



# Micro Focus Visual COBOL 2.0 Development Hub

---

A large, decorative graphic consisting of multiple overlapping, wavy blue lines that create a sense of motion and depth, positioned in the lower half of the page.

Release Notes

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

**Copyright © 2009-2012 Micro Focus. All rights reserved.**

**MICRO FOCUS, the Micro Focus logo and Visual COBOL are trademarks or registered trademarks of Micro Focus IP Development Limited or its subsidiaries or affiliated companies in the United States, United Kingdom and other countries.**

**All other marks are the property of their respective owners.**

**2012-04-18**

# Contents

<b>Micro Focus Visual COBOL 2.0 Development Hub Release Notes</b>	<b>4</b>
<b>Installation</b>	<b>5</b>
System Requirements for Visual COBOL Development Hub	5
Installing Visual COBOL Development Hub	6
Uninstalling	8
Configuring the Remote System Explorer Support	8
<b>Licensing Information</b>	<b>10</b>
<b>What's New</b>	<b>11</b>
New Features in Visual COBOL 2.0	11
Compiler Directives	11
JVM COBOL File Handler	11
Library Routines	11
Managed COBOL Language Features	11
OpenESQL	12
Run-Time Tunables	12
Vision Data File Searching	12
Features Added in Visual COBOL 2010 R4 Update 2	12
New Platforms Support	12
OO COBOL Class Library Reference	13
Features Added in Visual COBOL 2010 R4	13
ACUCOBOL-GT Compatibility	13
Embedded HTML	14
Language Improvements	14
RM/COBOL Compatibility	14
XML Extensions	15
<b>Known Issues</b>	<b>16</b>
Adding Projects to a COBOL JVM Project's Java Build Path	16
Background Parsing	16
Co-existing with Earlier Micro Focus Products	17
Compiling with Existing Classes on a Classpath	17
Debugging in Eclipse	17
Error Markers	17
JVM COBOL Support	18
Leaking File Handles	19
Remote Debugging	19
Remote Development using Remote System Explorer	20
Unsupported Features	20
Micro Focus Vision	20
Editing Remote JRE Settings	21
<b>Resolved Issues</b>	<b>22</b>
<b>Updates and SupportLine</b>	<b>28</b>
Further Information and Product Support	28
Information We Need	28
Creating Debug Files	29
<b>Disclaimer</b>	<b>30</b>

# Micro Focus Visual COBOL 2.0 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

#### Operating Systems:

AIX POWER 6.1/7.1 - 32/64-bit  
x86-64 running Red Hat Linux 5.5/5.7/6.1 - 32/64-bit  
SPARC running Solaris 10 - 32/64-bit  
x86-64 running SuSE SLES 11/11 SP1 - 32/64-bit  
HP IA 11.31 - 32/64-bit

COBOL for JVM is supported on 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  
x86-64 running SuSE SLES 11/11 SP1 - 32/64-bit

**Disk Space:** 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.

### Software Requirements



**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 will install 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 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.

- In order 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.

- If you use the full image for the installation, you need to first install Java Platform Standard Edition (Java SE) 6 Update 27. You can download Java SE from [www.oracle.com](http://www.oracle.com) and install it anywhere on your machine.
- On UNIX, you must have xterm installed. Xterm, the terminal emulator for the X Window System, is part of your Linux/UNIX distribution but is not installed by default. Use your Linux/UNIX installation media to install it.

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 will terminate. 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 in order to pick up localized messages. The LANG settings are English and Japanese only.

## Installing Visual COBOL Development Hub

### Downloading the Product

1. Click on the product download link in your delivery notice.

This opens the Micro Focus Electronic Product Delivery page for your order.

2. Before downloading your product, check the **Legal Information** tab and read the End User License Agreement.
3. Check the **Documentation** tab and read the documentation supplied for your product.
4. Check the respective links to download the setup file or the Web installer for your product for the appropriate platform to your machine.

### Installing

To use the Web Installer:

1. Give the file execute permissions as follows:

```
chmod +x webinstaller_visualcobol_devhub_2.0_platform
```

2. Run the installer with superuser permissions:

```
./webinstaller_visualcobol_devhub_2.0_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 to install the product as follows:

1. Execute the following to give execute permissions to the setup file:

```
chmod +x setup_visualcobol_devhub_2.0_platform
```

## 2. Run the installer with superuser permissions:

```
./setup_visualcobol_devhub_2.0_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_2.0_platform -installlocation="full path of new location"
```

or

```
./setup_visualcobol_devhub_2.0_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 may 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 will not be affected, you need to use `-skipsafenet` option which skips the installation of SafeNet:

```
./setup_visualcobol_devhub_2.0_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 will 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.

## 3. Set up the environment:

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



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

- a. To set up your product, execute:

```
./opt/microfocus/VisualCOBOL/bin/cobsetenv
```

- b. To verify that your product is installed and licensed, execute:

```
cob -Version
```



**Tip:** 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 on the [Micro Focus Infocenter](#).

## Installing as an Upgrade

**UNIX** 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 2.0 installer with the `-installlocation` option:

1. Execute the following command:

```
./InstallFile -installlocation="/opt/microfocus/VisualCOBOL2.0"
```

2. Execute `cobsetenv` to set the environment and point to the new install location:

```
./opt/microfocus/VisualCOBOL2.0/bin/cobsetenv
```

## Uninstalling

To remove Visual COBOL Development Hub from your machine:

1. Execute the `Uninstall_VisualCOBOL20.sh` script in the `COBDIR/bin` directory.



**Note:** Uninstalling Visual COBOL does not automatically uninstall the Micro Focus Licensing Manager or the prerequisite software.

To uninstall Micro Focus License Manager, run the `UnInstallMFLicenseServer.sh` script in the `/var/microfocuslicensing/bin` directory. The script will 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.

## Configuring the Remote System Explorer Support

The remote development support from the Eclipse IDE relies upon the 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

Some environment setting may need to be set up before starting the Daemon. This is because when a build or debug session is initiated on the development hub from 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.

# 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.

## License for COBOL Server for Unit Testing

You need to install this license to run and test your applications in the runtime environment. It is a timed license and expires after one year after the date of purchase. After this time you need to contact Micro Focus to request a renewal.

There is no evaluation period for this license and you need to use your Authorization Code to obtain a full license before you can test your applications. The Authorization Code can be found in the **Requested License Key** email that is supplied when you order your product. To obtain a full license use the code that is listed under **Visual COBOL - COBOL Server for Unit Testing**.

# 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 2.0

### 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:

<b>Local Variables</b>	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.

## OpenESQL

The JDBC preprocessor option enables you to access databases for applications running under the Java Virtual Machine (JVM). It is now available on Unix and Linux.

## 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\$CALLED BY
  - 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:

<b>Using copybooks</b>	The source code environments are different.
<b>Using environment variables</b>	If the project build is using environment variables, it is likely that they are different in the local and remote environments.
<b>Using the Pre-compiler</b>	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 a Visual 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 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.

## 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:

- Only the following library routines can be used in JVM projects:

CBL\_ABORT\_RUN\_UNIT  
CBL\_ALLOC\_DYN\_MEM  
CBL\_ALLOC\_MEM  
CBL\_ALLOC\_SHMEM  
CBL\_ALLOC\_THREAD\_MEM  
CBL\_AND  
CBL\_CANCEL  
CBL\_CANCEL\_PROC  
CBL\_CHECK\_FILE\_EXIST  
CBL\_CLOSE\_FILE  
CBL\_CLOSE\_VFILE  
CBL\_COPY\_FILE  
CBL\_CREATE\_DIR  
CBL\_CREATE\_FILE  
CBL\_CULL\_RUN\_UNITS  
CBL\_DELETE\_DIR  
CBL\_DELETE\_FILE  
CBL\_DIR\_SCAN\_END  
CBL\_DIR\_SCAN\_READ  
CBL\_DIR\_SCAN\_START  
CBL\_EQ  
CBL\_ERROR\_PROC  
CBL\_EXEC\_RUN\_UNIT  
CBL\_EXIT\_PROC  
CBL\_FILENAME\_CONVERT  
CBL\_FILENAME\_MAX\_LENGTH  
CBL\_FLUSH\_FILE  
CBL\_FREE\_DYN\_MEM  
CBL\_FREE\_MEM  
CBL\_FREE\_SHMEM  
CBL\_FREE\_THREAD\_MEM  
CBL\_GET\_CURRENT\_DIR  
CBL\_GET\_EXIT\_INFO  
CBL\_GET\_OS\_INFO  
CBL\_GET\_PROGRAM\_INFO  
CBL\_GET\_SHMEM\_PTR  
CBL\_IMP  
CBL\_JOIN\_FILENAME  
CBL\_LOCATE\_FILE  
