Micro Focus Visual COBOL 2.1 Development Hub

Release Notes

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
http://www.microfocus.com

2012-09-18

Contents

Micro Focus Visual COBOL Development Hub Release Notes

These release notes contain information that might not appear in the Help. Read them in their entirety before you install the product.

Note: This document contains a number of links to external Web sites. Micro Focus cannot be responsible for the contents of the Web site or for the contents of any site to which it might link. Web sites by their nature can change very rapidly and although we try to keep our links up-to-date, we cannot guarantee that they will always work as expected.

Installation

Note: If you are installing on Solaris, please read *UNIX Installer Issues* first.

System Requirements for Visual COBOL Development Hub

Hardware Requirements

The disk space requirements are:

- Between 26 and 33 MB for the Sentinel RMS license server depending on the platform.
- Between 206 and 427 MB for Micro Focus Development Hub depending on the platform.

Note: The installation requires extra disk space that equals the size of the product you install.

Operating Systems Supported

- POWER running AIX 6.1, 6.6, 7.2 - 32/64-bit
- x86-64 running Red Hat Linux 5.5, 5.8, 6.2, Oracle Linux 6.2 (Red Hat Kernel compatibility mode) - 32/64-bit
- Oracle Linux 6 Update 2 with Unbreakable Enterprise Kernel Release 2
- SPARC running Solaris 10, 11 - 32/64-bit
- x86-64 running SuSE SLES 11, 11 SP2, Oracle Linux 6.2 (Red Hat Kernel) - 32/64-bit
- 390 running SuSE SLES 11 SP1 - 32/64-bit
- Itanium running HP/UX 11.31 - 32/64-bit

Software Requirements

Before installing this product, you must have the following software installed on your computer:

- Before installing on Red Hat 6.1, you must have the 32-bit operating system libraries installed:

  GNU Standard C++ Library - libstdc++(i686 version)
  The object files for development using standard C libraries – glibc-devel (i686 version)

  Check the *Red Hat Web site* for more information.
- To use the Web installer on Red Hat Enterprise Linux 6.1, you must have the following bug fix updates for Red Hat installed:

  glibc-2.12-1.25.el6_1.3.i686.rpm
  openldap-2.4.23-15.el6.i686.rpm
  nss-pam-ldapd-0.7.5-7.el6.i686.rpm
  zlib-1.2.3-25.el6.i686.rpm
  nss-3.12.9-9.el6.i686.rpm
  nss-util-3.12.9-1.el6.i686.rpm
  cyrus-sasl-lib-2.1.23-8.el6.i686.rpm

You do not need these updates if you use the full product setup file to install the product.

1. Java Platform Standard Edition (Java SE) 6 Update 27 or Java 7 is required to execute COBOL JVM code and for native COBOL and Java interoperability.

   Note: On AIX 6.1 v2, the minimum required version is Java 6.0 SR10.

   You can download Java SE from *www.oracle.com* and install it anywhere on your machine.

- You need to install Xterm, the terminal emulator for the X Window System. Xterm is part of your Linux/UNIX distribution but is not installed by default. Use your Linux/UNIX installation media to install it.

Important: This release requires version 10000.2.990 or later of the Micro Focus licensing software. For local servers, you do not need to install it separately, as the setup file installs a new Visual COBOL client and a new licensing server on the same machine.

If you have a network server, you must update the license server before installing the product as the client is not able to communicate with license servers of versions older than 10000.2.660. To check the version of the license server on UNIX, run `/var/microfocuslicensing/bin/mfcesver` or `/var/microfocuslicensing/bin/cesadmintool.sh`.

You can download the new version of the license server software from the Micro Focus SupportLine Web site: *http://supportline.microfocus.com/websync/SLM.aspx*.

Before you start the installation, you need to set the environment as follows:

- Set the JAVA_HOME environment variable. When installing the product, set this variable to a 32-bit Java installation or the installation terminates. For example, execute the following:

  ```
  JAVA_HOME=java_install_dir
  ```

  where *java_install_dir* is the path to the JAVA installation directory such as `/usr/java/javan.n`
- Add `$JAVA_HOME/bin` to your system PATH variable. To do this, execute:

  ```
  export PATH=$JAVA_HOME/bin:$PATH
  ```
- Set the LANG environment variable to pick up localized messages. The LANG settings are English and Japanese only.

Installing Visual COBOL Development Hub

Downloading the Product

1. Use the download links in your Electronic Product Delivery email.

   For more information follow the links for the install instructions and the End User License Agreement.

Installing

To use the Web Installer:

1. Give the Web installer file execute permissions as follows:

   ```
   chmod +x webinstaller_visualcobol_devhub_2.1_platform
   ```
2. Run the installer with superuser permissions:

   ```
   ./webinstaller_visualcobol_devhub_2.1_platform
   ```

   If you don't run this as superuser you will be prompted to enter the superuser password during the installation.

Alternatively, you can use the setup file and install the product as follows:

1. Give execute permissions to the setup file:
   ```
   chmod +x setup_visualcobol_devhub_2.1_platform
   ```
2. Run the installer with superuser permissions:
   ```
   ./setup_visualcobol_devhub_2.1_platform
   ```

   If you don't run this as superuser you will be prompted to enter the superuser password during the installation.

   The COBOL environment is installed by default into `/opt/microfocus/VisualCOBOL`.

   To install in a different location use the `-installlocation="Location"` parameter to specify an alternative directory location. For example:
   ```
   ./webinstaller_visualcobol_devhub_2.1_platform -installlocation="full path of new location"
   ```

   or
   ```
   ./setup_visualcobol_devhub_2.1_platform -installlocation="full path of new location"
   ```

   You can see details about which additional parameters can be passed to the install script if you enter the `-help` option.

