non più sui fine pagina ma sulla rottura del valore letto).
FROM row TO...	Riga di inizio e fine campo/i
FROM col TO...	Colonna d’inizio e fine campo
AS varbuf	Variabile associata
tipoimp	Tipo impaccamento (facoltativo)

Istruzione che definisce per la pagina, la variabile e l’area del file da cui assegnare i dati alla variabile. È simile all’istruzione FIELD con dimensione anche comando <FROM rowstart TO rowend>, <FROM colstart TO colend AS varbuf [tipoimp]>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODOPEN

MODOPEN <tipo\$>,<nomefile\$>,<archivi\$>,<opzioni\$>,<righe>,<colonne>

Parametri	
tipo\$	Tipo di modulo di stampa; se = “S” è semplice con righe sequenziali senza pagine; se = “C” è complesso con pagine ed eventuale corpo pagina con la necessaria indicazione del numero di <righe> pagina;
nomefile\$	Nome del file testo;
archivi\$	Archivi da gestire. Archivi disponibili: “sc” = saldi contabili
opzioni\$	Opzioni del modulo separate da punto e virgola “;” (esempio: “des=Import Saldi;decsep=.”)
des	Eventuale descrizione del modulo di import;
decsep	Carattere separazione decimale virgola (,) o punto (.); il default è virgola;
cls	Codice classificazione (es. “SC” Saldi conto);

	newline	Codici numerici per i separatori di riga (max 2), se non definito si utilizza lo standard ASCII 13,10 (CHR(13), CHR(10));
righe	Numero di righe per pagina	
colonne	Numero di colonne	

Istruzione che apre il file testo dal nome <nomefile\$> relativo a modulo di stampa da importare di un certo <tipo\$> e con certo numero di <colonne>, al fine di importare il tipo di <archivi\$> da gestire, in base alle seguenti <opzioni\$> del modulo separate da punto e virgola “,”.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODPUT

MODPUT <archivi\$>

Parametri	
archivi\$	Archivi definiti nell'istruzione MODOPEN e indicati con sigle in minuscolo separate da virgola (es. "sc" = saldi conto). In caso di "sc", è attiva la variabile di struttura di primanota_PNNDO che, se è impostata, scrive il documento assegnandogli il numero, creando le relative scadenze.

Istruzione che scrive le variabili predefinite lette con MODGET negli <archivi\$> definiti in MODOPEN
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODSFIELD

MODSFIELD <opz\$>,<varbufori [tipoimp]>,<nomefile\$>,<obbligo>,<varbufdesori\$>

Parametri	
opz\$	Opzioni (parametro non ancora attivo);
varbufori [tipoimp]	Variabile per cui gestire le sostituzioni deve esistere nelle MODFIELD / MODBFIELD;
nomefile\$	Nome logico del file da gestire (conti predefiniti e generico);
obbligo	Se = "S" il flag _MODSCRIVI non viene attivato se la variabile <u>non</u> ha un valore associato, cioè l'eventuale variabile di descrizione di <varbufori> da presentare nella gestione delle assegnazioni sostituzioni (opzionale);
varbufdesori\$	Valore associato;

Istruzione che permette di associare ad un campo una tabella di sostituzioni (menu Servizi \ Personalizzazioni \ Importazione Moduli \ Tabelle Sostituzioni), in base ad opzioni.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MODVMSG

Istruzione che visualizza gli eventuali messaggi che sono contenuti nella variabile _MSGMOD\$() alla fine di un'elaborazione del modulo. Utilizzabile dopo MODOPEN.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

NCOLDISPFIN

NCOLDISPFIN(<barra_navigazione, pulsanti_scheda>)

Parametri	
barra_navigazione	1/0 per il calcolo delle colonne di una finestra con / senza barra di navigazione
pulsanti_scheda	1/0 per il calcolo delle colonne di una finestra con / senza pulsanti scheda

Funzione che restituisce il numero di colonne **disponibili** (utili) per una finestra.
 La funzione si rapporta alla variabile dizionario "sxc01" colonne a **video**.
 In base al tipo di interfaccia possono equivalere (interfacce < 2014) o differire di una colonna (interfacce > 2014).
 Per approfondimenti si rimanda alle indicazioni contenute in calce all'elenco della seguente tabella [Clicca qui](#)
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

NEXTKEY

NEXTKEY <numfile>,<numkey>

Parametri	
numfile	Numero File Indici
numkey	Numero chiave (da 1 a 4, cioè una fra quella primaria e le secondarie)

Istruzione che ritorna nelle variabili associate al buffer del file a indici aperto con numero <numfile>, il record successivo all'ultimo record letto (quello ritornato da GETKEY, NEXTKEY o PREVKEY) seguendo la chiave <numkey>. Se il record successivo non esiste tutte le variabili del buffer sono impostate con ASCII 0. Valorizza _NRDATI e _ERRIND.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

NOMECOLLAGE

NOMECOLLAGE\$(<num>)

Parametri	
num	Numero indice d'ordine del Collage in memoria (fra 1 e MAXCOLLAGE())

Funzione che ritorna il nome del file Collage con numero indice d'ordine <num>del Collage caricato in memoria.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

NOMEPASSAPP

NOMEPASSAPP\$()

Funzione che ritorna il nome completo della PassApp (CODICE+NOMEAPP).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

NRIGDISPFIN

NRIGDISPFIN(<pulsanti_azione>)

Parametri	
pulsanti_azione	1/0 per il calcolo delle righe di una finestra con / senza pulsanti azione

Funzione che restituisce il numero di righe **disponibili** (utili) per una finestra.

La funzione si rapporta alla variabile dizionario "sxrig" righe **a video**.

In base al tipo di interfaccia possono equivalere (interfacce < 2014) o differire di due righe (interfacce > 2014).

Per approfondimenti si rimanda alle indicazioni contenute in calce all'elenco della seguente tabella [Clicca qui](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

NUMPAGFIN

NUMPAGFIN <n1>,<n2>,<n3>,[s\$]

Parametri	
n1	Numero finestra;
n2	Numero pagina attuale
n3	Numero massimo di pagina
s\$	Stringa non attiva (parametro predisposto);

Istruzione per impostare il numero della pagina e delle pagine totali nella APRIFIN_EXT con la barra di navigazione attiva. Se nessuno dei due numeri supera il 9 verrà riportato ad esempio 1/2 altrimenti solo il numero della pagina.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

NUMPAGPDF

NUMPAGPDF(<pdf\$>)

Parametri	
pdf\$	Nome del file PDF completo di path

Funzione che restituisce il numero di pagine di un file PDF, quindi PG_COUNT=NUMPAGPDF("File.pdf"). Come per tutte le istruzioni che accedono ai file è possibile attivare la gestione degli errori e il controllo accesso utente al file tramite la variabile _ACCFILEUTE; in tal caso gli eventuali errori sono ritornati nella variabile _ERRFILE\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_BARCODE_NOMEINPUT

ON_BARCODE_NOMEINPUT:

Evento Sprix mobile di lettura barcode. Restituisce una array _WIBARCODE\$ (con quantità di elementi _WIBARNUM) di stringhe

contenente il tipo di codice letto `_WIBARTYPE$` e suddivisa in stringhe di 255 caratteri; se nell'evento viene effettuata una settext del campo di input, il valore sovrascrive quello letto dal lettore barcode.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_BEEP

ON_BEEP:

Evento all'emissione di segnale acustico di avviso impostato e definito nel file ".ini" della componente Client (MxDesk.ini). Consultare le Nozioni Generali del Manuale Generale dell'Utente.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_CTRLFILTRROMYDB

ON_CTRLFILTRROMYDB:

Evento richiamato all'uscita del filtro di anagrafica. Eseguito dopo i controlli del programma, ma prima dell'evento `ON_ENDFILTRROMYDB`. Sono attivi `MYDBGETATT` e `MYDBPUTATT` e la variabile `_TF` che contiene il codice del tasto di uscita. Se si modifica `_TF` o si esegue la `MYDBPUTATT` e si modifica `_TF`, il programma rientra in anagrafica per visualizzare le modifiche e rieseguire i controlli sull'anagrafica. Se si imposta `_TF` con un tasto valido, questo viene rielaborato. L'evento è alternativo all'evento `ON_CTRLMYDB`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_CTRLMYDB

ON_CTRLMYDB:

Evento richiamato all'uscita di anagrafica, ma prima degli eventi `ON_PUTMYDB`, `ON_DELMYDB` e `ON_ENDMYDB`. L'evento è alternativo all'evento `ON_CTRLFILTRROMYDB`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_DELMYDB

ON_DELMYDB:

Evento richiamato dopo la cancellazione fisica di un record, ma prima dell'evento `ON_ENDMYDB`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_END

ON_END:

Evento richiamato al termine del modulo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_ENDFILTRROMYDB

ON_ENDFILTRROMYDB:

Evento richiamato all'uscita del filtro di un'anagrafica.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_ENDMYDB

ON_ENDMYDB:

Evento richiamato alla chiusura della finestra di input di un'anagrafica o tabella.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_IN_CAMPO

ON_IN_CAMPO:

Evento Sprix mobile richiamato all'ingresso del "**CAMPO**", che rappresenta il nome assegnato dall'utente al campo di input nell'istruzione `WCREATEINPUT`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_LISTADOCMM

ON_LISTADOCMM:

Evento richiamato sulla lista dei documenti nel modulo Emissione Revisione Documenti. Tale evento viene attivato solo se i parametri "Compatibilità collage" e "Accesso emis./revis. Doc a lista" sono rispettivamente a "No" e "Si". Il vettore _LISTADOCMM\$, contiene i 999 caratteri presenti a video. Tali informazioni, sono suddivise nei 4 slot dell'array (250+250+250+249 = 999).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_LISTALOTTI

ON_LISTALOTTI:

Evento che, prima della scrittura del documento nuovo (oppure riscritto con ulteriori righe nuove con autoprelievo attivo), rende disponibili gli identificativi lotti di autoprelievo per la riga che si sta processando.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_OUT_CAMPO

ON_OUT_CAMPO:

Evento Sprix mobile richiamato all'uscita del "CAMPO", che rappresenta il nome assegnato dall'utente al campo di input nell'istruzione WCREATEINPUT.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_OUT_MDSELTEST

ON_OUT_MDSELTEST:

Evento su finestra [Shift+F8] di interrogazione dati MyDB di testata di un documento trasformato (esempio una bolla nata da più ordini) in cui viene chiesto di quale testata si vogliono vedere i dati. Questo evento consente di eseguire eventualmente una SCREEN\$ della posizione corrente, per sapere qual'è il documento scelto. Esempio:

ON_OUT_MDSELTEST:

IFF _TF=501

RI=GETPOSCURSY(2): CO=GETPOSCURSX(2)

X\$=SCREEN\$(_NFIN,RI,CO,100)

VIMSG 4,STR\$(_TF,0,0)+"|"+X\$

' Imposta i dati

ELSEF

X\$=""

' Azzera i dati

ENDF

END

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_PRESS_LISTA

ON_PRESS_LISTA:

Evento Sprix mobile richiamato quando viene premuto invio su una riga della lista "LISTA", che rappresenta il nome assegnato dall'utente alla lista nell'istruzione WCREATELIST. Viene valorizzata la variabile array _WLVALFLD\$() che contiene il valore dei campi della riga.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_PRESS_PULSANTE

ON_PRESS_PULSANTE:

Evento Sprix mobile richiamato alla pressione del pulsante definito dall'istruzione WCREATEINPUT; "PULSANTE" è il nome assegnato dall'utente al pulsante.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_PUTMYDB

ON_PUTMYDB:

Evento richiamato dopo la scrittura di un record MyDB, ma prima dell'evento ON_ENDMYDB.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_ROW_LISTA

ON_ROW_LISTA:

Evento Sprix mobile richiamato ad ogni riga quando viene disegnata la lista "LISTA", che rappresenta il nome assegnato dall'utente alla lista nell'istruzione WCREATELIST. Viene valorizzata la variabile _WLNUMROW numero della riga corrente. Tale numero viene utilizzato per la lettura della variabile array _WLVALFLD\$. Se la lista è legata a un archivio mobile all'evento tramite l'istruzione GETROWVALFLD possono essere letti tutti valori relativi al record dell'archivio.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_ROWFILTER_LISTA

ON_ROWFILTER_LISTA:

Evento Sprix mobile richiamato ad ogni riga quando viene disegnata la lista "LISTA", che è il nome assegnato dall'utente alla lista nell'istruzione WCREATELIST. Viene valorizzata la variabile _WLNUMROW numero della riga corrente. Tramite la variabile _WLROWOK\$ si ha la possibilità di impostare per i valori letti al momento se il dato deve essere inserito nella lista oppure no. Se il dato deve essere inserito _WLROWOK\$ = "S" se non deve essere inserito _WLROWOK\$ = "N".

L'evento ON_ROWFILTER deve essere gestito ogni qualvolta si ha la necessità di operare un filtraggio sulle righe della lista. Dentro l'etichetta che gestisce questo evento la riga i-esima è validata o meno impostando la variabile di struttura _WLROWOK\$="S" oppure _WLROWOK\$="N". Se la riga è stata validata (_WLROWOK\$="S") successivamente scatta l'evento ON_ROW. Nella routine di gestione di questo evento è possibile estrarre elementi della lista, popolare campi di una lista dinamica, o definire campi personalizzati per una lista estesa collegata ad archivio. La gestione dell'evento ON_ROWFILTER è facoltativo per tutte le liste, mentre la gestione dell'evento ON_ROW è obbligatorio per le liste dinamiche e per le estensioni di liste collegate ad archivio. In questi due casi infatti è proprio all'interno della routine di gestione dell'evento ON_ROW che vengono valorizzati i campi della lista. Per poter aggiornare i dati di una lista (liste dinamiche o estese) è necessario utilizzare una opportuna istruzione WCALL "REFRESHLIST", <id_oggetto>. Questo può essere utile nel caso i dati che compongono una lista dinamica o una estensione di lista collegata ad archivio, vengano modificati runtime e quindi si voglia procedere all'aggiornamento della lista che li contiene.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_SMSGARTMM

ON_SMSGARTMM:

Evento richiamato prima dell'apertura delle finestre di dettaglio articoli in emissione revisione documenti. Tale evento permette di gestire al meglio l'apertura e/o chiusura delle finestre rispettando la sequenza delle stesse.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_START

ON_START:

Evento richiamato all'avvio del modulo. E' il primo evento che viene processato per la personalizzazione che non sia compilata in percorso con voce di menu di tipo "AUTO" oppure che non sia richiamato con l'istruzione STARTCOLLAGE, dato che viene processato all'effettivo caricamento in memoria, eccetto una gestione nel codice.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_STARTFILTRROMYDB

ON_STARTFILTRROMYDB:

Evento richiamato all'ingresso del filtro di un'anagrafica.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_STARTMYDB

ON_STARTMYDB:

Evento richiamato prima dell'apertura della finestra di input di un'anagrafica o tabella.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ON_STARTMYDBRIC

ON_STARTMYDBRIC:

Evento richiamato all'ingresso della lista dei record visualizzati. Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

ON_STARTSTPEMAIL

ON_STARTSTPEMAIL:

Evento Collage successivo all'apertura della stampante di tipo Email SMTP impostata con tutti i relativi parametri per l'invio; tale etichetta identifica l'inizio della stampa su stampante di tipo Email SMTP.
--

Nota bene: l'evento ha effetti diversi sulle Variabili Collage di Modulistica.

Se il modulo è "grafico" questo viene composto DOPO che scatta l'evento in questione, pertanto le variabili sono valorizzate. Se il modulo è "tradizionale" questo viene composto PRIMA che scatti l'evento in questione, pertanto le variabili restano vuote; in alternativa usare i consueti eventi ON_STARTMM: oppure ON_PUTMM:

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

ON_START_SPRIX

ON_START_SPRIX:

Evento Sprix mobile richiamato all'entrata nella voce di menu, termina con la chiusura dell'applicazione o il lancio di un'altra voce di menu. Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

ON_STP_DOC

ON_STP_DOC:

Evento unico richiamato nelle sole Stampe Personalizzate di Magazzino "Movimenti di Magazzino" (menu SMM) e "Documenti di Magazzino" (menu SMF). L'evento è attivo anche sulla Stampa Ordini/Preventivi/Matrici (menu SMO). Non è applicabile nella stampa del modulo del documento di magazzino. Dopo che è stata attivata la label relativa all'evento, la lettura delle informazioni per le righe elaborate avviene tramite l'esecuzione del comando "GETATMM". La stampa è modificabile attraverso la valorizzazione delle variabili di stampa _STP e che hanno un funzionamento analogo a quello delle variabili Collage _MM della stampa del Modulo Documento di Magazzino.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

ON_STP_XML

ON_STP_XML:

Evento unico richiamato in Emissione revisione Documenti (menu MD) e " Emissione differita Documenti " (menu ME). Dopo che è stata attivata la label relativa all'evento, viene elaborato il file XML della fattura elettronica che sarà archiviato successivamente in docuvision. La modifica del file XML avviene tramite l'utilizzo delle variabili di ambiente _FTE???
--

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

OPEN

OPEN <tipo\$>,<numfile\$>,<nomefile\$>[,<nomefilek\$;pos1;lun1;pos2;lun2;...>,<lr>

Parametri	
tipo\$	Tipo di file aperto: <ul style="list-style-type: none"> • se = "R" è file sequenziale; • se = "K" è ad indici; • se = "K*" è ad indici con la riorganizzazione del file chiavi prima dell'apertura (consigliato se monoutenza altrimenti sarebbe meglio che fosse un'apposita funzione a parte);
numfile	Numero di file creato (vedere anche la funzione AUTOFIL). Si possono gestire fino a 15 file contemporaneamente di cui 5 possono essere con chiavi (le chiavi non vengono conteggiate nel numero di file gestibili).
nomefile\$	Nome del file; clicca su Nomenclatura File
nomefilek\$;posN;lunN	Se <tipo\$>="K" o "K*" sono parametri che servono per dichiarare e strutturare da 1 fino a 6 file di chiavi (in passato erano 4) dal nome <nomefilek\$> (numerabili progressivamente da sinistra a destra dal numero 1 -obbligatorio- e con chiave principale univoca, fino ad altri 2, 3, 4, 5, 6 facoltativi e con chiavi secondarie), ciascuno di essi con posizione e lunghezza <posN;lunN> della chiave all'interno del file <nomefile\$> (con N da 1 fino a 5 posizioni per ciascun <nomefilek\$> e la cui sommatoria dei vari 'lun' non può superare 90 bytes -max lunghezza di ciascuna chiave-). In pratica le chiavi gestibili sono 6 ed ogni chiave può essere fatta da 5 "parti" (non contigue) per un massimo di 90 byte a chiave.

lr	Lunghezza record del file <nomefile\$> da 1 a 4608 bytes (in passato era di 2048). Consentiti fino a 16.000.000 di records. La lunghezza di un campo del record è 255 byte (equivalentemente alla variabile).
----	---

Istruzione che apre un file sequenziale oppure ad indici (o con eventuale riorganizzazione delle chiavi). Terminate le operazioni della elaborazione del file successive all'apertura, il file deve essere chiuso con l'istruzione CLOSE <numfile> in modo da renderlo disponibile. Il termine del programma chiude il file.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

OPENCOM

OPENCOM <com\$>,<baudrate>,<parity>,<size\$>,<stopbit>,<timeout>,<flag\$>

Parametri	
com\$	Stringa relativa al numero di porta seriale che si vuole gestire; una sola porta alla volta; l'apertura di porta sul Client Remoto (cioè non localhost) richiede reindirizzamento su Client attraverso il simbolo "<", esempio: "<COM1";
baudrate	Velocità della porta (110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400);
parity	Parità della porta (0 = nessuna, 1 = dispari, 2 = pari);
size\$	Numero di bit ("5","6","7","8");
stopbit	Flag bit di stop (0 significa 1 stopbit; 2 significa 2 stopbit);
timeout	Tempo di attesa in numero di secondi/decimi/millesimi; Con le quantità positive significa secondi (esempi: 5 significa cinque secondi, 0 significa dieci secondi); con le quantità negative (precedute dal segno meno "-") significa decimi di secondo (in Linux) e millesimi di secondo (in Windows).
flag\$	Stringa <u>facoltativa</u> per specificare il comportamento di lettura del buffer in caso di un timeout; se = "0" (default) la variabile _BUFCON\$ non è valorizzata; se = "1" la _BUFCON\$ contiene i byte letti fino a quel momento (tempo di timeout);

Istruzione che apre la porta seriale <com\$>, con velocità <baudrate>, parità <parity>, stringa con numero di bit della parola <size\$>, flag bit di stop <stopbit>, numero secondi di attesa <timeout> in base al sistema operativo. Default in caso di parametri non validi: OPENCOM <com\$>,9600,2,"7",0,0. L'istruzione valorizza la variabile _ERRCOM\$. esempio in comandi per argomento.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

OPENCSV

<HND> = OPENCSV(<NomeFile\$>,<Separatore\$>)

Valore di ritorno	
HND	Handle che verrà passato alle altre funzioni per la lettura del file CSV
Parametri	
NomeFile\$	[Percorso/]NomeFile CSV da aprire
Separatore\$	Carattere utilizzato nel file CSV per separare i valori delle colonne

Funzione utile per leggere file CSV senza creare routine, parser, ecc. Le funzioni di libreria del Gestionale possono leggere i valori di una singola colonna di una riga di un file CSV, pertanto, creando un iteratore si possono scorrere le righe del CSV e, della riga letta, si può chiedere il valore di ogni singola colonna; lo stato dell'iteratore viene raggiunto dalle diverse funzioni comunicando a queste l'Handle associato al file ".CSV" ottenuto in fase di apertura.

Nello specifico, la funzione OPENCSV apre il file CSV e comunica il tipo di separatore dei campi del file CSV. Quando la funzione termina con successo viene ritornato un valore numerico che è l'handle con il quale si farà riferimento nelle operazioni successive al file CSV. HND vale 0 oppure un valore intero qualunque. Il numero di file contemporaneamente apribili valore di 8. L'apertura dei file con la funzione OPENCSV, utilizzano le stesse strutture dati utilizzate dall'istruzione OPEN, per cui occorre che vi sia la disponibilità di HANDLE di file SPRIX per il successo dell'operazione. La funzione valorizza le variabili di codice e descrizione errore _ERRCSV e _ERRCSV\$.

Esempio: HND = OPENCSV("bilue_economici.csv", ";") 'apertura file

[esempio in comandi per argomento](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

PARAM_REDD

PARAM_REDD <sog\$>,[mod\$]

Parametri	
sog\$	Parametro obbligatorio, valori consentiti: "C" oppure "D" oppure "" (null). Il doppio apice "" equivale a "D". Il parametro è relativo a Dichiarante / Coniuge, effettivo solo per il modello 730 (cioè modello = 3); gli altri modelli (4, 5, 6, 8), invece, hanno questo parametro sempre preimpostato a "D".
mod\$	Parametro opzionale e che, se immesso, è da scrivere in forma "MOD=???", dove ??? può assumere i seguenti valori: "730" corrisponde al modello con codice 3 "RPF" corrisponde al modello con codice 4

	<p>"RSP" corrisponde al modello con codice 5</p> <p>"RSC" corrisponde al modello con codice 6</p> <p>"RNC" corrisponde al modello con codice 8</p> <p>Questo parametro è obbligatorio per l'esecuzione del ciclo tabellare</p>
--	--

<p>Istruzione relativa ai Contratti di Locazione, valorizza le variabili _ERRREDSPX e _ERRREDSPXS\$.</p> <p>L'istruzione esegue anche il ciclotabellare.</p>
<p>Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>

PIUDATA

PIUDATA\$(<data\$>,<numgior>)

Parametri	
data\$	Data da elaborare; in formato "AAAAMMGG" compreso fra 19600101 e 2048
numgior	Numero di giorni di aumento o diminuzione; compreso fra -32.507 e 32.507; se > 0 aumenta, se < 0 diminuisce;

Funzione che aumenta o diminuisce la stringa <data\$>, del numero di giorni <numgior>.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

PREVKEY

PREVKEY <numfile>,<numkey>

Parametri	
numfile	Numero File Indici
numkey	Numero chiave (da 1 a 4, cioè una fra quella primaria e le secondarie)

<p>Istruzione che ritorna nelle variabili associate al buffer del file a indici aperto con numero <numfile>, il record <u>precedente</u> all'ultimo record letto (quello ritornato da GETKEY, NEXTKEY o PREVKEY) seguendo la chiave <numkey>. Se il record successivo non esiste tutte le variabili del buffer sono impostate con ASCII 0.Valorizza _NRDATI e _ERRIND.</p>
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

PRINT

PRINT <espr\$>[:]

Parametri	
espr\$	Espressione stringa da visualizzare nella finestra attiva;
;	Annulla il salto riga; se questo parametro è assente, dopo la visualizzazione effettua un salto riga;

<p>Istruzione che visualizza l'espressione della stringa <espr\$> nella finestra attiva alla posizione e colore attuale. Se assente il simbolo [:], dopo la visualizzazione, il cursore salta alla prima colonna nella riga successiva della finestra attiva (e se la visualizzazione è sull'ultima riga della finestra avviene lo scrolling di una riga verso l'alto). Non deve eccedere i limiti della finestra attiva e non consente scelta del font. Esempio: A\$="AA": P=12: N=12: M=-21: PRINT A\$+"Bb"; PRINT N+M;: PRINT " "+A\$:PRINT P</p>
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

PRINTD

PRINTD <str\$...:>

Parametri	
str\$...:	Stringa da visualizzare evidenziata;

<p>Istruzione che nell'interfaccia grafica visualizza evidenza la stringa <str\$> in modalità 'descrizione del campo'; elimina i puntini e due punti finali e non manda il cursore a capo (occorre LOCATE).</p>
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

PRINTF

PRINTF <n>,<nrip>

Parametri	
n	Carattere di fincatura; vale: 1 (orizzontale), 2 (verticale), 3 (angolo sup. sin.), 4 (angolo superiore destro), 5 (angolo inferiore sinistro), 6 (angolo inferiore destro), 7 (incrocio), 8 (raccordo in alto), 9 (raccordo in basso), 10 (raccordo a sinistra), 11 (raccordo a destra)
nrip	Numero di ripetizioni del carattere;

Istruzione che visualizza il carattere <n> di fincatura per <nrip> volte. Istruzioni alternative: STRING\$ o PRINT.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PRINTH

Istruzione che forza in modo incondizionato la stampa delle righe di intestazione (righe 'H') sulla stampante aperta.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PRINTICONA

PRINTICONA <icona\$>

Parametri	
icona\$	Icona da passare a video; può assumere le "DESCRIZIONI" della lista di icone elencate in SETTF (es. "CESTINO")

Istruzione che visualizza a video l'icona passata come parametro <icona\$>.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PROGARTMAG

PROGARTMAG <codicearticolo\$>,[nummagazzino]

Parametri	
codicearticolo\$	Codice dell'articolo di cui si richiedono i progressivi. Il parametro è obbligatorio.
nummagazzino	Numero di magazzino.

Istruzione che scrive nelle variabili struttura _PAM il risultato della ricerca dei progressivi articoli. L'istruzione richiede MDS.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUT

PUT <numfile>,<numrec>

Parametri	
numfile	Numero del file sequenziale (di tipo "R") aperto;
numrec	Numero di record del file sequenziale (di tipo "R") aperto; se <numrec> è inferiore ad 1 la scrittura non è eseguita, se <numrec> è superiore al numero di record presenti nel file, vengono accodati al file tanti record con i valori ASCII 0 fino raggiungere il numero di record richiesti (allargamento totale massimo 32.767 caratteri).

Istruzione che scrive il numero di record <numrec> nel file aperto di tipo "R" numero <numfile>, utilizzando le variabili del buffer associato: in scrittura, se la lunghezza dei valori delle variabili è superiore alle dimensioni dei buffer viene scritta solo la parte non eccedente, se è inferiore vengono scritti i caratteri ASCII 0 per la parte eccedente.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTABBIN

Istruzione che scrive/aggiorna i dati dei Codici Abbinati in base alla valorizzazione delle variabili di struttura archivi _AB.
 La PUTABBIN non scrive le categorie abbinati, devono essere già in tabella.
 L'istruzione valorizza le variabili _ERRAB e _ERRAB\$.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTAGP

PUTAGP [par1]

Parametri	
par1	Elemento dell'array della registrazione da effettuare; se non dichiarato, viene assunto _AGPNELE e se questo risulta 0, viene assunto valore 1.

Istruzione che scrive i dati delle righe dell'Agenda di Lavoro in base alla valorizzazione delle variabili di struttura archivi _AGP. L'istruzione valorizza le variabili _ERRAGP e _ERRAGP\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTALIAS

Istruzione che scrive/aggiorna i dati dei Codici Alias in base alla valorizzazione delle variabili di struttura archivi _AL. L'istruzione valorizza le variabili _ERRAL e _ERRAL\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTAR

PUTAR <nummag>

Parametri	
nummag	Numero magazzino;

Istruzione che scrive/aggiorna i dati dell'Anagrafica Articolo in base alla valorizzazione delle variabili di struttura _AR. La scrittura di un articolo nuovo richiede i dati obbligatori della versione del gestionale in uso; nel caso di un articolo esistente lo aggiorna (valorizzare sempre tutte le variabili di interesse, altrimenti alcune potrebbero azzerarsi). I progressivi sono in sola lettura (eccetto l'inventario nel primo anno dell'articolo) e modificabili solo con documenti rettificativi. L'istruzione valorizza le variabili _ERRAR e _ERRAR\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTAREA

Istruzione che scrive/aggiorna i dati di Anagrafica Aree in base alla valorizzazione delle variabili di struttura archivi _AE. L'istruzione valorizza le variabili _ERRAE e _ERRAE\$. La possibilità di cancellazione è attraverso il modulo Servizi di CdG.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTASSOCDV

PUTASSOCDV <tipo>,<keypass>,<iddoc>,[alleg]

Parametri	
tipo	Tipo di Entità del Gestionale;
keypass	<p>Chiave di ricerca sulle tabelle di associazione fra Gestionale e Docuvision, eccetto per le classi <tipo>=1 Stampa generica, =100 Nominativi, =200 Bilanci, =800 Dichiarativi/Versamenti, =900 Stampe Fiscali; =1000 Agenda di lavoro ed eccetto i casi in cui il livello di azienda o il codice contratto non consentono l'accesso ad alcune tabelle, come esempio le operazioni di classe 'Parcella/Notula' senza gestione parcellazione, classi di Magazzino con azienda Contabile, ecc...:</p> <p>Piano dei Conti: <tipo>=400 <keypass>="<codcon\$>" (codice conto del piano dei conti);</p> <p>Primanota: <tipo>=600 <keypass>="<dreg\$>;<prinr>;<cau\$>;<protr\$>;<prots>;<protn>;<docn>;<ddoc\$>" <dreg\$> (data di registrazione); <prinr> (n. progressivo registrazione primanota); <cau\$> (causale del documento); <protr\$>(registro di protocollazione); <prots> (serie del protocollo); <protn> (numero di protocollazione); <docn> (numero del documento); <ddoc> (data documento);</p> <p>Anagrafica Articoli: <tipo>=1200 <keypass>="<codart\$>" (codice articolo);</p> <p>Movimenti di Magazzino: <tipo>=1400 <keypass>="<sig\$>;<ser>;<num>;<clifor\$>;<ddoc\$>" <sig\$> (sigla documento); <ser> (serie del documento); <num> (numero del documento); <clifor\$> (codice conto intestatario); <ddoc\$> (data documento);</p> <p>Ordine/Preventivi/Matrici: <tipo>=1500 <keypass>="<sig\$>;<ser>;<num>;<ddoc\$>" <sig\$> (sigla documento); <ser> (serie del documento); <num> (numero del documento); <ddoc\$> (data documento);</p>

	<p>Bolla di Lavoro: <tipo>=1600 <keypass> = "<numeroBL>;<numeroSottoBL>" <numeroBL> (numero di bolla di lavoro); <numeroSottoBL> (numero di sottobolla di lavoro); Inserire "-1" per avere tutte le sottobolle, inserire "0" (o non specificare) per intendere la BL principale. In operazione DEL è richiesta anche la data della bolla.</p> <p>MyDB: <tipo>=1700 <keypass> = "<nome MyDB>;<numero elemento>" <nome MyDB> ("codice" (per mydb su app corrente) o "codiceApp@codice" (per mydb su altra app)); <numero elemento> (cod. numerico univoco che identifica l'n-esimo elemento di anagrafica (_MYDBID));</p> <p>Notule/Parcelle: <tipo>=1100 <keypass>="<cau\$>;<ser>;<num>;<clifor\$>;<ddoc\$>"; <cau\$> (causale del documento); <ser> (serie del documento); <num> (numero del documento); <clifor\$> (codice conto); <ddoc> (data documento);</p> <p>Controllo di gestione: <tipo>=1450 <keypass>="<anag\$>;<codice\$>" <anag\$> (tipo anagrafica); <codice\$> (codice dell'anagrafica);</p>
iddoc	ID del documento Docuvision
alleg	<p>Allegati (facoltativo), consente di variare la tipologia di allegato e il progressivo immagine attraverso la PUTASSOCDV. Il parametro ha la seguente sintassi "TPALL=x;PRGALL=y; STP=z" dove:</p> <ul style="list-style-type: none"> • "x" può assumere i seguenti valori null=Generico o I=Immagine o S=Scheda o D=Descrizione; • "y" può assumere valore numerico relativo al progressivo; • "z" può assumere valore SI o NO per alzare il flag di stampa al documento che si sta associando ad una delle seguenti classi <tipo> = 1100, 1400, 1500, 1600; per le restanti classi il parametro è ignorato; le classi 400, 600, 1200 non possiedono il flag di stampa. Nota: non si possono avere 2 documenti allegati alla stessa "entità" del gestionale con il flag di stampa attivato, pertanto, se si tenta di allegare un documento ad un movimento di magazzino stampato su Docuvision, l'istruzione rilascia l'errore "Esiste già" un documento Docuvision tra gli altri allegati con flag di stampa valorizzato". <p>Esempio: PUTASSOCDV 1200,"2.53.03.905",2,"TPALL=I;PRGALL=2;STP=SI"</p> <p>Essendo [alleg] un parametro facoltativo, in caso di omissione equivale al default Generico "TPALL=;"</p>

Istruzione che inserisce/aggiorna i dati delle associazioni fra il Documento Docuvision e l'Entità del Gestionale in base alla valorizzazione delle variabili struttura archivi _AD, in base all'ID del documento Docuvision <idDoc> e il <tipo> di Entità del Gestionale, determinata univocamente dalla <keypass> di ricerca sulle tabelle d'associazione fra Gestionale e Docuvision. Le Entità del Gestionale appartenenti ad archivi annuali devono essere relative all'anno aperto.

