



Cumulative Changes from Server Express to Micro Focus Visual COBOL for Eclipse (UNIX)

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

Copyright © Micro Focus 2018. 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.

2018-06-07

Contents

Cumulative changes from Server Express to Visual COBOL for Eclipse (UNIX)

.....	4
About this Guide	4
What was New	4
What was New in Visual COBOL 4.0	4
What was New in Visual COBOL 3.0	10
What was New in Visual COBOL 2.3 Update 2	16
What was New in Visual COBOL 2.3 Update 1	18
What was New in Visual COBOL 2.3	21
What was New in Visual COBOL 2.2 Update 2	28
What was New in Visual COBOL 2.2 Update 1	30
What was New in Visual COBOL 2.2	35
What was New in Visual COBOL 2.1 Update 1	40
What was New in Visual COBOL 2.1	41
What was New in Visual COBOL 2.0	43
What was New in Visual COBOL 2010	50
Significant Changes	53
Significant Changes in Visual COBOL 4.0	53
Significant Changes in Visual COBOL 3.0	58
Significant Changes in Visual COBOL 2.3 Update 2	60
Significant Changes in Visual COBOL 2.3 Update 1	61
Significant Changes in Visual COBOL 2.3	64
Significant Changes in Visual COBOL 2.2 Update 2	66
Significant Changes in Visual COBOL 2.2 Update 1	67
Significant changes in Visual COBOL 2.2	67
Significant Changes in Visual COBOL 2.1 Update 1	69
Significant Changes in Visual COBOL 2.1	69
Significant Changes in Visual COBOL 2.0	69
Unsupported or Deprecated Functionality	70
Unsupported or Deprecated at Visual COBOL 4.0	70
Unsupported or Deprecated at Visual COBOL 3.0	70
Unsupported or Deprecated at Visual COBOL 2.3 Update 2	71
Unsupported or Deprecated at Visual COBOL 2.3 Update 1	71
Unsupported or Deprecated at Visual COBOL 2010	71
Known Errors and Restrictions	72
Upgrading from Server Express	74
Advantages of Upgrading to Visual COBOL	75
Upgrading Your COBOL Application	75
Best Practice in Development Using Eclipse	79
Modernize Your Applications and Processes	80
Procedural COBOL Compared with Managed COBOL	81

Cumulative changes from Server Express to Visual COBOL for Eclipse (UNIX)

Welcome to Visual COBOL for Eclipse. This document combines information on Visual COBOL for Eclipse releases from the first release, Visual COBOL 2010 R1, to the most recent release. This information is taken from the various releases' Release Notes and other sources, and brought together here for your convenience.

You can use this document:

- If your Visual COBOL installation is up to date, to see the changes made in the latest release.
- If you already use Visual COBOL but your installation is not entirely up to date, to see the changes made over several updates, together in one place.
- If you are migrating from Server Express, to see everything that has changed in the life of Visual COBOL for Eclipse.

About this Guide

We recommend that you read all the sections of this publication, looking for information on the release that is most relevant to your needs. The main sections are as follows:

What was New	This section describes new functionality that was introduced in each successive release.
Significant Changes	This section describes, for each successive release, the changes in behavior or usage that could affect the behavior of existing applications or impact the way the tools are used.
Unsupported or Deprecated Functionality	This section describes any functionality that was discontinued or deprecated at a given release.
Known Errors and Restrictions	This section describes known errors affecting the latest release, and restrictions in its use.
Upgrading from Server Express to Visual COBOL	This section gives guidance on upgrading from earlier Micro Focus products.

Within each section, we recommend reading in the reverse chronological order in which it is presented here - that is, reading about the most recent release first, and then going back through the development of the products. That way, you can more easily see if anything was added in an earlier release and subsequently removed.

What was New

This section describes the new features that were introduced in each successive release of Visual COBOL.

What was New in Visual COBOL 4.0

This release provides enhancements in the following areas:

- [Integration with Eclipse](#)
- [Application Server JCA Support for Enterprise Server](#)

- [Application Workflow Manager](#)
- [Code Coverage](#)
- [Codeset support](#)
- [Compiler directives](#)
- [Data File Tools](#)
- [Debugging](#)
- [Docker](#)
- [Documentation on working with large applications](#)
- [Enterprise Server](#)
- [File Handler](#)
- [Managed COBOL](#)
- [Library routines](#)
- [Micro Focus Unit Test Framework](#)
- [OpenESQL](#)
- [Platform support](#)

Integration with Eclipse

[Back to Top](#)

This release provides the following general enhancements:

- Support for Eclipse 4.7 Oxygen (64-bit) and 64-bit project templates - the 64-bit Eclipse 4.7 is now installed by default. Project templates are available for both 32-bit and 64-bit applications. You can set a default project template (either 32-bit or 64-bit) to use so that all future new projects will use that as the preference. On platforms only capable of building to 64-bit, the 64-bit templates are the default ones.

The 32-bit Eclipse is still supported on 64-bit Windows platforms, however you need to manually install it and a 32-bit Java.

Support for Eclipse 4.4 and 4.5 has been discontinued.

- Support for SUSE 12 - only the 64-bit SUSE 12 platform is supported and you can only use the 64-bit Eclipse with 64-bit projects on it.
- (Native COBOL projects only) The **Project Settings** page in the project's properties now include a search option. This enables you to find files and build configurations that have settings different from the ones set at project level.
- **Remove File Directives** - a context menu command in the explorer view enables you to reset a file's directives.
- IVP diagnostic tool enhancements - it is now possible to run the server-side IVP diagnostic tool for diagnosing issues with a remote connection from within Eclipse installed at the client side. The IVP tool now also performs a check for whether XTERM is installed on the remote machine.
- IMTK service mappings are now automatically regenerated if you make a change to the interface fields in the underlying program.

The following enhancements are available for JVM COBOL projects and package handling:

- New wizards for creating ENUM, DELEGATE and VALUETYPE types.
- Additional code snippets for Method-Id and Property-Id.
- Showing and grouping packages in the COBOL Explorer view - use the **COBOL JVM Project Presentation** command from the **View** menu in the explorer to either display COBOL categories or packages.
- Refactoring by renaming the package name - available from the context menus in the COBOL Editor and in COBOL Explorer.
- Refactoring by moving source folders, packages or compilation units - available from the explorer context menu.
- Support for switching off the package name mapping using the **Each part of the package name corresponds to a subdirectory** option in the **Build Configuration** page.

- Support for packaging .class files in a .jar file when building COBOL JVM projects.
- **Open Type Hierarchy** and **Open Call Hierarchy** context menu commands - available in the COBOL editor, COBOL Explorer and in the outline views while seeing the packages presentation in the IDE.

The following editor improvements are available:

- A **Properties** context menu command - enables you to access a file's properties directly from the editor.
- A **Show In** context menu command - enables you to locate the file in the COBOL Explorer.
- **Edit > Convert Tabs to Spaces** command - enables you to convert any tabs in your COBOL source files to a specified number of spaces.

Automatic relinking of applications created with Visual COBOL 3.0:

- Visual COBOL 4.0 can automatically relink existing projects created with Visual COBOL 3.0 that have executable link artefacts. Eclipse displays a warning in the Problems view that the project requires relinking. It then offers a Quick Fix action for you to execute that will link your project with the most recent version of the run-time system.

Application Server JCA Support for Enterprise Server

[Back to Top](#)

This release includes the following enhancements:

- COBOL Resource Adapters now support WebSphere 9.0 and WebLogic 12.2.1.
- Tomcat 7.0 support for servlet generation with J2SEBeans.
- NullSearch utility - for COBOL resource adapters, this new utility provides assistance in locating NULL fields in mappings passed to Enterprise Server. When a large number of arguments is provided in the parameters passed to Enterprise Server, it is difficult to locate NULL fields, which are not allowed. The NullSearch utility isolates NULL fields, so the Java application can be corrected.

Application Workflow Manager

[Back to Top](#)

Improvements have been made in the following areas:

- Model editor - now includes:
 - A new **Edit** action. Enables you to edit any model component instead of working with the Properties view. The new action enables you to change the attribute values of several model components at a time.
 - An Outline and a Relationship Hierarchy view.
 - Improved backwards and forwards navigation in the model.
 - Automated sequence numbering in the model.
 - The context menu of the model editor has been restructured for more clarity.
 - Context help.
 - Tool creation has been improved. Dependent components are generated from the function package tool definition.
- Development of models:
 - Newly created empty models now contain all categories. The new models also validate without any errors or warnings.
 - A new AWM system type, "Custom System", is available. This system type supports the model development process by facilitating the creation, change, load and reload of a model.
- New modelling features:
 - The "Return Property Value" tool in the function package now supports several additional use cases.
 - The "Transfer File" tool in the function package offers improved support for mass processing.

- Support for properties with an application-wide scope.
- A modelling capability to structure the local file cache for copybooks and include files downloaded by the background parser in the editor. Additional tools are provided to check the existence of or clear the file cache for copybooks and include files.
- The product help includes some tutorials showing how to create new models and extend the functionality of existing models. See the *Tutorials Guide*.
- A number of template models are now available and can be used as a base to develop your own models.

Code coverage

[Back to Top](#)

This release provides the following enhancements:

- Support has been added in Eclipse for code coverage for procedural copybooks.

Codeset support

[Back to Top](#)

Support has been added to enable codeset mapping to additionally be configured to use IBM's Conversion Tables directly instead of the Micro Focus supplied tables. You need to download IBM's conversion tables from IBM's Web site. Then you can use the MFCODESET environment variable to convert between IBM's CCSIDs.

Compiler directives

[Back to Top](#)

The following Compiler directives are new in this release:

- **DISPSIGN** - determines the display output of numeric fields with included signs, under an IBM mainframe dialect only.
- **GNTLITLINKSTD** - stops the suppression of call-convention 8 when both call-convention 2 and call-convention 8 are in effect for a `.gnt` file in an Intel x86 32-bit environment.
- **ILSMARTTRIM** - trims any trailing spaces from a string item returned by the get property associated with an alphanumeric item processed by ILSMARTLINKAGE.
- **MAINFRAME-FLOATING-POINT** - specifies the format of a program's floating point data items: either IBM hexadecimal format or IEEE format. This directive is supported in managed code only.

The following Compiler directives contain new parameters in this release:

- **CHECKDIV** - a new parameter 'ACOS' now emulates a divide by zero operation on an ACOS mainframe system: the quotient and the remainder are set to the value of the dividend.
- **OOCTRL** - a new parameter, **L**, specifies whether to include directory location comments in `.cls` and `.ins` inheritance files.
- **NUMPROC** - a new parameter 'ACOS' provides partial compatibility with the behavior of NEC ACOS COBOL processing of invalid data in USAGE DISPLAY data items and invalid sign information in USAGE COMP-3 data items.

Data File Tools

[Back to Top](#)

It is now possible to export any filtered results. When filtering a data file, you can use the results to create a new data file - click **Search > Export Results** when a filter is applied to save the filtered records to a new file. You can:

- Save the filtered records to a new file.
- Save the records that match the specified filter (such as customer information or orders).
- Download a subset of the data from a remote file.
- Save a small portion of the data for testing purposes.

Debugging

[Back to Top](#)

This release includes the following enhancements:

The following enhancements have been made to reverse debugging and live recording:

- Reverse debugging and live recording have been enhanced significantly, and are now considered GA features. You can now debug using watchpoints and conditional breakpoints, and reset execution points. Debugging multi-threaded applications is now supported, and so are programs that contain OSVS performs and nested programs.
- A command line utility, `cobeslr`, has been introduced to enable you to configure live recording for particular services or application instances of an enterprise server region.
- You can now use the `CBL_DEBUG_START` and `CBL_DEBUG_STOP` library routines to start and stop a live recording session.

Docker

[Back to Top](#)

This release provides support to enable you to run your COBOL applications in Docker containers, taking advantage of the many benefits offered by the Docker platform such as portability, performance, agility, isolation, and scalability.

Documentation on working with large applications

[Back to Top](#)

The product help now includes a new section, *Working with a Large Code Base*, that includes recommendations and best practices for working with large applications inside the IDE. It includes tips on how to structure your projects, how to optimize the performance of the IDEs, and step-by-step workflow showing how to move an existing legacy application into Visual COBOL.

Enterprise Server

[Back to Top](#)

The following enhancements have been made to Enterprise Server:

- Conversation filtering - the Enterprise Server Communications Process (MFCS) can now restrict access to listeners by client address. You can specify any permitted or forbidden addresses either by IP address, network mask, or domain name, and use wildcards. Filters can be applied to individual listeners, communications processes, or to entire regions. More specific filter rules override any general ones.
- The Enterprise Server Security Facility now starts throttling Verify requests when it receives more than 100 requests per second.

This can be used to limit the effectiveness of denial-of-service and brute force attacks. You can configure the value where throttling occurs. See *Verify Request Throttling* for more information.
- (Technology Preview only) Support for adding, deleting, and modifying XA resources in a live Enterprise Server region.

It is now possible to add, edit, or delete XA resources while an enterprise server instance is running. Any changes made come into effect after any in-flight transactions have completed. The ability to make

these changes in a live environment comes under the control of the existing enterprise server permissions.

- XA-compliant Resources (XARs) - this release provides enhanced CTF tracing that allows more flexible reporting of warnings and errors on the RM switch module level.
- A new Communications Server resource class - enables you to control the access to the Enterprise Server Console Log and Communications Server Log when external security is in effect for an enterprise server region; see *Resource Classes for Communications Server* for more information.
- Improved catalog availability - there is now an improved resilience to temporary communication issues with the catalog and error reporting enabling a region to stay active if a region has multiple catalogs defined and one of the catalogs is not available.
- Enhanced SSL/TLS certificate support - for communications with TLS (formerly SSL), additional certificate and key file formats are supported. Servers may now be configured with both an RSA and an ECC key and certificate.
- Enhanced SSL/TLS cipher configuration - for communications with TLS (formerly SSL), the permitted cipher suites and their preferred order can now be configured. The minimum size of Diffie-Hellman groups for DH key exchange can also be configured. The defaults have been made more secure.

File Handler

[Back to Top](#)

This release provides the following enhancements:

- The DFSORT and SYNCSORT emulations now support the NULLOFL parameter of the OUTFIL statement.
- The **ASCIISOSI** configuration option is now available. It adds the required SOSI characters to the relevant EBCDIC DBCS character strings in order for them to be displayed or written out correctly.

Library routines

[Back to Top](#)

The following library routines contain new functionality:

- **CBL_GET_OS_INFO** - this library routine can now detect if the program is running within a Docker container: `cblte-osi-rts-capabilities` parameter, bit 7.
- **CBL_DEBUG_START** and **CBL_DEBUG_STOP** - these library routines have been enhanced to start and stop a live recording session.

Managed COBOL

[Back to Top](#)

Deploying JVM COBOL to an Application Server:

- Running JVM COBOL under WebSphere 9.0 and WebLogic 12.2.1 is now supported.

Micro Focus Unit Test Framework

[Back to Top](#)

This release provides support for the following functionality:

- Generation of unit test stubs for selected entry points within your program.

OpenESQL

[Back to Top](#)

This release provides the following new features:

- Support for SQL Server 2017.
- The SQL(TRANSACTION) compiler directive has been enhanced to clearly define transaction boundaries.
- A new SQL(NOWHERECURRENT) compiler directive that allows you to define updateable cursors that do not do positioned updates or deletes with PostgreSQL or MySQL.
- Larger communication area (PID) that accommodates longer plan and program names.
- SQL(OPTIMIZECURSORS) has been enhanced for consistent and better cursor performance across all OpenESQL backends.

Platform support

[Back to Top](#)

Note the following changes in platform support for this release:

- SUSE platforms - this release only supports SUSE Linux 12 SP2, 64-bit.

On this platform, you can only build COBOL programs to 64-bit executables. This applies when using Eclipse on SUSE or with remote projects when Visual COBOL Development Hub is installed on SUSE.

What was New in Visual COBOL 3.0

Visual COBOL 3.0 provided enhancements in the following areas:

- [Integration with Eclipse](#)
- [Application Server JCA support for Enterprise Server](#)
- [Application Workflow Manager](#)
- [Building applications](#)
- [Character encoding](#)
- [COBOL language enhancements](#)
- [Code analysis](#)
- [Code coverage](#)
- [Compiler control](#)
- [Data File Tools](#)
- [Database access - DB2](#)
- [Database access - MySQL](#)
- [Database access - OpenESQL](#)
- [Debugging applications](#)
- [Deployment on multiple platforms](#)
- [Documentation](#)
- [Enterprise Server](#)
- [iFileshare](#)
- [Micro Focus Unit Testing Framework](#)
- [XML processing](#)

Integration with Eclipse

Visual COBOL 3.0 provided enhancements in the following areas:

- Support for Eclipse 4.6 - Visual COBOL now ships with the 64-bit Eclipse 4.6. By default, Eclipse is configured to compile applications for 32-bit and the default target platform for new projects is 32-bit.



Note: Applications that were created using Visual COBOL 2.3 Update 2 and earlier must be recompiled in Visual COBOL 3.0 to ensure they will compile and execute properly.

The 32-bit Eclipse is still supported on 64-bit Windows platforms and you need to manually install the 32-bit IDE and a 32-bit Java

You can install the Visual COBOL plugin in other instances of Eclipse (32-bit or 64-bit). Support for Eclipse 4.2 and 4.3 has been discontinued.

- Rename refactoring - enables you to rename COBOL elements such as variables and identifiers, section and paragraph names, classes, and methods across a program or a workspace. Renaming helps improve the readability of an item or make its purpose clearer. Two preferences for renaming are supported in Eclipse - directly in the editor or using a **Rename** dialog box that offers a preview.
- Copybook paths - it is now possible to specify copybook paths to folders that are outside the project folder.
- Standalone files:
 - It is now possible to specify the locations where the IDE will search for the program symbol files (.idy) for standalone files: **Window > Preferences > Micro Focus > COBOL > Standalone Files > Program Symbols (.IDY)**.
 - It is now possible to use the Eclipse debug configurations to debug standalone files, including core dump files.

In previous versions of Visual COBOL, standalone files were known as "single files". References to "single files" in the IDE and the product help have been changed to "standalone files".

- Error reporting - errors reported in the **Console** view now include a link for opening the source file at the line where the error originates.
- Editor improvements - an option for removing trailing whitespaces has been added to the IDE preferences for the COBOL editor.

Application Server JCA support for Enterprise Server



Restriction: This feature applies only when the Enterprise Server feature is enabled.

In Visual COBOL 3.0, EJBGEN has been updated to generate an EAR file as a part of the COBOL deployment process, which enables you to deploy EJBs to Java Application Server.

Application Workflow Manager

Visual COBOL for Eclipse now provides the Application Workflow Manager (AWM) feature. AWM enables you to customize and extend the Eclipse UI and workbench functionality to create application workflows that meet your requirements.

The features comes with:

- A standard project model for local COBOL development that reflects the features available in COBOL Explorer. You can modify this model by adding or removing project features.
- **Application Explorer**, **Filter Definitions**, **Properties**, and **Table Results** views in the COBOL perspective.

Visual COBOL 3.0 includes improvements in the following areas:

- Application Workflow Manager model editor:
 - The definition of linked element types and properties has been improved. Some required model objects and relationships are now generated automatically.
 - The editor now detects unused references in a model.
- Application Workflow Manager modeling:
 - Basic resource processing has been extended to support mass processing where appropriate.
 - The AWM function package now includes the following new tools:
 - "Return Property Value" - enables you to map the values of input parameters to output parameters.
 - "Validate Value" - enables you to validate a filename against a specified naming convention.