   Note:

   - The installation of this product could affect the SafeNet Sentinel licensed components running on your machine. During installation licensing is shutdown to allow files to be updated. To ensure the processes running on your machine are not affected, you need to use the –skipsafenet option, which skips the installation of SafeNet:
     ```
     ./setup_visualcobol_devhub_2.1_platform -skipsafenet
     ```
   - To protect the SafeNet Sentinel installation from accidental updating you can create an empty file named `SKIP_SAFENET_INSTALL` in `/var/microfocuslicensing/` as follows:
     ```
     touch /var/microfocuslicensing/SKIP_SAFENET_INSTALL
     ```

     While the file is present, the SafeNet installer does not make changes to the installation or shutdown the running license daemons. If later licensing needs to be updated, remove the file and install Sentinel RMS server manually.

Set Up the Environment

When you have installed the product, you need to set the environment as described below.

1. To set up your product, execute:
   ```
   /opt/microfocus/VisualCOBOL/bin/cobsetenv
   ```
2. To verify that your product is installed, execute:
   ```
   cob -Version
   ```

Important: These commands set the environment only for the current shell. You need to execute them for each new shell that you start.

To avoid having to run cobsetenv for every shell, add these commands to the shell initialization files (`etc/profile`, `etc/bashrc`, etc.)

Note: For information about the Visual COBOL Development Hub, check the help for Visual COBOL for Eclipse that is available online on the *Micro Focus Infocenter*.

Installing as an Upgrade

This release works concurrently with version R4 of Visual COBOL, so you do not need to uninstall it. There are two options for installing the latest version in this case:

- Move the R4 installation to a different location and install the latest version to the default install location, `/opt/microfocus/VisualCOBOL`.

  This ensures you do not need to change your environment. To move the existing older installation to a different location:

  1. Execute the following command:
     ```
     [ as root ] mv /opt/microfocus/VisualCOBOL /opt/microfocus/VisualCOBOLR4
     ```
  2. Install the latest version as described in the section *Installing*.
- Install the latest version in a different location and set the environment to point to it. To do this, run the Visual COBOL 2.1 installer with the –installlocation option:

  1. Execute the following command:
     ```
     ./InstallFile -installlocation="/opt/microfocus/VisualCOBOL2.1"
     ```
  2. Execute cobsetenv to set the environment and point to the new install location:
     ```
     . /opt/microfocus/VisualCOBOL2.1/bin/cobsetenv
     ```

UNIX Installer Issues

License Infrastructure Installer

On some Solaris platforms, you can receive the following error message when SafeNet license server needs to be installed or upgraded on your machine:

```
tar: /safenet.tar: No such file or directory
```

To resolve this issue, wait for the installation to complete and then perform the following:

1. Navigate to the `safenet` directory in the COBDIR location.
2. With superuser permissions execute: `./MFLicenseServerInstall.sh`

License Server

On UNIX, you need to configure the computer hostname to ensure the license server will start properly.

To avoid performance issues, "localhost" and the computer hostname must not both be mapped to IP address 127.0.0.1. You should only map "localhost" to IP address 127.0.0.1.

The following is an example of how to specify these entries correctly in the etc/hosts file:

```
127.0.0.1 localhost.localdomain localhost
IP machinelonghostname machineshorthostname
```

where *IP* is the unique IP address of the computer in xx.xx.xx.xx format.

Configuring the Remote System Explorer Support

The remote development support from the Eclipse IDE relies upon Visual COBOL Development Hub running on the UNIX machine and handling all requests from the IDE for building and debugging programs. Visual COBOL Development Hub provides a UNIX daemon, the Remote Development Option (RDO) daemon, which initiates the RDO as Eclipse clients connect to it. Whichever environment is used to start the RDO daemon will be inherited for all servers and hence all build and debug sessions.

Starting the Daemon

Important: Before starting the daemon you must have the following on your UNIX machine:

- a version of Perl
- a version of Java

- the `as` (assembler) and `ld` (linking) programs on the path, as specified by the PATH environment variable

To start the daemon on the default port (4075) as a background process, perform this command with superuser authority:

```
$COBDIR/remotedev/startrdodaemon
```

The daemon will now listen for any Eclipse client processes connecting to that machine on port 4075. If you want to use another port, then specify another port number on the `startrdodaemon` command.

The daemon can also be configured to instantiate the servers on a specified port or range of ports. This is particularly relevant when you want to only open certain ports through a firewall. To do this, perform this command with superuser authority:

```
$COBDIR/remotedev/startrdodaemon [<port> | <low port>-<high port>]
```

where:

- <*port*> is the port number the daemon should use to listen for connections from Eclipse on the client machine. If no value is given, it will be assigned a default value of 4075. This value matches the value assigned within the Eclipse installation.

  For example,

  ```
  $COBDIR/remotedev/startrdodaemon 4999
  ```

  This command will start a daemon listening on port 4999 and will use random server ports.
- <*low port*>-<*high port*> is the range of ports on which the servers (launched by the daemon) should use to communicate with Eclipse on the client machine.

  For example,

  ```
  $COBDIR/remotedev/startrdodaemon 4080 4090-4999
  ```

  This command will start a daemon listening on port 4080 and server ports will be in the range 4090 to 4999.

Stopping the Daemon

To stop the daemon, type the following command (with superuser authority):

```
$COBDIR/remotedev/stoprdodaemon <port>
```

Configuring the Environment

You may need to configure some aspects of the environment before you start the daemon. This is because when a build or debug session is initiated on the Development Hubfrom one of the Eclipse clients, the environment used will be inherited from whatever was used to start the daemon. A typical example of the kind of environment that might need to be set up would include database locations and settings for SQL access at build/run time.

Repairing

If a file in the installation of the product becomes corrupt, or is missing, we recommend to reinstall the product.

Uninstalling

To uninstall this product:

1. Execute as root the `Uninstall_VisualCOBOL2.1.sh` script in the `$COBDIR/bin` directory.

Note: The installer creates separate installations for the product and for Micro Focus License Manager. Uninstalling the product does not automatically uninstall the Micro Focus Licensing Manager or the prerequisite software. To completely remove the product you must uninstall the Micro Focus Licensing Manager as well.