CBL-NLS\_CLOSE\_MSG\_FILE  
CBL-NLS\_COMPARE

CBL-NLS\_INFO  
CBL-NLS\_OPEN\_MSG\_FILE  
CBL-NLS\_READ\_MSG  
CBL\_NOT  
CBL\_OPEN\_FILE  
CBL\_OPEN\_VFILE  
CBL\_OR  
CBL\_PUT\_SHMEM\_PTR  
CBL\_READ\_FILE  
CBL\_READ\_KBD\_CHAR  
CBL\_READ\_VFILE  
CBL\_RENAME\_FILE  
CBL\_SPLIT\_FILENAME  
CBL\_TOLOWER  
CBL\_Toupper  
CBL\_WRITE\_FILE  
CBL\_WRITE\_VFILE  
CBL\_XOR  
CBL\_YIELD\_RUN\_UNIT  
JVM\_LOAD\_NATIVE  
X"F5"  
X"F4"

- 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 may 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.

## Leaking File Handles

The `startdodaemon` 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 is 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 startdodaemon 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 startdodaemon 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

### Projects compiled to .int, .gnt and .lbr files

The preferred executable file formats with Visual COBOL are .dll and .exe. The .int and .gnt file formats are still supported by the Compiler and the debugger but cannot be created directly by the IDE.

You can use the Net Express Project Import Wizard to convert existing Net Express projects into projects, and to import the COBOL source into the IDE. If you convert a project containing these code configurations, the following will happen:

- Any configurations that have .int, .gnt or .lbr will be automatically converted to .exe.
- A message will display confirming that the settings have been changed.

### Run-time Features

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

- Dialog System (Character and GUI)
- GUI Class Library
- HyHelp
- Panels Version 2

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

## 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.

- [Animator \(Character version\)](#)
- [Compiler](#)
- [Compiler ECM Support](#)
- [Documentation](#)
- [File Handling - External File Handler](#)
- [File Handling - Fileshare](#)
- [File Handling - Rebuild](#)
- [File Handling - Sort](#)
- [Interface Mapping General](#)
- [RTS](#)
- [Setup Issues \(UNIX\)](#)
- [SQL Support \(non-Compiler\)](#)
- [SQL: COBSQL](#)
- [SQL: DB2 ECM](#)
- [SQL: OpenESQL](#)
- [Vision File System](#)
- [XML Extensions](#)

## **Animator (Character version)**

- When using the Step function to debug a program running on Solaris that contains a DECLARATIVES section, the execution points are now correctly highlighted.  
1081749 (2544940)

## **Compiler**

- In constructs of type "STRING item1 item2 INTO identifier1", if the identifier uses a reference modification, the length from the picture clause definition was incorrectly used when determining if item1 and item2 fit into identifier1.  
586313 ( )
- Invalid syntax errors are no longer produced when compiling screen section items.  
1082549 (2549214)
- ACUCOBOL-GT reporting has been enhanced and now produces a Compiler error for the following syntax which is not yet supported: ACCEPT... CURSOR <data-item>.  
1081729 (2544855)
- An XML GENERATE statement on a record containing an ambiguous name whose case is distinct from the other declarations of that name now executes as expected.  
1081691 (2543133)
- The ACUCOBOL-GT syntax ACCEPT ... DISPLAY OMITTED previously caused an internal Compiler error.  
1081423 (2538935)

- Compiling a program which contains an XML GENERATE statement to exe no longer produces unresolved external symbol errors.  
1078019 (2498657)
- Illegal ODO items which reference many levels of qualification now display a syntax error.  
1080616 (2530541)
- Executing a statement using an intrinsic function with a reference modified argument when compiled with SSRANGE(2) no longer produces spurious run-time errors.  
1080275 (2526877)
- The RECORD-POSITION operator returned a value that was one less than the correct value.  
586672 ( )

### Compiler ECM Support

- A declaration of SQL TYPE ROWID was sometimes not processed correctly.  
1080477 (2529596)

### Documentation

- The following run-time system tunable is available: subsystem\_cancel\_mode. Specifies whether logical or physical cancels are used for CBL\_SUBSYSTEM cancels. Syntax:

```
>>-----set subsystem_cancel_mode---.-0-.----->< +-1-+
```

Parameters: 0 Physical cancels: any programs canceled with a CBL\_SUBSYSTEM cancel will be freed from memory, making more memory available. 1 Logical cancels: any programs canceled with a CBL\_SUBSYSTEM cancel will not be freed from memory. The programs are available to be reused and hence the performance of CALL/CANCEL statements is greatly increased. As far as your program is concerned, the behavior of logical and physical cancels is identical. Properties: Default: 0 on Net Express; 1 on Server Express

- 1080440 (2529528)
- 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.  
584421 ( )
- If you use the full image when installing Visual COBOL or the Development Hub, you must first install Java Platform Standard Edition (Java SE) 6 Update 27.  
1082268 (2550806)
- This release requires version 10000.2.660 or later of the Micro Focus licensing software. For local servers, you do not need to install it separately, as the setup file will install a new client (Visual COBOL) 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. You can check the version of your license server by clicking **Help > About** in the Micro Focus Licensing System Administration tool.  
1082175 (2549745)
- In order to successfully complete the example of Java Calling JVM COBOL in the Interoperability section of the documentation, you must add mfcobol.jar to the project's classpath.  
585406 ( )

### File Handling - External File Handler

- When reading a file cataloged as DISP=SHR the file handler now buffers the read for better performance.

- 1079491 (2518330)
- DUMMY files defined as indexed now open correctly on UNIX platforms.
- 1079483 (2516513)
- Dataset concatenations ending with DD DUMMY (or DSN=NULLFILE) no longer cause an abend when the file is closed.
- 1078960 (2511384)
- When opening a file for I-O under mainframe emulation, the file does not have to exist.
- 1078362 (2501743)
- A series of concatenated files will now all be opened with the same opcode so it is treated as a single file.
- 1078617 (2503276)
- Missing indexed files in the USE statement incorrectly resulted in a 114 error. A 9/013 I/O error is now correctly displayed.
- 583179 ( )
- SORT did not display correctly the report fields (such as sections, headers, trailers) when no OUTREC was specified.
- 583227 ( )
- An invalid input detected in the SORT FIELDS statement caused the application to crash. A 114 error message was incorrectly displayed. This has been changed and an error "Command statement error(s) detected" is now displayed.
- 583425 ( )
- Setting STRIPSPACE to OFF now correctly honors the record length and allows trailing spaces for LSEQ files.
- 573921 ( )
- Support has been added to allow trailing spaces in LSEQ files when used with the variable STRIPSPACE=OFF, when printing records via stdout or piping to a file.
- 585740 ( )
- Sequential files on a 64-bit machine are now correctly padded to fill the LRECL.
- 586461 ( )
- A memory leak could occur when performing file I-O operations inside a Web service.
- 1081398 (2540611)

#### **File Handling - Fileshare**

- An fsview close file now correctly closes the file.
- 1079724 (2508855)
- A CICS close on a file no longer fails if a transaction has started and updates are made to any file.
- 1079738 (2473753)
- Using a READ statement after a CLOSE REEL through Fileshare could result in a 9/077 error.
- 587474 ( )

#### **File Handling - Rebuild**

- If the checkpoint LAST is set and no checkpoint exists, BATCH BACKOUT now backs out of the entire log.
- 1081493 (2541220)
- Rebuild will no longer try to rebuild an indexed file without any keys, and will report the failure. Rebuild now also reports that there are no keys when performing file validation or file information on the file.



1078000 (2499296)

### File Handling - Sort

- The NUM function has been added to SORT.

1079136 (2512764)

- The TOOLMSG output "Number of records resulting from criteria:" now displays the number of records correctly.

1079280 (2513886)

- Users can now use the DATE+n field constants as well as comparing DATE constants within the INCLUDE/OMIT statements.

1078273 (2501816)

- NODETAIL now correctly prevents data records from being displayed in the OUTFIL. SORT will also correctly default to displaying the entire input record if neither NODETAIL or an OUTREC is specified.

1074166 (2443710)

- COUNT will now work correctly, representing a count of the number of data records in that section. The value of SUBMAX will now be output when defined in the OUTFIL.

1080528 (2443710)

- SORT will now regard the next line as a continued comment (and will therefore ignore everything on that line) if the previous line is a completed statement and has a comment in column 72.

1080262 (2527579)

- Using TOTAL or SUBTOTAL will now display the correct value instead of truncating the result.

584223 ( )

- Using SKIP=nL in the SECTIONS section of SORT will now cause the following behavior: 1) Under a mainframe sort, (n/3 rounded up) blank lines (when n>0) will be skipped and the appropriate carriage control character will be placed in column 1 of each blank line. 2) For sequential file organizations, the same behavior will occur but no carriage-control characters will be printed. 3) For line sequential files, n blank lines will be skipped.

