

Guida Rapida per la migrazione da MicroFocus COBOL verso isCOBOL Evolve.



Questo prodotto o documento è protetto da Copyright ed è distribuito sotto licenze che ne limitano l'uso, la copia, la distribuzione e la ricompilazione. Nessuna parte di questo prodotto o documento può essere riprodotta in qualsiasi forma e con qualunque mezzo, senza previa autorizzazione scritta di **Veryant**.

isCOBOL e Veryant sono marchi registrati di **Veryant LLC** negli Stati Uniti e negli altri paesi. Tutti gli altri marchi appartengono ai rispettivi proprietari.

Copyright© 2011 Veryant LLC.
9930 Derby Lane, Suite 202, Westchester, IL 60154 U.S.A.
All rights reserved.

Copyright© Veryant Italia s.r.l.
Via Pirandello 29 – 29121 Piacenza
Tutti i diritti riservati.

Questa guida intende aiutare a comprendere le somiglianze e le differenze tra la suite **isCOBOL™ Evolve** di **Veryant** e MicroFocus® COBOL; intende sottolineare la semplicità del processo di migrazione verso **isCOBOL**.

La migrazione di un'applicazione software, tra una piattaforma ed un'altra è un processo piuttosto complesso, delicato e costoso; tale attività necessita di una attenta analisi mirata ad individuare ed utilizzare quel prodotto che permetta di mantenere inalterato il patrimonio software, che permetta di ridurre al minimo i tempi ed i costi di sviluppo e che permetta al software stesso di sfruttare i benefici offerti dalle nuove tecnologie presenti sul mercato.

La guida comprende:

- una breve descrizione della suite **isCOBOL Evolve**;
- una tabella che riassume le direttive di compilazione di MicroFocus COBOL supportate da **isCOBOL Evolve** attraverso opportune opzioni di compilazione;
- una tabella che elenca le routine di MicroFocus COBOL supportate da **isCOBOL Evolve**;
- una tabella che elenca le differenze sintattiche e di comportamento tra **isCOBOL Evolve** e MicroFocus COBOL.

isCOBOL™ Evolve di **Veryant**, è un ambiente di sviluppo completo e potente che rispetta lo standard ANSI del COBOL e supporta i maggiori dialetti presenti sul mercato, per questo motivo offre la possibilità di migrare le applicazioni scritte con la suite MicroFocus® COBOL in modo semplice, trasparente e veloce grazie ad una serie di moduli che ne sostituiscono in tutto per tutto le funzionalità.



Inoltre, **isCOBOL™ Evolve** garantisce allo sviluppatore, totale portabilità, elevata efficienza, ampia scalabilità delle applicazioni, bassi costi di sviluppo e distribuzione, nonché una più efficace propensione all'uso delle nuove tecnologie grazie alla traduzione automaticamente del sorgente COBOL in codice Java ed alla generazione, sempre automatica, delle classi Java, eseguite poi con la Java Virtual Machine (JVM). Questo processo è totalmente trasparente per lo sviluppatore. Le classi Java generate potranno poi essere eseguite come programmi stand-alone o come parte di un'applicazione di tipo Enterprise. Il fatto che le classi Java derivino originariamente da codice scritto in COBOL non influisce in alcun modo sulla funzionalità del prodotto.

Con **isCOBOL™ Evolve** non è necessaria alcuna manutenzione o rifacimento del codice sorgente. Lo sforzo di modernizzazione dell'applicativo risulta pressoché nullo, l'interoperabilità e l'integrazione con gli altri software sono davvero facili.

L'infrastruttura di **isCOBOL™ Evolve** è basata su un'organizzazione a tre livelli (Three Tier), gestisce il processo di connessione tra Client e Server evitando, tra l'altro, l'installazione di applicazioni complesse sul Client. I Server possono essere di tipo Application-Server preposti all'elaborazione vera e propria, oppure Data-Server preposti alla gestione della base dati.