To uninstall Micro Focus License Manager:

1. Execute as root the `UnInstallMFLicenseServer.sh` script in the `/var/microfocuslicensing/bin` directory.

   The script does not remove some of the files as they contain certain system settings or licenses.

You can optionally remove the prerequisite software. For instructions, check the documentation of the respective software vendor.

Licensing Information

Note:

- This release uses the license keys for the Visual COBOL R4 release.
- This release requires the latest version of SafeNet licensing software. See *Software Requirements* in this document for more details.
- If you are unsure about what your licensing policy is or what sort of license you require, consult your System Administrator or Micro Focus SupportLine to obtain a valid license.

UNIX The Micro Focus Licensing System is installed into the `/var/microfocuslicensing` directory.

To license your software:

1. Run the Micro Focus Licensing Administration utility:
   ```
   sh /var/microfocuslicensing/bin/cesadmintool.sh
   ```

   Note:

   - You need permissions to write to the license file which normally means you need to log in as root.
   - Make sure Java is on the PATH before you run the utility. See *System Requirements* for more information.
2. Select an option as appropriate and press **Enter**:

   If you have access to the Internet:

   Select **Online Authorization**. You will be prompted to enter the Authorization Code supplied with your delivery notice.

   If you do not have access to the Internet:

   You need the license strings. In order to obtain them, you need your Authorization Code, your Machine Id and a machine with access to the Internet.

   a. Choose **Get Machine Id** and press **Enter**.
   b. Make a note of the information which is displayed.
   c. On the machine which is connected to the Internet, open *http://supportline.microfocus.com/activation/* in a browser.
   d. Follow the instructions to obtain the license strings and save them to a text file.
   e. Copy the text file to the first machine on which you are installing the product.
   f. At the command line, select **Manual License Installation** from the licensing options and enter the full name of the file containing the license strings.

What's New

The following sections outline the new features that have been added in this release of Visual COBOL.

New Features in Visual COBOL 2.1

ACUCOBOL-GT Data Types in Managed Code

ACUCOBOL-GT data types and sign() variants that were previously only available in native code are now supported in managed code. Use the Compiler directives COMP1 and COMP2 to set ACUCOBOL-GT behavior for those particular data types.

ACUCOBOL-GT Library Routines in Managed Code

ACUCOBOL-GT library routines that were previously only available in native code are now supported in managed code.

Compiler Directives

The following new Compiler directives are now available

- DISPLAY - Defines the default behavior of standard DISPLAY statements.
- COMP1 - Specifies the behavior of a COMP-1 data item.
- COMP2 - Specifies the behavior of a COMP-2 data item.
- RESTRICT-GOTO - Generates a syntax error for GO TO statements that transfer control to outside of the current section.
- ILSMARTRESTRICT - Limits the generation of properties in ILSMARTLINKAGE classes to non-redefining elementary items.

The following Compiler directives have changed:

- DATAMAP - Two new parameters allow you to display either the address or offset values for data items in your program.

`.int`, `.gnt` and `.lbr` File Types Support

Support has been added within the IDE for compiling native COBOL applications to the Micro Focus legacy formats `.int` and `.gnt`, and to package these files as a Micro Focus library file (`.lbr`). Improvements include:

- An option to compile all native COBOL projects to `.int` and `.gnt` code. You can set this in your project's properties.
- An option to package the `.int` and `.gnt` files produced by the project as a Micro Focus `.lbr` library files.
- Improvements to the Net Express Project Import wizard that enable you to convert existing Net Express projects to Visual COBOL projects that compile to `.int` and `.gnt` code.

JVM Class Library

A new class library, `com.microfocus.cobol.runtimeservices`, has been introduced to help you integrate JVM COBOL and Java in the same application.

The library contains the following classes:

CallableProgram	Provides a class and annotation that you can use to create a class that can be called from COBOL.
Interop	Provides a class that contains helper methods for loading, calling and cancelling a program.
RunUnit	Provides a container class that allows you to use COBOL programs in a multi user/session environment.

Full information on these classes is available in Javadoc format for integration in the Eclipse IDE.

Managed COBOL Enhancements

Delegates and Events

Delegates and events are now implemented on the JVM platform.

This release provides support for combining delegates, using the METHOD keyword to specify method groups, and implicit conversion from a method group or an anonymous method to the suitable delegate type.

Handling Invalid Numeric Data

The handling of invalid numeric data is controlled by a number of Compiler directives: HOSTNUMMOVE, HOSTNUMCOMPARE and SIGNFIXUP. These directives were previously only available in native code but are now supported in managed code.

Resolving Types

In this release, the Compiler attempts to resolve types to those defined in the current compilation unit wherever possible. The Compiler will attempt to resolve such types to an external name only if no suitable type exists in the current compilation unit. For example:

```
$set ilusing"System"
class-id MyNamespace.EventHandler.
01 o type EventHandler.
end class.
```

In this release, `01 o type EventHandler.` resolves to *MyNamespace*`.EventHandler` and not to `System.EventHandler`.

Specifying Properties

In previous versions of the products, properties declared using the PROPERTY keyword on a data item were generated as final properties. Starting with this release, they are generated as virtual properties by default. In order to make the properties final, you need to specify the word FINAL following PROPERTY. This change may affect the generation of Proxy classes, for example, if you are using WCF.

OpenESQL

SQL Compiler Directive Options

OpenESQL has been enhanced to support the the following new SQL compiler directive options:

DATE	Controls the reformatting of date values in output parameters and in input parameter character host variables when DETECTDATE is also specified.

TIME	Controls the reformatting of date values in output parameters and in input parameter character host variables when DETECTDATE is also used.
DATEDELIM	Specifies a single character as the delimiter between the year, month, and day components to override the default delimiter determined by the HCOSS DIALECT or DATE directive specification.
TIMEDELIM	Specifies a single character as the delimiter between the hour, minute, and second components to override the default delimiter determined by the HCOSS DIALECT or TIME directive specification.
TSTAMPSEP	Specifies a single character as the separator between the date and time parts of timestamp and date/time data.

