# Cumulative Changes from Server Express to Micro Focus Visual COBOL Development Hub 5.0

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

2019-06-20

# Contents

# Cumulative changes from Server Express to Visual COBOL Development Hub

Welcome to Visual COBOL Development Hube. This document combines information on Visual COBOL Development Hub 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 Development Hub.

## 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.

## Changes to Compiler Directives, Run-Time Tunables, Library Routines and Environment Variables

This section provides an index of items added, deprecated and changes since the first version of Visual COBOL. The changes are presented in a number of tables according to the type of feature involved. The first set of tables lists changes in alphabetical order. A second section lists changes in order of release.

### Changes in alphabetical order

The changes are presented in the following tables:

- *Compiler Directives*
- *Run-Time Tunables*
- *Library Routines*
- *Environment Variables*

For each change, the table lists:

- The name of feature that changed,
- The type of change,
- The version of Visual COBOL in which the change was introduced,
- A link to more information.

**Table 1: Changes to Compiler Directives**

| Name | Change | At Product Version | See |
|---|---|---|---|
| **01SHUFFLE** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **64KPARA** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **64KSPECT** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **ACU-UNDERSCORE** | New | 2.2 | *What was new in 2.2* |
| **ANIM** | Changed default | 5.0 | *Significant changes in 5.0* |
| **AUXOPT** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **CHANGE-MESSAGE** | Changed scope | 2.2 | *What was new in 2.2* |
| **CHECKDIV** | New parameter | 4.0 | *What was new in 4.0* |
| **CHIP** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **COMMAND-LINE-LINKAGE** | New | 2.3 Update 2 | *What was new at 2.3 Update 2* |
| **COMP1** | New | 2.1 | *What was new in 2.1* |
| **COMP2** | New | 2.1 | *What was new in 2.1* |
| **CONVERTRET** | Deprecated | 3.0 | *Unsupported or Deprecated at 3.0* |
| **DIALECT"RM"** | Changed effect | 2.2 | *What was new in 2.2* |
| **DISPLAY** | New | 2.1 | *What was new in 2.1* |
| **DISPSIGN** | New | 4.0 | *What was new in 4.0* |
| **DATALIT** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **DATAMAP** | New parameter | 2.1 | *What was new in 2.1* |

| Name | Change | At Product Version | See |
|---|---|---|---|
| **DBSPACE** | New parameter | 2.3 | *What was new in 2.3* |
| **EANIM** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **EBC-COL-SEQ** | New | 2.3 Update 2 | *What was new at 2.3 Update 2* |
| **EDITOR** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **ENSUITE** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **EOF-1A** | New | 2.3 | *What was new in 2.3* |
| **EXITPROGRAM** | New | 2.2 | *What was new in 2.2* |
| **EXPANDDATA** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **FIXING** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **FLAG-CHIP** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **GNTLITLINKSTD** | New | 4.0 | *What was new in 4.0* |
| **HIDE-MESSAGE** | Changed scope | 2.2 | *What was new in 2.2* |
| **HOSTSIGNS** | Unsupported | 3.0 | *Unsupported or Deprecated at 3.0* |
| **IDXFORMAT8** | New option | Visual COBOL Base release | |
| **IDXFORMAT12** | New option | 2.3 Update 1 | |
| **ILEXPONENTIATION** | New | 2.2 | *What was new in 2.2* |
| **IDYSRCPATH** | Deprecated | 3.0 | *Unsupported or Deprecated at 3.0* |
| **ILMAIN** | New | 2.3 Update 1 | *What was new in 2.3 Update 1* |
| **ILNORMALIZENAMES** | New | 5.0 | *What was new in 5.0* |
| **ILOBJECTIFY** | Deprecated | 3.0 | *Unsupported or Deprecated at 3.0* |
| **ILPARAMS** | New | 2.2 Update 1 | *What was new in 2.2 Update 1* |
| **ILSHOWPERFORMOVERLAP** | New | 2.2 | *What was new in 2.2* |
| **ILSMARTRESTRICT** | New | 2.1 | *What was new in 2.1* |
| **ILSMARTTRIM** | New | 4.0 | *What was new in 4.0* |
| **INIT-BY-TYPE** | New | 2.2 Update 1 | *What was new in 2.2 Update 1* |
| **JVMDECIMAL** | New | 2.3 | *What was new in 2.3* |
| **JVMTARGET** | New | 5.0 | *What was new in 5.0* |
| **MAINFRAME-FLOATING-POINT** | New | 4.0 | *What was new in 4.0* |
| **MASM** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |

| Name | Change | At Product Version | See |
|---|---|---|---|
| **MODEL** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **NATIONAL** | New | 2.3 Update 2 | *What was new at 2.3 Update 2* |
| **NLS-CURRENCY-LENGTH** | New | 2.3 | *What was new in 2.3* |
| **NULL-ESCAPE** | New | 2.3 | *What was new in 2.3* |
| **NUMPROC** | New parameter | 4.0 | *What was new in 4.0* |
| **OOCTRL** | New | 2.3 Update 1 | *What was new in 2.3 Update 1* |
| | New parameter | 4.0 | *What was new in 4.0* |
| **OPTIMIZECURSORS** | New | 3.0 | *Significant changes in 3.0* |
| **OPTION** | Deprecated | 3.0 | *Unsupported or Deprecated at 3.0* |
| **OPTSIZE** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **OPTSPEED** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **PARAS** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **PRESERVECASE** | Changed scope | 2.2 | *What was new in 2.2* |
| **PROTMODE** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **REGPARM** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **RESTRICT-GOTO** | New | 2.1 | *What was new in 2.1* |
| **SEGCROSS** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **SEGSIZE** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **SIGNCOMPARE** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **SMALLDD** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **SPZERO** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **TABLESEGCROSS** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **TRICKLE** | Deprecated | 3.0 | *Unsupported or Deprecated at 3.0* |
| **TRICKLECHECK** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |

| Name | Change | At Product Version | See |
|---|---|---|---|
| **WB2** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **WB3** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |
| **WB** | Deprecated | Visual COBOL Base release | *Unsupported or Deprecated at 2010* |