Allo sviluppatore ed all'utilizzatore, la suite **isCOBOL™ Evolve** mette a disposizione una serie di componenti che ottimizzano ed accelerano il processo di sviluppo, installazione ed uso del software applicativo.



isCOBOL Compiler. E' un ambiente di sviluppo 100% Java-based, compatibile con gli standard ANSI, in grado di generare codice orientato agli oggetti (OOP), capace di supportare la sintassi SQL standard all'interno del codice COBOL.

isCOBOL IDE. E' un ambiente visuale, flessibile e potente, basato sul framework Eclipse, con il quale è possibile gestire, in modo rapido e semplice, tutte le attività di sviluppo di qualsiasi tipo di applicazione (disegnare, codificare, testare, debuggare).

isCOBOL Debugger. E' un ambiente di debugging multi-piattaforma e completo, capace anche di analizzare e gestire programmi che risiedono su macchine diverse da quelle che ospitano i sorgenti (Remote Debugging).

Veryant Database Bridge. Basato su tecnologia 4GL, è in grado di leggere la struttura degli archivi COBOL e di generare programmi COBOL contenenti le chiamate SQL (ESQL) necessarie per la gestione di quegli archivi. L'interazione con i RDBMS avviene attraverso l'opportuno driver JDBC.

isCOBOL Runtime. E' un runtime in grado di funzionare su qualunque piattaforma che supporta il Java Runtime Environment (JRE); include una interfaccia utente multi-piattaforma, basata sui Java SWING che consente di gestire tutti i tipi di Look-And-Feel (LAF) disponibili.

isCOBOL Server. Gestisce i processi di comunicazione tra Server e Client, invia al Client l'interfaccia utente da visualizzare e ne gestisce il flusso, crea una sessione per ciascun utente connesso, lo identifica e lo abilita. Evita che sul Client si rendano necessarie installazioni complesse.

Veryant JISAM. Filesystem ISAM scritto in Java, snello e rapido, è adatto a tutte le piattaforme Java-compatibili soprattutto i dispositivi mobile in genere.

isCOBOL ISAM Server. Filesystem transazionale potente, sicuro e scalabile che può essere usato in ambiente standalone oppure su Server di rete, in questo caso opera anche come File-Server occupandosi di gestire il flusso di dati tra Client e Server in modo ottimizzato e sicuro.

isCOBOL SQL Server. Motore SQL per un accesso ai dati potente e flessibile, che offre la possibilità di gestire dal mondo esterno la base dati ISAM come se fosse un vero e proprio RDBMS senza obbligare alcun tipo di migrazione dei dati stessi.

isCOBOL ISAM ODBC. Questo driver garantisce un facile accesso ai dati COBOL direttamente dai più comuni strumenti di Office Automation e da applicazioni scritte con strumenti dotati di accesso ODBC. I dati sono raggiungibili via TCP/IP su qualsiasi macchina, locale o remota.

isCOBOL WEB Direct 2.0. Le applicazioni COBOL dotate di Screen-Section, saranno eseguite sul WEB grazie alla tecnologia AJAX; l'applicazione risiederà completamente sul Server e non sarà necessario scaricare nulla sul Client.

isCOBOL Load Balancer. Permette di incrementare velocità e stabilità grazie alla distribuzione del carico di lavoro tra i cluster dei Server in esecuzione. I cluster possono risiedere sulla stessa oppure su macchine differenti, la distribuzione del carico di lavoro aumenta la scalabilità orizzontale.

Le direttive di compilazione di MicroFocus COBOL non sono direttamente supportate da **isCOBOL™ Evolve**, ciò nonostante le loro funzionalità vengono ottenute usando opportune opzioni di compilazione.

Va ricordato che quando si compila un sorgente scritto con MicroFocus COBOL, per ottenere un alto livello di compatibilità sintattica e funzionale, devono essere sempre utilizzate le opzioni di compilazione **-cm** e **-dcm**.

Nella seguente tabella vengono elencate le direttive di compilazione di MicroFocus COBOL e la soluzione **isCOBOL™** che sortisce il medesimo risultato.