Esempio: supponendo di dovere aggiornare una primanota importata da import fatture emesse (fatture elettroniche da file Xml), è noto che la PUTPN non può aggiornare la primanota (per ragioni collegate allo scadenziario), piuttosto può solo scrivere ex-novo la registrazione di primanota. Di conseguenza la necessaria GETPN, con la successiva DELPN e l'opportuna modifica dei dati da riscrivere con PUTPN, azzerà l'associazione al documento XML. In tal caso, si rende necessario ricollegare la "nuova" primanota al documento elettronico.

```
'Leggi la Primanota
GETPN "20210928",0,"FE","V",01,000025,2/39,"20210928" : VIMSG 1, _ERRPN$

'Leggi la Associazione con la Primanota: <tipo>=600
VAR_1$="20210928;0;FE;V;01;000025;239;20210928"
GETASSOCDV 600,VAR_1$,0 : VIMSG 1, _ERRAD$

' Quale e' l'ID dell'Associazione?
IDASSOC = _ADDVDID(1) : VIMSG 1, STR$(IDASSOC,0,0)

' Cancella la Primanota
DELPN "20210928",0,"FE","V",01,000025,2/39,"20210928" : VIMSG 1, _ERRPN$

'Varia la Primanota (es. aggiungendo un agente con provvigione)
_PNAGE$="602.00001" : _PNPRO=333

'Ri-scrivi la Primanota
PUTPN : VIMSG 1, _ERRPN$

'Ri-Leggi la Primanota per ottenere il NUOVO progressivo
GETPN "20210928",0,"FE","V",01,000025,2/39,"20210928" : VIMSG 1, STR$(PROGR,0,0)
PROGR=_PNPRN

' RI-Associa la Primanota al documento XML
```

```
VAR_2$="20210928;" +STR$(PROGR,0,0)+"FE;V;01;000024;238;20210928"
PUTASSOC DV 600,VAR_2$,IDASSOC,"" : VIMSG 1, _ERRAD$
END
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTBL

Istruzione che inserisce/aggiorna i dati della Bolla di Lavorazione in base alla valorizzazione delle variabili struttura archivi _OB (testata) e _OL (righe). L'istruzione valorizza le variabili _ERRBL e _ERRBL\$. L'istruzione non prevede alcun tipo di automatismo presente nella funzione standard della procedura. La generazione di testata senza dettaglio e senza errori, significa che le variabili di riga non sono state tutte impostate oppure che il numero della riga in produzione per il magazzino non è stato valorizzato. Non si possono scrivere BL collegate a OC o sottobolla; per collegare vedere istruzione COLOCBL.

Sprix non ha gli stessi automatismi di avanzamento della Produzione del Gestionale; è tutto manuale e la procedura di avanzamento va ricostruita, in ogni caso non è possibile creare sottobolle. Fasi:

- leggere i prodotti con distinta base della bolla e delle relative sottobolle (GETBL)
- leggere gli impegni se presenti (GETIL), altrimenti SVILDB
- caricare la struttura in _MM per effettuare SL (PUTMM)
- cancellare gli impegni se presenti (DELIL)
- caricare la struttura in _MM per effettuare CL (PUTMM)
- avanzare la bolla per chiuderla (PUTBL) sulle fasi d'interesse (es. _OLFAS=100 -CL- e altre variabili _OL che sono necessarie)
- dalla v815 è possibile con un terminale Produzione scrivere le variabili della rintracciabilità di riga del movimento di magazzino
- dalla v816 le righe movimentate in conto lavoro non sono modificabili

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTCDG

Istruzione che scrive i dati dei movimenti extra di Contabilità Analitica del Controllo di Gestione in base alla valorizzazione delle variabili struttura archivi _CDG, La PUTCDG con codice commessa inesistente crea una nuova anagrafica commessa con codice e descrizione indicati; nel tipo 'R' avviene l'accodamento delle righe di analitica alle precedenti (l'aggiornamento richiede quindi una preventiva cancellazione). L'istruzione valorizza _ERRCDG e _ERRCDG\$.

Notare: la PUTCDG delle variabili _CDG è specificatamente relativa al modulo [Azienda - Controllo di Gestione - Contabilità Analitica - Inserimento/Revisione movimenti **extra**]; non vengono interessate e/o aggiornate le variabili di altri archivi (esempio _MM, _PN, ...), ovvero i moduli che avessero generato dei movimenti di analitica, pertanto, occorre procedere con le specifiche relative loro istruzioni (PUTMM, PUTPN, ...).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTCFR

Istruzione che scrive i riferimenti dei Referenti presenti nell'anagrafica della rubrica Clienti Fornitori (tasto funzione F4) o Contatto in base alla valorizzazione delle variabili struttura archivi _CFR in relazione ai valori del Cliente/Fornitore <cod\$> in base al <tipo> (se 0 = tutti i referenti, da 1 a 30 = un referente specifico). La scrittura inserisce nuovi referenti o aggiorna i dati di quelli preesistenti, che restano inalterati se non diversamente assegnati dalle variabili Sprix (che quindi possono non essere valorizzate). Se non tutti i referenti sono valorizzati, _CFRPRG() non corrisponde con la posizione dei referenti all'interno della finestra. Il collegamento fra referente registrato e array è stabilito da _CFRPRG(). L'istruzione valorizza le variabili _CFRERR e _CFRERR\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTCM

PUTCM

Istruzione che scrive dati nel Calendario dell'Agenda Attività Utente attraverso la valorizzazione delle variabili di struttura _CM. L'istruzione valorizza le variabili _ERRCM e _ERRCM\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTCOM

PUTCOM <a\$>,<n>

Parametri	
a\$	Buffer dati;

n	Quantità di byte;
---	-------------------

Istruzione che scrive sulla porta seriale aperta, la quantità di <n> byte (da 1 a 255, con default 1 se < 1 o 255 se > 255) prelevati dal buffer dati <a\$>. L'istruzione valorizza la variabile _ERRCOM\$.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTCOMM

Istruzione che scrive/aggiorna i dati di Anagrafica Commessa in base alla valorizzazione delle variabili struttura archivi _CO. L'istruzione valorizza le variabili _ERRCO e _ERRCO\$.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTCON

Istruzione che scrive i dati di Anagrafica Contatti in base alla valorizzazione delle variabili struttura _DC. L'istruzione valorizza le variabili _ERRDC e _ERRDC\$.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTDB

PUTDB <numfase>

Parametri	
numfase	Numero di fase della Distinta Base Primaria;

Istruzione che scrive i dati della Distinta Base Primaria in base alla valorizzazione delle variabili struttura archivi _TP in base al numero di fase <numfase> della DBP (1-99) e del codice articolo con DBP. Le operazioni sulla DBP valorizzano _NCOMP (numero dei componenti di una fase: max 5000 per fase). La scrittura è per numero di fase su articolo esistente (_TPCOD\$) con DBP a una/più fasi. L'istruzione valorizza le variabili _ERRDB e _ERRDB\$.
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTDDOC

Istruzione che inserisce/aggiorna i dati del Documento Docuvision in base alla valorizzazione delle variabili struttura archivi _DV. L'istruzione valorizza le variabili _ERRDV e _ERRDV\$.

Attenzione: la funzionalità è attiva per il modulo "Gestione documento", ma non per il modulo "Fatture elettroniche PA-B2B".

Nota su ALLEGATI

Sprix gestisce gli allegati relativi al Documento di Docuvision (fig. 1), non gestisce gli allegati relativi alla Fattura PA / B2B (fig. 2); il modo per intervenire sul file XML (della fattura elettronica) è consentito all'evento di stampa ON_STP_XML, ma non sono disponibili apposite variabili per codificare l'allegato dentro il file XML poichè occorre un'apposita codifica separata (fig. 2). Si ricorda che altri generici file XML sono gestiti in Sprix tramite apposite istruzioni XML. Per ottenere l'handle del documento in considerazione al fine di eseguirne le opportune modifiche con le istruzioni generali XML, avvalersi della variabile _FTE_DOCXML che consente l'accesso da collage al file _XML della FTE.

Fig. 1 (Allegati DOCUVISION)

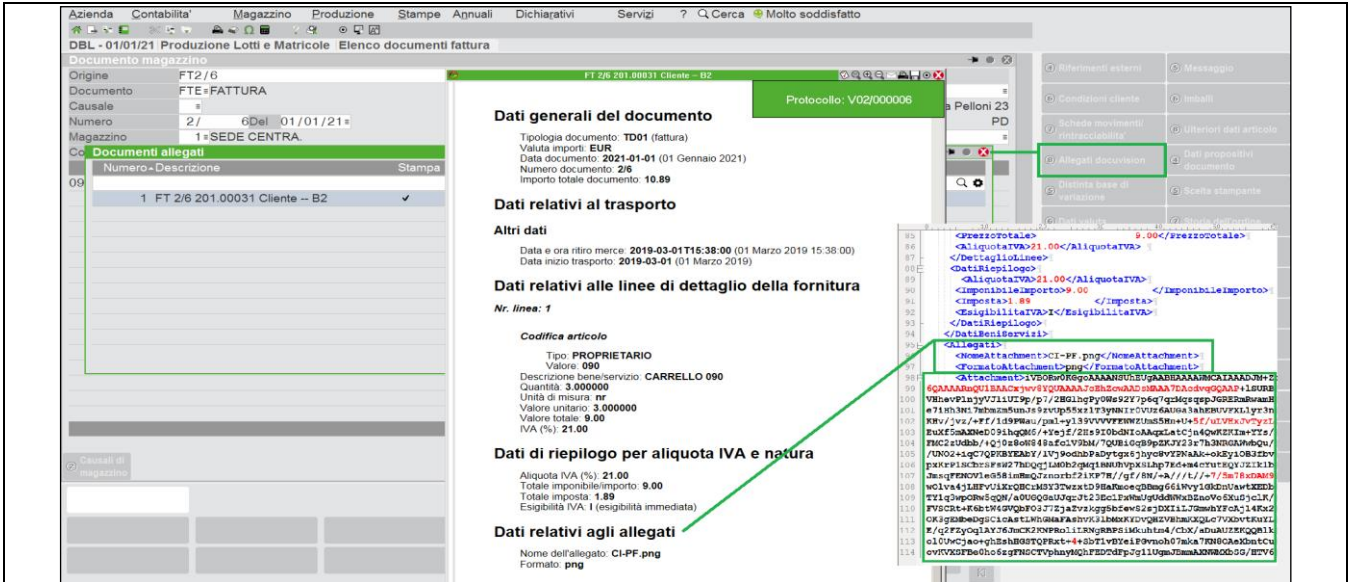
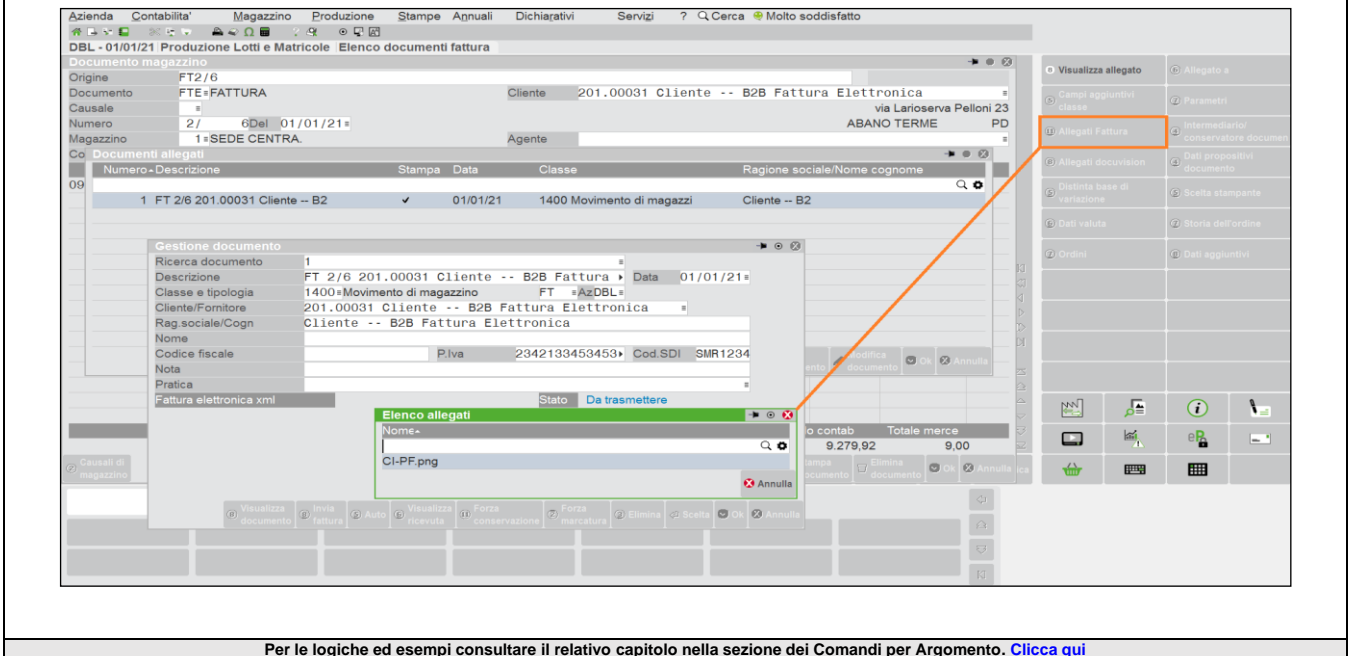


Fig. 2 (Allegati FATTURA)



Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTGM

Istruzione che scrive i dati anagrafici del Gruppo Merceologico in base alla valorizzazione delle variabili struttura archivi _GM. L'istruzione valorizza le variabili _ERRGM e _ERRGM\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTIL

Istruzione che inserisce/aggiorna i dati degli Impegni di Lavorazione in base alla valorizzazione delle variabili struttura archivi _IL. Se in scrittura viene cambiata una _IL? Della chiave della BL, viene inserita una nuova riga in Impegni di Lavorazione, identica a quella di partenza salvo per i valori cambiati; la scrittura non avviene se l'indice del vettore viene aumentato a più 1 per fare un nuovo inserimento; inserendo _ILNRB() = 65534 o 65533 si può inserire un RES_SL o RES_DL. L'istruzione valorizza le variabili _ERRIL e _ERRIL\$.

La scrittura di un impegno è strettamente legato ad una bolla di lavorazione, quindi la bolla di lavorazione deve esistere e occorre conoscere per ogni prodotto finito presente nella bolla i riferimenti univoci del prodotto finito.

Per inserire un impegno occorre dichiarare a quale prodotto finito quest'impegno è legato, il numero di riferimento univoco della bolla di lavorazione, una fase di lavorazione, un magazzino, un codice articolo, una quantità, e il tipo di unità di misura dell'articolo.

Variabile	Valore	Note
_ILNDO(1)	70	

_ILNSD(1)	0	
_ILNRB(1)	1	Se si dispone della struttura della bolla di lavorazione è _OLRll(); Se si utilizza il ciclo tabellare la variabile è olrii();
_ILFAS(1)	1	
_ILMAG(1)	1	
_ILAMP\$(1)	"ABC"	
_ILQTA(1)	20	
_ILTQT(1)	101	

Con questo esempio si sta inserendo per la bolla 70 sottobolla 0, riferimento prodotto finito 1, fase 1, magazzino 1, la materia prima "ABC" in quantità 20, in unità di misura primaria. Se l'impegno che stai inserendo esiste, per la chiave dell'impegno stesso, vengono sostituiti i valori non di chiave, ad esempio la quantità.

Se presente la bolla di lavorazione l'impegno non è soggetto a nessuna operazione. Lo stato dichiarato nella bolla, colonna impegni, è uno stato per capire che gli impegni sono stati fatti, ma da Sprix si può liberamente inserire o modificare dati: per verificare questa generazione dell'impegno è sufficiente creare una BL dal Gestionale e poi leggerne il numero "Rif.PF" presente nel dettaglio della riga della BL e, infine, inserire un impegno per quella BL per quel "Rif.PF".

Si ricorda che l'istruzione PUTIL opera in questo modo: se il campo chiave non varia ed il dato è presente avviene un aggiornamento mentre se non esiste viene scritto un nuovo record; ne consegue che se voglio modificare un dato presente in struttura proveniente da GETIL e il dato da modificare appartiene ad un campo che costituisce la chiave, bisogna procedere alla cancellazione dell'impegno esistente tramite DELIL. I campi che compongono la chiave per gli impegni sono: numero bolla, numero sottobolla, numero riferimento prodotto finito, fase, magazzino, codice articolo e ubicazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTIND

Istruzione che scrive i dati anagrafici degli indirizzi di Spedizione in base alla valorizzazione delle variabili di struttura archivi _DA. L'istruzione valorizza le variabili _ERRDA e _ERRDA\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTLOC

PUTLOC_STR <tipo>,<chiave>,<valore>

PUTLOC_NUM <tipo>,<chiave>,<valore>

PUTLOC_ARRAY <tipo>,<chiave>,<valore>

Parametri	
tipo	Tipo di memorizzazione: "T" = temporaneo, "P"=persistente;
chiave	Chiave dei dati;
valore	Variabile di interesse (stringa, numero, array)

Istruzione SprixMobile relativa al salvataggio dati in aree di memoria. L'istruzione scrive stringa, numero e array. La variabile di riferimento relativa a messaggi di errore è _ERRLOC\$. Esempio:

```
PUTLOC_STR "T", "Cliente", CLI$      (Temporaneo)
GETLOC_STR "T", "Cliente", OLDCLI$
PUTLOC_NUM "P", "Operazioni", NUMOP (Persistente)
STORELOC "P", salva in modo persistente i dati in memoria.
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTLOT

Istruzione che scrive i dati di Anagrafica Lotto in base alla valorizzazione delle variabili struttura archivi _LOT. Se il lotto è presente in archivio sono assunti i dati in archivio: il Tipo controllo carico; Fornitore/Data carico (se lotto caricato da movimento e tipo controllo carico 'F'/D'); dati utente gestiti dal lotto e non nelle variabili Sprix. Se il lotto invece è nuovo sono assunti i dati Sprix o dal default del tipo lotto associato: i dati utente non impostati; il carico movimento magazzino e il tipo controllo carico (non valorizzabili con Sprix); codice articolo che gestisca dei progressivi (se indicato uno che non li gestisce); codice utente lotto viene impostato con i dati personalizzati. Nella scrittura il dato personalizzato si azzera se valorizzato null "". A parità di dati aggiuntivi, la variazione dei soli dati generali comporta la sola scrittura di questi ultimi. Valorizza le variabili _ERRLOT e _ERRLOT\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTLSTDOC

Istruzione che scrive i dati della Lista di Prelievo relativi alla struttura archivi _LST. In fase di creazione lista ritorna nel campo _LSTTID l'id della lista creata. L'istruzione valorizza le variabili _ERRLST e _ERRLST\$. L'istruzione richiede MDS.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTLSTDOC_EXT

PUTLSTDOC_EXT [param\$]

Parametri	
param\$	Parametro facoltativo: impostato a "NOCTLSTATO" vengono esclusi i controlli di stato della testata. N.B: il parametro è ininfluente sulle liste inserite da procedura ZEROODO.

Istruzione che scrive i dati della Lista di Prelievo relativi alla struttura archivi _LST. In fase di creazione lista ritorna nel campo _LSTTID l'id della lista creata. L'istruzione valorizza le variabili _ERRLST e _ERRLST\$. L'istruzione richiede MDS.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTMM

PUTMM <scprn>

Parametri	
scprn	<p>Parametro numerico che consente di stabilire il tipo di registrazione:</p> <p>0 = registra in magazzino; 1 = registra in magazzino e primanota; 2 = registra in magazzino con stampa; 3 = registra in magazzino e primanota, con stampa; 4 = stampa senza registrazioni; 5 = stampa su Registratore Telematico (fa solamente una stampa);</p> <p>10 = come 0 ma relativo a beni usati; 11 = come 1 ma relativo a beni usati; 12 = come 2 ma relativo a beni usati; 13 = come 3 ma relativo a beni usati;</p> <p>14 genera documento magazzino, eventuale CDG; 15 genera documento di magazzino, eventuale registrazione di primanota (solo per registrazioni nuove), eventuale CDG; 16 genera documento di magazzino, eventuale registrazione di primanota (solo per registrazioni nuove), eventuale mov. bene usato (solo per registrazioni nuove), eventuale CDG; 17 come 14 con stampa documento; 18 come 15 con stampa documento; 19 come 16 con stampa documento;</p> <p>Notare: la PUTMM con parametro che esegue anche la stampa, richiede interazione I/O, pertanto non è schedabile in operazioni pianificate di tipo batch oppure con mxrsbatch.exe; per eseguire anche la stampa, impiegare altra routine (oppure altro Sprix) con l'istruzione APRISTAMP configurata per il dispositivo "silent" più opportuno (file, email,...).</p> <p>Per la ristampa consultare la variabile _MMRICSTP presente in fondo all'elenco variabili _MM.</p> <p>Per stampare un documento telematico di tipo (CO, RE, AN) è sufficiente utilizzare la funzione PUTMM con il parametro impostato con uno dei seguenti valori: 2, 3, 5, 12, 13. Notare bene: il valore 5 esegue <u>soltanto</u> la stampa sul registratore telematico. Anche la PUTMM come le altre funzioni di gestione registratore telematico, se è stata chiamata prima la funzione RTAPRI utilizza il registratore telematico aperto, altrimenti utilizza il registratore telematico di default impostato sul Client.</p>

Istruzione che scrive i dati del Documento di Magazzino in base alla valorizzazione delle variabili struttura archivi _MM. L'istruzione valorizza le variabili _ERRMM e _ERRMM\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTPAG

PUTPAG [g\$]

Parametri	
g\$	<p>Parametro stringa facoltativo per stabilire</p> <ul style="list-style-type: none"> • PUTPAG "+G" = modifica anche i dati generali; • PUTPAG "G" = modifica solo i dati generali; <p>L'uso del parametro [g\$] richiede l'esecuzione di una GETPAG 0 oppure una GETPAG di un pagamento esistente, in modo da avere anche i dati generali altrimenti saranno azzerati</p>

Istruzione che scrive i dati del codice pagamento nella tabella pagamenti in base ai valori impostati nelle variabili di struttura _PA?; riporta l'esito dell'operazione nelle variabili di errore _ERRPA e _ERRPA\$. Nota: non è possibile modificare/aggiornare un pagamento, poichè occorre eseguirne la cancellazione ed una nuova scrittura.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTPART

PUTPART

Istruzione che scrive/aggiorna i dati delle Particolarità (Prezzi, Sconti, Provvigioni, Articolo, Aliquota) in base alla valorizzazione delle variabili struttura archivi _PART. La PUTPART di particolarità nuova richiede i dati obbligatori della versione del gestionale in uso; se la particolarità esiste, viene aggiornata. L'istruzione valorizza le variabili _ERRPART e _ERRPART\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTPC

PUTPC

Istruzione che scrive i dati del Conto di Anagrafica Piano dei Conti o Rubrica Clienti Fornitori in base alla valorizzazione delle variabili di struttura archivi _PC. Valorizza le variabili _ERRPC, _ERRPC\$.

La PUTPC del conto nuovo richiede i dati obbligatori della versione del gestionale in uso; la PUTPC di un conto già esistente lo aggiorna (valorizzare sempre tutte le variabili di interesse, altrimenti alcune potrebbero azzerarsi).

Notare: in ambiente a file separati (tecnologia "BP"), per le aziende di gruppo, la PUTPC non è fruibile sui conti diversi da quelli di tipo C/F/B, perché gli unici conti che questa processa sono i clienti, fornitori e banche (cioè quelli propri dell'azienda in questione). Tramite la variabile _PCSOLOPICO\$ = "S/N" (con default = "N") si possono aggiornare tutte le variabili tipiche del conto aziendale anche per i conti di raggruppamento.

I dati della rubrica (RUDT) rimangono invariati. Per i SOLI conti presenti nell'azienda le variabili che possono essere modificate quando la variabile _PCSOLOPICO\$ è impostata = "S", sono contrassegnate con nota (*a) nella tabella variabili di struttura archivi _PC.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTPN

PUTPN

Istruzione che scrive i dati della registrazione di primanota in base alla valorizzazione delle variabili di struttura archivi _PN. Valorizza le variabili _ERRPN, _ERRPN\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTPNX

PUTPNX

Istruzione che scrive i dati della registrazione di primanota in base alla valorizzazione delle variabili di struttura archivi _PN. L'istruzione effettua la conversione Euro \leftrightarrow Lire: in valuta contabile Euro consente di esprimere importi in Lire e viceversa; non consente la registrazione nella medesima valuta oppure in doppia valuta (ma solo in quella contabile). Valorizza le variabili _ERRPN, _ERRPN\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTPU

PUTPU

Istruzione che scrive/aggiorna i dati della Anagrafica Ubicazioni tramite le variabili di struttura _PU

Se _PUID = 0 viene scritta una nuova ubicazione; se l'operazione va a buon fine, la variabile _PUID viene valorizzata con l'ID assegnato alla nuova ubicazione. Se _PUID > 0 viene aggiornata l'ubicazione esistente.

L'istruzione valorizza le variabili di errore _ERRPU e _ERRPU\$. La struttura viene azzerata con AZZVARSYS 72 oppure AZZCATVARSYS "PU"

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTREM

PUTREM_NUM <tag\$>,<numero>

PUTREM_STR <tag\$>,<stringa\$>

PUTREM_ARRAY <tag\$>,<array_di_stringhe\$ / array_di_numeri>

PUTREM_FILE <tag\$>,<percorso_file\$>

Parametri	
tag\$	Tag assegnato;
numero	Variabile numerica;
stringa\$	Variabile alfanumerica;
array stringhe/numeri	Vettore alfanumerico / numerico;
percorso file\$	Directory del file;

Istruzione Sprix Mobile per gestire i valori che dal Mobile arrivano al Server e/o viceversa. L'istruzione valorizza le variabili `_ERRPUTREM$` per la gestione dell'errore ritornato da una qualsiasi PUTREM...
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTREM_DATI

PUTREM_DATI <buff\$>

Parametri	
buff\$	Buffer del risultato per il file dati;

Istruzione di scrittura del buffer del risultato nel file dati <filed\$> del Communicator Locale, relativa all'esito dell'operazione di esecuzione dello Sprix Remoto attraverso l'esecuzione della funzione `SPXREMOTO$()`. L'istruzione non consente interazioni con il terminale o il video dell'utente. Il valore dell'istruzione non deve essere null ("") poiché causa errore nell'esecutore (esempio: PUTREM DATI "").
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTREM_ERR

PUTREMERR <buff\$>

Parametri	
buff\$	Buffer del risultato per il file errori;

Istruzione di scrittura del buffer degli errori nel file errori <filee\$> sul Communicator Locale, relativa all'esito dell'operazione di esecuzione dello Sprix Remoto attraverso l'esecuzione della funzione `SPXREMOTO$()`. L'istruzione non consente interazioni con il terminale o il video dell'utente. Il valore dell'istruzione non deve essere null ("") poiché causa errore nell'esecutore (esempio: PUTREM_ERR "").
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTSC

Istruzione che scrive o cancella registrazioni di primanota al pari di quanto avviene per le istruzioni PUTPN e DELPN

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTSORT

PUTSORT <str\$>

Parametri	
str\$	Stringa da inserire;

Istruzione che inserisce la stringa <str\$> in una lista di record da ordinare. La chiamata della PUTSORT successiva alla GETSORT azzerà tutti record precedentemente inseriti ordinati.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTTAST

PUTTAST <str\$>

Parametri	
str\$	Stringa da inviare alla tastiera;

Istruzione che invia la stringa <str\$> nel buffer della tastiera ad ogni successiva richiesta di input e senza alcuna pressione dei relativi tasti nell'ordine stabilito in <str\$>; la stringa <str\$> può contenere una sequenza di caratteri (es. PUTTAST "Sprix Pass") oppure di uno o più codici <cod> relativi a tasti funzione e in forma #<cod> (es. PUTTAST "#610").

Nella SETTF c'è una tabellina che elenca le combinazioni dei principali tasti funzione (non sono elencate tutte le combinazioni).

Se <str\$> contiene '@' seguito dal nome di una 'ETICHETTA.', sono eseguite le istruzioni a questa successive. Per evitare saturazioni del buffer di tastiera effettuare tante routine.

Per inviare caratteri '#' o '@' basta ripeterli, esempio PUTTAST "Spx1##Spx2@@".

Per evitare la saturazione del buffer di tastiera è consigliabile effettuare tante routine.

[esempio su listini](#)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTT PLOT

Istruzione che scrive i dati di Anagrafica Tipo Lotto relativi alle variabili struttura archivi _TPLOT. L'istruzione valorizza le variabili _ERRTPLOT e _ERRTPLOT\$. Nella scrittura, il dato personalizzato si azzerà se valorizzato null ""; se il Tipo Lotto è presente in archivio, i seguenti dati sono impostati con quelli in archivio:

- se lo stato è 'C' creato viene mantenuto l'ordine dei dati personalizzati immessi precedentemente;
- se è diverso da 'C':
 - lo Stato tipo lotto,
 - la Chiave univoca solo se 'N'
 - è mantenuto l'ordine dei dati personalizzati immessi precedentemente

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

PUTZA

PUTZA [param\$]

Parametri	
param\$	Parametro

Istruzione che scrive i dati Anagrafica Unica relativi alle variabili struttura archivi _ZA. Quando il parametro facoltativo <param\$> non viene specificato o viene impostato a "STOR=N" l'istruzione scrive/aggiorna i valori correnti di Anagrafica Unica; se valorizzato a "STOR=1" oppure "STOR=S" storicizza i cambiamenti in anagrafica sulla base della data di fine storicizzazione impostata nelle apposite variabili indicate nella sezione "Variabili utilizzate per la storicizzazione" presente in calce all'elenco della struttura _ZA; se valorizzato a "STOR=R" aggiorna l'anagrafica corrente e scrive anche lo storico utilizzando come data d'inizio quanto impostato nelle apposite variabili indicate nella sezione "Variabili utilizzate per la storicizzazione" presente in calce all'elenco della struttura _ZA.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

REM

REM

Istruzione che trasforma la riga in commento, da collocare ad inizio riga. In alternativa si può utilizzare l'apice " " posizionabile liberamente e trasforma in commento il testo alla sua destra fino alla riga successiva.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RICCONF

RICCONF <m\$>,<f\$>,[s\$]

Parametri	
m\$	Stringa di messaggio; se vuoto "" il messaggio default è "confermi prima di proseguire?" (con default Invio = No)
f\$	Flag S/N; "S" = default Invio "Si"; "N" = default Invio "No";
s\$	Stringa non attiva (parametro predisposto);

Istruzione che visualizza il messaggio <m\$> di conferma e con default su Invio. Il setting dei parametri imposta _TF con valore 0/1.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RIGHT

RIGHT\$(<str\$>,<numcar>)

Funzione che estrae gli ultimi caratteri <numcar> partendo da destra della stringa <str\$>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RIGMM

RIGMM <numriga>,<modo>,<numdecart>

Parametri	
numriga	Numero di riga valida del documento di magazzino e una testata valida;
modo	Modo di calcolo; se = 1 calcola per progressivi; se = 2 calcola per totale documento; la differenza di metodo è relazionata alle righe in um2 (unità di misura secondaria), ovvero, nel modo 1 si ottengono i valori (ad es. quantità e importo) di riga in unità di misura primaria e in valuta di gestione azienda, mentre invece, nel modo 2 gli stessi valori sono espressi in unità di misura secondaria (se gestita) ed in valuta del documento (se diversa da quella di valuta azienda).
numdecart	Numero di decimali articolo (obbligatorio se <modo>=1);

Istruzione che calcola i totali di riga _MMR se richiamata dopo la valorizzazione delle variabili _MM relative ad un documento valido con un numero di riga valida <numriga> e una testata valida, secondo un modo di calcolo <modo> e un numero di decimali articolo <numdecart>. L'istruzione ha gli stessi vincoli dell'istruzione PUTMM.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RIPTAFCT

RIPTAFCT("par")

Parametri	
par	parametro non ancora definito

Funzione che consente di ripristinare globalmente il setting di tutti i tasti funzione così come erano stati creati per la CREATELIST con l'array _CTTASTFUN\$. All'uscita dal codice di gestione di un tasto funzione, dopo che nel codice di gestione del tasto funzione venga variata la gestione dei tasti funzione per aver aperto una propria finestra, qualunque uso sia stato fatto dei tasti funzione, può ritornare un errore; attualmente il parametro non è ancora gestito.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RTAPRI

RTAPRI [n]

Parametri	
n	Numero registratore telematico da aprire; parametro opzionale, se non indicato o -1 apre il registratore telematico impostato per il Client.

Funzione per aprire il Registratore Telematico. Ritorna eventuali errori nella variabile _ERRRT\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RTAPRICASSETTO

RTAPRICASSETTO

Funzione per il comando apertura cassetto cassa. Ritorna eventuali errori nella variabile _ERRRT\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RTCHIUDI

RTCHIUDI

Funzione per chiudere il Registratore Telematico aperto con RTAPRI. Ritorna eventuali errori nella variabile _ERRRT\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RTCHIUSURAGIO

RTCHIUSURAGIO

Funzione di invio richiesta chiusura giornaliera al Registratore Telematico. La funzione ritorna i dati della chiusura nelle variabili di struttura _RT. Ritorna eventuali errori nella variabile _ERRRT\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RTDISPLAY

RTDISPLAY <str\$>

Parametri	
str\$	Array di 4 stringhe che sono visualizzate sul display nel seguente modo: <ul style="list-style-type: none"> la 1^ str\$ sulla prima riga del display allineata a sinistra la 2^ str\$ sulla prima riga del display allineata a destra (eventualmente sovrascrive la parte allineata a sinistra)

	<ul style="list-style-type: none"> la 3^a str\$ sulla seconda riga del display allineata a sinistra la 4^a str\$ sulla seconda riga del display allineata a destra (eventualmente sovrascrive la parte allineata a sinistra)
--	--

Funzione per invio informazioni sul display del Registratore Telematico. Ritorna eventuali errori nella variabile `_ERRRT$`.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RTELENCODISP

RTELENCODISP

Funzione che restituisce l'elenco dei Registratori Telematici configurati. La funzione ritorna i dati dei registratori telematici configurati nelle variabili di struttura `_RT`. Ritorna eventuali errori nella variabile `_ERRRT$`.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RTGETPAGAMENTI

RTGETPAGAMENTI

Funzione di invio richiesta pagamenti configurati per il Registratore Telematico. La funzione ritorna i dati dei reparti nelle variabili di struttura `_RT`. Ritorna eventuali errori nella variabile `_ERRRT$`.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

RTGETREPARTI

RTGETREPARTI

Funzione di invio richiesta reparti configurati per il Registratore Telematico. La funzione ritorna i dati dei reparti nelle variabili di struttura `_RT`. Ritorna eventuali errori nella variabile `_ERRRT$`.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SALTOPAG

SALTOPAG\$()

Funzione che ritorna una stringa di caratteri da inviare alla stampante per effettuare il salto pagina. Questa istruzione viene eseguita solo se prima è stata lanciata un'altra istruzione LPRINT. Esempio: LPRINT "" : LPRINT SALTOPAG\$().
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SBLOCCASEM

SBLOCCASEM <nomesem\$>

Parametri	
nomesem\$	Nome del semaforo da sbloccare; se "" sono sbloccati tutti i semafori che sono stati bloccati dal proprio terminale;

Istruzione che sblocca il semaforo con nome <nomesem\$> bloccato da BLOCCASEM. La semaforica per l'accesso di elaborazioni Sprix ai dati del gestionale è regolata come gli altri Client (concorrenza sul dato, regole di gestione).
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SCREEN

SCREEN\$(<nfin>,<rig>,<col>,<larg>)

Parametri	
nfin	Numero di finestra aperta (se indicato in negativo viene inteso come ordine di apertura: 0 = base, -1 = la prima aperta, -2 = la seconda aperta, ecc);
rig	Riga a video;
col	Colonna a video;
larg	Larghezza della stringa letta a video;

Funzione che ritorna una stringa di caratteri larga <larg>, letta a video sulla riga <rig> e colonna <col> del numero <nfin> di finestra aperta.

Nota bene: funzione vincolante, si raccomanda l'uso solo in caso di necessità, in quanto soggetta a variazioni in relazione a modifiche dell'interfaccia Gestionale.

Il video può cambiare formato/risoluzione, pertanto **le colonne possono cambiare**. In tal caso è utile identificare l'area finestra disponibile. Per esempio, per leggere il totale merce della riga del corpo del documento, si legge tutta la riga eliminando i blank a partire da destra, poi il dato di interesse fino al precedente blank.


```

ON_612_ALL:
NCOL = NCOLDISPFIN(1,1)
NRIG = NRIGDISPFIN(1)

X$=SCREEN$(_NFIN,NRIG,1,NCOL)
X$=CVA$(MKA$(X$,255),NCOL)
T$=""
FOR X=LEN(X$) TO 1 STEP -1
  Y$=MID$(X$,X,1): IF Y$=" " THEN X=X-1 ELSE IF Y$("<>") THEN T$=Y$+T$
NEXT
TOT=VAL(T$)
VIMSG 4,"X$="+X$+"|T$="+T$
END

```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SCROLL

SCROLL <rig>,<col>,<alt>,<larg>,<dir>

Parametri	
rig	Coordinata numero di riga della finestra attiva;
col	Coordinata numero di colonna della finestra attiva;
alt	Numero di righe in scroll;
larg	Quantità di caratteri in scroll;
dir	Direzione di scroll (1= in alto; -1=in basso; 2=verso sinistra; -2=verso destra).

Istruzione che scrolla nella finestra attiva a partire dall'angolo superiore sinistro con coordinate numero di riga <rig> e numero colonna <col>, per un numero di righe <alt> e di caratteri <larg> verso la direzione <dir>

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETDECFIX

SETDECFIX <ndec>

Parametri	
ndec	Se = -1 il FIX automatico a 6 decimali non viene eseguito; se = 0 – 6 rappresenta il numero dei decimali su cui deve essere eseguito il FIX automatico

Istruzione che in base al valore <ndec> (da -1 a 6) imposta il comportamento della FIX (che altrimenti sarebbe automatico con 6 decimali su tutti i risultati derivanti da calcoli) fino alla fine del programma o fino alla riga precedente del suo riutilizzo. Una chiamata esplicita di FIX ha priorità sul FIX eseguito automaticamente, quindi, anche su quanto impostato con SETDECFIX.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETFSM

SETFSM <nfin>

Parametri	
nfin	Numero finestra Sprix;

Istruzione che abilita nella finestra Sprix <nfin> la gestione del mouse. In Collage l'istruzione vale solo per le finestre della personalizzazione. Vedere anche SETZM e INPUT_EXT. L'istruzione consente di visualizzare l'icona a freccia elenco nella combobox di INPUT_EXT_EXT; attenzione all'eventuale "incompatibilità per sovrapposizione", es: SETTF "602@11Ricerca|Tooltip".

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETMAN

SETMAN <codvoce\$>

Parametri	
codvoce\$	<p>Codice alfanumerico, max lunga 8 caratteri key sensitive (maiuscolo/minuscolo).</p> <p>Il codice attiva la corrispondente voce del manuale utente relativo alla personalizzazione, visibile all'utente tramite la pressione del tasto funzione "Aiuto" [F1].</p> <p>Il codice è la chiave che collega il programma alla porzione dell'help in linea, editabile nel Editor Sprix/Collage con tasto funzione [Ctrl+F9], per un massimo di 250 voci.</p> <p>Il codice è richiamabile anche con variabile (es. V\$="voce1234": SETMAN V\$: INPUT.... : INPUT ...), valida per gli Input</p>

a seguire fino ad azzeramento SETMAN "" oppure ulteriore nuova SETMAN.

Con l'utilizzo di #APPENDISPX la definizione del <codvoce\$> va inserita nel codice sorgente e non nella Libreria.

Istruzione che consente, tramite tasto funzione "Aiuto" [F1] in campo di Input, di richiamare la <codvoce\$> corrispondente alla voce che è contenuta nel help in linea della personalizzazione e che è personalizzabile con il tasto funzione [Ctrl+F9] presente nell'Editor Sprix/Collage. L'elenco di voci aperto con [Ctrl+F9] parte sempre dalla prima voce libera. Ciascuna voce può contenere fino a 9000 caratteri testo, editabili tramite un semplice Editor dotato dei più comuni comandi (Canc, Undo, Back, Ins, Freccie di navigazione/selezione, Copia, Incolla, Taglia, Undo, Ricerca,...). Non sono disponibili comandi più avanzati (gestione font, colori, tabulazioni, elenchi, link, ecc..). La pressione di [Ctrl+F9] su SETMAN apre l'elenco alla voce indicata e se questa non esiste viene proposta in codifica. Le voci sono modificabili e cancellabili.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETNUM

SETNUM <TipoSigla\$>,<SiglaDoc\$>,<Anno>,<Serie>,<Data\$>,<Numeratore>

Parametri	
Tiposigla\$	Tipo di area: "C" = contabile, "M" = magazzino;
SiglaDoc\$	Tipo di sezionale A/V/S/C/R/X/F se l'area è "C" o la sigla del documento di magazzino se l'area è "M";
Anno	Anno solare di riferimento;
Serie	Numero di serie (1 – 99);
Data\$	Data di aggiornamento del numeratore;
Numeratore	Numeratore da 1 a 999999;

Istruzione che imposta il numeratore relativo all'area del <SiglaDoc\$> riferito all'anno solare <anno> e al numero di <Serie>. L'istruzione valorizza l'esito dell'operazione nella variabile _ERRNUM\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETPARCREAZMATR

SETPARCREAZMATR <str\$>

Parametri	
str\$	Parametro creazione matrice: "N" non viene creata; "S" viene creata (ha priorità rispetto al parametro impostato in anagrafica azienda).

Istruzione per la gestione del messaggio Creazione Matrice in modo prioritario rispetto a quanto è possibile impostare nel Gestionale con il parametro di magazzino "Creazione automatica ordine matrice" (in automatico o su richiesta). La scrittura di un ordine, infatti, prevede la possibilità di creare in automatico la matrice oppure con eventuale messaggio di richiesta, ma controllabile con tale istruzione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETPARSCRDOCNM

SETPARSCRDOCNM <str\$>

Parametri	
str\$	Parametro forzatura modifica documento; se vale "S" attiva la forzatura per modificare prima della scrittura

Istruzione che setta i parametri di scrittura del documento non modificabile. Inizializzando SETPARSCRDOCNM "S" prima della scrittura, l'istruzione forza la scrittura del documento elettronico quando questo non è più modificabile, altrimenti ci sarebbe un blocco con il rilascio del messaggio "Il movimento ha generato una fattura elettronica xml che risulta già trasmessa,! il documento non e' modificabile".

Notare: ovviamente il movimento aveva generato una fattura elettronica XML già trasmessa, pertanto le eventuali variazioni forzate sono memorizzate solo sul documento di magazzino e non sul file XML !

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETPARSOSTMATR

SETPARSOSTMATR <str\$>

Parametri	
str\$	Parametro sostituzione matrice: "N" non sostituita; "S" viene sostituita (ha priorità rispetto al parametro impostato in anagrafica azienda).

Istruzione per la gestione del messaggio Sostituzione Matrice in modo prioritario rispetto a quanto è possibile impostare nel Gestionale con il parametro di magazzino "Sostituzione automatica ordine matrice" (in automatico o su richiesta). La scrittura di un ordine già esistente, infatti, prevede la possibilità di sostituire in automatico la matrice oppure con eventuale messaggio di richiesta, ma controllabile con tale istruzione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETPRFIN

SETPRFIN <numfin>,<modo>

Parametri	
numfin	Numero di finestra attiva aperta (da 1-9);
modo	Modalità di visualizzazione: 0 = non proporzionale; 1 = proporzionale allineato a sinistra; 2 = proporzionale allineato a destra; 3 = proporzionale centrato

Istruzione che setta la modalità <modo> di visualizzazione nel numero <numfin> finestra attiva aperta con APRIFIN nell'interfaccia grafica

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETTATAST

SETTATAST <tast\$>,[ag\$]

Parametri	
tast\$	Nome della tastiera personalizzata creata da apposito menu del gestionale Servizi – Personalizzazioni – Tastiere; non può iniziare con “_”, deve essere nuovo e diverso dalla predefinita, anche se copiata/personalizzata.
ag\$	Modalità aggiuntiva della tastiera personalizzata in Collage: se il parametro [ag\$] = “a” la SETTATAST aggiunge la tastiera personalizzata a quella già presente sul campo; se il parametro [ag\$] = “” la SETTATAST opera in modalità sostitutiva della tastiera personalizzata al posto di quella del campo.

Istruzione che attiva setta la tastiera personalizzata dal nome <tast\$> creata da apposito menu del gestionale, per essere utilizzata in modo prioritario (viene sovrascritta) rispetto a quelle predefinite (riservate e che iniziano con il carattere ‘_’) e quelle personali. L'istruzione è consentita solo nella modalità di interfaccia “Extend” (modalità “Touch” con “visibilità area tastiere”).

In ambiente Sprix il settaggio è prima dell'istruzione INPUT con azzeramento subito dopo. La pressione del pulsante di <tast\$> copia il contenuto del campo Codice di <tast\$> nel campo di INPUT e ne esce in automatico se impostato anche il campo Tasto (501 o 602).

