



Transitioning from Micro Focus COBOL to Veryant

Solution Overview

Copyright © 2011 Veryant.

925 Vista Park Drive, Pittsburgh, PA 15205 U.S.A.

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution and recompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Veryant and its licensors, if any.

isCOBOL and Veryant are trademarks or registered trademarks of Veryant LLC in the U.S. and other countries. All other marks are property of their respective owners.

Table of Contents

Introduction	4
isCOBOL Evolve	5
Phased transition option	5
Migration process overview	5
Compiler options	7
Command line options	7
Language syntax	8
Runtime configuration variables (properties)	8
Environment variables	8
Accommodating variations	8
Data access	10
Sequential files	10
Relative files	10
Indexed files	10
Relational databases	10
External file systems	11
File status codes	11
User interfaces	12
Character-based user interfaces	12
Graphical user interfaces	13
Function keys	13
Interoperability	15
Library routines	15
Phased transitions	17
Conclusion	18
For more information	18

Introduction

Veryant delivers comprehensive COBOL application development, maintenance and modernization technologies that optimize IT resources, improve business performance, and dramatically lower cost.

Organizations are converting applications from Micro Focus® COBOL platforms such as Micro Focus Server Express™ and Net Express® to Veryant technology for many reasons:

- to gain the advantage of a COBOL development and Java deployment model without rewriting or retraining.
- to improve ROI and dramatically lower COBOL licensing fees.
- to deploy to multiple platforms using a single set of source code.
- to modernize applications with graphical thin client and web client options.
- to natively integrate with leading Web and application server technologies.
- to support a wide range of data access options and capabilities.

The Veryant COBOL product family includes isCOBOL™ Evolve. isCOBOL Evolve is an innovative COBOL platform for standalone applications that rely on COBOL screen user interfaces. The isCOBOL platform provides a comprehensive and cost-effective environment for COBOL application development, deployment, maintenance and modernization. With Veryant, all development and debugging tasks are performed in familiar, flexible and user-friendly COBOL environments, obviating the need for retraining or rewriting of application source code.

isCOBOL Evolve

isCOBOL Evolve is an innovative COBOL platform that simplifies application maintenance and modernization, while accelerating time-to-market and integration with Web initiatives and external JEE server environments on UNIX®, Linux® and Microsoft® Windows® based systems. In addition to a robust COBOL compiler, integrated development environment (IDE), graphical and remote source-level debugging facility, and numerous data access options, isCOBOL Evolve offers a portable graphical user interface (GUI) for standalone COBOL applications and is able to execute on any device running the Java Virtual Machine (JVM) – from mainframes to mobile phones. By uniquely blending proven COBOL development with the unlimited potential of Java platform deployment, isCOBOL delivers true portability, easy extensibility and a clear path forward for valuable application assets.

Phased transition option

With Veryant, organizations have the choice of either migrating an entire application codebase at once or of taking a phased approach which sets the pace of a conversion according to resource availability and desired timeframe. During a phased migration, an organization's development team can continue to maintain an application with the Micro Focus COBOL compiler, while simultaneously working on the same set of source code to compile and execute with a Veryant COBOL compiler. Veryant provides compiler and runtime compatibility settings to keep the overall level of effort required for conversion to a minimum.

Migration process overview

The key steps involved in a conversion to a Veryant COBOL platform are:

1. Set up the isCOBOL Software Development Kit (SDK)
2. Recompile the application; review compiler output and make minor syntax modifications where necessary
3. Migrate data using Veryant-supplied conversion utilities

4. Set up the target platform runtime environment, including environment variables and data connections
5. Integrate with external software libraries and routines such as those written in the C programming language

The remainder of this document examines how Micro Focus COBOL programs and data files can be rapidly converted to Veryant COBOL. The paper specifically looks at compiler options, data access, user interfaces, interoperability requirements and the phased transition approach.

Compiler options

Veryant SDKs include a 100% portable COBOL compiler which supports the latest ANSI standards as well as common legacy dialects and XML data exchange. Embedded SQL (ESQL) is supported either through database-specific pre-compilers, such as Oracle Pro*COBOL or DB2 PRECOMPILE, or through Java Database Connectivity (JDBC). In the latter case, no ESQL pre-compiler is required. Developers do not need to be familiar with the Java programming language in order to work with a Veryant compiler and all development and debugging is done in the COBOL language.

Veryant's COBOL compilers operate as command line utilities, just like Micro Focus COBOL. Existing Micro Focus build scripts can be used in a Veryant COBOL environment by changing relevant compiler executable names and command line options. Veryant offers a high degree of compatibility with Micro Focus COBOL syntax, and for those parts of an application that rely on Micro Focus-specific semantics, compatibility options for compilation and runtime behavior are provided.

Command line options

Veryant and Micro Focus largely use different conventions for specifying compiler options at the command-line and in application source code. Veryant technology uses standard command-line style options, whereas Micro Focus COBOL uses its own compiler directive style.

With Veryant, command line options start with a hyphen followed by an abbreviation, and then sometimes an equal sign '=' followed by a value. These command-line style options can be specified in a file using the "compiler.options" property or in the COBOL source code using the ">> IMP OPTION" directive.

