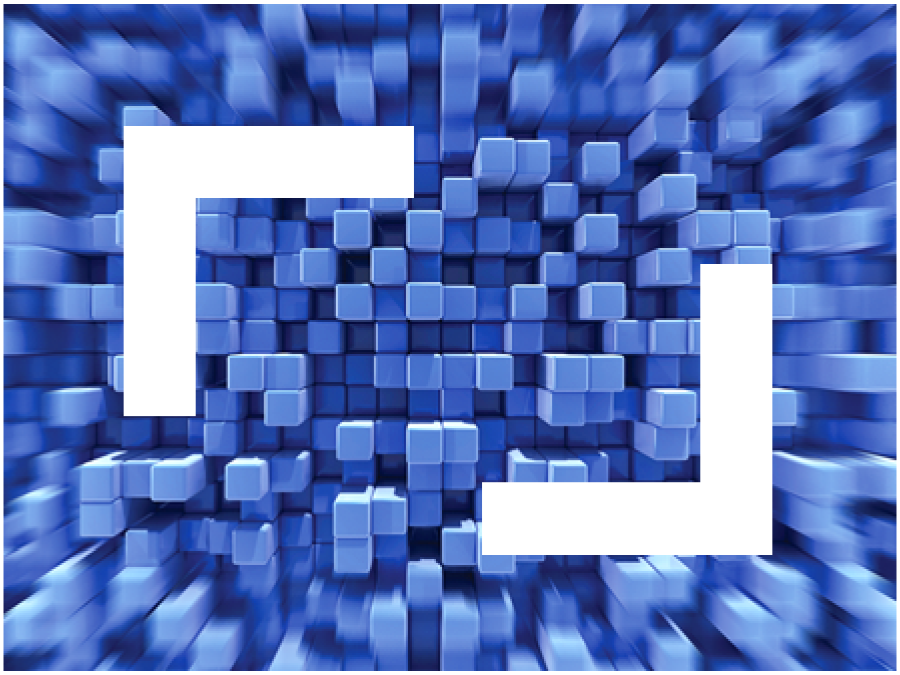

Micro Focus Visual COBOL 2010 R4 for Eclipse Update 1

Release Notes

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

2011-09-09

Contents

Micro Focus Visual COBOL 2010 R4 for Eclipse Update 1 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

System Requirements for Visual COBOL for Eclipse

Hardware Requirements

Windows Operating Systems:

Windows XP 64 bit SP2
Windows XP 32-bit SP3
Windows Vista SP1 32/64-bit
Windows 7 32/64-bit
Windows Server 2003 SP2 32/64-bit
Windows Server 2003 R2 32/64-bit
Windows Server 2008 SP2 32/64-bit
Windows Server 2008 R2 32/64-bit

Disk Space:

Approximately 60 Mb for the Sentinel RMS license server

Approximately 430 Mb for Visual COBOL. Note: This disk size includes the space needed to cache information locally so that you can modify the installation without the original source media.

UNIX Operating Systems:

Red Hat Linux x86-64 5.5 32/64-bit
SUSE 11 Linux x86-64 32/64-bit

Disk Space:

Approximately 60 Mb for the Sentinel RMS license server

Between 270 Mb and 350 Mb for Visual COBOL depending on the platform. Note: The installation requires extra disk space that equals the size of the product you install.

Software Requirements

You need to install Java Platform Standard Edition (Java SE) 6 or later (32-bit only) beforehand. You can download Java SE from *www.oracle.com* and install it anywhere on your machine.

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

Windows

- Set the JAVA_HOME environment variable. When installing the product, set this variable to a 32-bit Java installation or the installation will terminate.
- If you use XML Extensions, note that XML Extensions has the same requirements as RM/COBOL version 12 for 32-bit Windows. (See the *RM/COBOL User's Guide, Second Edition* or later.) Additionally, XML Extensions may be used in conjunction with Terminal Server.

UNIX

- 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 for Eclipse

Windows

Download the setup file on your machine and install the product as follows:

1. Click on the **Visual COBOL for Eclipse : Integrated Development Environment for Eclipse on Windows (Full Image)** link in your delivery notice and save the `visualcoboleclipser4_update1.msp` file to your computer.
2. Run the file and follow the wizard instructions to complete the installation.
3. Install the COBOL 2010 Runtime Deployment Environment for Testing Purposes in the same way.