Direttiva MicroFocus	Soluzione isCOBOL
ADDSYN.	Utilizzare l'opzione di compilazione -rm
ALIGN.	Utilizzare l'opzione di compilazione -align
ANIM	Utilizzare l'opzione di compilazione -d
APOST	Utilizzare l'opzione di compilazione -apost
ASSIGN	Utilizzare l'opzione di compilazione -cax
BOUND	Utilizzare l'opzione di compilazione -m1 e impostare a 1 la proprietà di configurazione iscobol.array_check
CHANGE-MESSAGE	Impostare opportunamente la proprietà di configurazione iscobol.compiler.messagelevel
COBFSTATCONV	Impostare opportunamente la proprietà di configurazione iscobol.file.status
COBOL370	Utilizzare l'opzione di compilazione -cv
CONSTANT	Impostare opportunamente la proprietà di configurazione iscobol.compiler.constant properly
CONVERTRET	Utilizzare l'opzione di compilazione -d5
COPYEXT	Utilizzare l'opzione di compilazione -ce=
COPYLIST	Utilizzare l'opzione di compilazione -lf
CURRENT-DATE	Funzionalità della classe com.iscobol.rts.CurrentDate

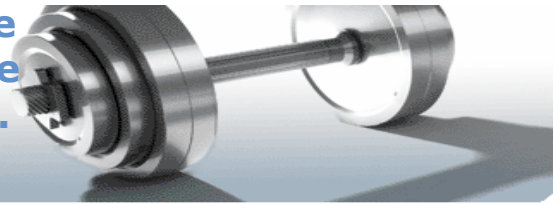
Corrispondenza tra le direttive di compilazione di MicroFocus COBOL e le opzioni di compilazione della suite **isCOBOL™ Evolve.**



Direttiva MicroFocus	Soluzione <i>isCOBOL</i>
DATAMAP	Utilizzare l'opzione di compilazione <i>-ld</i>
DEFAULTBYTE	Utilizzare l'opzione di compilazione <i>-di</i> e <i>-dv=</i>
DOS/VS	Utilizzare l'opzione di compilazione <i>-cv</i>
ERRLIST	Utilizzare l'opzione di compilazione <i>-ef</i>
FAULTFIND	Impostare a 1 la proprietà di configurazione <i>iscobol.exception.dump</i> e a 3 la proprietà <i>iscobol.exception_message</i>
FLAGQ	Utilizzare l'opzione di compilazione <i>-es</i>
HIDE-MESSAGE	Impostare opportunamente la proprietà di configurazione <i>iscobol.compiler.messagelevel</i>
IBMCOMP	Utilizzare l'opzione di compilazione <i>-dcmi</i> invece di <i>-dcm</i>
KEYCHECK	Impostare opportunamente la proprietà di configurazione <i>iscobol.file.extra_keys_ok</i>
LIST, LISTPATH, LISTWIDTH, LW	Utilizzare le opzioni di compilazione <i>-lf</i> e <i>-lo</i>
MAKESYN	Utilizzare l'opzione di compilazione <i>-rm</i>
ODOSLIDE	Utilizzare l'opzione di compilazione <i>-cod1</i>
OPTIONAL-FILE	Impostare a 1 la proprietà di configurazione <i>iscobol.file.io_creates</i> e la proprietà <i>iscobol.file.extend_creates</i>
OVERRIDE	Utilizzare l'opzione di compilazione <i>-rc</i>
PERFORM-TYPE	Utilizzare l'opzione di compilazione <i>-pt0</i> oppure <i>-pt1</i> oppure <i>-pt2</i>
PRINT	Utilizzare l'opzione di compilazione <i>-lo</i>
REMOVE	Utilizzare l'opzione di compilazione <i>-rw=</i>
RM	Utilizzare l'opzione di compilazione <i>-dci</i>
SEQUENTIAL	Utilizzare l'opzione di compilazione <i>-cfl</i>
SOURCEFORMAT	Utilizzare l'opzione di compilazione <i>-sa</i> oppure <i>-st</i>
TRACE	Impostare opportunamente la proprietà di configurazione <i>iscobol.tracelevel</i>

Le routine di MicroFocus COBOL più utilizzate sono supportate da *isCOBOL™ Evolve*, alcune di esse mantengono lo stesso nome e la stessa sintassi, altre hanno nome differente (che deve quindi essere sostituito nel sorgente COBOL) ma svolgono le stesse funzioni. La tabella sottostante elenca le routine MicroFocus e la corrispondente soluzione *isCOBOL™*.