- You can now specify enabling conditions in a more granular way. An additional condition type has been introduced to hide actions or properties. For example, hiding an action from the context menu of an element.
- The Eclipse linked element type now enables the label and label decorator to be automatically set to mirror their representation in other Eclipse views if they are not explicitly modeled.
- Table columns can be modeled as a combination of two or more properties - for example, to combine a date and time property value.
- Additional standard icons.
- Mainframe Access (MFA) integration and the MVS function package:
 - MVS resources are now available as linked element types.
 - The following tools have been added to the MVS function package:
 - Delete MVS data set or member
 - Rename MVS data set or member
 - Copy MVS file to clipboard, Paste MVS file from clipboard
 - Get data set or member attributes
 - Improvements have been made to copy and paste support in the MVS Explorer.
 - File mapping between z/OS and local files now supports additional local code pages.
 - The editor context menu contains a "Submit to Mainframe" action when a file is opened from the MVS Explorer with an extension which is mapped to JCL in the File Mappings view.
 - Some of the new MVS function package tools replace existing ISPF function package tools resulting in improved performance of the tools.
 - The sample model "MVS Projects Sample Application" has been improved by including some of the new AWM MVS function package features.
- AWM is now available as a standalone feature which you can install into a separate instance of the Eclipse IDE installed on your machine.

Building applications

Visual COBOL 3.0 provided the following improvements:

- Support for faster, parallel building on multi-CPU machines - support has been added for multi-processor compilation of the sources in native COBOL projects on multi-CPU machines.

You can specify the maximum number of concurrent compilations from the IDE preferences - **Window > Preferences > Micro Focus > Builder**.

Character Encoding

A new utility, cobutf8, is available. cobutf8 enables you to seamlessly run applications that require non-UTF-8 character encodings in a UNIX environment that is using a UTF-8 locale.

COBOL language enhancements

Visual COBOL 3.0 includes the following enhancements to the COBOL syntax:

- The DISPLAY-OF and NATIONAL-OF intrinsic functions are now able to process conversions using any IBM CCSID value.

The following enhancements are available in managed COBOL:

- To avoid an exception being thrown if an explicit conversion fails, use the AS IF syntax, which results in the target object being set to null and no exception thrown.

Code analysis

Visual COBOL 3.0 provided the following improvements:

- A new group of predefined rule sets for 64-bit readiness is now included in Visual COBOL.
- Support for importing code analysis reports produced with one of Micro Focus's advanced tools for code analysis, Enterprise Analyzer or COBOL Analyzer.

Code coverage

The following improvements are available within the IDE:

- Information about unexecuted programs - the code coverage reports in the **Code Coverage** window now show the unexecuted programs.
- Code coverage support for standalone COBOL files - you can import existing code coverage reports in the **Code Coverage** window and use it to supply code coverage information for standalone files.
- Remote code coverage files - in the **Code Coverage** window in Eclipse, it is now possible to import code coverage report files that are stored in remote locations.
- Test coverage files - it is now possible to use a test coverage file to debug applications.

If you are using Test Coverage from the command line, you can now use the following features:

- A new Compiler directive, COLLECTION - the directive enables test coverage to gather information about unexecuted programs. In the IDE, this directive is automatically set on a project when you enable code coverage for it.
- A new command line utility, tcutil - the utility enables you to convert the test coverage binary results file into XML format.
- It is now possible to integrate test coverage in a Continuous Integration (CI) system. You can use tcutil and an XSLT processor to transform test coverage data into a format suitable for including in a CI.

Compiler control

The following Compiler directive are new in this release:

- COLLECTION - provides a mechanism for code coverage to identify unexecuted programs.

The following Compiler directives have been updated:

- ALIGN - this directive has new parameters (FIXED and OPT) that can be used in conjunction with the integer taken, which can aid performance. The default is ALIGN"8 OPT"; see the Comments section of the *ALIGN* Compiler directive topic for details of its affect on memory boundaries.
- ARITH - this directive emulates the IBM mainframe option of the same name. Defines the maximum number of digits for numeric data items.
- FASTINIT - this directive is now on by default when setting the MF dialect; it remains not set by default for other dialects.
- SSRANGE - this directive now has an additional option (3), which permits zero-length reference modified items at run time when bounds checking.
- XMLPARSE - includes a change in the way entities are processed when XMLPARSE"COMPAT" is set

Data File Tools

The Data File Tools editor previously provided (from Visual COBOL 2.3) as a Technology Preview item was supported at GA level from Visual COBOL 3.0.

Visual COBOL 3.0 provided the following enhancements to Data File Tools:

- Opening files in shared mode - it is now possible to switch between read-only shared and edit modes. While a file is open in shared mode, others users can only open it in shared mode to ensure data consistency between users.
- Enterprise Server-level of security when accessing files - there is an improved level of security when exchanging data between Data File Tools and the targeted enterprise server instance. Users must now provide a user ID, group and a password when they try to access and view datasets in enterprise server instances. These are used for authentication and authorization checks to provide the same access level as Enterprise Server.

- Opening datasets using SSL - communication to a region is now possible using SSL. To enable the SSL communication, you need to provide a Java trust store which contains either a CA root certificate or a self-signed certificate of the region that it is communicating to. Java and the targeted region SSL configurations need to meet each other's standards in order for the communication to succeed.

This feature enables you to secure the information exchange between Data File Tools and the targeted enterprise server.

- Auditing of access and updates on datasets - Audit Manager now audits the access and updates on datasets via Data File Tools.
- Support for existing .pro files - enables you to use your existing editor profiles.
- Support for existing .str files - enables you to use your existing COBOL structure files.
- Automatic timeout - if no internal operations or external actions (such as a mouse click) have been detected for 30 minutes, Data File Tools now displays a countdown message. If the user does not take any decision within the specified period, Data File Tools closes all opened files.

Database access - DB2

Visual COBOL 3.0 provided a new DB2"QUALIFY-CALL" Compiler directive that enables stored procedure invocations to include a schema name.

Database access - MySQL

Visual COBOL 3.0 provided support for MySQL with ODBC.

Database access - OpenESQL

Visual COBOL 3.0 provided the following new features:

- Statement prefixes for the SQL"CHECK" Compiler directive that enable the creation of temporary tables and other SQL objects at compile time, ensuring full SQL syntax checking during compilation.
- SQL"OPTIMIZECURSORS" Compiler directive that enhances processing for traditional embedded SQL cursors that use WITH HOLD and FOR UPDATE clauses.
- SQL"CLOSE_ON_COMMIT" Compiler directive to leave cursors open for further result set processing after a commit.
- SQL"GEN-SQLCA" Compiler directive that generates an SQLCA similar to the z/OS DB2 directive STDSQL"YES".

Debugging applications

Visual COBOL 3.0 provided the following new features:

- Debugging core dump files without a project.
- Reverse debugging (Technology Preview) - it is now possible to step backwards through an application to view a recording of the previous steps executed in the current debug session. Input is not accepted while viewing recorded execution.



Restriction: This feature is supported on Red Hat Linux x86 platforms only. Additional restrictions apply. For details, see the *Reverse Debugging and Live Recording* topic in the *Known Issues and Restrictions* section of this documentation.

- COBOL Live Recording debug configuration (Technology Preview) - it is now possible to record an application execution and view its execution path in the debugger. This method of debugging enables stepping both forwards and backwards, but does not accept input.



Restriction: This feature is supported on Red Hat Linux x86 platforms only. Additional restrictions apply. For details, see the *Reverse Debugging and Live Recording* topic in the *Known Issues and Restrictions* section of this documentation.

Deployment on multiple platforms

Visual COBOL 3.0 provided support for deploying JVM COBOL applications on multiple platforms. You can compile an application on one platform (such as Windows) and then deploy its class files to a different platform (such as Linux or UNIX).

Some features of the COBOL language, however, are platform-specific and their behavior on different platforms might vary. See *Multi-Platform Deployment of JVM COBOL Applications* in your product Help for details.

Documentation

The following new sections have been added to the product help:

- *Where do I start?* - located on the launch page of the product help, this section provides the information you need in order to get started depending on which aspects of the product you need to get to grips with first.
- *Multi-Platform Deployment of JVM COBOL Applications* - includes information about how to ensure the portability of your JVM COBOL applications between Windows and UNIX or Linux platforms.

Enterprise Server

Improvements are available in the following areas:

Integration with Eclipse

- Exporting an enterprise server definition from the IDE in XML format.
- Importing an enterprise server into the IDE using its definition file.

Long user IDs and passwords:

- Enterprise Server now supports user IDs and passwords of up to 100 characters. It is possible to map IDs from long to short (or vice versa) to enable compatibility with programs that do not support long names.

SHA-256 support in DemoCA:

- By default, the Demonstration Certificate Authority (CA) now signs certificates with SHA-256. This ensures that the demonstration or evaluation certificates will be accepted by modern browsers and other software that has enhanced security requirements.

Syslog auditing:

- Enterprise Server now supports auditing using syslog events, which can be consumed by a wide range of Security Information and Event Management (SIEM) products. This replaces the Audit Manager auditing solution. Syslog auditing provides a much more efficient auditing mechanism, with significantly less impact on overall speed.

iFileshare

iFileshare, previously considered a Technology Preview feature, was supported at GA level from Visual COBOL 3.0. It also contained the following enhancements:

- An improved failover and recovery process. iFileshare now supports full recovery of nodes in the group. For high availability (HA-VSAM) groups, servers can now rejoin the group without the entire group having to be restarted. In addition:
 - A primary failover now results in a takeover from the most suitable node.
 - If configured, external clients will automatically reconnect to the new primary and will issue a notification if the transaction has been lost.
 - A failed node, when restarted, will rejoin the group, recover its files and request a log update from the current primary. Once this task has completed it will be considered an active hot-standby and will continue to process replication requests as normal.

- Users will experience a higher level of uptime/availability with their Fileshare configuration and will be able to recover from errors more easily.
- A new exit procedure, `ifsexitproc.cbl`, can be configured to automate some aspects of iFileshare behavior.
- The iFileshare Control page in ESMAC contains details of the current iFileshare high availability group.
- The following new iFileshare-specific environment variables are available:
 - FSWRKDIR - enables you to specify the Fileshare working directory, overriding the default, which is the system directory of the region.
 - FSCHKLFH - determines if a check is performed when a high availability group is started, to test the consistency of the data files within the group.
- The database reference file (`dbase.ref`) now supports wildcard matching for filenames, allowing you to perform operations on multiple files at once; for example: `fs /d dbase.ref /f data*` adds the entire contents of the data directory to the database reference file.

The Micro Focus Unit Testing Framework

The Micro Focus Unit Testing Framework is now available from within the IDE. It includes much of the architecture you would expect of an xUnit framework to create, compile, run and debug unit tests, including the following features:

- A unit test project template.
- A test creation wizard that enables you to generate tests from your source code.
- Code templates for each element of a test case.
- Support for running tests with Code Coverage enabled.
- The Micro Focus Unit Testing view, where you can manage your test runs and view test output.

There has been a number of enhancements to the command line version of the Micro Focus Unit Testing Framework. Support has been added for:

- Running test fixture files using Apache Ant.
- Applying traits to your test cases, then performing a test run based on those traits.
- Applying a high, medium, or low priority to test cases, which affects the order in which they are run.
- Adding coded command line options directly into your test code.
- Using a test run-specific configuration file, in which you can set environment variables.

XML processing

XML PARSE now works in a purely managed COBOL environment. It is now supported in JVM COBOL and, in both .NET and JVM COBOL, it has a fully managed implementation. XML PARSE working without calling out to native code ensures it can be used in restricted rights environments.

What was New in Visual COBOL 2.3 Update 2

Visual COBOL 2.3 Update 2 provided enhancements in the following areas:

- [Integration with the Eclipse IDE](#)
- [COBOL language enhancements](#)
- [Classic Data File Tools](#)
- [Compiler directives](#)
- [Editor writing assistance](#)
- [File handling](#)
- [Interface Mapping Toolkit](#)
- [Library routines](#)
- [Tutorials](#)

Integration with the Eclipse IDE

Visual COBOL 2.3 update 2 provided enhancements in the following areas:

- Using the search facility, **Search > Micro Focus**, you can now limit the search to the copybooks used by the current program rather than all copybooks in the project or the entire workspace.
- The editor provides a new context menu command, **Extract COBOL code to copybook**, that enables you to move a selected segment of the code to a new copybook file in your project. The segment of code moved to a copybook is replaced with a COPY statement in the original program.
- Support for Eclipse 4.5 - after installing Visual COBOL, you can add it as a plugin into a separate instance of Eclipse 4.5. See your product's *Installation* notes.

COBOL language enhancements

Numeric, edited and external floating point items can now specify USAGE NATIONAL when the NATIONAL"2" Compiler directive is in effect. Signed numeric items must be specified with the SIGN IS SEPARATE clause.

Classic Data File Tools

A new command line utility is available which enables you to initiate the following actions: open data files, create or open record layout files, create or open segment layout files, and open IMS databases using a DBD or PSB file. Note that although you can initiate these actions from the command line, you must complete them from within the IDE.

Compiler directives

The following Compiler directives are new in this release:

- COMMAND-LINE-LINKAGE - enables you to call a program and pass the command line to the main program as a parameter to be accessed via the Linkage Section. This offers equivalent functionality to the command_line_linkage tunable, which has now been deprecated.
- EBC-COL-SEQ - controls the behavior of an EBCDIC collating sequence, specified in a NATIVE"EBCDIC" program. EBC-COL-SEQ"1" (the default) maintains use of the long-standing fixed (platform-independent) EBCDIC collating sequence. EBC-COL-SEQ"2" prompts use of the latest CODESET table, which varies according to platform and user-controlled MFCODESET environment variable setting.
- NATIONAL - enables you to specify numeric, edited and external floating point items as USAGE NATIONAL.

Editor writing assistance

Visual COBOL 2.3 update 2 provided the following enhancements:

- Colorization of conditional compilation regions - by default, inactive code is now colored the same as Compiler directive elements. Inactive code is defined as code within conditional blocks that do not evaluate with the applied Compiler settings.
- \$REGION statement - support is provided for the \$REGION Compiler-control statement. You can use \$REGION - \$END-REGION to surround blocks of code that you want to fold or expand in the editor.
- AutoCorrect - you can configure the editor to automatically fix the most frequently misspelled words. You use the IDE preferences to specify a list of words that you sometimes mistype, and the correct spellings for them. Whenever you misspell that word, the editor automatically replaces it with correct version. This feature is enabled by default and can be configured from a new preference page in **Window > Preferences > Micro Focus > COBOL > Editor > AutoCorrect**.

File handling

MFJSORT ICETOOL now supports the USING parameter in the SELECT operator.

Interface Mapping Toolkit

Visual COBOL 2.3 update 2 supports JSON schemas for the generation of REST Web service clients in the Eclipse IDE.

Library routines

The following library routine contains new functionality:

- CBL_GET_PROGRAM_INFO - a new function (function 10) has been added for native COBOL which returns the path and program name, or the program name only of a particular program.

Tutorials

The product help includes the following new tutorial:

- *Tutorial: SQL - Deploying an Enterprise JavaBean Containing JVM COBOL to a JBoss Application Server* - that walks you through the process of deploying an EJB that contains JVM COBOL code.

What was New in Visual COBOL 2.3 Update 1

Visual COBOL 2.3 Update 1 provided enhancements in the following areas:

- [Integration with the Eclipse IDE](#)
- [Application Server JCA support for Enterprise Server](#)
- [Code Analysis](#)
- [Code Coverage](#)
- [Compiler directives](#)
- [Data File Tools](#)
- [Editor writing assistance](#)
- [Enterprise COBOL 5.2](#)
- [File Handling](#)
- [Library routines](#)
- [Managed COBOL Syntax](#)
- [Native COBOL Syntax](#)
- [RM/COBOL compatibility](#)
- [Rosetta Stone for COBOL, .NET and Java Developers](#)
- [UNIX and Linux platform support](#)
- [Windows Azure](#)
- [z/Server](#)

Integration with the Eclipse IDE

Visual COBOL 2.3 Update 1 provides the following enhancements in the integration of Micro Focus COBOL with the Eclipse IDE:

Editor:

- COBOL editor:
 - You can now specify the increment for the COBOL and the standard line numbering from **Window > Preferences > Micro Focus > COBOL > Editor > Line numbering**.
 - You can now toggle single or multiple lines between commented and uncommented states.

Building applications:

- Environment variables - a new page, **Build Environment**, in the project's properties enables you to specify environment variables for your applications. You can also specify environment variables that only apply at run time on the run or debug configuration for the application.

Code analysis

[Back to Top](#)

This release provides support for performing code analysis at the command line using Ant which enables the integration of code analysis in CI frameworks. Features include:

- Support for performing code analysis at the command line using the project's `.cobolBuild` Ant script and specifying a target.
- New Ant targets for code analysis - `analyze` and `build.and.analyze`. These enable you to only run analysis and produce analysis data or to build and produce build artifacts as well as analysis data.
- New Micro Focus Ant task, `analysis`, for the `.cobolBuild` file. A parameter for this task enables you to specify whether the build fails or continues when code analysis results are received.
- New parameters for code analysis for the cobol Ant task - `analysisData`, `analysisDataDir`.
- New Ant type, `ruleList`, for the `.cobolBuild` file - enables you to specify the rules to execute.
- Support for running analysis using a custom `.cobolBuild` file from outside of the project directory or the workspace.

See the [Micro Focus Ant User Manual](#) in the Micro Focus Infocenter for more details on the new task, types and parameters.

Code coverage

[Back to Top](#)

The code coverage reports are now integrated with the IDE and with the editor. Features include:

- A new Code Coverage view (Eclipse) showing the statistics of what percentage of the code has executed.
- Navigation from the Code Coverage view (Eclipse) to the missed and covered blocks in the editor.
- Colorization in the editor of blocks that were executed (covered blocks) or not (missed blocks).

Compiler directives

The following Compiler directives are new in this release:

- `ILMAIN` - you now specify the main entry point for the executable program, which can be specified either as `class-name::method-name`, or just as `method-name`. For example, `ILMAIN"classA::methodB"` or `ILMAIN"methodB"`. The first format can be used to distinguish between multiple methods with the same name in different classes.

This directive is now available for JVM COBOL.

- `OOCTRL` - a new parameter, `+/-A`, as been added. Set this parameter to `-A` to allow ActiveX controls in your COBOL application to use classes and methods in the OLE class library. The default is `+A`, which does not allow it

Data File Tools

This release provided improved security and increased support for more file types. Features include:

- Certain aspects of Enterprise Server security are honored when you attempt to access data sets. If the Enterprise Server region has security enabled, logon details must be authenticated before you can access the data set. If the details are unable to be authenticated, access is denied.
- When using a record layout, certain data is now validated at field level (to ensure the contents is compatible with its picture string) and record level (to ensure the record length matches the layout size).
- Full editing support has been added for variable block sequential files and relative files. Full editing is also available for line sequential files, as long as they do not contain any binary data

Enterprise COBOL 5.2

[Back to Top](#)

With the introduction of Enterprise COBOL 5.2, the following features were supported:

- The VOLATILE keyword is supported within the data entry description; although, this is treated as documentary. It has also become a reserved word when under the ENTCOBOL dialect.
- Format 2 of the SORT statement no longer treats the COLLATING SEQUENCE clause as documentary-only.
- The SUPPRESS clause of the XML GENERATE statement has been enhanced.
- The IBM z/OS JSON parser API, as documented for the IBM z/OS client web enablement toolkit.