Note:

- If you are installing onto a machine that has an existing Micro Focus product that uses an older Sentinel RMS License Manager, you might be prompted to remove it and install the Micro Focus License Manager. By doing this you will maintain the existing Sentinel RMS license files while adding the Micro Focus License Manager. If you are unsure about existing licenses on your computer or removing the Sentinel RMS License Manager, consult your System Administrator. If you want to proceed, remove Sentinel RMS License Manager by using Windows **Add or Remove Programs** and rerun the installation file.
- Trial licenses cannot be used with remote desktop services. If you want to use your product in this way, please contact Micro Focus SupportLine to obtain a relevant license.
- We recommend that you install any updates of the .NET Framework that are available at the *Microsoft Download* site.
- If you install JDK you might be prompted to install the latest update. The latest update is not required for use with Visual COBOL but you can install it if you wish.

UNIX

Download the setup file on your machine and install the product as follows:

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

   ```
   chmod +x setup_viscobolr4_deveclipse_update1
   ```
2. Run the installer with superuser permissions:

   ```
   COBDIR=[location of the base product]./
   setup_viscobolr4_deveclipse_update1
   ```

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

   COBDIR must be set or point to the location of the installed base product. If it is not set or if the installer does not find a valid product at that location, the installation will terminate.

To install in a different location use the `-installlocation="Location"` parameter to specify an alternative directory location. For example:

```
./setup_viscobolr4_deveclipse_update1 -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_viscobolr4_deveclipse_update1 -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:

   Important: 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.)

   c. To start Eclipse, double-click the icon for Micro Focus Visual COBOL for Eclipse (automatically installed on your desktop).

After You've Installed the Product

Eclipse perspective settings do not get reset after installation. To pick up any new features, you must reset the COBOL perspective after installation:

1. Make sure you are in the COBOL perspective by clicking **Window > Open Perspective > COBOL**.
2. Click **Window > Reset Perspective**.
3. Click **OK**.
4. Reapply any customizations.

Installing Silently

You can install Micro Focus products silently by using command line parameters to specify the installation directory, user information, and which features to install.

On Windows:

To install silently use the following command:

```
start /wait install-file.msp /q [parameters]
```

where *install-file* for the following products is as follows:

Visual COBOL	`visualcoboleclipser4_update1.msp`
COBOL 2010 Runtime	`cobol2010runtimer4_update1.msp`

After the installation is complete you can install the license silently. To do so execute:

```
start /wait <install-dir>\bin\cesadmintool –term activate AuthorizationCode
```

Directory Considerations (Windows)

- You must have read and write access for every directory accessed during the install.
- You can override the default installation folder using the INSTALLDIR parameter:
- If a path in a definition contains spaces, then the path must be preceded by a backslash and double quotation mark (\"). For example:

  ```
  INSTALLDIR=path
  INSTALLDIR=\"c:\MyProduct\"
  INSTALLDIR=\"c:\Program Files\Micro Focus\My Product\"
  ```
- Installing creates a log file in `%temp%\LogFilename` by default. To change the location or name, use the /l parameter on your Setup command line and specify the path and file name, for example:

  ```
  /l*v drive:\path\LogFilename
  ```

  The default names for the log files are as follows:

`VisualCOBOL_install_log.txt`	for the Visual COBOL wrapper.
`VisualCOBOLx??_install_log.txt`	for Visual COBOL, where *??* is "86" for 32-bit systems and "64" for 64-bit systems.
`lmsetup_install_log.txt`	for License Manager.
`COBOL2010Runtime_install_log.txt`	for the COBOL 2010 Runtime wrapper.
`COBOL2010Runtimex??_install_log.txt`	for COBOL 2010 Runtime, where *??* is "86" for 32-bit systems and "64" for 64-bit systems.
- The log filename and folder name cannot contain spaces
- The log file folder must exist before beginning the silent install

On UNIX:

You can use the following command line arguments to install silently on UNIX:

```
–silent -IacceptEULA
```

For example, execute:

```
setup_filename –silent -IacceptEULA
```

Installing X Window on Windows

Some features of Visual COBOL 2010 R4 Eclipse for Windows require an X Windows installation, hence Micro Focus ViewNowX is provided with the product. To install, run the file `ViewNow_X_Server.msi` in your Visual COBOL installation. By default this will be in the `C:\Program Files\Micro Focus \Visual COBOL 2010\ViewNowX` folder.

ViewNowX requires that your client machine has Microsoft Visual C++ 2008 SP1 Redistributable Package (x86) installed. If it is missing from your machine, the ViewNowX installation will offer a link to download the package.

Configuring the Environment for Developing RDBMS Applications on UNIX

Note:

- If you are working with remote projects, you need to configure the environment before you start the remote server process.
- On UNIX, if you are working with local projects, you need to configure the environment before you start Eclipse.

1. Ensure the COBOL and the third-party software environments are set.
2. Set up the RDBMS environment.

   Refer to your RDBMS vendor documentation for details.
3. Set COBCPY as required in order for the IDE to locate any copybooks that are external to your project.
4. If working with Pro*COBOL/Cobsql applications you need to set COBOPT. This sets the appropriate linker options for the platform and the COBOL working mode:

   In order to do this, execute the following at the command line:

   ```
   $COBDIR/src/oracle/set_cobopt_oracle
   COBOPT=$PWD/cobopt.ora
   export COBOPT
   ```

Repairing

On Windows, if any product files, registry settings or shortcuts are accidentally removed at any point, you can perform a Repair on the installation to replace them.

To repair your installation on Windows XP:

1. Click **Start Menu > Control Panel > Add/Remove Programs**.
2. Click your Micro Focus product in the list of installed programs.
3. Click **Click here for support information**.
4. Click **Repair**.

To repair your installation on Windows Vista or Windows 7:

1. Click **Start Menu > Control Panel**.
2. Click **Uninstall a program** under **Programs**.
3. Right-click your Micro Focus product and select **Repair**.

Uninstalling

Note: The installer creates separate installations for Micro Focus Visual COBOL (or COBOL 2010 Runtime) and Micro Focus License Manager. Uninstalling only Visual COBOL (or COBOL 2010 Runtime) does not automatically uninstall the Micro Focus Licensing Manager or any of the prerequisite software.

To completely remove the product you must uninstall the Micro Focus Licensing Manager as well.

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

Windows To remove Visual COBOL and Micro Focus License Manager, use the Windows `Add or Remove Programs` tool.

UNIX To remove Visual COBOL (or COBOL 2010 Runtime), execute as the root user the `Uninstall_VisualCOBOLR4.sh` script in the `/opt/microfocus/VisualCOBOL/bin` directory.

To uninstall Micro Focus License Manager, run the `UnInstallMFLicenseServer.sh` script in the `/microfocuslicensing/bin directory`. The script will not remove some of the files as they contain certain system settings or licenses.

Licensing Information

Tip: 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.

Windows Use the Authorization Code, supplied with your delivery notice, to license your product. To do this:

1. Click **Start** > **All Programs** > **Micro Focus License Manager** > **License Management System**.

   The dialog box for the Micro Focus License System Administration Tool will be displayed.
2. Click the **Authorize** tab.
3. Enter your Authorization Code and click **Authorize**.
4. Close the Micro Focus License System Administration Tool.

If you wish to install the license silently, execute the following after the product has been installed:

```
start /wait <install-dir>\bin\cesadmintool –term activate
AuthorizationCode
```

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 the Integrated Development Environment for Eclipse

Your product is supplied with an evaluation license that typically enables you to use your product for 30 days. While you are using the evaluation license, a dialog will display when you start your product prompting you to enter the Authorization Code. This code can be found in the **Requested License Key** email that is supplied when you order your product. Depending on your product order, you might have multiple license Authorization Codes. To obtain a full license for Visual COBOL, enter the code that is printed under **Visual COBOL - Visual COBOL 2010 R4 for Eclipse**.

Note: If you have purchased a Visual COBOL timed license, the license management system may show these as Trial licenses. This does not affect the license you have purchased.

Note: An XSERV963 license key is supplied for ViewNow X Server.

License for the COBOL 2010 Test Runtime Environment

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 2010 Test Runtime**.

What's New

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

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 routines can now be used with Visual COBOL:
  - 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.

COBOL for Java Virtual Machine (JVM) Support

In this release you can compile COBOL applications to JVM byte code (`.class` files) so that they can be run on a JVM. There is support in the IDE to edit, compile and debug JVM COBOL applications. This release includes Managed COBOL - COBOL with extensions to support the JVM framework plus OO syntax support.

Features include support for

- New managed COBOL syntax:
  - The SYNC statement, which marks a statement block as a critical section by obtaining the mutual-exclusion lock for a given object, executing a block of statements, and then releasing the lock.
  - Extension methods, enabling you to extend an existing class with new methods without the need to recompile the existing class
  - Java style inner classes, which define a nested class within another class. These follow all the methods belonging to the containing class.
  - The STATIC keyword, enabling you to mark methods and data as static.
  - Enumerators, which represent a list of constant values. You can declare an enum type that defines the values and symbolic names for them, and refer to the values by name in your code.
- Seamless interoperation between COBOL compiled for JVM and COBOL compiled for Java code
- The ability to add Java projects to the classpath for COBOL JVM projects, and COBOL JVM projects to the classpath for Java projects
- Red Hat, AIX and SUSE platforms

Improved Usability

The following improvements to COBOL development in Eclipse have been made:

Dialog box improvements The dialog boxes governing build configuration, debug configuration, and launch configuration are redesigned to make them easier to use.

Autocompletion and Content Assist The IDE now includes autocompletion and content assist editing features.

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.

Renamed Color Preferences

The COBOL Editor syntax-coloring scheme called "Micro Focus Net Express" is now called "Micro Focus Traditional".

The syntax-coloring element called "Identifiers" is now called "Identification Division Names".

Any existing settings for these options will not change.

Reporting of Linker Errors

Errors relating to linking, such as undefined entry points and libraries not being found, are now logged in Eclipse's Problems view.

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 routines can now be used with Visual COBOL:
  - C$Century
  - CConvertAnsiToOem
  - C$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

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 either COBCONFIG or COBCONFIG_ 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 Net Express or Studio Enterprise Edition installed and either COBCONFIG or 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 COBCONFIG_) or set it to another configuration file that does not contain the invalid tunable for the particular session you are running in.

Licensing error due to environment settings

The message "Micro Focus License Manager service is not running" can occur when you invoke a Net Express or Studio Enterprise Edition utility from Visual COBOL. This happens when the tool is invoked with Visual COBOL environment settings while it requires the Net Express or Studio Enterprise Edition ones.

This may happen when you edit files such as .dat that have a file association with Net Express or Studio Enterprise Edition.

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

- 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.
- A core dump file is not created for a stack overflow condition.

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.

Fileshare

- A write of a record with an alternate key where the alternate key is the first value for that key may now return a 0/2 file status rather than a 0/0 status if a record with that alternate key value has been deleted by uncommitted transaction (possibly the same transaction as that performing the write).

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
X"F5"
X"F4"

- 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 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.

Running Applications

- If you have installed JRE and not the JDK you might not be able to start COBOL programs under the control of the cobjrun VM. To resolve this, you need to add the JRE's libjvm.so file to the LD_LIBRARY_PATH using the following command:

  ```
  export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JAVA_HOME/jre/lib/i386/server
  ```

SQL Compiler Directive Options

- The SQL Directives dialog box provides a list of selectable SQL compiler directive options you can set for SQL applications. However, while all of the options are selectable from the list, not all are supported within the Eclipse development environment:

  - For the DBMAN option, only the ODBC value is supported, i.e., SQL(DBMAN=ODBC). The ADO and OADO values are not supported within the Eclipse environment.
  - Directives specifically for use with the HCO for SQL Server feature are not supported. They are:

    ACCESS
    BIND
    DATE
    DBRMLIB
    DIALECT
    MEMBER
    QUALIFIER
    TIME
    VERSION

Unsupported Features

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

The file types .int, .gnt and .lbr are not supported in Visual COBOL. If you open 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.

Using Java and COBOL Together

- On Linux, you need to use the Java/COBOL launcher, cobjrun, to call COBOL from Java. However, an attempt to call COBOL from Java using the Eclipse IDE might fail with error "Specified executable /opt/microfocus/VisualCOBOL/bin/cobjrun does not exist for jdk*version*". This is because Java searches for the cobjrun executable in the bin directory of the JDK installation. To work around this, you need to add a symbolic link to cobjrun in the bin directory of the JDK installation and then specify this link as the JAVA executable. Read *Java/COBOL Application Launcher* for more.

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.

XML Extensions

"50 Error - missing XML parser (MSXML6)" is displayed for XML Extensions

The XML parser could not be found. This error occurs only on Windows and indicates the MSXML 6.0 parser is not installed in the Windows system. The MSXML 6.0 parser is normally installed on most Windows systems, but could not be found on this system. MSXML 6.0 can be obtained by downloading it from Microsoft's web site or re-installing RM/COBOL.

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.

- *Adis*
- *Compiler*
- *Eclipse IDE*
- *Vision File System*

Adis

- An ACCEPT FUNCTION-KEY FROM ESCAPE statement called an incorrect ADIS routine and as a result the application waited for keyboard input.

  1080386 (2528881)

Compiler

- The width of Shift Jis characters in the sequence area was determined incorrectly and as a result the source code did not compile in the correct area.

  1080341 (2528695)
- Compiling a program containing DBCS characters produced spurious Compiler errors.

  1080336 (2528637)

Eclipse IDE

- If you previously added the Visual COBOL software update site to your Eclipse installation, you should perform the following steps in order to ensure you get the most up-to-date Visual COBOL features and plug-ins: 1. Start Visual COBOL for Eclipse. 2. Choose Help > Install New Software and deselect any existing Micro Focus Visual COBOL entries. 3. Choose Add > Archive, browse to the UpdateSite folder in your installation and choose the .zip file.

  584482 ( )

Vision File System

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

  1079788 (2521519)
- ACUFH has been enhanced to perform key mapping between the FCD passed to ACUFH and the keys available in the indexed file. This allows programs with differing alternate key orderings to open any indexed file with sufficient matching alternate keys.

  1080028 (2524660)

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 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 Software Support Identification Number (SHIN) if you have one (not used in all countries).
- 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.

On Windows, if you are reporting a protection violation you might be asked to provide a dump (`.dmp`) file. To produce a dump file you use the **Unexpected Error** dialog box that is displayed when a protection violation occurs. Unless requested by SupportLine, leave the dump setting as `Normal` (recommended), click **Dump**, then specify a location and name for the dump file. Once the dump file has been written you can email it to SupportLine.

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.

There are three debug files that the support representative could ask you to provide. The debug files, and the directives you use to create them, are as follows:

Directive	File Created	Contains
CHKECM(CTRACE)	ecmtrace.txt	Pseudo COBOL code that shows the code generated to replace the EXEC SQL statements. This file is equivalent to output out of the IBM COBOL precompiler.
CHKECM(TRACE)	ecmtrace.txt	Detailed information as to what information is passed between the preprocessor and the Compiler. If an error occurs that generates invalid syntax, this file will be required to help isolate where the problem occurred.
DB2(CTRACE)	sqltrace.txt	Detailed list of information passed to IBM precompiler services and the results. This file is very useful if an error may involve a bug in the DB2 system software as well as the preprocessor.

Supplementary Information

The following sections in the documentation have been changed since it was released:

Compiler Directives

The Compiler directive ILREF is applicable to JVM managed COBOL, as well as .NET managed COBOL. It permits the definition of extension methods and extension operators that are outside of the current compilation unit.

For the parameter, specify a .class or .jar filename that contains the required extension methods and extension operators. When you do:

- All .class files specified are scanned, and extension methods and extension operators are loaded into the compiler's type resolution system.
- All .jar files specified are read in memory, and any .class files that they contain are processed as above.
- All other files that do not end in the suffix .jar or .class are assumed to be in .jar format, and treated as such. This allows support for .zip, .war, .ear and other such file suffixes.

Any file specified that is not successfully processed by any of the above methods will result in an exception error during Compiler startup.

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.

Index