In ambiente Collage il settaggio è all'evento ON_IN_? Del campo di input con azzeramento all'evento ON_OUT_? Per consentire un altro caricamento. In Collage si può richiamare una tastiera nuove o predefinita personalizzata dall'utente, in cui definire pulsanti che richiamano specifiche etichette *anche se non sono caricate da uno specifico evento*, ma le regole dell'esecuzione restano immutate. Si consiglia una sola etichetta per singolo pulsante. Il richiamo all'etichetta è attraverso il campo Codice della <tast\$> in cui definire “ESEGUICOLLAGE:” seguito dal nome dell'etichetta da richiamare quando è premuto il pulsante.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETTF

SETTF <str\$>

Parametri																																																							
str\$	<p>Stringa di tasti funzione ed eventuali icone e descrizioni, separati dal carattere “,”.</p> <p>Sintassi della stringa <str\$>: “<cod>[@idico][desc], , <cod>[@idico][desc]”</p> <p>Nota: se il tasto e' stato dichiarato in apertura finestra [idico] e [desc] non vengono considerati</p> <table border="1" data-bbox="327 1534 1460 2105"> <thead> <tr> <th><cod></th> <th colspan="2">Codice numerico del tasto funzione</th> </tr> </thead> <tbody> <tr> <td></td> <td colspan="2">ATTENZIONE: la seguente tabella non riporta tutte le combinazioni possibili eseguibili con PUTTAST, ma elenca solo i tasti funzione che sono disponibili nella SETTF. I codici delle combinazioni non dichiarate in questa tabella significa che non sono utilizzabili in quanto riservati alle funzioni standard del gestionale. Per esempio: il tasto 634 (Ctrl+F10) è disponibile in PUTTAST, ma non in SETTF</td> </tr> <tr> <td>501 – Invio</td> <td>526 – Prima riga [Ctrl+Home]</td> <td>618 – Sh+F6</td> </tr> <tr> <td>502 – Back Space</td> <td>527 – Ultima riga [Ctrl+Fine]</td> <td>619 – Sh+F7</td> </tr> <tr> <td>503 – Tab</td> <td>528 – Shift+Canc</td> <td>620 – Sh+F8</td> </tr> <tr> <td>504 – Shift+Tab</td> <td>529 – Shift+PagSu</td> <td>623 – Sh+F11</td> </tr> <tr> <td>505 – Esc</td> <td>530 – Shift+PagGiù</td> <td>624 – Sh+F12</td> </tr> <tr> <td>506 – Ins</td> <td>537 – Ctrl+Sinistra</td> <td>625 – Ctrl+F1</td> </tr> <tr> <td>507 – Inizio riga [Home]</td> <td>538 – Ctrl+Destra</td> <td>626 – Ctrl+F2</td> </tr> <tr> <td>508 – Canc</td> <td>539 – Ctrl+PagSu</td> <td>627 – Ctrl+F3</td> </tr> <tr> <td>509 – Fine riga [Fine]</td> <td>540 – Ctrl+PagGiù</td> <td>628 – Ctrl+F4</td> </tr> <tr> <td>510 – Pag. indietro</td> <td>602 – F2</td> <td>629 – Ctrl+F5</td> </tr> <tr> <td>511 – Pag. avanti</td> <td>603 – F3</td> <td>630 – Ctrl+F6</td> </tr> <tr> <td>512 – Freccia su</td> <td>604 – F4</td> <td>631 – Ctrl+F7</td> </tr> <tr> <td>513 – Freccia giù</td> <td>605 – F5</td> <td>632 – Ctrl+F8</td> </tr> <tr> <td>514 – Freccia sinistra</td> <td>606 – F6</td> <td>633 – Ctrl+F9</td> </tr> <tr> <td>515 – Freccia destra</td> <td>607 – F7</td> <td>635 – Ctrl+F11</td> </tr> <tr> <td>516 – Shift+Su</td> <td>608 – F8</td> <td>636 – Ctrl+F12</td> </tr> </tbody> </table>	<cod>	Codice numerico del tasto funzione			ATTENZIONE: la seguente tabella non riporta tutte le combinazioni possibili eseguibili con PUTTAST, ma elenca solo i tasti funzione che sono disponibili nella SETTF. I codici delle combinazioni non dichiarate in questa tabella significa che non sono utilizzabili in quanto riservati alle funzioni standard del gestionale. Per esempio: il tasto 634 (Ctrl+F10) è disponibile in PUTTAST, ma non in SETTF		501 – Invio	526 – Prima riga [Ctrl+Home]	618 – Sh+F6	502 – Back Space	527 – Ultima riga [Ctrl+Fine]	619 – Sh+F7	503 – Tab	528 – Shift+Canc	620 – Sh+F8	504 – Shift+Tab	529 – Shift+PagSu	623 – Sh+F11	505 – Esc	530 – Shift+PagGiù	624 – Sh+F12	506 – Ins	537 – Ctrl+Sinistra	625 – Ctrl+F1	507 – Inizio riga [Home]	538 – Ctrl+Destra	626 – Ctrl+F2	508 – Canc	539 – Ctrl+PagSu	627 – Ctrl+F3	509 – Fine riga [Fine]	540 – Ctrl+PagGiù	628 – Ctrl+F4	510 – Pag. indietro	602 – F2	629 – Ctrl+F5	511 – Pag. avanti	603 – F3	630 – Ctrl+F6	512 – Freccia su	604 – F4	631 – Ctrl+F7	513 – Freccia giù	605 – F5	632 – Ctrl+F8	514 – Freccia sinistra	606 – F6	633 – Ctrl+F9	515 – Freccia destra	607 – F7	635 – Ctrl+F11	516 – Shift+Su	608 – F8	636 – Ctrl+F12
<cod>	Codice numerico del tasto funzione																																																						
	ATTENZIONE: la seguente tabella non riporta tutte le combinazioni possibili eseguibili con PUTTAST, ma elenca solo i tasti funzione che sono disponibili nella SETTF. I codici delle combinazioni non dichiarate in questa tabella significa che non sono utilizzabili in quanto riservati alle funzioni standard del gestionale. Per esempio: il tasto 634 (Ctrl+F10) è disponibile in PUTTAST, ma non in SETTF																																																						
501 – Invio	526 – Prima riga [Ctrl+Home]	618 – Sh+F6																																																					
502 – Back Space	527 – Ultima riga [Ctrl+Fine]	619 – Sh+F7																																																					
503 – Tab	528 – Shift+Canc	620 – Sh+F8																																																					
504 – Shift+Tab	529 – Shift+PagSu	623 – Sh+F11																																																					
505 – Esc	530 – Shift+PagGiù	624 – Sh+F12																																																					
506 – Ins	537 – Ctrl+Sinistra	625 – Ctrl+F1																																																					
507 – Inizio riga [Home]	538 – Ctrl+Destra	626 – Ctrl+F2																																																					
508 – Canc	539 – Ctrl+PagSu	627 – Ctrl+F3																																																					
509 – Fine riga [Fine]	540 – Ctrl+PagGiù	628 – Ctrl+F4																																																					
510 – Pag. indietro	602 – F2	629 – Ctrl+F5																																																					
511 – Pag. avanti	603 – F3	630 – Ctrl+F6																																																					
512 – Freccia su	604 – F4	631 – Ctrl+F7																																																					
513 – Freccia giù	605 – F5	632 – Ctrl+F8																																																					
514 – Freccia sinistra	606 – F6	633 – Ctrl+F9																																																					
515 – Freccia destra	607 – F7	635 – Ctrl+F11																																																					
516 – Shift+Su	608 – F8	636 – Ctrl+F12																																																					

		<table border="1"> <tr><td>517 – Shift+Giù</td><td>610 – F10</td><td></td></tr> <tr><td>518 – Shift+Sinistra</td><td>611 – F11</td><td></td></tr> <tr><td>519 – Shift+Destra</td><td>612 – F12 (solo Collage)</td><td></td></tr> <tr><td>520 – Shift+Back Space</td><td>615 – Sh+F3</td><td></td></tr> <tr><td>521 – Shift+Home</td><td>616 – Sh+F4</td><td></td></tr> <tr><td>522 – Shift+Fine</td><td>617 – Sh+F5</td><td></td></tr> </table>	517 – Shift+Giù	610 – F10		518 – Shift+Sinistra	611 – F11		519 – Shift+Destra	612 – F12 (solo Collage)		520 – Shift+Back Space	615 – Sh+F3		521 – Shift+Home	616 – Sh+F4		522 – Shift+Fine	617 – Sh+F5																																															
517 – Shift+Giù	610 – F10																																																																	
518 – Shift+Sinistra	611 – F11																																																																	
519 – Shift+Destra	612 – F12 (solo Collage)																																																																	
520 – Shift+Back Space	615 – Sh+F3																																																																	
521 – Shift+Home	616 – Sh+F4																																																																	
522 – Shift+Fine	617 – Sh+F5																																																																	
[idico]	[parametro facoltativo] - Eventuale ID dell'icona del pulsante;	<p>Nota bene: ulteriori codici icona sono visibili con il pulsante "Icône Video" [Shift+F7]</p> <table border="1"> <thead> <tr> <th>ID</th> <th>DESCRIZIONE</th> </tr> </thead> <tbody> <tr><td>1</td><td>"PALLINO_GIALLO"</td></tr> <tr><td>2</td><td>"PALLINO_ROSSO"</td></tr> <tr><td>3</td><td>"PALLINO_VERDE"</td></tr> <tr><td>4</td><td>"PALLINO_BLU"</td></tr> <tr><td>5</td><td>"PALLINO_CIANO"</td></tr> <tr><td>6</td><td>"PALLINO_MAGENTA"</td></tr> <tr><td>7</td><td>"PALLINO_GRIGIO"</td></tr> <tr><td>8</td><td>"PALLINO_BIANCO"</td></tr> <tr><td>9</td><td>"PALLINO_NERO"</td></tr> <tr><td>10</td><td>"PALLINO_BLU_I"</td></tr> <tr><td>11</td><td>"LENTE_DI_INGRANDIMENTO"</td></tr> <tr><td>12</td><td>"CESTINO"</td></tr> <tr><td>13</td><td>"MATITA"</td></tr> <tr><td>14</td><td>"SIMBOLO_DI_SPUNTA"</td></tr> <tr><td>15</td><td>"SIMBOLO_X_NERO"</td></tr> <tr><td>16</td><td>"LUCCHETTO"</td></tr> <tr><td>17</td><td>"LUCCHETTO_APERTO"</td></tr> <tr><td>18</td><td>"OK"</td></tr> <tr><td>19</td><td>"ANNULLA"</td></tr> <tr><td>20</td><td>"IMBUTO"</td></tr> <tr><td>21</td><td>"ROTELLA"</td></tr> <tr><td>22</td><td>"PALLINO_GIALLO_ESCLAMATIVO"</td></tr> <tr><td>23</td><td>"PALLINO_ROSSO_ESCLAMATIVO"</td></tr> <tr><td>24</td><td>"PALLINO_ROSSO_X"</td></tr> <tr><td>25</td><td>"SIMBOLO_PIU_ROSSO"</td></tr> <tr><td>26</td><td>"FRECCIA_DESTRA"</td></tr> <tr><td>27</td><td>"FRECCIA_SINISTRA"</td></tr> <tr><td>28</td><td>"CALENDARIO"</td></tr> <tr><td>29</td><td>"INTERNET"</td></tr> <tr><td>30</td><td>"BUSTA"</td></tr> <tr><td>Ecc...</td><td>ulteriori codici icona sono visibili con il pulsante "Icône Video" [Shift+F7]</td></tr> </tbody> </table> <p>I comandi che utilizzano gli ID o le DESCRIZIONI delle Icone sono i seguenti:</p> <ul style="list-style-type: none"> • APRIFIN_EXT (ID) • Pulstanti Finestra (ID) • Tasti Funzione (ID) • PRINTICONA ("DESCRIZIONE") • CREATELIST ("DESCRIZIONE") 	ID	DESCRIZIONE	1	"PALLINO_GIALLO"	2	"PALLINO_ROSSO"	3	"PALLINO_VERDE"	4	"PALLINO_BLU"	5	"PALLINO_CIANO"	6	"PALLINO_MAGENTA"	7	"PALLINO_GRIGIO"	8	"PALLINO_BIANCO"	9	"PALLINO_NERO"	10	"PALLINO_BLU_I"	11	"LENTE_DI_INGRANDIMENTO"	12	"CESTINO"	13	"MATITA"	14	"SIMBOLO_DI_SPUNTA"	15	"SIMBOLO_X_NERO"	16	"LUCCHETTO"	17	"LUCCHETTO_APERTO"	18	"OK"	19	"ANNULLA"	20	"IMBUTO"	21	"ROTELLA"	22	"PALLINO_GIALLO_ESCLAMATIVO"	23	"PALLINO_ROSSO_ESCLAMATIVO"	24	"PALLINO_ROSSO_X"	25	"SIMBOLO_PIU_ROSSO"	26	"FRECCIA_DESTRA"	27	"FRECCIA_SINISTRA"	28	"CALENDARIO"	29	"INTERNET"	30	"BUSTA"	Ecc...	ulteriori codici icona sono visibili con il pulsante "Icône Video" [Shift+F7]
ID	DESCRIZIONE																																																																	
1	"PALLINO_GIALLO"																																																																	
2	"PALLINO_ROSSO"																																																																	
3	"PALLINO_VERDE"																																																																	
4	"PALLINO_BLU"																																																																	
5	"PALLINO_CIANO"																																																																	
6	"PALLINO_MAGENTA"																																																																	
7	"PALLINO_GRIGIO"																																																																	
8	"PALLINO_BIANCO"																																																																	
9	"PALLINO_NERO"																																																																	
10	"PALLINO_BLU_I"																																																																	
11	"LENTE_DI_INGRANDIMENTO"																																																																	
12	"CESTINO"																																																																	
13	"MATITA"																																																																	
14	"SIMBOLO_DI_SPUNTA"																																																																	
15	"SIMBOLO_X_NERO"																																																																	
16	"LUCCHETTO"																																																																	
17	"LUCCHETTO_APERTO"																																																																	
18	"OK"																																																																	
19	"ANNULLA"																																																																	
20	"IMBUTO"																																																																	
21	"ROTELLA"																																																																	
22	"PALLINO_GIALLO_ESCLAMATIVO"																																																																	
23	"PALLINO_ROSSO_ESCLAMATIVO"																																																																	
24	"PALLINO_ROSSO_X"																																																																	
25	"SIMBOLO_PIU_ROSSO"																																																																	
26	"FRECCIA_DESTRA"																																																																	
27	"FRECCIA_SINISTRA"																																																																	
28	"CALENDARIO"																																																																	
29	"INTERNET"																																																																	
30	"BUSTA"																																																																	
Ecc...	ulteriori codici icona sono visibili con il pulsante "Icône Video" [Shift+F7]																																																																	
[desc]	[parametro facoltativo] - Eventuale Descrizione personalizzata (max 20 caratteri) e Tooltip del pulsante (separati da ' '). Nota: l'utilizzo di caratteri speciali può essere limitato, per esempio il carattere "\$" non è consentito, in quanto è un carattere riservato.																																																																	

Istruzione che setta i tasti funzione dichiarati nel parametro <str\$> nella forma "<cod>[@idico][desc], , <cod>[@idico][desc]". Ciascun tasto funzione è visualizzato con eventuale Icona, Descrizione e Tooltip nella barra dei tasti funzione attivi del gestionale.

Esempio: "505Uscita,508,501Proseguì".
Esempio: "505@19Uscita|Abbandonare,508,501Proseguì".

Nelle videate personalizzabili (riportabili e non, aggiuntive) il setting e controllo dei tasti funzione è limitato al set di tasti attivi nella videata, verificabili da variabile _TF; Collage può utilizzare il tasto funzione F12.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SETZM

SETZM <riga>,<colonna>,<lungzonamouse>,<numzonamouse>

Parametri	
riga	Numero di riga finestra;
colonna	Numero di colonna finestra
lungzonamouse	Lunghezza campo zona mouse;
numzonamouse	Numero della zona mouse; vale da 1-127 (0 = disabilita la zona mouse; -1 = disabilita tutte le zone mouse della finestra);

Istruzione che abilita la gestione del mouse nell'input a scorrimento in una determinata zona mouse con rilascio di un determinato codice numero zonamouse <numzonmouse>. In Collage l'istruzione vale solo per le finestre della personalizzazione. Vedere anche istruzione SETFSM, INPUT_EXT e le variabili _TMZ, _TMR, _TMC.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SEZIONEFIN

SEZIONEFIN <rig>,<col>,<alt>,<larg>,[tit\$]

Parametri	
rig	Riga inizio sezione (da correggere con valore -1)
col	Colonna inizio sezione (da correggere con valore -1)
alt	Altezza sezione (da correggere con valore +2)
larg	Larghezza sezione (da correggere con valore +2)
tit\$	Titolo della sezione, occupa la prima riga della sezione; il parametro è opzionale, se vuoto (senza titolo)

Istruzione che disegna delle sezioni nella finestra aperta. Occorre considerare le correzioni numeriche riportate da ciascun parametro delle dimensioni e posizionamento.

Esempio, per definire una "Sezione 1" che inizi alla riga 1 e colonna 1, alta 10 righe e larga 51 caratteri si imposta la seguente SEZIONEFIN (1-1),(1-1),(10+2),(51+2),"Sezione 1", ovvero SEZIONEFIN 0,0,12,53,"Sezione 1".

Per le sezioni **affiancate** è **obbligatorio** lasciare uno spazio tra le stesse:
SEZIONEFIN 00,00,12,52,"Sezione 1" : SEZIONEFIN 00,52,11,51,"Sezione 2"
La prima sezione finisce alla colonna 51 (52 - 1) e la seconda inizia alla 52.

Per le sezioni **sovrapposte** è possibile **non** lasciare spazi tra le stesse, ma solo se le sezioni hanno la stessa larghezza:
SEZIONEFIN 12,00,5,103,"Sezione 3" : SEZIONEFIN 16,00,6,103,"Sezione 4"
La prima sezione finisce alla riga 16 (12 + 5 - 1) e la seconda inizia alla stessa riga. In tal caso in un'unica riga vengono disegnate sia la riga di chiusura della prima sezione che la riga di apertura della seconda sezione.

E' anche possibile lasciare una (sola) riga vuota ed è anzi necessario (come nell'esempio) tra le Sezioni 1 e 2 e la Sezione 3. Per avere una maggiore armonia grafica, si consiglia di lasciare sempre una riga di separazione. Per cancellare le sezioni avvalersi dell'istruzione CLS_EXT.

Esempio:

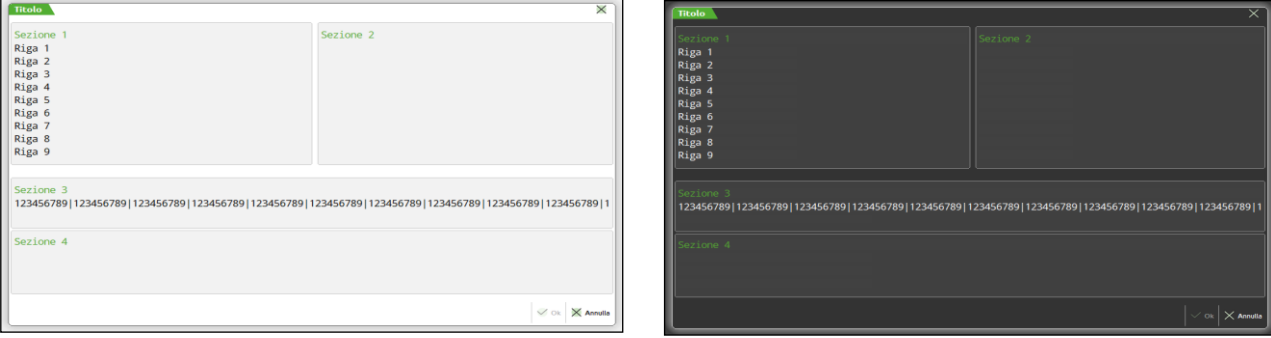
```
ALT=20 : LARG=101
W=AUTOFIN():
```

```
APRIFIN_EXT W,0,0,0,ALT,LARG,"Titolo","A505,A610","NOVISEZIONE"
ATTIVAFIN W
SEZIONEFIN 00,00,12,52,"Sezione 1"
SEZIONEFIN 00,52,12,52,"Sezione 2"
SEZIONEFIN 12,00,5,103,"Sezione 3"
SEZIONEFIN 16,00,6,103,"Sezione 4"
```

```
FOR R=1 TO 9
  LOCATE R+1,01: PRINT "Riga "+STR$(R,0,0);
NEXT
```

```
FOR C=1 TO 101
  LOCATE 14,C: PRINT STRING$(C MOD 10=0,"|") + STRING$(C MOD 10<>0, STR$(C MOD 10,0,0));
NEXT
```

```
T=INCAR(""):
CHIUDIFIN W
END
```



Nota: Il risultato in chiaro oppure scuro dipende dalle impostazioni del Gestionale (schema colori nell'aspetto dell'interfaccia)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SHELL

SHELL <comando\$>

Parametri	
comando\$	<p>Stringa di comando <comando\$> da eseguire a livello di sistema operativo.</p> <p>Per consentire la prosecuzione del programma dopo l'esecuzione SHELL, sistemi Microsoft Windows richiedono lo "START" nel <comando\$>; senza lo "START" o con l'uso di "START /WAIT" (o "START /W" nei sistemi NT) il programma attende il termine. Es: dato un file.bat installato nella directory "C:\\" (con un fittizio ping www.passepartout.net per testare la bontà della connessione ISP oppure con un semplice comando help) e data la variabile CMD\$="START C:\file.bat", la SHELL CMD\$ esegue senza fermarsi a differenza di quanto farebbe con CMD\$="START /W C:\file.bat".</p> <p>Nel sistema Client/Server non local host l'elaborazione sul Client remoto (per esempio l'installazione Live con Server in una Server Farm) è consentita tramite il carattere di reindirizzamento '<' da associare al <comando\$> (SHELL "<START /W C:\file.bat"); il default del Server si forza su Client anche con "*" (es. SHELL "**NOTEPAD"). In Linux il gestionale non reindirizza i comandi (<,>,>) e per eseguirli occorre costruire una shell nella directory prog dell'installazione da lanciare con un'altra SHELL. Ad esempio SHELL "sprixsh "+A\$ (dove A\$ è il comando desiderato)</p>

Istruzione che esegue la stringa di comando <comando\$> a livello di sistema operativo. Notare: l'esecuzione richiede accorgimenti in base al tipo di sistema operativo e nel rispetto dei requisiti di configurazione e sistemistici indicati nel manuale generale sez. "Nozioni Generali". Al termine dell'esecuzione del <comando\$> il controllo ritorna al gestionale dal punto dell'esecuzione, anche senza attendere il termine dell'elaborazione del <comando\$> in base a cosa si sta eseguendo, per esempio, a differenza di una SHELL "notepad.exe" l'esecuzione di A\$="C:\nuovacartella\" : SHELL "mkdir "+A\$ non richiede la chiusura di una finestra attivata dal comando; a tal proposito di ricorda che per la gestione di file e directory è disponibile l'istruzione INPUT_EXT.

Sistemi Windows 2000 e XP possono richiedere l'aggiunta nell'elenco di directory (in pannello di controllo – sistema – avanzate – variabili d'ambiente variabili di sistema) le directory %SystemRoot%\system32 e %SystemRoot%\system32\cmd.exe. Per eseguire programmi Windows (in VB o altro) utilizzare la SHELL con il percorso e il nome dell'eseguibile. L'unico modo di comunicare con un altro applicativo è l'utilizzo dei files.

Si possono settare i parametri del programma da lanciare, se sono settabili anche fuori dal gestionale. Per la definizione di directory complesse in forma abbreviata (esempio SHELL "c:\PROGRA~1\MICROS~2\OFFICE10\WINWORD.EXE") il carattere '~' si codifica con CHR\$(ALT+126 su comuni editor di testo).

Si ricorda che l'esecuzione di ulteriore Sprix avviene con l'istruzione SHELLSPRIX e in Collage l'esecuzione di un menu avviene con la istruzione ESECMENU.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SHELLSPRIX

SHELLSPRIX <comandospx\$>

Parametri <comandospx\$>	
[<NomeDft>][<t<NumTerm>][<o<NumOper>][<x<Data>][<a<SiglaAz>][<codSottAz\$>][<k=<Ute:Pwd> o -k<Chiave>]-p[#]<Sprix>[-v<ArgCmd>]	
<NomeDft>	Rappresenta il file di definizione del terminale in uso e senza estensione ".dft"
<t<NumTerm>	Rappresenta il numero di terminale da aprire, nella gestione con AUTOTERM, SHELLSPRIX apre un terminale tipo servizio, altrimenti occorre disporre di multiterminalità dove <NumTerm> assume valore numerico da 0 al massimo dichiarato nella configurazione dell'installazione; se = 0 collega al primo disponibile, se = -2 cerca il primo terminale aggiuntivo libero, se = -3 cerca in modo misto, prima sugli aggiuntivi e poi quelli definiti in configurazione);
<o<NumOper>	Rappresenta il numero di operatore in luogo di quello di default, 0 si collega al primo disponibile;
<x<Data>	Rappresenta la data odierna di apertura terminale (se assente è assunta l'ultima utilizzata dal terminale o in uso, se = 2 è la data di sistema, se = 3 è l'ultima utilizzata dal terminale, GGMMAAAA è una specifica;
<a<SiglaAz>	Rappresenta la sigla azienda da aprire, se assente è assunta quella del terminale che invoca SHELLSPRIX (<codSottAz\$> indica l'eventuale sotto azienda, v. APRIAZIE);
<k=<Ute:Pwd>	Rappresentano le credenziali di accesso agli archivi se attiva la gestione estesa degli utenti (se assente sono assunte quelle del terminale che invoca SHELLSPRIX) oppure la chiave d'accesso -k<Chiave> se non è attiva la gestione

	estesa (in entrambe i casi lo sprix da eseguire deve essere dichiarato a menù); Nota: il parametro -k può essere scritto anche nella seguente forma -k<Ute:Pwd>.
-p<Sprix>	<p>Unico parametro obbligatorio, che rappresenta lo Sprix da eseguire (da un Collage oppure da un altro Sprix)</p> <p>Il contenuto <Sprix> richiede una determinata sintassi a seconda del contesto di esecuzione; la sintassi ha ricevuto, infatti, varie modifiche negli anni a seconda del grado di evoluzione dell'ambiente di sviluppo.</p> <ul style="list-style-type: none"> Il metodo originario e "tradizionale" consisteva nell'indicare il <u>numero di posizione</u> del programma Sprix da eseguire fra quelli elencati in un determinato menu e <u>con prefisso "#"</u> (per esempio, se lo Sprix occupava la terza posizione del menu "Azienda" la sintassi era "-p#203") oppure consisteva nell'indicare il <u>nome del file da eseguire, ma senza estensione e senza il prefisso "#"</u>, (per esempio "-pspx203"), eccetto nel caso della gestione utenti <u>non attivata</u>, poiché era sufficiente che l'eseguibile fosse nella directory Sprix dell'azienda aperta. In ragione dell'evoluzione dell'ambiente di sviluppo (PassBuilder) questi metodi "tradizionali" sono in via di abbandono, ma sono attivi per consentire l'esecuzione di Sprix non appartenenti ad App generate con PassBuilder, ovvero Sprix non migrati nell'ambiente PassBuilder e che sono ancora elencati e numerati per ciascun menu. Il metodo relativo all'ambiente PassBuilder consiste nell'indicare il nome dell'App e dello Sprix da eseguire (e in essa contenuto) con prefisso "@" (ad esempio "-pApp@Sprix"); non specificando il nome dell'App viene eseguito lo Sprix con nome "Sprix" definito nel menu Sprix dello specifico gruppo oppure in quello generale (esempio "-p@Sprix"). <p>Esempio di sintassi di parametro -pCodiceApp@NomeFileSprix, per identificare Sprix/Collage contenuto in una App, ed in cui:</p> <ol style="list-style-type: none"> CodiceApp rappresenta il codice dell'App nella forma 999999APP (con 999999 = codice installazione, ad esempio 010999CUSCAREPRO nel caso dell'App "Customer Care Pro" rilasciata da Passepartout); NomeFileSprix rappresenta il nome del file sprix senza estensione (ad esempio sprix1 per lo sprix1.ss/sprix1.es) <p>Di conseguenza 010999CUSCAREPRO@sprix1 sarà la sintassi completa del parametro -p.</p> <ul style="list-style-type: none"> Un ulteriore metodo consiste nell'indicare il nome del solo programma "Sprix" senza il nome dell'App e senza il simbolo @ che si desidera eseguire (esempio "-pspx123"); in questo caso il lancio del programma è contestualizzato all'esecuzione, e pertanto lo Sprix all'interno dell'App stessa. Diversamente, invece, se SHELLSPRIX viene richiamata da uno Sprix/Collage (fuori da un'App, ovvero, dall'ambiente di sviluppo Sprix/Collage tradizionale) viene cercato lo Sprix definito nel menu Sprix (es. spx123). Dalla versione 64.0 la SHELLSPRIX in PassBuilder è in grado di eseguire uno Sprix non a menu (sprix000)
-v<ArgCmd>	Rappresenta la stringa da assegnare all'esecuzione dello Sprix e passata a questi prima del suo lancio tramite la variabile _ARGCMD\$. Attenzione: in presenza di spazi (blank) nella stringa del parametro occorre un trattamento per evitare il troncamento a destra dello spazio (es. CMD\$="BULL 080" → _ARGCMD\$="BULL") e consiste nel chiudere tra doppi apici ("), esempio: CMD\$=CHR\$(34)+"BULL 080"+CHR\$(34).

Istruzione che lancia l'esecuzione di un altro Sprix tramite il <comandospx\$> e con sintassi dei parametri da indicare tutta su una riga.

Per approfondimenti, consultare inoltre le configurazioni della Amministrazione Utenti contenute nel Manuale Generale e le Nozioni Generali (componenti client/server, configurazioni utenti/terminali), in particolare la Componente Client e l'esecuzione tramite il comando "-command=mxrs.exe seguito dai parametri Sprix". Es. CMD\$="mxrs.exe"+sxdft+" -t0 -o0 -a"+azsig+" -pspx302 -kADMIN -v"+ARGOM\$.

Notare: si ricorda che la sintassi per il parametro "-p" dei programmi "mxrs.exe" e "mxrsbatch.exe" necessaria per l'ambiente di sviluppo App in PassBuilder equivale a quella del parametro "-p" di questa istruzione SHELLSPRIX.

Gli eseguibili "mxrs.exe" e "mxrsbatch.exe" non possono eseguire uno Sprix che esegue una SHELLSPRIX, in tal caso è rilasciato l'errore SHELLSPRIX non disponibile su questo client - Elaborazione interrotta.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SLEEP

SLEEP(<n>)

Istruzione che sospende l'esecuzione del processo per <n> millisecondi.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SPRIGBL

SPRIGBL <NumBL>,<NumSottoBL>,<CodPF>,<NumRifPF>,<qta1>,...,<qta32>,[statidiriga\$]

Parametri	
NumBL	Numero Bolla di Lavorazione;
NumSottoBL	Numero Sotto Bolla di Lavorazione;

CodPf	Codice Prodotto Finito;
NumRifPF	Numero di riferimento al Prodotto Finito. Riguarda il dizionario olrii() delle righe di BL;
qtaN	Quantità/numero colli da spezzare; <qta1> e nel caso di taglie fino alla 32^ taglia <qta32>.
stadiriga\$	Questa variabile può contenere i campi STSRIGAORIG e STSRIGADEST che possono assumere i valori previsti dal campo "STATO DI RIGA DEL PRODOTTO FINITO", vedi help in linea della finestra "Dettaglio righe"-> colonna "S". Esempio utilizzo variabile: SPRIGBL 46,0, "SCARPAL",1,40,40,40,40,"STSRIGAORIG=I, STSRIGADEST=P".

Istruzione che spezza una riga di una BL in base a questi parametri obbligatori: il numero di Bolla di lavorazione <NumBL>, numero di Sottobolla di Lavorazione <NumSottoBL>, codice PF <CodPF>, numero di riferimento al PF <NumRifPF>, quantità/num. Colli da spezzare <qta>. L'istruzione valorizza ERRSPRIGBL e la ERRSPRIGBL\$.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SPXREMOTO

SPXREMOTO\$(<ind\$>,<par1>,<par2>,<filed\$>,<filee\$>,<sinc>)

Parametri	
<ind\$>	<p>Indirizzo e porta del Communicator Remoto (Client/Server Remoto), nei due formati alternativi:</p> <ul style="list-style-type: none"> formato socket manuale "nomemacchina:porta" oppure "IPadress:porta" formato "ID Installazione" del Server di destinazione (disponibile anche per metodo Shaker), anche se di una installazione inferiore alla versione V 62.4.
<par1>	<p>Stringa parametri di comando:</p> <p>"-aSiglaAzienda:SottoAzienda -tNumeroTerminale -xDataApertura -kUtente:Password -pNomeSprixRemoto"</p> <ul style="list-style-type: none"> se il Communicator Remoto gestisce Login di Mxserver aggiungere nella stringa le credenziali di accesso nella forma: "-login=nomeutente\$ -password=password\$" (non separate dai due punti ":"); se il Communicator Remoto è un'installazione Live Passepartout (Server Farm Passepartout) <par1\$> per forme di compatibilità deve assumere il valore "start.businesspass.passepartout.sm", mentre l'accesso da aggiungere è invece nella forma: "-login=nomeutente\$ -password=password\$" (credenziali di accesso al Predesk che sostituiscono il parametro "-k"). Prima della versione v62.4 occorre anticipare i parametri -login -password da ulteriori 3 parametri "-predesk=S -dominio=nomedominio\$ -idinst=nomeinstallazione\$". <p>NB: per approfondimenti consultare APRIAZIE e SHELLSPRIX, in particolare la sintassi del parametro "-p" per l'esecuzione di Sprix in ambiente PassBuilder; se questo parametro viene indicato senza specificare un contesto di appartenenza (ovvero "codiceApp@"), non essendo possibile la contestualizzazione dell'esecuzione (eseguita su altro server), lo Sprix con il nome "Sprix" viene cercato sia nei menu Sprix che nelle App che possono coesistere nel Server chiamato: nel caso in cui viene trovato un nome univoco viene eseguito lo Sprix desiderato, altrimenti, viene restituito un errore.</p> <p><u>Es Sprix nella stessa installazione non Live</u> X\$ = SPXREMOTO\$ ("localhost:9000", "-aSPX -t0 -x01012021 -kUtente:Password -pspx_rem2", ARG\$, "datispx.txt", "errspix.txt", 0)</p> <p><u>Es Sprix verso installazione non Live con Login=1</u> A\$ = SPXREMOTO\$ ("1234882000", "-kute:pwd123 -login=admin -password=adminpwd -aSTD -pspx123 -x01012022", "", "jfiled12", "jfilee12", 1)</p> <p><u>Es Sprix nella stessa installazione Live</u> X\$ = SPXREMOTO\$ ("1944687000", "-aSPX -t0 -x01012021 -login=ADMIN -password=123 -p907692CORSOSPXI0@spx_rem2", ARG\$, "j)datispx.txt", "j)errspix.txt", 0)</p> <p><u>Es Shaker verso installazione Live</u> SPXREM = MxSprix6_11.SPXREMOTO_S("1234882000", "-pspx123 -aSPX:G -kute:pwd123 -x31122021 -login=admin -password=adminpwd", "", "jfiled2", "jfilee2", 1)</p>
<par2>	Stringa facoltativa per l'esecuzione dello Sprix Remoto (con "" non si passa alcunché allo Sprix Remoto);
<filed\$>	<p>File per il deposito del risultato dell'elaborazione dello Sprix Remoto, esempio "datispx.txt" che riceve i dati; il file dati può ricevere eventuali errori contenuti in <filee\$>. La generazione del file sul Communicator Locale avviene tramite l'istruzione PUTREM_DATI.</p> <p>Nota bene: più chiamate in sequenza dell'istruzione SPXREMOTO\$() che inviano lunghezze superiori al primo invio, possono fare fallire la chiamata (warning: "impossibile allargare il file", "quantita' di record non ammessa"). Per ovviare, inviare una quantità costante di dati tramite la PUTREM_DATI.</p>
<filee\$>	File per l'errore generato dall'elaborazione dello Sprix Remoto, esempio "errspix.txt" riporta il messaggio "installazioni che non supportano questa esecuzione"; se la stringa = "" l'esecuzione è andata a buon fine. La generazione del file sul Communicator Locale avviene tramite l'istruzione PUTREM_ERR.

<code><sync></code>	Esecuzione sincrona o asincrona. Se il numero <code><sync>=0</code> il modo di esecuzione è sincrono (SPXREMOTO\$ Locale attende il trasferimento dei risultati provenienti dallo Sprix Remoto; terminata SPXREMOTO\$ sono disponibili il file dati/errori); se <code><sync>=1</code> è asincrono (SPXREMOTO\$() non attende risultati e termina, per cui occorre sapere quando trattarli).
---------------------------	--

<p>Funzione che, lanciata dal Communicator Locale (Client/Server Locale) uno Sprix Remoto cioè un servizio da eseguire lato Communicator Remoto (Client/Server Remoto o Control panel assistito) per ottenere sulla medesima installazione Locale i risultati provenienti dalle elaborazioni in Remoto (e le relative informazioni presenti lato Remoto). Richiede che sia abilitata l'assistenza remota da assistenti verso assistiti; dal 2005 si può eseguire anche fra versioni diverse.</p> <p>Lo Sprix Locale (relativo al Client/Server del Communicator Locale), lancia la SPXREMOTO\$() che contatta un predefinito Server Remoto (Client/Server del Communicator Remoto) attraverso la stringa <code><ind\$></code>, per eseguire lo Sprix Remoto sul Communicator Remoto tramite la stringa di parametri di comando <code><par1\$></code>.</p> <p>La funzione ritorna l'esito e l'eventuale errore dell'elaborazione dello Sprix Remoto in appositi file (dati/errore); nello Sprix (Communicator Remoto) associato alla SPXREMOTO\$() ed in grado di eseguire il servizio su richiesta del Communicator Locale, tramite delle istruzioni PUTREM si inseriscono i buffer <code><buff\$></code> contenenti il risultato/errore dell'elaborazione nei file dati/errori <code><filed\$>/<filee\$></code> nella directory dati (o dati utente) del Server Locale.</p> <p>Nota: SPXREMOTO\$() contenente l'istruzione PUTREM è un'esecuzione <u>batch</u>, pertanto <u>non</u> deve prevedere interazioni col terminale o il video dell'utente, pena l'interruzione dell'esecuzione con notifica dell'errore. Il valore della PUTREM non deve essere null ("") poiché causa un errore nell'esecutore (es. non inserire PUTREM_ERR "").</p> <p>In Sprix (oppure invio SMTP) gli accessi a file del Client con SPXREMOTO\$() sono preventivamente bloccati e con opportuna gestione di errore rilasciato nel log sottoforma di stream di errore, ma non per la _ERRFILE\$ o la _ERRDIR\$. Fa eccezione la ESISFILE che non torna errore, semplicemente non controlla se il file esiste e prosegue. Per approfondimenti consultare mxrsbatch nel manuale utente.</p> <p>Avvisi/Errori inviati con Email SMTP</p> <p>Non potendo visualizzare gli avvisi per assenza d'interfaccia utente, un invio mail SMTP generato da mxrs spxremoto/mxrsbatch registra solo sull'eventuale log tecnico attivo il risultato negativo di un controllo di sicurezza TLS. In tal modo, la stampa non si ferma e procede. Diversamente, in caso di esito negativo della transazione per un qualsiasi errore, il terminale chiude la propria esecuzione; allo stato dell'arte non è disponibile una variabile che intercetti l'eventuale errore ritornato. Analoga chiusura si verifica anche se un file allegato in configurazione risultasse inaccessibile durante la transazione. Per approfondimenti sulla gestione dell'EMAIL SMTP consultare il manuale utente (Servizi - Configurazioni - Stampanti).</p> <p style="text-align: center;">Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui</p>
--

SQL_ALLOCSTMT

SQL_ALLOCSTMT `<nc>`,`<ns>`

Parametri	
<code>nc</code>	Numero di connessione;
<code>ns</code>	Numero di statement;

Istruzione che nel numero connessione <code><nc></code> alloca la memoria per l'handle del <code><ns></code> numero di statement (allocazione memoria di riferimento, da 1 a 64) associato alla <code><nc></code> e che va liberato invocando SQL_FREESTMT. Esempio: SQL_ALLOCSTMT 1,1 : GOSUB GEST_ERR_SQL
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

SQL_AUTOCOMMIT

SQL_AUTOCOMMIT `<nc>`,`<tipo>`

Parametri	
<code>nc</code>	Numero di connessione;
<code>tipo</code>	Tipo di gestione delle transazioni;

Istruzione che nel numero connessione <code><nc></code> imposta l'eventuale <code><tipo></code> di gestione delle transazioni (>0 è SI ovvero il default, <=0 significa NO che richiede l'invocazione di SQL_TRANSACT per chiudere la transazione). Non è utilizzabile con pxjdbc.jar poiché agisce solo in modalità AUTOCOMMIT. La operazione inversa al termine della connessione evita gli errori. Esempio SQL_AUTOCOMMIT 1,1 (senza transazione) : SQL_AUTOCOMMIT 1,0 (con transazione)
Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui

SQL_BINDCOL

SQL_BINDCOL `<nc>`,`<ns>`,`<nbind>`,`<nomevar$>`[,`D`]

Parametri	
nc	Numero di connessione;
ns	Numero di statement;
nbind	Numero di colonna del result set (da 1 a n = ultima colonna del recordset; tanti per quanti sono i campi del result set, da 1 a 256)
nomevar\$	Variabile che riceve il contenuto di <nbind>
D	Flag da indicare in caso di colonne di tipo DATA per valorizzare la stringa in formato "AAAAMMGG".

Istruzione che nel numero connessione <nc> e di statement <ns>, assegna alla variabile <nomevar\$> il valore contenuto in <nbind> numero di colonna del result set (da 1 a n = ultima colonna del recordset; tanti per quanti sono i campi del result set, da 1 a 256) all'invocazione della SQL_FETCH e che in caso di colonne di tipo DATA richiede [,D] per valorizzare la stringa in formato "AAAAMMGG". Specifica dove devono essere ritornati i risultati della lettura SELECT) da effettuare per ogni colonna del result set specificando la variabile che verrà valorizzata da SQL_FETCH.

Esempio

```
SQL_BINDCOL 1,1,1,COD : GOSUB GEST_ERR_SQL
SQL_BINDCOL 1,1,2,DES$ : GOSUB GEST_ERR_SQL
SQL_BINDCOL 1,1,3,DAT$, D : GOSUB GEST_ERR_SQL
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_CONNECT

SQL_CONNECT <nc>,<DataSource\$>,<UserName\$>,<Password\$>

Parametri	
nc	Numero di connessione; numero da 1 a massimo 8 contemporanee, ed è il riferimento per le informazioni sulla connessione, lo stato della transazione e le condizioni d'errore;
DataSource\$	Profilo della connessione (nome o codice della sorgente dati per la connessione al driver ODBC per puntare al database di riferimento);
UserName\$	Utente;
Password\$	Chiave d'accesso;

Istruzione che realizza la connessione <nc> con il database tramite il profilo di connessione <DataSource\$>. Se non è specificato <UserName\$> e/o <Password\$> utilizzare "".

Se nel file "MxServer.ini" il parametro LOGIN = 1 occorre che gli utenti siano creati come utenti locali del Server, per l'autenticazione dell'utente all'apertura del Client, pena "Errore durante l'accesso: nome utente sconosciuto o password non valida". Se LOGIN = 0 non c'è autenticazione (se si utilizza un DSN utente, questo deve essere creato dall'utente proprietario dei processi in sede d'installazione del Server); la creazione di DSN Utente con un altro nome provoca errore "Nome origine dati non trovato"; l'utilizzo di un DSN di sistema non crea problemi. Il profilo ODBC va dichiarato sul Server con i flags di sola lettura ed accesso esclusivo impostati a "No" ed è consigliato che il profilo ODBC punti a un solo database e che sia dedicato per la singola azienda. Definire il DataSource\$ con il nome del DSN. Esempio, dichiarato il DSN\$="profdsn" (o "Access"): SQL_CONNECT 1,DSN\$,"ute","pwd" : GOSUB GEST_ERR_SQL.

Successivamente alla connessione, prima di eseguire ulteriori operazioni SQL occorre dichiarare la stringa di lettura/scrittura da eseguire. L'applicazione pone la stringa SQL in una variabile alfanumerica. Esempio:

```
STR_SQL1$ = "SELECT CODICE, DESCRIZIONE, DATA FROM Tabella"
STR_SQL2$ = "SELECT * FROM Tabella"
STR_SQL3$ = "SELECT NI_CONTO,CA_NOMERIC,DT_CREAZ FROM Rubrica"
STR_SQL4$ = "INSERT INTO Tabella(Codice,Descrizione) VALUES(1,"Pippo")"
STR_SQL5$ = "INSERT INTO "+TABELLA$+" (COLONNA) VALUES ("VAR$+"")"
```

L'interfaccia ODBC permette alle applicazioni di accedere ad un Database Relazionale utilizzando il linguaggio standard SQL. Con funzioni di ODBC l'applicazione può accedere indifferentemente a Database per cui è stato sviluppato un driver ODBC senza dover essere modificata. I comandi ODBC in Sprix/Collage adottano una sintassi simile a quella delle funzioni delle API ODBC di Microsoft, per usufruire della documentazione di ODBC disponibile. La chiave TipoODBC presente nel pxclient.ini può assumere i seguenti valori: 0 = DEFAULT (per Windows ODBC, per Linux JDBC); 1=ODBC (Windows), UnixODBC (in LINUX); 2=JDBC (per Windows e LINUX).

Si ricorda che nel sistema operativo Linux, il Gestionale non supporta collegamenti via JDBC (utilizzare collegamenti DSN via ODBC-MySQL) e che i comandi SQL in Server Farm è sottoposto a condizioni di Sistema: in caso di connessioni SQL via ODBC in ambiente Live (Server Farm) occorre considerare che in ambiente Live, lato Client sono consentiti consentiti reindirizzamenti "<c:\", per l'acquisizione di file personali (ascii, csv, jpg, ecc...), ma per il deposito lato Server sono consentiti i soli percorsi accessibili consentiti, ovvero:

```
PERCORSO <DATI GENERALI> "...dati\datigen\datiutente\" (abbreviabile con "!\"")
PERCORSO <DATI AZIENDA> "...dati\datiaz\azi\datiutente\" (abbreviabile con "!\")"
```

Al di fuori di queste modalità, in ambiente Live l'utilizzo di software diversi dal Gestionale che richiedono configurazioni particolari (es: operazioni SQL su file Xls/Mdb via ODBC, connessioni MySQL, ecc..) non sono disponibili. Nelle installazioni Live non è possibile accedere a database esterni tramite connessioni ODBC in quanto la parametrizzazione del DSN non è disponibile. Nello scambio dati da/a database esterni è però utile considerare che la componente Shaker ".Net" è disponibile in ambiente Live ed essendo questi un Client esterno al Server in Live, è in grado di eseguire in modo indipendente le sue connessioni (per scrivere o interrogare database esterni) per poi eseguire istruzioni Sprix di interesse per:

- la scrittura di file e loro copia su Server in Live nei suddetti percorsi accessibili per renderli disponibili alle procedure gestionali di servizio trasferimento archivi (ascii/csv)
- la scrittura diretta negli archivi del Gestionale tramite le istruzioni PUT?.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_DISCONNECT

SQL_DISCONNECT <nc>

Parametri	
nc	Numero di connessione,

Istruzione che chiude il numero connessione <nc> ed è obbligatoria; se è aperta una transazione associata questa viene chiusa con una ROLLBACK. Esempio SQL_DISCONNECT 1 : GOSUB GEST_ERR_SQL : END

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_DRIVERCONN

SQL_DRIVERCONN <modulo>,<ncon>,<tipo\$>,<driver\$>,<server\$>,<porta>,<nomedb\$>,<ute\$>,<pwd\$>

Parametri	
modulo	Numero del modulo. Valori accettati <ul style="list-style-type: none"> 0 (zero) modulo generico 3 connessione diretta al database di replica. Non richiede parametri, consente la connessione a qualunque tipo di database di replica installato che sia configurato dal menu [Servizi-Configurazione moduli-MexalDB]. Es. SQL_DRIVERCONN 3,1,"", "", "", "0", "", "" 4 connessione database Datatrek. Es. SQL_DRIVERCONN 4,1,"", "", "", "0", "", ""
ncon	Numero della connessione
tipo\$	Tipo RDBMS se modulo e' 0 allora "A" = ACCESS, "S" = SQL SERVER, "M" = MYSQL
driver\$	Nome del driver ODBC
server\$	Nome del server o IP/istanza
porta	Numero della porta 0=predefinita
nomedb\$	Nome del database (nel caso di uno sprix che utilizza Datatrek è obbligatorio il codice dell'elemento DataTrek (click sull'elemento DataTrek -> campo "Codice").
ute\$	Utente del database
pwd\$	Password del database

Istruzione che apre una connessione al database specificato. In caso di modulo impostato a 4 tutti i parametri possono essere impostati a "" o zero. Esempio SQL_DRIVERCONN 0,1,"S","SQL Server Native Client 11.0","srv-database","1443","miodb","sa","sa" : GOSUB GEST_ERR_SQL : END

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_EXECUTE

SQL_EXECUTE <nc>,<ns>

Parametri	
nc	Numero di connessione;
ns	Numero di statement;

Istruzione che nel numero connessione <nc> e di statement <ns> esegue uno statement "preparato". Esempio SQL_EXECUTE 1,1 : GOSUB GEST_ERR_SQL

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_EXEDIRECT

SQL_EXECDIRECT <nc>,<ns>,<str1\$>[,str2\$].....

Parametri	
nc	Numero di connessione;
ns	Numero di statement;
str1\$	Stringa SQL;

Istruzione che nel numero connessione <nc> e di statement <ns> esegue l'istruzione SQL di <str1\$>[,str2\$]...(stringa o stringhe SQL). Non sempre è consentita una isolata SQL_EXECDIRECT (ad esempio nell'accesso via pxjdbc.jar), perché viene richiesta SQL_PREPARE + SQL_EXECUTE (l'applicazione può scegliere di eseguire direttamente l'istruzione contenuta nella variabile oppure preparare la variabile per poi eseguirla). Esempio SQL_EXECDIRECT 1,1,STR_SQL\$

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_FREESTMT

SQL_FREESTMT <nc>,<ns>

Parametri	
nc	Numero di connessione;
ns	Numero di statement;

Istruzione che nel numero connessione <nc> chiude l'eventuale cursore aperto associato al numero di statement <ns>, scarta i risultati pendenti e libera tutte le risorse associate allo statement. Si consideri l'eventuale necessità di farla precedere da una allocazione (SQL_ALLOCSTMT) o come indicato dallo standard delle API ODBC. Esempio FINE_CONN: SQL_FREESTMT 1,1 : GOSUB GEST_ERR_SQL

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_FETCH

SQL_FETCH <nc>,<ns>

Parametri	
nc	Numero di connessione;
ns	Numero di statement;

Istruzione che nel numero connessione <nc> e di statement <ns> legge una riga dal result set; per leggerle tutte in sequenza dal primo record del recordset fino all'ultimo, occorre invocarla tante volte per il num. Di righe. Viene ritornato un valore per ciascuna colonna invocata da SQL_BINDCOL per leggere le informazioni in SELECT/INTERT. La SQL_FETCH imposta _ERRSQL_NDF che indica il fine dati nel cursore (termine del recordset). Ha valore 0 (false = righe non terminate) o 1 (true = lettura terminata). Esempio:

```
LEGGI_RECORDSET:
SQL_FETCH 1,1 : IF _ERRSQL_NDF=1 THEN GOTO FINE_CONN
PRINT STR$(COD,0,0)+" "+DESCR$
GOTO LEGGI_RECORDSET
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_PREPARE

SQL_PREPARE <nc>,<ns>,<str1\$>[,str2\$]...

