



Micro Focus® Visual COBOL® for Eclipse Distributed Edition

A large, decorative graphic consisting of several overlapping, wavy blue lines that create a sense of motion and depth. The lines are in various shades of blue, from light to dark, and are set against a light blue gradient background.

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.

2012-09-18

Contents

Micro Focus® Visual COBOL® for Eclipse Distributed Edition Release Notes	5
Installation	6
Installing Visual COBOL on Windows	6
System Requirements for Visual COBOL for Eclipse (Windows)	6
Installing Visual COBOL for Eclipse (Windows)	7
Installing on UNIX/Linux	9
System Requirements for Visual COBOL for Eclipse (UNIX)	9
Installing Visual COBOL for Eclipse (UNIX)	11
System Requirements for Visual COBOL Development Hub	15
Hardware Requirements	15
Operating Systems Supported	15
Software Requirements	15
Installing Visual COBOL Development Hub	16
Downloading the Product	16
Installing	16
Installing as an Upgrade	17
UNIX Installer Issues	18
Configuring the Remote System Explorer Support	18
Repairing	19
Uninstalling	19
Licensing Information	21
What's New	23
New Features in Visual COBOL 2.1	23
ACUCOBOL-GT Data Types in Managed Code	23
ACUCOBOL-GT Library Routines in Managed Code	23
Automatic Directives Detection and Setting	23
Compiler Directives	23
.int, .gnt and .lbr File Types Support	23
Just-in-time debugging	24
JVM Class Library	24
Managed COBOL Enhancements	24
OpenESQL	25
UNIX Platforms Support	25
Features Added in Visual COBOL 2.0	25
Automatic Directives Detection and Setting	25
COBOL Explorer View	25
COBOL File Search	27
Compiler Directives	28
Compiling Single Files	28
Enhancements to Developing Applications on a Remote Machine	28
Converting Net Express Projects	28
Copybook Context	29
Debugging Enhancements	29
Eclipse 3.7	29
JVM COBOL File Handler	30
Library Routines	30
Managed COBOL Language Features	30
Data Access	30
Remote COBOL JVM Projects - Early Release	31

Run-Time Tunables	31
Samples	31
Vision Data File Searching	31
Features Added in Visual COBOL 2010 R4 Update 2	31
New Platforms Support	31
OO COBOL Class Library Reference	32
OpenESQL	32
Features Added in Visual COBOL 2010 R4	32
ACUCOBOL-GT Compatibility	32
COBOL for Java Virtual Machine (JVM) Support	33
Embedded HTML	33
Improved Usability	34
Language Improvements	34
Renamed Color Preferences	34
Reporting of Linker Errors	34
RM/COBOL Compatibility	34
XML Extensions	35
Known Issues	36
Resolved Issues	38
Updates and SupportLine	44
Further Information and Product Support	44
Information We Need	44
Creating Debug Files	45
Disclaimer	46

Micro Focus® Visual COBOL® for Eclipse Distributed Edition Release Notes

The Micro Focus Visual COBOL for Eclipse Distributed Edition package comprises the following products which you can download from your Electronic Product Delivery Note:

- Visual COBOL for Eclipse - provides an Eclipse-based integrated COBOL development environment for Windows or Linux. Provides COBOL JVM support and enhanced compatibility with ACUCOBOL. Can be used standalone for developing local applications or in conjunction with the Development Hub to develop remote projects in Linux and UNIX.
- Visual COBOL Development Hub - provides a rich desktop development environment based on the Eclipse IDE with high-performance server-based tools for managing builds, source code access and debugger engine. The Development Hub also provides a central administration site, simplifying the development tool distribution and maintenance process.

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.



Important: Application executables that were compiled using earlier Micro Focus products must be recompiled from the sources using Visual COBOL.

Installation



Note: If you are installing on Solaris, please read [UNIX Installer Issues](#) first.

Installing Visual COBOL on Windows

System Requirements for Visual COBOL for Eclipse (Windows)

Hardware Requirements

The disk space requirements for Windows are:

- Approximately 42MB for the Sentinel RMS license server.
- At least 680MB for Visual COBOL.



Note: This includes the space needed to cache information locally so that you can modify the installation without the original source media.

The disk space requirements for UNIX/Linux are:

- Between 26 and 35 MB for the Sentinel RMS license server depending on the platform.
- Between 206 and 427 MB for Visual COBOL depending on the platform.



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

Operating Systems Supported



Note: You can produce 64-bit and 32-bit applications on 64-bit operating systems.

The supported Windows platforms are:

- 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

Software Requirements



Note: If you use the Micro Focus Web Installer to install this product, it checks your system and installs the missing prerequisite software.

- If you use XML Extensions, it 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.



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

If you have a network server, you must update the license server before installing the product as the client is not able to communicate with license servers of versions older than 10000.2.660. On Windows, you can check the version of your license server by clicking **Help > About** in the Micro Focus Licensing System Administration tool. 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>.

Installing Visual COBOL for Eclipse (Windows)

Installation Restrictions and Requirements

Before starting the installation, you should consider the following:

- Visual COBOL and COBOL Server cannot co-exist on the same machine.
- Visual COBOL and Enterprise Developer cannot co-exist on the same machine.

Downloading the Product

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

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

Installing

To use the Web Installer:

1. Double-click the `VisualCOBOLEclipse21_webinstaller.exe` file.
2. Click **Start** in the Installer window and follow the instructions to install the prerequisite software and the product.

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

1. Run the `VisualCOBOLEclipse21.exe` file and follow the wizard instructions to complete the installation.

A full version of Eclipse, with the Micro Focus plugins already installed, will be present in the `C:\Users\Public\Micro Focus\Product Name\eclipse` directory. On older versions of Windows, Eclipse might be installed in a different folder, such as for Windows XP, in `install_dir\eclipse`.



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 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 for 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.