Corrispondenza tra le routine di MicroFocus COBOL e quelle della suite *isCOBOL™ Evolve*.



Routine MicroFocus	Soluzione <i>isCOBOL</i>
CBL_ALLOC_DYN_MEM	M\$ALLOC
CBL_ALLOC_MEM	M\$ALLOC
CBL_ALLOC_SHMEM	M\$ALLOC
CBL_ALLOC_THREAD_MEM	M\$ALLOC
CBL_AND	CBL_AND
CBL_CHANGE_DIR	C\$CHDIR
CBL_CHECK_FILE_EXIST	C\$FILEINFO
CBL_CLEAR_SCR	Istruzione DISPLAY
CBL_COPY_FILE	CBL_COPY_FILE
CBL_CREATE_DIR	CBL_CREATE_DIR
CBL_DELETE_DIR	CBL_DELETE_DIR
CBL_DELETE_FILE	CBL_DELETE_FILE
CBL_ERROR_PROC	CBL_ERROR_PROC
CBL_EXEC_RUN_UNIT	C\$RUN
CBL_EXIT_PROC	CBL_EXIT_PROC
CBL_FREE_DYN_MEM	M\$FREE
CBL_FREE_MEM	M\$FREE
CBL_FREE_RECORD_LOCK	I\$IO
CBL_GET_CSR_POS	Special Name CURSOR
CBL_GET_OS_INFO	C\$SYSINFO
CBL_GET_RECORD_LOCK	I\$IO
CBL_LOCATE_FILE	C\$FULLNAME oppure C\$FSFULLNAME in funzione del tipo file.
CBL_NOT	CBL_NOT
CBL_OR	CBL_OR
CBL_RENAME_FILE	RENAME
CBL_TEST_RECORD_LOCK	I\$IO
CBL_TOLOWER	C\$TOLOWER
CBL_Toupper	C\$TOUPPER
CBL_XOR	CBL_XOR

Per semplificare e rendere più veloce il processo di migrazione vengono riportate nella seguente tabella le più importanti differenze sintattiche e di comportamento tra MicroFocus COBOL ed *isCOBOL™ Evolve*.

Differenze sintattiche e di comportamento tra MicroFocus COBOL e *isCOBOL™ Evolve*.



La direttiva \$SET OVERRIDE che cambia il significato delle parole riservate, non è supportata, ma lo stesso risultato può essere ottenuto con l'opzione di compilazione **-rc=word, customword..**

La sintassi per i commenti nelle linee di codice è **|commento** e non //commento.

LOCAL-STORAGE SECTION non è supportata, lo stesso risultato lo si ottiene con la proprietà di configurazione **iscobol.recursion_data_global**.

La sintassi z"String" non è supportata, per aggiungere x00 ad un testo si usa l'istruzione **STRING**.

USAGE COMP-5 non può essere applicata alle variabili alfanumeriche.

Lo Special name ALPHABETIC IS ASCII non è supportato, si ottiene lo stesso risultato rimpiazzando **ASCII** con **STANDARD-1**.

CALL_CONVENTION non è supportata, la proprietà **iscobol.dll_convention** nel file di configurazione sortisce lo stesso effetto.

Nel FILE-CONTROL, le sintassi ASSIGN TO FROM nome, ASSIGN TO LINE ADVANCING, ASSIGN TO LPT1, LPT2 ecc, ASSIGN TO PRN non sono supportate.

SUPPRESS non è supportato nella descrizione delle chiavi.