Parametri	
nc	Numero di connessione;
ns	Numero di statement;
str1\$	Stringa SQL;

Istruzione che nel numero connessione <nc> e di statement <ns>, prepara <str1\$>[,str2\$]... la (o le) stringa SQL per eseguire la query. Esempio SQL_PREPARE 1,1,STR_SQL\$: GOSUB GEST_ERR_SQL

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SQL_TRANSACT

SQL_TRANSACT <nc>,<tipo>

Parametri	
nc	Numero di connessione;
tipo	Tipo di transazione;

Istruzione che apre automaticamente l'eventuale transazione associata al numero connessione <nc>; richiede un'operazione di accettazione modifiche di <tipo> COMMIT (accetta modifiche) se > 0 o ROLLBACK (perde modifiche e riporta il database allo stato precedente l'apertura transazione) se <= 0. Esempio: SQL_TRANSACT 1,1 (commit) : SQL_TRANSACT 1,0 (rollback)

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SRVCALL

SRVCALL <nome_coll_remoto\$>,<etichetta_server\$>,<etichetta_client\$>

Parametri	
nome_coll_remoto\$	Collage Server Remoto; elemento che è da creare lato Server Remoto in cui implementare i servizi utili alla App Mobile all'interno di diverse etichette (<etichetta_server\$>); tale elemento, quindi, implementa i servizi necessari in una o più etichette;
etichetta_server\$	Etichetta del servizio chiamato dalla SRVCALL;
etichetta_client\$	Etichetta di gestione del risultato del servizio; questa etichetta va implementata in Sprix Mobile;

Istruzione Sprix Mobile per eseguire elaborazioni bidirezionali fra Mobile e Server (andata e ritorno), sull'azienda ed alla data di apertura indicate al login sul dispositivo Mobile. La funzionalità è quasi simile ad una combinazione fra l'istruzione SPRIXREMOTO e l'istruzione CALL. L'istruzione SRVCALL valorizza le variabili _ERRSRV e _ERRSRV\$ per la gestione dell'errore di comunicazione con il Server. La gestione in lettura e scrittura fra le parti avviene tramite le istruzioni GETREM e PUTREM; l'esecuzione del servizio, infatti, prevede un passaggio di valori da Mobile a Server e viceversa: in particolare possono essere passati numeri, stringhe, array e file.

Nel dettaglio, per usufruire di un servizio on demand, sono necessari i seguenti passi:

- Creare un Collage Server Remoto che implementa i servizi necessari, in una o più etichette;
- Implementare, su Sprix Mobile, l'etichetta di gestione del risultato del servizio (<etichetta_client\$> nella chiamata SRVCALL);
- Usare le <PUTREM_...> per i valori che arrivano da Mobile a Server e/o viceversa;
- Usare le <GETREM_...> per leggere i valori che si vogliono inviare al Mobile o al Server;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SRVSYNCDOC

SRVSYNCDOC <doc\$>,<conto\$>,<etichetta\$>

Parametri	
doc\$	Documento di magazzino da leggere. Identifica il documento di cui ottenere la sincronizzazione/stampa. Parametro equivalente al parametro di GETMM_EXT a cui si rimanda per chiarimenti.
conto\$	Conto Cliente/Fornitore relativo al documento di magazzino. Identifica il documento di cui ottenere la sincronizzazione/stampa. Equivalente al parametro di GETMM_EXT a cui si rimanda per chiarimenti.
etichetta\$	Etichetta che verrà eseguita in Sprix Mobile quando è disponibile l'esito della risposta da parte del Server.

Istruzione Sprix Mobile per la sincronizzazione del documento di magazzino fra Mobile e Server. Nel parametro etichetta sono forniti gli eventuali errori contenuti nelle variabili _ERRSSD e _ERRSSD\$, il nuovo numero del documento nella variabile _SSDNUMDOC ed il percorso del file risultato della stampa nella variabile _SSDFILESTP\$ da poter visualizzare con l'istruzione VISUALIZZAFIILE.

Si gestiscono le casistiche di sincronizzazione nei seguenti modi:

- documento nuovo su mobile → scrittura su server – sincronizzazione – invio stampa a mobile
- documento esistente con ultima modifica su mobile → scrittura su server – sincronizzazione – invio stampa a mobile
- documento esistente ma modificato su server → errore (impossibile ottenere la stampa)
- documento esistente e sincronizzato → invio stampa a mobile

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

STARTCOLLAGE

STARTCOLLAGE <nomefile\$>

Parametri	
nomefile\$	Nome del file Collage senza estensione.
Notare bene: Per approfondimenti sulla sintassi necessaria per l'ambiente di sviluppo App in PassBuilder consultare il parametro “-p” dell’istruzione SHELLSPRIX.	

Istruzione che carica nella memoria il nome del file Collage <nomefile\$>, che non sia già caricato e che, se assente dall'elenco dei Collage, sia nella directory sprix corrente. L'evento “ON_START.” è il primo evento processato se avviene l'effettivo caricamento in memoria.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

STOPCOLLAGE

STOPCOLLAGE <nomefile\$>

Parametri	
nomefile\$	Nome del file Collage senza estensione.
Notare bene: Per approfondimenti sulla sintassi necessaria per l'ambiente di sviluppo App in PassBuilder consultare il parametro “-p” dell’istruzione SHELLSPRIX.	

Istruzione che scarica dalla memoria il nome del file Collage <nomefile\$>. Se <nomefile\$> = null (“”), scarica quello che ha eseguito STOPCOLLAGE e non ne fa eseguire più alcuna sua routine (effetto analogo a quello che si otteneva in passato con _ENDCOLLAGE=1).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

STORD

STORD <stampante>,<serie>,<numero>,<anno1>,<anno2>

Parametri	
stampante	Numero stampante;
serie	Serie Ordine;
numero	Numero Ordine;
anno1	Anno inizio ricerca;
anno2	Anno fine ricerca;

Istruzione per stampare sulla stampante numero <stampante>, la storia ordine con <serie> e <numero> relativo all'anno <anno1> e <anno2> d'inizio ricerca sui movimenti di magazzino (l'evasione). La stampa equivale a quella standard del gestionale. Se gli anni sono distanti da quello in uso, si allungano i tempi di elaborazione. L'istruzione valorizza le variabili `_ERRSTORD` e `_ERRSTORD$`.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

STORELOC

STORELOC <tipo>

Parametri	
tipo	Tipo di memorizzazione: "T" = temporaneo, "P"=persistente;

Istruzione SprixMobile relativa al salvataggio dati in aree di memoria. I salvataggi automatici store persistenti sono:

1. Uscita dall'App
2. Cambio Sprix

Il salvataggio non viene effettuato a fronte di un crash del sistema. Lo store temporaneo viene cancellato in uscita dall'App. La variabile di riferimento relativa a messaggi di errore è `_ERRLOC$`. Esempio:

```

PUTLOC_STR "T", "Cliente", CLI$      (Temporaneo)
GETLOC_STR "T", "Cliente", OLDCLI$
PUTLOC_NUM "P", "Operazioni", NUMOP (Persistente)
STORELOC "P", salva in modo persistente i dati in memoria.
  
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

STR

STR\$(<num>,<numdec>,<punti>)

Parametri	
num	Numero da convertire;
numdec	Decimali; (da -1 a 6, se -1 è assunto automaticamente il numero di decimali di <num>)
punti	Punti separatori di migliaia (1=si, 0=no).

Funzione che converte il numero <num> in stringa contenente il <num> con <numdec> decimali e punti separatori di migliaia.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

STRCMP_CI

STRCMP_CI(<str\$1>,<str\$2>)

Parametri	
str1\$	Prima stringa del confronto dimensionale;
str2\$	Seconda stringa del confronto dimensionale;

Funzione che confronta in modalità case insensitive la stringa <str\$1> con la stringa <str\$2>. Ritorna: -1 se str\$1 < str\$2; 0 se str\$1 = str\$2; 1 se str\$1 > str\$2. Non distingue fra caratteri minuscoli e maiuscoli.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

STRDATA

STRDATA\$(<data\$>,<forma>,<separatore\$>)

Parametri	
data\$	Data da trasformare dal formato "AAAAMMGG" (forma tipica di dato parametrico in Sprix/Collage);
forma	Formato di trasformazione (1 = GGMM, 2 = GGMAA, 3 = GGMAAAA);
separatore\$	Carattere di separazione (" ", ".", "" ecc);

Funzione che trasforma la stringa <data\$> dal formato "AAAAMMGG" (forma tipica di dato parametrico in Sprix/Collage) nel formato <forma> e <separatore\$>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

STRING

STRING\$(<numrip>,<str\$>)

Parametri	
numrip	Numero di ripetizioni;
str\$	Stringa da ripetere;

Funzione che ripete una stringa <str\$> per <numrip> volte.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

SVILDB

SVILDB <codart\$>,[codpad\$],[codcli\$],[nrord],[qta1],..., [qta32],[datibi\$]

codart\$	Codice articolo con Distinta Base Primaria (obbligatorio);
codpad\$	Codice articolo padre Prodotto Finito (non obbligatorio); se non specificato (= "") è assunto = <codart\$>. Tale codice serve per le condizioni di tipo "A" / "S". Rappresenta il codice del prodotto finito a cui fare riferimento per le condizioni articolo / semilavorato. Tale codice non va confuso con il padre di articolo strutturato.
Codcli\$	Codice cliente per le eventuali condizioni e/o DBV; se non specificato (= "") è assunto quello indicato in DBP;
nrord	Numero di record di riga OC (ocnri()) di cui si vuole lo sviluppo di DBVO;
qta1,qta2...,qta32	Quantità numeriche di suddivisione per taglia e per articoli non a taglie, queste vanno sostituite dal parametro [V,F] (numero di pezzi V da produrre – max 9999 – con stessa dimensione/caratteristica F, es. 10 porte da 3,5 metri: V*F → 10,3.5) solo due valori, altri aggiunti verrebbero ignorati;
datibi\$	Parametro che consente lo sviluppo della DBV per bolla di lavoro. Si tratta di un unico parametro che contiene al suo interno i campi DTDOC, BOLLA, SOTTOBOLLA e RIFPF. Nel caso in cui parametri non coincidano con una riga in BL associata a codart\$ viene ritornato lo sviluppo della DBP. Esempio di utilizzo: SVILDB "ARTDB01", "", "501.3", 0, 100,"DTDOC=20220920, BOLLA=41, SOTTOBOLLA=0, RIFPF=1".

Istruzione che sviluppa la Distinta Base Primaria dell'articolo <codart\$>. Valorizza le variabili dizionario della tabella "os" (variabili array ad accesso diretto da leggere con ciclo FOR; tabella non utilizzabile per ciclotabellare, poichè è una tabella che riceve i risultati della SVILDB e non per esplodere i componenti dei prodotti finiti); Sprix non legge il Codice PF padre default (in ulteriori dati della DBP) poichè viene passato con il parametro [codpad\$]. Valorizza le variabili _NSVILDB, _ERRSVILDB\$, dalla versione v630 anche _MSGSVILDB e _MSGSVILDB\$(). Dalla versione 2023A2 se viene specificato numero bolla, sottobolla, riferimento prodotto finito, la distinta contiene dei semilavorati e in testata della BL, dentro il campo ulteriori dati, è stato specificato la categoria semilavorato, la funzione esplosa il dettaglio degli impegni anche per il semilavorato.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

TOTMM

Istruzione che calcola i totali del documento nelle variabili _MMT se richiamata dopo la valorizzazione delle variabili _MM relative ad un documento valido. L'istruzione ha gli stessi vincoli funzionali dell'istruzione PUTMM.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

UNISCIPDF

UNISCIPDF <elePDF\$>,<PDFgen\$>,[tit\$],[aut\$],[obj\$],[app],[DTcrea]

Parametri	
elePDF\$	Elenco dei file PDF di origine: File_PDF_Origine\$() Array monodimensionale di stringhe contenente i nomi dei file PDF (completi di path) da unire
PDFgen\$	File PDF generato: nome del file PDF (completo di path) da generare
tit\$	Stringa opzionale: Titolo da inserire nelle proprietà del file PDF generato
aut\$	Stringa opzionale: Autore da inserire nelle proprietà del file PDF generato
obj\$	Stringa opzionale: Oggetto da inserire nelle proprietà del file PDF generato
app	Numerico opzionale: Applicativo; se diverso da 0 nella proprietà "Applicazione" del file PDF generato viene inserito il nome del prodotto Passepartout
DTcrea	Numerico opzionale: Data e Tempo (ora) di creazione; se diverso da 0 nella proprietà "Creato il" del file PDF generato viene inserita la data e ora di creazione.

Istruzione per unire più file PDF. Come per tutte le istruzioni che accedono ai file è possibile attivare la gestione degli errori e il controllo accesso utente al file tramite la variabile _ACCFILEUTE; in tal caso gli eventuali errori sono ritornati nella variabile _ERRFILE\$
Esempio: UNISCIPDF File_PDF_Origine\$(), Nome_File_Risultante,"", "", "", 0,0

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

VAL

VAL(<str\$>)

Parametri	
str\$	Stringa da convertire;

Funzione che converte in numero la stringa <str\$> a partire da sinistra terminando al primo carattere non numerico incontrato; se il primo carattere di <str\$> è "-" o "+" lo considera segno; se fra i caratteri numerici c'è un "." o "," è assunto come punto decimale (massimo 6 decimali).

Attenzione: se nel codice alternativo dell'articolo è presente un codice alfanumerico con un carattere "E" oppure "D" viene considerato come esponente 10[^], con generazione di numero massimo pari a 18.446.774.073.709.552.000. Per esempio: se il codice alternativo di un articolo è "9E2" oppure "9D2", il VALORE=VAL(aralt("ART")) equivale a 900, ovvero 9*10[^]2.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

VARKEY

VARKEY <numfile>,<key\$>

Parametri	
numfile	Numero File Indici
key\$	Chiave principale

Istruzione che nel file a indici aperto con numero <numfile> varia il record in base alla chiave stringa principale <key\$> con i valori contenuti nelle variabili del buffer associato. In variazione se i valori delle variabili sono più lunghi dei buffer è scritta solo la parte non eccedente, diversamente (se inferiori) sono utilizzati ASCII 0 per l'eccedente; la variazione modifica anche la chiave principale se la nuova non esiste; se la principale esiste già la variazione non è eseguita; tutte le chiavi sono obbligatorie e i loro valori non possono essere tutti ASCII 0 o ASCII 255. Se i campi contenenti chiavi sono variati i relativi file chiavi vengono riordinati. Più chiavi secondarie uguali del nuovo record vengono inserite in ordine di chiave principale. Fare attenzione a tutto il buffer (la dimensione della chiave deve mantenere la lunghezza). Valorizza _NRDATI e _ERRIND.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

VIMSG

VIMSG <tipo>,<msg\$>

Parametri	
tipo	Numero di tipo di messaggio da visualizzare: 1 (normale), 2 (normale), 3 (aiuto), 4 (messaggio), 5 (errore);
msg\$	Stringa del messaggio da visualizzare;

Istruzione che visualizza il messaggio <msg\$> in finestra centrale auto dimensionante (sulla lunghezza di <msg\$>) di <tipo>. Sospende il programma fino alla pressione di Invio. Caratteri speciali:

- il carattere "l" (ASCII 124) effettua il salto riga;
- il prefisso (primo carattere di <msg\$>) "<" o ">" allinea il testo a sinistra oppure a destra;
- il prefisso (primo carattere di <msg\$>) "\$" visualizza dei caratteri non proporzionali allineati a sinistra;
- il prefisso (primo carattere di <msg\$>) "%" visualizza dei caratteri non proporzionali allineati a destra;
- il prefisso (primo carattere di <msg\$>) "!" visualizza dei caratteri non proporzionali allineati al centro;
- il prefisso (primo carattere di <msg\$>) "*" non visualizza il messaggio e la riga viene considerata come titolo della finestra;

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

VIMSG_EXT

VIMSG_EXT <m\$>,<t\$>,[s\$]

Parametri	
m\$	Stringa di messaggio
t\$	Tasti funzione (massimo 4); se non impostati il default è "501". La sintassi equivalente a "CodTasto+Descrizione,CodTasto+Descrizione,..."; (vedere SETTF e PUTTAST)
s\$	Stringa non attiva (parametro predisposto);

Istruzione che visualizza il messaggio <m\$>; imposta la variabile _TF.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

VISUALIZZAFI

VISUALIZZAFI <tipoeditor>,<file\$>

Parametri

tipoeditor	0 – Utilizza editor del sistema operativo 1 – Utilizza editor interno (attivo solo per PDF e immagini BMP, JPG, PNG, GIF, WMF, EMF);
file\$	Nome del file da aprire comprensivo del percorso Nomenclatura File

Istruzione che visualizza e/o apre un file specificato nel parametro, utilizzando l'editor interno di Passepartout, oppure un editor del sistema operativo. Per ulteriori gestioni di file consultare INPUT_EXT (tipo F/P) e altri comandi raggruppati per argomento "DATABASE – FILE – COM – SHELL". Se si utilizza l'editor esterno valorizza la variabile _ERRFILE\$ in caso di errore. Esempi: 'Visualizza il file presente sul filesystem con l'editor esterno VISUALIZZAFILE 0,"c:\tmp\immagine.jpg"; 'Visualizza il file presente nella cartella dati di PassBuilder con l'editor interno VISUALIZZAFILE 1,"cartella\$file.pdf".

Per accedere alle cartelle dati, utilizzare la sintassi NomeCartella\$NomeFile.est (Il percorso viene risolto con "\ " finale dopo il nome cartella) esempio fileapp\$immagine.jpg. Per ulteriori approfondimenti si rimanda al manuale PassBuilder.

In ambiente **Sprix Mobile**, il trasferimento delle immagini dal Server al dispositivo Mobile avviene in fase di sincronizzazione (opzionale), ma è possibile visualizzare le immagini nelle liste archivio anche tramite la visualizzazione di un file mediante il visualizzatore abbinato al tipo di file e presente sul dispositivo mobile tramite l'istruzione VISUALIZZAFILE a cui passare il percorso dell'immagine collegata all'articolo specificata nella variabile di struttura _ARIMM\$ con parametro 0 (visualizzatore esterno). Sprix Mobile può avvalersi anche della variabile _WLROWICO\$ per la gestione dell'icona dell'articolo nel catalogo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

WCALL

WCALL <azione\$>,<id>,<str\$>

Parametri	
azione\$	<p>Azione da eseguire:</p> <ul style="list-style-type: none"> - "REFRESHLIST",id - Esegue il refresh di una lista; - "PRINTFORM" (<i>obsoleta</i>) - Per eseguire la stampa offline di un form, non richiede l'esecuzione di WSHOWFORM; il risultato è una stampa del form in formato pdf, il form viene stampato esattamente come viene visualizzato a video. Nota: le funzioni INLINE insieme a WILENCAR permettono la stampa affiancata. <u>Sostituire</u> dall'istruzione che consente di passare più parametri WPRTFORMOPT <id_form> e relative variabili _ERRWP e _ERRWP\$ di errore _WPHEIGHT\$ = altezza (mm , senza virgole) es: _WPHEIGHT\$ = "200" _WPWIDTH\$ = larghezza (mm , senza virgole) es: _WPWIDTH\$ = "100" _WPDMFONT\$ = dimensione font (mm, senza virgole) "{titolo}:{header}:{testo}:{footer}" esempio: _WPDMFONT\$ = "10:10:8:10" - "CLOSEFORM",id - Forza la chiusura del form creato. Non ha effetto sui form creati a partire dal primo. - "FOCUSINPUT",id - Forza il focus su una particolare casella di input; l'utilizzo è consentito solo: <ul style="list-style-type: none"> o quando scatta un evento di ON_OUT di una casella di testo; o quando scatta un evento di ON_PRESS di un pulsante, a patto che contestualmente non si crei o visualizzi un altro form <p>Nota: se viene utilizzata una VIMSG il fuoco viene perso. Il modo corretto per gestire il focus in questo caso è quello di utilizzare una DIALOG e gestire il focus al ON_PRESS di uno dei pulsanti della dialog stessa.</p> <p>Notare: è possibile forzare il focus alla partenza di un form se l'istruzione viene invocata durante la creazione del form stesso, subito prima della WSHOWFORM. Per approfondimenti si rimanda al manuale Sprix Mobile.(v 3.5).</p> - "OPENMENU",0 - Forza l'apertura del menu laterale. È possibile forzare da codice l'apertura del menu laterale in cui sono presenti tutte le voci di menu della PassApp - "OPENMENU" ",0,"nome_sprix" - Forza l'avvio di un'altra voce di menu (il nome di un altro sprix-mobile e senza estensione .es); il form corrente verrà chiuso in favore del nuovo e non sarà possibile ripristinarlo tornando indietro con back. - "SETERROR",id,"stringa_errore" - Mostra un messaggio di errore sul campo di testo indicato da ID a seguito di un controllo. L'istruzione fa comparire un'icona di errore (icona rossa con punto esclamativo). Con un tap sull'icona verrà mostrato il messaggio di errore impostato in "Stringa errore". Appena si comincia a scrivere nel campo di testo il messaggio e l'icona di errore spariscono. - "SENDGPSDATA",min - Abilita l'invio temporizzato (o meno) del dato di localizzazione Passmobile (anche in background; comando da utilizzare lato_client (mobile); quando l'app è in background IOS può inviare aggiornamenti solo se cambiano le coordinate del dispositivo, non in modo temporizzato); per interrompere eseguire min = -1 (es WCALL "SENDGPSDATA",-1); l'eventuale errore di invio dati è riportato dalla variabile _ERRGPS\$ e le variabili restituite sono _GPS? (ovvero quelle da leggere lato_server) in cui il "numero di dati nell'array" è _GPSNUM (Il contatore _GPSNUM viene resettato ad ogni chiamata. - "GETGPSDATA",0 - Recupera i dati GPS (eseguibile lato_server); le variabili _GPS? sono restituite all'evento ON_GETGPSDATA: La variabile con l'errore è _ERRGPS\$. Nota: La richiesta dei dati GPS termina eventuali invii temporizzati di dati GPS al server tramite WCALL "SENDGPSDATA". Nell'evento di callback ON_GETGPSDATA sarà quindi necessario riavviare l'eventuale invio temporizzato. In Collage Server Remoto avvalersi dell'omonima istruzione GETGPSDATA (a cui si rimanda).
id	Numero della lista che si vuole aggiornare;
str\$	Stringa relativa al "nomesprix" oppure all'errore "stringa_errore"

Istruzione di servizio Sprix Mobile che esegue una funzione specifica per un oggetto grafico. Permette di interagire con gli elementi di

Passmobile in modo trasversale, nel senso che non è legata ad un ben preciso oggetto presente nel form (v. Manuale Sprix Mobile per il refresh delle liste dinamiche o per la chiusura forzata di un form), ma può essere utilizzata anche per altre funzioni di interazione ("REFRESHLIST", "CLOSEFORM" "OPENMENU",

Esempio: WCALL "REFRESHLIST", ID_LISTCF (ID_LISTCF identificativo della lista).
 Esempio: WCALL "PRINTFORM", ID_FORM

In ogni caso, per approfondimenti consultare anche il manuale Sprix Mobile.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

WCREATEDIALOG

WCREATEDIALOG <idnomedialog\$>

Parametri

idnomedialog\$	Nome di riferimento della finestra di dialogo
----------------	---

Istruzione che crea una finestra di dialogo modale. Il default è simile a VISMG (con un solo tasto "Ok"), altrimenti può prevedere fino a 3 tasti intercettabili con l'evento ON_PRESS_nome tasto. Parametri WCREATEDIALOG:

- Ingresso: _WDTITLE\$, _WDMESSAGE\$
- Uscita: _WDOID, _ERRWD, _ERRWD\$

Le finestre di dialogo modali sono per:

1. Visualizzare un messaggio da confermare
 2. Visualizzare una domanda con due o più risposte selezionabili mediante pulsanti (al max 3)
 3. Visualizzare una lista di due o più opzioni tra cui selezionarne una
- La 1 estende l'istruzione VISMG avendo a disposizione un evento di pressione pulsante per l'esecuzione di codice.
 La 2 e la 3 permettono di avere delle finestre più immediate dal punto di vista utente di un form.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

WCREATEFORM

WCREATEFORM <form\$>

Parametri

form\$	Identificativo univoco della finestra che si vuole aprire
--------	---

Istruzione che imposta la form di preparazione della finestra. Il disegno della finestra deve essere fatto successivamente con l'istruzione WSHOWFORM, dopo aver preparato i campi da inserire nella finestra. Vengono valorizzate le variabili _ERRWF, _ERRWF\$ che contengono il numero e la descrizione di un eventuale errore. Viene anche valorizzata la variabile _WFOID, numero univoco della finestra.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

WCREATELIST

WCREATELIST <lista\$>

Parametri

lista\$	Identificativo univoco della lista che si vuole aprire
---------	--

Istruzione che esegue il disegno della lista eseguita successivamente con l'istruzione WSHOWFORM, dopo aver preparato i campi da inserire nella lista. Vengono valorizzate le variabili _ERRWL, _ERRWL\$ che contengono il numero e la descrizione di un eventuale errore. Viene anche valorizzata la variabile _WLOID, numero univoco della lista. Tramite il comando si possono definire quattro tipi di lista

- Lista Dinamica dove il contenuto della lista deriva da campi elaborati dall'utente
- Lista Estesa dove il contenuto della lista deriva da campi elaborati dall'utente concatenabili tra loro
- Lista da Archivio
- Lista su Archivio collegato ad un campo d'input

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

WCREATEINPUT

WCREATEINPUT <input\$>

Parametri

input\$	Identificativo univoco dell'input
---------	-----------------------------------

Istruzione che definisce un input da inserire nel form che sarà visualizzato dall'istruzione WSHOWFORM. Vengono valorizzate le variabili _ERRWI, _ERRWI\$ che contengono il numero e la descrizione di un eventuale errore. Viene anche valorizzata la variabile _WIOID, numero univoco del pulsante. Tramite il comando si possono definire i tipi di input:

- Input di tipo pulsante, tramite l'assegnazione della variabile _WITYPE\$ con il valore "BUTTON". L'input di tipo BUTTON può

essere di due tipologie in base alla variabile `_WIPARENTZN$` che imposta il pulsante nella toolbar o nella form. I pulsanti definiti nella form sono posti uno accanto all'altro nella stessa riga. Nel caso tra i pulsanti ci sia un input allora i pulsanti definiti prima dell'input vengono posti nella riga precedente all'input, quelli definiti dopo vengono posti sulla riga successiva.

- Input di tipo `testo`, tramite l'assegnazione della variabile `_WITYPE$` con il valore "TEXT". L'input di tipo TEXT può essere anche una lista se impostate correttamente le variabili `_WILISTID` e `_WILISTNFLD`
- Input di tipo `data`, tramite l'assegnazione della variabile `_WITYPE$` con il valore "DATE". In questo caso l'input viene gestito con l'aiuto del calendario per impostare la data. L'input non fa, al momento, controlli sulla data valida del valore inputato, ma legge e riporta i primi 10 caratteri che trova. Il formato è GG/MM/AAAA.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

WGET

WGET <idobj>

Parametri	
idobj	Identificativo univoco dell'oggetto;

Istruzione che legge il valore della proprietà dichiarata. Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

WGETOID

WGETOID <obj\$>

Parametri	
obj	<p>Oggetto di cui si vuole l'ID.</p> <p>Per aggiungere un'etichetta sui menu di PassMobile il parametro <obj> può assumere il valore "PMMENU" in modo da ottenere l'ID del menu di PassMobile da gestire poi WSET. Esempio:</p> <p>WGETOID "PMMENU" → ottiene l'id del menu di PassMobile IDMENU = <code>_WOPVAL</code> <code>_WOPNAME\$</code> = "sprxm33.subtitle" → indica su quale sprxm mobile si vuole aggiungere il sottotitolo <code>_WOPVAL\$</code> = "stringa di dettaglio" WSET IDMENU</p>

Istruzione analoga a WGET, restituisce l'id dell'oggetto di cui si vuole ottenere l'ID. Dopo l'esecuzione della WGETOID viene valorizzata la variabile di struttura <code>_WOPVAL</code> con l'ID dell'oggetto. L'istruzione WGETOID è utile per interagire con un controllo di cui non si conosce l'ID. Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

WPRTFORMOPT

WPRTFORMOPT <id_form>

Parametri	
id_form	Identificativo univoco della form;

Istruzione che sostituisce la WCALL "PRINTFORM" per eseguire la stampa offline di un form, in base alla struttura <code>_WP?</code> Ritorna errore nelle variabili <code>_ERRWP</code> e <code>_ERRWP\$</code> Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

WSET

WSET <idobj>

Parametri	
idobj	Identificativo univoco dell'oggetto;

Istruzione che imposta il valore della proprietà dichiarata. Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. Clicca qui
--

WSHOWDIALOG

WSHOWDIALOG <iddialog>

Parametri	
iddialog	Id finestra dialogo;

Istruzione che mostra la finestra preparato con l'istruzione WCREATEDIALOG. Prima di disegnare la finestra occorre averla preparata con gli eventuali bottoni di comando (da 0 a 3). Vengono valorizzate le variabili _ERRWD, _ERRWD\$ che contengono il numero e la descrizione di un eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

WSHOWFORM

WSHOWFORM <idform>

Parametri	
idform	Numero univoco della finestra che si vuole aprire; viene ritornato dalla variabile _WFOID valorizzata a seguito della WCREATEFORM.

Istruzione che mostra il form preparato con l'istruzione WCREATEFORM. Prima di disegnare la finestra occorre aver preparato la finestra e gli eventuali campi, liste, ecc. che devono essere contenuti. Vengono valorizzate le variabili _ERRWF, _ERRWF\$ che contengono il numero e la descrizione di un eventuale errore. Per eseguire la stampa offline di un form consultare la WCALL.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLADDCDATA

XMLADDCDATA(<RifDoc>,<RifNodo>,<Nome\$>,<Valore\$>)

Parametri	
RifDoc	Riferimento Documento
RifNodo	Riferimento del Nodo
Nome\$	Nome del CDATA
Valore\$	Valore del CDATA

Funzione per la fase di **creazione** XML. Aggiunge nel documento <RifDoc> al nodo <RifNodo> il valore CDATA (testo semplice) con nome <Nome\$> e valore <Valore\$>. Ritorna il riferimento al nodo con il valore CDATA o zero se fallisce. La variabile _ERRXML\$ viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLADDATTR

XMLADDATTR(<RifNodo>,<Nome\$>,<Valore\$>)

Parametri	
RifNodo	Riferimento del Nodo
Nome\$	Nome dell'attributo
Valore\$	Valore dell'attributo

Funzione per la fase di **creazione** XML. Aggiunge l'attributo con nome <Nome\$> e valore <Valore\$> al nodo <RifNodo>. Ritorna il riferimento al nuovo attributo o zero se la creazione del nuovo attributo fallisce. La variabile _ERRXML\$ viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLADDELE

XMLADDELE(<RifNodo>,<NomeElemento\$>,<ValoreElemento\$>)

Parametri	
RifNodo	Riferimento del Nodo
NomeElemento\$	Nome dell'elemento
ValoreElemento\$	Valore dell'elemento

Funzione per la fase di **creazione** XML. Aggiunge un elemento con nome <NomeElemento\$> e valore <ValoreElemento\$> al nodo <RifNodo>. Ritorna il riferimento al nodo creato o zero se la creazione del nuovo elemento fallisce. La variabile _ERRXML\$ viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLADDNODVAL

XMLADDNODVAL <RifNodo>,<Valore\$>

Parametri	
RifNodo	Riferimento del Nodo
Valore\$	Stringa da aggiungere

Istruzione per la fase **creazione** XML. Aggiunge al valore del nodo <RifNodo> il testo <Valore\$>. La variabile `_ERRXML$` viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLADDNS

XMLADDNS(<RifNodo>,<NSPrefix\$>,<NSHref\$>)

Parametri	
RifNodo	Riferimento del Nodo
NSPrefix\$	Prefisso namespace; nel file XML è consentito inserire un namespace senza identificatore (cioè indicare un valore nullo <NSPrefix\$>= "") per ottenere attributi xmlns senza il carattere separatore ":". La specifica del parametro 'nullo' si ottiene tramite la sua valorizzazione con lo speciale carattere ASCII 255.
NSHref\$	HREF

Funzione per la fase **creazione** XML. Aggiunge il namespace con prefisso <NSPrefix\$> e HREF <NSHref\$> al nodo <RifNodo>. Ritorna il riferimento al namespace o zero se la creazione fallisce. La variabile `_ERRXML$` viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLCLEANUP

Istruzione per la fase di **creazione** XML. Rilascia l'ambiente per la gestione XML, può essere eseguita solo se precedentemente viene eseguita XMLINIT.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLDOCADDCOM

XMLDOCADDCOM <RifDocumento>,<TestoCommento\$>

Parametri	
RifDocumento	Riferimento del documento
TestoCommento\$	Stringa di commento

Istruzione per la fase di **creazione** XML. Aggiunge il commento <TestoCommento\$> al documento <RifDocumento>. La variabile `_ERRXML$` viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLDUMPFIL

XMLDUMPFIL(<RifDoc>,<NomeFile\$>,<Encoding\$>)

Parametri	
RifDoc	Riferimento del documento
NomeFile\$\$	File di scrittura
Encoding\$	Encoding

Funzione per la fase di **creazione** XML. Scrive sul file <NomeFile\$>, il documento <RifDoc> con l'encoding <Encoding\$>. Ritorna un valore diverso da zero se l'operazione è riuscita o zero se fallisce. La variabile `_ERRXML$` viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLELADDCOMM

XMLELADDCOMM(<RifNodo>,<TestoCommento\$>)

Parametri	
RifNodo	Riferimento del Nodo
TestoCommento\$	Stringa di commento

Funzione per la fase di **creazione** XML. Aggiunge il commento <TestoCommento\$> al nodo <RifNodo>. Ritorna il riferimento al commento creato o zero se la creazione del commento fallisce. La variabile `_ERRXML$` viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLFINDATTR

XMLFINDATTR(<RifNodo>,<Nome\$>)

Parametri	
RifNodo	Riferimento del nodo
Nome\$	Nome nodo

Funzione per la fase di **lettura/navigazione** XML. Cerca a partire nel nodo con riferimento <RifNodo>, l'attributo con nome <Nome\$> e ne ritorna il riferimento oppure zero se fallisce.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLFINDELEM

XMLFINDELEM(<RifNodo>,<Nome\$>)

Parametri	
RifNodo	Riferimento del nodo
Nome\$	Nome nodo

Funzione per la fase di **lettura/navigazione** XML. Cerca a partire dal nodo con riferimento <RifNodo>, il nodo con nome <Nome\$> e ne ritorna il riferimento oppure zero se fallisce.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLFINDELNS

XMLFINDELNS(<RifNodo>,<NomeElemento\$>,<NSPrefix\$>,<NSHref\$>,<Depth>)

Parametri	
RifNodo	Riferimento del nodo
NomeElemento\$	Nome nodo
NSPrefix\$	Prefisso namespace
NSHref\$	HREF
Depth	profondità

Funzione per la fase di **lettura/navigazione** XML. Ricerca di un nodo a partire dal nodo <RifNodo>, con nome <NomeElemento\$> namespace prefix <NSPrefix\$> e nameSpece HREF <NSHref\$>, per una profondità di iDepth livelli. Ritorna il riferimento al nodo trovato oppure zero se fallisce.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLFREEDOC

XMLFREEDOC(<RifDocumento>)

Parametri	
RifDocumento	Riferimento del documento

Istruzione per la fase di **creazione** XML. Viene rilasciato tutto il documento XML <RifDocumento>.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETATNAM

XMLGETATNAM\$(<RifAttr>)

Parametri	
RifAttr	Riferimento attributo

Funzione per la fase di **lettura/navigazione** XML. Acquisisce il nome dell'attributo con riferimento <RifAttr>, ne restituisce il valore o una stringa vuota se fallisce. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array _XMLSTRVAL\$() ed il numero degli elementi viene riportato in _XMLVALNR.

Se la funzione viene utilizzata per leggere il nome ed il contenuto dei nodi, piuttosto che per leggere nomi e valori degli attributi, può essere rilasciato errore (dalla libreria libxml2); occorre quindi l'utilizzo di XMLGETELNAM\$ anzichè XMLGETATNAM\$ (utilizzabile esclusivamente per acquisire il nome di un attributo).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETATTNS

XMLGETATTNS(<RifAttr>)

Parametri	
RifAttr	Riferimento attributo

Funzione per la fase di **lettura/navigazione** XML. Acquisisce il namespace dell'attributo con riferimento <RifAttr> e ne ritorna il riferimento oppure zero se fallisce.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETATVAL

XMLGETATVAL\$(<RifAttr>)

Parametri	
RifAttr	Riferimento attributo

Funzione per la fase di **lettura/navigazione** XML. Acquisisce il valore dell'attributo con riferimento <RifAttr> e ne ritorna il valore o una stringa vuota se fallisce. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array `_XMLSTRVAL$()` ed il numero degli elementi viene riportato in `_XMLVALNR`. Esempio:

`VALATTR$ = XMLGETATVAL$(NODO): IF VALATTR$="" OR _ERRXML$<>"" THEN VIMSG 5, _ERRXML$:`

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETCHILD

XMLGETCHILD(<RifNodo>)

Parametri	
RifNodo	Riferimento del nodo

Funzione per la fase di **lettura/navigazione** XML. Restituisce il riferimento al nodo figlio del nodo <RifNodo> oppure 0 se nessun nodo figlio esiste. La variabile `_ERRXML$` conterrà in questo caso una descrizione dell'errore. Esempio:

`NODO2 = XMLGETCHILD (NODO1)`

`IF NODO2=0 OR _ERRXML$<>"" THEN VIMSG 5," Impossibile trovare il nodo figlio: " + _ERRXML$:`

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETELNAM

XMLGETELNAM\$(<RifNodo>)

Parametri	
RifNodo	Riferimento del nodo

Funzione per la fase di **lettura/navigazione** XML. Restituisce il nome del nodo <RifNodo> o stringa vuota se fallisce. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array `_XMLSTRVAL$()` ed il numero degli elementi viene riportato in `_XMLVALNR`. Esempio:

`NOME$ = XMLGETELNAM$(NODO)`

`IF NOME$="" OR _ERRXML$<>"" THEN VIMSG 5," Impossibile leggere il nome del nodo: " + _ERRXML$:`

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETELTYPE

XMLGETELTYPE(<RifNodo>)

Parametri	
RifNodo	Riferimento del nodo

Funzione per la fase di **lettura/navigazione** XML. Restituisce il tipo del nodo <RifNodo> oppure zero in caso di errore. I tipi di nodo che possono essere ritornati sono:

0	MXML_UNKNOWN_NODE
1	MXML_ELEMENT_NODE
2	MXML_ATTRIBUTE_NODE
3	MXML_TEXT_NODE
4	MXML_CDATA_NODE
5	MXML_ENTITY_REFERENCE_NODE

6	MXML_ENTITY_NODE
7	MXML_PROCESSING_INSTRUCTION_NODE
8	MXML_COMMENT_NODE
9	MXML_DOCUMENT_NODE
10	MXML_DOCUMENT_TYPE_NODE
11	MXML_DOCUMENT_FRAGMENT_NODE
12	MXML_NOTATION_NODE
13	MXML_WHITESPACE_NODE
14	MXML_SIGNIFICANT_WHITESPACE_NODE
15	MXML_END_ELEMENT_NODE
16	MXML_END_ENTITY_NODE
17	MXML_XML_DECLARATION_NODE

Esempio:
TIPO = XMLGETELTYPE(NODO)
IFF _ERRXML\$<>""
 VIMSG 5," Impossibile leggere il tipo del nodo: " + _ERRXML\$:
ELSEF
 IF TIPO = 0 THEN ...
 IF TIPO = 1 THEN ...
 ...
 IF TIPO = 17 THEN ...
ENDF

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETELVAL

XMLGETELVAL\$(<RifNodo>)

Parametri	
RifNodo	Riferimento del nodo

Funzione per la fase di **lettura/navigazione** XML. Restituisce il valore del nodo <RifNodo> o stringa vuota se fallisce. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array _XMLSTRVAL\$() ed il numero degli elementi viene riportato in _XMLVALNR. Esempio:

VALORE\$ = XMLGETELVAL\$(NODO)
IFF VALORE \$="" OR _ERRXML\$<>"" THEN VIMSG 5," Impossibile leggere il valore del nodo: " + _ERRXML\$:

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETFSTATT

XMLGETFSTATT(<RifNodo>)

Parametri	
RifNodo	Riferimento del nodo

Funzione per la fase di **lettura/navigazione** XML. Acquisisce il riferimento al primo attributo del nodo <RifNodo>, ritorna il riferimento all'attributo oppure zero se l'operazione fallisce. Esempio:

ATTR = XMLGETFSTATT(NODO); IF ATTR=0 OR _ERRXML\$<>"" THEN VIMSG 5, _ERRXML\$:

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETNCONT

XMLGETNCONT\$(<RifDoc>,<RifNodo>)

Parametri	
RifDoc	Riferimento del documento
RifNodo	Riferimento del nodo

Funzione per la fase di **lettura/navigazione** XML. Acquisisce il contenuto del nodo identificato da Documento <RifDoc> e Nodo <RifNodo>. Ritorna una stringa con il contenuto o una stringa vuota se fallisce. Vengono inoltre valorizzate la variabile _XMLVALNR e l'array _XMLSTRVAL\$(). Nel caso il contenuto del nodo sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array _XMLSTRVAL\$() ed il numero degli elementi viene riportato in _XMLVALNR. Il valore restituito dalla funzione coincide con il valore del primo elemento dell'array. La variabile _ERRXML\$ contiene la descrizione di eventuali errori.

Esempio:
CONTENUTO\$ = XMLGETNCONT\$(XML_DOC, NODO1)
IFF CONTENUTO\$="" OR _ERRXML\$<>""
 VIMSG 5," Impossibile leggere il contenuto del nodo: " + _ERRXML\$:


```
ELSEF
  FOR I=1 TO _XMLVALNR
    LPRINT _XMLSTRVAL$(I)
  NEXT
ENDF
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETNODENS

XMLGETNODENS(<RifNodo>)

Parametri

RifNodo	Riferimento del nodo
---------	----------------------

Funzione per la fase di **lettura/navigazione** XML. Acquisisce il namespace del nodo <RifNodo> ne ritorna il riferimento oppure zero se fallisce.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETNSHRE

XMLGETNSHRE\$(<RifNameSpace>)

Parametri

RifNameSpace	Riferimento namespace
--------------	-----------------------

Funzione per la fase di **lettura/navigazione** XML. Acquisisce l'href del namespace con riferimento <RifNameSpace> e ne ritorna il valore oppure stringa vuota se fallisce. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array _XMLSTRVAL\$() ed il numero degli elementi viene riportato in _XMLVALNR.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETNSPRE

XMLGETNSPRE\$(<RifNameSpace>)

Parametri

RifNameSpace	Riferimento namespace
--------------	-----------------------

Funzione per la fase di **lettura/navigazione** XML. Acquisisce il prefix del namespace con riferimento <RifNameSpace>, ne ritorna il valore oppure stringa vuota se fallisce. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array _XMLSTRVAL\$() ed il numero degli elementi viene riportato in _XMLVALNR.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETNXTATT

XMLGETNXTATT(<RifAttr>)

Parametri

RifAttr	Riferimento attributo
---------	-----------------------

Funzione per la fase di **lettura/navigazione** XML. Acquisisce il riferimento all'attributo successivo a quello con riferimento <RifAttr>, ritorna il riferimento all'attributo oppure zero se l'operazione fallisce. Esempio:

```
ATTR = XMLGETNXTATT(NODO): IF ATTR=0 OR _ERRXML$<>"" THEN VMSG 5, _ERRXML$:
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETPARENT

XMLGETPARENT(<RifNodo>)

Parametri

RifNodo	Riferimento del nodo
---------	----------------------

Funzione per la fase di **lettura/navigazione** XML. Restituisce il riferimento al nodo padre del nodo <RifNodo> oppure 0 se il nodo padre non esiste. La variabile _ERRXML\$ conterrà in questo caso una descrizione dell'errore. Esempio:

```
NODO1 = XMLGETPARENT (NODO2)
```

```
IF NODO1=0 OR _ERRXML$<>"" THEN VMSG 5," Impossibile trovare il nodo padre: " + _ERRXML$:
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETROOT

XMLGETROOT(<RifDocumento\$>)

Parametri	
RifDocumento	Riferimento del documento

Funzione per la fase di **lettura/navigazione** XML. Restituisce il riferimento al nodo radice del documento XML <RifDocumento> oppure zero in caso di errore. La variabile _ERRXML\$ conterrà in questo caso una descrizione dell'errore. Esempio:
 ROOTNODE = XMLGETROOT(NEW_XML)
 IF ROOTNODE=0 OR _ERRXML\$<>"" THEN VIMSG 5," Impossibile trovare il nodo: " + _ERRXML\$:
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLGETSIBL

XMLGETSIBL(<RifNodo>)

Parametri	
RifNodo	Riferimento del Nodo

Funzione per la fase di **lettura/navigazione** XML. Restituisce il riferimento al nodo fratello del nodo <RifNodo> ritorna il riferimento al nodo fratello, 0 se nessun fratello esiste. La variabile _ERRXML\$ conterrà in questo caso una descrizione dell'errore. Esempio:
 NODO2 = XMLGETSIBL(NODO1)
 IF NODO2=0 OR _ERRXML\$<>"" THEN VIMSG 5," Impossibile trovare il nodo fratello: " + _ERRXML\$:
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLINIT

Istruzione per la fase **creazione** XML. Inizializza l'ambiente per la gestione XML
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLINSDATI

XMLINSDATI <FileXML\$>, <Tag\$()>, <FileOutput\$()>

Parametri	
FileXML\$	Percorso + Nome del file XML da variare
Tag\$()	Deve contenere il percorso ed il nome del nodo di cui si intende variare il valore; la sintassi consente anche di indicare qualora possano esistere più occorrenze dello stesso nodo, di quale occorrenza si vuole variare il valore
FileOutput\$()	Deve contenere il percorso + nome del file con il contenuto da inserire nel nodo il cui percorso di ricerca è indicato nel corrispondente elemento dell'Array <Tag\$()>. Il contenuto di questo file deve essere fornito con l'encoding corretto. Se il nodo indicato aveva nel file <FileXML\$> un precedente valore, questo viene sostituito dal nuovo valore.

Istruzione che consente la variazione del contenuto di uno o più nodi di un file XML senza allocarlo in memoria, ovvero, consente di inserire come contenuto di un nodo anche dati costruiti diversamente (per [esempio](#), come contenuto di un nodo un file .ZIP convertito codificato in Base64). Consente, quindi, di variare il contenuto di un nodo in un file XML anche quando questo, a causa delle sue dimensioni, non può essere caricato in memoria e/o successivamente modificato con le funzioni/istruzioni attualmente disponibili per la gestione di questo tipo di dati. L'istruzione valorizza le variabili _ERRXML e _ERRXML\$ relative all'esito dell'operazione.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLISBLANK

XMLISBLANK(<ValoreRaw\$>)

Parametri	
ValoreRaw\$	Stringa

Funzione per la fase di **lettura/navigazione** XML. Verifica se il valore contenuto in <ValoreRaw\$> contiene solo caratteri <SPAZIO>, <TAB>, <CR>, <LF> e ritorna 1 in caso positivo, 0 altrimenti.
 Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLLOAD

XMLLOAD(<NomeFile\$>)

Parametri	
NomeFile\$	File di lettura

Funzione per la fase di **lettura/navigazione** XML. Carica in memoria un documento XML leggendolo dal file <NomeFile\$>. Viene restituito il riferimento al file oppure zero se si sono verificati errori. La variabile `_ERRXML$` conterrà in questo caso una descrizione dell'errore. Nota bene: a prescindere dal formato di encoding del file di origine la funzione converte l'xml caricato in memoria nel formato UTF-8.

Esempio:

```
XMLDOC = XMLLOAD(<"C:\Prova.xml")
```

```
IF XMLDOC=0 OR _ERRXML$<"> THEN VIMSG 5," Impossibile aprire il file XML: " + _ERRXML$:
```

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLNEWDOC

XMLNEWDOC(<NomeNodoRadice\$>,<ValoreNodoRadice\$>)

Parametri

NomeNodoRadice\$	Nome assegnato al nodo radice
ValoreNodoRadice\$	Valore assegnato al nodo radice

Funzione per la fase **creazione** XML. Crea un nuovo documento XML con nodo radice che ha nome <NomeNodoRadice\$> e valore <ValoreNodoRadice\$>. Ritorna il riferimento <RifDocumento> al documento creato o zero in caso di errore, e viene restituito la descrizione dell'eventuale errore nella variabile `_ERRXML$`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSETNODVAL

XMLSETNODVAL <RifNodo>,<Valore\$>

Parametri

RifNodo	Riferimento del Nodo
Valore\$	Stringa per valorizzare

Istruzione per la fase **creazione** XML. Setta il valore del nodo <RifNodo> con il testo <Valore\$>. La variabile `_ERRXML$` viene valorizzata con la descrizione dell'eventuale errore. Istruzione applicabile solo a nodi di tipo `TEXT_NODE` e `CDATA_NODE`

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSNXTATTR

XMLSNXTATTR(<RifXmlSTREAM>)

Parametri

RifXmlSTREAM	Stream di lettura
--------------	-------------------

Funzione per la **lettura Xml modalità STREAM**. Acquisisce il prossimo attributo del nodo corrente. Ritorna 1 se l'attributo viene letto, 0 se non ci sono attributi, -1 in caso di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRDEPTH

XMLSRDEPTH(<RifXmlSTREAM>)

Parametri

RifXmlSTREAM	Stream di lettura
--------------	-------------------

Funzione per la **lettura Xml modalità STREAM**. Acquisisce la "profondità" del nodo corrente e ne ritorna il valore o -1 se fallisce.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRFREE

XMLSRFREE(<RifXmlSTREAM>)

Parametri

RifXmlSTREAM	Stream di lettura
--------------	-------------------

Istruzione per la **lettura Xml modalità STREAM**. Rilascia lo stream di lettura XML

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRFRSTATT

XMLSRFRSTATT(<RifXmlSTREAM>)

Parametri	
RifXmlSTREAM	Stream di lettura

Funzione per la **lettura Xml modalità STREAM**. Acquisisce il primo attributo del nodo corrente. Ritorna 1 se l'attributo viene letto, 0 se non ci sono attributi, -1 in caso di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRLOADFIL

XMLSRLOADFIL(<NomeFile\$>)

Parametri	
NomeFile\$	Documento XML

Funzione per la **lettura Xml modalità STREAM**. Inizializza la lettura del documento XML <NomeFile\$> in modalità STREAM. Ritorna il riferimento all'XmlStream oppure zero in caso di errore. La modalità stream non consente di navigare nel documento, ma di leggerlo sequenzialmente avanzando di nodo in nodo.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRNAME

XMLSRNAME\$(<RifXmlSTREAM>)

Parametri	
RifXmlSTREAM	Stream di lettura

Funzione per la **lettura Xml modalità STREAM**. Acquisisce e restituisce il nome del nodo corrente o stringa vuota se fallisce. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array `_XMLSTRVAL$()` ed il numero degli elementi viene riportato in `_XMLVALNR`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRNSATTR

XMLSRNSATTR(<RifXmlSTREAM>)

Parametri	
RifXmlSTREAM	Stream di lettura

Funzione per la **lettura Xml modalità STREAM**. Acquisisce il tipo oggetto del nodo corrente. Ritorna 1 se Dichiarazione namespace, 0 se attributo o altro tipo di nodo, -1 se errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRNSPREF

XMLSRNSPREF\$(<RifXmlSTREAM>)

Parametri	
RifXmlSTREAM	Stream di lettura

Funzione per la **lettura Xml modalità STREAM**. Acquisisce il namespace prefix del nodo corrente, ritorna il prefix se presente, stringa vuota altrimenti. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array `_XMLSTRVAL$()` ed il numero degli elementi viene riportato in `_XMLVALNR`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRNSURI

XMLSRNSURI\$(<RifXmlSTREAM>)

Parametri	
RifXmlSTREAM	Stream di lettura

Funzione per la **lettura Xml modalità STREAM**. Acquisisce il namespace uri del nodo corrente, ritorna l'URI se presente, stringa vuota altrimenti. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi

dell'array `_XMLSTRVAL$()` ed il numero degli elementi viene riportato in `_XMLVALNR`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRTYPE

XMLSRTYPE(<RifXmlISTREAM>)

Parametri

RifXmlISTREAM	Stream di lettura
---------------	-------------------

Funzione per la **lettura Xml modalità STREAM**. Acquisisce il tipo del nodo corrente e ne ritorna il valore numerico secondo la specifica tabella (vedi XMLGETELTYPE), oppure zero se sconosciuto.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRVALUE

XMLSRVALUE\$(<RifXmlISTREAM>)

Parametri

RifXmlISTREAM	Stream di lettura
---------------	-------------------

Funzione per la **lettura Xml modalità STREAM**. Acquisisce e restituisce il valore del nodo corrente o stringa vuota se fallisce. Nel caso il valore di ritorno sia una stringa di lunghezza superiore ai 255 caratteri lo stesso viene suddiviso negli elementi dell'array `_XMLSTRVAL$()` ed il numero degli elementi viene riportato in `_XMLVALNR`.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLSRWALK

XMLSRWALK(<RifXmlISTREAM>)

Parametri

RifXmlISTREAM	Stream di lettura
---------------	-------------------

Funzione per la **lettura in modalità STREAM**. Legge il prossimo nodo dello stream XML <RifXmlISTREAM>. Ritorna 1 se il nodo successivo viene letto, 0 se non ci sono più nodi, -1 in caso di errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLTRIMMA

XMLTRIMMA\$(<ValoreRaw\$>)

Parametri

ValoreRaw\$	Stringa
-------------	---------

Funzione per la fase di **lettura/navigazione XML**. Elimina i caratteri <SPAZIO>, <TAB>, <CR> ed <LF> presenti in testa ed in coda alla variabile <ValoreRaw\$> (stringa di lunghezza massima di 255 caratteri).

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

XMLUNLINKNOD

XMLUNLINKNOD(<RifNodo>)

Parametri

RifNodo	Riferimento del Nodo
---------	----------------------

Istruzione per la fase **creazione XML**. Elimina il nodo con riferimento <RifNodo>. La variabile `_ERRXML$` viene valorizzata con la descrizione dell'eventuale errore.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ZIP_CREA

ZIP_CREA <NomeArchivioZIP\$>, <Flag_Sovrascrivi_Aggiungi>, <PATH_da_eliminare>, <Array_File\$(>

Parametri

NomeArchivioZIP\$	Nome dell'archivio .ZIP da creare o a cui aggiungere file;
Flag_Sovrascrivi_Aggiungi	Valori ammessi: 1 = Sovrascrive il file se esistente; 0 = Se il file esiste, i file verranno aggiunti;
PATH_da_eliminare	Stringa contenente la parte iniziale del percorso dei file che non si vuole salvare nell'archivio .ZIP. Esempio:

	<p>Se si archiviassero i seguenti file...</p> <ul style="list-style-type: none"> • C:\DirectoryTemporanea\deploy\bin\comando.exe • C:\DirectoryTemporanea\deploy\lib\libreria.dll • C:\DirectoryTemporanea\deploy\conf\config.ini <p>...e si volesse ottenere un'estrazione dei file dall'archivio .ZIP nel percorso "D:\programmi\" in cui siano create le directory "\bin", "\lib", "\conf" e nessuna altra directory superiore a queste presente nel PATH di origine, occorre assegnare al parametro <PATH_da_eliminare> il valore "C:\DirectoryTemporanea\deploy\".</p>
Array_File\$()	Array (monodimensionale) contenente elenco file da inserire/aggiungere in archivio .ZIP <NomeArchivioZIP\$>.

Istruzione per gestire la creazione/implementazione di archivi file ".ZIP". Non consente redirectione lato Client (es. "<C:\TMP\FILE.ZIP"). La variabile _ERRZIP\$ viene valorizzata con la descrizione dell'eventuale errore occorso durante l'operazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ZIP_ESTRAI

ZIP_ESTRAI <NomeArchivioZIP\$>, <DirectoryEstrazione\$>, <FlagEliminaPath>

Parametri	
NomeArchivioZIP\$	Nome dell'archivio .ZIP da cui estrarre i file;
DirectoryEstrazione\$	Directory in cui estrarre i file;
FlagEliminaPath	Indica se conservare o eliminare il percorso dei singoli file durante l'estrazione; qualora si scelga di eliminare il path dei file, file con nomi identici che risiedevano in cartelle diverse creano un conflitto e verrà conservato solo l'ultimo che viene estratto. Valori consentiti: 1 = Elimina il PATH salvato; 0 = Conserva il PATH salvato per i singoli file.

Istruzione per gestire l'estrazione del contenuto di archivio ".ZIP". Non consente redirectione lato Client (es. "<C:\TMP\FILE.ZIP"). La variabile _ERRZIP\$ viene valorizzata con la descrizione dell'eventuale errore occorso durante l'operazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

ZIP_LISTA

ZIP_LISTA <NomeArchivioZIP\$>, <NomeFile\$()>, <DirectoryFile\$()>

Parametri	
NomeArchivioZIP\$	Nome dell'archivio .ZIP da cui estrarre i file;
NomeFile\$()	Array (monodimensionale) in cui vengono riportati i nomi di ogni file contenuto nell'archivio .ZIP. Esempio: il Nome salvato nell'elemento 1 è il nome file il cui corrispondente percorso è quello salvato nell'elemento 1 dell'array <DirectoryFile\$()>
DirectoryFile\$()	Directory del file di cui si vuole il contenuto;

Istruzione per gestire l'elencazione del contenuto di archivio ".ZIP". Non consente redirectione lato Client (es. "<C:\TMP\FILE.ZIP"). La variabile _ERRZIP\$ viene valorizzata con la descrizione dell'eventuale errore occorso durante l'operazione.

Per le logiche ed esempi consultare il relativo capitolo nella sezione dei Comandi per Argomento. [Clicca qui](#)

MDS - SHAKER

NOVITA' IMPORTANTI

**RIFERIMENTI ALLE CIRCOLARI DI RILASCIO ("CRS")
QUI RIPORTATE PER VERSIONE**

Versione 2019F (v 69.0 Gest)

La suite Shaker è disponibile anche a 64 bit (Java solo 64 bit e .NET a 32 bit e 64 bit). In ogni caso, le librerie 64 bit (.NET e JAVA) non consentono la gestione di file di dimensioni superiori a 2 GB.

File di Librerie distribuiti: MSprxDn.dll (v 7.0) - mxspx.dll (v 7.0). La versione .NET a 64 bit, ha una dipendenza implicita dalla libreria Visual Studio MSVCR100.dll, installabile direttamente con un applicativo Microsoft (Microsoft V.C++ 2010 SP1 Redistributable Package (x64)). La libreria JAVA è compatibile con la versione JDK 12, pertanto è distribuita solo a 64 bit. La libreria comune tra Windows e Linux è: MSprxJ.jar (v 7.0)

- Per Windows sono distribuite le seguenti librerie: mxspx.dll (v 7.0) - mxspxjni.dll (v 7.0)
- Per Linux sono distribuite le seguenti librerie: libmxspx.so.7.0.0.0 - libmxspxjni.so.7.0.0.0

Versione 2021A (v 80.5 Gest)

SHAKER 7.7

COMPONENTE .NET

La libreria è distribuita per sistemi a 32 bit ed a 64 bit. I file distribuiti sono:

- MSprxDn.dll (v 7.7.0.0)
- mxspx.dll (v 7.7.0.0)

COMPONENTE JAVA

La libreria principale è MSprxJ.jar (v 7.7.0.0) ed è distribuita solo a 64 bit per ambienti Windows e Linux. Compatibile con libreria java JDK 12.

Per Windows sono distribuite le librerie: mxspx.dll (v 7.7.0.0) e mxspxjni.dll (v 7.7.0.0)

Per Linux sono distribuite le librerie libmxspx.so.7.7.0.0 e libmxspxjni.so.7.7.0.0

Le versioni 64 bit (.NET e JAVA Windows) hanno una dipendenza implicita dalla libreria msvcr100.dll installabile mediante Microsoft Visual C++ 2010 SP1 Redistributable Package (x64). La versione a 64 bit, sia .NET, sia JAVA, non consente la gestione di file di dimensioni superiori a 2 GB.

INTRODUZIONE

Mexal Developer Suite è costituito da un insieme di tools per la realizzazione di verticalizzazioni sia in ambiente Passepartout che con i più evoluti ambienti di sviluppo. La Suite è composta da :

- SPRIX
- COLLAGE
- SHAKER
 1. .NET
 2. JAVA

Sprix è un linguaggio di sviluppo basic-like, integrato in Passepartout e permette la creazione di verticalizzazioni tramite una serie di primitive che permettono la gestione del video, l'accesso alla base dati di Passepartout, a data base SQL, e di dispositivi di I/O (seriali etc...).

Collage Permette di collegare script scritti in SPRIX ad eventi generati da Passepartout.

Da collage è possibile accedere allo stato della maschera (o della stampa) in esecuzione e variare ed inserire informazioni con logiche decise nella verticalizzazione.

L'ambiente di sviluppo per la realizzazione di verticalizzazioni SPRIX/COLLAGE è integrato in Passepartout. Esiste infatti un'apposita voce di menù la cui funzione è quella di lanciare l'ambiente di sviluppo che consente allo sviluppatore SPRIX/COLLAGE la realizzazione di programmi.

Shaker è una suite di componenti di sviluppo che consente di sviluppare programmi SPRIX ma con altri ambienti di sviluppo (OCX, .NET, JAVA). Richiede apposita Licenza MDS (Mexal Development Suite).

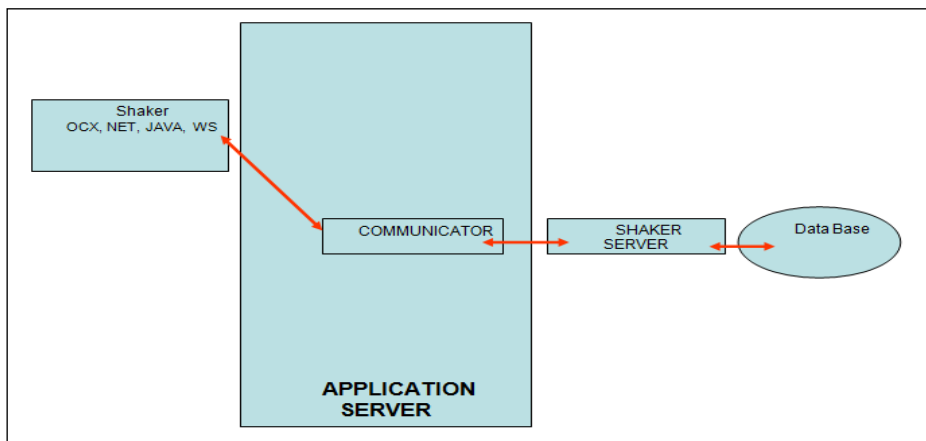
SPRIX e COLLAGE costituiscono un ottimo strumento per la realizzazione di verticalizzazioni Passepartout ed il linguaggio, basic-like, rende lo strumento di sviluppo semplice da utilizzare.

Attualmente sono presenti diversi e nuove versioni di strumenti di sviluppo tra cui Visual Basic e Java che si appoggiano alla tecnologia .NET. Risultano sempre più ricorrenti i termini di programmazione "object oriented" e programmazione ad eventi . Assume quindi un aspetto molto importante il fatto che, per quello che riguarda il mondo Microsoft, ci sia la possibilità realizzare applicazioni integrate con gli strumenti di OFFICE (tramite VBA).

SHAKER è costituito da un insieme di componenti per l'integrazione di Passepartout con i più moderni ambienti di sviluppo software.

- I componenti SHAKER permettono l'utilizzo delle primitive SPRIX in **Visual Basic 6, FoxPro, JAVA, .Net, Office, etc**
- I componenti SHAKER utilizzano la stessa tecnologia ed architettura Application Server di Passepartout, prevista per i client con interfaccia visuale proprietaria (iDesk, MobileDesk, WebDesk), connettendosi al servizio MxServer via socket TCP/IP.

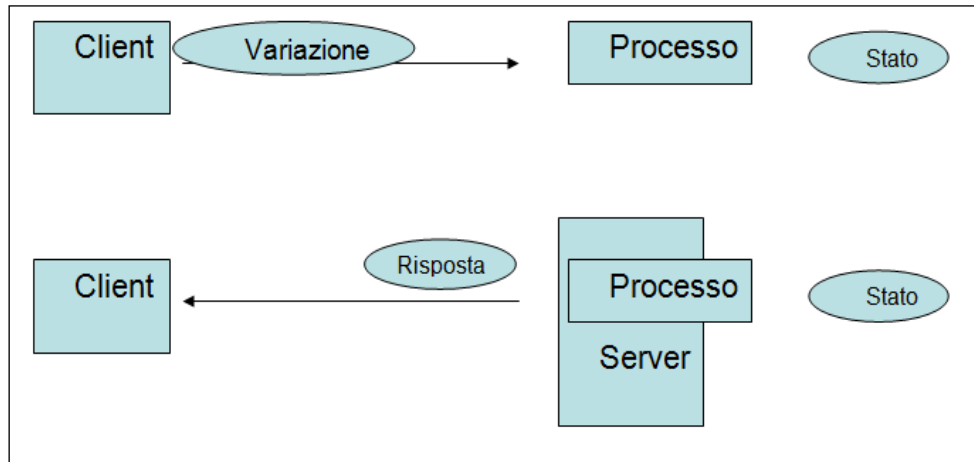
Architettura SHAKER



Caratteristiche tecniche

- Architettura basata sul modello State-Full
- Per ogni client c'è un processo in attesa
- Ad ogni cambiamento di stato nel client, questo deve comunicare esclusivamente le differenze rispetto allo stato precedente
- Sono stati implementati dei meccanismi di caching per minimizzare ulteriormente le comunicazioni

Modello State-Full



I componenti che costituiscono Shaker sono i seguenti:

- .NET
- JAVA

SHAKER - .NET

E' un componente scritto in C# utilizzabile da qualsiasi linguaggio della famiglia .Net

SHAKER VS SPRIX

I componenti SHAKER espongono un sottoinsieme dei comandi SPRIX.

Vengono espone SOLO istruzioni e funzioni che NON gestiscono l'I/O video e tastiera.

Noto l'utilizzo della tecnologia application server di Passepartout, sono state implementate nuove funzioni per la connessione al COMMUNICATOR.

Tutti i componenti SHAKER mettono a disposizione un insieme di proprietà e metodi.

PROPRIETÀ

Indichiamo con proprietà l'insieme delle variabili SPRIX di tipo _.

- _MMSIG\$ <componente>.MMSIG_S
- _MMNUM <componente>.MMNUM

Le variabili di tipo stringa che in SPRIX terminano con il carattere \$ nei componenti terminano con i caratteri _S.

METODI

Indichiamo con metodi l'insieme dei seguenti comandi SPRIX:

- Istruzioni: <componente>.GETMM_EXT("FT1")
- Funzioni: <componente>.DESART_S("AMARONE")
- Voci di dizionario: <componente>.DIZ("pcdes", 0, "201.1")

Per l'invocazione della lettura voci di dizionario il secondo parametro, tipo numerico che indica l'elemento dell'array da considerare, è obbligatorio.

I metodi permettono di:

- Utilizzare le istruzioni, le funzioni SPRIX e la lettura delle voci di dizionario (GETMM_EXT, PUTMM, GETAR, CAMBIOVAL...)
- Eseguire comandi SQL per RDBMS sul Server
- Eseguire operazioni su FILES del Server

Esempio di uso istruzioni, funzioni, lettura voci dizionario

- Istruzioni
<componente>.GETMM_EXT("FT1")
<componente>.GETMM_EXT("FT1","201.1") <componente>.GETAR("AMARONE",1)
- Funzioni
descArt = <componente>.DESART_S("AMARONE")
numStru = <componente>.GETNUMSTRUT("ARTICOLO")
val = <componente>.CAMBIOVAL(.....)
- Voci dizionario
desConto = <componente>.DIZ("pcdes", 0, "201.1")
saldoAnnoPrec = <componente>.DIZ("ppsal", 0, "201.1", 0)
saldoDare = <componente>.DIZ("ppdar", 0, "201.1", 0)

Il valore di ritorno delle funzioni e delle voci di dizionario è sempre di tipo stringa.

La conversione al tipo dato corretto viene lasciata al programmatore.

CONNESSIONE AL COMMUNICATOR PER INSTALLAZIONI NON LIVE

Il funzionamento di tutti i componenti SHAKER si basa sulla tecnologia Application Server di Passepartout.

E' quindi necessario connettersi al server COMMUNICATOR definendo quanto segue:

```
<componente>.INDIRIZZO = "<alias o IP server>"
<componente>.PORTA = <n.ro porta MxServer>
<componente>.LOGINMXSRV = "<utente SO>"
<componente>.PASSWORDMXSRV = "<password SO>"
<componente>.PASSWORDPASS = "<utente:password> gestionale"
<componente>.TERMINALE = "<n.ro terminale>"
<componente>.DATAAPTERM = "<data> nella forma gg/mm/aaaa"
<componente>.AZIENDA = "<sigla azienda>"
<componente>.AVVIACONNESSIONE
```

Esempio:

```
<componente>.INDIRIZZO = "mexal.rete.local"
<componente>.PORTA = 9000
<componente>.LOGINMXSRV = "mexaladm"
<componente>.PASSWORDMXSRV = "sK48pX82"
<componente>.PASSWORDPASS = "shaker001:mexal123"
<componente>.TERMINALE = "t0"
<componente>.DATAAPTERM = "31122020"
<componente>.AZIENDA = "PMF"
<componente>.AVVIACONNESSIONE
```

Il server deve consentire l'accesso in entrata alla porta per il client che desidera connettersi, mentre il client deve poter uscire sulla porta di destinazione del server che vuole contattare.

I metodi LOGINMXSRV e PASSWORDMXSRV indicano le credenziali per autenticarsi al sistema operativo, necessarie solo se MxServer prevede Login=1.

Il metodo PASSWORDPASS indica le credenziali per autenticarsi al gestionale e deve esprimersi nella forma "utente:password".

Ad ogni connessione corrisponde un terminale Passepartout.

Con gestione automatica del numero terminale (AUTOTERM), Shaker invoca sempre un terminale tipo ServizioL. In questo caso, il valore impostato nel metodo <componente>.TERMINALE non viene considerato dalla procedura.

In Passcom, Shaker risulta disponibile per utenti iDesk azienda AlwaysOn estesi ed estesi solution, mentre non lo risulta per iDesk azienda AlwaysOn base iDesk azienda base gestiti da studio.

Per aziende con gestione Art. 36 e sottoaziende definite, la sottoazienda da aprire viene dichiarata tramite metodo <componente>.AZIENDA = "<sigla azienda:codice>:<sottoazienda>".

Per il codice sotto azienda risultano ammessi valori numerici da 1 a 256 ed il carattere "G" indicante la sottoazienda generale. Qualora non sia indicato nulla, il terminale apre la sottoazienda prevalente.

Esempio:

```
MxSprix3_81.AZIENDA = "MAC:3"
```

Tutti gli errori relativi alla connessione sono presenti nella variabile

```
<componente>.ERRORE.
```

In questa proprietà vengono riportati :

eventuali errori di connessione

gli errori dati da Passepartout in fase di Start Up

gli errori di tipo SYSERR().

CONNESSIONE AL COMMUNICATOR PER INSTALLAZIONI LIVE

Risulta necessario abilitare in uscita le porte di destinazione:

- 443 verso gli host di destinazione:
 - start-live.passepartout.cloud;
 - start-live01.passepartout.cloud;
 - start-live02.passepartout.cloud;
- sul range 9000-9999.

Rispetto alle installazioni locali, ovvero installate presso la propria sede, l'accesso agli ambienti Live prevede tre metodi diversi per l'autenticazione ed uno distinto per la relativa gestione errore.

In particolare, non si utilizzano 5 metodi che definiscono il socket di MxServer (<componente>.INDIRIZZO, <componente>.PORTA), l'eventuale autenticazione di sistema operativo (<componente>.LOGINMXSRV, <componente>.PASSWORDMXSRV) e l'autenticazione al gestionale (<componente>.PASSWORDPASS).

L'autenticazione Live avviene impostando tre valori: dominio, utente e password, dove le credenziali rappresentano anche l'accesso automatico al gestionale. Pertanto, in ambiente Live si utilizzano tre distinti metodi:

```
<componente>.DOMINIOPREDESK = "<dominio Live>"  
<componente>.USERNAMEPREDESK = "<utente Live>"  
<componente>.PASSWORDPREDESK = "<password Live>"
```

L'avvio connessione avviene attraverso un metodo differente:
<componente>.AVVIACONNESSIONEPREDESK

A sua volta, la gestione errore di avvio connessione richiede un altro metodo:
<componente>.ERRORE_CONNESSIONEPREDESK

Esempio:

```
<componente>.DOMINIOPREDESK = "SIERRASISTEMI"  
<componente>.USERNAMEPREDESK = "MEXAL"  
<componente>.PASSWORDPREDESK = "MOCTAP1317"  
<componente>.TERMINALE = "t0"  
<componente>.DATAAPTERM = "31122020"  
<componente>.AZIENDA = "MA5:2"  
<componente>.AVVIACONNESSIONEPREDESK
```

Non variano utilizzo e significato di tutti gli altri metodi, tra cui <componente>.ERRORE per gestire errori di comunicazione dopo averla inizialmente stabilita (es. una chiusura incondizionata dallo strato fisico di rete).

DISCONNESSIONE DAL COMMUNICATOR

La corrispondente funzione associata alla <componente>.AVVIACONNESSIONE per installazioni non Live, oppure alla <componente>.AVVIACONNESSIONEPREDESK per ambienti Live, risulta:

<componente>.CHIUDICONNESSIONE

Nel caso in cui non si richiami esplicitamente il metodo per la chiusura connessione, l'applicazione, quando termina, richiama automaticamente la chiusura della connessione.

COMPONENTI

Versioni componenti OCX

Le diverse versioni dei componenti che verranno in futuro rilasciate richiederanno obbligatoriamente la ricompilazione delle verticalizzazioni distribuite. Questo vale solo per il componente OCX.

Utilizzo componente .NET: MSprixDn.dll

E' un componente scritto in C# che espone un insieme di metodi e proprietà "ereditati" da SPRIX ed è utilizzabile da qualsiasi linguaggio della famiglia .Net.

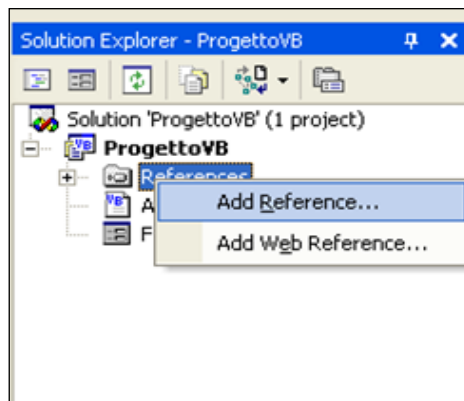
Per realizzare verticalizzazioni in ambienti di sviluppo della famiglia .NET è necessario quanto segue:

- Inserimento del componente .NET nell'ambiente di sviluppo
- Connessione al server COMMUNICATOR
- Richiamo di appropriati metodi SPRIX

Inserimento del componente .NET nell'ambiente di sviluppo

- Copiare in una cartella i files MSprixDn.dll e MxSpx.dll
- Includere il componente .NET come Reference
- Dichiarare il componente
Private componente as MSprixDn.MxSpxDotNet
- Istanziare il componente

Me.componente = New SprixDn.MxSpxDotNet(Me)



CONNESSIONI

Connessione al server COMMUNICATOR

Avviene specificando le proprietà richieste per la connessione e lanciando il metodo Me.componente.AVVIACONNESSIONE. L'eventuale errore di connessione viene riportato in Me.componente.ERRORE.

Chiusura connessione al server COMMUNICATOR

La chiusura della connessione con il server COMMUNICATOR può avvenire nei seguenti modi:

- Richiamando direttamente il metodo Me.componente.CHIUDICONNESSIONE
- Chiusura automatica con il Dispose.
Al richiamo del Dispose da parte del programma, automaticamente viene effettuato il Dispose del componente.
La dispose del componente effettua automaticamente la chiusura della connessione.
Si può comunque richiamare il metodo Me.componente.DISPOSE.

METODI

Richiamo di appropriati metodi SPRIX

In funzione del tipo di verticalizzazione che Si vuole realizzare verranno utilizzati gli opportuni metodi presenti nel componente.

Esempio di codice VB.NET per la lettura di movimenti di magazzino:

```
Dim iNumRigheDoc as Int32
```

```
Dim sDataDoc as String
Dim sCodiceArticolo(1000) as String
Dim i as Int32
```

```
Me.componente.GETMM_EXT ("FT1")
iNumRigheDoc = Me.componente.NMM
sDataDoc = Me.componente.MMDAT_S
For i = 1 To iNumRigheDoc
    sCodiceArticolo(i) = Me.componente.MMART_S(i)
Next i
```

I componenti SHAKER mettono a disposizione le istruzioni SPRIX per la gestione Files e l'esecuzione di statement SQL. L'esecuzione di questo set di istruzioni necessita, per il componente .NET, dell'utilizzo di:

1. tipo dato OBJECT
2. Definizione delle property
3. Metodo SETUTILIZZATORE(object)

Gli esempi seguenti, scritti in VB.NET, mostreranno come gestire tali istruzioni:

Letture del Codice articolo dalla tabella AR_ANAGRAFICHE

Definizione della PROPERTY

```
Public Class Form1
    Dim sCodArtV As String
    Public Property PsCodArtV() As String // As Object
        Get
            Return sCodArtV
        End Get
        Set(ByVal Value As String)
            sCodArtV = Value
        End Set
    End Property

    'Alternativa alla property precedente
    'Public WriteOnly Property PsCodArtV() As String
    '    Set(ByVal Value As String)
    '        sCodArtV = Value
    '    End Set
    'End Property
```

```
.....
.....
'Utilizzo del metodo SQL_BINDCOL con riferimento alla property "PsCodArtV"
SqlCommand = "SELECT CKY_ART FROM AR_ANAGRAFICHE"
Me.componente.SQL_ALLOCSTMT(1, 1)
Me.componente.SQL_PREPARE(1, 1, SqlCommand)
Me.componente.SQL_BINDCOL(1, 1, 1, "A", "PsCodArtV")
Me.componente.SQL_FETCH(1, 1)
sCodArticolo = sCodArtV
```

Istanze componente:metodo SETUTILIZZATORE(object)

Ipotizzando l'utilizzo di una unica istanza del componente da più classi, ogni classe che utilizza il componente e utilizza metodi di gestione Files o Statment SQL deve richiamare il metodo componente.SETUTILIZZATORE(Me) prima che la classe utilizzi il componente.

Questo metodo viene richiamato per indicare al componente la classe in cui sono state definiti gli attributi e le relative PROPERTY.

Esempi di verticalizzazione utilizzando il componente .NET

Ipotizziamo di realizzare un'applicazione VB che gestisce un solo form da cui si possono effettuare letture di documenti di magazzino, lettura e scrittura files ed esecuzione di query SQL.

```
Public Class Form1
    Inherits System.Windows.Forms.Form

    Dim campo1_1 As Object
    Dim campo1_2 As Object
    Dim campo2_1 As Object
    Dim campo2_2 As Object

    Dim sCodArtV As String
    Dim sUmV As String
    Dim sForAbV As String
    Dim nCstdV As Double

    Dim codV As Object
    Dim desV As Object
```

Dim des1V As Object
Dim saldodV As Object
Dim aStr As String
Dim bStr As String

DEFINIZIONE DELLE PROPERTY PER GESTIONE CAMPI FILES E CAMPI TABELLE

```
Public Property PCampo1_1() As Object
    Get
        Return campo1_1
    End Get
    Set(ByVal Value As Object)
        campo1_1 = Value
    End Set
End Property
```

```
Public Property PCampo1_2() As Object
    Get
        Return campo1_2
    End Get
    Set(ByVal Value As Object)
        campo1_2 = Value
    End Set
End Property
```

```
Public Property PCampo2_1() As Object
    Get
        Return campo2_1
    End Get
    Set(ByVal Value As Object)
        campo2_1 = Value
    End Set
End Property
```

```
Public Property PCampo2_2() As Object
    Get
        Return campo2_2
    End Get
    Set(ByVal Value As Object)
        campo2_2 = Value
    End Set
End Property
```

```
Public WriteOnly Property PsCodArtV() As String
    Set(ByVal Value As String)
        sCodArtV = Value
    End Set
End Property
```

```
Public Property PsUmV() As String
    Get
        Return sUmV
    End Get
    Set(ByVal Value As String)
        sUmV = Value
    End Set
End Property
```

```
Public Property PsForAbV() As String
    Get
        Return sForAbV
    End Get
    Set(ByVal Value As String)
        sForAbV = Value
    End Set
End Property
```

```
Public Property PnCstdV() As Double
    Get
        Return nCstdV
    End Get
    Set(ByVal Value As Double)
        nCstdV = Value
    End Set
End Property
```



```

Public Property PcodV() As Object
    Get
        Return codV
    End Get
    Set(ByVal Value As Object)
        codV = Value
    End Set
End Property

Public Property PdesV() As Object
    Get
        Return desV
    End Get
    Set(ByVal Value As Object)
        desV = Value
    End Set
End Property

Public Property Pdes1V() As Object
    Get
        Return des1V
    End Get
    Set(ByVal Value As Object)
        des1V = Value
    End Set
End Property

Public Property PsaldodV() As Object
    Get
        Return saldodV
    End Get
    Set(ByVal Value As Object)
        saldodV = Value
    End Set
End Property

Public Property PaStr() As String
    Get
        Return aStr
    End Get
    Set(ByVal Value As String)
        aStr = Value
    End Set
End Property

Public Property PbStr() As String
    Get
        Return bStr
    End Get
    Set(ByVal Value As String)
        bStr = Value
    End Set
End Property

```

FINE DEFINIZIONE DELLE PROPERTY PER GESTIONE CAMPI FILES E CAMPI TABELLE

```

#Region " Windows Form Designer generated code "
'DICHIARAZIONE DEL COMPONENTE
Public componente As MSprixDn.MxSpxDotNet

Public Sub New()
    MyBase.New()

    'This call is required by the Windows Form Designer.
    InitializeComponent()

    'Add any initialization after the InitializeComponent() call
    'IL COMPONENTE VIENE ISTANZIATO
    Me.componente = New MSprixDn.MxSpxDotNet(Me)
End Sub
'Form overrides dispose to clean up the component list.
Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
        If Not (components Is Nothing) Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(disposing)
End Sub

```

'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.