**Table 2: Changes to Run-time Tunables**

| Name | Change | At Product Version | See |
|---|---|---|---|
| **command_line_linkage** | Deprecated | 2.3 Update 2 | *Unsupported or Deprecated at 2.3 Update 2* |
| **default_cancel_mode** | New default value<br>New parameter | 2.3 | *What was new in 2.3 - Tunables* |
| **putenv_interface** | New | 2.3 | *What was new in 2.3 - Tunables* |
| **reduce_java_signals** | New | 2.2 Update 2 | *Significant changes in 2.2 Update 2* |
| **strict_file_locking** | New | 2.3 | *What was new in 2.3 - Tunables* |
| **subsystem_cancel_mode** | New | 2.0 | *What was new in 2.0 - Run-time Tunables* |
|  | New parameter | 2.3 | *What was new in 2.3 - Tunables* |

**Table 3: Changes to Library Routines**

| Name | Change | At Product Version | See |
|---|---|---|---|
| **CBL_CREATE_CORE** | New | 5.0 | *What was new in 5.0* |
| **CBL_CTF_COMP_PROPERTY_GET** | New | 2.0 | *What was new in 2.0* |
| **CBL_CTF_TRACE** | New | 2.0 | *What was new in 2.0* |
| **CBL_CTF_TRACER_LEVEL_GET** | New | 2.0 | *What was new in 2.0* |
| **CBL_CTF_TRACER_GET** | New | 2.0 | *What was new in 2.0* |
| **CBL_CTF_LEVEL** | New | 2.0 | *What was new in 2.0* |
| **CBL_CODESET_SET_MAPPING** | New | 2.3 Update 1 | *What was new in 2.3 Update 1* |
| **CBL_GET_DEBUG_START** | New | 4.0 | *What was new in 4,0* |
| **CBL_GET_DEBUG_STOP** | New | 4.0 | *What was new in 4,0* |
| **CBL_GET_OS_INFO** | New | 4.0 | *What was new in 4,0* |
| **CBL_LOCATE_FILE** | New parameter | 2.3 | *What was new in 2.3* |
| **CBL_MANAGED_SESSION_GET_USERDATA** | New | 2.3 | *What was new in 2.3* |

| Name | Change | At Product Version | See |
|---|---|---|---|
| **CBL_MANAGED_SESSION_SET_USERDATA** | New | 2.3 | *What was new in 2.3* |
| **CBL_RUNTIME_ERROR** | New | 2.3 Update 1 | *What was new in 2.3 Update 1* |
| **CBL_SEMAPHORE_ACQUIRE** | New parameter | 2.0 | *What was new in 2.0* |
| **CBL_CREATE_CORE** | New | 5.0 | *What was new in 5.0* |

**Table 4: Changes to Environment Variables**

| Name | Change | At Product Version | See |
|---|---|---|---|
| **COB_LIVE_RECORD_SIZE** | New | 5.0 | *What was new in 5.0* |
| **ES_CAT_RETRY_COUNT** | New | 4.0 | *Configuring Multiple Catalogs* |
| **ES_CAT_RETRY_WAIT** | New | 4.0 | *Configuring Multiple Catalogs* |
| **FSWRKDIR** | New | 3.0 | *What was new in 3.0* |
| **FSCHKLFH** | New | 3.0 | *What was new in 3.0* |
| **HCOBND** | Deprecated | 3.0 | *Unsupported or Deprecated at 3.0* |
| **MFALLOC_PCFILE** | Default value changes | 2.2 Update 2 | *Significant changes in 2.2 Update 2* |

# Changes in order of release

The changes are presented in the following tables:

- *Compiler Directives*
- *Run-Time Tunables*
- *Library Routines*
- *Environment Variables*

For each change, the table lists:

- The version of Visual COBOL in which the change was introduced,
- The name of feature that changed,
- The type of change,
- A link to more information.

**Table 5: Changes to Compiler Directives**

| At Product Version | Name | Change | See |
|---|---|---|---|
| Visual COBOL Base release | **01SHUFFLE** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **64KPARA** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **64KSPECT** | Deprecated | *Unsupported or Deprecated at 2010* |

| At Product Version | Name | Change | See |
|---|---|---|---|
| Visual COBOL Base release | **AUXOPT** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **CHIP** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **DATALIT** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **EANIM** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **EDITOR** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **ENSUITE** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **EXPANDDATA** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **FIXING** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **FLAG-CHIP** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **IDXFORMAT8** | New option | |
| Visual COBOL Base release | **MASM** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **MODEL** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **OPTSIZE** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **OPTSPEED** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **PARAS** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **PROTMODE** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **REGPARM** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **SEGCROSS** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **SEGSIZE** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **SIGNCOMPARE** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **SMALLDD** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **SPZERO** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **TABLESEGCROSS** | Deprecated | *Unsupported or Deprecated at 2010* |