Installing Into Different Eclipse Packages

Micro Focus Visual COBOL uses Eclipse 3.7. If you have an existing Eclipse 3.7 installation, no further action is necessary, but if you want to use Visual COBOL in other Eclipse packages based on version 3.6, you must also install the Visual COBOL update site, and the RSE and AspectJ plugins. Follow the procedure below to do this.



Important: We recommend you back up all existing Eclipse configuration files first.

1. Install Visual COBOL as directed above.
2. Copy the required Visual COBOL resources to your existing Eclipse as follows:

For Windows environments

1. Extract the contents of the following .zip files in %ProgramFiles%\Micro Focus\Visual COBOL\eclipse\installer to a temporary folder:
 - ajdt_2.1.3_for_eclipse_3.7.zip
 - RSE-runtime-3.3.zip
 2. Copy the extracted folders to *<your existing Eclipse>\eclipse\dropins*.
 3. Copy %ProgramFiles%\Micro Focus\Visual COBOLUpdateSite to *<your existing Eclipse>\eclipse\dropins*.
 4. Rename each of the folders:
 - From ajdt_2.1.3_for_eclipse_3.7 to AJDTUpdateSite
 - From RSE-runtime-3.3 to RSEUpdateSite
 - From UpdateSite to COBOLUpdateSite
 5. Move the contents of the *<your existing Eclipse>\eclipse\dropins\RSEUpdateSite\eclipse* folder up a level (that is, to *<your existing Eclipse>\eclipse\dropins\RSEUpdateSite*) and delete the empty folder.
 6. Copy the eclipse.ini file in %ProgramFiles%\Micro Focus\Visual COBOL\eclipse\installer to *<your existing Eclipse>\eclipse*, overwriting the existing file.
3. Restart Eclipse.

After Installing

If you have used Eclipse from the same workspace before, the Eclipse perspective settings are not reset after installing any Micro Focus product. 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 X Windows on Windows

Some features of Visual COBOL for Eclipse on Windows require an X Windows installation, hence Micro Focus ViewNowX is provided with the product. To install, run the file ViewNow_X_Server.exe in your

Visual COBOL installation. By default this will be in the %ProgramFiles%\Micro Focus\Visual COBOL\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.

Repairing

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 versions of Windows Vista or later:

1. From the **Control Panel**, click **Uninstall a program** under **Programs**.
2. Right-click your Micro Focus product and select **Repair**.

To repair your installation on older versions of Windows, such as 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**.

Uninstalling

To uninstall the product, you cannot simply delete its files from your hard disk. To uninstall the product:

1. Log in with the same user-ID as you used when you installed the product.
2. Click **Uninstall a program** under **Programs** (or **Add/Remove Programs** on older versions of Windows) in **Control Panel**.
3. On older versions of Windows such as Windows XP, ensure that **Show Updates** (at the top of the Add or Remove Programs dialog) is checked, so that any hot fixes or WrapPacks are listed.
4. On versions of Windows Vista and later, click **View installed updates** in the left-hand pane.
5. Select the product and click **Remove** or **Uninstall**, respectively.

When you uninstall, the only files deleted are those that the installation software installed. If the product directory has not been removed, delete any unwanted files and subdirectories within it using Windows Explorer.



Note: The installer creates separate installations for Micro Focus Visual COBOL and Micro Focus License Manager. Uninstalling only Visual COBOL 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.

Installing on UNIX/Linux

System Requirements for Visual COBOL for Eclipse (UNIX)

Hardware Requirements

The disk space requirements are:

- 33MB for the Sentinel RMS license server.
- Approximately 399MB for Visual COBOL.



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

Operating Systems Supported



Note: You can produce 64-bit and 32-bit applications on 64-bit operating systems.

- x86-64 running Red Hat Linux 5.8, 6.1, 6.2 - 32/64-bit
- Oracle Linux 6 Update 2 with Unbreakable Enterprise Kernel Release 2
- x86-64 running SuSE SLES 11, 11 SP2 - 32/64-bit

Software Requirements

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

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

GNU Standard C++ Library - libstdc++(i686 version)

The object files for development using standard C libraries – glibc-devel (i686 version)

Check the [Red Hat Web site](#) for more information.

- To use the Web installer on Red Hat Enterprise Linux 6.1, you must have the following bug fix updates for Red Hat installed:

glibc-2.12-1.25.el6_1.3.i686.rpm

openldap-2.4.23-15.el6.i686.rpm

nss-pam-ldapd-0.7.5-7.el6.i686.rpm

zlib-1.2.3-25.el6.i686.rpm

nss-3.12.9-9.el6.i686.rpm

nss-util-3.12.9-1.el6.i686.rpm

cyrus-sasl-lib-2.1.23-8.el6.i686.rpm

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

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



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

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

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



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

If you have a network server, you must update the license server before installing the product as the client is not able to communicate with license servers of versions older than 10000.2.660. On Windows, you can check the version of your license server by clicking **Help > About** in the Micro Focus Licensing System Administration tool. To check the version of the license server on UNIX, run `/var/microfocuslicensing/bin/mfcesver` or `/var/microfocuslicensing/bin/cesadmintool.sh`.

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

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

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

```
JAVA_HOME=java_install_dir
```

where *java_install_dir* is the path to the JAVA installation directory such as `/usr/java/javan.n`

- Add \$JAVA_HOME/bin to your system PATH variable. To do this, execute:

```
export PATH=$JAVA_HOME/bin:$PATH
```

- Set the LANG environment variable to pick up localized messages. The LANG settings are English and Japanese only.

Installing Visual COBOL for Eclipse (UNIX)

Downloading the Product

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

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

Installing

To use the Web Installer:

1. Give the Web installer file execute permissions:

```
chmod +x webinstaller_visualcobol_deveclipse_2.1_platform
```

2. Run the installer with superuser permissions:

```
./webinstaller_visualcobol_deveclipse_2.1_platform
```

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

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

1. Give execute permissions to the setup file:

```
chmod +x setup_visualcobol_deveclipse_2.1_platform
```

2. Run the installer with superuser permissions:

```
./setup_visualcobol_deveclipse_2.1_platform
```

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

The COBOL environment is installed by default into `/opt/microfocus/VisualCOBOL`. A full version of Eclipse, with the Micro Focus plugins already installed, is present in the `/opt/microfocus/VisualCOBOL/eclipse` directory.