#End Region

CONNESSIONE AL SERVER COMUNICATORE

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim errore As String

Me.componente.PORTA = 9100
Me.componente.INDIRIZZO = "gfiumana"
Me.componente.PASSWORDPASS = ""
Me.componente.LOGINMSRV = "utente1"
Me.componente.PASSWORDMSRV = "utente1"
Me.componente.DATAAPTERM = "31122001"
Me.componente.TERMINALE = "t0"
Me.componente.AZIENDA = "SUB"
Me.componente.AVVIACONNESSIONE()

errore = Me.componente.ERRORE
if(errore <> "") Then
 MessageBox.Show(errore)
End If

End Sub

INIZIO ESEMPI CODICE PER LA LETTURA MOVIMENTO DI MAGAZZINO

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

 Call EmRevDoc()
End Sub

Private Sub EmRevDoc()

Dim a As String
Dim i As Long
Dim numAbs As Double
Dim cod As String

Me.componente.AZZVARSYS(4)
Me.componente.AZZVARSYS(3)

Me.componente.GETMM_EXT(TBSiglaDoc.Text + TBNumero.Text)

a = Me.componente.ERRMM_S

If a <> String.Empty Then
 MessageBox.Show(a)
 Me.componente.ERRMM_S = String.Empty
Else
 Call VisualizzaRighe()
End If

End Sub

Private Sub VisualizzaRighe()

Dim i As Int32
Dim iNumRigheDoc As Double
Dim cod As String
Dim tot As Double
Dim a As String
Dim cnt As String
Dim mag As Double
Dim datadoc As String

Dim dt As New System.Data.DataTable()
Dim riga As System.Data.DataRow

```

Dim colonna1 As System.Data.DataColumn
Dim colonna2 As System.Data.DataColumn
Dim colonna3 As System.Data.DataColumn
Dim colonna4 As System.Data.DataColumn

colonna1 = New DataColumn("Codice articolo")
colonna1.DataType = System.Type.GetType("System.String")
colonna2 = New DataColumn("Quantita")
colonna2.DataType = System.Type.GetType("System.Int32")
colonna3 = New DataColumn("Prezzo")
colonna3.DataType = System.Type.GetType("System.Double")
colonna4 = New DataColumn("Sconto")
colonna4.DataType = System.Type.GetType("System.String")
dt.Columns.Add(colonna1)
dt.Columns.Add(colonna2)
dt.Columns.Add(colonna3)
dt.Columns.Add(colonna4)

```

```

TBSiglaDoc.Text = Me.componente.MMSIG_S
TBNumero.Text = Me.componente.MMNUM.ToString()
mag = Me.componente.MMMAG
iNumRigheDoc = Me.componente.NMM
datadoc = Me.componente.MMDAT_S

```

```

TBContoCF.Text = Me.componente.MMCLI_S
TBContoAgente.Text = Me.componente.MMAGE_S
cnt = TBContoCF.Text
TBDescrizioneCF.Text = Me.componente.DIZ("pcdes", 0, cnt)
cnt = TBContoAgente.Text
TBDescrizioneAgente.Text = Me.componente.DIZ("pcdes", 0, cnt)

```

```

Dim conto As String
Dim stringa As String
conto = Me.componente.MMCLI_S
stringa = Me.componente.DIZ("pcdes", 0, conto)
conto = Me.componente.MMAGE_S
stringa = Me.componente.DIZ("pcdes", 0, conto)

```

```

Dim j As Integer
For j = 1 To iNumRigheDoc
    riga = dt.NewRow()
    riga("Codice articolo") = Me.componente.MMART_S(j)
    riga("Quantita") = Me.componente.MMQTA(j)
    riga("Prezzo") = Me.componente.MMPRZ(j)
    riga("Sconto") = Me.componente.MMSCO_S(j)
    dt.Rows.Add(riga)
Next j

```

```

DGEEmRevDoc.SetDataBinding(dt, "")
End Sub

```

INIZIO ESEMPI CODICE PER LA GESTIONE FILES

```

'Apri file
Private Sub BApriFile_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BApriFile.Click

```

```

    'Apertura file non ad indici
    Me.componente.OPEN("R", 1, "C:\DANIELE\file.TXT", 100)
    Me.componente.FIELD(1, 10, "PCampo1_1", "")
    Me.componente.FIELD(1, 90, "PCampo2_1", "NUM2")
End Sub

```

```

'Scrivi file
Private Sub BScriviFile_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BScriviFile.Click
    Call ScriviFile()
End Sub

```

```

'Leggi file
Private Sub BLeggiFile_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BLeggiFile.Click
    Call LeggiFile()
End Sub

```

```

Private Sub ScriviFile()
    campo1_1 = TBCampo1.Text
    campo2_1 = Convert.ToDouble(TBCampo2.Text)
    Me.componente.PUT(1, 1)
End Sub
Private Sub LeggiFile()

```

```
TBFile.Text = String.Empty
campo1_1 = ""
campo2_1 = 0
Me.componente.GET(1, 1)
Dim s1 As String
s1 = CType(campo1_1, String)
Dim s2 As String
s2 = CType(campo2_1, String)
TBFile.Text = s1.Replace(Chr(0), " "c) & s2
End Sub
```

```
'Chiudi file
Private Sub BChiudiFile_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BChiudiFile.Click
    Call Me.componente.CLOSE(1)
End Sub
```

INIZIO ESEMPI CODICE PER LA LETTURA TABELLA DATA BASE

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
```

```
    Call LeggiDB()
End Sub
```

```
Private Sub LeggiDB()
```

```
    Dim i As Long
```

```
    Me.componente.SQL_CONNECT(1, "SUBDBMIRROR", "", "")
```

```
    If componente.ERRSQL_M_S <> "" Then
```

```
        MessageBox.Show("errore Connect")
```

```
    Else
```

```
        Dim SqlCommand As String
```

```
        SqlCommand = "SELECT CKY_ART , CSG_UNIMIS_PRI, NMP_COSTD, CKY_CNT_FOR_AB FROM AR_ANAGRAFICHE"
```

```
        Me.componente.SQL_ALLOCSTMT(1, 1)
```

```
        Me.componente.SQL_PREPARE(1, 1, SqlCommand)
```

```
        Me.componente.SQL_BINDCOL(1, 1, 1, "A", "PsCodArtV")
```

```
        Me.componente.SQL_BINDCOL(1, 1, 2, "A", "PsUmV")
```

```
        Me.componente.SQL_BINDCOL(1, 1, 4, "A", "PsForAbV")
```

```
        Me.componente.SQL_BINDCOL(1, 1, 3, "N", "PnCstdV")
```

```
        Me.componente.SQL_EXECUTE(1, 1)
```

```
        Dim dataTable As System.Data.DataTable
```

```
        Dim riga As System.Data.DataRow
```

```
        Dim colonna1 As System.Data.DataColumn
```

```
        Dim colonna2 As System.Data.DataColumn
```

```
        Dim colonna3 As System.Data.DataColumn
```

```
        Dim colonna4 As System.Data.DataColumn
```

```
        dataTable = New DataTable()
```

```
        colonna1 = New DataColumn("Codice articolo")
```

```
        colonna2 = New DataColumn("Unita misura")
```

```
        colonna3 = New DataColumn("Costo standard")
```

```
        colonna4 = New DataColumn("Fornitura abituale")
```

```
        dataTable.Columns.Add(colonna1)
```

```
        dataTable.Columns.Add(colonna2)
```

```
        dataTable.Columns.Add(colonna3)
```

```
        dataTable.Columns.Add(colonna4)
```

```
        Dim conta As Int32
```

```
        Dim ancora As Boolean
```

```
        conta = 0
```

```
        ancora = True
```

```
        While (ancora)
```

```
            Me.componente.SQL_FETCH(1, 1)
```

```
            riga = dataTable.NewRow()
```

```
            riga("Codice articolo") = sCodArtV
```

```
            riga("Unita misura") = sUmV
```

```
            riga("Fornitura abituale") = sForAbV
```

```
            riga("Costo standard") = nCstdV
```

```
            dataTable.Rows.Add(riga)
```

```
            conta = conta + 1
```

```
        If Me.componente.ERRSQL_NDF <> 0 Then
```

```
            ancora = False
```

```
        End If
```

```
    End While
```

```
    DGDatabase.SetDataBinding(dataTable, "")
```

```
End If
```

```
Me.componente.SQL_DISCONNECT(1)
End Sub
```

```
End Class
```

Esempio di applicazione MDI in cui tutte le FORM MDI CHILD utilizzano una sola istanza del componente istanziato nel FORM MDI CONTAINER.

- Dichiarazione del componente
- Istanziare il componente
- Creazione di un metodo (o property di get) pubblico che restituisce il componente
- Accesso al metodo che restituisce il componente da parte dei FORM figli
- Utilizzo del metodo SETUTILIZZATORE() dai FORM MDI

Codice relativo al form MDI CONTAINER:

```
Public Class Form1
    Inherits System.Windows.Forms.Form

    #Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()
        'Add any initialization after the InitializeComponent() call
        #2 Istanza del componente .NET
        Me.componente = New MSprixDn.MxSpxDotNet(Me)
    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
    #1 DICHIARAZIONE DEL COMPONENTE
    Private componente As MSprixDn.MxSpxDotNet

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents MainMenu1 As System.Windows.Forms.MainMenu
    Friend WithEvents MenuItem1 As System.Windows.Forms.MenuItem
    Friend WithEvents MenuItem2 As System.Windows.Forms.MenuItem
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.MainMenu1 = New System.Windows.Forms.MainMenu()
        Me.MenuItem1 = New System.Windows.Forms.MenuItem()
        Me.MenuItem2 = New System.Windows.Forms.MenuItem()
        '
        'MainMenu1
        '
        Me.MainMenu1.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.MenuItem1})
        '
        'MenuItem1
        '
        Me.MenuItem1.Index = 0
        Me.MenuItem1.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.MenuItem2})
        Me.MenuItem1.Text = "Magazzino"
        '
        'MenuItem2
        '
        Me.MenuItem2.Index = 0
        Me.MenuItem2.Text = "EmissioneRevisioneDocumenti"
        '
    End Sub

    Form1
End Class
```

```

Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(292, 273)
Me.IsMdiContainer = True
Me.Menu = Me.MainMenu1
Me.Name = "Form1"
Me.Text = "ProgettoProva"
Me.WindowState = System.Windows.Forms.FormWindowState.Maximized

```

End Sub

#End Region

'#3

**'Definisco una property per accedere al componente in sola lettura
'da parte delle form figlie del form contenitore MDI CONTAINER**

Public ReadOnly Property MxComponente() As MSprixDn.MxSpxDotNet

Get

Return Me.componente

End Get

End Property

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

'Connessione al server COMUNICATOR

Me.componente.PORTA = 9100

Me.componente.INDIRIZZO = "gfiumana"

Me.componente.LOGINMXSRV = "utente1"

Me.componente.PASSWORDMXSRV = "utente1"

Me.componente.PASSWORDPASS = ""

Me.componente.TERMINALE = "t0"

Me.componente.DATAAPTERM = "31122000"

Me.componente.AZIENDA = "SUB"

Me.componente.AVVIACONNESSIONE()

If (Me.componente.ERRORE <> "") Then

MsgBox(Me.componente.ERRORE)

End If

End Sub

Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem2.Click

Dim f1 As Form2

f1 = New Form2()

f1.MdiParent = Me

f1.Show()

End Sub

End Class

Codice relativo al form MDI CHILD:

Public Class Form2

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

Public Sub New()

MyBase.New()

'This call is required by the Windows Form Designer.

InitializeComponent()

'Add any initialization after the InitializeComponent() call

End Sub

'Form overrides dispose to clean up the component list.

Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)

If disposing Then

If Not (components Is Nothing) Then

components.Dispose()

End If

End If

MyBase.Dispose(disposing)

End Sub

'Required by the Windows Form Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

Friend WithEvents SigDoc As System.Windows.Forms.TextBox

Friend WithEvents NumDoc As System.Windows.Forms.TextBox

Friend WithEvents Button1 As System.Windows.Forms.Button

Friend WithEvents DGERevDoc As System.Windows.Forms.DataGrid

Friend WithEvents Conto As System.Windows.Forms.TextBox

```

Friend WithEvents DesConto As System.Windows.Forms.TextBox
<System.Diagnostics.DebuggerStepThrough(> Private Sub InitializeComponent()
    Me.SigDoc = New System.Windows.Forms.TextBox()
    Me.NumDoc = New System.Windows.Forms.TextBox()
    Me.Button1 = New System.Windows.Forms.Button()
    Me.DGEmRevDoc = New System.Windows.Forms.DataGrid()
    Me.Conto = New System.Windows.Forms.TextBox()
    Me.DesConto = New System.Windows.Forms.TextBox()
    CType(Me.DGEmRevDoc, System.ComponentModel.ISupportInitialize).BeginInit()
    Me.SuspendLayout()
    '
    'SigDoc
    '
    Me.SigDoc.Location = New System.Drawing.Point(40, 8)
    Me.SigDoc.Name = "SigDoc"
    Me.SigDoc.Size = New System.Drawing.Size(40, 20)
    Me.SigDoc.TabIndex = 0
    Me.SigDoc.Text = "FT"
    '
    'NumDoc
    '
    Me.NumDoc.Location = New System.Drawing.Point(40, 32)
    Me.NumDoc.Name = "NumDoc"
    Me.NumDoc.Size = New System.Drawing.Size(24, 20)
    Me.NumDoc.TabIndex = 1
    Me.NumDoc.Text = "1"
    '
    'Button1
    '
    Me.Button1.Location = New System.Drawing.Point(40, 200)
    Me.Button1.Name = "Button1"
    Me.Button1.Size = New System.Drawing.Size(64, 32)
    Me.Button1.TabIndex = 2
    Me.Button1.Text = "Button1"
    '
    'DGEmRevDoc
    '
    Me.DGEmRevDoc.Anchor = (((System.Windows.Forms.AnchorStyles.Top Or System.Windows.Forms.AnchorStyles.Bottom) _
        Or System.Windows.Forms.AnchorStyles.Left) _
        Or System.Windows.Forms.AnchorStyles.Right)
    Me.DGEmRevDoc.DataMember = ""
    Me.DGEmRevDoc.HeaderForeColor = System.Drawing.SystemColors.ControlText
    Me.DGEmRevDoc.Location = New System.Drawing.Point(288, 8)
    Me.DGEmRevDoc.Name = "DGEmRevDoc"
    Me.DGEmRevDoc.Size = New System.Drawing.Size(448, 372)
    Me.DGEmRevDoc.TabIndex = 3
    '
    'Conto
    '
    Me.Conto.Location = New System.Drawing.Point(40, 56)
    Me.Conto.Name = "Conto"
    Me.Conto.Size = New System.Drawing.Size(64, 20)
    Me.Conto.TabIndex = 4
    Me.Conto.Text = ""
    '
    'DesConto
    '
    Me.DesConto.Location = New System.Drawing.Point(112, 56)
    Me.DesConto.Name = "DesConto"
    Me.DesConto.Size = New System.Drawing.Size(168, 20)
    Me.DesConto.TabIndex = 5
    Me.DesConto.Text = ""
    '
    'Form2
    '
    Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
    Me.ClientSize = New System.Drawing.Size(744, 389)
    Me.Controls.AddRange(New System.Windows.Forms.Control() {Me.DesConto, Me.Conto, Me.DGEmRevDoc, Me.Button1, Me.NumDoc,
Me.SigDoc})
    Me.Name = "Form2"
    Me.Text = "EmissioneRevisioneDocumenti"
    CType(Me.DGEmRevDoc, System.ComponentModel.ISupportInitialize).EndInit()
    Me.ResumeLayout(False)

End Sub

```

#End Region

'PROPERTY per il ritorno del componente del Form padre tramite la property 'definita nel form padre non è necessaria. Volendo invece di richiamare 'Me.MxComponente. avremmo dovuto richiamare il codice 'all'interno della Get

```
Private ReadOnly Property MxComponente() As MSprixDn.MxSpxDotNet
```

Get

Dim f As Form1

f = Me.MdiParent

'#1 ritorno del componente del Form padre tramite la property 'definita nel form padre

Return f.MxComponente

End Get

End Property

'Se invece di effettuare una lettura di movimento di magazzino avessimo 'effettuato la lettura
'di una tabella del DB sarebbe stato necessario richiamare il metodo 'SETUTILIZZATORE(Me)

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim doc As String

Dim errGet As String

Dim errCon As String

Dim ok As Boolean

N.B. Se avessimo effettuato operazioni su file o DB avremmo dovuto eseguire

Me.MxComponente.SETUTILIZZATORE(Me)

ok = False

doc = SigDoc.Text + NumDoc.Text

Me.MxComponente.GETMM_EXT(doc)

errCon = Me.MxComponente.ERRORRE

If (errCon <> "") Then

MsgBox(errCon)

Else

errGet = Me.MxComponente.ERRMM_S

If (errGet <> "") Then

MsgBox(errGet)

Else

ok = True

End If

End If

If (ok = True) Then

VisualizzaDoc()

End If

End Sub

Private Sub VisualizzaDoc()

Dim iNumRigheDoc As Double

Dim cnt As String

Dim mag As Double

Dim datadoc As String

'Creo il datatable che sarà utilizzato per riempire il datagrid

Dim dt As New System.Data.DataTable()

Dim riga As System.Data.DataRow

Dim colonna1 As System.Data.DataColumn

Dim colonna2 As System.Data.DataColumn

Dim colonna3 As System.Data.DataColumn

Dim colonna4 As System.Data.DataColumn

'Vengono create tante colonne quanti sono i dati che voglio visualizzare

colonna1 = New DataColumn("Codice articolo")

colonna1.DataType = System.Type.GetType("System.String")

colonna2 = New DataColumn("Quantita")

colonna2.DataType = System.Type.GetType("System.Int32")

colonna3 = New DataColumn("Prezzo")

colonna3.DataType = System.Type.GetType("System.Double")

colonna4 = New DataColumn("Sconto")

colonna4.DataType = System.Type.GetType("System.String")

dt.Columns.Add(colonna1)

dt.Columns.Add(colonna2)

dt.Columns.Add(colonna3)

dt.Columns.Add(colonna4)

'SigDoc.Text = Me.MxComponente.MMSIG_S

'NumDoc.Text = Me.MxComponente.MMNUM.ToString()

'TBSerie.Text = Me.MxComponente.MMSER.ToString()

'mag = Me.MxComponente.MMMAG

iNumRigheDoc = Me.MxComponente.NMM

datadoc = Me.MxComponente.MMDAT_S


```

Conto.Text = Me.MxComponente.MMCLI_S
TBContoAgente.Text = Me.MxComponente.MMAGE_S
'cnt = TBContoCF.Text
DesConto.Text = Me.MxComponente.DIZ("pcdes", 0, Conto.Text)
'cnt = TBContoAgente.Text
'TBDescrizioneAgente.Text = Me.MxComponente.DIZ("pcdes", 0, cnt)

```

```

'Vengono create le righe e riempite con i dati presi dal componente
Dim j As Integer
For j = 1 To iNumRigheDoc
    riga = dt.NewRow()
    riga("Codice articolo") = Me.MxComponente.MMART_S(j)
    riga("Quantita") = Me.MxComponente.MMQTA(j)
    riga("Prezzo") = Me.MxComponente.MMPRZ(j)
    riga("Sconto") = Me.MxComponente.MMSCO_S(j)
    dt.Rows.Add(riga)
Next j

```

```
DGEmRevDoc.SetDataBinding(dt, "")
```

```
End Sub
```

```
End Class
```

WEB SERVICES

Permettono la comunicazione tra applicazioni indipendentemente dal linguaggio in cui sono state scritte o dal sistema operativo in cui stanno girando. Chi invoca un Web services non deve sapere come viene implementato il servizio; deve solo conoscere che parametri passargli e cosa aspettarsi come risultato

- Un Web Service NON ha mai un'Interfaccia Utente
- Ad esempio un Web Service scritto in JAVA che gira in Linux può essere invocato da un'applicazione .Net e viceversa
- Le funzioni esposte devono avere una "granularità" più alta per evitare troppe interazioni (costose!) tra chi invoca ed il server

La realizzazione di WEB SERVICES integrati con Mexal avviene tramite la costruzione di procedure realizzate con il componente .NET. Il metodo esposto dal WEB SERVICES non è altro che una verticalizzazione su Mexal.

Per esempio si può realizzare una GetMMMacro che esegue una GetMMe e ritorna le variabili in un unico file XML.

JAVA

Il componente Java MSPRIXJ permette di realizzare verticalizzazioni sia in sistemi operativi WINDOWS che LINUX. Esso, come tutti i componenti SHAKER, mette a disposizione un insieme di proprietà e metodi ereditati da SPRIX. Per il funzionamento del componente JAVA viene rilasciata, oltre alla DDL MxSpx.dll, la DLL JMxSpx.dll.

In generale, relativamente ai sistemi operativi WINDOWS e LINUX, è necessario utilizzare i seguenti files:

```

- WINDOWS
    MxSpx.dll
    JMxSpx.dll
    MSprjJ.jar (contenente la classe MSprjJ.class si deve effettuare l'import da JAVA.)

- LINUX
    libMxSpx.so
    libMxSpx.so.0
    libMxSpx.so.0.1
    libJMxSpx.so
    libJMxSpx.so.0
    libJMxSpx.so.1
    MSprjJ.jar (contenente la classe MSprjJ.class si deve effettuare l'import da JAVA.)

```

JAVA VIRTUAL MACHINE DA UTILIZZARE

Versione 1.4.2 distribuita da SUN. Collegarsi al sito java.sun.com e scaricare il relativo SDK per LINUX o WINDOWS.

Solo a titolo di esempio (in quanto materia che dovrebbe essere già conosciuta dallo sviluppatore) ricordiamo la sequenza di operazioni propedeutiche per evitare casi Could not initialize class sm.passepartout.MSprjJ.

Per l'uso di componente JAVA in LINUX occorre:

- 1) copiare le dll ed il file MSprjJ.jar nella cartella dove è collocato l'applicativo java realizzato.
- 2) creare manualmente i link (con "ln -s") alle librerie "madrì" libmxspx_s.so e libJMxSpx.so:

```
ln -s libJMxSpx.so.X.Y libJMxSpx.so
```

```
In -s libmxsp_x_s.so.X.Y libmxsp_x_s.so
```

3) impostare la variabile di ambiente LD_LIBRARY_PATH (affinchè la DLL sia visibile dal componente Java) nel seguente modo: export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:"percorso relativo alla cartella in cui sono collocati l'applicativo realizzato, il file MSprxJ.jar e le dll (tutte quelle relative al componente JAVA)"

4) impostare l'eventuale CLASSPATH (specificando il percorso in cui della directory bin della java virtual machine, come da esempio:set PATH=%PATH%;.;C:\Program Files\jdk\j2sdk_nb\j2sdk1.4.2\bin)

5) lancio dell'applicativo realizzato con il comando: java -cp MSprxJ.jar:."nome applicativo"

UTILIZZO DEL COMPONENTE MSPRIXJ

L'utilizzo del componente MSPRIXJ è lo stesso dei componenti OCX e .NET. A differenza dei componenti OCX e .NET la valorizzazione e la lettura dei valori delle variabili SPRIX avviene tramite metodi di set e get:

N.B. Indichiamo con <MSPRIXJ> un'istanza del componente JAVA creata nel seguente modo:

```
MSprxJ <MSPRIXJ> = new MSprxJ();
```

- es. Uso della variabile _MMART\$(i):

- Assegnamento del valore "ARTICOLO1" all'iesima riga di un movimento di magazzino

```
String codArt;  
codArt = "ARTICOLO1";  
<MSPRIXJ>.setMMART_S(i,codArt);
```

- Lettura dell'articolo relativo all'iesima riga di un movimento di magazzino

```
String codArt;  
codArt = <MSPRIXJ>.getMMART_S(i);
```

METODI E PARAMETRI FACOLTATIVI

La definizione dei parametri facoltativi, relativi ad alcuni metodi, avviene definendo degli array di oggetti:

- es. metodo DIZ

```
<MSPRIXJ>.DIZ("pcdes", 0, new Object[]{"501.1"} );
```

LA LETTURA E SCRITTURA DI FILES

La lettura e scrittura di files o l'esecuzione di query SQL avviene utilizzando due metodi:

<MSPRIXJ>.SetProperty() e <MSPRIXJ>.GetProperty().

- es. Lettura e scrittura di un file

- Apertura del file

```
<MSPRIXJ>.OPEN("R", 1, "C:\\PIPPA.TXT", 80, new Object[{}]);
```

- Definizione delle FIELD

```
String p1 = "Nome";  
String p2 = "Cognome";  
String p3 = "Indirizzo";  
String p4 = "Telefono";
```

```
<MSPRIXJ>.FIELD(1, 20, p1, "ALFA");  
<MSPRIXJ>.FIELD(1, 20, p2, "ALFA");  
<MSPRIXJ>.FIELD(1, 20, p3, "ALFA");  
<MSPRIXJ>.FIELD(1, 20, p4, "ALFA");
```

- Scrittura di un record nel file

La valorizzazione dei campi indicati nella FIELD deve avvenire utilizzando il metodo SetProperty.

I parametri da passare alla SetProperty sono i seguenti:

1 - Identificativo del campo (p1,p2,p3,p4 come indicato nella FIELD)

2 - Il valore del campo che deve essere obbligatoriamente uno dei seguenti tipi: Integer, Long, Float, Double, String.

In caso contrario viene sollevata un'eccezione.

```
String p1 = "Nome";  
String p2 = "Cognome";  
String p3 = "Indirizzo";  
String p4 = "Telefono";
```

```
<MSPRIXJ>.SetProperty(p1, "PIPPA");  
<MSPRIXJ>.SetProperty(p2, "PLUTO");  
<MSPRIXJ>.SetProperty(p3, "PAPERINO");  
<MSPRIXJ>.SetProperty(p4, "TOPOLINO");
```

- Lettura di un record dal file
 La lettura dei campi indicati nella FIELD deve avvenire utilizzando il metodo GetProperty.
 Il parametro da passare alla GetProperty è il seguente:
 1 - Identificativo del campo (p1,p2,p3,p4 come indicato nella FIELD)

```
String p1 = "Nome";
    String p2 = "Cognome";
    String p3 = "Indirizzo";
    String p4 = "Telefono";
String varStr;

<MSPRIXJ>.GET(1, 1);

    varStr = first.GetProperty(p1);
    .....
```

ESEMPIO JAVA che utilizza il componente SHAKER MSPRIXJ.

Esso contiene il codice relativo alla gestione di Documenti di magazzino e relativo alla lettura e scrittura di files

```
/*
 * ProvaComponente.java
 *
 * Created on 15 luglio 2004, 11.47
 */

import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowListener;
import java.awt.event.WindowEvent;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTabbedPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.border.TitledBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JSpinner;
import javax.swing.SpinnerNumberModel;

// IMPORT FONDAMENTALE PER UTILIZZO DEL COMPONENTE JAVA
import sm.passepartout.MSprixJ;

public class ProvaComponente
{
    public static void main(String[] args)
    {
        avviaConnessione();
    }

    public static void program(final MSprixJ first)
    {
        JFrame f = new JFrame();

        //f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                // MSprixJ
                // ULTIMO PASSO: chiusura della connessione.
                first.CHIUDICONNESSIONE();
                System.exit(0);
            }
        });

        f.getContentPane().setLayout(new GridBagLayout());

        JPanel documentiPanel = new JPanel();
        documentiPanel.setLayout(new GridBagLayout());
        GridBagConstraints gbc;
```

```
final JTextField documento = new JTextField("documento");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.gridwidth = 1;
gbc.gridheight = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
documentiPanel.add(documento, gbc);

final JTextField cliente = new JTextField("cliente");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 1;
gbc.gridwidth = 1;
gbc.gridheight = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
documentiPanel.add(cliente, gbc);

final JTextField descrizioneCliente = new JTextField("descrizioneCliente");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 1;
gbc.gridwidth = 1;
gbc.gridheight = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
documentiPanel.add(descrizioneCliente, gbc);

final JTextField agente = new JTextField("agente");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 2;
gbc.gridwidth = 1;
gbc.gridheight = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
documentiPanel.add(agente, gbc);

final JTextField descrizioneAgente = new JTextField("descrizioneAgente");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 2;
gbc.gridwidth = 1;
gbc.gridheight = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
documentiPanel.add(descrizioneAgente, gbc);

final JTable tabella = new JTable(100, 4);
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 3;
gbc.gridwidth = 1;
gbc.gridheight = 1;
gbc.gridwidth = 2;
gbc.fill = GridBagConstraints.BOTH;
gbc.weightx = 1.0;
gbc.weighty = 1.0;
documentiPanel.add(new JScrollPane(tabella), gbc);

JButton pulsante = new JButton("Leggi");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 4;
gbc.gridwidth = 1;
gbc.gridheight = 1;
//gbc.gridwidth = 1;
documentiPanel.add(pulsante, gbc);

pulsante.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        first.GETMM_EXT(documento.getText());
        if (!first.getERRMM_S().equals(""))
        {
            JOptionPane.showMessageDialog(null, first.getERRMM_S());
        }
        else
        {
            cliente.setText(first.getMMCLI_S());
            // METODO DIZ: i parametri "opzionali" devono essere passati in un array
        }
    }
});
```

```

// di Object
String desCliente = first.DIZ("pcdes", 0, new Object[] {cliente.getText()});
descrizioneCliente.setText(desCliente);
agente.setText(first.getMMAGE_S());
descrizioneAgente.setText(first.DIZ("pcdes", 0, new Object[] {agente.getText()});

double righeDocumento = first.getNMM();
tabella.setModel(new DefaultTableModel(100, 4));
for (int i = 1; i <= righeDocumento; i++)
{
    String codiceArticolo = first.getMMART_S(i);
    double qta = first.getMMQTA(i);
    double prz = first.getMMPRZ(i);
    tabella.setValueAt(codiceArticolo, i - 1, 0);
    tabella.setValueAt(new Double(qta), i - 1, 1);
    tabella.setValueAt(new Double(prz), i - 1, 2);
}
}
});

```

```

JPanel gestioneFilePanel = new JPanel();
gestioneFilePanel.setLayout(new GridBagLayout());
gbc = new GridBagConstraints();

```

```

JPanel filePanel = new JPanel();
filePanel.setLayout(new GridBagLayout());
filePanel.setBorder(new TitledBorder("File"));

```

```

JLabel tipoLabel = new JLabel("Tipo: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.anchor = GridBagConstraints.EAST;
filePanel.add(tipoLabel, gbc);

```

```

final JTextField tipo = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 0;
gbc.fill = GridBagConstraints.HORIZONTAL;
filePanel.add(tipo, gbc);
tipo.setText("R");

```

```

JLabel numLabel = new JLabel("Num: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 1;
gbc.anchor = GridBagConstraints.EAST;
filePanel.add(numLabel, gbc);

```

```

final JTextField num = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
filePanel.add(num, gbc);
num.setText("1");
JLabel nomeLabel = new JLabel("Nome: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 2;
gbc.anchor = GridBagConstraints.EAST;
filePanel.add(nomeLabel, gbc);

```

```

final JTextField nome = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 2;
gbc.fill = GridBagConstraints.HORIZONTAL;
filePanel.add(nome, gbc);
nome.setText("C:\\PIPPO.TXT");
JLabel lunghezzaLabel = new JLabel("Lunghezza record: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 3;
gbc.anchor = GridBagConstraints.EAST;
filePanel.add(lunghezzaLabel, gbc);

```

```

final JTextField lunghezza = new JTextField(10);
gbc = new GridBagConstraints();

```

```
gbc.gridx = 1;
gbc.gridy = 3;
gbc.fill = GridBagConstraints.HORIZONTAL;
filePanel.add(lunghezza, gbc);
lunghezza.setText("80");

JButton open = new JButton("Open");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 4;
filePanel.add(open, gbc);
open.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        first.OPEN(tipo.getText(), Integer.parseInt(num.getText()), nome.getText(), Integer.parseInt(lunghezza.getText()), new Object[]{});

        if (!first.getERRORE().equals(""))
        {
            JOptionPane.showMessageDialog(null, first.getERRORE());
        }
        else
        {
            {
            }
        }
    }
});
JButton close = new JButton("Close");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 5;
filePanel.add(close, gbc);

JPanel fieldPanel = new JPanel();
fieldPanel.setLayout(new GridBagLayout());
fieldPanel.setBorder(new TitledBorder("Field"));

// primo field

JLabel putColumn = new JLabel("Put Valori");
gbc = new GridBagConstraints();
gbc.gridx = 2;
gbc.gridy = 0;
gbc.insets = new Insets(0, 10, 5, 5);
fieldPanel.add(putColumn, gbc);

JLabel getColumn = new JLabel("Get Valori");
gbc = new GridBagConstraints();
gbc.gridx = 3;
gbc.gridy = 0;
gbc.insets = new Insets(0, 5, 5, 10);
fieldPanel.add(getColumn, gbc);

JLabel numero = new JLabel("Numero file: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 1;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(numero, gbc);

final JTextField file1 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(file1, gbc);
file1.setText("1");

JLabel lunghezzaCampoLabel = new JLabel("Lunghezza campo: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 2;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(lunghezzaCampoLabel, gbc);

final JTextField lunghezzaCampo = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 2;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(lunghezzaCampo, gbc);
```

```

lunghezzaCampo.setText("20");

JLabel valoreLabel = new JLabel("Valore: ");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 3;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(valoreLabel, gbc);

final JTextField putValori = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 2;
gbc.gridy = 3;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(putValori, gbc);

final JTextField getValori = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 3;
gbc.gridy = 3;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(getValori, gbc);

JLabel tipoImpaccamentoLabel = new JLabel("Tipo impaccamento: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 4;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(tipoImpaccamentoLabel, gbc);

final JTextField tipoImpaccamento = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 4;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(tipoImpaccamento, gbc);
    tipoImpaccamento.setText("ALFA");

// secondo field

JLabel numero2 = new JLabel("Numero file: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 5;
gbc.anchor = GridBagConstraints.EAST;
gbc.insets = new Insets(20, 0, 0, 0);

fieldPanel.add(numero2, gbc);

final JTextField file2 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 5;
gbc.fill = GridBagConstraints.HORIZONTAL;
gbc.insets = new Insets(20, 0, 0, 0);
fieldPanel.add(file2, gbc);

JLabel lunghezzaCampoLabel2 = new JLabel("Lunghezza campo: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 6;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(lunghezzaCampoLabel2, gbc);

final JTextField lunghezzaCampo2 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 6;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(lunghezzaCampo2, gbc);

JLabel valoreLabel2 = new JLabel("Valore: ");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 7;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(valoreLabel2, gbc);

final JTextField putValori2 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 2;

```

```
gbc.gridy = 7;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(putValori2, gbc);

final JTextField getValori2 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 3;
gbc.gridy = 7;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(getValori2, gbc);

JLabel tipolImpaccamentoLabel2 = new JLabel("Tipo impaccamento: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 8;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(tipolImpaccamentoLabel2, gbc);

final JTextField tipolImpaccamento2 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 8;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(tipolImpaccamento2, gbc);

// terzo field

JLabel numero3 = new JLabel("Numero file: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 9;
gbc.insets = new Insets(20, 0, 0, 0);
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(numero3, gbc);

final JTextField file3 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 9;
gbc.fill = GridBagConstraints.HORIZONTAL;
gbc.insets = new Insets(20, 0, 0, 0);
fieldPanel.add(file3, gbc);

JLabel lunghezzaCampoLabel3 = new JLabel("Lunghezza campo: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 10;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(lunghezzaCampoLabel3, gbc);

final JTextField lunghezzaCampo3 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 10;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(lunghezzaCampo3, gbc);

JLabel valoreLabel3 = new JLabel("Valore: ");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 11;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(valoreLabel3, gbc);

final JTextField putValori3 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 2;
gbc.gridy = 11;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(putValori3, gbc);

final JTextField getValori3 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 3;
gbc.gridy = 11;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(getValori3, gbc);

JLabel tipolImpaccamentoLabel3 = new JLabel("Tipo impaccamento: ");
gbc = new GridBagConstraints();
```



```

gbc.gridx = 0;
gbc.gridy = 12;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(tipolImpaccamentoLabel3, gbc);

final JTextField tipolImpaccamento3 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 12;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(tipolImpaccamento3, gbc);

// quarto field

JLabel numero4 = new JLabel("Numero file: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 13;
gbc.anchor = GridBagConstraints.EAST;
gbc.insets = new Insets(20, 0, 0, 0);

fieldPanel.add(numero4, gbc);

final JTextField file4 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 13;
gbc.fill = GridBagConstraints.HORIZONTAL;
gbc.insets = new Insets(20, 0, 0, 0);
fieldPanel.add(file4, gbc);

JLabel lunghezzaCampoLabel4 = new JLabel("Lunghezza campo: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 14;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(lunghezzaCampoLabel4, gbc);

final JTextField lunghezzaCampo4 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 14;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(lunghezzaCampo4, gbc);

JLabel valoreLabel4 = new JLabel("Valore: ");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 15;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(valoreLabel4, gbc);

final JTextField putValori4 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 2;
gbc.gridy = 15;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(putValori4, gbc);

final JTextField getValori4 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 3;
gbc.gridy = 15;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(getValori4, gbc);

JLabel tipolImpaccamentoLabel4 = new JLabel("Tipo impaccamento: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 16;
gbc.anchor = GridBagConstraints.EAST;
fieldPanel.add(tipolImpaccamentoLabel4, gbc);

final JTextField tipolImpaccamento4 = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 16;
gbc.fill = GridBagConstraints.HORIZONTAL;
fieldPanel.add(tipolImpaccamento4, gbc);

JButton buttonField = new JButton("Field");

```

```

gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 17;
fieldPanel.add(buttonField, gbc);
buttonField.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        // Nell'esecuzione delle FIELD deve essere passato l'identificatore del
        // campo.
        // Nell'esempio p1,p2,p3,p4 indicano tale identificativo.
        String p1 = "Nome";
        String p2 = "Cognome";
        String p3 = "Indirizzo";
        String p4 = "Telefono";
        String varStr;

tipolmpaccamento.getText();      first.FIELD(Integer.parseInt(file1.getText()),      Integer.parseInt(lunghezzaCampo.getText()),      p1,
        varStr = first.getERRORE();
        if (!varStr.equals(""))
        {
            JOptionPane.showMessageDialog(null, varStr);
            System.exit(0);
        }

tipolmpaccamento2.getText();      first.FIELD(Integer.parseInt(file2.getText()),      Integer.parseInt(lunghezzaCampo2.getText()),      p2,
        varStr = first.getERRORE();
        if (!varStr.equals(""))
        {
            JOptionPane.showMessageDialog(null, varStr);
            System.exit(0);
        }

tipolmpaccamento3.getText();      first.FIELD(Integer.parseInt(file3.getText()),      Integer.parseInt(lunghezzaCampo3.getText()),      p3,
        varStr = first.getERRORE();
        if (!varStr.equals(""))
        {
            JOptionPane.showMessageDialog(null, varStr);
            System.exit(0);
        }

tipolmpaccamento4.getText();      first.FIELD(Integer.parseInt(file4.getText()),      Integer.parseInt(lunghezzaCampo4.getText()),      p4,
        varStr = first.getERRORE();
        if (!varStr.equals(""))
        {
            JOptionPane.showMessageDialog(null, varStr);
            System.exit(0);
        }
    }
});

JPanel putPanel = new JPanel();
putPanel.setLayout(new GridBagLayout());
putPanel.setBorder(new TitledBorder("Put"));

JLabel putNumeroFileLabel = new JLabel("Numero file:");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.anchor = GridBagConstraints.EAST;
putPanel.add(putNumeroFileLabel, gbc);

final JTextField putNumeroFile = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 0;
gbc.fill = GridBagConstraints.HORIZONTAL;
putPanel.add(putNumeroFile, gbc);

JLabel putNumeroRecordLabel = new JLabel("Numero record:");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 1;
gbc.anchor = GridBagConstraints.EAST;
putPanel.add(putNumeroRecordLabel, gbc);

```

```

final JTextField putNumeroRecord = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
putPanel.add(putNumeroRecord, gbc);

JButton put = new JButton("Put");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 2;
putPanel.add(put, gbc);
put.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        // La valorizzazione dei campi indicati nella FIELD deve avvenire
        // utilizzando il metodo SetProperty.
        // I parametri da passare alla SetProperty sono i seguenti:
        // 1 - Identificativo del campo (p1,p2,p3,p4 come indicato nella FIELD )
        // 2 - Il valore del campo che deve essere obbligatoriamente uno dei
        //    seguenti tipi: Integer, Long, Float, Double, String.
        //    In caso contrario viene sollevata un'eccezione.