For example, to provide compatibility with the mainframe behavior of the PERFORM statement such as on OS/VS COBOL:

- Micro Focus COBOL requires the compiler directive PERFORM-TYPE"OSVS"
- Veryant requires the compiler command-line option -pt2

Language syntax

Small code changes may need to be made to items such as inline comments, for example:

- 01 WS-VAR PIC X. // Micro Focus inline comment
- 01 WS-VAR PIC X. | Veryant inline comment

Runtime configuration variables (properties)

At runtime, there are a number of Veryant properties that affect application behavior. These properties can be adjusted to obtain the same behavior as Micro Focus COBOL.

For example, a default behavior of Micro Focus COBOL is to strip trailing spaces from line sequential file records during WRITE operations. The same behavior can be obtained in a Veryant COBOL environment by setting the runtime property "file.strip_trailing_spaces=true".

Another example is accommodating the Micro Focus RTS Switch "N", which is default behavior in Micro Focus COBOL for including computational data in line sequential files. This behavior is obtained with Veryant COBOL by setting the property "file.linesequential_N=true".

Environment variables

No change is required to handle environment variables when moving from Micro Focus COBOL to Veryant. The same COBOL syntax is used to get and set environment variables.

Accommodating variations

Veryant's support team has considerable experience assisting customers in migration off the Micro Focus platform. In areas where Veryant does not support specific Micro Focus compiler syntax, the effect of a Micro Focus compiler directive can be obtained using a recommended Veryant compiler option or runtime setting.

The following list maps commonly used Micro Focus directives to the appropriate Veryant alternative:

Micro Focus directive	isCOBOL directive
ANIM	-d
APOST	-apost
ASSIGN(EXTERNAL)	-cax
COPYEXT(cpy)	-ce=cpy
DEFAULTBYTE(00)	-dv=0
IBMCOMP	-dcmi
PERFORM-TYPE(OSVS)	-pt2
REMOVE(NULL)	-rw=NULL
NOTRUNC	-dz

Data access

Veryant's innovative and modular design approach supports a wide range of data access options. All common COBOL file organizations and record types are supported by Veryant, including indexed, relative, binary sequential, and line sequential, with fixed and variable length records. Veryant's COBOL compilers have a built-in ESQL compiler that translates embedded SQL statements into JDBC calls. Veryant software also supports database-specific pre-compilers such as Oracle Pro*COBOL and DB2.

Most data files will need to be converted to an equivalent Veryant format before they can be used. A data migration utility is included with Veryant COBOL runtime environments.

Sequential files

Micro Focus and Veryant employ the same techniques to determine record lengths, so neither fixed length nor variable length sequential files require conversion.

Relative files

Relative files are converted with a utility provided by Veryant support.

Indexed files

Indexed files can be easily converted to Veryant™ JISAM, a 100% Java-based ISAM file system. Veryant JISAM includes a file-management utility called JUTIL that has an option for automatically migrating Micro Focus indexed data files into the Veryant JISAM file format.

Relational databases

No data migration is required for relational databases when moving to a Veryant COBOL platform -- current databases can continue to be accessed through existing interfaces without COBOL code changes. Veryant supports access to any database that offers a JDBC

driver or a COBOL/ESQL pre-compiler. Databases can be accessed through standard COBOL file I/O statements or by including embedded SQL in the COBOL source code.

External file systems

Veryant technology can be configured to use external file systems that contain indexed, relative or sequential files via an EXTFH (external file handler) interface. Developers can set the property, "extfh.libname" to specify the name of the EXTFH library. I/O statements remain the same in the COBOL source whether using native or EXTFH file system libraries.

File status codes

To further simplify the transition to Veryant, developers can configure the runtime to generate the same file status codes used by Micro Focus COBOL. This is accomplished by setting the property "file.status=com.iscobol.io.FileStatusMF".

User interfaces

Veryant supports a range of user interface options and access methods. The majority of user interfaces commonly associated with Micro Focus COBOL move smoothly to a Veryant COBOL platform with a basic recompile.

For standalone COBOL applications with a GUI implemented in the COBOL language itself, isCOBOL technology is typically deployed. The remainder of this section focuses on isCOBOL user interface capabilities.

Character-based user interfaces

The isCOBOL platform enables character-based user interfaces to be deployed on dumb terminals and terminal emulators, locally on a graphical Linux, Unix or Windows desktop, or by using isCOBOL software's thin client technology. isCOBOL™ Application Server offers a thin client character or graphical user interface implemented with Java Swing, and is thereby completely portable. The isCOBOL platform also offers numerous compile options and properties to fine tune appearance, including fonts and colors.

isCOBOL supports the following character-based user interface methods:

- Simple ACCEPT and DISPLAY
- Micro Focus enhanced ACCEPT and DISPLAY
- Windowing Syntax
- SCREEN SECTION

Screen Section code comes over as is and isCOBOL offers additional controls and properties to update and modernize such user interfaces.