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

```
./webinstaller_visualcobol_deveclipse_2.1_platform -installlocation="full path of new location"
```

or

```
./setup_visualcobol_deveclipse_2.1_platform -installlocation="full path of new location"
```

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

**Note:**

- The installation of this product could affect the SafeNet Sentinel licensed components running on your machine. During installation licensing is shutdown to allow files to be updated. To ensure the processes running on your machine are not affected, you need to use the `–skipsafenet` option, which skips the installation of SafeNet:

```
./setup_visualcobol_deveclipse_2.1_platform -skipsafenet
```

- To protect the SafeNet Sentinel installation from accidental updating you can create an empty file named `SKIP_SAFENET_INSTALL` in `/var/microfocuslicensing/` as follows:

```
touch /var/microfocuslicensing/SKIP_SAFENET_INSTALL
```

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

Set Up the Environment

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

1. To set up your product, execute:

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

2. To verify that your product is installed, execute:

```
cob -Version
```



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

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

Start the Product

To start Eclipse:

1. Double-click the product icon (automatically installed on your desktop).



Note: The installer automatically creates a shortcut icon for the product on the desktop for the user for which you ran the install script. If you need to create desktop icons for other users' desktops on the same machine, or if the icon was not created on the desktop for the root user, run the following shell script as the user you need to login as: `$COBDIR/bin/createdesktopicon.sh`

Installing as an Upgrade

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

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

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

1. Execute the following command:

```
[ as root ] mv /opt/microfocus/VisualCOBOL /opt/microfocus/VisualCOBOLR4
```

2. Install the latest version as described in the section *Installing*.

- Install the latest version in a different location and set the environment to point to it. To do this, run the Visual COBOL 2.1 installer with the `–installlocation` option:

1. Execute the following command:

```
./InstallFile -installlocation="/opt/microfocus/VisualCOBOL2.1"
```

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

```
. /opt/microfocus/VisualCOBOL2.1/bin/cobsetenv
```

Installing Into Different Eclipse Packages

Micro Focus Visual COBOL uses Eclipse 3.7. If you have an existing Eclipse 3.7 installation, no further action is necessary, but if you want to use Visual COBOL in other Eclipse packages based on version 3.6, you must also install the Visual COBOL update site, and the RSE and AspectJ plugins. Follow the procedure below to do this.



Important: We recommend you back up all existing Eclipse configuration files first.

1. Install Visual COBOL as directed above.
2. Copy the required Visual COBOL resources to your existing Eclipse as follows:

For UNIX environments

1. Extract the contents of the following `.zip` files in `/opt/microfocus/VisualCOBOL/eclipse` to a temporary folder:
 - `ajdt_2.1.3_for_eclipse_3.7.zip`
 - `RSE-runtime-3.3.zip`
 - `UpdateSite.zip`
 2. Copy the extracted folders to `<your existing Eclipse>/eclipse/dropins`.
 3. Rename each of the folders:
 - From `ajdt_2.1.3_for_eclipse_3.7` to `AspectJUpdateSite`
 - From `RSE-runtime-3.3` to `RSEUpdateSite`
 - From `UpdateSite` to `COBOLUpdateSite`
 4. Move the the contents of the `<your existing Eclipse>/eclipse/dropins/RSEUpdateSite/eclipse` folder up a level (that is, to `<your existing Eclipse>/eclipse/dropins/RSEUpdateSite`) and delete the empty folder.
 5. Copy the `eclipse.ini` file in `/opt/microfocus/VisualCOBOL/eclipse/eclipse` to `<your existing Eclipse>/eclipse`, overwriting the existing file.
 6. If necessary, edit `eclipse.ini` so that the `-vm` parameter is set to the requisite Java folder.
3. Restart Eclipse.

UNIX Installer Issues

License Infrastructure Installer

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

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

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

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

License Server

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

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

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

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

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

After Installing

If you have used Eclipse from the same workspace before, the Eclipse perspective settings are not reset after installing any Micro Focus product. 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.

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:

To do this, execute the following at the command line:

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

Repairing

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

Uninstalling

To uninstall this product:

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



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

To uninstall Micro Focus License Manager:

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

The script does not remove some of the files as they contain certain system settings or licenses. You can optionally remove the prerequisite software. For instructions, check the documentation of the respective software vendor.

System Requirements for Visual COBOL Development Hub

Hardware Requirements

The disk space requirements are:

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



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

Operating Systems Supported

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

Software Requirements

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

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

GNU Standard C++ Library - libstdc++(i686 version)

The object files for development using standard C libraries – glibc-devel (i686 version)

Check the [Red Hat Web site](#) for more information.

- To use the Web installer on Red Hat Enterprise Linux 6.1, you must have the following bug fix updates for Red Hat installed:

glibc-2.12-1.25.el6_1.3.i686.rpm

openldap-2.4.23-15.el6.i686.rpm

nss-pam-ldapd-0.7.5-7.el6.i686.rpm

zlib-1.2.3-25.el6.i686.rpm

nss-3.12.9-9.el6.i686.rpm

nss-util-3.12.9-1.el6.i686.rpm

cyrus-sasl-lib-2.1.23-8.el6.i686.rpm

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

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



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

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

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



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

If you have a network server, you must update the license server before installing the product as the client is not able to communicate with license servers of versions older than 10000.2.660. On Windows, you can check the version of your license server by clicking **Help > About** in the Micro Focus Licensing System Administration tool. To check the version of the license server on UNIX, run `/var/microfocuslicensing/bin/mfcesver` or `/var/microfocuslicensing/bin/cesadmintool.sh`.

