MICRO FOCUS

Cumulative Changes from Mainframe Express to Micro Focus Enterprise Developer for Visual Studio

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

2018-06-07

Contents

Cumulative Changes from Mainframe Express to Enterprise Developer for Visual Studio

Welcome to Enterprise Developer for Visual Studio. This document combines information on Enterprise Developer for Visual Studio releases from the first release to the most recent release, and on upgrading from Mainframe Express to Enterprise Developer. 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 you are migrating from Mainframe Express, to see everything that has changed in the life of Enterprise Developer for Visual Studio.
- If your Enterprise Developer installation is up to date, to see the changes made in the latest release.
- If you already use Enterprise Developer but your installation is not entirely up to date, to see the changes made over several updates, together in one place.

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 Mainframe Express to Enterprise Developer	This section gives guidance on upgrading from earlier Micro Focus products.

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

What was New

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

What was New in Enterprise Developer 4.0

This release provides enhancements in the following areas:

- *Integration with Visual Studio*
- *Application Server JCA Support for Enterprise Server*
- *Build Tools for Windows*
- *Code Coverage*
- *Codeset support*
- *Compiler directives*
- *Data File Tools*
- *Debugging*
- *Docker*
- *Documentation on working with large applications*
- *Enterprise Server*
- *Enterprise Server for .NET*
- *File Handler*
- *HCO for DB2 LUW*
- *IMS support*
- *JCL support*
- *Mainframe Access*
- *Managed COBOL*
- *Library routines*
- *Micro Focus Unit Test Framework*
- *OpenESQL*
- *Platform support*
- *PL/I support*
- *Web Service Clients*

Integration with Visual Studio

Back to Top

This release includes the following editor improvements:

- Inline rename refactoring - overwriting a variable name directly in the editor now invokes rename refactoring so you no longer need to select **Refactor** from the editor context menu.
- **Extract to section** - a new quick action (Visual Studio 2015 and later only) is available in the editor. It enables you to refactor your code by adding more sections in the code. This can help you create more entry points that you can use during unit testing.
- Automatic insertion of END-EXEC, END-TRY and END-PERFORM statements - the closing statements are now automatically inserted after you have typed the opening statements.
- IntelliSense filters (Visual Studio 2017 and later only) - when IntelliSense displays a list of suggestions, a number of icons at the bottom of the IntelliSense window now enable you to filter the suggestions.

Working with files:

- Copybook graphs - a context menu command, **Show Copybook Graph**, in Solution Explorer enables you to display the graphical representation of the copybook dependencies of COBOL programs.
- **Open Folder** mode (Visual Studio 2017 only) - support is now available for editing, compiling and debugging files opened in Solution Explorer in Open Folder mode without having to create projects.
- The file property pages now display a list of all Compiler directives that apply to the file.

New IDE configuration option:

- You can use the **Limit IntelliSense Search Scope** setting in **Tools > Options > Text Editor > Micro Focus COBOL > Advanced** that helps improve the editor performance.

Relinking existing applications:

- You can now configure Enterprise Developer to check whether applications created with older releases must be relinked. If the application uses an older version of the C runtime, Enterprise Developer can automatically relink the existing executable or .dll to the new version of the C runtime without the need to recompile the application first.

  If a project needs relinking, Visual Studio displays a message in the status bar (Visual Studio 2015 and 2017) or a dialog box (Visual Studio 2013) providing an option for you to choose and relink the project.

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.

Build Tools for Windows

Back to Top

This release includes Enterprise Developer Build Tools for Windows, a separately-installable component of Enterprise Developer that has been designed to be used in environments where you want to work with your COBOL and PL/I projects but you don't want the overheads associated with the Visual Studio IDE.

Build Tools provide a lightweight, easy-to-install development environment that is well-suited for use in Docker containers and continuous integration or continuous delivery systems.

Code coverage

Back to Top

This release provides the following enhancements:

- In Visual Studio, the Micro Focus Code Coverage window now offers a File View in addition to the existing Program View.

Codeset support

Back to Top

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

Compiler directives

Back to Top

The following Compiler directives are new in this release:

- **DISPSIGN** - determines the display output of numeric fields with included signs, under an IBM mainframe dialect only.
- **GNTLITLINKSTD** - stops the suppression of call-convention 8 when both call-convention 2 and call-convention 8 are in effect for a `.gnt` file in an Intel x86 32-bit environment.
- **ILSMARTANNOTATE** - adds attributes to the items generated by ILSMARTLINKAGE, based on their data type, which can be used to identify the size or range of the item in COBOL.

- **ILSMARTTRIM** - trims any trailing spaces from a string item returned by the get property associated with an alphanumeric item processed by ILSMARTLINKAGE.
- **MAINFRAME-FLOATING-POINT** - specifies the format of a program's floating point data items: either IBM hexadecimal format or IEEE format. This directive is supported in managed code only.

The following Compiler directives contain new parameters in this release:

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

Data File Tools

Back to Top

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

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

Debugging

Back to Top

This release includes the following enhancements:

- Enhanced .NET debugging in Visual Studio 2017 - a number of advanced debug features are now available when debugging .NET COBOL code. Features include support for performance tips, backwards debugging, an expression evaluator, integration with the Diagnostic Tools window, remote debugging of .NET COBOL applications, as well as support for IntelliSense in the debug windows (such as the Immediate, Watch, and QuickWatch windows).

- Debugging CICS channels and containers - you can now debug CICS COBOL applications that use channels and containers. The debugger now enables you to view the contents of any channels and containers, and to change content as you debug.

Docker

Back to Top

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

Documentation on working with large applications

Back to Top

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

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.

Enterprise Server for .NET

Back to Top

This release provides the following enhancements:

- Limited support has been added for TLS (SSL) - you can now configure TN3270 and MFBINP listener channels for TLS (SSL). The seeout command-line utility and the JES public class API also now support TLS.
- Enterprise Developer for Visual Studio includes a new wizard that enables you to convert native MSS projects to managed MSS ones. The wizard is invoked using the **Create Managed Mainframe Project** context-menu command at project level in Solution Explorer.

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.

HCO for DB2 LUW

Back to Top

This release includes the following new features:

- Support for CHARSET EBCDIC COBOL stored procedures
- Seamless integration with DB2 LUW SBCS databases, providing various EBCDIC collation sequences
- 64-bit HCO command-line tools are now available across all supported platforms
- HCO DCLGEN enhancements that provide more copybook generation options

IMS support

Back to Top

This release introduces the following new features:

- The IMS Database Forward Recovery feature has been updated to provide point-in-time forward recovery that enables you to recover all database transactions completed up to a specified point in time, including date and time of day.
- IMS support now includes MFS code for Light Pen devices. In your MFS, include a DEV statement with the FEAT=PEN parameter, and set the ATTR property on appropriate fields to DET or IDET to make them detectable by a light pen. See your IBM documentation for details.

  Note: MFS that contains light pen code must be regenerated to take effect if the MFS is part of an Enterprise Developer project created using an earlier release.
- Support for export and import of IMS-configured enterprise server regions through Server Explorer.

JCL support

Back to Top

This release introduces the following new features and enhancements:

- Micro Focus FTP utility - you can now configure MFFTP to not process trailing blank spaces for a GET command, by setting the new control variable MFFTP_PROCESS_TRAILS_ONGET=FALSE and setting the LOCSTAT value TRAILINGBLANKS=FALSE.
- A new API has been introduced - MVSSPLPB - that enables you to search all job information held in the JES spool control files. It is more comprehensive that the existing MVSCASPB API, and should instead be used for all new spool enquiries.
- A number of new environment variables have been introduced that enable you to configure the JES startup behavior with regards to opening user catalog files. You can configure JES to open user catalog files only when they are required (and not during system startup), and you can configure JES to retry opening a catalog file that has failed to open. These changes increase the resilience of the JES system, as you can mitigate against JES startup failures if the user catalog files are not readily available whilst the region starts. See *Configuring Multiple Catalogs* for more information.
- The public catalog API, MVSCATPB, has been enhanced to better report on file errors when opening catalog files. The following new functions have been added to `mfpubcat.cpy`: GCFS, GNFS, STNI, GCFE, and GNFE.

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.

Mainframe Access

Back to Top

The installation process has been changed as follows:

- A single installer for both Mainframe Access and the Mainframe Access z/Server support is provided.
- There is now only one XML configuration file for Mainframe Access. There is also a new configuration utility that guides you through the configuration process.
- Specify your own JES console character - it is now possible to configure the Mainframe Access installation to use a different JES console character instead of the default ("$"). To achieve this, use the new JES_CONCHAR configuration parameter.

Enhancements are available in the following areas:

- The Mainframe Call Generator now runs in an independent address space. Previously, it was possible for code run in the MFA address space to enter a supervisor state. MCGLIB is now the new address space's STEPLIB. This enables other user modules to be called without placing them in MFA's STEPLIB, as was previously necessary. See *Mainframe Call Generator: Calling Programs on the Mainframe* for more information.

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.
- Support has been added to Visual Studio for unit testing of managed procedural projects.

OpenESQL

Back to Top

This release provides the following new features:

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

Platform support

Back to Top

Note the following changes in platform support for this release:

- Windows 8 and Windows Server 2012 are no longer supported for developing applications. They are still supported for deployment.

PL/I support

Back to Top

Note: PL/I applications that were created using Enterprise Developer 3.0 and earlier must be recompiled in Enterprise Developer 4.0 to ensure they will compile and execute properly.

Support has been added to Eclipse for the following features in PL/I debugging:

- Displaying complex data structures in the Expressions and Variables views.
- Viewing the console window on non-Windows platforms so that output and input can be seen or entered.
- The Outline view in the default Debug perspective.
- Watchpoints on variables on the line where the variable has been declared.
- Conditional breakpoints.
- Skipping variables a defined number of times before the execution stops.
- Adding expressions to the Expressions view from the Outline view.
- Inspecting variables when referenced as part of a macro argument.
- Signals and conditions.

PL/I compiler and Run-Time System enhancements:

- Support for the following built-in functions - CHARVAL(), COMPARE(), CURRENTSIZE(), EDIT(), PICSPEC(), SOURCEFILE() and SOURCELINE().
- Support for additional DATETIME() format strings.
- Support for the FLUSH FILE(*), ITERATE, and RESIGNAL statement.
- Support for the XMLOMIT and XMOLATTR attributes on variable declarations.
- Support for the MARGINI compiler option.
- Support for the SIZE and STRINGSIZE prefix conditions.
- Improved support for the repeat factor on arrays of AREA variables.
- Improved pointer arithmetic on 64-bit platforms.
- Where necessary, MFPLX now recognizes selected *PROCESS statements and passes them to the appropriate preprocessors.

PL/I macro preprocessor:

- The Micro Focus macro preprocessor now provides a new default behavior that mimics IBM's RESCAN(ASIS) behavior. In addition, a new option, rescan_upper, has been introduced to enable you to optionally mimic IBM's RESCAN(UPPER) behavior.

Improvements in compatibility with IBM's Macro Preprocessor:

- Allows OTHER as an acceptable abbreviation for OTHERWISE on a macro SELECT statement.
- Supports the SCAN/NOSCAN option on declaration of macro variables.
- Implements the NOSCAN option on ANSWER statements.
- The %NOTE message format has been modified to match IBM's behavior.
- The SYSVERSION built-in function has been modified to always return a value of 22 bytes in length (space padded, if necessary).
- Relaxed diagnostics for scenarios where Enterprise Developer is generating a warning message and IBM does not.

(Technical Preview) Interaction with COBOL AMODE and 370 Assembler programs (Windows 32-bit only):

- A new compiler option is available which enables Open PL/I to interact with 370 Assembler, and with COBOL programs compiled with AMODE. See the product help for the current limitations to this support.

Web service clients

Back to Top

This release provides the following enhancements for Web service clients:

- Improved support for larger, more-complex WSDL files, including:
  - Configurable generation of variable arrays and repeated structures by specifying the Inline-Array-Size-Limit parameter.
  - Generation selection for service, port, and one or more operations.
- Support for SOAP version 1.2 in addition to version 1.1.

What was New in Enterprise Developer 3.0

Enterprise Developer 3.0 provided enhancements in the following areas:

- *Integration with Visual Studio*
- *Application Server JCA support for Enterprise Server*
- *Azure support*
- *Building applications*
- *CICS Web services*
- *COBOL language enhancements*
- *Code analysis*
- *Code coverage*
- *Compiler control*
- *Data File Tools*
- *Database access - HCO for DB2 LUW*
- *Database access - HCOSS*
- *Database access - MySQL*
- *Database access - OpenESQL*
- *Debugging applications*
- *Deployment on multiple platforms*
- *Documentation*
- *Enterprise COBOL 6.1 compatibility*
- *Enterprise Server*
- *Enterprise Server for .NET*
- *Interface Mapping Toolkit*
- *iFileshare*
- *IMS support*
- *Mainframe Access (MFA)*
- *Micro Focus Unit Testing Framework*
- *PL/I support*
- *Rumba terminal emulator*
- *Transaction Classes*
- *XML processing*

Integration with Visual Studio

Enterprise Developer 3.0 provided enhancements in the following areas:

- Integration with Microsoft's Visual Studio 2017, the most recent version of the world’s most popular integrated development environment (IDE) for the Windows platform, offering significant benefits for developers and businesses developing software for Windows.

- COBOL Editor:
  - Brace completion - a closing quote mark or bracket is automatically inserted when you type the opening quote mark or bracket at the end of a line.
  - Brace matching for IF and EVALUATE statements - clicking on any of the clauses of these statements highlights the opening and closing statements and all clauses. Use **CTRL +]** to move the cursor to the next clause in the statement.
  - Collapse to definitions - closes all sections, entry points and methods in an open file. Use the **Outlining > Collapse to Definitions** editor context menu command.
  - Colorization is now available for QuickInfo, completion tooltips and signature help.
  - Completion of statements - IF and EVALUATE statements are now automatically aligned and completed.
  - Indentation - smart indentation is now available to control how lines indent within IF and PERFORM statements and after continuation statements.
  - Locate definition, , on the COBOL toolbar enables you to search for data items and identifiers by specifying any string of characters that might be part of their name. This command replaces **Edit > Go To Location**.
  - Outlining - outlining is now available for 01 level group items (for expanding and collapsing the entire group) and for EXEC blocks
  - Quick actions (a feature of Visual Studio versions 2015 and later) - Enterprise Developer now supports Visual Studio's quick actions for setting the CICSECM Compiler directive, rename refactoring, and finding or creating missing copybooks.
  - Tooltips are now available for managed COBOL members.
- COBOL toolbar - the commands on the COBOL toolbar have been updated in order to optimize your editing experience. A number of commands (renumber and unnumber) have been removed from the context menu in the editor and are now only available from the toolbar.
- Directives determination - there is now a preview dialog box when the IDE scans your sources for Compiler directives. This enables you to review and approve the directives before they are set.
- Error reporting - you can now sort the errors and warnings in the **Error List** window by their COBOL error code.
- Import, export and synchronization of the IDE options - you can now export a number of the COBOL and PL/I global options as a Visual Studio .settings file. You can import the file into other instances of Enterprise Developer.

  Enterprise Developer for Visual Studio 2015 and later now supports the automatic synchronization of a number of COBOL settings across the copies of Visual Studio installed on different machines. The settings are synchronized through your MSDN account. See the MSDN for more details and for information on any restrictions to the functionality.
- **Project Details** window - provides a number of enhancements, including performance improvement. It is now possible to filter on project or solution, perform a search, and sync with Solution Explorer. Tooltips are available in the Overrides column to show the override directives.
- Net Express import wizard - provides a new option for including copybooks in a project.
- Rename refactoring - this release introduces rename refactoring for COBOL code that can help to improve the readability of COBOL elements or make their purpose clearer. You can rename elements such as variables and identifiers, section and paragraph names, classes, and methods across a program or a solution.
- Support for standalone files - various improvements have been made to how you work with files that are not part of a project:
  - When you first open a standalone file, the editor displays an alert bar with options to browse for a symbols (`.idy`) file to use to compile the standalone file or to configure the Compiler directives.
  - You can now use the file's properties window to edit the Compiler directives and to specify options for debugging.

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

Application Server JCA support for Enterprise Server

In Enterprise Developer 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.

Azure support

Enterprise Developer now supports the following versions of Microsoft Azure SDK - version 2.9.6 with Visual Studio 2013 and version 3.0 with Visual Studio 2015 and later.