| At Product Version | Name | Change | See |
|---|---|---|---|
| Visual COBOL Base release | **TRICKLECHECK** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **WB2** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **WB3** | Deprecated | *Unsupported or Deprecated at 2010* |
| Visual COBOL Base release | **WB** | Deprecated | *Unsupported or Deprecated at 2010* |
| 2.1 | **COMP1** | New | *What was new in 2.1* |
| 2.1 | **COMP2** | New | *What was new in 2.1* |
| 2.1 | **DISPLAY** | New | *What was new in 2.1* |
| 2.1 | **DATAMAP** | New parameter | *What was new in 2.1* |
| 2.1 | **ILSMARTRESTRICT** | New | *What was new in 2.1* |
| 2.1 | **RESTRICT-GOTO** | New | *What was new in 2.1* |
| 2.2 | **ACU-UNDERSCORE** | New | *What was new in 2.2* |
| 2.2 | **CHANGE-MESSAGE** | Changed scope | *What was new in 2.2* |
| 2.2 | **DIALECT"RM"** | Changed effect | *What was new in 2.2* |
| 2.2 | **EXITPROGRAM** | New | *What was new in 2.2* |
| 2.2 | **HIDE-MESSAGE** | Changed scope | *What was new in 2.2* |
| 2.2 | **ILEXPONENTIATION** | New | *What was new in 2.2* |
| 2.2 | **ILSHOWPERFORMOVERLAP** | New | *What was new in 2.2* |
| 2.2 | **PRESERVECASE** | Changed scope | *What was new in 2.2* |
| 2.2 Update 1 | **ILPARAMS** | New | *What was new in 2.2 Update 1* |
| 2.2 Update 1 | **INIT-BY-TYPE** | New | *What was new in 2.2 Update 1* |
| 2.3 | **DBSPACE** | New parameter | *What was new in 2.3* |
| 2.3 | **EOF-1A** | New | *What was new in 2.3* |
| 2.3 | **JVMDECIMAL** | New | *What was new in 2.3* |
| 2.3 | **NLS-CURRENCY-LENGTH** | New | *What was new in 2.3* |
| 2.3 | **NULL-ESCAPE** | New | *What was new in 2.3* |
| 2.3 Update 1 | **IDXFORMAT12** | New option | |
| 2.3 Update 1 | **ILMAIN** | New | *What was new in 2.3 Update 1* |
| 2.3 Update 1 | **OOCTRL** | New | *What was new in 2.3 Update 1* |
| 2.3 Update 2 | **EBC-COL-SEQ** | New | *What was new at 2.3 Update 2* |
| 2.3 Update 2 | **COMMAND-LINE-LINKAGE** | New | *What was new at 2.3 Update 2* |
| 2.3 Update 2 | **NATIONAL** | New | *What was new at 2.3 Update 2* |
| 3.0 | **CONVERTRET** | Deprecated | *Unsupported or Deprecated at 3.0* |

| At Product Version | Name | Change | See |
|---|---|---|---|
| 3.0 | **HOSTSIGNS** | Unsupported | *Unsupported or Deprecated at 3.0* |
| 3.0 | **IDYSRCPATH** | Deprecated | *Unsupported or Deprecated at 3.0* |
| 3.0 | **ILOBJECTIFY** | Deprecated | *Unsupported or Deprecated at 3.0* |
| 3.0 | **OPTIMIZECURSORS** | New | *Significant changes in 3.0* |
| 3.0 | **OPTION** | Deprecated | *Unsupported or Deprecated at 3.0* |
| 3.0 | **TRICKLE** | Deprecated | *Unsupported or Deprecated at 3.0* |
| 4.0 | **CHECKDIV** | New parameter | *What was new in 4.0* |
| 4.0 | **DISPSIGN** | New | *What was new in 4.0* |
| 4.0 | **GNTLITLINKSTD** | New | *What was new in 4.0* |
| 4.0 | **ILSMARTTRIM** | New | *What was new in 4.0* |
| 4.0 | **MAINFRAME-FLOATING-POINT** | New | *What was new in 4.0* |
| 4.0 | **NUMPROC** | New parameter | *What was new in 4.0* |
| 4.0 | **OOCTRL** | New parameter | *What was new in 4.0* |
| 5.0 | **ILNORMALIZENAMES** | New | *What was new in 5.0* |
| 5.0 | **JVMTARGET** | New | *What was new in 5.0* |
| 5.0 | **ANIM** | Changed default | *Significant changes in 5.0* |

**Table 6: Changes to Run-time Tunables**

| At Product Version | Name | Change | See |
|---|---|---|---|
| 2.0 | **subsystem_cancel_mode** | New | *What was new in 2.0 - Run-time Tunables* |
| 2.2 Update 2 | **reduce_java_signals** | New | *Significant changes in 2.2 Update 2* |
| 2.3 | **default_cancel_mode** | New default value<br>New parameter | *What was new in 2.3 - Tunables* |
| 2.3 | **putenv_interface** | New | *What was new in 2.3 - Tunables* |
| 2.3 | **strict_file_locking** | New | *What was new in 2.3 - Tunables* |
| 2.3 | **subsystem_cancel_mode** | New parameter | *What was new in 2.3 - Tunables* |
| 2.3 Update 2 | **command_line_linkage** | Deprecated | *Unsupported or Deprecated at 2.3 Update 2* |

**Table 7: Changes to Library Routines**

| At Product Version | Name | Change | See |
|---|---|---|---|
| 2.0 | **CBL_CTF_COMP_PROPERTY_GET** | New | *What was new in 2.0* |
| 2.0 | **CBL_CTF_TRACE** | New | *What was new in 2.0* |

| At Product Version | Name | Change | See |
|---|---|---|---|
| 2.0 | **CBL_CTF_TRACER_LEVEL_GET** | New | *What was new in 2.0* |
| 2.0 | **CBL_CTF_TRACER_GET** | New | *What was new in 2.0* |
| 2.0 | **CBL_CTF_LEVEL** | New | *What was new in 2.0* |
| 2.0 | **CBL_SEMAPHORE_ACQUIRE** | New parameter | *What was new in 2.0* |
| 2.3 | **CBL_LOCATE_FILE** | New parameter | *What was new in 2.3* |
| 2.3 | **CBL_MANAGED_SESSION_GET_USERDATA** | New | *What was new in 2.3* |
| 2.3 | **CBL_MANAGED_SESSION_SET_USERDATA** | New | *What was new in 2.3* |
| 2.3 Update 1 | **CBL_CODESET_SET_MAPPING** | New | *What was new in 2.3 Update 1* |
| 2.3 Update 1 | **CBL_RUNTIME_ERROR** | New | *What was new in 2.3 Update 1* |
| 4.0 | **CBL_GET_DEBUG_START** | New | *What was new in 4,0* |
| 4.0 | **CBL_GET_DEBUG_STOP** | New | *What was new in 4,0* |
| 4.0 | **CBL_GET_OS_INFO** | New | *What was new in 4,0* |
| 5.0 | **CBL_CREATE_CORE** | New | *What was new in 5.0* |