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

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

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

```
JAVA_HOME=java_install_dir
```

where *java_install_dir* is the path to the JAVA installation directory such as `/usr/java/javan.n`

- Add `$JAVA_HOME/bin` to your system PATH variable. To do this, execute:

```
export PATH=$JAVA_HOME/bin:$PATH
```

- Set the LANG environment variable to pick up localized messages. The LANG settings are English and Japanese only.

Installing Visual COBOL Development Hub

Downloading the Product

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

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

Installing

To use the Web Installer:

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

```
chmod +x webinstaller_visualcobol_devhub_2.1_platform
```

2. Run the installer with superuser permissions:

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

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

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

1. Give execute permissions to the setup file:

```
chmod +x setup_visualcobol_devhub_2.1_platform
```


2. Run the installer with superuser permissions:

```
./setup_visualcobol_devhub_2.1_platform
```

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

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

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

```
./webinstaller_visualcobol_devhub_2.1_platform -installlocation="full path of new location"
```

or

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

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



Note:

- The installation of this product could affect the SafeNet Sentinel licensed components running on your machine. During installation licensing is shutdown to allow files to be updated. To ensure the processes running on your machine are not affected, you need to use the `-skipsafenet` option, which skips the installation of SafeNet:

```
./setup_visualcobol_devhub_2.1_platform -skipsafenet
```

- To protect the SafeNet Sentinel installation from accidental updating you can create an empty file named `SKIP_SAFENET_INSTALL` in `/var/microfocuslicensing/` as follows:

```
touch /var/microfocuslicensing/SKIP_SAFENET_INSTALL
```

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

Set Up the Environment

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

1. To set up your product, execute:

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

2. To verify that your product is installed, execute:

```
cob -Version
```



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

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



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

Installing as an Upgrade

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

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

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

1. Execute the following command:

```
[ as root ] mv /opt/microfocus/VisualCOBOL /opt/microfocus/VisualCOBOLR4
```

2. Install the latest version as described in the section *Installing*.

- Install the latest version in a different location and set the environment to point to it. To do this, run the Visual COBOL 2.1 installer with the `-installlocation` option:

1. Execute the following command:

```
./InstallFile -installlocation="/opt/microfocus/VisualCOBOL2.1"
```

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

```
./opt/microfocus/VisualCOBOL2.1/bin/cobsetenv
```

UNIX Installer Issues

License Infrastructure Installer

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

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

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

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

License Server

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

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

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


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

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

Configuring the Remote System Explorer Support

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

Starting the Daemon

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

- a version of Perl
- a version of Java
- the `as` (assembler) and `ld` (linking) programs on the path, as specified by the `PATH` environment variable

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

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

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

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

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

where:

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

For example,

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

This command will start a daemon listening on port 4999 and will use random server ports.

- `<low port>-<high port>` is the range of ports on which the servers (launched by the daemon) should use to communicate with Eclipse on the client machine.

For example,

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

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

Stopping the Daemon

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

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

Configuring the Environment

You may need to configure some aspects of the environment before you start the daemon. This is because when a build or debug session is initiated on the Development 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.

Repairing

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

Uninstalling

To uninstall this product:

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



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

To uninstall Micro Focus License Manager:

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

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

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

Licensing Information



Note:

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

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 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 XSERV964 license key is supplied for ViewNow X Server.

What's New

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

New Features in Visual COBOL 2.1

ACUCOBOL-GT Data Types in Managed Code

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

ACUCOBOL-GT Library Routines in Managed Code

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

Automatic Directives Detection and Setting

The IDE automatically determines and sets the COBOL dialect, and the CICS and SQL directives on local or remote native COBOL projects. You can also start a directives scan from within COBOL Explorer - select **Determine Directives** from the context menu for the projects or the COBOL source files. This triggers a scan to determine the COBOL dialect, the CICS and SQL settings, and sets them as Compiler directives on a file or project level respectively. At the end of the scan, you view the results and choose to apply the changes.

Compiler Directives

The following new Compiler directives are now available

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

The following Compiler directives have changed:

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

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

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

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

Just-in-time debugging

Visual COBOL now supports "Just-in-time" debugging: when a run-time error occurs, or an application calls `CBL_DEBUGBREAK`, the IDE can start automatically with the debugger attached to the failed process.

JVM Class Library

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

The library contains the following classes:

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

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

Managed COBOL Enhancements

Delegates and Events	Delegates and events are now implemented on the JVM platform. This release provides support for combining delegates, using the <code>METHOD</code> keyword to specify method groups, and implicit conversion from a method group or an anonymous method to the suitable delegate type.
Handling Invalid Numeric Data	The handling of invalid numeric data is controlled by a number of Compiler directives: <code>HOSTNUMMOVE</code> , <code>HOSTNUMCOMPARE</code> and <code>SIGNFIXUP</code> . These directives were previously only available in native code but are now supported in managed code.
Resolving Types	In this release, the Compiler attempts to resolve types to those defined in the current compilation unit wherever possible. The Compiler will attempt to resolve such types to an external name only if no suitable type exists in the current compilation unit. For example: <pre>\$set ilusing"System" class-id MyNamespace.EventHandler. 01 o type EventHandler. end class.</pre> In this release, <code>01 o type EventHandler.</code> resolves to <code>MyNamespace.EventHandler</code> and not to <code>System.EventHandler</code> .
Specifying Properties	In previous versions of the products, properties declared using the <code>PROPERTY</code> keyword on a data item were generated as final properties. Starting with this release, they are generated as virtual properties by default. In order to make the properties final, you need to specify the word <code>FINAL</code> following <code>PROPERTY</code> . This change may affect the generation of Proxy classes, for example, if you are using WCF.