Building applications

Enterprise Developer 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 - **Tools > Options > Micro Focus > Projects**. In Visual Studio, multi-processor compilation must also be enabled in the project's properties.

CICS Web Services

From Enterprise Developer 3.0, CICS Web Services was made generally available; it had previously been available (from Enterprise Developer 2.2 Update 2) as a Technology Preview feature.

From Enterprise Developer 3.0, support also includes the following additional features:

- CICS JSON Web services for request/response and RESTful JSON. CICS Web Services support has been extended to allow JSON/REST requests to be used in both requestor and provider mode.
- Use of container-based mappings

COBOL language enhancements

Enterprise Developer 3.0 includes the following enhancements to the COBOL syntax:

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

The following enhancements are available in managed COBOL:

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

Code analysis

Enterprise Developer 3.0 provided the following improvements:

- A new group of predefined rule sets for 64-bit readiness is now included in Enterprise Developer.
- 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 directives are new in this release:

- COLLECTION - provides a mechanism for code coverage to identify unexecuted programs.
- ILSTDLIB (.NET COBOL only) - helps you to ensure that your .NET COBOL code compiles with the correct version of the .NET Framework Microsoft Common Object assembly `mscorlib.dll`. When building from Visual Studio or using MSBuild to compile a COBOL project, the NOILSTDLIB directive is set, and an ILREF is generated pointing at the version of mscorlib appropriate for the target framework selected in the project's properties.

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 Enterprise Developer 2.3) as a Technology Preview item was supported at GA level from Enterprise Developer 3.0.

Enterprise Developer 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 and PL/I 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 - HCO for DB2 LUW

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

Database access - HCOSS

Enterprise Developer 3.0 provided a new Offline Schema Extract feature – a batch solution that migrates a z/OS DB2 catalog to SQL Server without direct access to z/OS DB2.

Database access - MySQL

Enterprise Developer 3.0 provided support for MySQL with ODBC.

Database access - OpenESQL

Enterprise Developer 3.0 provided the following new features:

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

Debugging applications

Enterprise Developer 3.0 provided the following new features:

- A new drop-down option, **Any** on the **Debug** tab in the project's properties of native Mainframe Subsystem application projects, enables you to debug without specifying the type of application.

Deployment on multiple platforms

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

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

Documentation

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

- *Where do I start?* - located on the launch page of the product help, this section provides the information you need in order to get started depending on which aspects of the product you need to get to grips with first.
- *Multi-Platform Deployment of JVM COBOL Applications* - includes information about how to ensure the portability of your JVM COBOL applications between Windows and UNIX or Linux platforms.
- *Configuring an SSL Client Connection to a Mainframe* - includes instructions about how to configure client applications to connect to MFA and z/Server mainframe via AT-TLS in a zOS environment.

Enterprise COBOL 6.1 compatibility

The following features have been added to support Enterprise COBOL 6.1:

- A new statement, JSON GENERATE, which enables you to convert data to JSON format.
- A new special register, JSON-CODE, and its resulting exception codes to support the new statement.
- Two further new statements, ALLOCATE and FREE, to obtain and release dynamic storage.
- A new MF level, MFLEVEL"19", which contains the following new additions to the reserved words list:
  - JSON
  - END-JSON
  - ALLOCATE
  - FREE
  - INITIALIZED (context-sensitive reserved word)

  Note: These keywords are also in effect under the ENT dialect.
- The SSRANGE Compiler directive has an additional option "3", which permits zero-length reference modified items at run time when bounds checking.
- Support for the INITIALIZE statement under the ENT dialect has been enhanced.

Enterprise Server

Improvements are available in the following areas:

Integration with Visual Studio

- Exporting an enterprise server definition from the IDE in XML format.
- Importing an enterprise server into the IDE using its definition file.
- In Visual Studio, it is now possible to configure an enterprise server to produce a core dump file directly from the IDE, from the server properties.

EXEC CICS TRANSFORM:

- Enterprise Server now supports the XMLTODATA and DATATOXML options of this API.

JCL spool

The JCL spool repository infrastructure has been redesigned to significantly improve its performance. The improvements include:

- A more disk-efficient infrastructure, reducing the space required by the spool files by up to 75% in some cases.
- Improved internal indexing, resulting in quicker generation of messages and sysout records during processing.
- Improved navigation of spool queues: job status **Complete** replaces the **Output**, **Out Hold**, and **Printed** statuses, and completed jobs with any combination of sysout files can be displayed in a single list.
- Increased range of job numbers, from five digits to seven digits.

  Important: These changes are automatically in place for new enterprise servers that you create. To enable these updates to existing enterprise servers, you must run a conversion utility before any JCL processing is permitted. Follow the **To run the spool conversion utility** process, detailed in *Significant Changes in Behavior*, for each existing JES-enabled enterprise server you have.

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.

Multi-factor sign-on:

- Multi-factor authentication enables users to sign on to Enterprise Server CICS without entering a username and password. A secure user identifier, such as a certificate, is mapped to a username and is used to generate a passtoken. It is possible to use smartcards, biometrics, or other authentication systems to identify yourself to the system. The username and passtoken can then be used to log on to CICS in an automated way.

  The supported sign-on mechanisms are Express Logon Feature (ELF) and Automated Sign-On for Mainframe (ASM). See the product Help for examples of how to configure a region to use the two mechanisms.

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.

Enterprise Server for .NET

Enterprise Developer 3.0 provided enhancements in the following areas:

- Public class library - offers enhanced functionality for applications to interact with a running JES system:
  - Dynamic dataset allocation/deallocation
  - KSDS/ESDS/RRDS file handling support
  - ENQ/DEQ support
- Web User Interface (Technology Preview) - this is a Technology Preview of a web-based UI for the administration of an Enterprise Server for .NET system. The UI enables you to administer an Enterprise Server for .NET system from any device supporting a web browser and without the need to install Enterprise Server for .NET. The following features are available:
  - Starting and stopping of regions
  - The following views - Console, Statistics, Local trace tables, Monitor, Dynamic listener control and the HSF view for CICS.
- Region-specific external file handler (EXTFH) configuration - it is now possible to specify EXTFH configuration on a per-region basis using the MMC UI region start-up file editor. Previously, you had to specify EXTFH configuration on a per-machine basis and it was global to all regions. This could have lead to inconsistency in application behavior if one or more machines in a scale-out had different configurations defined.
- HSF performance improvements - various improvements have been made to HSF to reduce the amount of database processing required. This lowers the impact on application performance when HSF is enabled for a region.
- Accepting input from the system console - system console input can now be provided from the MMC UI console view, or using the seesys command-line utility. The input is provided to JES applications using ACCEPT FROM CONSOLE and to CICS applications using EXEC CICS WRITE OPERATOR REPLY.
- JES seven-digit job numbers - you can now configure JES regions to allow up to seven-digit job numbers to be used. This enhancement is also available in the native Enterprise Server product.

The following new features were initially provided as HotFix updates of Enterprise Server for .NET 2.3 Update 2 and are now available in this release:

- Dataset name filtering by the MMC UI Catalog view - it is now possible to filter catalog name entries through the use of wildcards in the MMC UI Catalog view. This helps reduce the time to display dataset names when a catalog contains a great number of entries.
- MMC UI file viewer enhancements:
  - Options for closing all open files, closing all but the currently selected file, and for loading all of the remaining lines of a file.
  - Viewing the contents of SYSOUT files being written by active job steps.
- Improved task recovery - Enterprise Server for .NET now automatically detects when a SEP pool process dies (for example, when you stop debugging, or the machine shuts down), or is no longer responsive, and recovers any running CICS transactions or JES jobs. This provides improved system integrity by ensuring that the responsiveness and work through a running system is maintained following a SEP pool failure.

Interface Mapping Toolkit

The Interface Mapping Toolkit (IMTK) contains the following enhancements in Enterprise Developer 3.0:

- Cross Origin Resource Sharing (CORS) support for REST Web services, enabling Web service access from a CORS-supported Web browser.

iFileshare

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

- An improved failover and recovery process. iFileshare now supports full recovery of nodes in the group. For high availability (HA-VSAM) groups, servers can now rejoin the group without the entire group having to be restarted. In addition:
  - A primary failover now results in a takeover from the most suitable node.
  - If configured, external clients will automatically reconnect to the new primary and will issue a notification if the transaction has been lost.
  - A failed node, when restarted, will rejoin the group, recover its files and request a log update from the current primary. Once this task has completed it will be considered an active hot-standby and will continue to process replication requests as normal.
  - Users will experience a higher level of uptime/availability with their Fileshare configuration and will be able to recover from errors more easily.
- A new exit procedure, `ifsexitproc.cbl`, can be configured to automate some aspects of iFileshare behavior.
- The iFileshare Control page in ESMAC contains details of the current iFileshare high availability group.
- The following new iFileshare-specific environment variables are available:
  - FSWRKDIR - enables you to specify the Fileshare working directory, overriding the default, which is the system directory of the region.
  - FSCHKLFH - determines if a check is performed when a high availability group is started, to test the consistency of the data files within the group.
- The database reference file (`dbase.ref`) now supports wildcard matching for filenames, allowing you to perform operations on multiple files at once; for example: `fs /d dbase.ref /f data\*` adds the entire contents of the data directory to the database reference file.

IMS support

A new MFSGEN Compiler directive, CONTMVS, has been added to enable the compiler to parse MFS continuation lines (those continued without a terminating comma) by treating the line that follows as a comment line, just as is done on the mainframe.

Mainframe Access (MFA)

Enterprise Developer 3.0 provided the following improvements:

- Running MFA client in 64-bit processes - you can now call the MFA client modules from 64-bit applications and take advantage of the wider memory addressing available to those modules.
- MFA client CTF tracing - MFDAS now supports CTF tracing. This provides better diagnostics of issues and helps increase turnaround on issues. CTF is common to a number of Micro Focus modules so cross-component logging can be turned on from a single place.
- Endevor functions - the following functions are now available to Enterprise Sync and MFDAS: DELETE, DIRX, GENERATE, GETENQI, LIST, MOVE, SIGNIN and TRANSFER.

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

- Generating NUnit-style reports.
- 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.

PL/I support

Enhancements are available in the following areas:

PL/I compiler and run-time system

Support has been added to enable the following:

- The UNSIGNED data type - the PL/I Compiler now supports the UNSIGNED attribute for FIXED BINARY data items. In addition to the basic language support, the SIGNED, UNSIGNED, ISIGNED, and IUNSIGNED new built-in functions have been implemented. These features offer an improved compatibility with other languages such as C or C++.
- New built-in functions - UTF8(), UTF8TOCHAR(), UTF8TOWCHAR(), and SUBTRACT(). These new build-in functions make it easier to modernize migrated PL/I applications.
- The ENTRY LIMITED data attribute - the Open PL/I Compiler now accepts and fully supports the LIMITED attribute on ENTRY declarations. This increases the compatibility of the Micro Focus PL/I Compiler with IBM's PL/I for applications that you are migrating from the mainframe.
- Support for processing of JSON data - a number of new built-in functions are available for processing JSON text. This enables you to modernize your PL/I applications by enabling them to interpret, create and process JSON data.

JSONGETARRAYEND	JSONPUTARRAYEND	JSONGETMEMBER
JSONGETARRAYSTART	JSONPUTARRAYSTART	JSONGETVALUE
JSONGETCOLON	JSONPUTCOLON	JSONPUTMEMBER

JSONGETCOMMA
JSONGETOBJECTEND
JSONGETOBJECTSTART
JSONPUTCOMMA
JSONPUTOBJECTEND
JSONPUTOBJECTSTART
JSONPUTVALUE
JSONVALID

PL/I macro preprocessor

A number of enhancements have been made to improve the compatibility between the PL/I macro preprocessor and IBM's macro preprocessor which helps ease the effort required to migrate PL/I applications to new platforms. Support is available for:

- RESCAN in the ANSWER statement.
- The shorthand "<>" syntax for indicating "Not Equal To".
- Break characters in hex literal constants to improve readability of code.
- A new option, -tagdbg, that annotates macros and %FILE/%ENDFILE statements with line numbers and the source file which injected them.
- The -caseasis option which allows source code to flow through to the `mdeck.pp` file retaining the case as entered.
- Macros with a null parameter list.
- The LEAVE statement.
- Improved macro preprocessing speed.
- New built-in functions: SYSENDIAN() and MFPHOSTNAME().

Rumba terminal emulator

Enterprise Developer now ships with Micro Focus Rumba 9.5.

Back to Top

Transaction Classes

Transaction classes, which provide a mechanism to limit the number of CICS® tasks in your system and were previously a Technology Preview item, was supported at GA level from Enterprise Developer 3.0.

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 Enterprise Developer 2.3 Update 2

Enterprise Developer 2.3 Update 2 provided enhancements in the following areas:

- *Integration with the Visual Studio IDE*
- *Building applications*
- *COBOL language enhancements*
- *Classic Data File Tools*
- *Compiler directives*
- *Database access - DSNREXX*
- *Debugging applications*
- *Dialog System GUI*
- *Enterprise Server with .NET*
- *File handling*
- *IMS support*

- *Library routines*
- *PL/I support*
- *Tutorials*

Integration with the Visual Studio IDE

Enterprise Developer 2.3 update 2 provided the following enhancements in the integration of Micro Focus COBOL with the Visual Studio IDE:

- A new menu command **SQL Option for DB2** on the **Tools** menu enables you to access all the SQL Option for DB2 utilities.
- It is now possible to specify options for compiling native resource files (.rc) by accessing the file's properties from within the IDE.
- Editor improvements:
  - Support for showing the margins and the ruler is now available for mainframe subsystem files: Assembler, BMS, JCL, and IMS.
  - Colorization of the code in the ToolTips for collapsed regions in the COBOL editor has been added.
  - The Time and Date Warp options are now available on the **Run-Time Configuration** tab when editing an application configuration file for native COBOL projects.
  - **Go To Next Method** and **Go To Previous Method** editor context menu commands are now available for native Object-Oriented and for managed COBOL code.
  - Improvements have been made to editing performance.

Building applications

It is now possible to use the 64-bit version of MSBuild to build COBOL projects.

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.

Database access - DSNREXX

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.

DSNREXX support is available for use with Enterprise Server using DB2 LUW and SQL Server. See *DSNREXX Support* for more information.

Debugging applications

- You can configure the debug tooltip for subscripted OCCURS items to either display the elements of arrays or to display the value of the current expression. To do this, right-click in the editor, click **COBOL Debug Tooltip Style** and either enable or disable **Show COBOL OCCURS or reference modification value**.

Dialog System GUI

Installing the version of the Compatibility AddPack for Visual COBOL provided with this release adds a new menu item, **Micro Focus Dialog System**, to the **Tools** menu of Visual Studio for starting the Dialog System GUI from within the IDE.

Enterprise Server for .NET

Enterprise Developer 2.3 update 2 provided enhancements in the following areas:

- Dynamic configuration of running regions using a new seesys command-line utility for re-cycling AppDomains such as when new versions of programs have been deployed. seesys enables you to replace any previously loaded programs without having to restart the region, or remove any unused SEPs.
- Performance improvements - it is now possible to configure JES regions to reuse AppDomains that were previously used to run a job's steps. The default behaviour is to unload the AppDomain at the end of each job. Reusing AppDomains improves a job's throughput by eliminating assembly reloads.
- JES security support equivalent to that of the native Enterprise Server (i.e. external security managers (ESMs)). Customers can now secure their JES regions using mechanisms similar to the native enterprise server.
- JES spool housekeeping - you can now use the MVSSPLHK utility to archive JES spool output file data.
- Share catalog/spool between JES regions - A JES catalog and spool file can now be shared between JES regions.
- Improvements to the system administration capabilities using the Microsoft Management Console (MMC):
  - A seemmcconfig command-line utility that enables you to configure regions.
  - A number of enhancements in the spool view such as additional filters and columns, sortable columns.
  - Shared catalog configuration and security configuration
  - JES program aliases
- Public class library enhancements - it is now possible to interact with the region to access the spool functions and the security options.
- FTP and sFTP support for JES jobs that enables you to perform FTP functions from job steps, with or without a secure layer.
- Support for IDCAMS DEFINE CLUSTER SHAREOPTIONS - provides improved mainframe emulation of sharing VSAM files between tasks or jobs.

File handling

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

IMS support

Enterprise Developer 2.3 update 2 supports the PROMPT keyword in MFS source files.

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.