The following user interface methods require transformation in order to be migrated to the isCOBOL platform:

- Dialog system for character user interfaces
- Panels
- COBOL System Library Routines (e.g. CBL_WRITE_SCR_CHATTRS)

Graphical user interfaces

The isCOBOL platform delivers innovative, next-generation graphical user interface (GUI) programming capabilities directly in the COBOL language, no need to learn .NET or any other language. Organizations updating legacy COBOL applications with modern Web 2.0 user interfaces can build dynamic GUIs with the graphical isCOBOL™ Integrated Development Environment (IDE). Developers can also transform a green-screen into a GUI or add JavaBean graphical controls directly in the COBOL language.

The following graphical user interface methods require transformation when moving to the isCOBOL platform:

- Graphical User Interface API
- Panels V2 (Version 2)
- Dialog system for graphical user interfaces
- .NET programming using WinForms or WebForms

Similar to Micro Focus COBOL, isCOBOL supports third-party screen I/O packages.

Function keys

The isCOBOL platform supports all function keys. Slight code changes may be required to respond as desired. For example,

In Micro Focus COBOL:

```
special-names.  
    crt status is key-status.  
  
    01 key-status.  
        03 key-type      pic x.  
        03 key-code-1   pic 9(2) comp-x.  
        03 key-code-2   pic 9(2) comp-x.  
  
accept data-item at 0101  
if key-type = "1"  
    if key-code-1 = 1  
        display "F1 was pressed"  
    end-if  
end-if.
```

In isCOBOL:

```
special-names.  
  crt status is key-status.  
  
  01 key-status pic 9(4).  
  
  accept data-item at 0101  
  on exception  
    if key-status = 1  
      display "F1 was pressed"  
    end-if  
  end-accept.
```

Interoperability

Programs compiled with a Veryant COBOL can call external language routines without the need for code changes. Runtime framework properties are used to point to the external libraries.

Library routines

Veryant implements many library routines that provide equivalent functionality to the Micro Focus “call by name” routines (CBL_*).

The following Micro Focus library routines are supported by Veryant:

CBL_AND

CBL_COPY_FILE

CBL_CREATE_DIR

CBL_DELETE_DIR

CBL_DELETE_FILE

CBL_ERROR_PROC

CBL_EXIT_PROC

CBL_NOT

CBL_OR

CBL_XOR

Where necessary, Veryant’s support team can write or assist in writing required subroutines, C routines or Java programs using the equivalent Veryant COBOL syntax.

For example, a COBOL subprogram that could be used to replace the Micro Focus CBL_CHANGE_DIR routine would be:

```
IDENTIFICATION DIVISION.  
    PROGRAM-ID. CBL_CHANGE_DIR.  
ENVIRONMENT DIVISION.
```

```
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 STATUS-CODE          PIC S9(9) COMP-4.  
LINKAGE SECTION.  
01 LNK-PATHNAME        PIC X ANY LENGTH.  
PROCEDURE DIVISION USING LNK-PATHNAME.  
MAIN.  
    CALL "C$CHDIR" USING LNK-PATHNAME, STATUS-CODE.  
    GOBACK STATUS-CODE.
```


Phased transitions

With Veryant software compiler and runtime compatibility settings, the overall level of effort required to move from Micro Focus COBOL can be kept to a minimum. Some organizations still elect to undergo a phased migration to further minimize business disruption when migrating to Veryant.

During a phased transition, developers continue to maintain an application with the Micro Focus COBOL compiler and to support current customers under the Micro Focus platform, while simultaneously working on the same set of source code in isCOBOL. If specific features are desired for use with a Veryant platform, conditional compilation options are employed and a check is performed at runtime to determine which subprogram to call.

Some of the Veryant features that make phased transition possible include:

- The compiler operates as a command-line utility, like Micro Focus COBOL. So current build scripts can be used by changing only the compiler executable name and command-line options.
- Veryant software's compatibility options for proprietary Micro Focus brand features.
- Veryant compiler support for conditional compilation with Micro Focus style syntax, allowing the preservation of current COBOL syntax and behavior.
- Runtime environment support for checking at run-time to determine whether the execution environment is set for Veryant or Micro Focus COBOL. This ability to do dual-coding within the same program allows you have a single set of source code for your application, even as you convert.

Overall, Veryant technology is stricter in some areas about ANSI COBOL standards than other compilers and might therefore catch new programming errors. Once an application's standards-compliance has been improved, the code will compile cleanly with Micro Focus COBOL as well as with Veryant.

Conclusion

Veryant's innovative COBOL technology is highly compatible with today's common COBOL dialects, enabling quick and efficient migration from Micro Focus COBOL. Whether an organization is focused on evolving existing applications through SOA, migrating from mainframes to open systems, improving quality and distribution processes, or simply improving the bottom line, Veryant software can increase business productivity and operational effectiveness.

When considering a move to Veryant, our support team can help assess the level of effort required. If desired, Veryant also offers services to build a limited representative prototype of an application running in a Veryant environment. This exercise includes a thorough analysis process that helps determine how straightforward the transition to Veryant will be for an organization.

For more information

For more information on migrating to Veryant, visit us online at www.veryant.com or email info@veryant.com.