1080092 (2525670)

- An RTS 252 error no longer occurs if RTS memory checking is enabled and outfil processing is used in a sort card.

1081337 (2540109)

- Setting SORTSPACE to the same value as SORTTEMPSPACE no longer causes an RTS 252 to occur.

1081277 (2539633)

- SORTCOMPRESS=5 will no longer cause the records output from SORT to differ from the records that are taken in.

1080006 (2518730)

- MFSORT will now accept the following constants in the outrec, header, and footer: \* DATE1+n \* DATE1-n \* DATE1(c)+n \* DATE1(c)-n \* DATE2+n \* DATE2-n \* DATE2(c)+n \* DATE2(c)-n \* DATE3+n \* DATE3-n \* DATE3(c)+n \* DATE3(c)-n \* DATE1P+n \* DATE1P-n \* DATE2P+n \* DATE2P-n \* DATE3P+n \* DATE3P-n where n represents the number of days for DATE1 and DATE3, and number of months for DATE2, and where c represents the date separator.

1079771 (2521778)

- The keywords CENTWIN and Y2PAST are now both accepted under both DFSORT and SYNC SORT emulation.

585369 ( )

- Sort now closes down correctly after an RTS 163 error, and is now able to carry on and process further jobs.

1082133 (2544148)

- When combining OUTFIL and SORTOUT files where headers and trailers were used, SORTOUT sometimes contained mainframe printer control characters.  
1082048 (2548488)
- When writing to EBCDIC files, DATE constants will now correctly be converted into EBCDIC format.  
586917 ( )
- A SORT performed from REXX no longer causes a COBRTS 114 error.  
1081942 (2546794)
- SORT will now correctly compare INCLUDE/OMIT parameters when SUM FIELDS=NONE.  
1083014 (2558088)
- Setting MFJSSTRICTSORT=ON will now cause records that are shorter than the output file's LRECL to be padded with low values instead of spaces.  
1081919 (2547118)
- If a SUBSYS is specified and does not exist, SORT reverts to using EXTFH and emits a warning message.  
1081570 (2542992)
- PARM RC16 is ignored but the comment line reports a warning.  
1081203 (2537395)

### Interface Mapping General

- User Exit 5 now allows the Web service response to be overridden.  
1081015 (2516533)

### RTS

- The Micro Focus run-time system is no longer unloaded prematurely when using IBM TXSeries, and does not produce a COBRT 173 error.  
1079371 (2516765)
- The run-time text window no longer returns incorrect data when content is copied from it using Edit/Mark.  
1078975 (2511669)
- The CBL\_RENAME\_FILE and CBL\_COPY\_FILE routines no longer cause a memory leak on UNIX platforms.  
1076845 (2484033)
- If you run 32-bit XML Extensions on AIX and you are exporting a very large data item (over 60 Mb), you can encounter error 72[6]: "Data conversion error text: Memory allocation error". To workaround this issue, you need to set the LDR\_CNTRL environment variable before running the program as follows:  
LDR\_CNTRL=MAXDATA=0x40000000; export LDR\_CNTRL  
586691 ( )
- Use of PC\_PRINTER\_DRAW\_RECTANGLE, PC\_PRINTER\_GET\_BMP\_INFO, PC\_PRINTER\_LINE\_FROM\_TO, PC\_PRINTER\_SET\_BACKGROUND\_MIX\_MODE, PC\_PRINTER\_SET\_PEN or PC\_PRINTER\_SET\_POSITION no longer give an unresolved symbol error when linking as a 32-bit application.  
1072571 (2418196)

### Setup Issues (UNIX)

- During installation, the install script checks a list of expected locations for the GCC compiler, and when found, updates the cobopt file with that location. This list now includes locations appropriate for different platforms and different levels of GCC compiler.

1081011 (2535964)

### **SQL Support (non-Compiler)**

- The ODBC pre-compiler has been modified to properly support UTF-8 coded host variable names in EXEC SQL statements.

1079019 (2512255)

### **SQL: COBSQL**

- The Cobsql preprocessor has been updated to improve robustness of error handling when working with Oracle Pro\*COBOL on UNIX.

1082474 (2552830)

### **SQL: DB2 ECM**

- The DB2 ECM has been updated to resolve run-time errors returned when compiling against mainframe databases in 64-bit mode.

1082441 (2549058)

### **SQL: OpenESQL**

- OpenESQL has been updated to support 64-bit operation for INSERT and FETCH statements.

1079765 (2499575)

### **Vision File System**

- There is a new configuration option - RMFM\_RUN\_INDEX\_FILES\_PREVENT\_WRITE\_STARVATION. The default value of this option is set to NO, but when set to YES, it enables the new additional locking scheme that prevents write starvation when there are many active readers of the same file.

589086 ( )

- ACUFH incorrectly returned status 00 on OPEN INPUT of a non-existent OPTIONAL file. This has been fixed so that ACUFH returns the correct status 05.

1082400 (2552367)

- An alternate key numbering mismatch occurred during the execution of the OPEN statement for an RM indexed file.

1079788 (2521519)

### **XML Extensions**

- XML EXPORT failed to output some occurrences from an OCCURS data item nested in another OCCURS data item.

1080752 (2533412)

# 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](http://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 installed Micro 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.

Copyright © Micro Focus 1984-2012. All rights reserved.