PL/I support

Enterprise Developer 2.3 update 2 provided the following enhancements:

- New built-in functions - support has been added for the ONCONDID() and SAMEKEY() built-in functions.
- Support for strings declared with the VARYINGZ attribute - the Open PL/I compiler and run-time system now support the use of VARYINGZ strings as variables, parameters, and targets for built-in functions. This provides an improved compatibility with IBM's mainframe compiler, and the ability to more easily interact with other languages such as ANSI C.
- Improved PL/I macro preprocessor compatibility - the Open PL/I compiler now supports all macro preprocessor built-in functions used by IBM's PL/I Macro Preprocessor.
- Improved compatibility with IBM's Structure Alignment rules - the -zalign option of the Open PL/I compiler now maps structure elements of the BIT data type that do not consume a full byte to the same boundaries as IBM's PL/I compiler. This helps make the migration of mainframe data files to a distributed platform easier and improves the program execution compatibility.
- Improved floating point support - the Open PL/I Code Generator now utilizes the newer Intel floating point operands for improved speed. In addition, the Code Generator now provides improved compatibility with passing parameters between C and COBOL.
- PL/I editor improvements:
  - The drop-down list of types and members in the Visual Studio editor now shows the PL/I types and members.

Tutorials

The product help includes the following new tutorials:

- *Tutorial: SQL - Deploying an Enterprise JavaBean Containing JVM COBOL to a JBoss Application Server* - an Eclipse-based tutorial that walks you through the process of deploying an EJB that contains JVM COBOL code.
- *Tutorial: DSNREXX*, that walks you through the process of executing DB2 LUW queries using DSNREXX.

What was New in Enterprise Developer 2.3 Update 1

Enterprise Developer 2.3 Update 1 provided enhancements in the following areas:

- *Integration with the Visual Studio IDE*
- *Application Server JCA support for Enterprise Server*
- *CICS Web services (Technology Preview)*
- *Code Analysis*
- *Code Coverage*
- *Compiler directives*
- *Database Access - HCO for SQL Server*
- *Database Access - Mainframe Batch Database Tools (MBDT)*

- *Database Access - OpenESQL*
- *Data File Tools*
- *Debugging applications*
- *Dialog System Applications*
- *Editor writing assistance*
- *Enterprise COBOL 5.2*
- *Enterprise Server with .NET*
- *File Handling*
- *Library routines*
- *Managed COBOL Syntax*
- *Native COBOL Syntax*
- *PL/I support*
- *Rosetta Stone for COBOL, .NET and Java Developers*

Integration with the Visual Studio IDE

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

- The **Go To location** and **Go To Procedure Division** commands are now available from the **Edit** menu. In addition, the **Go To Procedure Division** has a shortcut key - **Ctrl+K, Ctrl+J**.
- Object Browser - now provides support for .NET COBOL code. You can view the members of objects in your project and the definition of classes and methods as specified by the XML documentation comments in your code. The window also provides support for the **Go To Definition** command.
- A new search option, **Current COBOL Program**, has been added to the **Find in Files** dialog box. This enables you to search in the COBOL program currently opened in the editor and in any copybooks referenced by that program.
- Solution Explorer - a new context menu command, **Add Folder to Copybook Paths**, is now available for the subfolders of your projects. This enables you to add the subfolders to the copybook path of the project. For Mainframe Subsystem Projects, similar commands, **Add Folder to Assembler Macro Paths** and **Add Folder to IMS Copy Paths**, are now available to help add subfolders that include Assembler macro paths or IMS copy file to the Dependency Paths of the project.

Editor improvements:

- Line numbers - a new option for configuring the increment for the COBOL and the standard line numbers when you use Renumber and Unnumber has been added. You can specify the increment from **Tools > Options > Text Editor > Micro Focus COBOL > Line Numbering**.
- Outlining - a new IDE option enables you to switch outlining in the COBOL editor on or off. See the **Enable Outlining** option available on the **Advanced** page in **Tools > Options > Text Editor > Micro Focus COBOL**.
- Program breakpoints - a new context menu command, **Add COBOL Program Breakpoint**, enables you to set program breakpoints directly from the editor.

Application Server JCA support for Enterprise Server

Back to Top

This release provides support for:

- Automatic connection recovery to an active Java application server when an enterprise server region is restarted. This applies to:
  - COBOL resource adapters
  - CICS resource adapters
  - All CICS API based programs
- The IBM implementation of CICS resource adapter for WebSphere 8.5 and JBoss 7.1.1.

CICS Web services (Technology Preview)

Note: This was introduced as 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 provided an initial support for CICS Web services (CWS) using an SOAP-based provider and requester modes and using HTTP or MQ as the transport mechanism. CICS Web services support the Coded Character Set Identifier (CCSID) and configuring the CCSID is a prerequisite for CWS in all environments.

Restriction: Security and Web Service Atomic Transactions (WSAT) are not currently supported.

Code analysis

Back to Top

This release provides support for performing code analysis at the command line using Microsoft's MSBuild utility which enables the integration of code analysis in CI frameworks.

Support is available for various MSBuild parameters. You can use MSBuild with the /p switch and with the following Micro Focus-specific command line options:

- - `RunMicroFocusCodeAnalysisAfterBuild` - controls whether code analysis is performed with MSBuild or not regardless of what code analysis options are specified the project's properties.
  - `ActiveRulesets` - specifies which rule sets should be used when running code analysis from the command line.

Code coverage

Back to Top

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

- A new Code Coverage window showing the statistics of what percentage of the code has executed.
- Navigation from the Code Coverage window 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 - HCO for SQL Server

Back to Top

This release provides the following new features:

- End-to-end demonstrations that show several methods for submitting a JCL batch job that calls COBOL stored procedures. To view these demos, see the *SQL* demonstrations in the Mainframe Samples browser.

- The SPCALLLOCAL SQL compiler directive option, used to convert an EXEC SQL CALL into a call to a local COBOL subroutine instead of a call to a SQL Server SQLCLR stored procedure.
- Extended support for conversion of the DB2 CONCAT function.

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.

Database Access - OpenESQL

Back to Top

This release provides the following new features:

- Demonstrations for using OpenESQL in .NET applications. To view these demos, see the *SQL* demonstrations in the Visual COBOL Samples browser.
- Support for the Oracle fully managed ODP.NET driver.

Database Access - SQL Option for DB2

Back to Top

This release provides the following new features:

- Import/export of XML with SQLWizard/DSNUTILB(XUTLB)
- New XDBSetup utility for TS/Citrix environments
- Using Migrate to move XML tables from z/OS DB2 to XDB
- Support for Temporal 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

Debugging applications

- You can configure the debug tooltip for subscripted OCCURS items to either display the elements of arrays or to display the value of the current expression. To do this, right-click in the editor, click **COBOL Debug Tooltip Style** and either enable or disable **Show COBOL OCCURS or reference modification value**.

Dialog System Applications

Back to Top

The version of the Compatibility AddPack for Visual COBOL released with Enterprise Developer 2.3 Update 1 enables you to give your Dialog System applications a refreshed user interface. Features include:

- Support for a modern look and feel for Dialog System applications.

  Dialog System now supports Microsoft's visual styles for controls and fonts to give your existing Dialog System applications a modern look that is native to the Windows version the applications are running on.

  You use a new environment variable, MFVSSW, to switch the visual styles on. Alternatively, for applications that compile to an executable, to switch the new styles, you can add an application manifest file to the applications' project files.

  See the section *Modernizing Dialog System Applications* in your product help for details on how to enable the visual styles, and for information about any possible changes in behavior or appearance of the common controls.
- A modern look and feel of the Dialog System painter.

  There are new configuration options in the Dialog System painter for applying the visual styles and fonts to the painter.
- Visual Studio project templates for creating Dialog System applications with a modern look out-of-the box.

  The Compatibility AddPack for Visual COBOL now installs the following new project templates:

  - **Dialog System Application (Modern)** for creating Dialog System applications that use Microsoft's visual styles by default.
  - **Dialog System Application (Classic)** for creating Dialog System applications that use the classic fonts and look.
- Additional samples:

  All of the Dialog System samples previously available with Net Express are now included with the Compatibility AddPack for Visual COBOL. The samples have been converted to Visual Studio solutions.

For more information see the section *Modernizing Dialog System Applications* in your product help and the Release Notes for the Compatibility AddPack for Visual COBOL.

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.

Enterprise Server for .NET

Enterprise Developer 2.3 Update 1 provides enhancements in the following areas:

- Support is provided for DSNRLI, the language interface load module for the DB2 Resource Recovery Services attachment facility (RRSAF). The interface to RRSAF is only through the DSNRLI entry point which provides explicit service connection requests for any XA resource, including WebSphere MQ.
- HSF support has been extended and is now available in JES regions.
- The EXEC CICS POST and WAIT API's are now implemented.
- Support for EZASOKET is now available in both CICS and JES regions.
- SQL API diagnostic events are now being recorded and displayed in the local trace tables.

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.
- PC_PRINTER_INFO_DOTNET (.NET COBOL only) - enables the access to the native Hdc for a printer.

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 for JVM COBOL 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.

PL/I support

Enterprise Developer 2.3 Update 1 provides the following enhancements:

- FIXED BIN(63) data type - PL/I programs that utilize FIXED BIN data types that are larger than FIXED BIN(31) are now supported which enables you to migrate such programs more easily to distributed platforms.
- TRIM() built-in functions - the Open PL/I macro preprocessor now supports the TRIM() built-in function.
- A 64-bit Open PL/I peephole optimizer - when compiling 64-bit Open PL/I programs optimization for speed, a peephole optimizer is now run against the intermediate code to improve your applications' performance. Any code which has attributes that benefit from peephole optimizations will then run faster.

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.

Windows Azure

Back to Top

Enterprise Developer support on Microsoft Azure has been updated to version 2.8 of the Microsoft Azure SDK.

⚠ **Important:**

- Starting with this release, versions of the Microsoft Azure SDK earlier than version 2.8 are no longer supported.
- Support for the Microsoft Azure SDK is no longer provided in Enterprise Developer for Visual Studio 2012. If you have COBOL Azure projects created with earlier versions of Enterprise Developer for Visual Studio 2012, to maintain them, use Enterprise Developer for Visual Studio 2013 or 2015.

What was New in Enterprise Developer 2.3

Enterprise Developer 2.3 provided enhancements in the following areas:

- *Integration with Visual Studio 2015*
- *General IDE enhancements*
- *Building applications in Visual Studio*
- *COBOL editor in Visual Studio*
- *Code analysis*
- *Code coverage*
- *Command Line Compilation and Linkage*
- *Compiler directives*
- *Converting additional directives to project's properties*
- *Data File Structure command line utility*
- *Data File Tools (Technology Preview)*
- *Database access*
- *File handling*
- *Library routines*
- *Managed COBOL syntax*
- *Micro Focus Infocenter*
- *Micro Focus Unit Testing Framework*
- *Microsoft Azure*
- *Preprocessors*
- *Profiler*
- *REST service interfaces*
- *RM/COBOL Compatibility*
- *Single file support*
- *Tunables*
- *Updated run-time system*

Integration with Visual Studio 2015

This release provides a new Enterprise Developer flavor that takes advantage of the new features of Visual Studio 2015, including:

Light Bulbs	You can use the light bulb feature (💡) to quickly implement unimplemented interfaces or to resolve unknown types.
Compatibility (Project Round-Tripping)	Enterprise Developer supports the Visual Studio project round-tripping feature which enables you to work with COBOL projects in Visual Studio 2012, Visual Studio 2013 or in Visual Studio 2015 without the need to upgrade the project file.
.NET Framework 4.6 Compatibility	Provides support for creating managed COBOL applications that target version 4.6 of the .NET Framework.
Microsoft Help Viewer 2.2	Provides support for the Micro Focus Help in Microsoft Help Viewer 2.2.

General IDE enhancements

In Visual Studio:

- Menu items - the following Micro Focus utilities were previously available under the **Tools > Micro Focus** menu and have been moved as follows:
  - **ADO.NET Connection Editor** - now accessed from **View > Micro Focus SQL Tools**.
  - **MFA Compare and Synchronization Monitor** - now available directly under the **Tools** menu.
  - **MFA Server Access Configuration** - now available directly under the **Tools** menu.
  - **OpenESQL Assistant** - now accessed from **View > Micro Focus SQL Tools**.
  - **OpenESQL Configuration Utility** - now accessed from **View > Micro Focus SQL Tools**.
  - **Web Service Client** - now accessed by creating a new project of type "Web Service Client Application from JSON/REST" - click **File > New > Project > COBOL > Native > Web Service Client Application from JSON/REST**.
- Managed COBOL project properties - a new setting, **Expose group linkage items to managed code**, is now available on the **COBOL** tab in the properties of managed COBOL projects. Checking this sets the ILSMARTLINKAGE directive that specifies that group linkage items in your code are to be exposed as properties of a class. A related button, **Options**, enables you to also specify whether to remove specific prefixes from the names of COBOL data items, whether to make the new classes serializable, expose Linkage Section items as nested classes of the program class or to limit the property generation to non-redefining elementary items.
- PL/I project properties - there is now a new page, **Assembler**, in the project properties of PL/I Mainframe Subsystem Projects that enables you to set properties for configuring PL/I Assembler applications.
- Resetting the file properties - a new button, **Use Project Defaults**, on the file property pages of COBOL files in native projects now enables you to reset the file properties of COBOL files in native COBOL projects.

Building applications

Enterprise Developer now supports Visual Studio's parallel builds for COBOL projects. Parallel building enables you to build multiple projects faster on multi-CPU machines. Following this change, in order for your multi-project solutions to build in parallel successfully, ensure that the project dependencies and build order are set correctly for your solution using **Project > Project Dependencies**. For more information about parallel building, see *Tips on Building COBOL* in your product help and your product Release Notes.

Note: Parallel builds are not supported with Mainframe Subsystem COBOL projects.

COBOL editor

The COBOL editor includes the following enhancements and new features:

- Improved Intellisense writing assistance for COBOL in both native and managed COBOL:
  - Context sensitive suggestions - IntelliSense only shows suggestions that are relevant for the position of the cursor in the code or for the type of project.
  - Enhanced completion lists - lists include any relevant COBOL verbs, clauses and words, copybooks, code templates, data items and section and paragraph names.
  - Intelligent assistance with completing statements - when you have entered a COBOL verb, IntelliSense shows suggestions for the relevant clauses and identifiers that you can use to complete the statement.
  - Automatic completion for single items - IntelliSense automatically inserts single suggestions in the code.
  - Qualifying non-unique names - IntelliSense qualifies data items whose names are not unique.
  - Configuration preferences for IntelliSense - enable you to configure what suggestions appear in the completion lists, whether suggestions are added in insert or overwrite mode, and the case of the inserted words.
  - Snippets - code snippets are now included in the IntelliSense suggestions.
- $IF-$ELSE-$END statement colorization - parts of a $IF Compiler-control statement that are not executed are now colored in grey.
- $REGION statement - support is provided for the $REGION Compiler-control statement. You can use $REGION - $END-REGION to surround blocks of code that you want to folded or expand in the editor.
- Copybook glyphs (▷) in the left-hand margin on a copy line - indicate that a copybook can be expanded inline.
- Creating copybooks from selection of code - a new context menu command, **Extract to Copybook** in the editor enables you to move a selection of COBOL code into a new copybook file. The file is added to the project and the code in the original program is replaced with a COPY statement that refers the new copybook.
- Expanded copybook view - provides indicators for read-only copybooks
- Outlining - outlining is now available on comment blocks, paragraphs and on $REGION and $IF statements.
- Task List comments - the COBOL editor now provides support for Visual Studio-style Task List comments in COBOL. Typing TODO, HACK or UNDONE in the code immediately after the declaration of the COBOL comment (*>, *>> or * in column 7) creates a task that appears in the **Task List** window.

Code analysis

Enterprise Developer 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

Enterprise Developer 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.

Command Line Compilation and Linkage

When using the `cbllink` command to compile and link, there is a new `-y` option. Use this option to create an executable that includes support to be able to run on Windows XP and Windows Server 2003.

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.

Converting additional directives to project's properties

The **Update Project Properties** utility allows you to convert additional directives in your project to project properties. This option is presented on opening a project. You can select to not show this dialog again by unchecking **Check Additional Directives for project properties** in **Tools > Options > Micro Focus > General**.

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 Enterprise Developer's command prompt or a terminal.