        String p1 = "Nome";
        String p2 = "Cognome";
        String p3 = "Indirizzo";
        String p4 = "Telefono";

        String varStr;

        first.SetProperty(p1, putValori.getText());
        first.SetProperty(p2, putValori2.getText());
        first.SetProperty(p3, putValori3.getText());
        first.SetProperty(p4, putValori4.getText());

        int numfil = Integer.parseInt(putNumeroFile.getText());
        int numrec = Integer.parseInt(putNumeroRecord.getText());
        //first.PUT(Integer.parseInt(putNumeroFile.getText()), Integer.parseInt(putNumeroRecord.getText()));

        first.PUT(numfil, numrec);

        varStr = first.getERRORE();
        if (!varStr.equals(""))
        {
            JOptionPane.showMessageDialog(null, varStr);
            System.exit(0);
        }
    }
});

JPanel getPanel = new JPanel();
getPanel.setLayout(new GridBagLayout());
getPanel.setBorder(new TitledBorder("Get"));

JLabel getNumeroFileLabel = new JLabel("Numero file:");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.anchor = GridBagConstraints.EAST;
getPanel.add(getNumeroFileLabel, gbc);

final JTextField getNumeroFile = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 0;
gbc.fill = GridBagConstraints.HORIZONTAL;
getPanel.add(getNumeroFile, gbc);

JLabel getNumeroRecordLabel = new JLabel("Numero record:");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 1;
gbc.anchor = GridBagConstraints.EAST;
getPanel.add(getNumeroRecordLabel, gbc);

final JSpinner getNumeroRecord = new JSpinner(new SpinnerNumberModel(1, 1, 3, 1)); // JTextField
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;

```

```

getPanel.add(getNumeroRecord, gbc);

JButton get = new JButton("Get");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 2;
getPanel.add(get, gbc);
get.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        // La lettura dei campi indicati nella FIELD deve avvenire
        // utilizzando il metodo GetProperty.
        // Il parametro da passare alla GetProperty è il seguente:
        // 1 - Identificativo del campo (p1,p2,p3,p4 come indicato nella FIELD )

        String p1 = "Nome";
        String p2 = "Cognome";
        String p3 = "Indirizzo";
        String p4 = "Telefono";

        first.GET(Integer.parseInt(getNumeroFile.getText()), ((Integer)getNumeroRecord.getValue()).intValue());

        getValori.setText((String) first.GetProperty(p1));
        getValori2.setText((String) first.GetProperty(p2));
        getValori3.setText((String) first.GetProperty(p3));
        getValori4.setText((String)first.GetProperty(p4));
    }
});

// pannello generale

gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.anchor = GridBagConstraints.NORTH;
gestioneFilePanel.add(filePanel, gbc);

gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 0;
gestioneFilePanel.add(fieldPanel, gbc);

JPanel thirdPanel = new JPanel();
thirdPanel.setLayout(new GridBagLayout());
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
thirdPanel.add(putPanel, gbc);

gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 1;
thirdPanel.add(getPanel, gbc);

gbc = new GridBagConstraints();
gbc.gridx = 2;
gbc.gridy = 0;
gbc.anchor = GridBagConstraints.NORTH;
gestioneFilePanel.add(thirdPanel, gbc);

JTabbedPane tabbedPane = new JTabbedPane();
tabbedPane.add("Emissione/Revisione Documenti", documentiPanel);
tabbedPane.add("Gestione File", gestioneFilePanel);

f.getContentPane().add(tabbedPane);

f.pack();
f.show();
}

public static void avviaConnessione()
{
    // MSprixJ
    // 1 PASSO: creazione di una nuova istanza del componente Java.
    final MSprixJ first = new MSprixJ();

    final JFrame f = new JFrame();

    f.getContentPane().setLayout(new GridBagLayout());

```

```

GridBagConstraints gbc;

// MSprixJ
// 2 PASSO: definizione dei parametri di connessione al server MEXAL
// tramite le relative proprietà di set.

JLabel PORTALabel = new JLabel("PORTA: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 0;
gbc.anchor = GridBagConstraints.EAST;
f.getContentPane().add(PORTALabel, gbc);

final JTextField PORTA = new JTextField(10);
PORTA.setText("9000");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 0;
gbc.fill = GridBagConstraints.HORIZONTAL;
f.getContentPane().add(PORTA, gbc);

JLabel INDIRIZZOLabel = new JLabel("INDIRIZZO: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 1;
gbc.anchor = GridBagConstraints.EAST;
f.getContentPane().add(INDIRIZZOLabel, gbc);

final JTextField INDIRIZZO = new JTextField(10);
INDIRIZZO.setText("localhost");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 1;
gbc.fill = GridBagConstraints.HORIZONTAL;
f.getContentPane().add(INDIRIZZO, gbc);

JLabel LOGINMXSRVLabel = new JLabel("LOGINMXSRV: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 2;
gbc.anchor = GridBagConstraints.EAST;
f.getContentPane().add(LOGINMXSRVLabel, gbc);

final JTextField LOGINMXSRV = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 2;
gbc.fill = GridBagConstraints.HORIZONTAL;
f.getContentPane().add(LOGINMXSRV, gbc);

JLabel PASSWORDMXSRVLabel = new JLabel("PASSWORDMXSRV: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 3;
gbc.anchor = GridBagConstraints.EAST;
f.getContentPane().add(PASSWORDMXSRVLabel, gbc);

final JTextField PASSWORDMXSRV = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 3;
gbc.fill = GridBagConstraints.HORIZONTAL;
f.getContentPane().add(PASSWORDMXSRV, gbc);

JLabel PASSWORDPASSLabel = new JLabel("PASSWORDPASS: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 4;
gbc.anchor = GridBagConstraints.EAST;
f.getContentPane().add(PASSWORDPASSLabel, gbc);

final JTextField PASSWORDPASS = new JTextField(10);
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 4;
gbc.fill = GridBagConstraints.HORIZONTAL;
f.getContentPane().add(PASSWORDPASS, gbc);

JLabel AZIENDALabel = new JLabel("AZIENDA: ");
gbc = new GridBagConstraints();

```

```

gbc.gridx = 0;
gbc.gridy = 5;
gbc.anchor = GridBagConstraints.EAST;
f.getContentPane().add(AZIENDALabel, gbc);

final JTextField AZIENDA = new JTextField(10);
AZIENDA.setText("SUB");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 5;
gbc.fill = GridBagConstraints.HORIZONTAL;
f.getContentPane().add(AZIENDA, gbc);

JLabel TERMINALELabel = new JLabel("TERMINALE: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 6;
gbc.anchor = GridBagConstraints.EAST;
f.getContentPane().add(TERMINALELabel, gbc);

final JTextField TERMINALE = new JTextField(10);
TERMINALE.setText("t0");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 6;
gbc.fill = GridBagConstraints.HORIZONTAL;
f.getContentPane().add(TERMINALE, gbc);

JLabel DATAAPTERMLabel = new JLabel("DATAAPTERM: ");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 7;
gbc.anchor = GridBagConstraints.EAST;
f.getContentPane().add(DATAAPTERMLabel, gbc);

final JTextField DATAAPTERM = new JTextField(10);
DATAAPTERM.setText("31 122000");
gbc = new GridBagConstraints();
gbc.gridx = 1;
gbc.gridy = 7;
gbc.fill = GridBagConstraints.HORIZONTAL;
f.getContentPane().add(DATAAPTERM, gbc);

JButton connect = new JButton("Connect");
gbc = new GridBagConstraints();
gbc.gridx = 0;
gbc.gridy = 8;
gbc.gridwidth = 2;
f.getContentPane().add(connect, gbc);
connect.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        first.setPORTA(Integer.parseInt(PORTA.getText()));
        first.setINDIRIZZO(INDIRIZZO.getText());
        first.setLOGINMXSRV(LOGINMXSRV.getText());
        first.setPASSWORDMXSRV(PASSWORDMXSRV.getText());
        first.setPASSWORDPASS(PASSWORDPASS.getText());
        first.setAZIENDA(AZIENDA.getText());
        first.setTERMINALE(TERMINALE.getText());
        first.setDATAAPTERM(DATAAPTERM.getText());

        // MSprixJ
        // 3 PASSO: Avvio della connessione
        first.AVVIACONNESSIONE();

        // MSprixJ
        // 4 PASSO: controllo dell'avvenuta connessione
        String varStr = first.getERRORE();

        if (!varStr.equals(""))
        {
            JOptionPane.showMessageDialog(null, varStr);
            System.exit(0);
        }

        f.setVisible(false);
        f.dispose();
        program(first);
    }
});

```

```

    }
  });

  f.addWindowListener(new WindowAdapter()
  {
      public void windowClosing(WindowEvent e)
      {
          System.exit(0);
      }
  });

  f.pack();
  f.show();
}
}

```

Letture scrittura file

Dichiarazione variabili globali

```

Option Explicit
Dim campo1V As Variant
Dim campo2V As Variant
Dim campo3V As Variant
Dim campo4V As Variant

```

Apertura file "file.txt"

```

Private Sub bOpenFile()
Dim Tipo As String
Dim nFil As Integer
Dim nomeFil As String
Dim IRec As Integer

Tipo = "R"
nFil = 1
nomeFil = "file.txt"
IRec = 100
Call MxSprix1.OPEN(Tipo, nFil, nomeFil, IRec)
End Sub

```

Definizione del tracciato record

```

Private Sub bField ()
Dim numFileField As Integer
Dim lunCampo As Integer
Dim tipolmp As String

campo1V = ""
campo2V = ""
campo3V = ""
campo4V = ""

numFileField = 1
lunCampo = 10
tipolmp = "ALFA"
Call MxSprix1.FIELD(numFileField, lunCampo, campo1V, tipolmp)

numFileField = 2
lunCampo = 6
tipolmp = "NUM0"
Call MxSprix1.FIELD(numFileField, lunCampo, campo2V, tipolmp)

numFileField = 3
lunCampo = 84
tipolmp = "ALFA"
Call MxSprix1.FIELD(numFileField, lunCampo, campo3V, tipolmp)
End Sub

```

Letture record file

```

Private Sub bLeggiRecord()
Dim nFileGet As Integer
Dim nRecGet As Integer
Dim dNum As Double
Dim str1 As String
Dim str2 As String
Dim i As Integer

nFileGet = 1
nRecGet = 1

```

Call MxSprix1.Get(nFileGet, nRecGet)

```
str1 = campo1V
dNum = campo2V
str2 = campo3V
End Sub
```

Scrittura Record

```
Private Sub ScriviRecord()
Dim nFilePut As Integer
Dim nRecPut As Integer
```

```
campo1V = "BRAVO"
campo2V = 10
campo3V = "CHARLY"
```

```
nFilePut = 1
nRecPut = 1
Call MxSprix1.Put(nFilePut, nRecPut)
End Sub
```

Statement SQL: FETCH

Letture campi tabella AR_ANAGRAFICHE

```
Private Sub LeggiTabellaArticoli()
Dim ancora As Boolean
Dim SqlCommand As String
Dim nStmt As Integer
Dim StrErr As String
Dim sCodArtV As Variant
Dim sUmV As Variant
Dim nCstdV As Variant
Dim sForAbV As Variant
Dim sCodArt As String
Dim sUm As String
Dim nCstd As Double
Dim sForAb As String
```

```
nStmt = 1
ancora = True
```

```
'Connessione al DB
Call MxSprix1.SQL_CONNECT(1, "SUBDBMIRROR", "", "")
If (MxSprix1.ERRORSQL_M_S <> "") Then
MsgBox "ERRORE CONNESSIONE DB"
End If
```

```
SqlCommand = "SELECT CKY_ART , CSG_UNIMIS_PRI, NMP_COSTD, CKY_CNT_FOR_AB FROM AR_ANAGRAFICHE"
```

```
Call MxSprix1.SQL_ALLOCSTMT(1, nStmt)
Call MxSprix1.SQL_PREPARE(1, nStmt, SqlCommand)
Call MxSprix1.SQL_BINDCOL(1, nStmt, 1, "A", sCodArtV)
Call MxSprix1.SQL_BINDCOL(1, nStmt, 2, "A", sUmV)
Call MxSprix1.SQL_BINDCOL(1, nStmt, 3, "N", nCstdV)
Call MxSprix1.SQL_BINDCOL(1, nStmt, 4, "A", sForAbV)
i = MxSprix1.SQL_EXECUTE(1, nStmt)
```

```
While (ancora = True)
Call MxSprix1.SQL_FETCH(1, nStmt)
If (MxSprix1.ERRORSQL_NDF <> 0) Or (MxSprix1.ERRORSQL_M_S <> "") Then
ancora = False
Else
sCodArt = sCodArtV
sUm = sUmV
sForAb = sForAbV
nCstd = nCstdV
End If
Wend
```