SQL Server We now support Microsoft SQL Server 2012.

Features Added in Visual COBOL 2.0

Compiler Directives

The following new directives are now available:

- COPYSEARCH - enables you to specify how copybooks are located. You can choose between usual Micro Focus COBOL behavior or usual RM/COBOL behavior.
- ILSMARTNEST - enables you to nest ILSMARTLINKAGE classes inside the program class in which they are defined. This makes it possible to have multiple programs in a single compilation unit that include linkage records with the same name.

The following directives have been changed:

- DIALECT(RM) - now accepts a new parameter, RM, which enables the RM-compatible functionality that the RM directive used to enable.
- ILREF - can only specify a .class as a parameter, and not a .jar file or other file types.
- ILUSING - when set on a single file using the SET statement, `$set ilusing`, the directive only affects that file.

JVM COBOL File Handler

Use the JVM COBOL File Handler, a File Handler written in purely JVM COBOL managed code, when you are deploying to environments that do not allow the use of native code such as the default Micro Focus File Handler.

Library Routines

The following CTF library routines are now available in COBOL for JVM:

CBL_CTF_COMP_PROPERTY_GET
CBL_CTF_TRACE
CBL_CTF_TRACER_LEVEL_GET
CBL_CTF_TRACER_GET
CBL_CTF_LEVEL

The following routine has been enhanced:

- The CBL_SEMAPHORE_ACQUIRE routine now accepts a `timeout` parameter.

Managed COBOL Language Features

The following new syntax elements are now available in managed COBOL:

Local Variables	In managed COBOL, Data items can now be declared in the procedure division, using the DECLARE statement. In addition, they can be declared inline as the iterator in a PERFORM statement, or as an exception message in a TRY … CATCH … FINALLY statement block.
Collections	There are two new collection types in managed COBOL: LIST and DICTIONARY. For a LIST, you can add elements to a list, retrieve the nth element of the list, replace the nth element, iterate through the list and clear the list. For a DICTIONARY, you can add key value pairs, retrieve a value corresponding to a key, to replace the value corresponding to a key, iterate through the dictionary and clear the dictionary.
Properties	In managed COBOL, a property can now be defined using PROPERTY-ID and GETTER and SETTER phrases to access to the property. The previous technique of specifying the keyword PROPERTY on a data declaration is still available.
Indexers	In managed COBOL, an indexer can now be defined using INDEXER-ID and GETTER and SETTER phrases to access the indexer value. Indexers are similar to properties, except that their accessors take parameters. Indexers allow instances of a class or valuetype to be indexed just like arrays.
Zero-based Indexing	The managed COBOL syntax for arrays now uses zero-base indexing to access arrays when square brackets are specified. For backward compatibility, one-base indexing is used when round parentheses are specified.

Run-Time Tunables

This release provides the following new tunable:

- subsystem_cancel_mode - use this to override the default cancel mode when you use the CBL_SUBSYSTEM library routine to cancel a subsystem.

Vision Data File Searching

This release provides the following new ACUCOBOL-GT compatible environment variables to help search for Vision data files at run time:

APPLY_FILE_PATH
FILE_CASE
FILE_PREFIX
FILE_SUFFIX

Features Added in Visual COBOL 2010 R4 Update 2

New Platforms Support

Support for Visual COBOL for Eclipse has been added for the following platforms:

- x86-64 running Red Hat Enterprise Linux 5.7/6.1
- x86-64 running SuSE SLES 11 SP1

Support for Visual COBOL Development Hub has been added for the following platforms:

- x86-64 running Red Hat Enterprise Linux 5.7/6.1
- x86-64 running SuSE SLES 11 SP1

Support for COBOL for JVM has been added for the following platforms:

HP IA 11.31 - 32/64-bit
x86-64 running Red Hat Linux 5.6/6.1 - 32/64-bit
SPARC running Solaris 10 - 32/64-bit

OO COBOL Class Library Reference

Help for the OO COBOL class libraries are available from the Micro Focus SupportLine Web site, as follows:

1. Go to the Server Express documentation, at *http://supportline.microfocus.com/documentation/books/sx51ws02/sx51indx.htm*.
2. Click *Reference* > *OO COBOL*.
3. Expand *OO COBOL Class Library Reference*.

Features Added in Visual COBOL 2010 R4

ACUCOBOL-GT Compatibility

The Compiler and run-time continue to provide support for ACUCOBOL-GT. The directive ACU is the main switch for turning on ACUCOBOL-GT compatibility. The ACU directive enables various ACUCOBOL-GT syntax extensions and other language elements. Additional ACUCOBOL-GT compatibility features include the following:

- When using a CALL statement, the USING and GIVING/RETURNING phrases can now appear in either order.
- The following ACUCOBOL-GT standard library routines can now be used with Visual COBOL in native code:
  - C$CALLEDBY
  - C$CALLERR
  - C$CHDIR
  - C$MAKEDIR
  - C$MEMCPY
  - C$MYFILE
  - C$PARAMSIZE
  - C$RERR
  - M$ALLOC
  - M$FREE
  - M$COPY
  - M$FILL
  - M$GET
  - M$PUT
  - WIN$VERSION
- The following ACUCOBOL-GT 'ccbl' compiler options can now be used with Visual COBOL:
  - -E, -V
  - -Cv
  - -Da, -Db, -Dd31, -DL1/2/4/8, -Dq, -FpRounding

- -La, -Li, -Lc, -Lf, -Ll, -Lo, -Ls, -Lw

  Note: The output that these list options provide differs in Visual COBOL.
- -Qm
- -Rc, -Rn, -Rw
- -Sa, -St, -Sd, -Sp, -S1...-S9
- -noTRUNC, -truncANSI, -Dz
- -Td, -Te
- -Vc
- -Za, -Zc, -Zl, -Zn, -Zs, -Zi, -Zr1, -Zy, -arithmeticVSC2

Full ACUCOBOL-GT compatibility is documented under the *Programming* section in the product help.