To use the new editor directly from the Visual Studio IDE, clear the **Use classic data file tools** check box, available from **Tools** > **Options** > **Micro Focus** > **Data File Tools**. When cleared, the Data File Tools (Technology Preview) version is used, when possible. When this editor does not support the action you are attempting to complete, Classic Data File Tools is used instead. This check box is selected by default.

Database Access

Enterprise Developer version 2.3 provides the following enhancements to database access:

COBSQL Enterprise Developer version 2.3 provides:

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

HCO for DB2 LUW

Enterprise Developer version 2.3 provides:

- GUI versions of data migration tooling
- A new tool that enables you to bind or rebind all packages. See *BindAll Packages Tool* for details.
- 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.
- 64-bit support for PL/I on appropriate platforms. See *Additional Software Requirements on Windows* for details.

HCO for SQL Server

Enterprise Developer version 2.3 provides:

- Support for the DATA-CHANGE-TABLE-REFERENCE clause.
- The following HCOSS Add-Ins have been converted to VSPackage Extensions:
  - HCO for SQL Server Tools, now available under **View > Micro Focus SQL Tools**
  - Options for HCO for SQL Server Tools, now available under **Tools > Options > Micro Focus**
- Context Help is now provided for:
  - Options for HCO for SQL Server tools
  - For each tool in the HCO for SQL Server user interface

OpenESQL

ADO.NET Connection Editor

In this release:

- The ADO.NET Connection Editor has been redesigned using a series of Wizards that guide you through the processes of adding, copying, and removing connections.
- Context Help is now provided for the main window, and each Wizard page and dialog box.

Date/Time Processing

This release provides streamlined datetime processing for ODBC and JDBC and ADO.NET.

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
- Npgsql ADO.NET 4.0 driver version 2.2.5

PostgreSQL 9.4 has been tested with OpenESQL and OpenESQL Assistant on the following Windows platforms:

- Windows 32-bit
- Windows 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 Enterprise Developer version 2.3 provides support for the SQL Server OUTPUT clause.

Mainframe Batch Database Tools (MBDT) Windows only. Enterprise Developer version 2.3 includes mainframe batch utilities that provide the following functionality for Oracle, DB2 LUW, and SQL Server agnostically, as well as specifically for HCO for SQL Server:

- DSNTEP2
- DSNUTILB LOAD
- DSNUTILB UNLOAD
- 32-bit and 64-bit support
- New batch configuration utility

SQL Option for DB2 Enterprise Developer version 2.3 provides the following z/OS DB2 version 10 enhancements:

- Extended support for implicit casting
- Greater timestamp precision for applications
- Support for TIMESTAMP WITH TIME ZONE

This release provides initial XML support for the XDB engine.

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 CICS and 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.

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 an Enterprise Developer command prompt. A test fixture or suite is a

COBOL program compiled to `.dll` 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.

Microsoft Azure support

Enterprise Developer support for Microsoft Azure has been updated to version 2.6 of the Microsoft Azure SDK.

Support has been added to the product for making any future versions of the Microsoft Azure SDK available before the next major release of Enterprise Developer. Micro Focus will deliver support for these only upon customers' requests.

Note:

2.3 HotFix 1 update: This HotFix provides support for version 2.7 of the Microsoft Azure SDK in Enterprise Developer for Visual Studio 2013 and 2015.

Preprocessors

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

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

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

Profiler

Enterprise Developer 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 Enterprise Developer.

Single file support

The recommended way to work with files within Enterprise Developer is to include them in a project. For situations where you might want to quickly open edit a single file, Enterprise Developer 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.

To enable full IDE support for single files, Enterprise Developer provides a path for creating projects from them - right-click such files in the editor, and click **Create COBOL Project**.

Tunables

Enterprise Developer 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

Enterprise 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 Enterprise Developer, you should ensure that the executable is rebuilt and packaged with the new run-time system. This does not affect mainframe subsystem projects deployed to Enterprise Developer. You can rebuild from the IDE or the command line.

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

What was New in Enterprise Developer 2.2 Update 2

Enterprise Developer 2.2 Update 2 provided enhancements in the following areas:

- *Visual Studio*
- *Character Set Enhancements*
- *Code Analysis*
- *Database Access*
- *Enterprise COBOL 5.1 compatibility*
- *External Call Interface (ECI)*
- *Micro Focus COBOL enhancements*
- *External Security Facility (ESF)*
- *Enterprise Server for .NET*
- *Enterprise Server MQ-IMS Bridge*
- *IMS Connect*
- *PL/I General Enhancements*
- *RUMBA 9.2*
- *Transaction Classes (Technology Preview)*
- *Tunables*

Visual Studio

Enterprise Developer 2.2 Update 2 provides enhancements in the following areas:

- Call Hierarchy - support for the Call Hierarchy window has been enhanced and it now shows types and members across the entire solution.
- Debugging enhancements - includes enhanced support for the Autos windows for native and managed COBOL, support for querying EBCDIC data in managed code, and support for the standard visualizers for data items and groups in native COBOL.
- Expanded Copybook View - now supports Intellisense, collapsing and expanding of outlining regions, and code snippets.
- Find All References - you can now configure the scope of the Find All References command. The default behavior is to search for references in the current COBOL project and now, you can enable Find All References to search for matches in all managed COBOL projects that are part of your solution.

- Net Express Project Import wizard - now includes a number of usability enhancements.
- Run-time configuration - you can now use the application configuration file, `Application.config`, in native COBOL projects to set all run-time 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

Enterprise Developer for Visual Studio can produce the following COBOL reports:

- Dead Code - finds unreferenced items or any piece of code that can't be reached during execution.
- Unreferenced Data - finds any data items that are not explicitly referenced in the Procedure Division of a program.
- Undeclared Procedures - finds any procedures that are referred to but not defined.
- Copybook Structure - displays the hierarchy of any copybooks defined in a program.
- Program Statistics - provides general information, such as number of source code lines, number of data items, and size of data items.
- Unexecuted Procedures - finds any procedures that are defined but not referred to.

Database Access

The following new features are available in database access support:

COBSQL

In Visual Studio:

- Pro*COBOL support - you can now select and configure the Pro*COBOL COBSQL preprocessor for compiling COBSQL applications on the **SQL** tab in the project's properties.
- KEEPCOMP - the new KEEPCOMP directive resolves COMP/COMP-5 issues with Oracle applications on little-endian platforms.

HCO for DB2 LUW

Enterprise Developer 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.

HCO for SQL Server (HCOSS)

Enterprise Developer 2.2 Update 2 provides the following new HCOSS features:

- DATE/TIME formatting - the OpenESQL Configuration Utility now supports DATE/TIME formatting for HCOSS batch utilities.
- Multi-row INSERT - supports a multi-row INSERT statement.
- Sample applications - Enterprise Developer for Visual Studio includes the following new mainframe HCOSS sample applications:
  - Array Insert - demonstrates how array inserts work with SQL Server using mainframe DB2 syntax.

    - Multiple Result Sets - demonstrates how to create a SQL Server SQL CLR stored procedure with multiple result sets and how to access them from a client program using mainframe DB2 syntax.
- Sample application - Enterprise Developer for Eclipse includes the following new mainframe HCOSS sample application: the **Array Insert** sample demonstrates how array inserts work with SQL Server using mainframe DB2 syntax.

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.

SQL Option for DB2

Support for the following has been added in this version:

- DSNUTILB LOAD LOG=NO option
- RID scalar function
- XML data type via XDB Link to mainframe DB2

XA switch modules

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.
- SQL Server switch module:
    - Enables you to specify which XA resource definitions use User Impersonation.
    - Now compiled with one source file, rather than two.

Enterprise COBOL 5.1 compatibility

There have been a number of enhancements to make this version of Enterprise Developer more compatible with Enterprise COBOL 5.1.

Note: These enhancements are available under both DIALECT(MF) and DIALECT(ENTCOBOL).

- The following phrases have been added to the XML GENERATE statement:
    - NAME
    - SUPPRESS
    - TYPE
- The following intrinsic functions have been added:
    - ULENGTH

    - UPOS
    - USUBSTR
    - USUPPLEMENTARY
    - UVALID
    - UWIDTH
- A number of reserved and context-sensitive reserved words have been added; see the topics *Reserved Words Table* and *Context-sensitive Words Table* for complete lists.

External Call Interface (ECI)

Enhancements to the ECI include:

- The ECI for Java clients is no longer restricted to the 32K Commarea. You can now transfer virtually unlimited containers of any size in a channel to a CICS server program. See *ECI Java Interface* for more information.
- Java support for IBM's implementation of ECI, which, in turn, uses the Micro Focus implementation.
- A ECI RA for WebSphere; however, in this release, no IMTK tooling is provided for servlets or JSP.

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 for .NET

Enhancements are available in the following areas:

- Azure deployment - the Azure deployment now deploys to the latest worker role guest OS version, Windows Server 2012 R2, which has the latest .NET frameworks (versions 4.0, 4.5 and 4.5.1) installed.
- CICS/JCL Subsystem Started Task Manager - Subsystem Started Task Manager (SSTM) provides JES functionality within a CICS region and enables the region to submit batch work to a JES region. The JES functionality is equivalent to that of a task started under z/OS and must support CICS spooling.

CICS spooling enables spooling files to the JES region or submitting JCL jobs. CICS spooling does this using its internal reader or using the SPOOLOPEN, SPOOLWRITE and SPOOLCLOSE APIs.
- Diagnostics - the diagnostic collection process now provides more detailed diagnostic information.
- JCL initiator node affinity - node affinity enables a particular initiator to be associated with a particular machine and execute certain types of work only on that machine. An example of this is running a particular job on a machine with specific third-party software installed, for which only one license exists.
- MQ Series support - the WebSphere MQ .NET managed code client only provides a class library interface and not the traditional MQI (MQ programming interface). The MQI interface is now implemented within Enterprise Server for .NET which enables you to run applications that utilize the MQI without any code changes.
- System administration - this release includes two new views, Sessions view and Request view, that enable system administrators to see which sessions are attached to a region, which tasks are active and which ones are queued.

Enterprise Server for .NET provides a better integration with Visual Studio as follows:

- Associating projects with regions - support for associating your managed CICS or JCL projects with a region is available from within Visual Studio. This is to provide the ports required for JVCL submission and for TN3270 integration.
- Submit JCL from within Visual Studio - for a project that is associated with a JCL-enabled region, you can now submit JCL files that are part of the project directly from within Visual Studio.
- Embedded TN3270 Rumba mainframe display - you can now use the **TN3270 Rumba mainframe display** embedded in Enterprise Developer to run managed CICS applications from within Visual Studio.
- Viewing the console log in Visual Studio - you can now display the console log of a region associated with your project in the **Output** window in Visual Studio.

Enterprise Server MQ-IMS Bridge

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

IMS Connect

Visual COBOL 2.2 Update 2 now supports Java IMS Connect messages except when also using the IBM-supplied HWSJAVA0 I/O exit, excluding customizations to that exit.

PL/I General Enhancements

Enhancements are provided in the following areas:

- Attributes - the PL/I DECLARE statement now supports the OPTIONAL attribute as part of the parameter-descriptor list or as an attribute in a parameter declaration. This provides an improved functionality and a greater language compatibility for customers looking to migrate applications from z/OS environments.
- Built-in functions - the PL/I Compiler and run-time system now support the PRESENT and OMITTED built-in functions which provides an improved functionality and a greater language compatibility if you are looking to migrate applications from z/OS environments.
- CodeWatch - this release includes the following enhancements:
    - CodeWatch is now supported in native 64-bit environments which ensures that the CodeWatch debugger runs properly in native 64-bit mode. This enables the PL/I debugger to take advantage of faster speeds and to debug larger programs which have access to larger amounts of storage.
    - Notifications for the CodeWatch conditions - you can use the command line and the UI to turn notifications on or off for all of the currently documented PL/I conditions. The use of notifications enables you to investigate the state of programs and optionally set breakpoints before continuing program execution.

- Compound operators - the PL/I macro preprocessor now supports the compound operators += , -=, *=, and /= within macro assignment statements.
- Error handling - you can now use the EXEC CICS HANDLE ABEND check which provides a consistent error handling technique for PL/I programs running under CICS that is consistent with error handling on the mainframe. EXEC CICS HANDLE ABEND determines error handling for programs running under CICS. Conditions detected by CICS cause an abend if established using EXEC CICS HANDLE ABEND, and PL/I ON units do not get control. If EXEC CICS HANDLE ABEND is not established, CICS defines that the action taken will be as defined by CICS, which normally leads to a transaction abend.
- PL/I data structures - this release provides enhanced support for PL/I data structures in the Data File Tools. You can now easily generate a PL/I structure map (.str) from within the IDE, and the Data File Tools can then use to generate a formatted layout for a PL/I data file. This enables you to see what is in each field within a PL/I Data File Record, and to modify the records in a logical way without having to know the offset, length and raw format of the data item.

Rumba 9.2

Visual COBOL 2.2 Update 2 of Enterprise Developer includes Micro Focus Rumba 9.2 which you can optionally install as part of the setup process. The license for Enterprise Developer will license all components of Rumba.

Transaction Classes (Technology Preview)

Transaction classes provide a mechanism to limit the number of CICS® tasks in your system. By spreading your tasks across a number of transaction classes and configuring the maximum number of tasks that can be dispatched within each transaction class, you can control resource contention between tasks and limit the number of tasks that CICS considers eligible for dispatching at task attach.

Tunables

Enterprise Developer 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 Enterprise Developer 2.2 Update 1

Enterprise Developer 2.2 Update 1 provided enhancements in the following areas:

- *Visual Studio IDE*
- *Micro Focus Heartbleed Update*
- *ACUCOBOL-GT Compatibility*
- *Btrieve Support*
- *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*
- *Local Catalog*
- *Mainframe Emulation*
- *Managed COBOL*
- *Managed COBOL Syntax*
- *PL/I*

- *Preprocessors*
- *Rumba Integration with Visual Studio*
- *Run-time Launch Configuration Files*

Visual Studio IDE

Enterprise Developer 2.2 Update 1 provided the following enhancements to COBOL support in the IDE:

Expanded Copybook View

The Expanded Copybook View functionality in the editor was added in the 2.2 release and this was enhanced in 2.2 Update 1 so it is now possible to debug into the expanded copybooks. The functionality includes:

- Stepping inline in the expanded copybook view when debugging - when stepping into or hitting a breakpoint in the copybook, the copybook automatically expands. You can disable this feature from the IDE preferences.
- Showing Pinned Data tips inside the expanded copybook.
- Support for **QuickWatch** and **Add Watch** inside the expanded copybook.
- Executing some of the debugger commands from the context menu from the expanded copybook view - for example, **Show Next Statement**, **Run To Cursor** and **Set Next Statement**.

Some limitations apply.

Call Hierarchy

Enterprise Developer now supports the Visual Studio **Call Hierarchy** window for analyzing COBOL PERFORM statements.

Find All References

The Find All References functionality now works for managed OO COBOL code.

Navigation

You can now use **Alt+Shift**+Up Arrow (or Down Arrow) to navigate the variables in the editor.

You can use the drop-down types and members to navigate to the different sections (Linkage section, File section etc.) in your source files.

Project Details Window

The **Project Details Window** now includes columns showing the COBOL dialect of the files and the SQL properties set on them.

Run-Time Configuration

- You can now use the application configuration file (`application.config`) in native COBOL projects to specify some additional settings such as the search order for called programs, command line handling, or file handling. You can do this from the new **Run-Time Configuration** tab.
- When an executable is built using the IDE, the application configuration file is automatically copied to the output folder and is renamed as `<`*`executable-base-name`*`>.exe.mfgcf`.

Samples

Visual COBOL 2.2 Update 1 includes the following new samples:

- Airport Demo (managed) - shows how to create a COBOL WCF REST service and then use a client application to consume it.
- Airport Demo (native) - a basic lookup program that reads the information about airports from a .dat file and outputs the distance between two airports.

The Sandcastle sample has been updated and now uses Sandcastle Help File Builder v1.9.8.0.

Solution Explorer

Support is now available in Solution Explorer for the following features:

- Errors and warnings filters - click the arrow next to the (Pending Changes Filter) icon in the Solution Explorer toolbar, and either click **Errors Filter** or

Errors and Warnings Filter to show the files that result in errors or also cause warnings.

- Searching in Solution Explorer - You can use the search field in Solution Explorer to search for files in your solution. Searching also finds copybooks in the copybook dependency view even if they are not part of any of the projects in the solution.

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.

Btrieve Support

Support for the Btrieve file handling system from Pervasive Software Inc. has been added into Enterprise Developer.