**Table 8: Changes to Environment Variables**

| At Product Version | Name | Change | See |
|---|---|---|---|
| 2.2 Update 2 | **MFALLOC_PCFILE** | Default value changes | *Significant changes in 2.2 Update 2* |
| 3.0 | **FSWRKDIR** | New | *What was new in 3.0* |
| 3.0 | **FSCHKLFH** | New | *What was new in 3.0* |
| 3.0 | **HCOBND** | Deprecated | *Unsupported or Deprecated at 3.0* |
| 4.0 | **ES_CAT_RETRY_COUNT** | New | *Configuring Multiple Catalogs* |
| 4.0 | **ES_CAT_RETRY_WAIT** | New | *Configuring Multiple Catalogs* |
| 5.0 | **COB_LIVE_RECORD_SIZE** | New | *What was new in 5.0* |

# What was New

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

## New Features in 5.0

This release provides enhancements in the following areas:

- *COBOL Formatting*
- *Code Analysis*
- *Enterprise Server Common Web Administration*
- *COBOL Language Enhancements*
- *Compiler Directives*
- *Data File Tools*
- *Debugging*

- *Enterprise Server Security*
- *Micro Focus Unit Testing Framework*
- *Platform Support*
- *Product Documentation*

## COBOL Formatting

*Back to Top*

You can now reformat your COBOL code in the editor according to your preference and standards. Formatting sets the indentation based on the configuration options which you can set in .

You can find the COBOL formatting features on the menu.

## Code Analysis

*Back to Top*

You can now create the following views for your code:

- Program Flow Graph - enables you to view the overall structure of the COBOL program with nodes representing sections and paragraphs. PERFORM statements between them which affect the program flow are represented by links joining the nodes. External calls to other programs are shown but cannot be expanded in the Program Flow Graph view. Hovering over the nodes shows a tooltip containing the code representing the respective sections or paragraph. Clicking a node or a link joining said nodes positions the code editor to that point in the source, allowing quick and easy navigation of the program.
- Data Flow Analysis Tree - enables you to select a data item and then either statically trace what might change its value or what other data items it might affect.

## Enterprise Server and Scale Out Architecture

*Back to Top*

**Note:** This feature is in Early Adopter Product (EAP) release status.

This release offers an enhanced and improved availability, and serviceability of Enterprise Server through the ability to administer and maintain a cross-system group of regions as a single system image with the new Performance and Availability Cluster (PAC).

In order for different Enterprise Server instances to be able to work together, they need to be able to share data. This is achieved through the use of a Scale Out Repository (SOR). All ES instances in a PAC will have a common SOR (PSOR) which is used to store CICS resources (limited to PCTs, PPTs, FCTs, DCTs and TSTs in this release) as well as internal system data to facilitate synchronisation between instances. Temporary Storage Queues and Transient Data Queues can also be shared between Enterprise Server instances by directing them to a SOR. Benefits of using PACs and SORs in this way include:

- Reduced hardware costs - taking advantage of the PAC to scale-out rather than scaling-up for more efficient use of processor resources.
- Easier maintenance - Dynamically adding or removing of regions to the PAC for system maintenance.
- Increased availability - in the event of the failure of an Enterprise Server instance, the PAC can continue to operate with reduced capacity. Enterprise Server instances can reside on different machines, improving availability further.
- Dynamic scaling - Enterprise Server instances can be added to, or removed from, the PAC depending on demand.
- Better performance - throughput is no longer restricted by the resources on a single machine (scale-up).
- Improved serviceability - you can now administer the PAC and any associated regions as a single image from a new contemporary web-based administration interface (ESCWA).

## Enterprise Server Common Web Administration

*Back to Top*

**Note:** This feature is in Early Adopter Product (EAP) release status.

Visual COBOL now includes a new Enterprise Server Common Web Administration interface (ESCWA). ESCWA is a web user interface and server for modern administration, monitoring and control of Enterprise Server. It offers improved usability that consolidates the different Enterprise Server user interfaces so that native and managed regions, and security stores can be managed in one place. Features include:

- Administering directory servers across multiple hosts.
- Monitoring and control of Enterprise Server instances.
- Configuring and administering a security store, defined in an LDAP-compatible security manager such as Microsoft Active Directory or OpenLDAP.
- Administering the Scale-Out features - enable you to specify logical groups of Enterprise Server instances, and configure and run Performance Availability Clusters (PACs) and their related Scale-Out Repositories (SORs).
- Administering, monitoring and control of Enterprise Server for .NET regions and listeners.
- The use of current web frameworks that have a greater focus on security.

## COBOL Language Enhancements

*Back to Top*

The following enhancements have been made to the COBOL syntax:

- ISO2002 conditional compilation - an additional mechanism for conditional compilation, the ISO2002 Conditional Compilation method, is now available. This has been implemented as part of the support for Enterprise COBOL 6.2.

  The following compiler directing statements are supported:

  ```
  >>DEFINE
  ```

  ```
  >>IF/>>ELSE/>>END-IF
  ```

  ```
  >>EVALUATE/>>WHEN/>>WHEN OTHER/>>END-EVALUATE
  ```
- The JSON PARSE statement enables you to convert JSON text into COBOL data formats. This has been implemented as part of the support for Enterprise COBOL 6.2.
- INSPECT statement - the performance of INSPECT... CONVERTING has been improved, particularly for cases where `source-alphabet` is long.
- Support has been added for the portable syntax for SET - set[string].

## Compiler Directives

*Back to Top*

The following Compiler directives are new in this release:

**JVMTARGET** Specifies the JVM version number targeted by the class files generated in this compilation unit.

**ILNORMALIZENAMES** Determines the program elements that are normalized when a program is compiled to .NET or JVM COBOL. Normalization in this context results in replacing all $ and - (hyphen) characters with _ (underscore) characters in the name of the generated class file.