Embedded HTML

We now support the use of Embedded HTML (EHTML) in COBOL CGI programs, which enables you to output HTML directly from your applications.

Language Improvements

The following improvements have been made to managed COBOL:

Extension methods and extending operators	Managed COBOL now supports extension methods. This feature enables you to add methods to existing types without the need to edit or recompile the code. You can also extend operators.
The SYNC modifier for methods	The SYNC modifier locks the values of the arguments sent to the method, so that they do not change while the method is processing.
Nested classes	In managed COBOL, a nested class can now be defined so that it can access the instance fields, properties and methods in its containing class. To allow this, you add the optional SHARING PARENT phrase to the nested class definition.

RM/COBOL Compatibility

The Compiler and run-time continue to provide support for RM/COBOL. Additional RM/COBOL compatibility features include the following:

- The following RM/COBOL standard library routines can now be used with Visual COBOL in native code:
  - C$Century
  - C$ConvertAnsiToOem
  - C$ConvertOemToAnsi
  - C$DARG
  - C$Delay
  - C$GetEnv
  - C$GetNativeCharset
  - C$LogicalAnd
  - C$LogicalComplement
  - C$LogicalOr
  - C$LogicalShiftLeft
  - C$LogicalShiftRight
  - C$LogicalXor
  - C$NARG
  - C$SetEnv

    - C$RERR
    - DELETE
    - RENAME
- The RM/COBOL file handler can now be used with Visual COBOL, enabled by using the CALLFH(ACUFH) Compiler directive, and then configuring an add-on to the Vision file handler.

Full RM/COBOL compatibility is documented under the *Programming* section in the product help.

XML Extensions

Note: This functionality is supported in native COBOL only.

You can now use XML Extensions, the system that enables your COBOL applications to interact with XML documents, with Visual COBOL.

XML Extensions has many capabilities. The major features support the ability to import and export XML documents to and from COBOL working storage. Specifically, XML Extensions allows data to be imported from an XML document by converting data elements (as necessary) and storing the results into a matching COBOL data structure. Similarly, data is exported from a COBOL data structure by converting the COBOL data elements (as necessary) and storing the results in an XML document.

For more information about XML Extensions, refer to the *XML Extensions User's Guide*, available from the RM/COBOL product documentation set, in the SupportLine section of the Micro Focus Web site.

Known Issues

Adding Projects to a COBOL JVM Project's Java Build Path

If a project you add to the Java build path of a COBOL JVM project has its default output folder set to its root and not a sub-folder, you will get errors when building the COBOL JVM project.

To avoid this problem, make sure set the output folder of the added project is a sub-folder.

Background Parsing

When you work with a remote COBOL project, most of the processing such as program compilation and debugging takes place on the remote server. However, background parsing takes place on your local client machine and this architecture might result in parsing errors due to differences in local and remote environments.

The following are examples of when you might encounter parsing errors:

Using copybooks	The source code environments are different.
Using environment variables	If the project build is using environment variables, it is likely that they are different in the local and remote environments.
Using the Pre-complier	The software environment used in the build is different in the local and remote environments.

These parsing errors do not affect the program build. However, you can remedy the problem depending on the nature of the parsing error. For example, if you encounter parsing errors when using copybooks, you can create a softlink to the copybook on the remote machine.

From a command line prompt on the remote machine do the following:

1. Change to the project directory:
   ```
   cd projectdir
   ```
2. Create a softlink on the required directory or individual file:
   a. On a directory:
      ```
      ln -s $COBDIR/cpylib/ cpylib
      ```
   b. On an individual file:
      ```
      ln -s $COBDIR/cpylib/copybook.cpy copybook.cpy
      ```

Background parsing does not trigger in all cases

This happens only to files that are already opened and are being edited in the **COBOL Editor**. The background parsing is not triggered (in all cases). Sometimes it shows errors that do not exist or hides existing ones.

For example:

1. Open two COBOL classes in the editor (`CobolClass1` and `CobolClass2`. And `CobolClass1` uses an artefact (*A2*) from `CobolClass2`).

2. Delete *A2* from `CobolClass2`.
3. Switch to the `CobolClass1` window in the editor.

   No error will be displayed in the editor until background parser is triggered for `CobolClass1` (when you edit `CobolClass1` or reopen it).

Note: If you start a build the error won't be visible in the **Problems view** and the **Console** but the error will still be visible in the editor until the background parser is triggered.

Co-existing with Earlier Micro Focus Products

Run-time system error due to COBCONFIG

A run-time system error occurs if the COBCONFIG environment variable is set when you run aVisual COBOL application or when you use Visual COBOL to edit or create projects and the configuration file it refers to contains entries that are not valid for Visual COBOL.

For example, this might happen if you have Studio Enterprise Edition installed and COBCONFIG is set for it.

To work around this issue, ensure that Visual COBOL is not running and then modify the configuration file by doing one of the following:

- If the invalid tunable is not needed by another application, remove it from the run-time configuration file.
- Add the following as the first line in the configuration file:

  ```
  set cobconfig_error_report=false
  ```
- Unset COBCONFIG or set it to another configuration file that does not contain the invalid tunable for the particular session you are running in.

Compiling JVM COBOL Applications

Compiling with Existing Classes on a Classpath

If a compilation unit includes a class, and a version of the class already exists on the classpath, the results are undefined.

To avoid this problem, either keep a separate runtime and compilation classpath, or delete built artifacts before compilation.

Managed File Handler

The execution of managed JVM COBOL applications compiled for 64-bit may fail on 64-bit platforms with high bitism, such as HP or AIX.

To work around this issue, you need to compile your code for 32-bit platforms.

Debugging in Eclipse

- If you delete a project while you are debugging it, the delete action will only be partially successful and errors will occur. You must stop the debug process before deleting a project.
- If you import the COBOL JVM /Extension methods demo, set a breakpoint and start debugging, the main source is found but when you step into twice you get a report that the source for `StringExtension.cbl` cannot be found (even though it is located in the same directory as the main source and is on the source lookup path.