Support is restricted to native COBOL, in a Windows environment.

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 Enterprise Developer 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:

HCO for DB2 LUW

- Support added for DB2 LUW version 10.5.
- Enhanced RETURN-CODE processing.
- The HCO DDL processor now accepts column names with a length greater than 18 characters.
- The HCO Import Data tool now processes CLOB data less than 32K in size.
- The HCO Import Data tool can now convert DSNTIAUL import files to delimited format, enabling the actual import to be done by DB2 LUW IMPORT and LOAD utilities. This is done using the new SAVEDEL and CHARDEL parameters.

HCO for SQL Server (HCOSS)

- Support for the mainframe SET CURRENT SQLID statement.
- Support for the mainframe LEFT and RIGHT scalar functions.
- Data Migration tools now support the mainframe ALIAS and SYNONYM objects.

OpenESQL

- Enhanced internationalization support for UNICODE, DBCS and MBCS.
- Enhanced GET DIAGNOSTICS statement support.
- Enhanced LOB support for CLOB, BLOB and DBCLOB data types.
- ADO.NET connection editor now provides context help.
- 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 the Output window.

Enterprise Server MQ-IMS Bridge (Technology Preview)

Note: At Enterprise Developer 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.

Enterprise Developer 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

Enterprise Developer 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.

Local Catalog

From within Server Explorer in the IDE, you can add one or more local catalogs, and then use them in conjunction with Drag and Drop to drag down data set items from your mainframe, edit them and then move them back to the mainframe or to your testing environment.

Mainframe Emulation

Previously, MGMTCLAS was supported by simply saving it on the catalog record when a dataset was allocated in JCL using the MGMTCLAS parameter on the DD statement. Starting with this release, you can use batch jobs to define MGMT classes and to specify the EXPIRE AFTER time (in days).

Managed COBOL

Enterprise Developer 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

Enterprise Developer 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.

PL/I

PL/I support within the IDE now includes the following enhancements:

- Support for new built-in functions

  The PL/I Compiler and Runtime now support the following new built-in functions:

  DAYS
  DAYSTODATE
  SECSTODATE
  SECS
  WEEKDAY
  DAYSTOSECS
  ONSUBCODE
  REPATTERN
  IAND
  IOR
  INOT
  OMITTED
  XMLCHAR

  This provides an improved functionality and a greater language compatibility when you migrate applications from a z/OS environment.
- Support for %OPTION in the PL/I macro preprocessor

  You no longer need to modify application code that uses %OPTION in order for it to compile successfully.
- Support for QUOTE() and COMMENT() built in functions.

  This enables the use of PL/I macros that generate code using these built-in functions.
- Support for IBM Style %INCLUDE(s) that utilize PDS name as part of include file names.

  The PL/I macro preprocessor now parses the %INCLUDE syntax of the form shown below and utilizes the –isuffix and –ipath Compiler directives to locate the “member”.

  ```
  %INCLUDE MYPDS(MYMEMBER);
  ```

  This provides an improved language compatibility with the application code without the need to modify the original underlying source code.
- Support for PL/I AREA data items up to 2Gb.

  Open PL/I now supports the declaration and usage of variables of type AREA up to 2Gb in size. This provides you with improved functionality and greater language compatibility when migrating applications from a z/OS environment as there is no need to modify the code in order to deal with a maximum area size of 32Kb.
- Enhanced compatibility with the DISPLAY() REPLY() syntax.

  The PL/I Compiler and runtime now provide a DISPLAY() REPLY() syntax that is no longer dependent upon the use of the SYSIN and SYSOUT DD’s and that no longer causes behavioral differences if stream IO to SYSIN/SYSOUT is intermingled with programs using DISPLAY() REPLY().

- Support for big-endian style bit strings on little-endian platforms.

  Using a new Compiler directive –bitsltr to compile programs that have logic that is dependent on big-endian bitstring ordering enables you to utilize the code unchanged on Open PL/I little-endian platforms (such as Windows, Red Hat Linux, SUSE Linux). This enables you to migrate applications that make an assumption about bit ordering on the underlying operating system or hardware and to run them unchanged.
- Support for the –initcall Compiler directive.

  Compiling a PL/I program with the –initcall directive causes the specified program to be invoked prior to any user logic. This enables you to customize your environments and do things such as tracking which routines have been invoked, detect memory corruption prior to routine startup, establish database connectivity.
- Support for the “F” option in PLIDUMP.

  A PLIDUMP which was generated using the “F” option will now attempt to list all known information about files which have been accessed during the execution of the program and includes information about the last IO operation, key values, and a dump of buffers, if available. This enables you to identify more easily the cause of file-related failures and what file operations were in progress at the time of failure.
- Technology Preview support for PL/I data structures Data File Tools.

  In the IDE, you can now generate a PL/I structure map (.str) which can be utilized by the Data File Tools utility to generate a formatted layout for a PL/I Data File. The PL/I data structure must be compiled in advance with the -adt Compiler directive. This enables you to see what is in each field within a PL/I data file record, and to modify them in a logical way without having to know the offset, length and raw format of the data item.
- Support for PL/I debugger notification for ON Unit invocation.

  You can now request that the debugger stops when a PL/I ON Unit is about to be triggered. This enables you to locate the ON Unit, establish appropriate breakpoints within the ON Unit and determine the cause of the ON Unit being triggered.
- Support for PL/I debugger notifications for the load of shared libraries and .dll files.

  You can now request that the debugger issues a notification when a shared object or a .dll file has been dynamically loaded. This enables you to stop a program when a dynamic program has been loaded and to establish the appropriate breakpoints, watch lists, etc.
- Alternative Index Performance Improvements for the CASSPOOL command.
- Support of the SHAREOPTIONS property in a VSAM cluster.
- Support for setting directives automatically for PL/I files.

  In Visual Studio, click **Tools > Options > Micro Focus > Directives > PL/I**.
- A new page in the project and file properties enables you to set SQL directives on PL/I source files.

Preprocessors

IDE support is now available for selecting and using third-party preprocessors installed on your machine and integrated with Enterprise Developer for building native COBOL Mainframe Subsystem applications.

A guide for third-party vendors and administrators showing how to integrate additional preprocessors with Enterprise Developer is now available in the product help in the *General Reference > Integrated Preprocessor Interface* section.

Rumba Integration with Visual Studio

Visual Studio now offers context menu commands for launching the Rumba mainframe display (Desktop or embedded) from Solution Explorer or from Server Explorer. The embedded Rumba display now provides keyboard mapping and color settings.

You can use the display of choice both in debugging and when running applications.

Run-time Launch Configuration Files

Use a run-time launch configuration file to ensure an application can be launched when it is deployed in a separate location to the run-time system (in the case of dynamically bound applications), or when the licensing daemon is not already running.

What was New in Enterprise Developer 2.2

Enterprise Developer 2.2 provided enhancements in the following areas:

- *Enhancements for Enterprise Developer 2.2 for Visual Studio 2012*
- *ACUCOBOL-GT Compatibility*
- *RM/COBOL Compatibility*
- *Application Configuration*
- *COBOL Source Information (CSI)*
- *Compiler Directives*
- *Consolidated Tracing Facility*
- *Converting Net Express Projects*
- *Enhanced Accept and Display Statements*
- *Debugging*
- *Grouping Files in Virtual Folders in Solution Explorer*
- *File Handling*
- *Interface Mapping Toolkit*
- *Managed COBOL*
-
- *Searching in Copybooks*
- *COBOL Editor Enhancements*
- *Upgrading from Net Express to Visual COBOL*
- *WCF REST Service Application projects*
- *Microsoft Windows Azure*
- *XML Extensions*

Enhancements for Enterprise Developer 2.2 for Visual Studio 2012

The following enhancements are applicable to Enterprise Developer for Visual Studio 2012:

- Copybook Dependencies - Solution Explorer shows a tree view of the dependencies of the COBOL programs and copybooks on copybooks in a project to help you navigate easily around the source code.
- Expanded Copybook View: Technical Preview - The COBOL editor supports showing the contents of copybooks inline at the place where they are referenced in a program or a copybook. Features include:
  - Support for copybooks referenced using any of the following statements - COPY, -INC, ++INCLUDE, EXEC SQL INCLUDE.
  - Support for editing copybooks in the expanded copybook view (except for copybooks shown with replaced values or such for which the user does not have write access).
  - For copybooks referenced using a COPY... REPLACING statement, you can show either the original contents of the copybook file or show its contents with replaced values. The expanded copybook view with replaced values is read-only which is indicated by a yellow background color.

    Note: You can configure the background color of the read-only view from **Tools > Options > Environment > Fonts and Colors**, and changing the background color for the **COBOL Read-Only Background** item.
  - **Go To Definition**, **Find All References**, navigation using the drop-down types and members, and navigating to an error from the **Error List** window all find results or locate information in the expanded copybook view.

- Support for Intellisense and code snippets when editing copybooks in the expanded copybook view.
- Support for the Visual Studio outlining feature (collapsing and expanding regions of code) within the expanded copybook view and for regions that include expanded copybooks.

Note: The following limitations apply to the expanded copybook view:

- Bookmarks are not supported in the expanded copybook view. You can add bookmarks in copybooks that are opened as standalone files.
- It is not possible to edit copybooks referenced through a COPY... REPLACING statement in expanded copybook view when the view shows the replaced values. In this case, the expanded copybook view is read-only.
- The read-only status of expanded copybooks is not preserved when you close the project. Next time you open the project, the copybook is shown in the Expanded Copybook view but not as read-only.
- If you used the OF or IN phrases with a COPY statement, it is not possible to show the corresponding copybook in the Expanded Copybook view or open it using the **Open "*CopybookName*" in New Window** command if the copybook is stored in a library file (where you have used the COPYLBR Compiler directive).
- Navigating to items in the expanded copybook view is not supported from the **Class View** or using the **Navigate To** command.
- References to file locations in the output windows such as pointing to the build output, or in the results received by **Find in Files**, do not point to items in the expanded copybook view.
- Selecting the currently selected statement in the call stack in an expanded copybook may show the actual copybook. You can use the **Show Next Statement** command of the debugger to return the actual expanded copybook position.
- The Breakpoints window always shows the line number of a breakpoint within the file containing it, not the line number within expanded copybook view.
- The **Call Stack**, **Threads** and **IntelliTrace** windows all display line numbers corresponding to the file containing the statement, not the line number within expanded copybook view.
- If you move the caret to an invalid position in the expanded copybook view, the **Set Next Statement** command will display an ‘Unable to set the next statement to this location’ error message and the copybook will be opened as a separate document.
- When stepping through managed code, if the next statement to step onto is within the same class or program, and on the same line number, but in a different source file, the debugger steps over that statement. For example, if you try to step from line 10 of `Program1.cbl` to line 10 of `Copybook1.cpy`, the debugger steps over the statement in the copybook.

ACUCOBOL-GT Compatibility

The following enhancements are applicable to Enterprise Developer:

- 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 Enterprise Developer in this release:

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

Application Configuration

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

COBOL Source Information (CSI)

COBOL Source Information (CSI) provides a quick and easy way of providing you with information about your program when you are working on it. You enter a query in the CSI query control and CSI returns the results of the query in the Micro Focus Code Analysis window.

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

Converting Net Express Projects

Conversion of Net Express projects has been enhanced. You now use the Visual Studio conversion wizard to import and convert Net Express projects (.app files) to Visual Studio solutions.

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 Enterprise Developer. 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

Enterprise Developer 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

Enterprise Developer 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*.

Procedural Multi-Output Project type

A new .NET managed project template, Procedural Multi-Output Project, is now available. The project compiles procedural programs to individual managed assemblies or executables as opposed to the standard managed project templates that produce a single assembly. This project type is useful when you are moving existing procedural applications that consist of multiple programs to Enterprise Developer and to managed code. When calling a subprogram, the COBOL run-time system can locate the assembly using the name of the called program which will match the assembly file name. In this case, the assembly does not have to be preloaded. See *Types of COBOL Project Templates* for more details and for a list of the limitations of this project type.

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 Enterprise Developer.

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.

Searching in Copybooks

The search in copybook files has been enhanced. Visual Studio's **Find in Files** now enables you to perform a search in all copybooks - the ones that are part of the project, and in the ones that are found in the paths defined on the **Dependency Paths** tab in the project's properties. To do so, set the search scope in the **Find and Replace** dialog to **COBOL Project Copybook Paths**.

COBOL Editor Enhancements

Navigating in the editor is now more similar to the editor navigation in Net Express. You can configure things like word wrap and indentation, the movement of the cursor using the **Home** and **End** keys, the style of the ruler and the colors of the text and the margins:

Smart editing mode

The default mode in the editor is now Smart edit mode. It controls the word wrapping and the indentation in the different COBOL areas. You can configure the Smart edit mode from the **Tools > Options > Text Editor > Micro Focus COBOL > Margins** page. To toggle the mode, click in the COBOL toolbar.

Home and End keys navigation

You can configure how the **Home** and **End** keys move the cursor in the editor on the **Tools > Options > Text Editor > Micro Focus COBOL > Margins** page. To toggle the mode, click in the COBOL toolbar.

Configuring the ruler

You can specify whether to display the horizontal ruler in the editor and also choose its style, a default style or a mainframe style for which the ruler indicates COBOL area A/B, from **Tools > Options > Text Editor > Micro Focus COBOL > Margins** and check **Show ruler**.

Support for SOA

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

Enterprise Developer 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 Enterprise Developer, 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 Enterprise Developer Personal Edition license does not support the IMTK functionality.

Upgrading from Net Express to Enterprise Developer

A new section in the product help, *Upgrading from Net Express to Enterprise Developer for Visual Studio 2017*, provides guidance on how to move existing applications either developed or debugged in the Net Express IDE into the Visual Studio IDE.

WCF REST Service Application projects

Enterprise Developer now provides a project template for creating COBOL WCF REST Application services. The requirements are - Visual Studio Professional, Premium, or Ultimate (Visual Studio Shell does not support WCF applications), and versions of the .NET Framework 4.0 and later. For more information about WCF and REST, check Microsoft's MSDN.

Microsoft Windows Azure

Enterprise Developer for Visual Studio now enables you to create, build, debug, run, and package COBOL applications for use with Windows Azure. It provides:

- Project template for a COBOL cloud service. This creates a solution containing a project for the cloud service and projects for the web and worker roles you specify.
- Demonstration projects, showing a web and worker role in COBOL, with the business logic remaining in unchanged COBOL programs.
- Help. This includes a tutorial that shows how to create a simple cloud service for COBOL and how to deploy and run the service as an off-premises Microsoft Azure instance.

This functionality was previously available as an AddPack but is now integrated into Enterprise Developer.

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 Enterprise Developer 2.1 Update 1

Enterprise Developer 2.1 Update 1 provided enhancements in the following areas:

- *Compiler Directives*

- *DB2 ECM*
- *Copybooks Enhancements*
- *Enterprise Server*
- *HCO for DB2 LUW*
- *HCO for SQL Server*
- *OpenESQL Assistant*
- *PL/I Support*
- *SQL Option for DB2*
- *Debugging Enhancements*

Compiler Directives

You can now set SQL and PL/I-related 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 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

Copybooks Enhancements

For variables in a copybook that are modified by COPY… REPLACING statements in your code, the Autos window displays all values defined in the source code. When there are multiple COBOL programs in your project that perform a COPY… REPLACING in a copybook, the Autos window only uses the replacing values found in the first COBOL program. In addition, a new command, **Open *copybook* with replaced values**, is now available from the editor for the copybooks that the COPY... REPLACING statements modify.

Enterprise Server

The following new features and enhancements are available:

Clustering	Enterprise 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 Enterprise 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	Enterprise Server now allows clients and servers to identify themselves through X.509 certificates and participate in SSL-enabled conversations.

HCO for DB2 LUW

- Support for 64-bit 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.

HCO for SQL Server

- Data Transfer tool enables you to:
  - Specify the number of error rows to identify before committing them to the log file.
  - Transfer tables in one SSIS package instead of one package per table (optional).
  - Execute DDL for tables
  - Transfer DDL
- The Generate DDL tool provides you the option of using SQL Server database collation or preserving mainframe DB2 collation with SQL Server.
- Migration of DB2 temporary tables has been enhanced.

OpenESQL Assistant

Support for generation of PL/I SQL queries and copybooks.