## Data File Tools

*Back to Top*

This release provides the following enhancements:

- The Data File Editor now includes a Compare Files tool that enables you to compare the contents of two data files side-by-side.
- Structure files, and the layouts within them, can now be created within the Data File Editor; you no longer need to use the Classic Data File Tools utility to manage your layouts.
- When connecting to a VSAM dataset stored in an enterprise server region, you can store any passwords required for access, for the duration of your current session.
- You can view archived JES spool jobs that have been merged into one spool file using the merging archived spool files process.
- You can now quickly duplicate records in non-indexed files, using the Duplicate Record option.

## Debugging

*Back to Top*

This release includes the following enhancements:

- Conditional watchpoints - you can now specify conditions and hit counts for COBOL watchpoints. A condition can be in the form of an expression or it can be a hit count (for the number of times memory associated with a data item has been updated). The condition is evaluated each time the data item being watched changes. If the condition evaluates to true, then program execution stops. This enables more fine-tuning of the current watchpoint support and is useful when debugging large, complex programs.
- Dynamic core dumps - you can now invoke a core dump programmatically and continue execution of your program. A new library routine, CBL_CREATE_CORE, enables you to produce a core file for the current process or a process owned by the same user that owns the current process.
- Improved visibility of the PERFORM stack - the PERFORM stack and PERFORM range names are visible in system debuggers such as gdb in many cases on 32-bit Linux with Intel architecture and Solaris with Intel architecture. This was already the case on AIX, 64-bit Solaris-Intel, and all other Linux platforms (including 64-bit Linux-Intel).
- Live Recording - a new environment variable, COB_LIVE_RECORD_SIZE, is now available to aid performance during the creation of a live recording file. Use it to specify the amount of memory (in MB) that is to be used to store events that are to be written to the live recording file. When the limit is reached, the buffer is cycled so that the oldest events are removed to make space for the most recent events.
- In order to include source line numbers in run-time error messages, which makes the diagnosis of issues in your program easier, the COBOL compiler now produces `.idy` files by default when building from the command line, unless the NOANIM directive is specified. You can safely remove `.idy` files after compilation if you do not need them; they do not need to be distributed with your application unless you want to debug your code. However, unless you specify the ANIM directive, generated executables such as `.dll` and `.gnt` will not be debuggable, which is consistent with prior behavior.

  **Note:** Behavior has not changed if you are using an IDE or build scripts generated by an IDE.
- Stackdump files (Linux Intel 64-bit, Solaris Intel 64-bit, and Linux/390 platforms only) - you can produce a stack trace for applications in a production environment that encounter a Run-Time system error. Use the stackdump_on_error and stackdump_filename tunables to configure the production of a stackdump file when an error occurs, from which you can locate the exact line of failing code in your program.
- A new configurable tool is now available that dumps to a file stack traces of all threads in process on any Run-Time System error. [Linux Intel, Solaris Intel, zLinux only – requires ANIM on 32-bit Linux Intel]. This is only supported for native COBOL.
- Line number information is now being output for optimized builds (on Linux Intel 64-bit, Solaris Intel 64-bit, PowerLinux, zLinux, and Solaris SPARC platforms only). This is only supported for native COBOL. Because of what optimization does, the information is imperfect, but it remains highly indicative. This gives the following advantages:

  - Enables the use of the stackdump utility (where available) for optimised builds.
  - Provides location information should a signal occur when running in a system debugger such as gdb.
  - Provides some limited ability to step through optimized code in system debuggers.

## Enterprise Server Security

*Back to Top*

This release includes the following enhancements:

- Security - security features can now be employed when developers and administrators install new COBOL services (web services and EJBs) into an enterprise server instance over the network. There are a number of authentication and authorization options that can be enabled. See *Deployment Listeners* and *The .mfdeploy File*.
- Vault Facility - a new security feature has been added that enables some Enterprise Server components to keep certain sensitive information in a form of storage defined as a vault, accessible via a configurable vault provider. The default vault provider stores data in encrypted format on disk.
- OpenSSL 1.1.1 - the OpenSSL security provider has been updated to OpenSSL version 1.1.1.

  This is the stable Long Term Support version of OpenSSL.

  - Added support for the ratified TLS protocol version 1.3. TLS 1.3 benefits include:

    - Much shorter initial connection negotiation sequence. This reduces the time taken to establish a link before starting to transmit data.
    - Using only the most secure ciphers and hash methods.
    - TLS 1.3 will be negotiated in preference to the older TLS protocols.
  - Added support for new Ciphers and Key Exchange groups in line with TLS 1.3 requirements.
  - The default security level for previously configured endpoints has been moved from Security Level 0 to Security Level 1. This removes the ability to accidentally make use of known-weak elements such as SSL3 and MD5. Similar changes to the default Security Level have recently happened to Java, Chrome, Firefox, and other systems providing secure connections.
- Fileshare Security - the Fileshare Secure TCP/IP transport provider now supports the trusted use of X509 certificates bearing the name of the Fileshare service as the Common Name element of the certificate.