File Handling

- A new indexed file format, IDXFORMAT12, has been introduced to improve file maintenance and recovery procedures when using the rebuild utility. This file format is similar in structure and use to IDXFORMAT8. Where the two formats differ is that an IDXFORMAT12 file has an accompanying side file (.idx file) containing the indexed key information.

You can use this type of file with the new `rebuild /q` option. This rebuild process is considerably quicker than other rebuild processes such as a data scrape or `rebuild /p`.

- Faster SORT operations for fixed block records - when using the DFSORT emulation, the performance when sorting fixed block records has greatly improved.

Library routines

The following library routine were new at this release:

- CBL_CODESET_SET_MAPPING - enables you to change the codeset in effect.
- CBL_RUNTIME_ERROR - forces an application to terminate with a run-time error condition.

Managed COBOL syntax

[Back to Top](#)

The following enhancements have been made to the managed COBOL syntax:

- A new command line utility, `mfjarprogmapp`, is available to allow you to create the necessary Java property file when calling COBOL programs that have been compiled as part of a package.
- You can now create generic iterators.
- You can now use the Profiler utility to obtain detailed statistics on the run-time performance of managed COBOL applications.

Native COBOL Syntax

[Back to Top](#)

The following items are new features of the native COBOL syntax:

Class condition tests New and updated class condition tests are available for DBCS, KANJI, and JAPANESE.

RM/COBOL compatibility

[Back to Top](#)

The RM/Panels syntax is now supported in Micro Focus COBOL applications.

Rosetta Stone for COBOL, .NET and Java Developers

[Back to Top](#)

The product Help now includes a quick and easy to use syntax guide for developers who need to learn OO COBOL syntax when modernizing COBOL applications for the Java or .NET platforms. The guide includes side-by-side equivalent syntax for COBOL, C#, VB and Java.

UNIX and Linux platform support

[Back to Top](#)

This release is now supported on SUSE and Red Hat platforms that are running the little-endian PowerLinux architecture.



Note: These are 64-bit platforms only.

There are a few restrictions when running in this environment:

- The `cob` flag `-p`, which enables profiling, is not supported on Red Hat platforms.
- The `cobmode` utility is not supported.
- SQL functionality is restricted to OpenESQL support (ODBC and JDBC) only.
- The RM File Manager (RMFM) is not supported.

z/Server

[Back to Top](#)

The z/Server Configuration Utility is now installed as part of COBOL Server and is not a Technology Preview download. The user interface has been streamlined for the creation of a default working configuration.

What was New in Visual COBOL 2.3

Visual COBOL 2.3 provided enhancements in the following areas:

- [Integration with the Eclipse IDE](#)
- [General IDE enhancements](#)
- [Building JVM COBOL Projects Incrementally](#)
- [COBOL Editor in Eclipse](#)
- [Code analysis](#)
- [Code coverage](#)
- [Compiler directives](#)
- [Data File Structure command line utility](#)
- [Data File Tools \(Technology Preview\)](#)
- [Database access](#)
- [File locking](#)
- [File handling](#)
- [Library routines](#)
- [Managed COBOL syntax](#)
- [Micro Focus Infocenter](#)
- [Micro Focus Unit Testing Framework](#)
- [Personal edition licensing](#)
- [Preprocessors](#)
- [Profiler](#)
- [Remote Connections](#)
- [REST service interfaces](#)
- [RM/COBOL Compatibility](#)
- [Single file support](#)

- [Tunables](#)
- [Updated run-time system](#)

Integration with the Eclipse IDE

This release shipped with Eclipse version 4.4.2.

Support for Eclipse versions 3.7 and 3.8 has been deprecated. This affects any applications that were created using an earlier version of Visual COBOL that have a JVM COBOL part. Such applications must be rebuilt using Visual COBOL 2.3 to avoid receiving errors during compilation or execution.

In addition, when Visual COBOL is installed, you can optionally install the Visual COBOL plugin in other instances of Eclipse installed on your machine (supported versions of Eclipse are 4.2, 4.3 and 4.4 for the 32-bit IDE).

General IDE enhancements

In Eclipse:

- The **Variables** view now shows the file status of an internal file name.
- The **Find All References** and **Go To Definition** commands are now supported for JVM COBOL applications.
- Support is provided for the SOCKS5 proxy server for debugger communication over SSL.

Building JVM COBOL Projects Incrementally

To minimize the number of modules to compile when building JVM COBOL projects, Visual COBOL now enables you to configure your projects so that they are built incrementally - the IDE only rebuilds the files that have changed.

To enable incremental builds for JVM COBOL applications that contain namespaces, check the **Use incremental build (Technical preview feature)** option on the project build configuration tab in the project's properties.

For JVM COBOL applications that do not contain namespaces, you can use the **Use dynamic calls** option available on the build configuration tab in the project's properties. When the project is built with this setting, calls to modules are resolved at run time rather than during compilation. This has the effect of not requiring every module to be compiled when rebuilding the application.

COBOL Editor in Eclipse

Content Assist support for COBOL includes various enhancements and is now also available in JVM COBOL:

- Context sensitive proposal - Content Assist only shows proposal that are relevant for the position of the cursor in the code or for the type of project
- Enhanced proposal lists - lists include any relevant COBOL verbs, clauses and words, copybooks, code templates, data items and section and paragraph names
- Intelligent assistance with completing statements - when you have entered a COBOL verb, Content Assist shows proposals for the relevant clauses and identifiers that you can use to complete the statement.
- Automatic completion for items - Content Assist automatically inserts single suggestions in the code.
- Qualifying non-unique names - Content Assist qualifies data items whose names are not unique.
- Configuration preferences for Content Assist - enable you to configure what suggestions appear in the completion lists, whether suggestions are added in insert or overwrite mode, and the case of the inserted words.
- Code templates - code templates are now included in the Content Assist proposals.

Code analysis

Visual COBOL now offers more advanced code analysis features and enables you to run various analysis queries (rules and groups of rules called rule sets) against your code to ensure adherence to standards such as standards for coding or performance.

You can run analysis rules against programs in a project in the IDE at user request or you can run analysis rules at the end of a project's build.

Code coverage

Visual COBOL now provides support for code coverage of native COBOL applications directly from within the IDE where code coverage uses the Test Coverage functionality. You can produce code coverage reports for applications running in the COBOL run-time and for applications that run in Enterprise Server.

To produce reports, you need to enable code coverage in a project's, a build configuration's, or a file's properties, compile your application and then run your application with code coverage to produce the relevant reports. For applications that require an Enterprise Server instance, you start the enterprise server with code coverage.

Compiler directives

The following Compiler directives are new in this release:

EOF-1A	Treats a 0x1a character in the source file as the end of file.
JVMDECIMAL	Determines the type in which certain items are exposed. This directive affects COBOL data items of type 'decimal' and non-integral numeric items exposed as a result of either ILSMARTLINKAGE usage or the PROPERTY keyword.
NLS-CURRENCY-LENGTH	Specifies the number of bytes to allocate for the currency symbol in a PIC field.
NULL-ESCAPE	Treats a 0x00 character in the source file as an escape character for other non-printable characters in the source code.

The following Compiler directives contain new parameters in this release:

DBSPACE The new parameter 'MIXED' extends the DBSPACE directive to be able to evaluate data items in programs that contain a mix of single-byte and double-byte strings.

Data File Structure command line utility

The Data File Structure Command Line (DFSTRCL) utility is a DOS-based command line utility that enables you to create record layout (.str) files from COBOL debug information (.idy) files. You can use the utility to process a single .idy file or batch process up to 100 .idy files.

Data File Tools (Technology Preview)



Note: This is a technology preview feature only. It is being made available to allow you to test and provide feedback on this new capability; however, this feature is not intended for production use and it is not supported as such. Furthermore, Micro Focus does not guarantee that this feature will be delivered at a GA level and if it is, then the functionality provided might differ considerably from this technology preview.

The Data File Tools (Technology Preview) is a new standalone text editor in which you can create and edit data files. By nature of it being a 'technology preview' product, it does not currently include all the functionality that was available in the previous version of Data File Tools - now referred to as Classic Data File Tools. If you require any of the functionality not provided in this version, you can still use the classic version by accessing it in the usual way.

To run Data File Tools (Technology Preview), type `mfdatatools2` from Visual COBOL's command prompt or a terminal.

To use the new editor directly from the Eclipse IDE, use the **Open with** option on the shortcut menu when selecting a data file or structure file, and select **Data File Tools**. Eclipse remembers the last tool used for a particular file type, and so will use Data File Tools (Technology Preview) until you select a different editor.

Database Access

Visual COBOL version 2.3 provides the following enhancements to database access:

COBSQL Visual COBOL version 2.3 provides:

- Selection and configuration of the Oracle Pro*COBOL preprocessor for compiling COBSQL applications in project properties on the **SQL Preprocessor** tab and in the build configuration settings.
- Support for COBOL directives SOURCEFORMAT=TERMINAL and SOURCEFORMAT=VARIABLE for Pro*COBOL applications.

HCO for DB2 LUW Visual COBOL version 2.3 provides:

- Support for MFHCO mode across all platforms by default via the new HCO "NOHCO" DB2 compiler directive option. See the *HCO DB2 compiler directive option* topic for details.
- A new DB2 compiler directive option, OPTPER "NOOPTPER", that enhances performance for CHARSET EBCDIC processing. See the *OPTPER DB2 compiler directive option* topic for details.
- A new DB2 directive option, BINDDIR, which specifies an alternative directory in which to write the DBRM file created during compilation. You can set BINDDIR from the command line or specify it in your project properties. See the *BINDDIR DB2 compiler directive option* topic, and the *Binding* topic for details.

OpenESQL	Date/Time Processing	This release provides streamlined datetime processing for ODBC and JDBC.
	Performance	This release includes a new SQL compiler directive option, OPTPER "NOOPTPER", that enhances performance for CHARSET EBCDIC processing. See the <i>OPTPER SQL compiler directive option</i> topic for details.
	PL/I	This version provides 64-bit support for PL/I on appropriate platforms. See <i>Additional Software Requirements</i> for details.
	PostgreSQL	In this release, PostgreSQL 9.4 has been tested with OpenESQL and OpenESQL Assistant with the following PostgreSQL software: Server software PostgreSQL EnterpriseDB version 9.4.1-3 Client software <ul style="list-style-type: none"> • psqLODBC driver version 09.03.04.00 • JDBC41 PostgreSQL driver version 9.4-1201

PostgreSQL 9.4 has been tested with OpenESQL on the following UNIX platforms:

- X86-64 running Red Hat Linux, 32- and 64-bit
- X86-64 running SuSE Linux, 32- and 64-bit



Note: Micro Focus provides compatibility for PostgreSQL but does not directly contribute to or support the PostgreSQL open source project. Any issues relating to PostgreSQL

functionality should be addressed through an open source support vendor.

SQL Server Visual COBOL version 2.3 provides support for the SQL Server OUTPUT clause.

XA Switch Modules

In this release, the XA interface has been redesigned to provide:

- Consistent look and feel for SQL Server, DB2, and Oracle user personalization
- Consistent look and feel for both RM dynamic and static registration (SQL Server, DB2, Oracle, generic one-phase commit)
- Additional support for two instances of the same switch module using Web Services applications via the new XAID compiler directive
- Using a specified XA resource only with batch applications executing under Enterprise Server

File handling

This release contains the following new configuration options:

ACUFH Enables or disables the use of the ACU file handler (ACUFH), which is required to handle Vision and RM/COBOL indexed files.

ESACUFH Enables or disables the use of the ACU file handler (ACUFH) for file handling operations running under Enterprise Server. ACUFH must also be enabled for this option to take effect.

File Locking

In versions prior to Visual COBOL 2.3, the semantics of the sharing phrase specified in an OPEN statement or used within a call to CBL_OPEN_FILE were not correctly applied in some cases on UNIX and Linux platforms. From version 2.3 onwards, the sharing phrase is correctly honored when the tunable `strict_file_locking=true` is set, which is the default setting.

Example of potential changes in behavior:

- *Process-A* opens a file with read-only access and a sharing mode that denies other processes write access (SHARING WITH READ ONLY).
- *Process-B* then attempts to open the file with read-only access and a sharing mode that denies other processes read access (SHARING WITH NO OTHER).

With `strict_file_locking=true`, *Process-B* is unable to open the file, because *Process-A* has successfully opened the file allowing only read access.

With `strict_file_locking=false`, *Process-B* successfully opens the file.

If your application encounters unexpected OPEN conditions or fails to open files, it might be as a result of the new file locking behavior. In such circumstances, we recommend that you review the file locking and sharing requirements of your application and refactor your source code to work with the default setting. The original file locking and sharing behavior can be restored by setting `strict_file_locking=false`.

Library routines

The following library routines are new in this release:

CBL_MANAGED_SESSION_GET_USERDATA Retrieves user data saved in the current RunUnit.

CBL_MANAGED_SESSION_SET_USERDATA Sets user data in the current RunUnit.

The following library routines contain new parameters in this release:

CBL_LOCATE_FILE You can now specify a file name that is a null-terminating string, which has resulted in three new values available for the `user-mode` parameter.



Managed COBOL syntax

The following enhancements have been made to the managed COBOL syntax:

- The `TYPE OF type-name[ANY...]` syntax enables you to obtain the `System.Type` (.NET) or `java.lang.Class` (JVM) object for a generic class, interface, or delegate.
- The `self::` or `super::` syntax is no longer required to access inherited data within a subclass.
- The `ATTRIBUTE-ID` syntax enables you to define new attribute types, which can be used in various contexts.

Micro Focus Infocenter

The Micro Focus Infocenter Web site (<http://documentation.microfocus.com>) has been upgraded and now includes the following improvements:

- Scope being persisted when you select a product documentation in the Product Documentation section on the Micro Focus SupportLine Web site and choose to view the documentation in the Micro Focus Infocenter.
- Updated **Scope** settings - provides the ability to nest four levels deep when setting a scope.
- Scope being persisted between browser sessions once it has been set.
- Creating automatic scopes using the **Search Topics** icon, .
- A link to change the scope from the search results when there are too many results.
- Improved Boolean search expressions.
- Details included with the search results.
- Help on how to use the Infocenter and how to construct search expressions - available using the Infocenter Help button, .

Micro Focus Unit Testing Framework



Note: This is a technology preview feature only. It is being made available to allow you to test and provide feedback on this new capability, but it is not intended for production use and is not supported as such. Furthermore, Micro Focus does not guarantee that this feature will be delivered at a GA level and if it is, then the functionality provided might differ considerably from this technology preview. During the preview, you are encouraged to share your feedback and experiences via the Micro Focus community forum - <http://community.microfocus.com/microfocus/>.

The Micro Focus Unit Testing Framework is an xUnit style testing framework, available from the command line, for procedural COBOL applications.

It includes much of the architecture you would expect in an xUnit framework. The test runner is a 32- or 64-bit executable that you run from a Visual COBOL shell prompt. A test fixture or suite is a COBOL program compiled to `.so` that can include the setup, the test case code, and the teardown associated with the test case.

Test results are available in a number of formats. By default, results are displayed to screen and to a `.txt` file, but you can use additional parameters on the command line to produce reports in JUnit format.

Personal edition licensing

This release includes a Personal Edition licensing option for Visual COBOL for Eclipse.

Preprocessors

Support has been added in the IDE for enabling and using multiple preprocessors with your projects.

A new page, **Additional Preprocessors**, has been added to the project's and the files' properties of native COBOL applications to enable you to choose one or more preprocessors to use when building your application and to specify their order of execution.

New reporting capability is now available for user preprocessors: resp-main code 18 indicates that a buffer contains a data name to be marked as modified by the immediately preceding preprocessed line. The data name may be qualified and resp-more contains the column information for the reference.

Profiler

Visual COBOL now provides support for Profiler for native COBOL applications directly from within the IDE. To produce reports, you need to:

1. Enable Profiler in the COBOL property page for a project, a build configuration, or a file.
2. Compile your application to apply the changes.
3. Create a run configuration that has Profiler enabled.
4. Run your application with Profiler to produce the relevant reports.

Remote Connections

This release provided the following enhancements:

- An improved diagnostic tooling to help determine connection problems - enhancements are available for both the client and the server installations.
- It is now possible to specify the Remote System Explorer (RSE) type of connections to create remote mainframe COBOL and PL/I projects. This is to cater for scenarios when it is not possible to use SAMBA or NFS connections within your environment.



Note: The following features and utilities are not supported when remote projects use the RSE connection type:

- The file layout editor and the file editor in the Classic Data File Tools and Data File Tools (Technology Preview) utilities.
- Changing the type of a remote connection - it is now possible to change the type of the remote connection from RSE to NFS and vice versa using the **Remote Settings** for remote projects.

REST service interfaces

RESTful service interfaces utilizing JSON as the media type in request and response messages are now supported using the Interface Mapping Toolkit. This enables you to extend COBOL applications using modern transport payloads and protocols.

RM/COBOL Compatibility

This release includes improved support for RM dialect applications. Please consult with Micro Focus before considering a transition from RM/COBOL to Visual COBOL.

Single file support

The recommended way to work with files within Visual COBOL is to include them in a project. For situations where you might want to quickly open edit a single file, Visual COBOL now provides support for native COBOL files in the IDE when the file is not opened as part of a project. There is limited support for the IDE editing, compiling and debugging features as full support requires a project file.

Tunables

Visual COBOL version 2.3 contains the following new tunables:

putenv_interface Provides backward compatibility for UNIX systems in which the operating system's `putenv()` function is required when setting environment variables.

strict_file_locking Enables a new, more reliable method of file locking for UNIX systems. See *File Locking* for more information.

Visual COBOL version 2.3 contains the following updates to tunables:

default_cancel_mode A new parameter, and default, has been introduced for this tunable; see *default_cancel_mode* for more information.

subsystem_cancel_mode A new parameter has been introduced for this tunable; see *subsystem_cancel_mode* for more information.

Updated run-time system

COBOL Server has been updated to provide an execution environment capable of running applications that were each built using different development products. A consequence of this is that if your application has a main COBOL executable (.exe) that was built with a previous version of Visual COBOL, you should ensure that the executable is rebuilt and packaged with the new run-time system. You can rebuild from the IDE or the command line.

Other COBOL subprograms built with previous versions of Visual COBOL are not required to be rebuilt.

What was New in Visual COBOL 2.2 Update 2

Visual COBOL 2.2 Update 2 provided enhancements in the following areas:

- [Eclipse](#)
- [Character Set Enhancements](#)
- [Code Analysis](#)
- [Database Access](#)
- [Micro Focus COBOL enhancements](#)
- [External Security Facility \(ESF\)](#)
- [Enterprise Server MQ-IMS Bridge](#)
- [Tunables](#)

Eclipse

Visual COBOL 2.2 Update 2 provides the following new functionality and improvements:

- Support for Eclipse 4.2 and 4.3 - Visual COBOL 2.2 Update 2 ships with Eclipse 3.8 but also supports Eclipse 4.2 and 4.3 (the 32-bit IDE only). To use Visual COBOL with a newer version of Eclipse, you need to install Visual COBOL first and then use the `installeclipseplugins` script in `/opt/microfocus/VisualCOBOL/installer`. See *Installing into other instances of Eclipse* for more details.
- Remote JVM COBOL projects - this release provides enhanced support for remote JVM COBOL projects.
- Remote project connections - the diagnosis tool for remote connection issues has been improved. There is now a client-side diagnosis tool and a server-side diagnosis tool for diagnosing connection problems to remote projects and connections to your Visual COBOL Development Hub. You should run both tools for a complete diagnosis.