OpenESQL

SQL Compiler Directive Options	OpenESQL has been enhanced to support the the following new SQL compiler directive options:
DATE	Controls the reformatting of date values in output parameters and in input parameter character host variables when DETECTDATE is also specified.
TIME	Controls the reformatting of date values in output parameters and in input parameter character host variables when DETECTDATE is also used.
DATEDELIM	Specifies a single character as the delimiter between the year, month, and day components to override the default delimiter determined by the HCOSS DIALECT or DATE directive specification.
TIMEDELIM	Specifies a single character as the delimiter between the hour, minute, and second components to override the default delimiter determined by the HCOSS DIALECT or TIME directive specification.
TSTAMPSEP	Specifies a single character as the separator between the date and time parts of timestamp and date/time data.

SQL Server We now support Microsoft SQL Server 2012.

UNIX Platforms Support

Support for remote development and deployment of projects has been added for the Linux/390 platform. Development Hub now supports Oracle Linux 6 Update 2 with Unbreakable Enterprise Kernel Release 2.

Features Added in Visual COBOL 2.0

Automatic Directives Detection and Setting

It is now possible to automatically determine and set the COBOL dialect and SQL directives on native COBOL source code. In the COBOL Explorer, select **Determine Directives** from the context menu for the projects or the COBOL source files. This triggers a scan to determine the COBOL dialect and SQL settings and sets them as Compiler directives on a file or project level respectively. At the end of the scan, you view the results and choose to apply the changes.

This feature works with local or remote native COBOL projects only.

COBOL Explorer View


Visual COBOL now includes the COBOL Explorer view in the Eclipse IDE. This allows you to navigate around COBOL projects in a more useful and convenient way than Eclipse's own Navigator view.

The COBOL Explorer view includes the following features to help you manage your projects:

- For COBOL Project, Remote Project, and Mainframe Project types, COBOL Explorer adds category folders that automatically group together your project's COBOL programs, copybooks, and output files. (These folders are not physical folders on the disk, but effectively headings for certain file types.)
- To simplify navigation around your project, you can hide some files that are part of the project, such as the `.cobolBuild`, `.cobolProj`, and `.project` files, and the `.settings` folder and its content in

JVM projects. You do this by opening the view menu, selecting **Customize View** and choosing from the options.

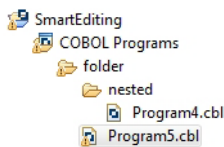
Filters Choose types of content to hide in COBOL Explorer.

Option	Description
.*resources	Any files of type . * such as .cobolBuild, .cobolProj
Closed projects	Resources in closed projects never display, but you can choose to hide all closed project icons  too
Empty folders inside COBOL categories	Folders mapped to the COBOL Programs and Copybooks that contain no COBOL program or copybook files
Non-COBOL projects	Projects of types other than those under Micro Focus COBOL - for example AspectJ, Plug-in, or Update-site projects
RSE Internal projects	Projects generated by the RSE plug-in

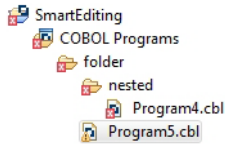
Content Choose types of content to show in COBOL Explorer.

Option	Description
Working Sets	Working sets are subsets of workspace resources you can choose to show or perform options on. To define a working set, click Project > Build Working Set > Select Working Set > New .
COBOL Elements	Non-resource types that are COBOL-specific: <ol style="list-style-type: none"> x category folders, including those for COBOL programs and copybooks icons for the different types of COBOL files. overlays for build errors and warnings some context menu items
Resources	COBOL projects, COBOL programs and copybooks. Has effect only if the COBOL Elements option is unchecked.

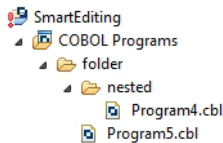
- COBOL Explorer helps you fix problems by using icons to identify files and containers that cause build errors and warnings.
- A file that generates a warning, and any containing folders and categories, is marked with a yellow warning sign. For example:



- A file that causes an error, and any containing folders and categories, is marked with a cross. In this example the icons indicate the most severe problem is the error caused by `Program4.cbl`, and the folders are marked with error icons despite `Program5.cbl` generating a warning:



- A project that suffers from a dependency error is marked with a red exclamation mark; its contents remain unmarked. In this example the project depends on another project that is closed, causing a build path problem:



Also, the context menus are reorganized and include some additional tasks such as the **Determine Directives** command.

You access COBOL Explorer in the same way as other Eclipse views, by selecting **Window > Show View**.

COBOL File Search

Visual COBOL now includes the COBOL File Search feature in the Eclipse IDE.

There are two ways to use the COBOL File Search feature to make it easier to find certain files within your projects.

- Select **Search > COBOL File**, and type the filename search string.
You can filter results based on
 - file type: choose from COBOL programs, copybooks.
 - only those files that cause build errors
 - only those files that cause build warnings
- Select **Search > Run Stored COBOL Search Query** and choose to find either:
 - only those files that cause build errors
 - only those files whose properties override the project's properties (for example **Language dialect** or **Compile for debugging** settings).

The results are shown in the Search view.

You can edit the properties of multiple files in the search results simultaneously if they come from the same project. Highlight the files you want to change, right-click and select **Properties**, and make sure **Enable file specific settings** is selected.



Tip: You can save your search criteria and give it a label, which adds it to the list of stored queries available when you select **Search > Run Stored COBOL Search Query**. When saving a search you can also assign it a keyboard shortcut by entering the required keystrokes in the **Binding** text box.

Save and edit your search either in the Search view, or the **Windows > Preferences > Micro Focus COBOL > Search** dialog box.

Make sure you don't open the **Keys** preferences page (**Windows > Preferences > General > Keys**) before or after you edit the binding in the **Search** preferences page, and before clicking **OK** to save any changes. Opening the **Keys** preferences page will prevent any changes you make to the search bindings being saved.

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.

Compiling Single Files

It is now possible to compile individual COBOL source files without rebuilding your entire project. **Build Automatically** on the **Project** menu must be turned off.

Enhancements to Developing Applications on a Remote Machine

Previously, this product depended on Samba or NFS to transfer and manage the project files on a remote UNIX machine. Remote Server Explorer (RSE) was only used to build the applications on remote machines.