  In previous releases, a secure connection to a Fileshare server was made using a certificate that represented the network location upon which the Fileshare service was located. This method is still supported, but does not distinguish between the exact Fileshare server that is being connected to when more than one service can exist on a single host system. With this change, individual Fileshare services can identify themselves by using a unique certificate. While running on the same host and registered with the same network endpoint.
- Support for Active Directory user groups and group name mapping - the Enterprise Server External Security Facility's MLDAP ESM Module can now use Active Directory user group objects for Enterprise Server user groups. Also, the module can now map long group names to the 8-character-maximum names required for mainframe emulation.
- Selective auditing - administrators can audit only security activity of particular interest, reducing audit overhead and the volume of events. The Enterprise Server External Security Facility's MLDAP ESM Module can now enable ESF Audit events only for particular users, groups, and resources.
- Improved interaction with LDAP client libraries resulting in fewer LDAP-related issues and easier diagnostics - the Enterprise Server External Security Facility's MLDAP ESM Module has improved interoperation with LDAP client libraries:

  - The client library vendor and version information is logged after the library is loaded
  - The module has better heuristics for loading the correct library supplied by the OS vendor, so the "provider" configuration option can generally be omitted
  - For OpenLDAP, the module sets its proprietary "connect timeout" option

## Enterprise Server for .NET

*Back to Top*

This release includes the following enhancements:

- Improved CICS compatibility - CICS EIBRCODE handling in Enterprise Server for .NET now matches the mainframe specification. CICS BMS support in Enterprise Server for .NET now matches the behavior on the mainframe and in native Enterprise Server more closely.
- MMC administration UI support for ESF security - the MMC snap-in for Enterprise Server for .NET administration now supports the External Security Facility. This means you can use LDAP-based or custom security solutions to provide administrator authentication and access control.
- TLS (SSL) support - the seelistener utility now supports TLS (also known as SSL) for its TCP communications channels. These include all channel types except MQ. The Enterprise Server for .NET client classes also have been enabled for TLS. For example, the seeout utility can now be used with a TLS-enabled MFBINP channel.
- Performance improvements - this release comes with an improved performance in WCF, system database operations, and VSAM database operations.

## Micro Focus Unit Testing Framework

*Back to Top*

This release provides support for the following functionality:

- Data-driven unit tests - a new type of test suited to testing data where values read from a source are passed through the same tests.
- Test output colorization - test output can now include basic colorization for supported terminal emulations. This feature must be run from the command line.

## Platform Support

*Back to Top*

Support is now available for the following additional or updated operating systems:

- SUSE (Power) 11 SP3, 12
- Red Hat (Power) 7.2, 7.3
- Windows Server 2019

For a full list of supported platforms, check *https://supportline.microfocus.com/prodavail.aspx*.

## Product Documentation

*Back to Top*

The following is new in this release:

- The *Comparison of COBOL with Other Managed Languages* document has been updated to reflect the recent enhancements in the .NET and JVM COBOL support.

# What was New in Visual COBOL 4.0

This release provides enhancements in the following areas:

- *Application Server JCA Support for Enterprise Server*
- *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*

### 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.

For more information see *Related Information* at the end of this topic.

### Code coverage

*Back to Top*

This release provides the following enhancements:

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

# What was New in Visual COBOL 3.0

Visual COBOL 3.0 provided enhancements in the following areas:

- *Application Server JCA support for Enterprise Server*
- *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*
- *Deployment on multiple platforms*
- *Documentation*
- *Enterprise Server*
- *Micro Focus Unit Testing Framework*
- *XML processing*

### 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.

### 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 - .

### 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. See *To install a CCSID Table* for details on installing the required CCSID table.

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. See *cast expressions* in the *Expressions* topic for more information.

### 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.

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".

### 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* 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:

- 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.

### 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 for each element of a test case.
- Support for running tests with Code Coverage enabled.
- The Micro Focus Unit Testing, 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:

- *COBOL language enhancements*
- *Classic Data File Tools*
- *Compiler directives*
- *File handling*
- *Library routines*
- *Tutorials*

### 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.

### File handling

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

For more information see *Related Information* at the end of this topic.

### 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.

For more information see *Related Information* at the end of this topic.

### 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:

- *Application Server JCA support for Enterprise Server*
- *Code Analysis*
- *Code Coverage*
- *Compiler directives*
- *Database Access - Mainframe Batch Database Tools (MBDT)*

- *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*

### Code analysis

*Back to Top*

### Code coverage

*Back to Top*

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

- A new showing the statistics of what percentage of the code has executed.
- Navigation from the 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

### Database Access - Mainframe Batch Database Tools (MBDT)

*Back to Top*

**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.

This release provides MBDT support for DB2 LUW partitioned tables.

### 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, `mfjarprogmap`, 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.

# What was New in Visual COBOL 2.3

Visual COBOL 2.3 provided enhancements in the following areas:

- *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*
- *Preprocessors*
- *Profiler*
- *REST service interfaces*
- *RM/COBOL Compatibility*
- *Single file support*
- *Tunables*
- *Updated run-time system*

### 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 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.

### 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 tab.
- 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. See the *BINDDIR* DB2 compiler directive option 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 *OPTPER* SQL compiler directive option topic for details.

**PL/I**

This version provides 64-bit support for PL/I on appropriate platforms. See *Additional Software Requirements* 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**

- 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.

### Preprocessors

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

A new page, , 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.
2. Compile your application to apply the changes.
3. Run your application with Profiler to produce the relevant reports.

### 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:

- *Character Set Enhancements*
- *Code Analysis*
- *Database Access*
- *Micro Focus COBOL enhancements*

- *External Security Facility (ESF)*
- *Enterprise Server MQ-IMS Bridge*
- *Tunables*

### 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

### Database Access

The following new features are available in database access support:

**COBSQL**

**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*
- *COBOL Source Information*
- *Compare and Synchronization Monitor*
- *Compiler Directives*
- *Database Access*
- *Enterprise Server Integration in the IDE*
- *Enterprise Server MQ-IMS Bridge (Technology Preview)*
- *Environment Variables*
- *Fileshare Recovery*
- *Line Numbering for COBOL Programs*

- *Managed COBOL*
- *Managed COBOL Syntax*
- *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.

### COBOL Source Information

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

### Compare and Synchronization Monitor

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.

### 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 .

### 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.

### 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. |

### 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*
- 
- *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 and CSI returns the results of the query in the .

### 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.

**ILEXPONENTIATION** 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.

### 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 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*

### 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.

# 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*
- *Managed code enhancements*
- *OpenESQL*
- *UNIX Platforms Support*

### 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.

### 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 following new SQL compiler directive options:

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

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

**DATEDELIM** Specifies a single character as the delimiter between the year, month, and day components to override the default delimiter determined by the HCOSS DIALECT or DATE directive specification.