PL/I Support

- Support for executing PL/I programs using IBM MQ Series software
- DEFINE ALIAS support for the Open PL/I Compiler for XML
- Support for TSO/IDCAMS
- Improved file I/O speed when using the PL/I language to read or write sequential, indexed and relative record files
- To enable PL/I for CICS, JCL, IMS you only need to set `ES_PLI_SUPPORT=Y`. You no longer need to set `ES_PL1_MFFH_JCL=Y`.
- Support for character picture data types A,X
- Support for %XINCLUDE (macro preprocessor and compiler)
- Support for %OPTION (macro preprocessor)
- Tolerance for source code containing INLINE and ABNORMAL attributes.
- Support for the TINY and HUGE built-in functions
- New PLICTF and PLICTFHX built-in functions
- Support for the PARAMETER attribute
- Support for the NONINIT attribute

SQL Option for DB2

- Background parsing support
- Enhanced XDB engine LOB support
- UTF-8 support for client-side expressions
- Support for indices on expressions

Debugging Enhancements

For variables in a copybook that are modified by COPY… REPLACING statements in your code, the Autos window displays all values defined in the source code. When there are multiple COBOL programs in your project that perform a COPY… REPLACING in a copybook, the Autos window only uses the replacing values found in the first COBOL program. In addition, a new command, **Open *copybook* with replaced values**, is now available from the editor for the copybooks that the COPY... REPLACING statements modify.

What was New in Enterprise Developer 2.1

Enterprise Developer 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*
- *IMS Support*
- *Managed code enhancements*
- *OpenESQL*

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.

Associating file extensions with the COBOL language

This release includes enhancements to the way you associate file extensions and extensionless files with the COBOL language.

If you are importing existing COBOL applications into Visual Studio, it is recommended that you create associations with COBOL within the IDE for any extensions that are not traditionally used in COBOL.

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 on the Application page in your project's properties.
- A new native COBOL project template, Micro Focus INT/GNT.
- 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 Enterprise Developer projects that compile to `.int` and `.gnt` code.

IMS Support

Mainframe projects contain categories for IMS resources (`.dbd`, `.mfs`, and `.psb` files), in which files with these extensions are automatically put when you add them to the project. Each file type has its own editor that colorizes tokens to aid development. Building a project containing these file types automatically generates DBD, PSB, and MFS files.

Managed code enhancements

Delegates and Events

Delegates and events are now implemented on the JVM platform.

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

Handling Invalid Numeric Data

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

Resolving Types

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

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

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

Specifying Properties

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

OpenESQL

JDBC

JDBC has been enhanced to support two new directives:

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

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

ODBC

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

SQL Compiler Directive Options

OpenESQL has been enhanced to support the the following new SQL compiler directive options:

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

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

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

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.

What was New in Enterprise Developer 2.0

Enterprise Developer 2.0 provided enhancements in the following areas:

- *Building Projects to Multiple Output Files*
- *Compiler Directives*
- *Debugging Enhancements*
- *JVM COBOL File Handler*
- *Library Routines*
- *Managed COBOL Language Features*
- *Data Access*
- *Run-Time Tunables*
- *Samples*
- *Vision Data File Searching*
- *XML Support*

Building Projects to Multiple Output Files

You can now build your native Link Library projects to multiple output files. To configure this, you need to go in the project properties and set **Output To** to **Multiple Libraries**.

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.

Debugging Enhancements

Enterprise Developer 2.0 provides the following enhancements to debugging:

Core dump files

On Windows, the core dump files created by this product are Microsoft minidump files. If you have a mixed-language application, you can use the minidump file to debug the other languages involved.

Debugging native COBOL Link Library projects

You can now debug native COBOL link library projects that get built to a single .dll file.

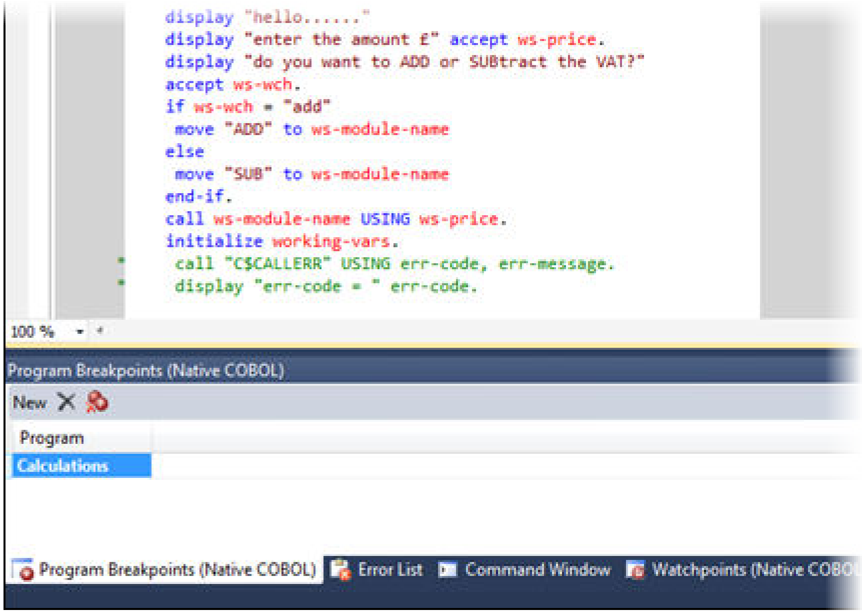
Program Breakpoints

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

Use the **Program Breakpoints (Native COBOL)** window to add program breakpoints to your applications. When a program breakpoint is set the application breaks whenever the program or one of its entry points is called.

In the **Program Breakpoints (Native COBOL)** window, click **New** and type the name of the source file, without the extension. For multi-program source files, to set a break for a sub-program, use its program-id.

To display the **Program Breakpoints (Native COBOL)** window, click **Debug** > **Windows** > **Program Breakpoints (Native COBOL)**.



```
  display "hello......"
  display "enter the amount £" accept ws-price.
  display "do you want to ADD or SUBtract the VAT?"
  accept ws-wch.
  if ws-wch = "add"
   move "ADD" to ws-module-name
  else
   move "SUB" to ws-module-name
  end-if.
  call ws-module-name USING ws-price.
  initialize working-vars.
*  call "C$CALLERR" USING err-code, err-message.
*  display "err-code = " err-code.
```

100 %
Program Breakpoints (Native COBOL)
New

Program
Calculations

Program Breakpoints (Native COBOL)
Error List
Command Window
Watchpoints (Native COBO

Step out of OSVS perform statements

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

Support for debug attaching to programs launched by services

Enterprise Developer now supports debugging of native CGI applications by waiting for attachment on a directory.

Wait for debuggable attachment option for native COBOL

Wait for debuggable attachment on the Debug page in the properties of native properties has been enhanced. You can now specify one the following options for it - Wait for any program, Wait for directory, and Wait for ID.

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.

Data Access

Enterprise Developer version 2.0 provides the following enhancements:

- Improved IDE integration with SQL directives - now supports handling of deprecated and removed directives. Also supports filtering of the choices offered to the user by product type, project type, and platform.
- OpenESQL has been enhanced and it now:
  - defaults to optimal performance
  - supports 64bit ODBC across all platforms
- The OpenESQL ADO.NET managed run-time has been redesigned so it now works faster. It also supports multiple result sets with stored procedures.

Run-Time Tunables

Enterprise Developer 2.0 provides the following new tunables:

- printer_raw_redirection - use this to redirect WRITE statements through the Windows print spooler as RAW data types.
- subsystem_cancel_mode - use this to override the default cancel mode when you use the CBL_SUBSYSTEM library routine to cancel a subsystem.

Samples

The following new samples are now available:

- CGI demonstrations:
  - Complex CGI application - demonstrates how to use native COBOL to create CGI programs which accept data from a form on a Web page and then redisplay that data in another Web page
  - Simple CGI application - demonstrates how to use native COBOL to create CGI programs which accept data from a form on a Web page and then redisplay that data in another Web page
- The following samples have been added to the COBOL for .NET section:
  - Collections - demonstrates the new LIST and DICTIONARY collection types
  - Local Variables - shows how to declare data items in the procedure division in the DECLARE, PERFORM and TRY statements
- The code in the Properties sample in the **COBOL for .NET** section of Samples Browser has been enhanced to use the new PROPERTY-ID syntax. The sample also includes a sample program for Indexers which illustrates the new INDEXER-ID syntax.

Vision Data File Searching

Enterprise Developer 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

XML Support

The IBM-style XML GENERATE syntax is now supported in .NET managed COBOL.

What was New in Enterprise Developer 2010

Enterprise Developer 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 Enterprise Developer 2010 R4 Update 2

Documentation for the Dialog System AddPack

Documentation for the Dialog System AddPack is now available and is integrated with the Enterprise Developer documentation.

This documentation describes the AddPack, which enables you to modernize Dialog System applications within Enterprise Developer. You migrate an application to Enterprise Developer and from there you can run the application without change, or modernize it over time.

The documentation describes some modernization techniques such as:

- A Windows Forms form replacing a Dialog System dialog, where the form can contain .NET controls. See the Customer + .NET WinForm sample `CustomerWinForm.sln`.
- A Windows Forms control wrapped as an ActiveX control and used on a Dialog System dialog. See the Customer + .NET GridView User Control sample `custgrid.sln`.
- A WPF user control hosted by a Windows Forms user control, which is then exposed as ActiveX ready for use by Dialog System. See the Customer + .NET WPF GridView User Control sample `CustGridWPF.sln`
- A .NET managed code application interacting with Dialog System as native COBOL `.dll`. See the Managed Customer sample `ManagedCustomer.sln`.

OO COBOL Class Library Reference

Help for the following OO COBOL class libraries is available:

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

The Help is available in the file `nxrclr.chm`, which is installed in the Help folder of your installation. The default location is`%ProgramFiles(x86)%\Micro Focus\Enterprise Developer\Help`.

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

Net Express Project Import Wizard

Enterprise Developer for Visual Studio includes a Net Express Project Import Wizard that converts NetExpress projects into Visual Studio solutions. The wizard analyzes the Net Express project file and its configuration settings, creates one or more Visual Studio native projects based on this information, imports the existing source code into them and sets the appropriate directives. When the process is complete, it can optionally display a detailed conversion report.

Features Added in Enterprise Developer 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 Enterprise Developer 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.

ADO.NET Runtime Improvements

The OpenESQL ADO.NET runtime has been re-engineered, offering several advantages over previous versions, providing:

- Better performance **-** the new runtime is much faster and can as much as double your performance, depending on the specifics of your application.
- Support for multiple run unit processing.
- Support for both client applications and SQL CLR stored procedures.
- For stored procedure applications:
  - Support for input/output parameters defined as COBOL data types, in addition to the previously supported .NET data types
  - Support that enables stored procedures to return more than one result set.
  - Static SQL Support for the DIALECT=MAINFRAME Compiler directive.
- For client-based applications:
  - Support for dynamic SQL that uses an SQLDA structure to pass parameter and result host variables, specifically including the following embedded SQL statements:
    - PREPARE **-** For example, the following statement passes a prepared statement into the SQLDA from a host variable:

      ```
      EXEC SQL
           PREPARE stmt1 INTO :sqlda FROM :stmt-buf
      END-EXEC
      ```
    - EXECUTE **-** For example, the following statement executes a prepared statement using a previously constructed SQLDA structure:

      ```
      EXEC SQL
           EXECUTE stmt1 USING DESCRIPTOR :sqlda-structure
      END-EXEC
      ```

    - OPEN - For example, the following statement opens a cursor using a previously constructed SQLDA structure:

      ```
      EXEC SQL
           OPEN c1 USING DESCRIPTOR :sqlda-structure
      END-EXEC
      ```

  For complete information on using these statements, see the *PREPARE*, *EXECUTE*, and *OPEN* topics.
- Additional support for result set processing that allows you to receive more than one result set back from a stored procedure.
- Support for applications built to target .NET Framework versions 2.0, 3.0 and 3.5, in addition to .NET Framework version 4.0.

Important: To take advantage of these improvements you must recompile your applications with the DBMAN=ADO Compiler directive. However, if you want to continue using the functionality available in previous releases, we now provide the DBMAN=OADO directive to use in place of the DBMAN=ADO directive. DBMAN=OADO provides the same functionality as was available with previous versions of the DBMAN=ADO Compiler directive, and has 100% backward compatibility with applications designed and developed with Visual COBOL R3.

Debugging Enhancements

The ability to load core dump files in Visual Studio has been added. This feature works with native COBOL only.

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.

HCO for SQL Server (HCOSS)

Host Compatibility Option for SQL Server (HCOSS) is a new component in Enterprise Developer for Visual Studio 2017. However, it was previously available in the Enterprise Developer - Enterprise Server for .NET product. The following are HCOSS features that are new since its previous release in Enterprise Developer - Enterprise Server for .NET:

- HCOSS supports client-based applications compiled with the DBMAN=ADO and DIALECT=MAINFRAME compiler directives when the application targets .NET Framework versions 2.0, 3.0 and 3.5, or .NET Framework version 4.0, but only when the application does not contain dynamic SQL that uses an SQLDA structure.
- HCOSS supports dynamic SQL that uses the SQLDA structure in client-based applications compiled with the DBMAN=ADO and DIALECT=MAINFRAME compiler directives, but only when the application is built to target .NET Framework version 4.0.

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 Enterprise Developer 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 Enterprise Developer, 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 Enterprise Developer.

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

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

Significant Changes

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

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

Significant Changes in Enterprise Developer 4.0

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

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

- *Application Server JCA Support*
- *CICS Support*
- *Code Coverage*
- *Codeset Support*
- *Common Communications Interface*
- *Communications Server*
- *Compiler*
- *Data Tools*
- *Documentation*
- *Enterprise Server*
- *Enterprise Server Auditing*
- *Executables require relinking*
- *File Handling*
- *HCO for Microsoft SQL Server*
- *JCL Support*
- *Mainframe Access*
- *MF Server Administrator (GUI)*
- *Micro Focus Directory Server*
- *PL/I Compiler*
- *Run-Time System*

Application Server JCA Support for Enterprise Server

- With this release, all CICS-related Java artifacts have been delivered using a new directory structure under the `javaee-ccl` directory of your Enterprise Developer installation directory. The `javaee-ccl` directory structure mimics the structure of the `javaee` directory. For example, `mfccl.jar` is no longer located in the `bin` subdirectory, but can be found in the `javaee-ccl\common` directory.

CICS Support

Back to the list

- Client ECI timeouts now propagate CTG requests to ABEND AITH the ECI server program.
- The update resource definition file utility, CASRDTUP, now supports the DELETE command. Note that the REMOVE option is not currently supported.

  2885020 (1108004)

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)
- 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)
- CICS Web services has been updated to include the ES_TRANCLASS_CWI environment variable, which enables you to prioritize and limit Web requests in a TRANCLASS-enabled region. When the ES_TRANCLASS_CWI environment variable has been set to either TCPIPSERVICE (default) or URIMAP, a TRANCLASS-enabled region processes the elected transaction according to its priority and associated TRANCLASS.

  Note: When using CICS® Transaction Server for z/OS®, Version 3 Release 1, and IBM CICS initiates the ATTACH transaction followed by the ALIAS transaction for a single Web request, this is intended to enable the application of different limitations and priorities on these transactions. Enterprise Server does not support this. Instead, a TRANCLASS-enabled region processes the requests as a single ATTACH transaction, and uses the ALIAS transaction for security checks.

  3126381 (1111349)
- Web access to the Enterprise Server Console Log and Communications Server Log is now restricted when the enterprise server 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 the resources do not exist, the default behavior is to allow 'read' access on entry of a valid username and password.

  An example of the new resource class and resource definitions can be found in the `es_default_ldap_msuser.ldf` file, which is located in the `bin` sub-directory of your product directory.

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.

Executables require relinking or recompiling

- Due to an internal change in version 4.0 of your product, you must at least relink any executable programs compiled prior to this version, to make them compatible with the latest run-time system. However, a full recompilation of your source code is the recommended action, to allow your executables to benefit from the product's latest programming and performance enhancements.

  Relinking an executable without recompiling means using the original object code with the `cbllink` utility. Original object code is typically the binary file output (usually containing the `.obj` extension) produced during the original compilation process. An application can contain one or more binary files.

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)

- For .NET applications, set XFHLOG=DEFAULT to create the XFHLOG file in the current directory; otherwise it is created in C:\ProgramData\Micro Focus\[ED/VS]\[Release]. From release 4.0 onwards, this variable also has an effect for native applications.