Error Markers

The IDE does not set correctly the error markers in files that are not associated with the Micro Focus COBOL project types. Instead of setting the markers on the exact lines that cause a problem, the IDE sets all of them on the first line of the respective files. To work around this issue, you need to add the unknown file extensions or the file names as file associations for the COBOL project types. To do this, click **Window > Preferences**. In the left-hand pane, click **General > Content**. Expand **Text > Micro Focus COBOL Project Types**. Add the file extensions in the **File association** pane.

JVM COBOL Support

This release includes the ability to edit, debug, and compile COBOL applications to JVM byte code (.class files) so that they can be run on a Java Virtual Machine. The following limitations and restrictions apply:

- In JVM_LOAD_NATIVE, the `myLibrary` parameter must be passed as a literal; it cannot be declared in a COBOL storage section.
- You cannot use the COBCONFIG or COBCONFIG_ environment variables. You must use COBCONFIGJVM instead.
- The format of a JVM tunables file must be that of a standard Java properties file and not the format used by the native runtime.
- You cannot use XML, delegates, or generics in your JVM COBOL code.
- COBOL class package and the directory structure is enforced. You might get errors finding source code.

  For example:

  If a `COBOLClass1.cbl` is within a package `com.microfocus.core` then it should be in the `Project/src/com/microfocus/core/COBOLClass1.cbl` directory. In the case the `com.microfocus.core` package is mapped to `com/microfocus/core` directory.
- You cannot pass a JVM procedure-pointer to a native COBOL application.
- There is a known issue with IBM's JVM support for the Java™ Attach API which causes the API to leak semaphores under certain conditions. If you are running IBM's JVM on AIX and Redhat or Suse Linux on z390, this can cause your machine to run out of semaphore space resulting in the machine running slower or causing intermittent failures.

  To determine whether you are experiencing this issue, type the following at the command line to check the number of semaphores that start with the 0xa1 prefix:

  ```
  "ipcs -s -r | grep 0xa1 | wc –l"
  ```

  If this number is very high and keeps growing you need to follow the cleanup procedure described in the "IBM Developer Kit and Runtime Environment Diagnostics Guide", "Chapter 19. Attach API problem determination", ftp://public.dhe.ibm.com/software/dw/jdk/diagnosis/diag50.pdf

  You can disable the Java™ Attach APIs to avoid the issue. Note, however, that when Attach API is disabled, it will not be possible for other Java application to attach to your application and features such as jtop, jstat, jconsole, JMX agents, JVMTI agents will cease to function.

  The Attach API is enabled by default for Java 6 SR6 and later. To disable the feature you need to set the IBM JVM property com.ibm.tools.attach.enable to "no".

  You can use this property with any Java trigger, or with IBM's environment variable IBM_JAVA_OPTIONS which allows you to pass extra arguments to the JVM without changing any command lines.

  You can set the variable as follows:

  ```
  export IBM_JAVA_OPTIONS=-Dcom.ibm.tools.attach.enable=no
  ```

Leaking File Handles

The startrdodaemon daemon leaks file handles on every RSE connect and disconnect. As a result, the number of file handles increases until the limit for a process in reached and an error is reported.

It is recommended that you increase the number of file descriptors per process. Consult your operating system documentation for guidelines on how to do this.

Remote Debugging

- If, when you try to start debugging a remote application, you are logged on the remote machine (using connectremote) and X server is running, the debugger may encounter an error launching the application and return Debugger Error Code: -1. This may occur if you have started the RDO daemon from a user session on your hub (remote) machine.

  To work around this issue, you need to do the following:

  - Stop the startrdodaemon daemon and disconnect the RSE connection in Eclipse.
  - Either stop X server on the remote machine or establish a new connection to the remote machine using SSH.
  - Start the startrdodaemon daemon and debug the remote application.

Remote Development using Remote System Explorer

The following problems can occur when you are developing a project using Remote System Explorer (RSE) to browse the file system of a remote machine.

- You might get poor performance of RSE if you have the Web Standards Tool (WST) Eclipse plug-in installed.
- If you close Eclipse while a remote project is open, the following errors occur when you restart Eclipse: Parent Error: Workspace restored, but some problems occurred. Error1: Could not read metadata for '<remote project name>'.
  When this happens you must reopen the project.

Unsupported Features

Run-time Features

The run-time features of your COBOL development system not supported include:

- GUI Class Library
- HyHelp
- Panels Version 2

Note that some of these features might compile correctly, but won't run.

Dialog System (Character and GUI) applications are supported if you install the following AddPacks for Visual COBOL - Dialog System GUI AddPack, Visual COBOL Dialog System Character Mode. You can download the AddPacks from the *Product Updates* section on the Micro Focus SupportLine Web site.

Micro Focus Vision

- Starting the Vision utilities `vutil`, `vio`, and `logutil` installed in `/opt/microfocus/VisualCOBOL/bin/vutil` might fail with an error similar to "vutil: symbol lookup error: /opt/microfocus/VisualCOBOL/lib/libcobacme64.so: undefined symbol: cobme". This is an issue with a common shared library. To work around this problem you need to use the same utilities from the installation of ACUCOBOL-GT Extend release.

Editing Remote JRE Settings

If you are developing a remote COBOL JVM project, problems can occur if you use a remote JRE and change its settings in the project's JVM Build Path dialog box.

Resolved Issues

The resolved issues that customers have reported are listed in this section. The numbers that follow each issue are the Reported Problem Incident number followed by the Customer Incident Numbers (in parentheses). RPIs that have numbers only (and no text) are included to confirm that the RPIs have been fixed, since no further information is required.

Compiler

- The DBCSSOSI directive now compiles correctly.

  590357 ( )
- DISPLAY MESSAGE BOX ... RETURNING syntax now returns an E-level error message stating that it is unsupported in this COBOL system.

  584600 ( )
- You can now configure the DATAMAP clause to output either addresses or offsets. DATAMAP(ADDR) or DATAMAP with no option shows addresses. DATAMAP(OFFSET) shows offsets. Compile managed COBOL code sets DATAMAP(OFFSET) at the end of directive processing as addresses are not available for managed code.