**TIMEDELIM** Specifies a single character as the delimiter between the hour, minute, and second components to override the default delimiter determined by the HCOSS DIALECT or TIME directive specification.

**TSTAMPSEP** Specifies a single character as the separator between the date and time parts of timestamp and date/time data.

**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.

# What was New in Visual COBOL 2.0

Visual COBOL 2.0 provided enhancements in the following areas:

- *Compiler Directives*
- *JVM COBOL File Handler*
- *Library Routines*
- *Managed COBOL Language Features*
- *Run-Time Tunables*
- *Vision Data File Searching*

## Compiler Directives

The following new directives are now available:

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

The following directives have been changed:

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

## JVM COBOL File Handler

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

## Library Routines

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

CBL_CTF_COMP_PROPERTY_GET
CBL_CTF_TRACE
CBL_CTF_TRACER_LEVEL_GET
CBL_CTF_TRACER_GET
CBL_CTF_LEVEL

The following routine has been enhanced:

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

### Managed COBOL Language Features

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

**Local Variables** In managed COBOL, Data items can now be declared in the procedure division, using the DECLARE statement. In addition, they can be declared inline as the iterator in a PERFORM statement, or as an exception message in a TRY … CATCH … FINALLY statement block.

**Collections** There are two new collection types in managed COBOL: LIST and DICTIONARY. For a LIST, you can add elements to a list, retrieve the nth element of the list, replace the nth element, iterate through the list and clear the list. For a DICTIONARY, you can add key value pairs, retrieve a value corresponding to a key, to replace the value corresponding to a key, iterate through the dictionary and clear the dictionary.

**Properties** In managed COBOL, a property can now be defined using PROPERTY-ID and GETTER and SETTER phrases to access to the property. The previous technique of specifying the keyword PROPERTY on a data declaration is still available.

**Indexers** In managed COBOL, an indexer can now be defined using INDEXER-ID and GETTER and SETTER phrases to access the indexer value. Indexers are similar to properties, except that their accessors take parameters. Indexers allow instances of a class or valuetype to be indexed just like arrays.

**Zero-based Indexing** The managed COBOL syntax for arrays now uses zero-base indexing to access arrays when square brackets are specified. For backward compatibility, one-base indexing is used when round parentheses are specified.

### Run-Time Tunables

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.

### 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$CALLEDBY
  - C$CALLERR
  - C$CHDIR
  - C$MAKEDIR
  - C$MEMCPY
  - C$MYFILE
  - C$PARAMSIZE
  - C$RERR
  - M$ALLOC

    - M$FREE
    - M$COPY
    - M$FILL
    - M$GET
    - M$PUT
    - WIN$VERSION
- The following ACUCOBOL-GT 'ccbl' compiler options can now be used with Visual COBOL:
    - -E, -V
    - -Cv
    - -Da, -Db, -Dd31, -DL1/2/4/8, -Dq, -FpRounding
    - -La, -Li, -Lc, -Lf, -Ll, -Lo, -Ls, -Lw

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

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

### Embedded HTML

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

### Language Improvements

The following improvements have been made to managed COBOL:

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

### RM/COBOL Compatibility

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

- The following RM/COBOL standard library routines can now be used with Visual COBOL in native code:
    - C$Century
    - C$ConvertAnsiToOem
    - C$ConvertOemToAnsi
    - C$DARG
    - C$Delay
    - C$GetEnv

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

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

### XML Extensions

**Note:** This functionality is supported in native COBOL only.

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

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

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

# 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.

## Significant Changes in Visual COBOL 5.0

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).

- *Common Communications Interface*
- *Compiler*
- *Data Tools*
- *Documentation*
- *Enterprise Server*
- *Enterprise Server Auditing*
- *File Handling*
- *Interface Mapping Toolkit*

- *Micro Focus Directory Server*
- *MFCRYPTLIB*
- *Request Packet Interface API*

### Common Communications Interface

*Back to the list*

- Certificate subject names are now correctly checked against the name the client is attempting to connect to. Certificates registered to "localhost" should now work as expected.
- The default Cipher Suites List now includes @SECLEVEL=0. This indicates that backwards compatibility with old certificates such as MD5 is to be retained. As a result of this change, you need to review your use of the Cipher Suites List and of any old certificates.
- If using Micro Focus Common Client(MFCC) or Micro Focus Communications Server(MFCS) to connect to a non-existent remote host, it was possible to create a connection to a server listening on the same port on the local system. This is no longer possible and a bad address error is reported in the error logs.
- By default, in this and previous releases an SSL/TLS CCI client can trust a server which resides on an IP address that is not an intended connection target. You can now specify the ssl_only_verify_literally option in your CCI.INI file to restrict the certificate check to improve security. See “CCITCP Security Options in the CCI.INI File” in your product Help for more information. When connecting to a non-CCI protocol server, all clients perform strict certificate checking as the new default behavior

### Compiler

*Back to the list*

- The Search order has been updated, to allow the Compiler to find Java.

### Data Tools

*Back to the list*

- Accessing a dataset through the Data File Editor now obtains a shared lock on Enterprise Server when you open the file in shared mode.

  3170479 (1115346)

### Documentation

*Back to the list*

- The ANIM Compiler directive has changed its default when compiling from the command line on Windows platforms. The default is now ANIM. (Compiling from the IDE still defaults to ANIM.)
- The validation of server certificates for TLS (also known as SSL) connections has been corrected. This is likely to cause connection failures in cases where previously connections were incorrectly allowed. For example, if Fileshare is secured with TLS, client applications must be configured to use the hostname of the Fileshare server as it appears in the server's certificate. If the Micro Focus Directory Server is secured with TLS, then MFDS clients, such as the casstart command-line utility, will need to connect to it using a hostname that appears in the certificate MFDS is using. (Note that a certificate can contain multiple hostnames.) Consult your Certification Authority administrator for more information.