Character Set Enhancements

The following character sets, available using the `MFCODESET` environment variable, have been enhanced or added in this release:

- Thai Extended (0066) - new
- Korean (0082)
- Simplified Chinese (0086)
- Traditional Chinese (0886)

There are also a number of double-byte character sets that are now capable of mixed single-byte and double-byte character conversion; see the definition of MFCODESET in *Environment Variables in Alphabetical Order* for more information.

Code Analysis

Visual COBOL version 2.2 Update 2 provides Dead Code analysis for COBOL programs that enables you to find unreferenced items or any piece of code that can't be reached .

Database Access

The following new features are available in database access support:

- COBSQL** In Eclipse, the new KEEPCOMP directive resolves COMP/COMP-5 issues with Oracle applications on little-endian platforms.
- HCO for DB2 LUW** Visual COBOL version 2.2 Update 2 introduces GEN-HV-FROM-GROUP - a new DB2 ECM compiler directive option, that generates host variables for all elementary data items when a multiple-level group variable is used in a FETCH or singleton SELECT DB2 statement.
- OpenESQL** This version provides the following new OpenESQL features:
- Support for SQL Server 2014.
 - New SQL Compiler directive options:
 - DETECTDATE=SERVER - resolves host variables alignment with column data types in an SQL table.
 - GEN-HV-FROM-GROUP - generates host variables for all elementary data items when a multiple-level group variable is used in a FETCH or singleton SELECT SQL statement.
 - Sample applications - the following native COBOL SQL sample applications are new with this version:
 - Get Diagnostics - demonstrates how to use GET DIAGNOSTICS EXEC SQL calls to get diagnostic information from various DBMSs.
 - LOB Data Types - Demonstrates how to INSERT and SELECT LOB data in a native application using various DBMSs.

XA switch modules



Restriction: This feature applies only when the Enterprise Server feature is enabled.

The following XA switch module updates are available in this version:

- Oracle switch module:
 - Supports User Impersonation when statically registered.
 - Enables you to specify which XA resource definitions use User Impersonation.
 - Now compiled with one source file, rather than two.

Micro Focus COBOL enhancements

The following enhancements have been made to Micro Focus COBOL:

- The following phrases have been added to the XML GENERATE statement:

- NAME
- TYPE
- SUPPRESS
- The following intrinsic functions have been added:
 - ULENGTH
 - UPOS
 - USUBSTR
 - USUPPLEMENTARY
 - UVALID
 - UWIDTH

External Security Facility (ESF)

The Enterprise Server External Security Facility (ESF) now supports caching the results of some security queries. This can improve the performance of enterprise server instances and of the MFDS when they are configured to use external security.

To enable caching, you need to set non-zero values for the **Cache limit** (maximum size of the cache) and **Cache TTL** (Time To Live, or how long before a cached result expires) settings on the **MFDS Security** tab, the **Default ES Security** tab, or on the **Security** tab for an individual enterprise server. (Currently, the cache settings for Security Managers have no effect; you need to set cache parameters on one of the three Security pages mentioned earlier.)

For more information, see <http://supportline.microfocus.com/examplesandutilities/doxygen/caching.html>.

Enterprise Server MQ-IMS Bridge

At Visual COBOL 2.2 Update 2 the Enterprise Server MQ-IMS Bridge was supported at GA level. It had previously (from Visual COBOL 2.2 Update 1) been available as a Technology Preview item only.

Tunables

Visual COBOL 2.2 Update 2 includes the following new tunable:

- `reduce_java_signals` - specifies the options that are passed to a JVM when mixing Java and COBOL.

What was New in Visual COBOL 2.2 Update 1

Visual COBOL 2.2 Update 1 provided enhancements in the following areas:

- [Micro Focus Heartbleed Update](#)
- [ACUCOBOL-GT Compatibility](#)
- [Assembler Support](#)
- [COBOL Source Information](#)
- [Compare and Synchronization Monitor](#)
- [Compiler Directives](#)
- [Database Access](#)
- [Eclipse IDE](#)
- [Enterprise Server Integration in the IDE](#)
- [Enterprise Server MQ-IMS Bridge \(Technology Preview\)](#)
- [Environment Variables](#)
- [Fileshare Recovery](#)
- [Line Numbering for COBOL Programs](#)
- [IMTK](#)
- [Working with the Mainframe](#)

- [Managed COBOL](#)
- [Managed COBOL Syntax](#)
- [Project and Item Templates](#)
- [Remote Connection](#)
- [Rumba Integration with Eclipse](#)
- [Terminfo Files](#)

Micro Focus Heartbleed Update

The OpenSSL library used in this product was updated to version 1.0.1g to fix the "Heartbleed" vulnerability with TLS heartbeat requests.

ACUCOBOL-GT Compatibility

The following ACUCOBOL-GT support has been added in this release:

- **-Di compiler option** The -Di compiler option, which initializes Working-Storage data items based in their type, is now supported.

Assembler Support

Visual COBOL 2.2 Update 1 provided Assembler support from within the IDE. Features include:

- Support for creating Assembler programs (.mlc extension), macro files (.mac or .cpy extensions) or Assembler linker files (.lin) from within the IDE.
- Support for building Assembler programs and for linking Assembler linker files from within the IDE. You can choose to exclude certain files from the build.
- Assembler editors for Assembler and macro files - includes content assist for reserved words and colorization.
- Support for using Ant for building .asm, .mlc and .cap files and for linking .lin files.
- IDE support for configuring the Assembler compiler and linker options on project, build configuration and file levels.
- Configuring the macro paths.

COBOL Source Information

The **Quick Browse** option is now available as a context menu command in the editor.

Compare and Synchronization Monitor



Note: This feature is only available in a Windows environment.

With the release of Visual COBOL 2.2 Update 1, the Compare and Synchronization Monitor has been updated to version 2.

Version 2 is greatly improved in terms of performance, especially during initial checkout of partitioned data sets or when synchronizing a large number of members. Also, the user interface has been improved, and some of the functions available in the old version have now changed or become obsolete.

Compiler Directives


The following Compiler directives have been added in this release:

- **ILPARAMS** Determines the way in which you call a method that contains an array as its last receiving parameter.

- INIT-BY-TYPE** Initializes Working-Storage Section data items to a default value, according to their type.
- Alphabetic, alphanumeric, alphanumeric edited, and numeric edited items are initialized to spaces.
 - Numeric items are initialized to zero.
 - Pointer items are initialized to null.
 - Index items are initialized to the value 1.

Database Access

The following new features have been added as part of database access support:

- DB2 ECM**
- Support added for DB2 LUW version 10.5.
 - Enhanced RETURN-CODE processing.
- OpenESQL**
- Enhanced internationalization support for UNICODE, DBCS and MBCS.
 - Enhanced GET DIAGNOSTICS statement support.
 - Enhanced LOB support for CLOB, BLOB and DBCLOB data types.
 - Enhanced IDE support for OPTION directives.
 - Now provides support for the creation of save points and rolling back to save points.
- XA Switch Modules**
-  **Restriction:** This feature applies only when the Enterprise Server feature is enabled.
- New two-phase commit module for SQL Server based on Microsoft's XA switch. This provides support for xa_recover.
 - Support for DB2 LUW version 10.5.
 - Support for Oracle version 12.1.

Eclipse IDE

Assigning memory to Linkage Section items If the debugger steps on a line with an unassigned linkage item (for example, if you are debugging only a part of your application and no memory has been allocated to that linkage item), debugging terminates. To assign linkage to that data item and continue debugging, you need to select the data item, right-click it and click **Inspect COBOL**. When prompted, confirm and assign a value to the data item. Alternatively, to assign linkage, you can right-click the data item in the **Variables** view and click **Change Value**.

Indicating that a file is a copybook or a COBOL program In COBOL Explorer, you can now use two new file context menu commands to indicate that a COBOL program is a copybook (**Transform Program to Copybook**) and that a copybook is a COBOL program (**Transform Copybook to Program**). You may need to use these in situations when you imported existing COBOL source code in the Eclipse IDE and some of the files were incorrectly identified as either a copybook or a COBOL program.

Enterprise Server Integration in the IDE

You can now use the context menu for the servers in Server Explorer to enable the display of the Enterprise Server log information in the Console view.

Enterprise Server MQ-IMS Bridge (Technology Preview)



Note: At Visual COBOL 2.2 Update 1 this was provided as a technology preview feature only. It was made available to allow you to test and provide feedback on this new capability; however, this feature was not intended for production use and was not supported as such.

Visual COBOL version 2.2 Update 1 provided support that enables WebSphere MQ applications to communicate with IMS applications in an Enterprise Server region.

Environment Variables

The following environment variable has been added in this release:

strictvsam strictvsam enables strict mainframe emulation when processing VSAM files.

When set to ON and running under mainframe emulation, file status 37 is returned for an existing VSAM file when opened for OUTPUT if the file has data or previously had data written to it, or if the file is of a different format to the file on disk. When set to OFF, file status 0 is returned and a new file is created when an existing VSAM file is opened for OUTPUT. This variable is set to OFF by default.

Fileshare Recovery

Recovery of Fileshare data files has been enhanced.

Rollback recovery is a faster process that aims to fix the files from their failed state.

This process cannot be used in all scenarios, but a new user exit has also been introduced that allows you to programmatically control which files you wish to recover with this process.

Hot backups are also a new introduction, which allow you to perform a backup without having to shut down Fileshare.

Line Numbering for COBOL Programs

Visual COBOL version 2.2 Update 1 provided options for auto-inserting or removing line numbers in source files open the editor. Features include:

- COBOL numbering - line numbers are inserted in the sequence area of the code (columns 1 - 6), starting by default at 000100 at the first line, incrementing by 100 by default.
Micro Focus recommends that you use COBOL numbering only if your files are in fixed or variable source format.
- Standard numbering - line numbers are inserted immediately to the right of area B, in columns 73 - 80, starting by default at 00000100 at the first line, incrementing by 100 by default.
Micro Focus recommends that you use Standard numbering only if your files are in fixed format.
- The **Renumber** and **Unnumber** commands available from the context menu in the editor.

IMTK

You can now use the Interface Mapping Toolkit to create Web Services and Java Interfaces for remote COBOL projects.

Working with the Mainframe

Visual COBOL version 2.2 Update 1 includes the "Launch ISPF" functionality as a technical preview. This feature is supported only under z/OS 1.13.

Managed COBOL

Visual COBOL now provides support for Java managed beans (MBean) in JVM COBOL code that enable you to manage and monitor RunUnits, and to identify certain issues such as leaks and long-running RunUnits.

- You can enable an MBean only for a particular RunUnit level or for all RunUnits you create.
- You can view and use MBeans from programs such as Oracle's Java Mission Control or JConsole.
- MBeans include the `LogicalRunUnitCount` and `LiveRunUnitCount` attributes that enable a visual indication of how many RunUnits are live. If the values of these two attributes are different, this might indicate some issues.

Visual COBOL 2.2 Update 1 also includes the following tutorials for JVM COBOL:

Deploying JVM COBOL to an Application Server	Using some ready-made sample projects, this tutorial guides you through implementing your JVM COBOL code into an Enterprise JavaBean (EJB), then deploying it to a JBoss application server. Instructions are also included on how to deploy the application to WebSphere and WebLogic application servers.
---	---

Managed COBOL Syntax

Visual COBOL version 2.2 Update 1 includes the following enhancements to the managed COBOL syntax:

Specifying parameters in the method signature	You can now specify passing parameters and returning items in the method signature, instead of using a Procedure Division header. This applies to methods, indexers, iterators, constructors and delegates.
CONSTANT keyword	Use the CONSTANT keyword on a field to protect it from being altered.
Operations on string fields	You can now use the STRING, UNSTRING and INSPECT statements on fields of type string.

Project and Item Templates

Support is now available for using existing projects and files as custom templates to create new projects and files. You create and configure projects that include the files and settings you would like to use as templates.

Remote Connections

Visual COBOL now provides a new connection type, **Micro Focus DevHub using SSH**, that uses a Secure Shell daemon process to launch a server on the remote host.

You can use this type of connection when the UNIX machine you are connecting to uses LDAP authentication which is not supported by the DevHub daemon.

Using this connection also means you do not need to run the DevHub daemon process with root privileges. It also gives you greater flexibility in setting environment variables needed for building or debugging on the remote server.

Rumba Integration with Eclipse

Visual COBOL 2.2 Update 1 provided enhanced integration with the Micro Focus Rumba application for running and debugging Mainframe Subsystem applications that require a TN3270 emulator to run. You can now configure Eclipse to launch a Rumba mainframe display embedded in the IDE or a mainframe session of Rumba Desktop.

Terminfo Files

The following terminfo files have been added:

- *ansi80x25* - this is based on the old *ansi* file. A newer version of *ansi* exists in this release that has no function key support, which is consistent with *ansi* terminfo files on various other UNIX platforms. If you currently use *ansi* and require function key support, you should instead set the TERM environment variable to *ansi80x25* to continue previous behavior.
- *xterm-color* and *kterm-color* - these are now available on all UNIX platforms - previously, they were only available on Linux.
- *aixterm-old* (AIX systems only) - this has similar capabilities to the AIX OS terminfo file of the same name. It differs from the existing *aixterm* file, because it has no line drawing capability. Line drawing is only possible with *aixterm* if it is displayed on an appropriate display (X server).
- *vt220-w* - this is the wide (132-column) version of the vt220 file, and is based on the vt100-w file. For more information, see *Wide Terminal Mode* in the documentation referenced at the bottom of this section.

There have been a number of additions and fixes to existing terminfo files; refer to the *Terminfo Database and Terminal Devices* section of the documentation for full details.

There have also been a number of terminfo files that have been removed; refer to the *Backward Compatibility* section for a complete list.

What was New in Visual COBOL 2.2

Visual COBOL 2.2 provided enhancements in the following areas:

- [ACUCOBOL-GT Compatibility](#)
- [RM/COBOL Compatibility](#)
- [Application Configuration](#)
- [COBOL Source Information \(CSI\)](#)
- [Compiler Directives](#)
- [Consolidated Tracing Facility](#)
- [Enhanced Accept and Display Statements](#)
- [Debugging](#)
- [Grouping Files in Virtual Folders in Solution Explorer](#)
- [File Handling](#)
- [Interface Mapping Toolkit](#)
- [Managed COBOL](#)
- [Problems View](#)
- [Upgrading from Net Express to Visual COBOL](#)
- [XML Extensions](#)

ACUCOBOL-GT Compatibility

The following enhancements are applicable to Visual COBOL:

- Accessing data files through AcuServer - You can now access your ACUCOBOL-GT data files, both sequential and Vision files, through AcuServer.
- Standard library routines - Support for the following library routines has been added:
 - C\$GETPID
 - C\$JUSTIFY
 - C\$LIST-DIRECTORY
 - C\$LOCKPID
 - C\$REGEXP
 - C\$RUN
 - C\$SLEEP
 - C\$SYSTEM

- C\$TOLOWER
- C\$TOUPPER
- I\$IO
- Using Vision files with Micro Focus Data File Tools - You can now use some of the Data File Tools functionality with Vision files. You can:
 - Convert Vision files to Micro Focus format using the Data File Converter and the DFCONV command line utility.
 - Edit Vision files using the Data File Editor.



Note: For more information about the **Data File Tools** utility, see *Data Tools*.

RM/COBOL Compatibility

The following support has been added to Visual COBOL in this release:

- Subprograms - Support for the following subprograms (referred to as library routines in Visual COBOL) has been added:
 - C\$OSLockInfo
 - C\$SecureHash
- recover1 - The recover1 utility, RM/COBOL's indexed file recovery utility, is now distributed with Visual COBOL. Refer to the *RM/COBOL File Handling* section of *RM/COBOL Compatibility* for details of its use.

Application Configuration

You can now set environment variables for when you run native projects from within the IDE from the project's properties - click **Environment** on the **Application** tab in the project properties.

COBOL Source Information (CSI)

COBOL Source Information (CSI) provides a quick and easy way of providing you with information about your program when you are working on it. You enter a query in the **Quick Browse** dialog box and CSI returns the results of the query in the Search view.

Compiler Directives

The following Compiler directives are new:

ACU-UNDERSCORE	This directive treats underscores in COBOL words as hyphens.
ILSHOWPERFORMOVERLAP	This managed COBOL-only directive generates a warning when an overlapping PERFORM range is detected in the program.
IEXPONENTIATION	This managed COBOL-only directive enables you to optimize exponential arithmetic operations by specifying the calculation method used.
EXITPROGRAM	This directive determines how the EXIT PROGRAM statement is executed.

The following Compiler directives have changed

CHANGE-MESSAGE	The scope of this directive has been widened to allow you to change the severity of different types of error messages, not just syntax checking messages.
DIALECT"RM"	DIALECT"RM" now sets PERFORM-TYPE"RM". If you recompile an application that uses DIALECT"RM", the behavior may change for nested PERFORM statements. If that is the case, explicitly set PERFORM-TYPE"MF" after DIALECT"RM" to continue with the previous behavior.

HIDE-MESSAGE The scope of this directive has been widened to allow you to hide different types of error messages, not just syntax checking messages.

PRESERVECASE This directive now defaults to PRESERVECASE when compiling native COBOL; managed COBOL compilation already defaults to PRESERVECASE. This results in externally visible identifiers preserving their case instead of being converted to uppercase.

Consolidated Tracing Facility

The following changes have been made to the Consolidated Tracing Facility (CTF):

CTF for JVM COBOL application CTF tracing is now supported in JVM COBOL applications.

New emitters A new emitter, JAVALOGGER, is available for JVM COBOL web applications. This emitter passes details to the Java logging API that is available through your web server. Use this emitter if you are unable to configure the TEXTFILE or BINFILE emitters due to web server permission restrictions.

New properties and variables for existing emitters The following support has been added to existing emitters.

Properties The following property has been added to the BINFILE emitter:

Property	Description
RunUnitID	Controls whether the RunUnit information is included in the trace.

Variables Four new pseudo-variables for the FILE property have been added to the BINFILE and TEXTFILE emitters:

pseudo-variable	Description
\$(PLATFORM)	A platform specific constant, useful when two run-time systems are in the same process, and you require separate trace files
\$(RUNUNIT)	A unique number that represents the managed RunUnit ID
\$(RUNUNIT_SESSIONNAME)	The session name passed to the managed RunUnit
\$(RUNUNIT_GUID)	The globally unique identifier associated with the managed RunUnit

Enhanced Accept and Display Statements

Two of the existing Enhanced ACCEPT and DISPLAY settings available through Adis have additional values, which are aimed at RM/COBOL users migrating their source code to Visual COBOL. The new values are:

- Emulation of RM/COBOL-85 style data entry for numeric data entry on ACCEPT statements.
- Emulation of an RM/COBOL backspace in free format fields when in replacement editing mode, in that deleted characters are removed and characters to the right are shifted left, the same as when in insertion editing mode.

For more information on how to set these values, refer to *Configuring Enhanced ACCEPT and DISPLAY*.

Debugging

Displaying debug information for managed applications

You can set the DEBUG constant for managed COBOL projects on the **COBOL** tab in the project properties. This enables you to use the System.Diagnostics.Debug class in your applications to ensure they write diagnostic information in the Output window for projects compiled for Debug but not for projects compiled for Release.

Changing the display format for individual items in the Watch window

It is now possible to change the display format for individual items in the Watch window in COBOL. To do this, click a row, press **F2**, and type: *Variable,h* or *Variable,x* to always display the values in hexadecimal format; *Variable,d* to always display the values of variables in decimal format, and of strings - as text.