  1054182 (2093752)
- Compilation of ACU DISPLAY windowing syntax will no longer receive spurious internal error messages during a background syntax check.

  591227 ( )
- Compiling with a large number of ADDSYN directives specified now works as expected.

  1083917 (2567767)
- The maximum value for the PAGE LIMIT clause has been increased from 999 to 9999.

  1084391 (2576431)
- DISPLAY(CONSOLE) is a new directive available to override the behavior set by RM or ACU directives, so that standard DISPLAY syntax is processed as standard ANSI DISPLAYs, directed to CONSOLE. RM and ACU directives set DISPLAY(CRT) immediately, allowing you to set DISPLAY(CONSOLE) after either of those directives, if required.

  1084496 (2577799)

- A compiler bug that produced illegal int-code for "call ... returning pointer-item" under DIALECT(RM) has been fixed.

  1085375 (2586367)

Documentation

- When entering the path of the IBM MQ libraries for the ES_MQ_LIB and ES_MQ_LIB_XA environment variables, in AIX environments the library must be an object inside a shared object.

  1084692 (2578515)

File Handling - External File Handler

- Reading IDXFORMAT"9" records non-transactionally over fileshare no longer causes any issues.

  1083646 (2566578)
- When opening a mainframe file for input, the file is now not optional as per the mainframe.

  1082362 (2506971)
- When attempting to OPEN I/O a read-only file using the RM/COBOL File Handler, a 37, 07 error message is displayed if the program is running in ANSI85 mode. The error message is not displayed if the program is running in ANSI74 mode.

  1084329 (2573607)
- Cobfhrepro now works correctly when session id is specified.

  590271 ( )
- Invalid XML syntax in MF.MFFH.XML has been corrected.

  589406 ( )
- The maximum field length that MFSORT supports for PD summary fields is extended from 9 bytes to 18 bytes.

  1084961 (2581353)

File Handling - Fileshare

- If you are accessing RM/COBOL files through Fileshare, the RETRYLOCK option is now working correctly.

  1084744 (2579930)
- Closing a file using FSVIEW now correctly removes the file from fileshare's open file table.

  587706 ( )
- The FSVIEW option 'stats get' gives the statistics for current users, peak users, file opens and peak file opens. The corresponding FSVIEW API is FSV-C-get-stats.

  1083664 (2567737)
- FSVIEW will no longer process the last command in a command file twice.

  589764 ( )

File Handling - Sort / JCL Sort

- The informational "Operand 'VLSHRT' Ignored" message has been removed from the sort sysout.

  1083225 (2562501)
- Using SYMNAMES sometimes caused SORT to use the wrong field position and lengths.

  1083320 (2563738)
- Data format FI is now supported in the OUTREC edit fields section.

  1083495 (2565051)

- SORT now works correctly when there are concatenated VB files with different record lengths in SORTIN.

  1084194 (2573990)
- SORT now gracefully handles the error when there is an INCLUDE/OMIT condition with invalid HEX/Binary digit.

  1084231 (2574465)
- SORT now ignores the operand 'WORK' along with its value.

  1081023 (2533397)
- A sort using SORTTEMPSPACE no longer ever results in a COBRTS 252 error.

  1081943 (2546898)
- CENTWIN and Y2PAST are now supported as PARMS.

  1080717 (2531364)
- The MFSORT help screen now shows that OPTION is supported.

  590260 ( )
- SORT worked incorrectly for multiple fields to be converted in INREC/OUTREC/OUTFIL OVERLAY syntax.

  1084481 (2577157)
- The TOTAL field length calculation now works correctly when TOTALs are zeros.

  1082887 (2558079)
- Trailers are now included in the SYSOUT outfil record count.

  1084193 (2573272)
- Records following HEADER2 and HEADER3 will now be the correct length.

  590270 ( )
- Header lengths are now calculated correctly, taking line feeds into account instead of giving a SORT099I error. Headers will now be printed for outfiles that do not have any records.

  1083701 (2565909)
- SORT now works correctly when SYSIN is given as LSEQ PDS Member.

  1082529 (2550678)
- When using MFJTOOL with a VB input file, an appropriate error is thrown when the output file is not VB or has not worked successfully when the output file is VB. Previously, a COBRTS 139 error was thrown.

  1084567 (2577626)
- Sort now works correctly when OPTION COPY is given before the SORT FIELDS in SYSIN.

  1083679 (2565053)
- SORT caused different sort processing for large sort cards. This has been fixed so SORT now returns an error message and exits the sort processing when the sort card has more than 1024 INCLUDE/OMIT conditions.

  1084065 (2571660)
- When using MFSORT, smaller records are padded to the size of the minimum record length of a VB file and the record length is changed to the minimum of sortout.

  1082943 (2554482)
- Sort now displays HEADER2 at the start and TRAILER2 at the end of each page.

  1083702 (2566545)

JVM Compiler

- Java verification errors or incorrect program flow occurring in very large JVM COBOL programs is now fixed. Previously, in very large programs, the compiler would process GO TO statement incorrectly, and could jump to the wrong program locations.

  1084569 (2578716)
- Some problems with the use of nested classes defined in COBOL have been resolved.

  1083183 (2561420)

JVM Run-Time System

- A stack overflow no longer occurs if you execute a call to a method inside a constructor and this method contains a COBOL statement. Previously, the runtime would jump back to the constructor and go into a loop, producing a stack overflow.

  1083181 (2561418)
- The JVM COBOL library routines CBL_DIR_SCAN_ now return the correct results when searching for directories.

  590261 ( )

MVS REXX Emulation

- The DATE() function now formats the year correctly when converting between a Julian and a standard date format.

  1083437 (2564240)
- The LISTDSI external function now correctly sets values for the SYSREFDATE and SYSMGMTCLASS variables.