  3192497 (1117068)

### Enterprise Server

*Back to the list*

- A number of audit events now contain some additional information.

  3140310 (1112666)

- Previously, the /x option of the CASRDTEX utility had been duplicated to drive XML output. This prevented the export of transactions which was the original meaning of /x. To generate an XML export, you must now specify the option /xm.

  3180233 (1116108)
- casfhsf could hang when processing HSF files that contain records from tasks that ran in different months.

  3177385 (1115884)
- A new system transaction, CRCN, is now available to monitor the state of XA connections for each resource manager (RM) entry defined in the system. CRCN is a replacement for the former XA Reconnect facility that recycled SEPs on connection loss. The XA Reconnect facility was configured by setting the ES_XA_RECONNECT environment variable. CRCN also uses this environment variable, although its usage differs slightly. For more information see your product Help.

  3174272 (1115727)

### Enterprise Server Auditing

*Back to the list*

- Auditing on big-endian platforms now correctly interprets numeric audit data.

### File Handling

*Back to the list*

- RELFORMAT=MF|RM is now enabled for FOLDER and other filename tags.

  3164916 (1114803)
- The SORT-CONTROL special register is now supported during a SORT statement.

  3159740 (1114389)
- To direct the handling of relative files to a particular file handler (either MF COBOL or RM/COBOL), use the RELFORMAT=MF or RELFORMAT=RM configuration options within the [REL-DEFAULT] tag of the EXTFH.CFG file. (The default is RELFORMAT=MF.)

  3157247 (1114149)

### Interface Mapping Toolkit

*Back to the list*

- If you try to deploy a stateful EJB SVI from the Visual Studio IDE after unchecking the “Reuse container after application termination” check box (in the Deployment Characteristics tab of the Runtime Environment Configuration dialog box), you receive a warning message informing you that the “Reuse container after application termination” setting will be set to true for the deployment in order to prevent the crash. In the Eclipse IDE, the unchecking of this option will just be ignored, and the option re-selected. A warning to not uncheck the option for stateful EJB SVIs is also displayed on the prior page (to the dialog box).

  2663812 (1090987)

- WSDL and JSON Schema files generated for bottom-up CICS Web services now include a "maxLength" parameter for all string fields.

  3175039 (1116241)

### Micro Focus Directory Server

*Back to the list*

- The event codes for several MFDS audit points have changed.

### MFCRYPTLIB

*Back to the list*

- Passing an incorrect keyfile passphrase no longer results in receiving an RTS115 error in MFDS.

  3181743 (1116186)

### Request Packet Interface API

*Back to the list*

- The default behaviour for TLS certificate Common Name (CN) matching has changed for release 5.0. It now requires a strict match on the CN value or any Subject Alternate Name (SAN). For example, 127.0.0.1 and localhost are no longer implicitly equivalent, nor are hostname and its IP address or fully-qualified domain name (FQDN) variants. When starting a region from the Enterprise Server Administration Web interface, a resolved IP address is used for the casstart -m parameter value. A resolved IP address is also used even if a hostname is specified on a command line startup.

  To change this behavior and use a resolved hostname value (as determined by the machine TCP configuration, for example, hosts file entries.) set the environment variable MFDS_DNS_RESOLVE=Y. To use an unresolved string literal hostname, for example, a value explicitly passed in via the casstart –m command line parameter, set MFDS_DNS_RESOLVE=N.

  3194613 (1117203)

# Significant Changes in Visual COBOL 4.0

This section describes significant changes in behavior or usage in Visual COBOL 4.0. 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).

- *Code Coverage*
- *Codeset Support*
- *Common Communications Interface*
- *Communications Server*
- *Compiler*
- *Data Tools*
- *Documentation*
- *Enterprise Server*
- *Enterprise Server Auditing*
- *File Handling*
- *MF Server Administrator (GUI)*
- *Micro Focus Directory Server*
- *Run-Time System*

### 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=MQL*,CN=MFESMAC,CN=Enterprise Server Resources,CN=Micro
  Focus,CN=Program Data,DC=X
  changetype: add
  cn: MQL*
  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 adds 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)

### 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 mfds -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

### 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*
- *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.

### 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 mfds 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 (CASMG0001I).

  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:

- *Data Tools*
- *Editor Writing Assistance*
- *Run-Time System*
- *SQL: OpenESQL*
- *SQL Option for DB2*

### 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.

### 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 `%ALLUSERSPROFILE%\Micro Focus\Enterprise Developer\hcoss \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

### 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*
- *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.

### 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 mfds 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.0Z), 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*

### 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.

## 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

### 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

# Unsupported or Deprecated Functionality

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

# Unsupported or Deprecated at Visual COBOL 5.0

This section includes information about features or functionality that are no longer supported.

- 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. This is a change since version 3.0 of this product.

# Unsupported or Deprecated at Visual COBOL 4.0

There are no features or functionality unsupported or deprecated at version 4.0.

# 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
DATALIT
EANIM
EDITOR
ENSUITE
EXPANDDATA
FIXING
FLAG-CHIP
MASM
MODEL
OPTSIZE
OPTSPEED
PARAS
PROTMODE
REGPARM
SEGCROSS
SEGSIZE
SIGNCOMPARE
SMALLDD
TABLESEGCROSS
TRICKLECHECK
WB2
WB3
WB

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

BASENAME
FILENAME
PATH
TARGETDIR

# Known Errors and Restrictions

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 5.0:

# Upgrading from Server Express to Visual COBOL Development Hub

You can upgrade existing distributed applications created with Server Express to Visual COBOL Development Hub.

For more information on how to upgrade, check the product help.

**Note:** The process of upgrading Server Express is very similar. Therefore, references to Visual COBOL for Eclipse and Visual COBOL Development Hub apply equally to Enterprise Developer for Eclipse and Enterprise Developer UNIX Components, respectively.

# 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 Jyphon) 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.