  3115116 (1110323)
- 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)

- The TO parameter is now supported for TOT|TOTAL, MIN, MAX, and AVG in the TRAILER3 trailer item.

  3135298 (1112194)
- The OUTFIL statement of a SORT operation now supports the NULLOFL parameter.

  3110364 (1109822)
- The MFJSORT OUTFIL emulation now supports the VLTRIM parameter.

  3107105 (1109820)

HCO for Microsoft SQL Server

Back to the list

- The HCOSS Manage Connections tool was incorrectly limited to using the default CCSID for mainframe DB2 connections, causing problems when mainframe database did not use the default CCSID of 37. This has been corrected by the addition of a text box that enables the setting of a specific CCSID value for the connection.

  3130137 (1112180)
- When using the DIALECT=MAINFRAME setting, the OpenESQL STRICTPARSE SQL compiler directive option is now turned on by default for ODBC (DBMAN=ODBC). This ensures strict compliance with z/OS DB2 syntax.

  If your applications require the OpenESQL preprocessor to behave as it did in an earlier release, compile using NOSTRICTPARSE.

JCL Support

Back to the list

- You can now configure MFFTP to not process trailing blank spaces for a GET command, by setting the new control variable MFFTP_PROCESS_TRAILS_ONGET=FALSE and setting the LOCSTAT value TRAILINGBLANKS=FALSE.

  3138375 (1112486)
- ICEGENER now does not copy the RECFM from SYSUT1 to SYSUT2 if SYSUT2 RECFM is not specified in JCL and when SYSUT1 is VSAM. It defaults to F/FB if SYSUT2 is non-VSAM, or V/VB if SYSUT2 is VSAM.

  3131268 (1111805)
- An error in identifying the IDENT in IEBGENER when the record included high values at the start of the record has been corrected.

  3109443 (1109759)
- FOR ED 2.3.2 HF7: The following changes have been made to the MF_UCC11 environment variable: MF_UCC11=Y - when running a job with DISP=(NEW,CATLG), it will not result in job failure caused by the data set already being cataloged. MF_UCC11=YR - when restarting a job with DISP=(NEW,CATLG), it will not result in job failure caused by the data set already being cataloged. For ED 3.0 HF1: The following changes have been made to the MF_UCC11 environment variable: The default behavior of MF_UCC11=Y has changed: DISP=NEW is changed to DISP=MOD only when the job is restarted. MF_UCC11=YA reverts to the old behavior of changing DISP=NEW to DISP=MOD even when the job is first run.

  3107409 (1109722)
- The introduction of a new environment variable, ES_OLD_DYN_PDS, has changed the resulting behavior of dynamic PDS members when specified in a JCL DD statement. Default behavior (ES_OLD_DYN_PDS=N) no longer creates physical files or catalogs the PDS members when specified in a JCL DD statement. Set this variable to Y to enable the previous default behavior of creating the physical files and cataloging PDS members when specified in a JCL DD statement.

  2845176 (1102528)
- To prevent unintended deletion of a GDG Bias, a GDG in use by one job may not be accessed by a second job until the first job has completed.

  2816745 (1101569)

Mainframe Access

Back to the list

- Calling mfmon2.exe with the following options has the given effect:

  1. If you do not specify "EXTENSION" or specify "EXTENSION=" it will result in files downloaded without an extension.

2. If you specify "EXTENSION=*" you will get an extension that matches the last part of the data set name. For example, if the source data set is a.b.c.d then the file extension will be .d.
3. Everything else results in the extension specified. For example, "EXTENSION=txt" will result in files downloaded with a .txt extension.

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)
- 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.

PL/I Compiler

Back to the list

- Previously, storing PL/I named constants in big-endian form was incorrect and impaired some instances of restricted expression evaluation. In ED 4.0, named constants are stored in native form when the -bigendian compiler option is in effect.

  3150931 (1113613)
- Open PL/I now supports the FLUSH statement.

  3134294 (1112117)
- The %PROCESS statement now includes new compile-time options, -pporder and -options. See the product help for more information.

  3107667 (1109861)
- Pointer subtraction in 64-bit mode now works as expected.

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 Enterprise Developer 3.0

Enterprise Developer version 3.0 includes significant changes in the following areas:

- *CICS Support*
- *Compatibility AddPack*
- *Compiler*
- *Documentation*
- *Enterprise Server*
- *Enterprise Server for .NET*
- *File Handling*

- *IDE*
- *IMS Support*
- *JCL Support*
- *JES-enabled Enterprise Server Regions*
- *Micro Focus Batch Scheduler Integration*
- *Micro Focus Directory Server*
- *OpenESQL*
- *PL/I Support*
- *Reserved words*
- *SQL: HCO for SQL Server*
- *HCO for DB2 LUW*

CICS Support

- The value specified in the LENGTH field of the SEND TEXT command is now honored when the target terminal has its SOSI or PS attributes switched on. This means that trailing spaces will no longer be removed for SOSI/PS terminals.

  2884877 (1107983)

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 Enterprise Developer for Visual Studio. The run-time components are installed as part of Enterprise 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)

Enterprise Server for .NET

- A problem that caused an RTS 173 error while executing a batch job step that performed a COBOL sort (as opposed to EXEC PGM=SORT)when the program terminated using STOP RUN instead of GOBACK has been fixed.

  3105738 (1109359)

IDE

- Disabling CICS support from the properties of a file in your project within the IDE now correctly sets the NOCICSECM Compiler directive on that file.

File Handling

- 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)
- Under Mainframe emulation, when an empty virgin VSAM file is opened for input or for input/output, a file status of 35 is returned, regardless of the access mode specified.

  2692826 (1093604)
- New PARM option SIGN-EBCDIC documented for MFJSORT.

  2880037 (1107256)
- The MFJSORT ICETOOL emulation now supports the USING parameter in the SELECT operator.

  2821176 (1099529)
- An empty, space-filler record is no longer produced at the end of an ICETOOL listing file.

  (622789)

IMS Support

- Inconsistent results returned to a GET call issued from a PROCOPT=GO application when segments within the same database record were updated and locked by another session. Symptoms included an invalid (GE) status code or the wrong segment returned to a GN call. This has been fixed.

  2862879 (1104899)
- The IMS control page is now secured according to the IMS* entity in MFESMAC.

  2877405 (1107099)

JCL Support

- An emulation of the LE run-time option CBLQDA has been implemented in the form of the environment variable MF_CBLQDA. The default is OFF, but when set ON, QSAM files can be dynamically created.

  3104665 (1109251)
- A new event call to the JCL user exit (MFJUXIT) has been included. The event 'job-queued' (event code 27) occurs when a JCL job has been parsed and put on either the Input Ready queue or the Input Hold queue.

  2990395 (1108734)
- When reading commands, MFFTP now checks the charset (EBCDIC or ASCII) of each concatenated INPUT dataset to determine if it should be translated. The output is written in the charset which appropriate for the dataset.

  2876407 (1106795)
- The PHYSFILE security setting will now be checked when a dataset entry is inserted into the catalog, or when a dynamic PDS member is created.

2857819 (1104844)

JES-enabled Enterprise Server Regions

⚠ **Important:** Existing JES-enabled Enterprise Server regions must perform this upgrade to continue using JES functionality. If you start a region without upgrading, the following message is displayed in the console log, and JES functionality is disabled:

```
JES000058E JES validation - Product requires new spool control files. Run
spool conversion utility, splconv, to create them.
```

In this release, the infrastructure of the JES spool repository has been upgraded in order to improve performance (see the *Enterprise Server* section of *What's New* for more details). For each JES-enabled region, before you can submit any jobs to JES, you must run the spool conversion utility; any newly created regions will automatically be configured to use the new spool files.

The conversion process converts the information in the existing spool files to a set of new files required under the new system. There should be no loss of data in moving to the new system.

⚠ **Important:** After converting to the new JES spool infrastructure, we recommend that you do not employ the increased range of job numbers until you have thoroughly tested the new infrastructure. Reverting back to the previous JES spool files is possible, but requires that you remove all job entries using more than 5 digits, for which there is no easy process.

To run the spool conversion utility

1. At the Enterprise Developer command prompt, set the following environment variables, as required:

Variable	Description
MFSYSCAT	The location and name of the current JES catalog. This is mandatory. If multiple catalog files are being used (that is, one or more user catalogs are defined), this environment variable should point to the catalog where JES spool datasets are cataloged.
MVSSPOOLDIR	The location of the current JES spool files. This is only mandatory if the location is different to the location of the JES catalog.
NEWSPOOLDIR	The location for the resulting JES spool files (that is, the SPL* files if running the conversion, or the old-style spool files if running the regression). This is only mandatory if the location is to be different to the location of the existing spool files.
MFSYSCAT_ERRS	The number of catalog errors that will be tolerated during the conversion process before the conversion is terminated. This is optional, and defaults to 100. If the process is terminated, the following error is produced: MVSCATIO Error limit exceeded = *nnnn* (where *nnnn* is the error count)

2. Type `splconv`, then press **Enter**.

   The conversion process runs, and details of the files and records converted are displayed to screen.
3. If MVSSPOOLDIR and NEWSPOOLDIR were set to different locations prior to running the conversion, you must now set MVSSPOOLDIR to the value of NEWSPOOLDIR.

Note: If you do not perform this step, you will receive the following error when you start your region, and JES fails to initialize:
JES000058E JES validation - Product requires new spool control files. Run spool conversion utility, splconv, to create them.

4. You can now start the region identified by the variables, and begin to submit jobs.

Reverting to the previous JES spool repository

If you plan to downgrade to an older version of your current product, any enterprise server region that has been converted or created with the current product will not be compatible. Before you can use them, they must be reverted to using the old JES spool repository.

Use this process if you are wanting to use a JES-enabled region in a previous version of the product:

1. At the Enterprise Developer command prompt, set the environment variables listed in the *To run the spool conversion utility* process, as required.
2. Set an additional variable, SPLREGR_TARGET, to `PRE30`.
3. If the region has started to use 6- or 7-digit job numbers (an enhancement new to version 3.0 of the product), then before you run the regression, you must remove the jobs that are using these numbers, and ensure that the MF_MVSJOB environment variable (or the `SPLJNO.dat` file if the variable is not used) does not specify an upper limit greater than 5 digits.

   Note: If using the MF_MVSJOB environment variable, to revert to 5-digit job number support, set MF_MVSJOB=<*low number*>+<*high number*>, where *high number* is less than 0099999, and ensuring that the + sign is used to separate the low and high values.
4. Type `splregr`, then press **Enter**.

   The regression process runs, and details of the files and records converted are displayed to screen.

   You can now use the region identified by the variables in previous versions of Enterprise Developer.

Exporting the spool files to Enterprise Server for .NET

If you plan to export a region to Enterprise Server for .NET, any enterprise server region that has been converted or created with the current product will not be compatible. Before you can use them, they must be reverted to using the old JES spool repository.

Use this process if you are exporting a JES-enabled region to Enterprise Server for .NET:

1. At the Enterprise Developer command prompt, set the environment variables listed in the *To run the spool conversion utility* process, as required.
2. Set an additional variable, SPLREGR_TARGET, to the appropriate value:

   DOTNET5

   Create spool files for use on Enterprise Server for .NET, using a maximum of 5 digits for job numbers.

   DOTNET7

   Create spool files for use on Enterprise Server for .NET, using a maximum of 7 digits for job numbers.
3. Type `splregr`, then press **Enter**.

   The regression process runs, and details of the files and records converted are displayed to screen.

   You can now use the region identified by the variables in Enterprise Server for .NET.

Micro Focus Batch Scheduler Integration

- MFBSI no more requires that the name of the last folder specified in the path in MFBSI_DIR matches the name of the target Enterprise Server region. It is now possible to specify any directory path in MFBSI_DIR. You must use the same value for MFBSI_DIR in both the Enterprise Server configuration and for MFBSIJCL execution. When the name of the last folder specified in the path in MFBSI_DIR does not match the name of the target Enterprise Server region, you need to specify the region name in one of the following ways: 1. From the mfbsijcl command line, using the parameter

REGION=ESRegionName. 2. In an MFBSI configuration file, in a line containing REGION=ESRegionName. You can specify this either in the default configuration file, mfbsi.cfg located using the path specified by MFBSI_DIR, or using a custom location or configuration file and the following parameter: CFG=[path]/name[.ext] See your product help for more details.

2869805 (1105939)

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 both ADO.NET (DBMAN=ADO) and ODBC (DBMAN=ODBC). For ADO.NET, this reduces cursor memory consumption thereby providing optimal performance. This also ensures that, for ODBC, 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.

PL/I Support

- PUT output-items of type POINTER are now output as a hexadecimal string rather than a fixed bin integer value.

  (623184)
- Enterprise Developer 3.0 includes some changes that result in a significant performance improvement when assigning a null character string '' to arithmetic data items, pointers, and structures. In versions of the product earlier than 3.0, pointer = ''; assigned the value sysnull(). This has been fixed and null() is now assigned. See the product help for details about the -setnull compiler option.

  (627517)
- Previously, if writing output to a CTLASA (PRINT) file with the optional LINE(x) parameter having a value of x that was less than the current line, PL/I did not generate ASA Control characters to end of page and emit the output on the first line of the new page. This no longer occurs.

  2871278 (1106114)

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: HCO for SQL Server

- The HCOSS Assess Application Migration tool now enables you to view the original DB2 SQL and the converted TSQL together in both the HCOSS tool itself and in any exported CSV files.

HCO for DB2 LUW

- 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 Enterprise Developer 2.3 Update 2

Enterprise Developer version 2.3 update 2 includes significant changes in the following areas:

- *Enterprise Server*
- *Compiler*
- *MF Directory Server*
- *Monitoring and Management*
- *JCL Support*
- *PL/I Support*
- *Run-Time System*

Compiler

- The ILPINVOKE directive is now allowed only in an initial $SET. This is in line with similar directives like ILREF and ILTARGET. Previously, the ILPINVOKE directive was allowed in other positions in the source code, but could later result in 'Insufficient memory' errors.

  2860347 (1104422)
- 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)

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)

JCL Support

- A printer exit can now evaluate the spool copybook field lk-dest-source-is to find the source of the value in the spool DEST-NODE and DEST-USERID fields, to determine if it is from a SYSOUT 'DEST=' value or from a /*ROUTE PRINT value.

  2852302 (1103304)

PL/I Support

- A problem with PUT STRING when compiled with the -bigendian option in effect has been fixed. Note that when upgrading to ED 2.3 U2 you must recompile any program using PUT SPRING.

  2857374 (1104039)
- Previously, if using the -incafter option of the macro preprocessor to infuse a %INCLUDE with macro logic, you were required to schedule a second macro pass to preprocess the contents of the %INCLUDE. This is no longer necessary and the macro logic will be preprocessed on the first macro pass as if it was part of the original source.
- Previously, if writing output to a CTLASA (PRINT) file with the optional LINE(x) parameter having a value of x that was less than the current line, PL/I did not generate ASA Control characters to end of page and emit the output on the first line of the new page. This no longer occurs.

  2871278 (1106114)
- Previously, if SIGPIPE was triggered when a PL/I program was in the callstack, then PL/I error handling got control and processed the ERROR. This no longer occurs. Users who rely upon catching SIGPIPE can re-enable it by adding a call to cobpostsighandler() to their code to turn it back on.

  2859980 (1104399)

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 Enterprise Developer 2.3 Update 1

Enterprise Developer version 2.3 update 1 includes significant changes in the following areas:

- *Communications Server*
- *Data Tools*
- *Dialog System*
- *Editor Writing Assistance*
- *Enterprise Server*
- *IBM Language Environment for OS/390 & VM Support*
- *JCL Support*
- *Run-Time System*
- *SQL: OpenESQL*

- *SQL Option for DB2*

Communications Server

- A new option, `[Operation] synchronous=yes` (or 1, or Yes, or y or Y) is now supported in the Fileshare listener configuration.

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.

Dialog System

- Versions of the Micro Focus Compatibility AddPack released with version 2.3 of Enterprise Developer or Visual COBOL 2.3 or with earlier versions supported the use of the "MFOLECL_NO_THREAD_INIT" environment variable. Using this variable, you could disable the default OLE Class Library COM threading initialization. This helped avoid issues that could manifest as hangs and crashes, especially on Microsoft's Windows 8.x or 10 and with applications that are a hybrid between Dialog System, OLE class library and .NET elements - see *http://community.microfocus.com/microfocus/cobol/visual_cobol/w/knowledge_base/20715.exception-occurs-when-native-dialog-system-program-calls-managed-winform.aspx*.