Grouping Files in Virtual Folders in Solution Explorer

Visual COBOL now provides a Virtual View of a project within Solution Explorer. In the Virtual View you use virtual folders to improve navigation by logically grouping the files that make up the project. You can also create your own virtual folders to group files of your choice (a file can only belong to one virtual folder). The files can be of different file types.

File Handling

New features include:

- Converting and editing Vision and RM/COBOL indexed data files using the Data File tools is now supported.
- Access to data files (either sequential or indexed) through AcuServer is now supported.
- Access to Vision and RM/COBOL indexed data files through Enterprise Server is now supported.

Interface Mapping Toolkit

Visual COBOL now supports the creation and deployment of COBOL program-based services using the Interface Mapping Toolkit (IMTK).

Managed COBOL

Documentation A guide that provides a basic introduction to Object-Oriented Programming (OOP) for COBOL developers, *An Introduction to Object-Oriented Programming for COBOL Developers*, with examples is now available from the *Product Documentation* section on the Micro Focus SupportLine Web site - [click here to download it](#).

Named and optional parameters

Two new types of parameter have been introduced for use during method invocation:

Named parameters

As part of the invocation expression, you can define a value for a parameter named in the method definition. The named argument must be specified after any positional arguments, and must not correspond to any of those preceding arguments.

Optional parameters

Optional parameters are parameters defined with a default value in the procedure division header of the invoked method. If none of the arguments passed in during invocation correspond to this parameter, the default value is used in the method; if an argument does correspond, the value that was passed in is used.

Delegates and events

A number of new features have been added that relate to delegates and events:



Note: Some of these features were also available in previous versions of Visual COBOL.

The ATTACH and DETACH statements	Use these statements to attach or detach a delegate, method group or an anonymous method to or from an event.
The RUN statement	Use this statement to invoke a delegate once it has been created.
Combining delegates	Use the '+' operator to add a method group, anonymous method or another delegate to a delegate, and use the '-' operator to remove a method or another delegate from a delegate.
Method groups conversions	Use the METHOD keyword to specify a compatible method from a method group, and convert it to a delegate.

Problems View

The **Problems** view now has a 'Program' column that displays the name of the program in which the problem occurred. Click this column heading to sort the errors by program.

Program	Description	Resource	Path
	Errors (200 of 262 items)		
ZBNKE15.CBL	COBCH0008S Unknown copybook CABENDD spec...	ZBNKE15.CBL	/Mainframe1
SBANK00P.CBL	COBCH0008S Unknown copybook CBANKDAT spe...	SBANK00P.CBL	/Mainframe1
SBANK00P.CBL	COBCH0217S Preceding item at this level has zero ...	SBANK00P.CBL	/Mainframe1
SBANK00P.CBL	COBCH0008S Unknown copybook CBANEXT spe...	SBANK00P.CBL	/Mainframe1
SBANK00P.CBL	COBCH0217S Preceding item at this level has zero ...	SBANK00P.CBL	/Mainframe1
SBANK00P.CBL	COBCH0008S Unknown copybook CABENDD spec...	SBANK00P.CBL	/Mainframe1
SBANK00P.CBL	COBCH0012S Operand BANK-ENV-CICS is not dec...	SBANK00P.CBL	/Mainframe1
SBANK00P.CBL	COBCH0012S Operand BANK-COLOUR-SETTING i...	SBANK00P.CBL	/Mainframe1

If the column is not present in the view by default, you can add it through the **Configure Columns** dialog box in the **View Menu**.

Support for SOA



Restriction: This topic applies only when the Enterprise Server feature is enabled.

Visual COBOL now includes support for creating Web service and Enterprise Java Bean applications using the Interface Mapping Toolkit (IMTK) in conjunction with Enterprise Server. If you are upgrading to this release from an earlier version of Visual COBOL, you may need to apply for a new authorization code in order to access the functionality - please contact Micro Focus SupportLine to receive an updated authorization code. Note that the Visual COBOL Personal Edition license does not support the IMTK functionality.

Upgrading from Net Express to Visual COBOL

A new section in the product help, *Upgrading from Net Express to Visual COBOL for Eclipse*, provides guidance on how to move existing applications either developed or debugged in the Net Express IDE into the Eclipse IDE.

XML Extensions

You can now use XML Extensions in your managed COBOL projects.

Use XML Extensions 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.

While importing or exporting data to or from XML documents, you can apply XSLT transforms to the data by using XSLT stylesheets.

For more information, refer to the XML Extensions User's Guide, available from the product documentation section of the SupportLine website (<http://SupportLine.MicroFocus.com/ProductDoc.aspx>)

What was New in Visual COBOL 2.1 Update 1

Visual COBOL 2.1 Update 1 provided enhancements in the following areas:

- [Compiler Directives](#)
- [DB2 ECM](#)
- [Enterprise Server](#)
- [DB2 ECM](#)
- [Mainframe Compatibility](#)
- [Debugging Enhancements](#)

Compiler Directives

You can now set SQL Compiler directives and their values more easily, using a table of tick boxes in a project's Properties dialog box.

DB2 ECM

- Support for 64-bit compile and runtime
- Support for DB2 10.1
- New DB2 SQL compiler directive option, BGP, to enable background parsing

Enterprise Server

The following new features and enhancements are available:

Clustering	COBOL Server Clustering allows the scaling-out of work units, so that an increased number of operating system images can share the workload, resulting in high-performance, multi-system data sharing across all platforms.
Historical Statistics Facility	The Historical Statistics Facility has been extended to include the generation of JCL file records, increasing the amount of information customers have available to assist them in monitoring and tuning their COBOL Server installations.
Recovery of in-doubt XA transactions	Some events in XA environments can result in 'in-doubt' transactions, where all parts of a composite transaction are not committed through all participating resource managers. The recovery of such in-doubt transactions is now supported.
SSL Support for the CICS Web Interface	COBOL Server now allows clients and servers to identify themselves through X.509 certificates and participate in SSL-enabled conversations.


DB2 ECM

- Support for 64-bit compile and runtime
- Support for DB2 10.1
- New DB2 SQL compiler directive option, BGP, to enable background parsing.

Mainframe Compatibility

is compatible with IBM System z9 mainframe hardware and later.

Debugging Enhancements

You can create a breakpoint for any program in the workspace that your application uses by clicking **Run > Add Program Breakpoint**, or clicking the  icon in the Breakpoints view, and entering the name of the file.

What was New in Visual COBOL 2.1

Visual COBOL 2.1 provided enhancements in the following areas:

- [ACUCOBOL-GT Data Types in Managed Code](#)
- [ACUCOBOL-GT Library Routines in Managed Code](#)
- [Associating file extensions with the COBOL language](#)
- [Compiler Directives](#)
- [.int, .gnt and .lbr File Types Support](#)
- [Just-in-time Debugging](#)
- [Managed code enhancements](#)
- [OpenESQL](#)
- [UNIX Platforms Support](#)
- [Automatic Directives Detection and Setting](#)

ACUCOBOL-GT Data Types in Managed Code

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

ACUCOBOL-GT Library Routines in Managed Code

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

Compiler Directives

The following new Compiler directives are now available:

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

The following Compiler directive has 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.

Managed code enhancements

Delegates and Events

Delegates and events are now implemented on the JVM platform.

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

Handling Invalid Numeric Data

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

Resolving Types

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

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

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

Specifying Properties

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

OpenESQL

JDBC

JDBC has been enhanced to support two new directives:

JNDI Enables you to specify a JNDI class that looks up connection strings.

JNDIENC Enables applications to use the JNDI Environment Naming Context (ENC) when looking up JDBC data source names using JNDI.

ODBC

Added support for a generic one-phase commit for ODBC XA switch module.

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.
TIMDELIM	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.

OpenESQL Assistant OESQL Assistant now supports updateable cursors.

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.

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.

What was New in Visual COBOL 2.0

Visual COBOL 2.0 provided enhancements in the following areas:

- [Automatic Directives Detection and Setting](#)
- [COBOL Explorer view](#)
- [COBOL File Search](#)
- [Compiler Directives](#)
- [Compiling Single Files](#)
- [Enhancements to Developing Applications on a Remote Machine](#)
- [Copybook Context](#)
- [Debugging Enhancements](#)
- [Eclipse 3.7](#)
- [JVM COBOL File Handler](#)
- [Library Routines](#)
- [Managed COBOL Language Features](#)
- [Data Access](#)
- [Remote COBOL JVM Projects - Early Release](#)
- [Run-Time Tunables](#)
- [Samples](#)
- [Vision Data File Searching](#)

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.

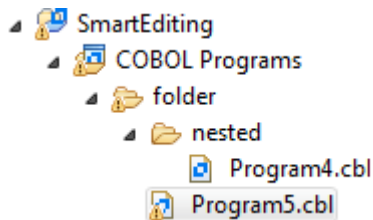
Option	Description
.*resources	Any files of type <code>. *</code> such as <code>.cobolBuild</code> , <code>.cobolProj</code>
Closed projects	Resources in closed projects never display, but you can choose to hide all closed project icons  too
Empty folders inside COBOL categories	Folders mapped to the 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.

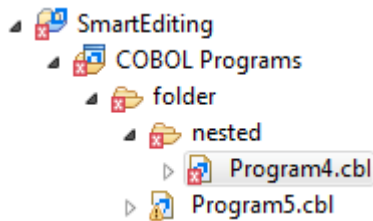
Option	Description
Working Sets	Working sets are subsets of workspace resources you can choose to show or perform options on. To define a working set, click Project > Build Working Set > Select Working Set > New .
COBOL Elements	Non-resource types that are COBOL-specific: <ol style="list-style-type: none">1. <code>x</code>2. category folders, including those for COBOL programs and copybooks3. icons for the different types of COBOL files.

Option	Description
Resources	<p>4. overlays for build errors and warnings</p> <p>5. some context menu items</p> <p>COBOL projects, COBOL programs and copybooks. Has effect only if the COBOL Elements option is unchecked.</p>

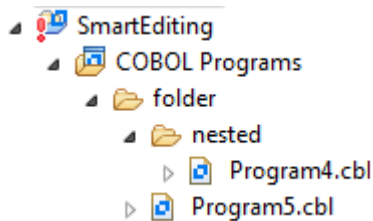
- COBOL Explorer helps you fix problems by using icons to identify files and containers that cause build errors and warnings.
- A file that generates a warning, and any containing folders and categories, is marked with a yellow warning sign. For example:



- 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.

You can use the Micro Focus search feature to make it easier to find files within your projects:

- Select **Search > Micro Focus**, and type the file search pattern.

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.

You can control the scope of your search:

- Workspace - searches entire workspace.
- Selected resources - searches only the resource currently being edited or highlighted in the tree view.
- Enclosing projects - if the editor is focussed on a resource or resources are highlighted in the tree view, then the search is within the entire project containing the resources.
- Current program - searches the resource currently being edited, as well as within any copybooks used by the resource.
- Working set - see *Working Sets* for more information.
- Select **Search > Run Stored Micro Focus 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).

Whichever way you use the Micro Focus search the results are shown in the **Search** view.

After you have performed a search you can edit the properties of any of the program files listed in the search view. You can even simultaneously edit the properties of multiple program files which have compilation settings available on them, that is program files rather than copybooks. Highlight the files you want to change, right-click **Properties**, and then check **Enable file specific settings**. Change the file properties as required and then click **OK**.

You can save your search criteria and give it a label which is added to the list of stored queries available when you select **Search > Run Stored Micro Focus Search Query**:

1. In the Search tab, click  **Save the Current Search**.

This opens the **Add Micro Focus Search** dialog box.

2. In the **Search label** field, type a search label.

- You can assign a keyboard shortcut by typing the required keystrokes in the **Binding** field.
- Optionally, you can modify the current search criteria.

3. Click **OK**.

The search label is added to the list under **Search > Run Stored Micro Focus Search Query**.

You can also add, edit, and remove your searches from **Window > Preferences > Micro Focus > Search**.



Note: Do not open the **Keys** preferences page (**Window > Preferences > General > Keys**) while editing the binding in the **Search** preferences page. Opening the **Keys** preferences page will prevent any changes you make to the search bindings from 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.
- ILSUSING - 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.

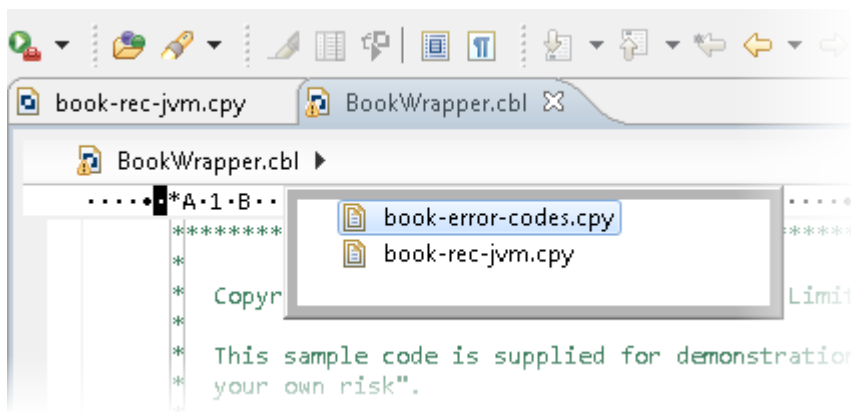
Copybook Context

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


Visual COBOL 2.0 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

Visual COBOL 2.0 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

Visual COBOL version 2.0 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

Visual COBOL 2.0 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

Visual COBOL 2.0 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
```

What was New in Visual COBOL 2010

Visual COBOL 2010 provided enhancements as part of the following releases:

- [New Features in Enterprise Developer 2010 R4 Update 2](#)
- [New Features in Enterprise Developer 2010 R4](#)

New Features 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
- SPARC running Solaris 10
- x86-64 running SuSE SLES 11 SP1

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

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

OO COBOL Class Library Reference

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

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

Features Added in Visual COBOL 2010 R4

ACUCOBOL-GT Compatibility

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

- When using a CALL statement, the USING and GIVING/RETURNING phrases can now appear in either order.

- The following ACUCOBOL-GT standard library routines can now be used with Visual COBOL in native code:
 - C\$CALLED BY
 - C\$CALLERR
 - C\$CHDIR
 - C\$MAKEDIR
 - C\$MEMCPY
 - C\$MYFILE
 - C\$PARAMSIZE
 - C\$RERR
 - M\$ALLOC
 - M\$FREE
 - M\$COPY
 - M\$FILL
 - M\$GET
 - M\$PUT
 - WIN\$VERSION
- The following ACUCOBOL-GT 'ccbl' compiler options can now be used with Visual COBOL:
 - -E, -V
 - -Cv
 - -Da, -Db, -Dd31, -DL1/2/4/8, -Dq, -FpRounding
 - -La, -Li, -Lc, -Lf, -Ll, -Lo, -Ls, -Lw

Note: The output that these list options provide differs in Visual COBOL.

 - -Qm
 - -Rc, -Rn, -Rw
 - -Sa, -St, -Sd, -Sp, -S1...-S9
 - -noTRUNC, -truncANSI, -Dz
 - -Td, -Te
 - -Vc
 - -Za, -Zc, -Zl, -Zn, -Zs, -Zi, -Zr1, -Zy, -arithmeticVSC2

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

COBOL for Java Virtual Machine (JVM) Support

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

Features include support for

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

- Seamless interoperation between COBOL compiled for JVM and COBOL compiled for Java code
- The ability to add Java projects to the classpath for COBOL JVM projects, and COBOL JVM projects to the classpath for Java projects
- Red Hat, AIX and SUSE platforms

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.

Significant Changes

This section describes significant changes in behavior or usage in each successive release. These changes could potentially affect the behavior of existing applications or impact the way the tools are used.

Where present, the numbers that follow each issue are the Support Incident Numbers followed by the Reported Problem Incident (RPI) number (in parentheses).

Significant Changes in Visual COBOL 4.0

This section describes significant changes in behavior or usage. These changes could potentially affect the behavior of existing applications or impact the way the tools are used.

Where present, the numbers that follow each issue are the Support Incident Numbers followed by the Reported Problem Incident (RPI) number (in parentheses).

- [Application Workflow Modeller](#)
- [Code Coverage](#)
- [Codeset Support](#)
- [Common Communications Interface](#)

- [Communications Server](#)
- [Compiler](#)
- [Data Tools](#)
- [Documentation](#)
- [Enterprise Server](#)
- [Enterprise Server Auditing](#)
- [File Handling](#)
- [IDE](#)
- [Interface Mapping Toolkit](#)
- [MF Server Administrator \(GUI\)](#)
- [Micro Focus Directory Server](#)
- [Run-Time System](#)

Application Workflow Modeller

[Back to the list](#)

- Source or listing files are now opened in read-only mode when a browse action is executed in the ChangeMan attachment model.
3138354 (1112546)
- A new function package, Micro Focus Background Parser, enables you to define the SYSLIB or PROCLIB concatenation in which to search for COBOL copybooks, PL/I or JCL include files on the mainframe or on network drives. The package also enables you to define any identifiers which determine the structure of the local cache for COBOL copybooks, PL/I or JCL include files.
3132249 (1111922)
- The Endeavor attachment application now validates the input of a CCID or a comment. They must not be blank.
- Added the new "Modeled Edit Sessions Only" AWM model attribute to control whether modeled editor actions should be available in non-modeled edit sessions.

Code Coverage

[Back to the list](#)

- Schema changes that affect the test coverage results generated from the `tcutil` utility mean that if you propagate the results to a third-party application (for example, an XSLT processor), and rely on the `<copyFileCoverage>` element, you need to alter your transformations to focus on `<sourceFileCoverage>` instead. The element was renamed to more appropriately reflect its contents, as `tcutil` now gives global coverage for all source files (not just copybooks).

Codeset Support

[Back to the list](#)

- Code-set mappings between ASCII and EBCDIC have been updated when Simplified Chinese is the language in effect. ASCII table 5210 now maps to EBCDIC CCSID 836 for SBCS conversions. This replaces the previously conversion (where ASCII table 1042 was used), which would convert the “\” character to “\$”.
3124321 (1111464)
- New single-byte character set tables for MFCODESET have been added in order to improve support for DB2 LUW - both for off-mainframe databases and for access to z/OS DB2. A number of existing MFCODESET mappings have also been updated. See 'Supported Country Codes' for a full listing of ASCII/ANSI <-> EBCDIC mappings.
3111843 (1109984)

Common Communications Interface

[Back to the list](#)

- You can now configure the Micro Focus Directory Server and enterprise server region's listeners to only use the server's configured SSL and TLS protocols and define a priority ordered cipher suite collection. This forces connecting clients to use the server's preferred ordered list of cipher suites when using the specified protocols.

2866265 (1105526)

- In some circumstances it was possible for a connection to incorrectly accept the identity of an SSL/TLS peer and allow a connection to complete when the connection should have been denied. This occurred due to a failure to check the peer's entire identity certificate chain. This has now been fixed. NOTE: You might need to correct your system's configured certificate chains that fail verification checks at secure connection creation time.
- In some circumstances it was possible to crash the CCITCP module when it was attempting to obtain detailed error information about a closed connection. This has been fixed.

Communications Server

[Back to the list](#)

- TN3270 conversations to Enterprise Server now correctly handle the receive (idle) timeout setting configured for the listener. There are also two new settings for configuring TN3270 timeouts, "Printers time out" and "Output resets timeout". See the online product documentation for more information.

3144133 (1113024)

- Web access to the Enterprise Server Console Log and Communications Server Log is now restricted when the region is secured using external security. Users will be required to provide a valid username and password in order to view either log file. Web access to the logs can be controlled using the standard ACL definitions under the new "Communications Server" Resource Class, with resources "Enterprise Server Console Log" and "Communications Server Log". If these resources exist, users require read access to be allowed to view the logs. If they do not exist, the default behavior is to allow read access. The new resource class and resource definitions can be found in the "es_default_ldap(_msuser/_unix).ldf" file in the bin or etc sub-directory of your product install directory.

3113539 (1110155)

- MFCS listeners can now be SSL-enabled without the need to have DemoCA installed.

2868627 (1105777)

- MFCS no longer initializes the Security Facility if there are no External Security Managers defined for the region.

Compiler

[Back to the list](#)

- Programs containing EVALUATE statements of the form: EVALUATE true | false WHEN conditional-expression where conditional-expression included inline method invokes would give an RTS 114 error when run as .int code, and an "Illegal .int code" error when generated. This has been fixed. Also, short circuit evaluation is now correctly observed, such that when evaluating condition-1 AND condition-2, if condition-2 contains an inline method invoke and condition-1 is false, then the inline method invoke is not executed. Previously, despite being correctly evaluated, the inline method invoke in condition-2 was being executed. Similar behavior relating to OR evaluations has also been corrected.

3138510 (1112492)

- During compilation, characters within literals that are unknown in the current locale are now less likely to cause spurious errors. However, the correct (and safest) solution is to ensure that the locale has been set correctly, to match the source encoding of these characters. On UNIX, this means setting LANG, LC_CTYPE, or LC_ALL appropriately; each of these variables takes precedence over the former. On Windows, this means setting the system locale in the 'Region and Language' section of Control Panel.

3123935 (1111148)

- The Compiler now produces an E level message - COBCH1888 Typedef is defined differently in another external program - if different external programs have conflicting definitions of the same typedef name. (To restore the previous behavior, where the earlier definition was ignored, use the directive `HIDEMESSAGE"1888".`)
- An issue with the Compiler has been fixed so that in the `RECORD VARYING` clause, if the minimum and maximum lengths are specified, the maximum length must be greater than the minimum length.

Data Tools

[Back to the list](#)

- Records with an invalid value for a conditional field will no longer result in a match for that conditional layout.

2853226 (1103406)

Documentation

[Back to the list](#)

- You can use 'byte' or 'BYTE' as a synonym for the binary-char unsigned data type. As a result, 'byte' and 'BYTE' are now reserved words in Managed COBOL. Use the `REMOVE"BYTE"` Compiler directive to prevent an error being produced for existing programs that use the reserved word as a user-defined word.

3147576 (1113323)

- As of version 3.0, references to types within an assembly other than `mscorlib` need to be explicitly referenced. You can achieve this by using the `ILREF` Compiler directive. (Previously, in certain circumstances, the Compiler would allow access to types within the `System.dll` assembly without the need for an `ILREF"System"` directive.)

3121002 (1111373)

Enterprise Server

[Back to the list](#)

- The External Security Facility (ESF) can now be configured to throttle large volumes of incoming Verify (user authentication / signon) requests to improve resilience to denial-of-service and brute force attacks. See "Verify Request Throttling" for more information.

3113639 (1110160)

- The LDIF files used to create the sample configuration for Enterprise Server LDAP-based security no longer create an empty "PHYSFILE" resource class. Changes in the JCL engine as of ES 3.0 caused most jobs to fail when submitted to a security-enabled region using such a configuration. See the product help for more information.
- The MLDAP ESM Module, part of the Enterprise Server External Security Facility, now supports the Argon2 hash algorithm for creating password verifiers. See MLDAP ESM Module Custom Configuration Information in the product help for more information. NOTE: The Argon2 hash is optional and not enabled by default.
- The MQ pages in ESMAC are now controlled by a new security resource, MQL. This enables you to either restrict or grant users access. 'MQL*' is a new resource that needs to be added under MFESMAC similar to existing resources such as 'PCT*' or 'XAT*'. The following is a sample export of the LDAP repository:

```
*****
# Sample security definitions for ESMAC MQ Listeners/Writers pages

#####
##### MQL*          ##
```



```
#####
dn: CN=SQL*,CN=MFESMAC,CN=Enterprise Server Resources,CN=Micro
Focus,CN=Program Data,DC=X
changetype: add
cn: SQL*
objectClass: microfocus-MFDS-Resource
microfocus-MFDS-Resource-Class: MFESMAC
microfocus-MFDS-Resource-ACE: allow:SYSADM group:alter
microfocus-MFDS-Resource-ACE: deny:*:execute
microfocus-MFDS-UID: mfuid
#description: Allow full access any ESMAC MQ Listeners/Writers Screen

*****
```

3143258 (1112990)

- You can now use the `ECIResponse.getReturnCode()` method to obtain the return code for any errors from Enterprise Server.

3142092 (1113248)

- Communication with the console daemon has been improved. Messages are displayed more quickly and requests are being processed more efficiently and, as a result, times for initialization and shutdown might be reduced.

3136867 (1112483)

- Administrators can now add, delete or modify XA resources in the Enterprise Server Administration Web UI while a region is running.

2589624 (1085625)

- In UNIX environments, DB2 and ODBC switch modules can now be enabled together in the same region if the DB2 switch is built with the "-o" option.

3137455 (1112398)

Enterprise Server Auditing

[Back to the list](#)

- The `maxRetryTime` value in the audit configuration file now treats 0 as a no timeout time, and any negative number as an infinite timeout.

3150566 (1113592)

- Any extra information that was added to the syslog messages will now correctly appear in the structured data items.

File Handling

[Back to the list](#)

- In some cases the ESF LDAP Security Administration Web Interface inadvertently removed users from groups when changing their password. This has been fixed.

3124294 (1111259)

- The ESF LDAP Security Administration Web Interface can now filter by Class and Resource name, description, and ACL. Previously, you could only filter on Class name.

2871549 (1106119)

- Setting the configuration option `ASCIISOSI=ON` will add the required SOSI characters to the relevant EBCDIC DBCS character strings, in order for them to be displayed or written out correctly.

3113802 (1110183)

- The OPEN mode of SYSOUT files now honors the DISP specified in the JCL.

3109432 (1109745)

- OPEN I-O of a virgin ESDS file now correctly returns a file status of 35, as it does on the mainframe.
2887724 (1108443)

IDE

[Back to the list](#)

- New functionality to support the debugging of CICS channels and containers has been added to this release.
2810448 (1098047)

Interface Mapping Toolkit

[Back to the list](#)

- Refresh Resources functionality has been added to regenerate existing service interfaces for COBOL programs whose linkage section has been updated after the service interface was initially created.
2849403 (1103036)
- Due to changes made to the **Configure Runtime Environment** dialog box in this release, after you upgrade your system, you must reenter all previously set values on this dialog box before running a service.

MF Server Administrator (GUI)

[Back to the list](#)

- The total number of active sessions or clients in MFDS is now limited to 2000.

Micro Focus Directory Server

[Back to the list](#)

- The mfd -g options D, O, and S have been added to the product Help.
2848627 (1102864)
- On UNIX, specifying an invalid user ID in the MFDS "General" options value for "Default process user ID" no longer causes the child process (such as a region start or stop request launched from the Web interface) to fail. The user ID under which the MFDS process was started will be used instead.
- MFDS now disables and limits the scope of Web listeners on add. It also emits a warning if any insecure Web listeners are displayed in the validate and listener tables.

Run-Time System

[Back to the list](#)

- The run-time system now produces a more precise error message if a shared object of the wrong bitism is loaded.
- scan64 is no longer available. This has been superseded by the COBOL Analysis functionality in the IDE.

Significant Changes in Visual COBOL 3.0

Visual COBOL version 3.0 includes significant changes in the following areas:

- [Compatibility AddPack](#)
- [Compiler](#)
- [Documentation](#)
- [Enterprise Server](#)

- [File Handling](#)
- [IDE](#)
- [Micro Focus Directory Server](#)
- [OpenESQL](#)
- [Reserved words](#)
- [SQL Option for DB2](#)

Compatibility AddPack for Visual COBOL

Compatibility AddPack for Visual COBOL is now deprecated and will not be available with release 3.0 and later.

The Dialog System GUI and run-time components and Dialog System Character Mode (on Windows and UNIX) which were part of the AddPack are now installed as part of Visual COBOL for Visual Studio. The run-time components are installed as part of COBOL Server. These are only included for backward compatibility and Micro Focus does not recommend that you use them for new development.

The other components which were part of the AddPack, the Character-Based Data File Editor, CSBIND and Screens, will be available upon request from Micro Focus SupportLine.

Compiler

- Replacing a partial token no longer causes the second part of the token to appear on a new line. This could happen if the new text was larger than the text being replaced.

2869185 (1105763)

Documentation

- There have been a number of new reserved words added to the language in this release; these are all in effect under MFLEVEL"19", which is the default level when running under the MF dialect. Any of the following words are now not allowed under default conditions, and you will need to remove/rename them, or specifically configure your environment to allow them: ALLOCATE FREE JSON END-JSON

Enterprise Server

- WEB CONVERSE now supports a value of 0 for the USERLEN and PASSWORDLEN options which matches the behavior on the mainframe. There is no change to the behavior of WEB SEND (client) which is to return LENGERR 139/140 when USERLEN or PASSWORDLEN are 0.

2989188 (1108602)

IDE

- You can now open data files in RSE configured remote projects using the mfdatatools2 on the remote machine displaying back to an X display. You can configure the remote tools display by clicking Window > Preferences > Micro Focus > X Display. In the X Display (DISPLAY) field, type the display details. In addition, COBOL Explorer displays the Open With Remote Data File Editor context menu item for remote data files.

2852872 (1104085)

- Enterprise Server sign on credentials which were created on Windows when using the 32-bit versions of Enterprise Developer or Visual COBOL for Eclipse are not compatible with the 64-bit versions of those products, and vice versa. Users are required to re-enter credentials which are not compatible with the current version of the product. You only need to do this the first time they are used in the current product.

(625819)

File Handling

- Under certain circumstances, retry-lock requests on UNIX systems were sleeping for a second before re-attempting to acquire the lock. This no longer happens.

2988222 (1108521)

- A problem that generated a 39 error when attempting to access a VSAM file via an alternate index PATH element has been fixed.

2874622 (1106562)

Micro Focus Directory Server

- In the Enterprise Server Administration HTML GUI, the "Scripts" page functionality is only available if administration access is restricted and the logged on user has sufficient authority.

3101625 (1109025)

- Some additional CSRF security measures have been added to the Enterprise Server Administration HTML GUI.

3101205 (1108916)

OpenESQL

The new OpenESQL OPTIMIZECURSORS SQL compiler directive option is turned on by default for ODBC (DBMAN=ODBC). This ensures that embedded SQL cursors that use WITH HOLD and FOR UPDATE clauses have the same data integrity across all databases.

If your applications require the OpenESQL preprocessor to use the behavior provided in an earlier release, compile them using OPTIMIZECURSORS=NO.

Reserved words

- There have been a number of new reserved words added to the COBOL language; these are all in effect under MFLEVEL"19", which is the default level when running under the MF dialect. Any of the following words are now not allowed under default conditions, and you will need to remove/rename them, or specifically configure your environment to allow them:

ALLOCATE

FREE

JSON

END-JSON

SQL Option for DB2

- Help buttons previously available on the XDB Server Configuration Utility, XDB Service Controller, Options Dialog, Bind Utility, and Linker Config (Link Profile) UIs have been removed with the exception of error messages in the SQLWizard, Migrate, and Declaration Generator.

Significant Changes in Visual COBOL 2.3 Update 2

Visual COBOL version 2.3 update 2 includes significant changes in the following areas:

- [Enterprise Server](#)
- [Compiler](#)
- [Eclipse IDE](#)
- [MF Directory Server](#)
- [Monitoring and Management](#)
- [Run-Time System](#)

Compiler

- Replacing a partial token no longer causes the second part of the token to appear on a new line. This could happen if the new text was larger than the text being replaced.
2869185 (1105763)
- There is no longer a problem opening an RM/COBOL indexed file when the program has a RECORD CONTAINS n CHARACTERS clause and there are record descriptions with lengths less than n. This situation previously caused a 39 error on the OPEN (other than OPEN OUTPUT) because there was a mismatch in the minimum record length.

Eclipse IDE

- This update modifies any existing connections defined in an Eclipse workspace. If the connections had any non-default values, those values could revert to their original default setting. After installing this release, before you use any remote connections for the first time, you need to check the settings and amend them as necessary.
2852872 (1103699)

Enterprise Server

- Previously, it was possible to install groups that should not have been installed. If a group name, as defined in the Startup List, did not exist in the list of Groups then the next Group in the alphabetical order would be loaded instead. Now, if a Group is not defined in the list of Groups, a warning that the Group could not be loaded is issued.
2869848 (619107)
- On UNIX, if the "File Path" setting was not specified in the configuration of an Enterprise Server, the environment variable TXFILEP was defaulting to \$COBDIR/etc/cas. This has been changed and TXFILEP is not populated when the "File Path" is not specified.
(618668)

MF Directory Server

- The "-n" option for the mfd command now supports hostnames as the network addresses in addition to IPv4 addresses.
2816871 (1099564)

Monitoring and Management

- Messages that are written to the console log by applications that perform "display upon console" now contain a standard message ID (CASMG0001).
2854207 (1103659)

Run-Time System

- The command_line_linkage tunable has been deprecated; equivalent functionality can be achieved by using the COMMAND-LINE-LINKAGE Compiler directive instead.
2838118 (1101539)

Significant Changes in Visual COBOL 2.3 Update 1

Visual COBOL version 2.3 update 1 includes significant changes in the following areas:

- [Application Workflow Modeller](#)
- [Data Tools](#)

- [Editor Writing Assistance](#)
- [IDE](#)
- [Run-Time System](#)
- [SQL: OpenESQL](#)
- [SQL Option for DB2](#)

Application Workflow Modeller

- Tools used in the File_Descriptor_Has_Action relationship now support Resource Processing. This means tools defined within a dialog table action can modify properties of the selected table rows.
2835290 (1101493)
- The standard ED project model now contains additional context menu actions under "Open With..." for IMS DBD and PSP files to open either the DBD, the PSP or the IMS Database editors.
- You no longer receive a REXX error when recompiling a component from a baseline.

Data Tools

- When filtering a data file, if there is no valid temporary directory set, you are prompted to set one using the option in the Preferences dialog box.
- The editor no longer allows you to open a file if the file size (without header size) is not a multiple of the record size on disk; an error is produced instead.
- The editor no longer allows you to open a file if the file size without header size is not a multiple of the record size on disk; an error is produced instead.
- The level numbers displayed in a record layout correspond to the levels used in the .idy file that was used when the structure file was created.

Editor Writing Assistance

- IntelliSense (Visual Studio) or Content Assist (Eclipse) suggestions are no longer offered if you start typing numbers and automatic triggering of suggestions is enabled.

IDE

- When compiling to multiple executables from the command line, you must specify the -logger parameter to enable the correct log to be output to the console.
2848855 (1102932)
- For a file added to a JVM COBOL project by adding the folder that stores the file (using the **Source** tab on the **Micro Focus > JVM Build Path > Source page** in the projects' properties), there is now a new context-menu command, **Copy to Output Directory**, in COBOL Explorer. Choosing this command on a file triggers a build of the project and copies the file into the output directory.
2699374 (1094326)

Run-Time System

- The Audit Manager contains a new TIMEOUT option. When a client sends an audit event using the 'CBL_AUDIT_EVENT' API, the event gets placed in the next available slot in a shared memory block. If shared memory is full (i.e. no slots are available), the event is re-tried until a slot becomes available.

If no Audit Manager is running, no events are removed from shared memory, and no slots will ever become available. Therefore, use the new TIMEOUT option so that a client will only retry sending until the TIMEOUT duration is reached; after which, it will stop sending audit events. If Audit Manager is recycled, events will start to be sent again.

To set the TIMEOUT for all Audit Manager clients, specify the following line in the Audit Manager configuration file:

```
mfaudit.timeout = n
```

Where n is the timeout value in milliseconds.

To set the TIMEOUT for an individual Audit Manager client, use the 'CBL_AUDIT_CONFIG_PROPERTY_SET' API. It takes an integer property-value, which should be the timeout value in milliseconds.

If TIMEOUT is set using both methods, the client property TIMEOUT takes precedence, unless this property is set to zero; in such cases, the TIMEOUT in the configuration file is used. If you use the 'CBL_AUDIT_CONFIG_PROPERTY_GET' API on the 'TIMEOUT' property, it only returns the TIMEOUT value for the client property; it does not return the value set in the configuration file.

2838689 (1101685)

- Several changes have been made to the implementation of IS DBCS, IS KANJI and IS JAPANESE class condition tests:

- IS [NOT] DBCS

When CHARSET"EBCDIC" is in effect, the IS DBCS test returns true when each character in the string is deemed to be a valid DBCS character. A valid character has its first byte in the range 0x41 through 0xFE, and the second byte in the range 0x41 through 0xFE, or the character is an EBCDIC space (0x4040). When CHARSET"ASCII" is in effect, the DBCS test uses an OS call to determine if the string contains only valid double-byte character, and returns true if valid.

- IS [NOT] KANJI

When CHARSET"EBCDIC" is in effect, the IS KANJI test returns true when each character in the string is deemed to be a valid Kanji character. A valid character has its first byte in the range 0x41 through 0x7F, and the second byte in the range 0x41 through 0xFE, or the character is an EBCDIC space (0x4040). When CHARSET"ASCII" is in effect, the IS KANJI test uses an OS call to determine if the string contains only valid Kanji character, and returns true if valid.

- IS [NOT] JAPANESE

When CHARSET"EBCDIC" is in effect, the IS JAPANESE test is not supported, and will generate a COBCH1806 Feature not supported in selected charset message on compilation.

When CHARSET"ASCII" is in effect, the IS JAPANESE test returns true when the string contains only double-byte Japanese characters or single-byte Japanese Katakana characters, and returns true if valid. When NSYMBOL"NATIONAL" is in effect, these class tests are not supported, and will generate a COBCH0303 Operand has wrong data-type message on compilation.

2812895 (1098401)

SQL: OpenESQL

- The DB2 CONCAT function and operator now convert to SQL Server using the HCOSS-supplied dbo.CONCAT for character, numeric and datetime data. If you are using BINARY or VARBINARY data, you must apply the HCOSS-supplied dbo.CONCAT_BINARY function. HCOSS applications deployed with earlier versions of Visual COBOL are affected, if they use string or binary concatenation. The mainframe dialect DB2 || operator and CONCAT function now call a new SQL Server scalar function dbo.CONCAT(). All existing programs with dialect=mainframe that use DB2 concatenation syntax should be recompiled. All existing SQL Server databases that are accessed by these programs must have dbo.CONCAT installed. To create the new function in your application's SQL Server database, you can either:

- Run a DSN bind against the customer database. Or:
- Execute the %ALLUSERPROFILE%\Micro Focus\Enterprise Developer\hross\InstallDigitsFunction.sql script.

This is a one-time only change to the database.

2843818 (1102248)

SQL Option for DB2

- Spurious errors were sometimes returned while querying using an ALIAS.

2830383 (1100609)

Significant Changes in Visual COBOL 2.3

Visual COBOL version 2.3 includes significant changes in the following areas:

- [CAS Security](#)
- [CAS XA Switch modules](#)
- [Compiler](#)
- [File Handling - External File Handler](#)
- [File Locking](#)
- [IDE](#)
- [J2EE Connector](#)
- [MF Server Administrator \(GUI\)](#)
- [Updated Run-Time System](#)

CAS Security

- The Enterprise Server External Security Facility now includes MLDAP ESM Module 2.0, with a new algorithm for identifying the best-matching resource-access rule and ACE for resource-access security checks. This algorithm is faster and matches most customers' expectations. The new algorithm also provides an optional "username substitution" feature. It can be enabled by setting "rule substitutions" to "yes" in the [Operation] section in the Security Manager configuration text area. When this is enabled, the string "\${user}" in a resource-rule name will be replaced with the name of the user that makes the request. For example, a DATASET rule named "USERS.\${user}.*" would apply to datasets with the requesting user's name as the second qualifier. In rare cases, customers with complex, ambiguous resource-access security rules might see experience changes in behavior as a result of the new algorithm. The old algorithm is still supported and can be enabled by setting "version 1 authentication" to "yes" in the [Operation] section of the Security Manager configuration.

2807531 (1097783)

CAS XA Switch modules

- The XA switch modules now support dynamic registration.

2682101 (1092325)

- The XA switch modules now support batch-only operations when multiple XA Resource Managers have been defined.

2664675 (1091082)

- In Visual COBOL 2.2 update 2, Micro Focus identified undefined run-time behavior when the following combination of directives was specified: SIGN"EBCDIC", CHARSET"ASCII", and one of the following: HOST-NUMMOVE, HOST-NUMCOMPARE or SIGN-FIXUP. Previously (Visual COBOL 2.2 update 1 and earlier), if this combination was specified, the SIGN"EBCDIC" directive should have been ignored, to avoid a mixture of ASCII and EBCDIC characters; however, SIGN"EBCDIC" was still being honored, resulting in undefined run-time behavior. Therefore, this combination of directives is now invalid for Visual COBOL 2.2 update 2 or later, and if specified, will be rejected at compile time.

2786397 (1095265)

Compiler

- For native COBOL, the size limit of the Data Division now stands at 2GB -1.
2796076 (1096384)
- Previously, it was not possible to specify sign(EBCDIC) with sign-fixup, host-num-move or with host-num-compare. This combination is now supported in native COBOL but remains invalid for managed COBOL code. This is applicable to version 2.2 U2 HotFix 10 onwards.
2824577 (1100823)

File Handling - External File Handler

- Custom file handlers (using DYNREDIR) are now called for each part of a concatenated file.
2795077 (1096322)

File Locking

- In versions prior to Visual COBOL 2.3, the semantics of the sharing phrase specified in an OPEN statement or used within a call to CBL_OPEN_FILE were not correctly applied in some cases on UNIX and Linux platforms. From version 2.3 onwards, the sharing phrase is correctly honored when the tunable `strict_file_locking=true` is set, which is the default setting.

Example of potential changes in behavior:

- *Process-A* opens a file with read-only access and a sharing mode that denies other processes write access (SHARING WITH READ ONLY).
- *Process-B* then attempts to open the file with read-only access and a sharing mode that denies other processes read access (SHARING WITH NO OTHER).

With `strict_file_locking=true`, *Process-B* is unable to open the file, because *Process-A* has successfully opened the file allowing only read access.

With `strict_file_locking=false`, *Process-B* successfully opens the file.

If your application encounters unexpected OPEN conditions or fails to open files, it might be as a result of the new file locking behavior. In such circumstances, we recommend that you review the file locking and sharing requirements of your application and refactor your source code to work with the default setting. The original file locking and sharing behavior can be restored by setting `strict_file_locking=false`.

IDE

- Visual COBOL for Eclipse now ships with Eclipse 4.4.2 (Luna). If your applications contain JVM COBOL code that was built with a previous version of the product, those parts of your application must be rebuilt; otherwise you will experience errors at either compilation or run time.
(609469)
- A project can have one of two connection modes: NFS/Samba, where the target location is mounted as a local drive, and RSE, which is a purely remote connection to the target location. For project types such as Mainframe COBOL, NFS/Samba is required to use some tools which do not support RSE. For most other projects, it is possible to switch between connection modes using the **Remote Settings** context menu option. In the dialog box, there are radio buttons allowing for selection of connection modes. On changing mode, you must select an existing connection of the appropriate type, or create a new one. If switching to NFS/Samba mode, you must specify the local path to the project.
2792882 (1096196)
- By setting `"-Denable.projectrepair=true"` in the eclipse.ini file, .cobolProj and .pliProj files will be repaired to reflect the workspace on project refresh.
2696707 (1095994)

J2EE Connector

- Visual COBOL version 2.3 provided a new command-line argument to Java, `mf.ssl.algorithm`, which can be set to an appropriate algorithm.

2799213 (1096684)

MF Server Administrator (GUI)

- Passwords that entered through either the MFDS or the ESMAC interface now use the same encoding.

2792382 (1096011)

Updated Run-Time System

- COBOL Server now provides an execution environment capable of running applications that were each built using different development products. A consequence of this is that if your application has a main COBOL executable (.exe) that was built with a version of Visual COBOL prior to version 2.3, you should ensure that the executable is rebuilt and packaged with the new run-time system. You can rebuild from the IDE or the command line.

Other COBOL subprograms built with previous versions of Visual COBOL are not required to be rebuilt.

Significant Changes in Visual COBOL 2.2 Update 2

Visual COBOL version 2.2 update 2 includes significant changes in the following areas:

- [Compiler](#)
- [Compiler Front-end](#)
- [Documentation](#)
- [J2EE Connector](#)

Compiler

- When using the HOSTRW directive with the mainframe dialect, Report Writer will now produce the full range of ASA control characters and will emulate mainframe print files.

2697615 (1094527)

Compiler Front-end

- Fixed Binary (`p<=7`) is now an 8-bit, signed, 2's complement binary integer by default.

Documentation

- The default setting for the MFALLOC_PCFILE environment variable has changed; the default is now set to Y, which means that when cataloguing a file that has a DCB attribute of DSORG=PS, a physical file is created for it if one does not exist. Previously, the default was set to N, which meant that a file was not created.

2697571 (1094370)

J2EE Connector

- The `listSystem.properties` file in package `com.ibm.ctg.client` was missing documentation for some sections.

(606556)

Significant Changes in Visual COBOL 2.2 Update 1

Visual COBOL version 2.2 update 1 includes significant changes in the following areas:

- [SQL: COBSQL](#)

SQL: COBSQL

- COBSQL now displays appropriate COBOL syntax errors after encountering EXEC SQL statement errors.
2673619 (1093197)

Significant changes in Visual COBOL 2.2

Visual COBOL version 2.2 included significant changes in the following areas:

- [Compiler](#)
- [Eclipse IDE](#)
- [Interface Mapping Toolkit](#)
- [MF Directory Server](#)
- [MF Server Administrator \(GUI\)](#)
- [MFBSI](#)
-
-
-
-
-
-
- [Request Handler](#)

Compiler

- The default for the NSYMBOL directive under DIALECT(ENTCOBOL) has been changed to NSYMBOL(NATIONAL) to emulate the equivalent IBM default.
2657471 (1090355)
- To improve RM/COBOL and ACUCOBOL compatibility, the SIGN clause at a group level is no longer applied to non-DISPLAY usage signed numeric data items within the group, just as it is not applied to unsigned numeric data items and non-numeric data items within the group.
2549904 (1082171)
- Previously, even though no code was generated, the Compiler allowed the ON EXCEPTION and NOT ON EXCEPTION phrases in the DISPLAY statement in formats that do not allow these phrases. As a result, if the DISPLAY statement was in the ON EXCEPTION phrase of another statement, the NOT ON EXCEPTION phrase would bind incorrectly with the DISPLAY statement instead of with the intended containing statement - for example, ACCEPT or CALL.

Eclipse IDE

- The Problems view now has a 'Program' column that displays the name of the program in which the problem occurred. If the column is not present in the view by default, it can be added through the Configure Columns dialog box in the View Menu.
2608496 (1088530)
- The outline of the ruler in the COBOL editor now changes on modelChanged depending on the current source format.

2488419 (1077143)

- Deleting a Web service or a Java interface mapping now deletes the files relating to it from the repos directory and, if it is empty after the files have been deleted, will remove the repos directory itself.
- The IDE now performs an automatic directive determination when files are added to a project. You can also use a command to perform directive determination of the project manually.

Interface Mapping Toolkit

- For program-based Service Interfaces, if the program-id name in the COBOL source is in lowercase and is not surrounded by quotes, its corresponding entry-point name is now forced to uppercase when used in a Service Interface Operation. Previously, the case was preserved. As a result of this change, existing Service Interfaces will become invalidated if you refresh their program's annotations because of the new spelling of the entry-point name. To avoid this, you need to surround the program-id name in the COBOL source with quotes before you refresh the annotations.

MF Directory Server

- The mfdx command line option for exporting registered Enterprise Server definitions to an XML file now supports the "*" option. This exports all registered servers rather than a specified server. Multiple server definitions are now exported into the target directory and saved into a file with the default name ALLSERVERS.xml. The import option now also supports the import of multiple server definitions from a single XML file.

2641890 (1088838)

- mdump now supports a new option, -e, to help you query the Security Manager configuration details. The possible values of the option are: "1" - shows security configuration that applies to any returned enterprise servers; "2" - shows security configuration for MFDS and the default Enterprise Server security configuration. This requires MFDS version 1.15.00 or higher; "3" - returns the properties of all configured external Security Managers.

2487164 (1081693)

MF Server Administrator (GUI)

- When adding a user to an external security manager, you can now include a password expiry time in the Advanced Configuration section of the Add New User wizard in Enterprise Server Administration. The field value is specified using generalized time format (YYYYMMDDHHMMSS.OZ), and can be used by the MLDAP ESM for calculating whether a user's password has expired and requires updating. This value may only be specified using this page when adding a user. You need to use an external directory services configuration tool to edit it.

2562118 (1083203)

Request Handler

- A problem that caused BIS to create log files in a directory named C:\ProgramData\AcuCorp\BIS\LogFiles was fixed. BIS no longer creates log files unless specified and the BIS logging service is now disabled by default. To enable it, you need to use the following global environment variable:
BIS_LOG=[OFF | ON | <directory>] Where the values are:
 - OFF - disables logging (the same as if BIS_LOG is not specified or is left blank)
 - ON - enables logging and directs the log files into the default location, which must not be read-only.
 - <directory> - enables logging and directs the log files into the specified directory. The user must ensure that the BIS request handler has write rights for this directory. The directory must be an absolute path or network path. If the specified directory does not exist, BIS will attempt to create it. The containing directory must exist.

The BIS_LOG variable is only examined when the BIS application pool is started or recycled. After setting or changing BIS_LOG, IIS must be restarted in order for the variable to take effect.

Significant Changes in Visual COBOL 2.1 Update 1

Visual COBOL version 2.1 update 1 includes significant changes in the following areas:

- [Documentation](#)
- [IDE](#)

Documentation

- To ensure no loss of functionality when accessing Vision and RM/COBOL data files, you should use the appropriate IDXFORMAT Compiler directive setting or file handling option, and not use the CALLFH(ACUFH) Compiler directive. See 'Configuring Access to Vision Files' and 'Configuring Access to RM/COBOL Data Files' for more information.

IDE

- You can now set program breakpoints as follows - click "Add Program Breakpoint" in the Run menu or on the Breakpoints view toolbar, and enter the program name in the dialog that is displayed.
- Local Enterprise Server regions without secure user credentials are now started with casstart and stopped with casstop commands.

Significant Changes in Visual COBOL 2.1

Visual COBOL version 2.1 includes significant changes in the following areas:

- [IDE](#)
- [Run-Time System](#)
- [Vision File System](#)

IDE

- 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)

Run-Time System

- 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.

2579335 (1084817)

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.

2553438 (1082469)

Significant Changes in Visual COBOL 2.0

Visual COBOL version 2.0 includes significant changes in the following areas:

- [Compiler](#)
- [DB2](#)
- [File Handling](#)
- [IDE](#)

Compiler

- The scope of the ILUSING Compiler directive when used in a \$set command has changed. The scope of the directive is now limited only to the source file it is set in, and not globally. This new behavior may mean that your source files no longer compile. To resolve this, add the required ILUSING statements to the required individual source files, or add the ILUSING directive on the command line. Alternatively, use the IDE to achieve the required behavior: in Visual Studio, use the Namespaces tab; in Eclipse, set the directive in the Additional Directives field.

DB2

- The DB2 ECM has been updated to resolve run-time errors returned when compiling against mainframe databases in 64-bit mode.
2549058 (1082441)

File Handling

- When reading a file cataloged as DISP=SHR the file handler now buffers the read for better performance.
2518330 (1079491)

IDE

- Typing in the sequence area moves text after the cursor to the right and doesn't delete the characters at area end. To overwrite characters in the area, switch to insert mode by pressing the Insert key. Pressing the Tab key in the sequence area indents the text if the option "Pressing Tab in the sequence number area moves caret to area A" is not selected. Pressing the Tab key when there is a multiline selection aligns the text to the area margins.
- Typing in the sequence area now works like typing in a regular text editor - it moves to the right all the text after the cursor and doesn't delete the characters at area end. If you want to overwrite characters in the area, you can switch insert mode to overwrite by using "Insert" key.

Unsupported or Deprecated Functionality

The following topics describe functionality that was removed or deprecated at each product release.

Unsupported or Deprecated at Visual COBOL 4.0

The following features or functionality are no longer supported or are deprecated at version 4.0:

- Support for Eclipse 4.4 and 4.5 has been discontinued.

Unsupported or Deprecated at Visual COBOL 3.0

The following features or functionality are no longer supported or are deprecated at version 3.0:

- The HOSTSIGNS Compiler directive is no longer supported. Micro Focus recommends that you use the following Compiler directives instead: SIGN-FIXUP, HOST-NUMMOVE, and HOST-NUMCOMPARE.
- Compatibility AddPack for Visual COBOL - this is now deprecated and will not be available with release 3.0 and later.

The Dialog System GUI and run-time components and Dialog System Character Mode (on Windows and UNIX) which were part of the AddPack are now installed as part of Visual COBOL for Visual Studio. The run-time components are installed as part of COBOL Server. These are only included for backward compatibility and Micro Focus does not recommend that you use them for new development.

The other components which were part of the AddPack, the Character-Based Data File Editor, CSBIND and Screens, will be available upon request from Micro Focus SupportLine.

- Audit Manager is deprecated and provided for backward compatibility only. We recommend that you use syslog events instead. See *Enterprise Server Auditing* for more information.
- The following DB2 environment variables are deprecated at version 3.0, and provided for backward compatibility only.
 - HCOBND - Micro Focus recommend you use either the BIND or the BINDDIR compiler directive option.
- The following compiler directives are deprecated at version 3.0, and provided for backward compatibility only.
 - CONVERTRET
 - IDYSRCPATH
 - ILOBJECTIFY
 - OPTION
 - SPZERO
 - TRICKLE
- The following file handling options are deprecated at version 3.0, and provided for backward compatibility only. Micro Focus recommend you use IDXFORMAT"8" instead:
 - STRIPING
 - MAXSTRIPEDIGITS
 - MAXSTRIPEFILES
 - MAXSTRIPESIZE
 - STRIPE-X
 - STRIPENAMETYPE -

Unsupported or Deprecated at Visual COBOL 2.3 Update 2

The following features or functionality are no longer supported or are deprecated at version 2.3 Update 2:

- The `command_line_linkage` tunable has been deprecated; equivalent functionality can be achieved by using the `COMMAND-LINE-LINKAGE` Compiler directive instead.

Unsupported or Deprecated at Visual COBOL 2.3 Update 1

The following features or functionality are no longer supported or are deprecated at version 2.3 update 1:

Unsupported or Deprecated at Visual COBOL 2010

The following Net Express features or functionality were not supported or were deprecated from the first version of Visual COBOL:

These Net Express compiler directives are not supported by Visual COBOL :

01SHUFFLE
64KPARA

64KSECT
AUXOPT
CHIP
COBIDY
DATALIT
EANIM
EDITOR



Note: The directive SPZERO was already deprecated from Net Express 5.0 onwards, and is provided for backward compatibility only. You should instead use *SIGN-FIXUP*.

and the pseudovariabes of the following Net Express environment variables are obsolete and cannot be used:

BASENAME
FILENAME
PATH
TARGETDIR

Known Errors and Restrictions

Errors and restrictions in previous versions are not listed in this document; those that still apply in version 4.0 are included here, while any that have been fixed may occur in the *What was New* or *Significant Changes* sections.

Refer also to the *Known Issues and Restrictions* topic in the *Product Information* section of your product Help.

The following errors and restrictions are known to apply to Visual COBOL version 4.0:

- [Application Workflow Manager](#)
- [Commuter licenses](#)
- [Debugging](#)
- [Database Access](#)
- [Enterprise Server](#)
- [imtkmake](#)
- [Installation](#)
- [OpenSSL](#)
- [Resource Adapters](#)
- [REST/JSON IMTK implementation and the same-origin policy \(SOP\)](#)
- [Viewing the Product Help](#)

Application Workflow Manager

-

Commuter licenses

If there are a number of entries displayed in the License Administration tool, these might be either for different products or for the same product but for different serial numbers, for example.

When a commute request is made using the license manager, that request is serviced from a pool of licenses that could be provided by any number of Micro Focus product licenses stored on that server. The commute request cannot target a particular RMS license by product name, serial number, or number of cores.

The products to which you apply the commuter licenses will be licensed correctly but the number of licenses left in the License Administration tool might appear to be less than expected. This issue is more likely to occur with multiple entries for the same product.

If you are using a number of different products, Micro Focus recommends that you use a different licensing server for each product in order to help minimize the issue.

Debugging

- When debugging a remote program, the performance of the network, specifically the latency between the local and remote machines, will have an effect on the responsiveness of Eclipse. Whenever Eclipse has to show the new execution position, a number of messages are sent from the remote machine, over the network, to get the required information. The effect of this is that it can take a few seconds after a Step command is issued before Eclipse has updated the screen and is ready for the next command.

You can mitigate this delay by closing any unnecessary debug views. For instance, by closing the Expressions view and the Variables view, the number of messages that are sent is reduced, which allows Eclipse to update quicker.

- Large programs can suffer from a large delay the first time that the program is displayed in the debugger and an expression is evaluated. You can reduce this delay by following these steps:

1. Create a text file called `debugconfig.xml`.
2. Add the following lines, and then save the file:

```
<?xml version="1.0" encoding="utf-8" ?>
<NativeDebuggerOptions>
  <DebugOption Option="REMOTECHECKER" Value="yes" />
</NativeDebuggerOptions>
```

3. At a command or shell prompt, set the environment variable `DEBUG_CONFIG` to the full path name for `debugconfig.xml`.
 4. Start Eclipse from the same command or shell prompt.
- Remote debugging does not work for programs running on AIX or HP machines, if you are trying to debug using Visual COBOL installed on a Linux machine.
 - `core_on_error=3` does not work correctly on AIX 7.2.

Database Access

- If you have a remote COBOL project under Eclipse that uses DB2 ECM in Visual COBOL or HCO for DB2 LUW in Enterprise Developer and the DB2 software is not installed on the client machine where you are using the Eclipse IDE, you receive background parsing errors in your application. To resolve the issue, you can do either one of the following:
 - Disable background parsing **Window > Preferences > Micro Focus > COBOL > Editor** and disabling the checkbox for **Background parsing**.
 - Install the IBM DB2 client-side software on the machine on which you are running Eclipse. Go to the IBM Support Home and locate the page entitled "IBM Download Fix Packs for IBM Data Server Client Packages" .

Enterprise Server

- The Historical Statistics Facility may generate incorrect records for SSTM-enabled enterprise servers.

imtkmake

- On AIX v7.2, you must install IBM's patch APAR IJ04075 in order for `imtkmake -genclient` to work correctly. See http://www.microfocus.com/docs/links.asp?vc=IBM_APAR_IJ04075 for details about how to install the patch.

Installation

- Installing this product fails if Micro Focus License Manager version 10000.2.02070 (April 2015) or older is already installed on your machine. The setup files does not upgrade the License Manager automatically to the latest version. You must uninstall the existing version before you start the installation.

OpenSSL

The version of OpenSSL included in Visual COBOL and Visual COBOL 4.0 is intolerant of expired X.509 certificates in the CA Root Certificate collection in some circumstances. Specifically, if you configure a PEM file that contains multiple certificates as the collection of trusted CA certificates, and one or more of the certificates in that file have expired, OpenSSL might not be able to load and use other certificates in the file which are still valid. This can cause certificate validation errors and connection failures for SSL and TLS connections.

To work around this, you need to remove any expired certificates from such collections. If necessary, replaced them with any valid certificates that have the same Distinguished Name and Modulus (key). This is a known issue with OpenSSL. Micro Focus is investigating for other possible workarounds.

Resource Adapters

- Trying to deploy the local resource adaptor `mfcobol-localtx.rar` to WebLogic may fail with a `ClassCastException`. To work around this issue, you need to deploy `mfcobol-xa.rar` first, then need to undeploy this file and deploy the local one, `mfcobol-localtx.rar`. If there are issues deploying using the WebLogic GUI, you can use the command line. If there are issues with this as well, try reducing the length of the command (for example, by moving the file to a location with a shorter path).

REST/JSON IMTK implementation and the same-origin policy (SOP)

- HTTP requests sent from scripts within a web browser to REST services deployed on an enterprise server might fail due to the same-origin policy (SOP). Some browsers might implement (or support plugins that implement) techniques for relaxing SOP such as cross-origin resource sharing (CORS) that enable sending cross-origin requests successfully. For REST services, Enterprise Server does not implement a method for relaxing SOP. This means that browsers that implement CORS (or any other SOP relaxation technique) might still forbid requests made from scripts due to Enterprise Server not implementing the equivalent technique on the server side.

Viewing the Product Help

- A known issue in JRE 1.7.0_45 (issue JDK-8028111) causes problems with searching and indexing the help in the Eclipse help viewer if you are running Eclipse on a machine that has this version of JRE installed.

Upgrading from Server Express

You upgrade existing distributed applications created with Server Express to Visual COBOL for Eclipse.

The Visual COBOL suite includes the following two independent products that are designed to be used together for maximizing the developers' productivity:

- Visual COBOL Development Hub is the counterpart of Server Express and must be installed on the machine that has your application sources.
- Visual COBOL for Eclipse provides a rich and highly effective integrated development environment that can be connected to Visual COBOL Development Hub to enable you to edit, build and debug the applications' sources directly on the remote machine.

The following sections provide information on how to upgrade.

Advantages of Upgrading to Visual COBOL

Visual COBOL offers numerous enhancements and a modern IDE that offers advantages such as:

Improved productivity	<p>This is achieved through:</p> <ul style="list-style-type: none">• Using the very latest technology of integrated development environments (IDEs) in a powerful IDE.• Fast and easy customization with major source control systems, simplifying integration with existing development lifecycle leading to higher developer acceptance and productivity gains.• Providing the tools to allow you to edit, compile, debug, execute and test COBOL applications on all supported Micro Focus UNIX and Linux environments. One development tool can now meet your development requirements today and in the future.• Enabling cross-language application development which results in better quality and increased competitiveness.• Achieving faster time to market.
Improved efficiency	<p>Improve efficiency when performing day-to-day development activities, enabling access to Eclipse, smart editor, instantaneous compilations, advanced test and debugging capabilities.</p>
Addressing of the COBOL skills shortage	<p>Using COBOL from within a standard IDE helps reduce the gap between the COBOL and Java or C# developers by providing a unifying, collaborative and dynamic development environment. Non-COBOL developers can quickly and easily adapt to the COBOL language, and get up to speed in no time.</p>

For a full list of the advantages of using Visual COBOL, see [the Visual COBOL data sheet and white papers on the Micro Focus Web site](#).

Upgrading Your COBOL Application



Note: Upgrading your existing applications might be a complex task, therefore we recommend you contact Micro Focus Professional Services for assistance with this process.

If you've never used Eclipse before, Micro Focus recommends that you read the chapter *Introduction to Eclipse* in the product help of Visual COBOL for Eclipse to get familiar with the IDE before you start upgrading your applications.

Prerequisites

In most situations when you are upgrading to Visual COBOL, you will:

- Keep your source code on the UNIX machine. You need to install Visual COBOL Development Hub on this machine.
- Install Visual COBOL for Eclipse on the your local Windows machine. You need to create a connection in Visual COBOL to the remote UNIX machine and then create remote projects for the application's source files.

On single-user Linux machines you do not need to install Visual COBOL Development Hub. Instead, you can install Visual COBOL for Eclipse on the same machine as the application files and maintain the application directly on the Linux machine.

Upgrading your applications

The following outline shows an example approach to upgrading an application to Visual COBOL:

1. Assess your existing processes for building and testing your applications.

Discover any similarities and differences between your existing applications and what programming issues they resolve. Determine what project types and scenarios you need to handle in Visual COBOL for Eclipse.

Review the Compiler directives that are used to build the applications. Review these to ensure they are still valid.



Note: You can use Micro Focus COBOL Analyzer to check your existing applications and assess what might need to be changed or what functionality is no longer supported.

2. Compile your existing sources at the command line:

- a. Ensure Visual COBOL Development Hub is installed on the machine that stores your sources.
- b. From a terminal window, set the environment for Visual COBOL Development Hub and build the existing application. Resolve any compilation issues and ensure that the application behaves as it did previously.

3. Create a Visual COBOL project to handle your source code:

- a. Install Visual COBOL for Eclipse on a local Windows machine.
- b. In Eclipse, create a connection to the machine that includes your sources and create a remote project.
- c. Add the application sources to the new project and set the following Compiler directives on the project:

```
SOURCETABSTOP"8"  
COPYEXT",CBL,CPY"
```

- d. Add the Compiler directives that were used to compile the original project and are still valid to the new project.
- e. Build and debug the application using Eclipse.

Ensure you get the same results using Eclipse as you did when you used Visual COBOL Development Hub on the remote machine (see step 2b).

4. Add more projects, programs and copybooks as required.
5. Review the *Best Practice in Development Using Eclipse* information in this document to ensure you achieve maximum productivity.
6. Experiment with various debugging options to find those that best meet your requirements.
7. Compile for debug and for release and ensure that the application performs as expected.
8. Create one or more template projects and distribute those across the entire development team.
9. Examine ways in which Eclipse can help to modernize your applications and processes.

For more details about these steps see the following sections.

Compile Your Existing Programs without a Project

Visual COBOL includes a lot of enhancements and changes and this is why applications that were compiled using earlier Micro Focus products must be recompiled from the sources using Visual COBOL.

Micro Focus recommends you first compile without a project as you would have done in Server Express. This will help you focus on resolving any compilation issues following the changes in the environment. Then, you can look at creating a Visual COBOL project in order to take advantage of all the benefits offered by the IDE.

To compile your existing programs from a terminal window directly on the UNIX machine:

1. Ensure that Visual COBOL Development Hub is installed on the remote machine where you store your existing COBOL programs.
2. Log onto the remote machine and from a terminal window, configure the environment and compile your programs using the `cob` command or the build scripts or makefiles used to build the application with Server Express.

Remember to review and reuse all Compiler directives that were previously used when compiling the application in Server Express.

Most programs created with Server Express should compile cleanly using your existing build scripts and makefiles without any changes to your code, as Visual COBOL also provides the `cob` utility to create `.int`, `.gnt` or `.o` files.

Fixing compilation issues

Because Micro Focus continues to enhance the COBOL language, you might encounter some problems when compiling your existing Server Express applications in Visual COBOL.

For example, the list of reserved COBOL words changes with each new level of the COBOL language, so an application created with an older Micro Focus product might use data names that are now reserved keywords in Visual COBOL. If this happens, a COBOL syntax error COBCH0666 ("Reserved word used as data name or unknown data description qualifier") occurs. See *Reserved Words Table* in the product help of Visual COBOL for a comprehensive list of reserved words and the levels at which they are supported.

Also, these Server Express Compiler directives are no longer supported:

01SHUFFLE	ENSUITE	REGPARAM
64KPARA	EXPANDDATA	SEGCROSS
64KSECT	FIXING	SEGSIZE
AUXOPT	FLAG-CHIP	SIGNCOMPARE
CHIP	MASM	SMALLDD
COBIDY	MODEL	TABLESEGCROSS
DATALIT	OPTSIZE	TRICKLECHECK
EANIM	OPTSPEED	WB2
EDITOR	PARAS	WB3
	PROTMODE	WB

and the pseudovariabes of the following Server Express environment variables are obsolete and can't be used.

BASENAME
FILENAME
PATH
TARGETDIR

You should consider using the following methods to solve such issues:

- Rewrite the source to avoid using these keywords in your code and directives files.
- Use the REMOVE Compiler directive to remove individual keywords from the reserved words list.
- Use the MF or MFLEVEL compiler directive to select an earlier version of Micro Focus COBOL that your code is compatible with. For example, setting `MFLEVEL "12"` ensures compatibility with Mainframe Express 3.0 and 3.1, Net Express 4.0, 5.0, and 5.1, and Server Expresss 4.0, 5.0, and 5.1. Refer to *Reserved Words Table* in the product help of Visual COBOL for the value to use to ensure support for your existing reserved words.

Setting REMOVE and MFLEVEL Compiler directives at the command line

To use REMOVE from the terminal window (or later in the editor in Eclipse), type the following:

```
cobol myprogram.cbl remove(title) ;
```

The command above removes TITLE as a keyword from the language so you can use it as an identifier in a COBOL program.

To use the set of reserved words that was used for Server Express v5.1 WrapPack 5, use this command line:

```
cobol myprogram.cbl mflevel"15" ;
```

Setting REMOVE and MFLEVEL directives in the source code

To set either one of the directives in your source code, type the following starting with \$ in the indication area of your COBOL program:

```
$set remove "title"
```

Or:

```
$set mflevel"15"
```

Create a Project and Import Your Sources

Once you have ensured your programs compile cleanly in the Visual COBOL environment, you can create a project for them so that you can take full advantage of all of the benefits offered by the Eclipse debugging environment. You need Visual COBOL for Eclipse installed on your local Windows machine and you need to create a connection in it to the remote UNIX machine.

Connect Visual COBOL to the remote machine

From your local installation of Visual COBOL create a new remote connection to Visual COBOL Development Hub on the UNIX machine which has your existing COBOL programs. See *Creating connections to remote hosts* in the product help of Visual COBOL.

Create a remote project

You need to create a project that will store the source files on the remote machine:

1. Start Visual COBOL.

If this is the first time you have started Visual COBOL you see the **Welcome** page in the IDE. You can also open the **Welcome** page by clicking **Help > Welcome**.

2. Click **Open COBOL perspective** in the **Welcome** page.

In this perspective, Eclipse offers you the views and commands that are best suited for COBOL development.

3. Click **File > New > Remote COBOL Project**.

Choose the remote connection to use and follow the instructions to create a remote native COBOL project in a directory on the remote machine.

Import the COBOL source files

Add the source files on the remote machine to your project as follows:

1. Right-click your project in the **COBOL Explorer** view and click **Import > Import**.
2. In the **Import** wizard, expand **General**, click **File System**, and then click **Next**.
3. Click **Browse**, navigate to the location of your source files on the remote machine, and click **OK** to confirm the path.
4. In the **Import** wizard, enable the check boxes for the files you want to import in your project, and click **Finish**.

This adds the files to your project and, if Eclipse uses the default settings, starts a build. You might need to set some Compiler directives to resolve any compilation issues, similarly to when you resolved any issues when compiling from a terminal window.

Configure your project

To configure your project and specify any Compiler directives:

1. Right-click your project in COBOL Explorer, and click **Micro Focus > Project Settings > COBOL**.
2. In the **Additional directives** field, specify any directives that you had to use to resolve any issues that occurred when you compiled without a project. These could be the Compiler directives that were used to compile the application in Server Express prior to the upgrade or new ones.
3. Also specify the following Compiler directives:

```
SOURCETABSTOP"8"  
COPYEXT",CBL,CPY"
```

If your source code contains tab stops, compilation might fail, because although a COBOL tab is eight characters long, the IDE's tab is four characters long, and lines of code might be starting in the sequence number and indicator areas section (columns one to seven) of the program instead of from column eight.

You can fix this problem using the `SOURCETABSTOP(n)` Compiler directive, where *n* is the number of space characters by which to expand tab characters during compilation.

4. Click **OK**.

Build the project

By default, Eclipse builds your source files each time you make a configuration change or add or edit the files. If you have changed this preference, to build the project click **Project > Build Project**.

Add copybooks

To add existing copybooks to your project, right-click the project and use the **Import > Import** command.

Copybooks are not compiled at build time and the files' **Build Action** property is automatically set to **None**. (You can also set this property for COBOL source files to keep a file in the project but not include a built version in any output.)

By default, Visual COBOL identifies files as copybooks by their `.cpy` extension. You can specify other file extensions as copybooks in the IDE preferences - click **Window > Preferences > Content Types**, click **COBOL Copybook** and add any new file associations using this dialog box.

Best Practice in Development Using Eclipse

Here is a list of recommendations about how to make the most of using Eclipse in real-life scenarios.

Break down large projects

Projects that contain a large number of source files and build artefacts can be hard to navigate and slow to build. If you find this is the case with your application, Micro Focus recommends that you review the contents of large projects and split them into separate projects in which you group items that are logically related.

For example:

- If you have different versions of a product for different customers, keep common source in one project and have a separate project for each customer.
- If you have core code that is rarely changed or recompiled, keep that in one project and have separate projects for those areas that change regularly.

Reference common sources

Use the COBOL Copybook project type to store all your copybooks and add this project as a dependency to the projects that contain any source programs that reference the copybooks. Use this approach in preference to using linked resources.

Use linked folders to reference resources where it is not possible to restructure your projects (to use the COBOL Copybook project type) or use a remote project (to access any sources on remote machines).

Use existing Compiler directives files

To avoid repetition and reduce maintenance effort, consider keeping all your Compiler directive settings in a directives file. To reference this file in each project, specify the `USE "filename"` Compiler directive in the **Additional directives** field in each project's properties.

Use relative paths

Keep your sources relative to a base path and avoid specifying full paths in you projects. This ensures your code is portable and easy to use with source control systems.

Access sources on remote machines

- For optimal performance, create a remote project from the available project types to access any sources on remote machines.
- Use RSE or SSH in preference to Samba type remote connections. Only use the Samba type connection when it is not possible to use the other types of connections. This is because the Samba connection has the slowest performance.
- Avoid using network shares or drives to directly access any resources on remote machines from your local projects.

If for some reason, it is not possible to use local versions of the sources, you can use network shares as linked folders but the connection will be very slow. In such cases, you might be able to improve the performance by splitting the remote shared folder into smaller chunks then set the `COBCPY` environment variable before you start Visual COBOL for Eclipse on your local machine.

Eclipse workspace

- To avoid performance issues, your Eclipse workspace must not be on a shared drive.
- Do not commit the Eclipse workspaces into a source control system, as they contain metadata which is updated frequently.

Configure Visual COBOL

To boost your productivity, enable some additional views in Eclipse that are suitable for the programming tasks you're dealing with within the current active perspective. To do this click **Window > Show View** and either select a view from the ones that are displayed, or click **Other** to choose from all available ones.

For example, in the Debug perspective, enable the **Expressions** view that lists any data items you want to monitor. Or enable the **Remote Systems** view to see all remote systems that have a connection defined in Eclipse and to launch terminal sessions from within the IDE.

Modernize Your Applications and Processes

Following industry-standard development practices

Many source code control systems and Agile tools can be integrated into the Eclipse IDE.

You should also consider using continuous integration, which involves the automatic building and testing of an application after a change occurs to the source code. This method traps errors sooner in the development life cycle and can greatly improve efficiency and reduce costs.

Multi-user applications

Visual COBOL includes a Run Unit API to enable multiple users to simultaneously use an application based on COBOL code that was designed originally for a single user.

Developing Web-based applications

You can use Visual COBOL to migrate existing, core applications to a service-oriented architecture as Web services, and deploy them using Micro Focus COBOL Server and Enterprise Server, so that you can develop COBOL-based software components to be invoked across the Web.

Developing JVM applications

You can compile both new and existing COBOL as JVM COBOL managed code. This enables you to:

- Reuse existing COBOL business logic and data access in the Java Virtual Machine
- Access Java classes and features from COBOL applications
- Create and extend composite applications consisting of COBOL and Java code
- Reuse and extend Open ESQL applications

Both procedural and Object-Oriented (OO) COBOL are supported within the JVM. OO COBOL classes can inherit classes written in Java and vice versa.

The managed COBOL syntax includes many extensions to the COBOL language to support Java features; for example, the TRY ... CATCH syntax to enable exception handling in COBOL.

There are also certain directives that help integrate your managed COBOL with Java. For example, you can now expose the Linkage Section and entry points in your COBOL to other managed languages by compiling with the ILSMARTLINKAGE directive.

Procedural COBOL Compared with Managed COBOL

Procedural COBOL is regular COBOL without any of the new syntax that has been added for .NET and JVM. This is the COBOL that will have been used to write Net Express, Server Express and Mainframe Express applications, and it is still actively supported today.

You can compile to native or (in most cases) managed code. The core COBOL syntax is supported in managed code. However, there are some features that are not supported (for example Panels V2, Dialog System and ACUCOBOL-GT). This means that you can take most existing COBOL applications and recompile to create managed applications.

Managed COBOL

Managed COBOL is the collective term for .NET COBOL and JVM COBOL.

Managed COBOL is COBOL with extensions to support the .NET and JVM frameworks. It offers OO syntax support and syntax to allow access to the available class libraries.

When you compile managed COBOL the compiler generates managed code: `.class` or `.jar` for JVM which will run on the Java Virtual Machine.

Managed Code and Native Code

You can compile your COBOL program to managed code using the `jvmgen` compiler directive. From within the IDE this happens automatically if you are using a managed COBOL COBOL JVM project.

The compiler has now created an intermediate language (JVM byte code `.class/.jar`).

COBOL and all other JVM languages (for example Java, JRuby and Jython) compile to this format, which makes mixed language applications easy to write.

You can also create native code applications. In Eclipse the default COBOL project compiles to native code.

The compiler generates `.exe/.dlls` as the result of a native compilation.

The native COBOL application has to call the appropriate management services available for the operating system, whereas a managed application can take advantage of the management services provided by the run time such as exception handling, garbage collection, and thread management.

Run Time

The JVM byte code (`.class/.jar` files) can be deployed to a JVM for execution.

The JVM's just-in-time (JIT) compiler compiles the byte code into code native to the operating system. The JVM provides additional services including memory management, exception handling, garbage collection and thread management.

Developing Native and Managed Applications

You use the IDE to develop, compile and debug both native and managed applications. You can write new COBOL code or you can recompile existing COBOL applications to managed or native code, potentially without any code changes.

You can deploy and further debug the application under the run-time system provided by COBOL Server.

JVM COBOL applications are deployed to a Java Virtual Machine for execution.