"Remote file system (RSE)" is now the default option in the Create Remote COBOL project wizard for managing the files of remote COBOL projects. RSE establishes a connection with the remote machine and is used to transfer and manage the project files on it.



Important:

- You must use SSH connection and not DStore for remote COBOL projects created with Remote file system (RSE).
- You can still use Samba and NFS as file system providers. On system where SSH is not allowed, you can use SAMBA and DStore connection to transfer and manage the files on the remote system.

Converting Net Express Projects



Note: This topic applies to Windows environments only.

It is now possible to convert Net Express projects into Eclipse projects and use them with Visual COBOL. To do this, use the Net Express Project Import and Convert Wizard available from **File > Import > Convert NetExpress Projects to COBOL Projects**. The wizard analyzes the Net Express project file and its configuration settings, creates Eclipse projects based on this information, imports the existing source code into them, and sets the requisite project and file properties from the original Net Express project.



Note: If you import a Net Express project that uses functionality that is not supported in Visual COBOL, the wizard will still produce an Eclipse project. In some cases you might be able to perform additional steps (such as editing source files, installing AddPacks, or reworking parts of the

application) in order to successfully compile and run it. See *Converting Net Express Projects to Eclipse Projects* for more information about the limitations of the wizard.


Copybook Context

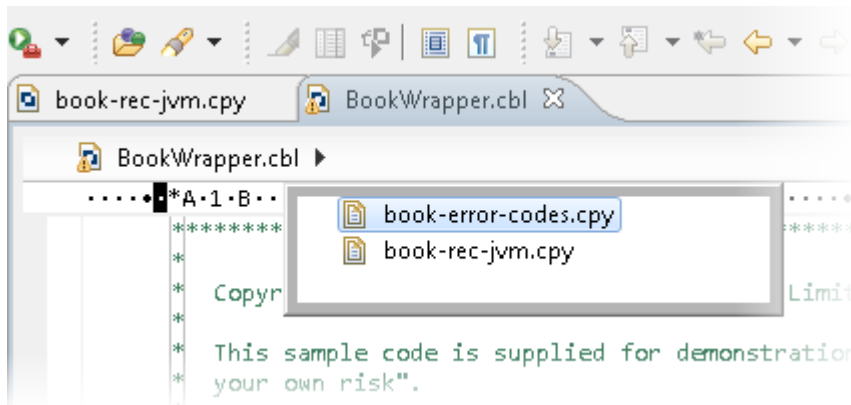
When you view or edit a copybook, you need to see it in the context of the program that references it, as its appearance and usage can change depending on the program. For example, data items that are not used in the program are struck through, and horizontal lines indicating the start and end of code lines can show depending on the source format of the program. The Program Outline view can differ between contexts too.

You can choose which context to open a copybook by right-clicking on the file in COBOL Explorer and selecting **Open In Context**. A list is displayed of all the programs that include a COPY statement referencing the copybook. You can also choose to open the copybook in no particular context.

Whichever you choose becomes the copybook's context whenever you right-click the copybook in COBOL Explorer and select **Open**. It also determines which program it is displayed within when you right-click in the copybook source and select **Open in Copy View**.

A breadcrumb trail at the top of the Editor indicates which program and copybooks are the current context -

click Toggle Context Breadcrumbs  in the Eclipse toolbar and use the drop-down list to view and open the dependent copybooks.



Debugging Enhancements


This release provides the following enhancements to debugging:

Program Breakpoints



Note: Program breakpoints are supported in native COBOL only, and are not supported with nested programs.

You can now set program breakpoints, which break into a program whenever it or one of its entry points is called.

To set or toggle a program breakpoint, double-click in the left margin, next to the Procedure Division heading. The  icon appears when the program breakpoint is set.

Step out of OSVS perform statements

You can now step out of a perform statement when PERFORM-TYPE(OSVS) or DIALECT(OSVS) is set.

Eclipse 3.7

This release of Visual COBOL uses Eclipse 3.7.1 Indigo.

JVM COBOL File Handler

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

Library Routines

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

```
CBL_CTF_COMP_PROPERTY_GET
CBL_CTF_TRACE
CBL_CTF_TRACER_LEVEL_GET
CBL_CTF_TRACER_GET
CBL_CTF_LEVEL
```

The following routine has been enhanced:

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

Managed COBOL Language Features

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

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

The Help now contains code examples comparing managed COBOL and Java.

Data Access

This release provides the following enhancements:

- Improved IDE integration with SQL directives - now supports handling of deprecated and removed directives. Also supports filtering of the choices offered to the user by product type, project type, and platform.

- OpenESQL has been enhanced and it now:
 - defaults to optimal performance
 - supports 64bit ODBC across all platforms
 - OpenESQL now supports JDBC across all platforms

Remote COBOL JVM Projects - Early Release

You can use Eclipse to develop COBOL JVM projects on a remote UNIX/Linux machine. The source code resides and is being compiled and debugged on the remote machine.

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.

Samples

The following new samples are now available:

- Collections - demonstrates the managed COBOL collections syntax
- Local Variables - shows how to declare data items in the procedure division in the `DECLARE`, `PERFORM` and `TRY` statements
- The code in the Properties sample has been enhanced to use the new `PROPERTY-ID` syntax. The sample also includes a sample program for Indexers which illustrates the new `INDEXER-ID` syntax.

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

On Windows, Help for the following OO COBOL class libraries is available:

- Base class library
- GUI class library
- OLE class library
- OLE Automation class library

The Help is available in the file `nxrclr.chm`, which is installed in the Help folder of your installation. The default location is `%ProgramFiles%\Micro Focus\Visual COBOL\Help`.

To open the help, double-click `nxrclr.chm` in Windows Explorer.

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*.

OpenESQL

OpenESQL now includes the JDBC preprocessor option that you can use to access databases for applications running under the Java Virtual Machine (JVM).