  1084902 (2581611)

Run-Time System

- The call MVS_REGISTER_DDNAME is an MFE-only call used by the IDE. FileHandler no longer calls this routine.

  1083028 (2559292)
- IF NUMERIC validation of COMP-3 slack nibbles (when there is an even number of digits in the picture clause) is now done in a way that is compatible with that on the mainframe.

  1083801 (2569340)
- The RUN program now accepts command line arguments of up to 1023 characters long.

  1083215 (2561567)
- When running a full-screen application inside a terminal emulator on Linux, the actual size of the terminal is read at startup and reread when the terminal is resized. This behaviour is also supported on AIX, HP/UX, and Solaris. The Micro Focus vt220 terminfo entry now correctly describes a 24-line display. A vt220-25 terminfo entry is included for compatibility with the previous behaviour.

  1084817 (2579335)

SQL: COBSQL

- The Cobsql preprocessor was updated to no longer misinterpret WORKING-STORAGE items whose definitions were spread across multiple source lines.

  1080078 (2507684)
- The Cobsql preprocessor has been updated to correctly process variables defined as USAGE COMP after an EXEC SQL INCLUDE SQLCA when the CP preprocessor directive NOSQL is specified.

1084463 (2577593)

- The Cobsql preprocessor has been updated to correctly process Pro*COBOL-generated data items when the first WORKING-STORAGE variable in user code contains a VALUE clause with the literal value on a separate source line.

  1084753 (2579264)

SQL: DB2 ECM

- The DB2 pre-compiler now generates GOBACK instead of STOP RUN at the end of program source, so that poorly coded programs do not fail when running under IMS or other transaction monitors.

  1083235 (2559616)

SQL: OpenESQL

- The OpenESQL preprocessor has been updated such that when a program or application is compiled with the SQL(DBMAN=ADO) compiler directive, the calls generated do not modify user data when a null value is returned.

  1083563 (2561563)
- The OpenESQL preprocessor no longer accepts singleton select statements that have no INTO clause, or host variables that are not preceded by a colon, apart from SQLDA references in dynamic SQL. In rare cases where a singleton select without an INTO clause is required, code it by placing the SELECT statement inside of a BEGIN/BEND block.

  1084430 (2576585)
- The ODBC Compiler did not identify specialized classes such as ENUMS, and incorrectly generated SQL interface code for them that caused Compiler errors. The ODBC Compiler has been enhanced so that it no longer generates SQL interface code for those classes.

  1083144 (2561349)
- Previously, the SQL pre-compilers did not always recognized the EXEC SQL INCLUDE statements during the syntax checking phase and returned incorrect error messages.

  1085057 (2582967)

Vision File System

- When you configure your application to return RM/COBOL file status codes, by setting COBFSTATCONV=rmstat, the codes returned are ANSI'85 codes.

  1082469 (2553438)

XDB Server

- Support has been added to enable the execution of DESCRIBE INPUT statements.

  1079821 (2508551)
- A mutex deadly embrace no longer results in a server hang.

  1085199 (2583889)

XML syntax support runtime

- An issue with the PREXML preprocessor has been resolved and it now does not truncate data names longer than 30 characters.

  1085342 (2586384)
- In previous releases, the documentation for XML Input/Output was missing from the online help. This documentation is now included in the online help.

1084119 (2572393)

Updates and SupportLine

Our Web site gives up-to-date details of contact numbers and addresses.

Further Information and Product Support

Additional technical information or advice is available from several sources.

The product support pages contain a considerable amount of additional information, such as:

- The WebSync service, where you can download fixes and documentation updates.
- The Knowledge Base, a large collection of product tips and workarounds.
- Examples and Utilities, including demos and additional product documentation.

To connect, enter *http://www.microfocus.com* in your browser to go to the Micro Focus home page.

Note: Some information may be available only to customers who have maintenance agreements.

If you obtained this product directly from Micro Focus, contact us as described on the Micro Focus Web site, *www.microfocus.com*. If you obtained the product from another source, such as an authorized distributor, contact them for help first. If they are unable to help, contact us.

Information We Need

However you contact us, please try to include the information below, if you have it. The more information you can give, the better Micro Focus SupportLine can help you. But if you don't know all the answers, or you think some are irrelevant to your problem, please give whatever information you have.

- The name and version number of all products that you think might be causing a problem.
- Your computer make and model.
- Your operating system version number and details of any networking software you are using.
- The amount of memory in your computer.
- The relevant page reference or section in the documentation.
- Your serial number. To find out these numbers, look in the subject line and body of your Electronic Product Delivery Notice email that you received from Micro Focus.

Alternatively, you might be asked to provide a log file created by the Consolidated Tracing Facility (CTF) - a tracing infrastructure that enables you to quickly and easily produce diagnostic information detailing the operation of a number of Micro Focus software components.

On UNIX, you can use the Micro Focus UNIX Support Scan Utility, mfsupport, to create a log file that contains the details about your environment, product, and settings. The mfsupport script is stored in `$COBDIR/bin`.

To run mfsupport:

1. Start a UNIX shell.
2. Set COBDIR to the product with issues.
3. Execute `mfsupport` from a directory where you have write permissions.

   This creates a log file, `mfpoll.txt`, in that directory.
4. When the script finishes, send the `mfpoll.txt` file to your Micro Focus SupportLine representative.

Note:

If COBDIR is set to a location which does not contain `etc/cobver`, the script outputs the contents of `/opt/microfocus/logs/MicroFocusProductRegistry.dat` which keeps a list of the installedMicro Focus products.

Creating Debug Files

If you encounter an error when compiling a program that requires you to contact Micro Focus technical support, your support representative might request that you provide additional debug files (as well as source and data files) to help us determine the cause of the problem. If so, they will advise you how to create them.

Disclaimer

This software is provided "as is" without warranty of any kind. Micro Focus disclaims all warranties, either express or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall Micro Focus or its suppliers be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if Micro Focus or its suppliers have been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of liability for consequential or incidental damages so the foregoing limitation may not apply.

Micro Focus is a registered trademark.