```
Call MxSprix1.SQL_DISCONNECT(1)
```

```
End Sub
```


APPENDICE: Comandi per Versione (GEST/DR)

AVVERTENZE

La seguente sezione elenca i comandi rilasciati per ciascuna versione Gestionale/Redditi.

Se vengono indicati dei contenuti in una tabella relativa ad una versione non ancora rilasciata, significa che le relative informazioni nella tabella sono esclusivamente in anteprima (work in progress) e non sono utilizzabili fino al rilascio ufficiale, poiché non funzionanti e/o suscettibili di variazioni, pertanto sono potenzialmente dannose.

Passepartout declina ogni responsabilità in caso di utilizzo non autorizzato.

VERSIONE 2024C (v86.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Tasto F12 non attivo dopo esecuzione COLLAGE con CREATELIST e tasti personalizzati (Sol. 59544).	
Nuova variabile vettore <u>_WSQRYSTRA\$()</u> (Sol. 73789).	
Nuova variabile vettore <u>_MMCAURIGA()</u> e relative <u>mmcsr()</u> , <u>occsr()</u> , <u>ofcsr()</u> , <u>omcsr()</u> , <u>opcsr()</u> e <u>orcsr()</u> . Le variabili dizionario <u>occar()</u> , <u>ofcar()</u> , <u>omcar()</u> , <u>opcar()</u> e <u>orcar()</u> assumono la nuova descrizione "Numero causale/contropartita".	L'introduzione della nuova variabile modifica la finestra di emissione/revisione documenti a cui è associato il valore: prima del rilascio della presente versione era presente un unico campo con etichetta MDCAUCONTR, mentre ora ve ne sono due distinti. Nel caso che la tabella delle contropartite sia attiva è necessario procedere alla revisione di eventuali procedure collage che ne fanno uso in quanto dopo l'aggiornamento MDCAUCONTR conterrà il dato della causale di riga mentre la contropartita risiederà nel campo con etichetta MDCONTROPT.
Nuova variabile vettore <u>_LSTRARDES\$()</u> .	

VERSIONE 2024B2 (v86.1b GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
La modifica tramite struttura <u>_FTE</u> dell'xml della fattura elettronica puo' portare alla perdita di sezioni dell'xml (Sol. 73832).	

VERSIONE 2024B (v86.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Editor PassBuilder: estesa la capacità di ricerca/sostituzione testo agli elementi app "Collage controllo App" e "Collage controllo App Mobile".	
AVANZAPF: aggiunta possibilità di impostare il magazzino di carico dei prodotti valorizzando la variabile <u>APFMAG()</u> .	
Abilitato Docuvision nei terminali factory per le tipologie ammesse da questo tipo di terminale.	
CREATELIST: mancato ordinamento righe per colonne di tipo (I)cona (Sol. 73307).	
Sprix Mobile/GETSIGN: impossibilita' di firmare documenti di magazzino (Sol. 73390).	
GETCONDDOC tronca la variabile degli sconti a 7 caratteri (Sol. 73421).	

VERSIONE 2024A5 (v86.0e GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Anomalia variabili dizionario (Sol. 73346).	In seguito all'inserimento posizionale non corretto di nuove voci di dizionario, l'accesso diretto alle variabili arstc e arpwm può ritornare errore o dati non coerenti con la variabile interrogata. Per correggere l'anomalia sarà necessario ricompilare le procedure che ne fanno uso dopo aver installato il presente aggiornamento.

VERSIONE 2024A4 (v86.0d GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Anomalia su ricerca anagrafiche mydb in anagrafica collegata (Sol. 73236).	
GETMULTILOT: la ricerca eseguita utilizzando il tipo lotto non ritorna dati anche se presenti (Sol. 73243).	

VERSIONE 2024A1 (v86.0a GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Errore editor sprix su ciclo tabellare "ol" (Sol. 73015).	
Riorganizzazione MYDB: eliminazione righe multiple associate alla stessa chiave per gli archivi estensione dei documenti (Sol. 71690).	

VERSIONE 2024A (v86.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Nuova variabile _ILPOSDBP() e corrispondente dizionario oipdb.	
PUTGM segnala carattere errato nel codice (Sol. 72635).	
Gestione storico archivio lotti.	Per maggiori informazioni si rimanda a queste note .
Nuova variabile _ARPESWMS\$ e relativa arpwm di dizionario.	
Nuove variabili dizionario per sconto 130 omaggio con rivalsa iva (Sol. 67826): mmrdz, mmruz, ocrdz, ocruz, ofrdz, ofruz, omrdz, omruz, oprdz, opruz, orrdz, orruz.	
Editor Sprix: modificata la struttura della finestra dizionario con l'introduzione della visualizzazione come finestra elenco. La modalità è attivata di default su tutte le installazioni (CTRL+SHIFT+F4 per ripristinare la visualizzazione precedente).	
Nuova istruzione PUTLSTODC_EXT e nuovo parametro facoltativo per DELLSTDOC.	

VERSIONE 2023H (v85.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
GETATTPN non valorizza le variabili PROPIETA' DOCUMENTO (Sol. 71789).	
Nuova variabile _ARDTIMG\$ e relativa ardim() (Sol. 72145).	
Introduzione archivio storico bolle di lavorazione. I dati dello storico sono accessibili in lettura tramite GETBL e le tabelle dizionario ov e oz (Sol. 72061).	
Tipo documento collegato precedentemente errato (deve essere OF) (Sol. 72352).	
Variabile _MMIDLISTA(R) rilasciata in sola lettura (Sol. 71869).	
Storicizzazione automatica anagrafica unica (PUTZA, parametro "STOR=R"). Aggiunte nuove variabili: _ZA_TPSOG\$ ora collegata a zazzses(), _ZA_DESC\$ ora collegata a zanom(), _ZADIV\$, _ZA_ADAI\$, _ZA_RDAI\$, _ZA_IDAI\$, _ZA_EDAI\$, _ZA_LDAI\$, _ZA_TDAI\$, _ZA_FDAI\$, _ZA_GDAI\$ e _ZA_NDAI\$.	
Visualizzazione scheda articolo campionario webdesk copre accesso a tasti funzione (Sol. 71942).	
Accesso diretto tramite variabili dizionario ai dati della tabella pr.	
Editor sprix: facilitazione inserimento decodifiche per i campi dizionario.	

VERSIONE 2023G3 (v85.0c GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Leggendo e riscrivendo in sprix un OC o superiore che deriva da preventivo, i riferimenti al preventivo presenti nelle righe descrittive/testo vengono persi.	

VERSIONE 2023G1 (v85.0a GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Anomalia in accesso anagrafica articolo da campo di INPUT sprix (Sol. 71880).	

VERSIONE 2023G (v85.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Valorizzazione variabile <code>_MMNCRERIGA()</code> nell'evento <code>ON_STP_DOC</code> (Sol. 71181).	
Nuova variabile <code>_MMSCMERCE(T)</code> (Sol. 52451).	
PUTMM non segnala errore ma documento non presente in lista (Sol. 71506).	
Nuova variabile <code>_ARRCM\$</code> e rispettiva <code>arrcm()</code> .	
AVANZAPF: implementazione campo vettore maggior quantità prodotta per articoli con taglie (Sol.71387).	
Nuova istruzione <code>LPRINTQRCODE</code> .	
PROGARTMAG: errore violazione di memoria (Sol.71470).	
Scrittura <code>DBVO</code> in trasformazione nativa documenti (Sol. 71541).	
Errore 6 generato con operazioni su lotti di articoli strutturati (Sol. 71741).	
Nuove variabili <code>_MMTDOCP\$(R)</code> , <code>_MMSOFP(R)</code> e <code>_MMNOFP(R)</code> e relative <code>mmsdc()</code> , <code>mmsec()</code> , <code>mmndc()</code> , <code>ocsdc()</code> , <code>ocsec()</code> , <code>ocndc()</code> , <code>orsdc()</code> , <code>orsec()</code> e <code>orndc()</code> .	
Nuova variabile <code>_MMR_PSMER\$(9)</code> e implementazione comportamento <code>_MMSCO\$(R)</code> per gestire maggiorazioni di riga.	
Nuove funzioni <code>GETATTTOTMM</code> e <code>GETATTRIGMM</code> per il calcolo dei totali documento e di riga da collage.	
Anagrafica unica: variabili struttura <code>_ZA</code> e relative istruzioni <code>GETZA</code> , <code>PUTZA</code> , <code>DELZA</code> , <code>AZZVARSYS 36</code> , <code>AZZCATVARSYS "ZA"</code> e variabili <code>_ERRZA</code> e <code>_ERRZA\$</code> . Ciclo tabellare interessato: "za", rinominato da rubrica aziendale a anagrafica unica e contestualmente esteso con le nuove variabili, e "zs" storico anagrafica unica.	

VERSIONE 2023F4 (v84.5d GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
SPRIGBL: la variabile stati di riga non può essere espressa come variabile stringa quando sono presenti quantità per 32 taglie (Sol. 71385).	
PUTLOT: errata attribuzione nuovo codice lotto in fase di aggiornamento (Sol. 71429).	

VERSIONE 2023F3 (v84.5c GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabile <code>ampde</code> non associata a campo univoco (Sol. 71368).	

VERSIONE 2023F1 (V 84.5A GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Nuova variabile <code>_PNCTF\$()</code> : Riferimento al conto di costo/ricavo per dichiarazioni iva.	

VERSIONE 2023F (V 84.5 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
CSVROWVAL causa chiusura del client senza messaggi di errore (Sol. 70892).	
Errore filtro veloce in elenco da istruzione CREATELIST (Sol. 71043).	

VERSIONE 2023E2 (V 84.4B GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Nuova variabile flag osert(), esclusione dal rilevamento per articoli di tipo "L", in sviluppo distinta base.	
Errore nelle stampe da terminale non grafico e server linux (sol. 70886).	

VERSIONE 2023E (V 84.4 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Articolo DBA in lista di prelievo da SPRIX (Sol. 70279).	
Nuova variabile dizionario arstc(), categoria statistica completa della parte alfabetica e numerica (Sol. 70188).	
Nuove variabili _MMNOBV\$, _MMNOADDBV\$ e relative variabili dizionario mmblv e mmabv (Sol. 64748). Le variabili sono state rese disponibili anche negli export DBMIRROR e ASCMIRROR della tabella MM_TESTATE (campi mmblv, mmabv e IFL_NODOCBV, IFL_NOADDBV).	
Anomalia GETCONDDOC nel calcolo degli sconti e delle provvigioni (Sol. 70636).	
Anomalia su anagrafica MYDB estensione testata documenti (Sol. 70639).	

VERSIONE 2023D2 (V 84.3B GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Lavorazione da rilevare caratteri errati (Sol. 70502).	In seguito a questa anomalia vengono documentate le variabili _ILLAVDARIL\$() e _ILLAVRILEV\$().

VERSIONE 2023D1 (V 84.3A GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Utilizzando la funzione ASCMIRROR_EXT filtrando tramite i parametri <slot> o <att> o <var>, i documenti con più testate (cioè documenti che derivano da più di un documento come ad esempio FT che deriva da più OC) venivano comunque considerati. Questo provocava la ripetuta sincronizzazione verso Passweb di tali documenti anche se in realtà non erano variati (Sol. 70428).	

VERSIONE 2023D (V 84.3 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
PUTPN/GETPN in doppia valuta e nuove variabili _PNSIV(), _PNPRD e _PNMPD().	Aziende con scadenzario a partite: la variabile _PNSIP(), importo da pagare/compensare, deve essere valorizzata esclusivamente nella valuta del documento. Compensazioni di rate negative e pagamenti, relative allo stesso conto Cli/For, non sono piu' ammessi all'interno dello

	stesso documento se questo e' in valuta diversa da quella di gestione (e' necessario creare piu' documenti distinti).
CREATELIST: ordinamento decrescente ordina per colonna diversa da quella impostata nella variabile CTNUMCOLORD (Sol. 70230).	
Ciclo tabellare movimenti di magazzino va in errore con messaggio mmag record non valido (Sol. 70270).	

VERSIONE 2023C1 (V 84.2A GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
La funzione AVANZAPF accetta i decimali nel parametro delle quantita' di sovrapproduzione (Sol. 70038).	
MYDB estensione testate documenti: perdita modifica note al secondo accesso in anagrafica in fase di trasformazione (Sol. 70083).	

VERSIONE 2023C (V 84.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
datnrg() data non valida 0/0/0 (Sol. 69485).	
Anomalia stampa anagrafiche mydb (Sol. 68265).	
Anomalia ciclo tabellare 'or' (Sol. 69951).	

VERSIONE 2023B5 (V 84.1E GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Introdotti nuovi parametri per la funzione LEGGIDESC e variabili dizionario mmttt, mmtar, octtt, octar, ofttt, oftar, omitt, omtar, opttt, optar, orttt, ortar. Il ciclo tabellare sulle tabelle mm, oc, of, om, op e or ora ritornano il numero di creazione, nelle rispettive variabili, per le righe documento di tipo descrizione e testo. In mancanza di ordinamento specificato, il ciclo tabellare mm ritorna di default con l'ordinamento per chiave mmtsi()+mmtse()+mmtnu()+mmcli()+mmprg()+mmord().	

VERSIONE 2023B4 (V 84.1D GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
SS_DOCREV semaforo di tipo stringa: numero di stringhe non valido (Sol. 69627).	
CREATELIST di tipo 1, tasto conferma assente (Sol. 69729).	
Nuova variabile _MMINMAGAZ\$.	
Nuova variabile _LSTRINM\$().	

VERSIONE 2023B1 (V 84.1A GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabile mmspl Split Payment non ritorna il dato corretto (Sol. 69483).	

VERSIONE 2023B (V 84.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Nuova variabile dizionario amstt (PROPONI STATO RIGA IN TRASFORMAZIONE DA OF).	
Nuova variabile dizionario pnir (progressivo numerico univoco RATEO/RISCONTI).	
CDG: abilitata lettura e scrittura DOCUVISION da sprix.	

Gestione di 999 documenti nella GETMM e GETMM2 e GETMM_EXT.	
Anomalia Estensione MYDB su testata documenti compilati nella fase di trasformazione documento (PR->OC) (Sol. 69259).	
Anomalia valorizzazione _MMNCRERIGA() (Sol. 69241).	
Istruzione PUTCOMM obbliga inserimento date anche se flag nei parametri risulta spento (Sol. 68812).	
MYDB: errore violazione memoria a seguito di export Excel (Sol. 69265).	

VERSIONE 2023A2 (V 84.0B GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
MYDB: errore violazione memoria a seguito di export Excel (Sol. 69209).	
MYDB: duplicazione anagrafica articolo con MYDB inserito non duplica dati MYDB (Sol. 69211).	

VERSIONE 2023A1 (V 84.0A GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Corretta anomalia istruzione MYDBPUTATT (Sol. 69022).	
Corretta anomalia CREATELIST modalità 1 – tasto di conferma disabilitato (Sol. 69095).	
Corretta anomalia l'evento SCF_RIDISEGNA_CANCEL_SELEZ non cancella le righe selezionate (Sol. 69109).	
Corretta anomalia GETCONDDOC - non restituisce la corretta provvigione in caso di particolarità basata su sconti (Sol. 69128).	

VERSIONE 2023A (V 84.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Nuova variabile _ARMAC\$ e corrispondente voce dizionario armac.	
Abitato variabile _MMPICKING\$ su documenti OF per gestione campo di testata "Stato".	
App attive: variabili di struttura _APPSTS e relative istruzioni ELENCASTATOAPP, AZZVARSYS 76 e AZZCATVARSYS "APPSTS".	
Nuova variabile _OLUM\$, descrizione unità di misura del prodotto finito.	
Corretta anomalia GETATTAR (Sol. 67924).	
Corretta anomalia _CTTOTRIGSEL ritorna valore errato (Sol. 68820).	
Nuovi parametri PUTMM per scrittura CGD e relative variabili _MMDTANL\$, _MMCOMMANL\$ e _MMAREAANL\$.	
Nuovi parametri DELMM_EXT per cancellazione movimento bene usato e CGD.	
Nuove variabili metadati ubicazioni: _PUTDATCRE\$, _PUTORACRE\$, _PUTUTECRE\$, _PUTMENCRE\$, _PUTDATMOD\$, _PUTORAMOD\$, _PUTUTEMOD\$, _PUTMENMOD\$ e relative variabili dizionario pucds, pucos, pucid, pucin, pucme, pumet, pumeo, puvid, puvin e puvme.	

VERSIONE 2022J3 (V 83.1C GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Abitato funzione AVANZAPF in collage server remoto.	
Anomalia: _ILL_IDLOT(L)=-2 genera errore non correggibile (Sol. 68652).	
Anomalia: Errore violazione di memoria in esecuzione comando PUTLSTDOC se la variabile _LSTRIDLOT contiene un ID lotto non esistente (Sol. 68720).	

VERSIONE 2022J (V 83.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Nuova variabile _ILCCFAS() e relativa variabile dizionario "oiccf".	
Attivato eventi collage ON_IN_* e ON_OUT_* su campi finestra anagrafica lotti quando richiamata da record MYDB (Sol. 68034).	
Implementato la funzione di esportazione/importazione da editor Passbuilder per gli elementi collage server	

remoto e collage controllo app. (Sol. 59803).	
Implementato campo data ultima modifica a fianco di ogni elemento delle app Passbuilder (Sol. 63961).	
Implementato nuovo parametro progressivo scheda "PRGSCH=(1..99)" per funzione PUTASSOCDV per gestire il dato "Progressivo scheda articoli".	Accetta valori da 1 a 99 e richiede TPALL=S
MYDB: archivio Mydb con campo lotto, tentando di entrare nell'anagrafica del lotto errore lotto non trovato (Sol. 68061).	
Anomalia MYDBVIEW - problema con filtri di selezione (Sol. 68336)	
Nuova variabile _PCASIXML\$ e relativa variabile dizionario pcasx.	
Nuova variabile _PNESO\$ e relativa variabile dizionario pnios.	
Nuove variabili _ARDTCUL\$ e _ARDTCUP\$ e relative variabili dizionario ardcu e ardcp.	
Reso possibile l'aggiornamento degli ordini in consegna da sprix con archivi storici attivi.	
Nuove variabili metadati struttura _LST e relative variabili di dizionario.	

VERSIONE 202216 (V 83.0F GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Corretta anomalia "tpdes": Il file della tabella delle descizioni di fasi è chiuso (Sol. 68073).	
Corretta "anomalia collegata alla soluzione 68054 corretta nella 202214 si ripresenta nella versione 202215" (Sol. 68243).	

VERSIONE 202215 (V 83.0E GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Corretta anomalia errore in scrittura impegni (Sol. 68094).	
Corretta anomalia descrizione fase _TPDES\$ errata (Sol. 68120).	

VERSIONE 202214 (V 83.0D GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Corretta anomalia causata da errato posizionamento della tabella dizionario "gd" (Sol. 68054). L'inserimento della tabella "gd", a partire dalla versione 20221, ha invalidato le seguenti tabelle: "ta", "te", "tf", "tb", "ti", "tz", "tt" e "tr". Inoltre il posizionamento del campo "tzdes" nella versione 2022D, ha invalidato i campi della tabella "tz" a partire dalla variabile "tzprn". Gli eseguibili sprix ricompilati a partire da queste versioni e che includono l'utilizzo delle tabelle e variabili indicate, vanno ricompilati a partire da questa versione per ripristinarne il comportamento standard. Segnali evidenti che la procedura necessita di essere ricompilata includono i messaggi d'errore "Voce di dizionario non ammessa" e "File errato o non aperto".	

VERSIONE 20221 (V 83.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Istruzione DELCFR resa disponibile in collage server remoto (Sol. 67502).	
Aggiunto variabile ID lotto a struttura _PAM (_PAMIDLOTTO).	
Nuove variabili _MMTOTVALID\$, _MMTOTIVA(), _MMTOTDIOC(), _MMTDAPAGARE() e relative variabili dizionario mmtdv, mmtiv, mmtdo, mmtdp, octdv, octiv, octdo, octdp, oftdv, oftiv, oftdo, oftdp, omtdv, omtiv, omtdo, .omtdp, optdv, optiv, optdo, optdp, ortdv, ortiv, ortdo, ortdp.	Sui documenti inseriti prima della versione 20221 i totali non sono immediatamente disponibili a meno di aver fatto girare la funzione 'Servizi > Variazioni > Magazzino > Calcola totali documenti (Alt+ZVMM)' in alternativa è sufficiente leggere (GETMM) e riscrivere (PUTMM) il documento. La variabile MMTOTVALID\$ (S/N) indica se i totali sono valorizzati per la testata.
Nuovi parametri facoltativi nella funzione SPRIGBL.	
Nuovi parametri facoltativi nella funzione SVILDB.	

Avanzamenti di produzione: variabili di struttura _APF e relative istruzioni AVANZAPF, AZZVARSYS 73, AZZCATVARSYS "APF" e variabili _ERRAPF, _ERRAPF\$, _NLOTTIAPF e _NAPF.	
La lettura dati della tabella 'nu' può ritornare valori non corretti (Sol. 67653).	
Condizioni documento: variabili di struttura _PZD e relative istruzioni GETCONDDOC, AZZVARSYS 58, AZZCATVARSYS "PZD" e variabili _ERRPZD e _ERRPZD\$.	
Nuove variabili struttura _MM per trasformazione nativa documenti: _MMSIGTRASS\$, _MMCOLR(R), _MMQTAR(R), _MMTAGR(R,T), _MMLQTAR(R,L), _MMLQTATGR(R,L,T), _MML_COLLI(R,L), _MML_LCOLLIR(L), _MML_LQTAR(L), _MML_LQTATGR(L,T), _MMUBISURES\$(R).	
Nuova variabile _PCBARICO e relativa variabile dizionario pcbri.	
Nuove variabili dizionario aqesi e aqdin.	
Anomalia collage, etichetta ON_STARTST_WIBCESP: (Sol. 67791).	
Corretta anomalia: PUTMM consente scrittura documento anche se un documento con la stessa chiave è presente nell'archivio documenti storici (Sol. 66345).	
Nuove variabili _TPCICLO e _TPCOLLEGA\$	

VERSIONE 2022H2 (V 82.5B GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Anomalia CREATELIST che si presenta nelle liste a selezione multipla: la variabile _CTTOTRIGSEL ritorna 0 su generico evento di callback quando ci sono righe selezionate (Sol. 67537).	
Anomalia FTE modifica campi vettore 2.1.9.1.3.1 e 2.1.9.1.3.5 (Sol. 67600).	

VERSIONE 2022H (V 82.5 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Collage: attivato eventi ON_IN_MALISTIN e ON_OUT_MALISTIN in anagrafica articoli->condizioni commerciali->listini (Sol. 67195).	
MYDB: corretto errore non correggibile in stampa personalizzata lanciata da MYDBVIEW richiamata da procedura collage (Sol. 67077).	
DBP, allineato funzioni sprix al comportamento del gestionale: eliminata la possibilità di cancellare tutte le righe della DBP senza cancellare la distinta base vuota.	
Struttura _LST: nuova variabile _LSTRRESSTR\$() e relativa variabile dizionario lrrst.	
Dizionario: corretta mancata lettura diretta cartella abbinamenti in caso di abbinato articolo (Sol. 67110).	
MyDB: campo in anagrafica collegato ad archivio pagamenti non riporta valore corretto con codice pagamento maggiore di 255 (Sol.67400).	
CREATELIST: anomalia su eventi di callback (Sol. 67468).	

VERSIONE 2022G (V 82.4 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Nuove variabili _ARSMN\$ e arsmn per Gestione RSM: Non rimborsabile TM1 e TM2	
Progressivi articoli: variabili di struttura _PAM e relative istruzioni PROGARTMAG, AZZVARSYS 74, AZZCATVARSYS "PAM" e variabili _ERRPAM e _ERRPAM\$	
Liste di prelievo: variabili di struttura _LST e relative istruzioni GETLSTDOC, PUTLSTDOC, DELLSTDOC, AZZVARSYS 75, AZZCATVARSYS "LST" e variabili _ERRLST e _ERRLST\$. Ciclotabellare interessato: "ls", "lr". Aggiuto variabile di magazzino di riga associato ad ID lista: _MMIDLISTA(R).	
Nuova istruzione NOMEPASSAPP\$() che ritorna il nome completo della PassApp.	

VERSIONE 2022F (V 82.3 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
CREATELIST: I campi numerici delle liste vengono stampati come tali nelle stampe excel formato xml.	Richiesta numero 66169.
Nuovo flag disponibile in proposta evadibilità ordini (variabili _PCIEV\$ e pcoie): utilizzare "P" per Picking.	
Nuova variabile flag _MMPICKING\$ (accetta valori S/N)	

VERSIONE 2022E (V 82.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Movimenti di magazzino: <code>_MMTIPBOEDI\$, mmtbo, _MMT_TBOEDI\$</code> .	
Nuovi eventi collage <code>ON_IN_CIDRIGA</code> e <code>ON_OUT_CIDRIGA</code> .	Richiesta numero 15135.

VERSIONE 2022D (V 82.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Piano dei conti: <code>_PCRPAE\$, _PCRPI\$, _PCRDEN\$, _PCRCOG\$, _PCRNOM\$, _PCRCAP\$, _PCRIND\$, _PCRCVI\$, _PCRLOC\$, _PCRPRO\$, _PCRNAZ\$, pafst, pafpi, pafde, pafco, pafno, pasca, pasin, pasci, paslo, paspr, passt</code> .	
Aggiunto la possibilità di impostare nella scrittura dei movimenti il default relativo alle ubicazioni articolo, valorizzando con -1 le variabili <code>_MMRUBI(R)</code> e <code>_MMRAUBI(R)</code> .	
Movimenti di magazzino: <code>_MMDESAGGVU\$(R)</code> , <code>mmdvu</code> , <code>ocdvu</code> , <code>ofdvu</code> , <code>omdvu</code> , <code>opdvu</code> , <code>ordvu</code> .	
Particolarità articoli/clienti: <code>_PARTDES\$</code> e <code>tzdes</code> .	
<code>_MMRIFTPA\$</code> e <code>_MMRIFDATA\$</code> ora disponibili per documenti FF e NF.	
Aggiunto all'elenco delle variabili dizionario disponibili quelle relative alla porta di connessione per Docuvision (<code>sxpdv</code>) e MexalDB (<code>sxpdb</code>).	
Anagrafica ubicazioni: <code>_PUPERCORSO</code> e <code>puper</code>	

VERSIONE 2022C (V 82.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Distinta base primaria: <code>_TPFOL\$</code> , <code>_TPMAL</code> , <code>tpfol</code> , <code>tpmal</code> .	
Aggiunta l'Applicazione Sprix n.W0 - MyDB CRM (Az. SPX) nella cartella "Esempi di App in Sprix e Collage". L'esempio dispone del relativo video corso in Area Riservata Sviluppatori; i sorgenti (Sprix, Collage, MyDB e Moduli) possono richiedere aggiornamenti.	

VERSIONE 2022B4 (V 81.9D GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Intracce 2022: <code>_PNNTB\$, _PNPOR\$, _MMNBTR\$, _MMPAEM\$, _MMTOCOS\$, mminb, mmpmi, mmtoc</code>	
Aggiornamento della cartella "Esempi di App in Sprix e Collage" scaricabile dal sito Passepartout.net al link "Allegati e Documenti" presente alla voce di menu "Area Sviluppatori" dell'Area Riservata.	

VERSIONE 2022A (V 81.8 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Modifica alle voci di menu: sono eliminate le voci 'Stampe' e 'Dichiarativi'; la voce 'Annuali' precede la nuova voce 'Fiscali' e che contiene gli Sprix all'origine configurati nella vecchia voce 'Stampe'. Gli Sprix configurati nelle altre voci di menu, vengono collocati nelle medesime voci; l'impostazione dei menù di esecuzione Sprix in configurazione ed i relativi collage restano inalterati e garantiscono la retrocompatibilità. La licenza MDS consente di creare un'App in cui importare i soli eseguibili tramite funzione [Sh+F4] e sono riconfigurabili con la funzione [F7] "Configura" per cambiare la voce di menu.	
Dizionario: allineamento cicli tabellari alla struttura: <code>o?ntt (_MMTAGTAB)</code> <code>o?tir (_MMTPR\$)</code> <code>o?umi (_MMTIP\$)</code> <code>o?str (_MMORD\$)</code> <code>mmtir (_MMTPR\$)</code> <code>mmumi (_MMTIP\$)</code>	

VERSIONE 2021J (V 81.7 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;

- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabile <code>_IL_INIZLOT</code> consente di gestire lo start dell'array che contiene l'elemento del lotto	

VERSIONE 2021I (V 81.6 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
PUTBL: da questa versione le righe movimentate in conto lavoro non saranno modificabili	
<code>_TP?</code> : estensione delle note della fase e dei componenti a 1000 caratteri (25 elementi x 40 caratteri)	
<code>_ILL_?</code> : variabili di gestione Lotti e Matricole negli Impegni di Lavorazione	
Sprx Mobile: Istruzioni DELLOC e LSTLOC, variabili <code>_WL?:_WLICOFGL\$()</code> , <code>_WLNUMROW</code> , <code>_WLCARTADD\$</code> , <code>_WLHIDEFLD\$(n)</code> ; variabili <code>_WI?:_WICLEARFLD\$</code> . Per approfondimenti si rimanda al Manuale Sprx Mobile.	
Aggiornamento della cartella "Esempi di App in Sprx e Collage" scaricabile dal sito Passepartout.net al link "Allegati e Documenti" presente alla voce di menu "Area Sviluppatori" dell'Area Riservata.	

VERSIONE 2021H (V 81.5 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabile <code>_ILIDUBI</code> Id ubicazione impegni di produzione; viene tornata solo con la gestione della logistica	
GETATTCOMM istruzione che legge i dati di Anagrafica Commessa presente a video	
Anagrafica Ubicazioni: variabili di struttura <code>_PU</code> e relative istruzioni GETPU, GETATTPU, PUTPU, DELPU, AZZVARSYS 72, AZZCATVARSYS "PU" e variabili <code>_ERRPU</code> e <code>_ERRPU\$</code> . Ciclotabellare interessato: "pu", "pq", "pg", "po". Altro Dizionario interessato: "ab", "mm", "oc", "of", "om", "op", "or".	
Variabili struttura <code>_AR</code> per Ubicazioni nei "DATI MAGAZZINO" e Variabili di struttura di riga <code>_MM</code> per Ubicazioni	
Le INPUT vengono dotate di tipo dato "U" per l'utilizzo dei codici ubicazione.	
Istruzione CTRARTTOUBI per i controlli di inserimento dell'ubicazione sull'articolo.	
GETATTMM in avanzamento di produzione (sostituisce il workaround SHELLSPRIX + GETMM/Ciclotabellare)	

VERSIONE 2021G (V 81.4 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabili primanota per aziende RSM: <code>_PNDPDS\$</code> "esclusione da distinta" e <code>_PNTPM(200)</code> "num. tipo merce"	
Variabili numeriche <code>_ARSME</code> e <code>_ARSMR</code> e di riga documento <code>_MMTIPM</code> 'Tipo merce' per aziende RSM	

VERSIONE 2021F (V 81.3 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabile <code>_PNPAEO\$</code> codice paese gestione IVA OSS	

VERSIONE 2021E (V 81.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Stampa del sorgente dal compilatore [F10]: aggiunto flag per numerare le righe della stampa (il default per quella che prevede l'elenco variabili è "S")	

VERSIONE 2021D (V 81.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Collage su dispositivo di stampa Email (SMTP) con la vecchia interfaccia utente non è più disponibile. La gestione in Collage del suddetto dispositivo richiede un aggiornamento del sorgente in funzione della nuova interfaccia "Mail Editor" (dalla v810), tramite i seguenti specifici comandi: <ul style="list-style-type: none"> • Evento ON_STARTSTPEMAIL: etichetta all'apertura della stampante Email; • Istruzioni per leggere e scrivere le variabili di struttura dati _EMAIL??? della stampante di tipo Email da inviare al motore di stampa: EMAILGETATT e EMAILPUTATT • Variabili struttura _EMAIL??? 	
Variabili _MM e _MMT per la gestione del Bonus Edilizio (se attivata la gestione da parametri di magazzino)	
Variabili _DVVRELEASE\$, _DVVPRODOTT\$ e _DVVORIGINE\$ dei file gestiti a documento o pagine	

VERSIONE 2021C3 (V 81.0C GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Istruzione APRISTAMP_EMAIL_EXT per stampare su stampante email e tra i vari parametri gli si può passare il percorso di un file HTML che diventerà (quindi sostituirà) il corpo dell'email.	Richiesta n. 51437

VERSIONE 2021C1 (V 81.0A GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Tabella Dizionario 'sw' con informazioni relative ai siti PASSWEB di anagrafica azienda	

VERSIONE 2021C (V 81.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
_PCRIMIO\$ e _PCRIMIS\$ - variabili conti automatici dipo "R" nel Piano Conti 80, mastri 204, 205 e 206;	
_ARDSE\$() - array di 5 elementi da 200 caratteri ciascuno per descrizioni dettagliate articolo (1000 caratteri);	
_MMDESAGG\$(R,1-5) array di riga (5 elementi da 200 caratteri) per descrizioni dettagliate =< 1000 caratteri; in caso di variazione della descrizione dettagliata il dizionario mmdse(7)() riporta 7 elementi da 145 caratteri;	
CALLWEBSVC - metodo PATCH per la _WSMETHOD\$;	
Per i protocolli https, imap, smtp e altre comunicazioni sopra al TLS, la GETINTERNET cerca di impostare automaticamente il Server di destinazione come SNI, in modo da coprire la maggior parte dei casi;	Richiesta n. 56942

VERSIONE 2021B (V 80.6 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabili per riferimenti per Resi (RE) e Annulli (AN): _MMCALTRIF\$ e _MMDALTRIF\$	
PUTPN può compensare rate negative di scadenzario <u>non abbinare ad alcuna riga positiva.</u>	

VERSIONE 2021A (V 80.5 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabili di testata Magazzino e nuove colonne MexalDB _MMTANNOLET(), _MMTPROGLET(), _MMSPTRASP(), _MMTCAUVEN()	

Aggiornamento _DVVESTFIL\$(): array in sola lettura con le estensioni dei file di presa visione allegati al documento. L'indice dell'array è il numero di versione del documento	
Sprix Mobile <ul style="list-style-type: none"> Implementata la possibilità di settare il focus su un determinato Input quando si visualizza un form. Implementato nuovo elemento PassBuilder "Collage Controllo App Mobile"; consente di utilizzare etichette per gestire eventi di: <ul style="list-style-type: none"> ON_NUOVA_APP: ogni volta che è installata una app sul dispositivo mobile; ON_MODIFICA_APP: ogni volta che l'app è modificata in PassBuilder e poi aggiornata sul dispositivo mobile; ON_POST_SYNC_APP: ogni volta che la sincronizzazione con il server produce risultati o errori; Istruzione GETSIGN per la gestione della firma di presa visione relativa ad un dato documento. La firma verrà allegata al documento e, se necessario, sincronizzata lato gestionale con Docuvision. <p>Per approfondimenti si rimanda al manuale Sprix Mobile (v 3.5).</p>	
Librerie componenti SHAKER 7.7 .NET e JAVA - consultare il paragrafo NOVITA' IMPORTANTI	
Abilitazione istruzione GETPC2 in Shaker	

VERSIONE 2020M (V 80.4 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabili _PCLLOTTE\$ (diz. paltt, pcltt) e _MMCLOTTRT\$ (diz- mmclt) per gestire il Codice Lotteria in MxRT.	

VERSIONE 2020L5 (V 80.3E GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Etichetta MDETDPORTO del campo descrizione aggiuntiva del porto in presenza di diverse testate	Sol. 60672

VERSIONE 2020L (V 80.3 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
GETATTIND istruzione Collage per leggere i dati anagrafici di Indirizzi di Spedizione presenti a video	Richiesta n. 23921
Variabili per ricostruire il numero documento dal Tag <Numero> del Xml: PCXMLTPNR\$, _PCXMLPOSNR()	
Funzione "Sostituzione testo" [Ctrl+F6] nella funzione "Cerca/Sostituisci" [F11] dell'Editor Sprix	
Funzione DOCMODCONT per conoscere lo stato di aggiornamento contabile del documento in revisione	
Funzione JSONADDNULL per settare il valore NULL su un oggetto JSON	
Variabile _AUNUMLIS\$ di GETAUTORIZUTE che ritorna la stringa dei listini da visualizzare	

VERSIONE 2020K5 (V 80.2E GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Evento ON_DELBL può eseguire lettura tramite GETATTBL	

VERSIONE 2020K4 (V 80.2D GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabile contabile sullo stato dell'import XML: _PCXMLSTIMP\$	

VERSIONE 2020J3 (V 80.1C GEST)

Notare:

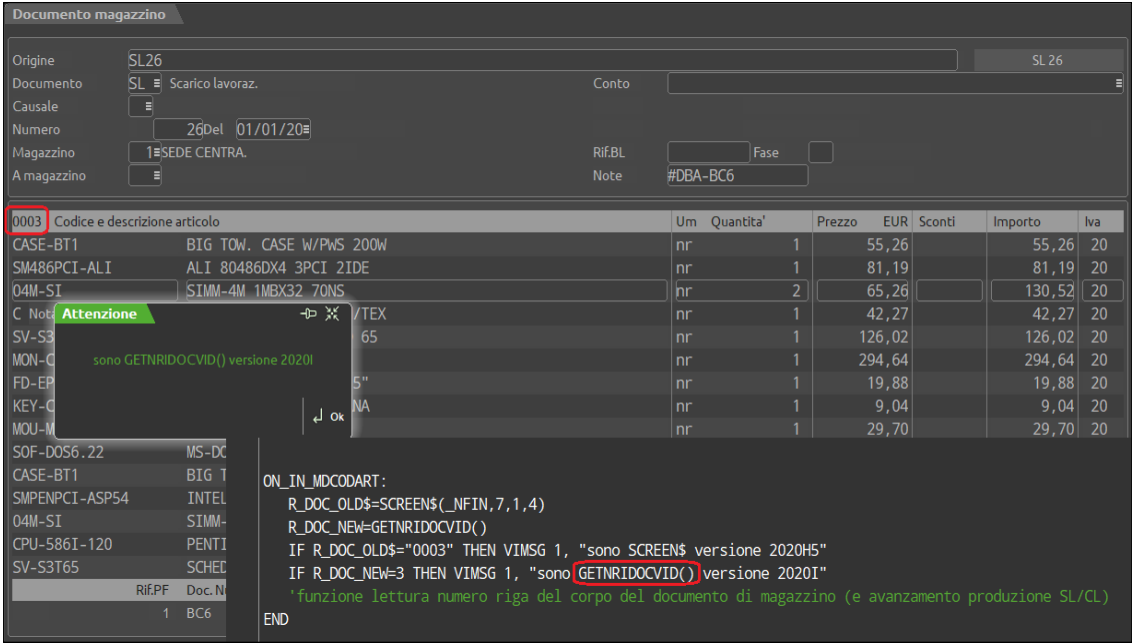
- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabile <code>_CTTASTFUNES</code> attivare i tasti "501" e/o "505" in <code>CREATELIST "0"</code>	Sol. 59545
Sprix Mobile: <ul style="list-style-type: none"> variabili <code>_WL???</code> attributi Icona Gestione dei dati GPS (WCALL "SENDGPSDATA" e "GETGPSDATA", evento <code>ON_GETGPSDATA</code>: variabili <code>_GPS</code>, istruzione Collage Server Remoto <code>GETGPSDATA</code> variabile <code>_WIMAXCAR</code> quantità massima di caratteri in input Istruzione <code>WPRTFORMOPT <id_form></code> in sostituzione di <code>WCALL "PRINTFORM"</code> Input "CHECKBOX" in <code>:_WITYPE\$</code> e relativo evento <code>ON_CHECK</code> alla pressione del checkbox 	

VERSIONE 2020J (V 80.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
<p>Funzioni Collage che rilasciano il progressivo di riga in cui l'utente è posizionato nel corpo della registrazione. Il numero progressivo di riga è quello visualizzato a sinistra nella barra dei titoli delle colonne della sezione relativa al corpo della registrazione.</p> <p>GETNRIPRNVID() - numero riga a video del corpo di primanota GETNRIDOCVID() - numero riga a video del corpo documento magazzino e avanzamento produzione SL/CL GETNRIBLVID() - numero riga a video del corpo della bolla di lavoro</p> <p>Esempio per magazzino</p>  <p>Funzione di lettura del titolo della finestra <code>GETFINTIT\$(<numfin>)</code> funzione di lettura e restituzione del titolo della finestra <numfin> (se = <code>_NFIN</code> è quella attiva)</p>	(Rif. 2020Ia beta)
<p>Note aggiuntive alle funzioni di cui sopra.</p> <p>Tali funzioni hanno lo scopo di sostituirsi all'utilizzo della <code>SCREEN\$</code>, per una progressiva sostituzione della funzione <code>SCREEN\$</code> all'interno delle personalizzazioni a favore delle nuove istruzioni più portabili che verranno implementate.</p> <p>Quanto alla portabilità dei Collage preesistenti: l'adozione della nuova interfaccia non pregiudica le personalizzazioni Collage basate sui nomi dei campi. Nel caso invece in cui il collage utilizzi l'istruzione <code>SCREEN\$</code> ossia la lettura dei valori riportati a video, per mantenere il più possibile la portabilità senza interventi sul sorgente sono disponibili le seguenti eccezioni transitorie e temporanee.</p> <ul style="list-style-type: none"> EM. REV. DOC.: <code>SCREEN\$(_NFIN,7,1,4)</code> viene interpretata come <code>SCREEN\$(_NFIN, 8,1,4)</code> leggendo quanto riportato a video nella riga "8" 	

- IMM./REVIS. PRIMANOTA.: SCREEN\$(_NFIN,5,1,3) viene interpretata come SCREEN\$(_NFIN, 6,1,3) leggendo quanto riportato a video nella riga "6"
- SCREEN\$ utilizzata per leggere il titolo delle finestre (operante solo su alcune finestre) riuscirà a leggere il nome anche dopo il cambio di posizione causato dall'adozione della nuova interfaccia.

Le suddette eccezioni verranno dismesse a partire dalla prima release 2021 del Gestionale. In sostituzione della SCREEN\$ sarà necessario utilizzare le nuove istruzioni che consentono di individuare rispettivamente la riga in cui è posizionato l'operatore nella registrazione oppure il titolo della finestra.

Tabella casistiche di compatibilità Collage

Funzionalità	Utilizzo SCREEN\$ per	Comp. vers. precedente	Alternativa
Em. Rev. Documenti Finestra principale dati testata e corpo	Numero riga visualizzato (posiz. video rig. 7 col. 1)	SI solo per numero di riga visualizzato (temporanea)	Funz. GETNRIDOCVID() ottiene num. posiz. riga
Em. Rev. Documenti Finestra principale dati testata e corpo	Info. righe corpo doc. (videata corpo documento ha una riga in meno)	NO	GETATTMM
Em. Rev. Documenti Finestra principale dati testata e corpo	Lettura valori dell'ultima riga di finestra principale (esistente, disponibile, tot. merce...)	SI - NB: il parametro di config. utente "dimensione finestra emis./rev. Docum: fissa/dinamica"	
Immiss./Revis. Primanota	Numero riga visualizzato (posiz video rig. 5 col. 1)	SI solo per numero di riga visualizzato (temporanea)	Funz. GETNRIPRNVID() ottiene num. posiz. riga
Immiss./Revis. Primanota	Ultima riga (tot. imponibile totale imposta...)	SI	
Immiss./Revis. Primanota	Info. righe corpo doc. (videata corpo documento ha una riga in meno)	NO	
Generale	Riga 1 per finestre in cui la funzione restituisce il titolo (non tutte le finestre hanno questa possibilità)	SI	Funzione GETFINTIT\$() per tutte le finestre rilascia il titolo
Generale	Riga 1 per finestre dove la funzione restituisce il contenuto della prima riga e non il titolo	SI - per tutte le finestre che hanno unica sezione grafica e non è cambiato il layout di info. contenute	

VERSIONE 2020I (V 80.0 GEST) - BETA

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Potenziamento del motore di ricerca nel Editor Sprix con la ricerca globale "cerca tutti" e le "ultime ricerche"	
Istruzioni di messaggio con impostazione tasti e richiesta conferma: VIMSG_EXT e RICCONF	
Istruzione GRIGLIAFIN per impostare la finestra con la griglia di righe "chiaro/scuro" alternate	
Istruzione NUMPAGFIN per impostare il numero di pagina e pagine totali nella barra di navigazione di finestra	rif. v751a
Sezioni finestra: Par. "NOVISEZIONE" e "SFONDOAPP" di APRIFIN_EXT e Istruz. SEZIONEFIN e CLS_EXT	
Istruzioni di gestione della TABELLA PAGAMENTI (GETPAG - PUTPAG - DELPAG) e relative variabili _PA?	Richiesta n. 48978
CREATELIST: gestione di tasti funzione personalizzati tramite nuove variabili _CT e funzione RIPTAFCT()	
Istruzione APRISTAMP_EMAIL disponibile in Collage Server Remoto. Stesse condizioni APRISTAMP_EXT	vers. beta
Funzione FIRMAFILE per eseguire la firma digitale di un file mediante SmartCard	
Opzione "Numero ID" per la lettura con MYDBGGET dell'ultimo record in archivio	
Variabile _PCRIFSTD\$ Conto di collegamento al PDC standard 80/81 (righe redditi) - solo lettura -	
Variabile _MMTRACKNUM\$(T) Track number (tracciabilità o tracking) movimenti di magazzino	
Aggiornamento variabili _MMRINTR??? relative alla rintracciabilità in fase PUTMM (solo livello Produzione)	
Dizionario conti automatici sopravvenienze (A/P) e costo spese banche: 'casoa', 'casop', 'cacsb'.	
Dizionario 'az' e 'sa': 'azcel', 'azdtp', 'sadtpr', 'azdpr', 'sadtpr', 'azdin', 'sadin', 'azdan', 'sadan', 'azd77', 'sad77'	
Aggiornamento documentazione Collage DataTrek con Manuale DataTrek (v. PassBuilder in Manuale Utente).	

VERSIONE 2020H (V 78.3 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Evento ON_OUT_MDSELTEST: su finestra [Shift+F8] di interrogazione dati MyDB per documenti trasformati	Richiesta n. 58995

VERSIONE 2020F (V 78.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
<p style="text-align: center;">AVVISO IMPORTANTE RELATIVO AL CAMPO MDTDOC (prima di aggiornare occorre verificare la retrocompatibilità e la disponibilità dei sorgenti)</p> <p>Con l'implementazione dell'estensione del tipo di inserimento consentito nella Sigla Documento "MDTDOC" di Emissione Revisione Documenti, cambiano gli effetti in seguito agli eventi ad esso relativi.</p> <p>La pressione di [F12] (ON_612) inizializza la _SINP\$ con l'intero valore del campo a video (esempio FF TD20 Regulariz./integraz. FR non ricev/errate).</p> <p>La pressione di [F2] sul campo non scatena più l'evento ON_OUT, poiché questo avviene alla chiusura dell'elenco in seguito alla selezione del documento.</p> <p>La variabile _SINP\$ mantiene i 3 caratteri (2SigDoc+1ModStp) all'evento ON_IN e se viene modificata con una sigla documento riconosciuta valida (es. FT o FTE) viene impostata con tale sigla e si scatena il successivo evento ON_OUT e la _SINP\$ contiene tali caratteri.</p> <p>Diversamente, se _SINP\$ viene modificata con una qualsiasi altra stringa (es. tipo documento xml "TD04" o "fattura") tale per cui consente di rintracciare univocamente la corrispondente sigla documento di 2/3 caratteri, nella _SINP\$ verrà impostata quest'ultima e consegue l'evento ON_OUT, altrimenti viene proposto l'elenco dei risultati al fine di eseguire la scelta del documento, successivamente alla quale consegue l'evento ON_OUT con riporto della sigla nella _SINP\$.</p>	
Variabili _DVITPELETT\$ e _MMTIPELETT\$ (e dizionario mmtde) per il tipo documento elettronico (es.TD20).	
Parametri "RIGA" e "COLONNA" per MYDBVIEW e MYDBINPUT	

VERSIONE 2020E (V 78.0 GEST/DR)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
SQL_DRIVERCONN con parametro <modulo>=3. Consente la connessione diretta al database di replica. Eccetto il parametro tipo di connessione 3 non è richiesta la valorizzazione dei restanti parametri; consente la connessione a qualunque tipo di database di replica installato, configurato dal menu [Servizi-Configurazione moduli-MexalDB].	
Variabili _ERRIMAP e _ERRIMAP\$ di CHIUDISTAMP_EXT per la gestione dell'errore di memorizzazione di posta inviata (vedere anche documentazione del manuale utente, argomento: configurazione automatica per i principali account di posta elettronica in dispositivo stampante email (SMTP)	

VERSIONE 2020C (V 75.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Attivazione in Collage Server Remoto delle funzioni: ACAPO\$(), LUNGMOD\$(), SALTOPAG\$(), INIZCO\$(), FINECO\$(), INIZES\$(), FINEES\$(), INIZGR\$(), FINEGR\$(), INIZSO\$(), FINESOS\$(), INIZIT\$(), FINEIT\$()	
Dizionario 'azcba' relativo al codice conto della banca aziendale predefinita	Richiesta n. 58382
BLOCCASEM adeguamenti lunghezze e case (si rimanda alle nozioni generali del manuale utente)	

VERSIONE 2020B2 (V 75.1B GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
DBMIRROR/ASCMIRROR: colonna CDS_DENOM "Denominazione" (diz. paden/pcden) Rubrica CLI/FOR	

VERSIONE 2020B (V 75.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;

- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabili _MMT_IMPCPR? per la TOTMM dell'Imponibile cassa previdenza.	
Variabile _PNPTTSS\$ per la Tessera Sanitaria - Pagamento Tracciato; dizionario pnppt()	
Sprix Mobile: Eventi ON_BARCODE, ON_PRECLOSE, ON_CLOSE; Variabili _WIBARCODE\$, _WIBARNUM, _WIBARTYPE\$, _WIOUTONLY\$, _WINOTEMPTY\$, _WICKHEMPTY\$, _WITYPE\$ "TIME" / "TEXTAREA". Istruzione WCALL "OPENMENU" / "SETERROR".	

VERSIONE 2020A (V 75.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
APRISTAMP e APRISTAMP_EXT disponibili in Collage Server Remoto, stampe immediate su file e stampe fisiche, ma solamente se in assenza di interazioni I/O. Per approfondimenti consultare APRISTAMP.	esempio srvcall
SCHEDULATORE SPRIX per pianificare l'esecuzione automatica di Sprix in Live e Locale. Per gli ambienti Live, si ricorda di tenere in considerazione il periodo di fermo macchine per il backup. Si rimanda alla documentazione del menu Servizi - Configurazioni - Operazioni Pianificate.	
Adeguamento a 6 caratteri dei campi dizionario 'dtnpa', 'dunpa', 'cenpa' relativi alla "natura" fattura XML	
Adeguamento a 10 caratteri dei campi dizionario 'mmndr' e 'mmrtn'.	
_PCDENOM\$ variabile denominazione ditte individuali (dizionario 'paden' e 'pcden')	Richiesta n. 56858
Configurazione App: flag di aggiornamento App Azienda con Collage caricati, ma non in esecuzione	
Licenza MDS: consente la replica DB Datatrek in chiaro (vedere le "politiche" nelle prime pagine del manuale)	

VERSIONE 2019K (V 71.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
SETMAN del Help in Linea Sprix [Ctrl+F9] aggiornato con i comuni comandi editor	Richieste n. 1394 e 57285
Parametro "VISEST" di MYDBVIEW per avere in prima posizione la visualizzazione dell'estensione	
Compilazione massiva elementi Sprix/Collage dell'App (pulsante funzione [F12] nell'elenco elementi App)	
Abilitate le istruzioni APRIAZIE e APRIAZIE_EXT nell'esecuzione dell'elemento Collage Server Remoto della App PassBuilder, utile anche per il cambio della data nelle interrogazioni dei movimenti di magazzino in anni diversi da quello aperto e in uso.	

VERSIONE 2019J (V 71.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
INPUT di tipo password ("W" oppure "WP") con possibilità di attivare la conferma della password nelle INPUT che hanno parametri (INPUT_EXT, INPUT_EXT_EXT, INPUT3).	Richiesta numero 28751 esempio
LEGGIDIR: parametro per l'accesso alla catella dati dell'App	
_MMCBZPAG\$ variabile per codificare/ricamare "Banca Azienda" dai PAGAMENTI del piede documento	Richiesta numero 55027
GETCODART e variabili _CODARTNUM, _CODART\$, _ERRCODART\$ per la ricerca di codici articoli di anagrafica articoli accomunati in base all'input di vario tipo (descrizione, codice alternativo, alias, abbinato, chiave lotto, codice fornitore).	esempio
PUTMM non di stampa (0/1) può generare documento CO equivalente a quello generato con i parametri di stampa, ovvero, con la sezione "DOCUMENTO COMMERCIALE" e relativi campi "Fiscale" e "Non Fiscale".	Richiesta numero 57274
Parametro [GETPH] in GETDVIDOC <numdoc>, <numrev>, [[DIRLOC],[,GETPH=S]] che consente di interrogare PassHub per eseguire il download del file Xml (se disponibile e/o consentito).	Richieste n. 57177 e 54665
DELMM_EXT consente di eliminare anche documenti da Cruscotto (quindi non solo da Gestione Documenti)	Richiesta numero 57233
CREACF e variabili _CDF per la creazione/scorporo del Codice Fiscale	Richiesta numero 16616

VERSIONE 2019I1 (V 70.1A GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
MxRT: variabili _MMNODOCCOM\$ e _MMDOCGESS\$ per scontrino non fiscale/cortesia	

VERSIONE 2019I (V 70.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Dizionario 'sxml' GESTIONE ALIAS 'sxggm' GESTIONE GRUPPI MERCEOLOGICI	

VERSIONE 2019H (V 70.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Debugger: implementazione dei tasti Pagine di navigazione (Pag su - Pag giù - Home - Fine)	Richiesta numero 1618

VERSIONE 2019G2 (V 69.1B GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Ripristino di Libcurl.dll per compilazioni del Gestionale a 32bit. Non è garantito il funzionamento in macchine e/o versioni di S.O. non recenti. Resta valido, comunque, il consiglio dell'utilizzo di compilazioni a 64bit.	

VERSIONE 2019G (V 69.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
GETATTIL - funzione Collage che valorizza le variabili _IL? presenti a video. Solo menu/etichette 'PV'.	
_FTEOPNODO(R,100) - parametro "C" cancella nodo	https://easypass.passepartout.sm/Soluzione/67608
MXRT funzioni e variabili _RT? per gestire il Registratore Telematico e relative _MM? per PUTMM / GETMM	
Dizionario e Variabili _AR? - _MM? - _MMT? per la gestione ENPAM / ENPALS	
Versione v070a Shaker	

VERSIONE 2019F (V 69.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
_MYDBDIN(\$) variabile array di _MYDBELE elementi come _MYDB\$(). Contiene il flag di riportabilità sui documenti. Relativo alle anagrafiche DINAMICHE.	
Impegni di Lavorazione: variabili _ILTPVIN() e _ILORAVIN()	
Dizionario e Variabili _MM per deguamento MxÉcr (documenti AN / RE per scontrino elettronico)	
_MMNCGDOCRIF _MMNDOCRIF _MMDDOCRIF\$ _MMNMATRRT\$	
Sprix Mobile: parametro BARCODE in _WITYPE\$ e parametro FOCUSINPUT in WCALL	
Shaker a 64 bit (Java solo 64 bit e .NET a 32 bit e 64 bit). File di Librerie distribuiti: MSprixDn.dll (v 7.0) - mxspx.dll (v 7.0).	
La versione .NET a 64 bit, ha una dipendenza implicita dalla libreria Visual Studio MSVCR100.dll, installabile direttamente con un applicativo Microsoft (Microsoft V.C++ 2010 SP1 Redistributable Package (x64)).	
La libreria JAVA è compatibile con la versione JDK 12, pertanto è distribuita solo a 64 bit. La libreria comune tra Windows e Linux è: MSprixJ.jar (v 7.0)	
<ul style="list-style-type: none"> • Per Windows sono distribuite le seguenti librerie: mxspx.dll (v 7.0) - mxspxjni.dll (v 7.0) • Per Linux sono distribuite le seguenti librerie: libmxspx.so.7.0.0.0 - libmxspxjni.so.7.0.0.0 	
In ogni caso, le librerie 64 bit (.NET e JAVA) non consentono la gestione di file di dimensioni superiori a 2 GB.	

VERSIONE 2019C (V 68.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Funzioni e variabili di lettura file CSV: OPENCSV(), CSVITERSTR(), CSVITERNXT(), CSVROWVAL\$, CLOSECSV(), _ERRCSV, _ERRCSV\$	(rif. v66.2) esempio e tabella per argomento Clicca qui
GETINTERNET aggiornamento parametri del timeout di lettura/scrittura http e riduzione del default a 15 min: TimeoutLetturaGetInternet = <secondi> (default 900); TimeoutLetturaGetInternet = <secondi> (default 900);	
Aggiornamento tabelle dizionario per archivi pagamenti e parcellazione: "ct", "cp", "rt", "df"	
Funzione DECODP7M per decodificare i file firmati p7m (CADES)	
Variabili Struttura e Dizionario per la gestione Ritenuta d'Acconto, Gestione separata Inps, Cassa Previdenza. _PCALQRDA, _PCQIMPRDA, _PCCAUPRDA, _MMSTUDASS\$, _MMALQRDA, _MMQIMPRDA, _MMSPBANRDA\$, _MMCAUPRDA, _MMINPSS(R), _MMTPCAINPSS(R), _MMCAPROS(R), _MMPERCAPRO2, _MMPERCAPRO3, _MMQUCAPRO1, _MMQUCAPRO2, _MMQUCAPRO3, _MMTPCAPROS(R), _MMTPCAPRO2\$(R), _MMTPCAPRO3\$(R), _MMCALCRDA, _MMCALCCAPRO, _MMCALCINPS Attenzione! non agiscono retroattivamente su documenti progressivi , cioè, la modifica dei documenti di magazzino tramite Sprix (GETMM + nuove _MM + PUTMM) segue le logiche di funzionamento dello standard del menu di Emissione Revisione Documenti, per cui la modifica di documenti emessi con versioni precedenti, pur valorizzandone i nuovi campi, non effettua il ricalcolo del totale documento.	
Variabile _FTE_DOCXML che consente l'accesso da Collage al file _XML della FTE. Riporta l'handle del documento in quel momento per eseguire le opportune istruzioni generi XML al fine di modificare il file XML, in quanto la personalizzazione tramite le variabili _FTE all'evento ON_STP_XML: non consente l'intervento sul medesimo Tag, anche se apparentemente sembra composto da più elementi.	esempio

VERSIONE 2019B (V 67.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
_MMCDMYDB\$(T) se valorizzata indica che i dati MYDB sono quelli del codice destinatario	
IDFIN (identificativo nome finestra) nei parametri dell'aspetto di APRIFIN_EXT	
SETPARSCRDOCNM forza la scrittura di documento non modificabile se l'elettronico è già stato trasmesso	

VERSIONE 2019A (V 67.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
JSONGETSTRF legge dati di tipo JSON_STRING e scrive direttamente in un file	
INPUT3 estensione del comando INPUT_EXT_EXT gestisce automaticamente i messaggi di errore	
_MMARM\$() codice articolo proprietario dei progressivi.	
_PNCOMI\$ Competenza iva periodo precedente	
_FTEOPNODO\$(R,100) imposta l'operazione da eseguire nel nodo specificato	
variabili formato e self billing: _PCALLXML\$, _PCRALLFRM\$, _PCSELFBILL\$, _PCRGFISC\$, _PCIVADIFF\$	

VERSIONE 2018M (V 66.4 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
ON_STP_XML: etichetta collage che permette la modifica/integrazione del file Xml della fattura elettronica	
_FTE??? Variabili di struttura per la gestione della modifica dei file Xml della fattura elettronica	
_PCESTEROM\$() Trattamento soggetto in Esterometro	

VERSIONE 2018K (V 66.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
MYDBDEL TAB\$,"TOTALE" con azzeramento ID tabella azzeramento progressivo _MYDBID	Richiesta numero 54290

APRIAZIE_EXT parametro "FINATT"	
---------------------------------	--

VERSIONE 2018J (V 66.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Variabili codici Dare/Avere (bilancio RSM in formato XBRL) _PCRSMDA\$ / parda() e _PCRSMAV\$ / parav()	
Nuovi ID e relative DESCRIZIONI delle icone elencate in SETTF ed utilizzate da PRINTICONA, CREATELIST, APRIFIN_EXT	
Finestra attiva: numero e identificativo tramite le funzioni CERCACODFIN() e GETCODFIN\$(). Le funzioni sono implementate in sostituzione della lettura diretta del titolo finestra	

VERSIONE 2018I (V 66.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
APRIFIN_EXT estensione finestra con le Icone e Tooltip	
INPUT_EXT_EXT estensione dell'istruzione INPUT_EXT	
INPUT_TXT input multi-linee	
NRIGDISPFIN() e NCOLDISPFIN(,) numero righe/colonne disponibili	
SETTF (estensione del parametro <str\$> per le Icone e Tooltip).	
INPUT_EXT (modifica sintassi parametri e aggiunta di SOLOCLI)	
Variabili Alias: _ALID() - _ALTPUM\$() - ALTPPUM	
Associazione Docuvision - MyDB: classe <1700>	

VERSIONE 2018H (V 65.4 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Estensione eventi ON_START/PUT/DEL della struttura PC disponibili nelle Variazioni Contabili (menu Servizi) per le funzioni di variazione: Anagrafiche piano dei conti, Condizioni clienti/fornitori, Codici bilancio CEE, Conversione da codici bilancio CEE a codici UE	Richiesta numero 53376
MYDBGGET: aggiunto parametro "NODEF" per evitare valori iniziali.	
ARCHVIEW: nuovo parametro arch=3 Tab. Codici Catastali comuni	
CHIURDISTAMP_EXT, _ERRCHSTAMP/\$ stampa e-mail.	
Eventi Collage	
ON_ENDMSG: uscita della finestra dei messaggi differiti	
ON_STARTZA / PUTZA / ENDZA: Nominativi Aziendali	
ON_STARTTZ / DELTZ / PUTTZ / ENDTZ: Particolarità Prezzi	

VERSIONE 2018G4 (V 65.3D GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Fattura elettronica: variabile _PCDFATTEL\$ data inizio gestione	Richiesta numero 53322
MYDBGGET: sintassi a 3 cifre per l'utilizzo dei sezionali estesi	

VERSIONE 2018F (V 65.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Sostituzione del tasto funzione [F2] con [F3] per la conversione del testo MAIUSCOLO/minuscolo nella funzione di ricerca testo nell'editor [F11].	
Istruzione CONVPDFA per convertire file PDF in file PDF/A.	
Sprix Mobile: parametro CLOSEFORM per l'istruzione WCALL; input DIVIDER per variabile _WTYPE; linea di separazione in _WIVALUE\$; ID menu "PMMENU" di PassMobile per l'istruzione WGETOID.	

Particolarità: variabili _PARTCOND\$ e _PARTAPPLI\$	
JSON: set di istruzioni, funzioni, variabili per gestire file JSON	

VERSIONE 2018E (V 65.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
<p>Editor: indentazioni rapide e gestione a colonne del testo:</p> <ul style="list-style-type: none"> • tasto [Tab] per tabulare più righe selezionate; • selezione a colonne di testo tramite la pressione combinata dei tasti [Shift+Alt] con frecce di navigazione e dei tasti per la gestione del testo: [Shift+F11], [Tab], [Shift+Tab], [Canc], [BackSpace], [Ctrl+C/V/Z/X], [Invio], [Caratteri] <p>Nota: durante la selezione della colonna [Shit+Alt+Freccia.] nel editor compare una colonna "grigia" che indica il posizionamento verticale di gestione del testo, cioè il punto d'inserimento/cancellazione multiplo. La successiva selezione del testo con le frecce di navigazione diviene in reverse "blu" poichè è pronto ad essere gestito, anche oltre il fine riga.</p>	
ARCHVIEW: istruzioni e variabili di visualizzazione tabelle ABI e CAP	
SprixMobile: Carrello (_WLTYPE="CART") e relativi eventi (CART), iteratore archivi (ITERARC), input _WITYPE di tipo taglie ("ARQTA") e numerico ("NUMBER"), variabile back form (_WFPCFEVT), variabili icone personalizzate (_WIICO\$ e _WLVALICO\$)	
_MMIDALIAS(R) numero alias nelle righe documento di magazzino	Richiesta numero 33145

VERSIONE 2018D (V 65.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
_MMRIFAMMPA\$ (rif. amm. PA) valorizzata nel documento BC	Richiesta numero 50722
Variabili _PC??? e Dizionario dei dati trasporto dell'anagrafica cliente da riportare nei documenti	Richiesta numero 43983
Estensione del parametro <tpart\$> per la selezione di Particolarità in base a nuove variabili di struttura (_PART) e dizionario (tz).	Richiesta numero 43995
Istruzione e ciclotabellare per Contratti di Locazione PARAM_REDD <sog\$>.[mod\$]	
Dati CDG nelle variabili _MM???	Richiesta numero 51763
Parametri Download e Upload per INPUT_EXT in Client WebDesk	Richiesta numero 38938
GETINTERNET anche per protocollo "https://"	
GETINTERNET sottoposta al controllo accesso utente	
SPXREMOTO\$() avvisi / errori tramite invio EMAIL SMTP	
Gestione file .ZIP (ZIP_CREA, ZIP_ESTRAI, ZIP_LISTA, _ERRZIP\$)	
Codifica/Decodifica file in Base64 (CODIF_BASE64)	
Variazione contenuto nodi file XML (XMLINSDATI)	
CREAUUID\$(): funzione per generare un UUID	

VERSIONE 2018C (V 64.5 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Duplicazione degli elementi nella medesima App	richiede licenza PassBuilder
File log PassBuilder dedicato: app_passbuilder_ANNOMESE.log	richiede licenza PassBuilder
Variabili _WI??? per la gestione dei Codici Strutturati	richiede licenza PassBuilder
Variabile Sprix Mobile _WFPCFMSG\$ messaggio di chiusura form (e relativi eventi)	richiede licenza PassBuilder
PUTASSOCDV con parametro di stampa	

VERSIONE 2018A (V 64.3 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Istruzioni LPRINTMG e ESISFILE accedono alle cartelle dati di PassBuilder con la sintassi cartella\$nomefile.	richiede licenza PassBuilder
GETAUTORIZUTE in relazione ai conti Clienti / Fornitori	
NUMPAGPDF: funzione che restituisce il numero pagine di file PDF	
UNISCIPDF: istruzione per unire più file pdf	

VERSIONE 2017K (V 64.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Aggiornamento Split Payment: Nuova array _PCDTSPL\$(1-6) e nuovi valori da "A" in "S" per _PCCATSPLIT\$ e per _MMSPLITPAY\$.	
Gestione ultimo modulo in APRISTAMP_EXT e APRISTAMP_EMAIL	richiede licenza PassBuilder
Istruzione APRISTAMP_FILE per PDF / CSV	richiede licenza PassBuilder
Abilitazione nell'esecuzione COLLAGE SERVER REMOTO (elemento componente PassBuilder) delle istruzioni LPRINT? e CHIUDISTAMP	richiede licenza PassBuilder
HTTPHDRTAG\$() funzione Web Service per gestire header superiori a 255 caratteri (massimo valore variabile Sprix)	richiede licenza PassBuilder
Istruzione FTPSECCMD e struttura _FTP per interagire da SPRIX/COLLAGE con server FTP / SFTP / FTSP (per SFTP è consigliata compilazione Gest 64bit, poiché in S.O. 32bit non recenti può non funzionare).	richiede licenza PassBuilder
Istruzione GETMODSTPGRAF per caricare moduli di stampa Sprix predefiniti e personalizzati fuori dall'App.	richiede licenza PassBuilder
Dizionario sycpl e sycpp (credenziali Clicca e Posta)	richiede licenza PassBuilder
Modifica "a caldo" di App (con azienda aperta); tranne MyDB/Datatrek	richiede licenza PassBuilder
Duplicazione App	richiede licenza PassBuilder
MyDB accessibili da App diverse	richiede licenza PassBuilder
Ordinamento Collage all'interno della stessa App	richiede licenza PassBuilder

VERSIONE 2017I (V 64.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
SHELLSPRIX che esegue sprix non a menu (Sprix000)	richiede licenza PassBuilder
Istruzioni e variabili Calendario (struttura _CM)	richiede licenza PassBuilder
Variabili Sprix Mobile _WL di ordinamento e filtri nelle liste dinamiche	richiede licenza PassBuilder
Parametro [modstsp] per le istruzioni di APRISTAMP_??? e relative variabili LPRINTGRAF_??? per le stampe del modulo grafico Sprix in App PassBuilder.	richiede licenza PassBuilder
Variabili _WS per gestire timeout e funzione TCP KeepAlive	richiede licenza PassBuilder
Variabili _AR per gestire i listini estesi	richiede licenza PassBuilder
Aggiornamento parametro codice <codcon\$>="0" in GETPARTICOL per particolarità senza conto e senza categoria.	

VERSIONE 2017G (V 63.3 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
MYDBDEL con parametro "TOTALE"	
Variabile Sprix Mobile _WIICO\$ per l'inserimento delle icone	richiede licenza PassBuilder
Istruzione Sprix Mobile ACQUISISICIFILE per l'acquisizione del file da dispositivo mobile e relativo oggetto acquisito _ACQFILE\$	richiede licenza PassBuilder
Variabile Sprix Mobile _WLFMTFLD\$() per la formattazione della visualizzazione dei numerici	richiede licenza PassBuilder
Utilizzo di variabile _MMSCO\$(R) in Sprix Mobile e gestione archivio Omaggi/Abbuoni/Garanzie	richiede licenza PassBuilder
Variabile _SSDTPSTP\$ di stampa con invio email PassMobile	richiede licenza PassBuilder
Variabile _WLVALICO\$ per le icone nelle liste dinamiche	richiede licenza PassBuilder

VERSIONE 2017F (V 63.2 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
Sprix Mobile: azione "PRINTFORM" nell'istruzione WCALL per la stampa offline di un form.	richiede licenza PassBuilder
Sprix Mobile: istruzioni SRVCALL, GETREM, PUTREM e le relative _ERR per la chiamata al Server.	richiede licenza PassBuilder
Sprix Mobile: istruzione SRVSYNCDoc e relative variabili _SSD e _ERR per sincronizzare il documento.	richiede licenza PassBuilder
Sprix Mobile: dizionario mfsmx, mfbcs, mqscn, mqqta, sxute, sxnut, azpri, azult, ammag	richiede licenza PassBuilder

Sprix Mobile: istruzione GETAUTORIZUTE gestione utenti	richiede licenza PassBuilder
Sprix Mobile: istruzione GEOMAPPA gestione mappa	richiede licenza PassBuilder
Sprix Mobile: istruzione VISUALIZZAFIle gestione file	richiede licenza PassBuilder
Sprix Mobile: variabile _WLROWICO\$ gestione icona	richiede licenza PassBuilder
_MMSPLITPAY con valori 'A', 'a', 'Z'	richiede licenza PassBuilder
Sintassi del Nome Libreria in istruzione #APPENDISPX	richiede licenza PassBuilder
Sintassi Nome File in SHELLSPRIX, SPXREMOTO, CARCOLLAGE, STARTCOLLAGE, STOPCOLLAGE, ABILCOLLAUTO, CALL. Per tutte le istruzioni la sintassi è riconducibile a quanto specificato nel parametro "-p" di SHELLSPRIX.	richiede licenza PassBuilder
Elaborazioni batch di App PassBuilder ed esecuzione tramite mxrs o mxrsbatch di Sprix residente in una PassApp. Per la consultazione si rimanda al capitolo "Nozioni Generali" del Manuale Generale Utente.	richiede licenza PassBuilder
Funzione LIVLICAPP() per il livello di App sul PassStore	richiede licenza PassBuilder
MYDB campo di tipo Password	

VERSIONE 2017E (V 63.1 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
_STPLUNGRIG; _STPLARGCOL; _STPLUNGRMM; _STPLARGCMM;	richiede licenza PassBuilder
LPRINTLINE	richiede licenza PassBuilder
CREATELIST	richiede licenza PassBuilder
GEOMAPPA	richiede licenza PassBuilder
INPUT Internet, Email, Telefono	
_ARTSC\$; _ARTST\$;	
_MMANNOLETT(R); _MMPROGLETT(R);	
_PCRATING\$; _PCGGRIT(2-13);	
GETAUTORIZUTE con filtro "_PCCOD" per il 3° parametro e nuovo 4° parametro [filtro\$]; vettore _AUVAL\$() con valori multipli;	

VERSIONE 2017D (V 63.0 GEST)

Notare:

- le correzioni di anomalie sono indicate nella Circolare di rilascio della R&S;
- l'organizzazione logica e gli esempi relativi agli argomenti presenti nella sottostante tabella è contenuto nella relativa sezione "COMANDI PER ARGOMENTO";
- approfondimenti su ciascun argomento si trovano in ciascun singolo comando elencato in ordine alfabetico nella sezione "LIBRERIA DI ISTRUZIONI - FUNZIONI - VARIABILI".

	nota
_MSGSVILDB; _MSGSVILDB\$;	richiede licenza PassBuilder
GETPARTICOL con parametro [mod]=2	richiede licenza PassBuilder
MYDBVIEW con parametro "NOPUT"	
MYDBVIEW con parametro "RIGHE=??"	