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$CALLED`
 - `C$CALLED`
 - `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 JVM COBOL and native COBOL
- 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

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.

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 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

Please refer to the *Known Errors and Restrictions* topic in the *Product Information* section of your product Help.

In addition, please note the following:

COBOL Watchpoints

The debugger ignores a COBOL watchpoint that is hit if there is no statement following the statement that modifies the data on which that watchpoint is set.

Co-existing with Earlier Micro Focus Products

Run-time system error due to COBCONFIG A run-time system error occurs if either the COBCONFIG or 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 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.

Debugging

There is an issue with "Wait for attachment" when you use Visual COBOL for Eclipse to debug applications that run on some Linux/Unix platforms. Eclipse connects to the debugger on the remote machine, but might not attach to the process to debug the code.

To work around this issue, ensure that on the remote machine the TMPDIR environment variable is unset or has the same value for both Micro Focus Visual COBOL Development Hub server and for the running process you wish to debug. The Micro Focus Visual COBOL Development Hub server is the server which you started either directly with the \$COBDIR/remotedev/startdoserver script or indirectly using the daemon which is started with the \$COBDIR/remotedev/startdodaemon script.

To check the variable used by the Micro Focus Visual COBOL Development Hub server:

1. Open Remote Systems view in Eclipse on the Windows machine.
2. Right-click the **Shells** element of the server connection to be tested and click **Launch Shell**.
3. In the **Remote Shell** view, type `echo $TMPDIR` in the **Command** field, and press **Enter**.

The value of the TMPDIR environment variable is shown - ensure it is the same as the one used by the process to be debugged.



Note: The value of TMPDIR used by the Development Hub server cannot be changed in the remote shell and must be set before you start the daemon or server.

File Handling

When using a CGI/ISAPI application developed with the Net Express HTML Forms Designer in Visual COBOL, you may receive an error “unresolved external symbol _NMCNVRTI” or “unresolved external symbol _NMCNVRT0”.

To work around this issue, add the corresponding .obj file to the **Additional Directives** in the project properties, or use them from the command line to link the application: `cbllink filename.cbl NMCNVRTI.OBJ` (or `cbllink filename.cbl NMCNVRT0.OBJ`, respectively).

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).

Resolved Issues

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

- [Compiler](#)
- [Data Tools editor](#)
- [Documentation](#)
- [Eclipse IDE](#)
- [File Handling - External File Handler](#)
- [File Handling - Fileshare](#)
- [File Handling - Sort / JCL Sort](#)
- [JVM Compiler](#)
- [JVM Run-Time System](#)
- [MVS REXX Emulation](#)
- [Run-Time System](#)
- [SQL: COBSQL](#)
- [SQL: DB2 ECM](#)
- [SQL: OpenESQL](#)
- [Vision File System](#)
- [XDB DCLGEN Utility](#)
- [XDB Server](#)
- [XML syntax support runtime](#)

Compiler

- The DBCSSOSI directive now compiles correctly.
590357 ()
- DISPLAY MESSAGE BOX ... RETURNING syntax now returns an E-level error message stating that it is unsupported in this COBOL system.
584600 ()
- You can now configure the DATAMAP clause to output either addresses or offsets. DATAMAP(ADDR) or DATAMAP with no option shows addresses. DATAMAP(OFFSET) shows offsets. Compile managed COBOL code sets DATAMAP(OFFSET) at the end of directive processing as addresses are not available for managed code.
1054182 (2093752)
- Compilation of ACU DISPLAY windowing syntax will no longer receive spurious internal error messages during a background syntax check.
591227 ()
- Compiling with a large number of ADDSYN directives specified now works as expected.
1083917 (2567767)
- The maximum value for the PAGE LIMIT clause has been increased from 999 to 9999.
1084391 (2576431)
- DISPLAY(CONSOLE) is a new directive available to override the behavior set by RM or ACU directives, so that standard DISPLAY syntax is processed as standard ANSI DISPLAYs, directed to CONSOLE.

RM and ACU directives set DISPLAY(CRT) immediately, allowing you to set DISPLAY(CONSOLE) after either of those directives, if required.

1084496 (2577799)

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

1085375 (2586367)

Data Tools Editor

- The Data File Editor now allows editing of an EBCDIC data file on a DBCS OS.

1083716 (2567163)

Documentation

- Information detailing the process of XML-enabling COBOL applications that was previously left out of the Visual COBOL documentation has been reinstated and is now available under Programming > COBOL Programming > XML Input/Output.

1084142 (2573343)

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

1084692 (2578515)

Eclipse IDE

- Specifying "use(cobol.dir)" in the COBOL build settings no longer generates errors when a program is open in the Editor.

1082324 (2548143)

- The values that the Variables View shows when you debug with "Display hexadecimal values" or "Display byte values" settings enabled have been changed.

589220 ()

- You can now specify in the IDE whether directories that you add to the project should be added to the copypath. You configure this in the project's properties under Micro Focus COBOL > Build Paths, and on the Copybook Paths tab.

589096 ()

File Handling - External File Handler

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

1083646 (2566578)

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

1082362 (2506971)

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

1084329 (2573607)

- Cobfhrepro now works correctly when session id is specified.

590271 ()

- Invalid XML syntax in MF.MFFH.XML has been corrected.

589406 ()

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

1084961 (2581353)

File Handling - Fileshare

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

1084744 (2579930)

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

587706 ()

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

1083664 (2567737)

- FSVIEW will no longer process the last command in a command file twice.

589764 ()

File Handling - Sort / JCL Sort

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

1083225 (2562501)

- Using SYMNames sometimes caused SORT to use the wrong field position and lengths.

1083320 (2563738)

- Data format FI is now supported in the OUTREC edit fields section.

1083495 (2565051)

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

1084194 (2573990)

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

1084231 (2574465)

- SORT now ignores the operand 'WORK' along with its value.