La clausola RETURN non è supportata nel paragrafo IO-CONTROL.

La clausola VALUE non è supportata in REDEFINES.

COMMUNICATION SECTION non è supportata.

BY REFERENCE, BY VALUE, BY CONTENT non possono essere usate nelle istruzioni INVOKE, CHAIN e ENTRY, questa sintassi è supportata solo dall'istruzione **CALL**.

L'istruzione ACCEPT non mostra il valore iniziale se non si specifica la clausola WITH UPDATE. L'opzione di compilazione **-vu** usa la clausola in tutte le ACCEPT.

ACCEPT FROM EXCEPTION STATUS non è supportato, un effetto simile lo si ottiene specificando la clausola **ON EXCEPTION** oppure **NOT ON EXCEPTION** nell'istruzione **ACCEPT**.

ACCEPT FROM USER NAME non è supportata, l'istruzione **ACCEPT FROM ENVIRONMENT** restituisce il contenuto della proprietà **iscobol.user_id**.

ACCEPT FROM CRT non è supportata.									
La clausola MODE IS BLOCK non è supportata nell'istruzione ACCEPT									
ADDRESS OF data-item non può essere passato come parametro tra i programmi e neanche essere usato come exit status.									
La clausola WITH TIMEOUT non è supportata nell'istruzione ACCEPT, può essere rimpiazzata con la clausola BEFORE TIME .									
La clausola FOR REMOVAL nell'istruzione CLOSE non è supportata, può essere rimpiazzata con la clausola WITH LOCK per ottenere lo stesso effetto.									
DISPLAY SPACES non pulisce lo schermo, bisogna aggiungere la clausola ERASE EOS .									
Le istruzioni OPEN e CLOSE non possono essere usate nei puntatori, ma solo nei file.									
L'istruzione EXHIBIT non è supportata, può essere rimpiazzata dall'opportuna istruzione DISPLAY per raggiungere lo stesso scopo.									
La clausola WITH WAIT non è supportata dall'istruzione READ.									
La clausola ADVANCINT TAB/FORMATTED non è supportata dall'istruzione WRITE.									
La sintassi VALUE NEXT, per riprodurre lo stesso effetto di MF deve essere accompagnata dalle opzioni -m1 e -align8 .									
La classe speciale com.iscobol.rts.CurrentDate Class permette di ottenere lo stesso effetto dei seguenti entry di configurazione di MF									
<table> <tr> <td>current_day</td> <td>current_hour</td> <td>current_minute</td> <td>current_month</td> </tr> <tr> <td>current_second</td> <td>current_year</td> <td>datewarp_dynamic</td> <td></td> </tr> </table>	current_day	current_hour	current_minute	current_month	current_second	current_year	datewarp_dynamic		
current_day	current_hour	current_minute	current_month						
current_second	current_year	datewarp_dynamic							
I file ISAM ad indici di MF non sono direttamente supportati da isCOBOL ma possono facilmente essere convertiti utilizzando le utility di conversione dati CTUTIL e JUTIL .									
I file-status Code restituiti da isCOBOL sono quelli indicati nell'ANSI 2002. Per poter ottenere quelli restituiti da MF deve essere aggiunta la seguente riga di configurazione: iscobol.file.status=com.iscobol.io.FileStatusMF									
Le seguenti funzioni intrinseche non sono supportate da isCOBOL									
<table> <tr> <td>CHAR- NATIONAL</td> <td>DISPLAY-OF</td> <td>E</td> </tr> <tr> <td>EXP</td> <td>FRACTION-PART</td> <td>LENGTH-AN</td> </tr> <tr> <td>NATIONAL-OF</td> <td>PI</td> <td>SIGN</td> </tr> </table>	CHAR- NATIONAL	DISPLAY-OF	E	EXP	FRACTION-PART	LENGTH-AN	NATIONAL-OF	PI	SIGN
CHAR- NATIONAL	DISPLAY-OF	E							
EXP	FRACTION-PART	LENGTH-AN							
NATIONAL-OF	PI	SIGN							