1081023 (2533397)

- A sort using SORTTEMPSPACE no longer ever results in a COBRTS 252 error.

1081943 (2546898)

- CENTWIN and Y2PAST are now supported as PARMS.

1080717 (2531364)

- The MFSORT help screen now shows that OPTION is supported.

590260 ()

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

1084481 (2577157)

- The TOTAL field length calculation now works correctly when TOTALs are zeros.

1082887 (2558079)

- Trailers are now included in the SYSOUT outfil record count.

1084193 (2573272)

- Records following HEADER2 and HEADER3 will now be the correct length.
590270 ()
- Header lengths are now calculated correctly, taking line feeds into account instead of giving a SORT099I error. Headers will now be printed for outfiles that do not have any records.
1083701 (2565909)
- SORT now works correctly when SYSIN is given as LSEQ PDS Member.
1082529 (2550678)
- When using MFJTOOL with a VB input file, an appropriate error is thrown when the output file is not VB or has not worked successfully when the output file is VB. Previously, a COBRTS 139 error was thrown.
1084567 (2577626)
- Sort now works correctly when OPTION COPY is given before the SORT FIELDS in SYSIN.
1083679 (2565053)
- SORT caused different sort processing for large sort cards. This has been fixed so SORT now returns an error message and exits the sort processing when the sort card has more than 1024 INCLUDE/OMIT conditions.
1084065 (2571660)
- When using MFSORT, smaller records are padded to the size of the minimum record length of a VB file and the record length is changed to the minimum of sortout.
1082943 (2554482)
- Sort now displays HEADER2 at the start and TRAILER2 at the end of each page.
1083702 (2566545)

JVM Compiler

- Java verification errors or incorrect program flow occurring in very large JVM COBOL programs is now fixed. Previously, in very large programs, the compiler would process GO TO statement incorrectly, and could jump to the wrong program locations.
1084569 (2578716)
- Some problems with the use of nested classes defined in COBOL have been resolved.
1083183 (2561420)

JVM Run-Time System

- A stack overflow no longer occurs if you execute a call to a method inside a constructor and this method contains a COBOL statement. Previously, the runtime would jump back to the constructor and go into a loop, producing a stack overflow.
1083181 (2561418)
- The JVM COBOL library routines CBL_DIR_SCAN_ now return the correct results when searching for directories.
590261 ()

MVS REXX Emulation

- The DATE() function now formats the year correctly when converting between a Julian and a standard date format.
1083437 (2564240)
- The LISTDSI external function now correctly sets values for the SYSREFDATE and SYSMGMTCLASS variables.
1084902 (2581611)

Run-Time System

- The call MVS_REGISTER_DDNAME is an MFE-only call used by the IDE. FileHandler no longer calls this routine.
1083028 (2559292)
- IF NUMERIC validation of COMP-3 slack nibbles (when there is an even number of digits in the picture clause) is now done in a way that is compatible with that on the mainframe.
1083801 (2569340)
- The RUN program now accepts command line arguments of up to 1023 characters long.
1083215 (2561567)
- When running a full-screen application inside a terminal emulator on Linux, the actual size of the terminal is read at startup and reread when the terminal is resized. This behaviour is also supported on AIX, HP/UX, and Solaris. The Micro Focus vt220 terminfo entry now correctly describes a 24-line display. A vt220-25 terminfo entry is included for compatibility with the previous behaviour.
1084817 (2579335)

SQL: COBSQL

- The Cobsql preprocessor was updated to no longer misinterpret WORKING-STORAGE items whose definitions were spread across multiple source lines.
1080078 (2507684)
- The Cobsql preprocessor has been updated to correctly process variables defined as USAGE COMP after an EXEC SQL INCLUDE SQLCA when the CP preprocessor directive NOSQL is specified.
1084463 (2577593)
- The Cobsql preprocessor has been updated to correctly process Pro*COBOL-generated data items when the first WORKING-STORAGE variable in user code contains a VALUE clause with the literal value on a separate source line.
1084753 (2579264)

SQL: DB2 ECM

- The DB2 pre-compiler now generates GOBACK instead of STOP RUN at the end of program source, so that poorly coded programs do not fail when running under IMS or other transaction monitors.
1083235 (2559616)

SQL: OpenESQL

- The OpenESQL preprocessor has been updated such that when a program or application is compiled with the SQL(DBMAN=ADO) compiler directive, the calls generated do not modify user data when a null value is returned.
1083563 (2561563)
- The OpenESQL preprocessor no longer accepts singleton select statements that have no INTO clause, or host variables that are not preceded by a colon, apart from SQLDA references in dynamic SQL. In rare cases where a singleton select without an INTO clause is required, code it by placing the SELECT statement inside of a BEGIN/BEND block.
1084430 (2576585)
- The ODBC Compiler did not identify specialized classes such as ENUMS, and incorrectly generated SQL interface code for them that caused Compiler errors. The ODBC Compiler has been enhanced so that it no longer generates SQL interface code for those classes.
1083144 (2561349)

- Previously, the SQL pre-compilers did not always recognize the EXEC SQL INCLUDE statements during the syntax checking phase and returned incorrect error messages.

1085057 (2582967)

Vision File System

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

1082469 (2553438)

XDB DCLGEN Utility

- A protection violation sometimes occurred when you used the XDB Declaration Generator tool on 64-bit Windows platforms.

591030 ()

XDB Server

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

1079821 (2508551)

- A mutex deadly embrace no longer results in a server hang.

1085199 (2583889)

XML syntax support runtime

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

1085342 (2586384)

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

1084119 (2572393)

Updates and SupportLine

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

Further Information and Product Support

Additional technical information or advice is available from several sources.

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

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

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



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

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

Information We Need

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

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

On Windows, if you are reporting a protection violation you might be asked to provide a dump (`.dump`) file. To produce a dump file you use the **Unexpected Error** dialog box that is displayed when a protection violation occurs. Unless requested by Micro Focus 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 Micro Focus 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. 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.