  In the version of the Micro Focus Compatibility AddPack released with Enterprise Developer 2.3.1 or Visual COBOL 2.3.1, this environment variable is now enabled by default and no longer needs to be set exclusively. To restore the previous behavior, use a new environment variable, MFOLECL_THREAD_INIT, and set it to Yes.

  2848875 (1102920)

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.
- Pressing TAB in the Visual Studio editor now always inserts the highlighted item in the IntelliSense list of suggestions.

Enterprise Server

- In situations where the crossregion option of the shareoptions parameter is either 1 or 2, both a CICS and a JCL job could have the same file open for IO. This has been fixed. In addition, the '-oo' (and the synonymous '-o ') option of the casfile command has been deprecated. If '-oo' is used, you will now receive a return code 4 (warning). Instead of '-oo', you need to use the '-ooi' option (open immediate). If -oo is used with a region configured for VSAM shareoption use, the '-ooi' option is now forced.

  2839272 (1101672)
- When using an external security manager to secure a region, the EXEC CICS START TRANSID API now works as expected and honors the USERID option. Previously, the started transaction would run under the authority of the user that executed the EXEC CICS START API.

  2835112 (1101201)

IBM Language Environment for OS/390 & VM Support

- When calling SETENV, SYSOUT(?) can be specified. This designates the file as a spool file but will not update its CLASS.

  2841220 (1101955)

JCL Support

- The return code from a JCL printer exit will now be checked. If it is non-zero, the dataset that was sent to the printer exit will be moved to the OUT-HOLD queue rather than to the PRINTED queue. To release the dataset, click "Release" in the ESMAC view - this will also present the dataset again to the printer exit.

  2834206 (1101079)
- Under Enterprise Server and Enterprise Server for .NET, flushing an active job previously resulted in all associated spool records for the job being deleted, but left the job running. A subsequent attempt to kill the running job using casout/seeout failed due to the job spool record no longer existing. This change causes the flush request to fail if the job is active. You must kill the active job first before attempting to flush it.
- The catalog search facility has been enhanced to provide a quicker return when using wildcards in the dataset name selection.

  2698700 (1094266)
- REXX can now be executed in VSE via IKJEFT01 and IRXJCL using in-stream data.

  2831984 (1100883)

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 Enterprise Developer 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 Enterprise Developer 2.3

Enterprise Developer version 2.3 includes significant changes in the following areas:

- *Assembler Support*
- *Building*
- *Converting Additional Directives to Projects' Properties*
- *CAS Security*
- *CAS XA Switch modules*
- *Compiler*
- *Data Tools*
- *Discontinued Support for cwgui*

- *File Handling - External File Handler*
- *IDE*
- *J2EE Connector*
- *JCL Support*
- *Mainframe Access*
- *MF Server Administrator (GUI)*
- *PL/I Support*
- *SQL - HCO for SQL Server*
- *Updated Run-Time System*

Assembler Support

- Previously, the Assembler run-time error SOC4 (Invalid 370 address abend) was mapped to the COBOL run-time error 200 (Internal logic error) which was misleading. It is now mapped to the COBOL run-time error 205 (COBRT205 Invalid mainframe pointer value (Fatal)). Note that, in most cases, you receive an Assembler SOC4 abend if the calling COBOL program was compiled without the AMODE"24" or the AMODE"31" Compiler directives.

  2806239 (1097738)

Building

- Enterprise Developer now supports Visual Studio parallel builds for COBOL projects. Parallel builds enable you to build multiple projects faster on multi-CPU machines.

  If, after upgrading to this version of Enterprise Developer, you start receiving unexpected build errors when compiling an existing multi-project solution, this may be a result of enabling support for parallel project builds. These are a couple of examples of issues that might be causing these errors:

  - Using file references to project outputs in the same solution. You need to use project-to-project references instead.

    Use **Project > Project Dependencies** to manage the project dependencies and build order within your solution.
  - A customized build process such as one that is using pre- or post- build events.

  If resolving any of these issues does not help resolve the build errors, consider disabling the parallel build support - click **Tools > Options > Projects and Solutions > Build and Run** and **set maximum number of parallel project builds** to `1`.

Note: Parallel builds are not supported with Personal Edition licensing

Converting Additional Directives to projects' properties

Starting with this release, an **Update Project Properties** dialog box might start to appear when you are opening existing COBOL solutions. The dialog box recommends converting some of the directives specified in **Additional directives** to project's properties. This is to address an issue where you might try to set file properties that differ from the project directives and the directives specified in **Additional directives** that have a property equivalent take precedence over the file properties. As a result of this you might receive unexpected build issues when building your applications.

This dialog box helps eliminate the issue by converting the directives specified in **Additional directives** to project properties.

If you do not want the IDE to perform this check, click **Tools > Options > Micro Focus > General** and uncheck **Check Additional Directives for project properties**.

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 Enterprise Developer 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 (Enterprise Developer 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 Enterprise Developer 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)
- COBDATA has no effect on compilation. The output of the Compiler is the same location regardless of whether COBDATA is set.

  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)

Data Tools

- DFCONV now returns the correct return-code; previously, it would always return 0.

Discontinued support for cwgui

The cwgui debugger is no longer available. You can now use the cw_java debugger which provides a richer graphical debugging experience.

File Handling - External File Handler

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

2795077 (1096322)

IDE

- Changes in the C compiler in Visual Studio 2015 affect the way you link COBOL object code and C object code built with that version of Visual Studio in the same executable. In this scenario, you must use the Microsoft link utility and the C runtime libraries directly from Visual Studio, rather than the Micro Focus cbllink utility, the Microsoft link utility and the libraries supplied with Enterprise Developer. You might also need to specify some additional C runtime libraries - see the Microsoft documentation for more details.

  Note that when using COBOL and C object code together, Micro Focus recommends you build and maintain the COBOL and C executables as separate projects, and use import libraries and the Micro Focus C functions for calling COBOL (see "C functions for calling COBOL" in the product help) to resolve calls between them.
- The default warning level for new COBOL projects in Enterprise Developer for Visual Studio is now "Include recoverable errors (Level E)". This also includes "Severe errors only (Level S)" and unrecoverable (Level U) errors. Micro Focus recommends you set the warning level on the COBOL page in a project's properties to "Include warnings (Level W)" where appropriate to help avoid potential coding problems.

J2EE Connector

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

  2799213 (1096684)

JCL Support

- When allocating a dataset to a step using MVSCTLBP, the values returned for the dataset attributes are those defined in the catalog. If these are unset in the catalog, then the values defined in the MVSCTLBP call are returned.

  2816179 (1098882)
- The program attributes for the SSTM job have been changed to default to a mainframe dialect, AMODE 31 program. These attributes will be used when constructing the MVS Control Blocks for the job, which means that the pointers to the sub-block areas will be in mainframe format rather than native format.

  2815695 (1098825)
- Any OUTPUT statement information passed to the printer exit now includes the correct class of that statement.

  2814249 (1098636)
- In ESMAC, when deleting the Spool, the physical spool files and the entries in casspool.dat are deleted.

  2812899 (1098412)
- New functionality has been added to the printer exit interface that enables you to delete the printer spool file after a file has been printed. Set (ws)-prn-file-disp, which is defined in cascbprn.cpy, to 1 to delete the spool file after printing. Additionally, the environment variable ES_JESYSMSG_OUTPUT=Y now correctly routes spool files to the Output queue. Previously, ESMAC still showed the job on the HELD display, even when there were no spool files HELD.

  2812899 (1099966)
- A new environment variable has been introduced - ES_JESYSMSG_RESTRICTED. JESYSMSG spool files are normally routed to HELD. This can be overridden by setting the environment variable ES_JESYSMSG_OUTPUT=Y, which causes all JESYSMSG spool files to be routed to OUTPUT. If ES_JESYSMSG_OUTPUT=Y is set but you require specific classes to be held, set the environment variable ES_JESYSMSG_RESTRICTED to a string containing the classes to be HELD.

  2812878 (1098417)

- The IEBDG PICTURE value length must match the length specified. The value can now contain any characters including quotes as on the mainframe.

  2794881 (1096221)
- Changes to the IDCAMS ALTER processing prohibit applying MGMTCLASS changes to GDGs and to a VSAM element other than a Cluster. Shareoption settings in a SMS DATACLASS are now applied to the VSAM elements correctly.

  2789939 (1095803)

Mainframe Access

Mainframe Access is now built using the IBM XL/C compiler rather than the SAS/C compiler that was used previously. This change requires you to update your production JCL procedures with reference to the supplied samples MFA, MFAS, and MFAAS. In each JCL procedure, change the STEPLIB DD statement and add a new one, CEEOPTS.

The web administration feature is no longer available in Enterprise Developer.

To use the Mainframe Access support for Endevor you must be using Endevor R14 or later. CA stabilized the Endevor API at R14, so if you upgrade from R14 to a later version you will not need to update the support module that Mainframe Access uses.

MF Server Administrator (GUI)

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

  2792382 (1096011)

PL/I Support

- For BASED variables declared using the form BASED (ADDR(x)), storage checking is now performed on the base reference “x”. This behavior requires the compiler option –nolaxbased. See the Open-PLI User guide for more information on the options –[no]laxbased and –[no]laxdefined.

  2819663 (1099293)
- The behavior of the –zp1 option has been reverted to that of before Enterprise Developer 2.2 Update 1, with an additional correction relating to Char Varying data items. For a full description of the –zp1 option, refer to the Open-PLI User’s Guide in your product help. If you are installing ED 2.2 U1 HF5 as an upgrade to ED 2.2 U1, you must rebuild any applications that are compiled using the –zp1 option after the upgrade.

  2789213 (1095636)
- Previously, when using any of the complier options -mvs, -dli, -ims, or -cics, the BINARY attribute was incorrectly applied if the FIXED attribute is specified without BINARY or DECIMAL. The default attribute DECIMAL is now applied.
- The BIGENDIAN attribute, when explicitly applied to a structure, is now propagated to its CHAR VARYING and WIDECHAR VARYING members.
- Declarations using DEFINED storage are now checked for compatible types between the DEFINED variable and its base variable. This behavior requires the compiler option –nolaxdefined. See the Open-PLI User guide for more information on the options –[no]laxbased and –[no]laxdefined.
- The PL/I Macro preprocessor has been enhanced to support the use of the ANSWER statement from within a PL/I Macro.

  2698537 (1094243)
- The macro preprocessor has been enhanced to support arrays of CHAR and FIXED variables. New macro builtin functions have been implemented (HBOUND, LBOUND, DIMENSION).

  2696129 (1094011)

SQL: HCO for SQL Server

- The return type for the HCOSS DB2 CHAR() function has changed from CHAR(255) to VARCHAR(255).

  2804069 (1097253)

Updated Run-Time System

- Enterprise 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 Enterprise Developer prior to version 2.3, you should ensure that the executable is rebuilt and packaged with the new run-time system. This does not affect mainframe subsystem projects deployed to Enterprise Server. You can rebuild from the IDE or the command line.

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

Significant Changes in Enterprise Developer 2.2 Update 2

Enterprise Developer version 2.2 update 2 includes significant changes in the following areas:

- *.NET CICS (General)*
- *Compiler*
- *Compiler Front-end*
- *Documentation*
- *Enterprise Server*
- *File Handling - External File Handler*
- *J2EE Connector*
- *MTO: CICS Communications*
- *MTO: IMS DB*
- *MTO: IMS MFS*
- *MTO: JCL MVS*
- *MTO: JCL Utils*
- *Open PL/I Compiler*
- *Open PL/I Debugger*

.NET CICS (General)

- The data file location list in the region startup file now supports both on-disk and database folders – for example, c:/project/bin/debug and sql://(dbinst)/VSAM/?folder=FOLDER1;type=folder.

  When searching for a file, CICS will check in all of the provided locations in the order specified. Files that are accessed using fileshare or the catalog are not affected by these changes and continue to work like they did before. You can configure how CICS locates the underlying data files using a new set of properties. When using the CICS Resource File Editor to explore the ResourceDefinitions.config file, FCTs now include a File Location Overrides section with the following properties:

  - Filename – specify a filename or a filename with an extension (for example, ACCTFIL.DAT) that CICS will search for instead of using the FCT name.
  - File Extension –specify a file extension. The FCT name must be used as the file name.
  - Folder – specify where CICS should search for files.
  - If this is set to the absolute path to a folder on your machine or to a database folder, CICS only searches for files in that folder and does not use the data file locations.

- If this is set to a relative path (such as "/subfolder1/subfolder2/"), CICS applies that relative path to each of the data file locations in which it searches for the file.

  Note: Previously, the Folder property was called Path and had to be set to the full absolute path to the data file. These three properties are compatible with the native Enterprise Server and can be enabled in any combination.

Previously, in order for CICS to locate a file in the database, you had to set the Format property to RDBMS and the Path property of an FCT to the name of the datastore. This is no longer required.

Instead, you need to add the datastore folder that contains the file either to the data file locations or to the Folder property in the File Location Overrides section in the FCT properties. You can still use the RDBMS format value as a hint to configure CICS to create the file (for example, using CFCR) in the database and not on disk.

If you install this release, you must modify any FCTs that were using the old Path property. For example:

If you previously had:

FCT (Name=fct1 Path="c:\folder\file1.dat")

This must be converted to: FCT (Name=fct1 FileName="file1.dat")

And the data file location, c:\folder\, must be added to the region startup file.

2780059 (1094576)

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)

Enterprise Server

Starting with this release, IMSCONFG.DAT is no longer used for configuring enterprise servers. Instead, you need to use environment variables to control the following fields:

Field	Environment Variable
LANG=PL/I PCB address lists	ES_IMS_PLI_INDIRECT_PCBADDR=D|Y|N Where: **D** Dynamic (default) use indirect PCB address list when PSB language is PL/I and main program is PL/I. **Y**

Field	Environment Variable
	Always use indirect PCB address list if PSB language is PL/I. **N** Never use indirect PCB address list.
IBM Platform	ES_IMS_IBMPLATFORM=M|D Where: **M** MVS (default) **D** DOSVS Required by some DOS/VS customers.
Secondary Index Sparse exit language	ES_IMS_SPARSE_EXIT_LANG=C|A Where: **C** COBOL **A** Assembler **Not set (default)** Sparse exits disabled

File Handling - External File Handler

- The ES_IMS_TLOG_FLUSH environment variable is now deprecated. To control TLOG flushing, use the following environment variable: ES_IMS_DB_TLOG_WRITETHRU=0|1

  0 Forces the flushing of TLOG buffers to disk on COMMIT only (default)

  1 Forces the flushing of TLOG buffers to disk on all database I/O

  To control database flushing, use the following environment variable: ES_IMS_DB_COMMIT_FLUSH=0|1

  0 Forces the flushing of database buffers to disk on CLOSE only (default)

  1 Forces the flushing of database buffers to disk on COMMIT only

  Depending on the number of database updates, the frequency of COMMITs, and other concurrent computer activities, use of these environment variables could cause significant performance degradation.

  2784949 (1095190)
- The use of the environment setting MFJ_INPUTDS_ERROR=N has been extended so that an input file for a JCL step is now considered as optional and you no longer receive an error when the file is missing.

  2784622 (1095971)

J2EE Connector

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

(606556)

MTO: CICS Communications

- In CRTE mode, the transactions are now run using the correct userid if the user has signed on after running CRTE.

  2663890 (1091979)

MTO: IMS DB

- The NODCX mfims dbdgen directive has been added to enable you to compile DBD source without executing data capture exit routines contained in the DBD source. Previously, data capture exit routines contained in the DBD source but that were not found during compilation were ignored. The new default behavior is to process all data capture exit routines unless NODCX has been specified.

  2579600 (1084675)

MTO: IMS MFS

- All existing IMS Global Physical Terminal edit routines (DFSGPIX0) in use must be recompiled with charset EBCDIC instead of charset ASCII.

  (606142)

MTO: JCL MVS

- Following the detection of an unrecoverable file status error on the casspool file, a message is displayed on the console and an ACCEPT statement is issued. This stops processing, allowing you to fix the underlying file problem, before continuing the job.

  2651654 (1090287)

MTO: JCL Utils

- You can now activate support for the VSAM Shareoption for batch jobs and for CICS files that use a catalog entry. For more details, see your product help.

  2660651 (1091290)

Open PL/I Compiler

- ⚠ **Important:** If you are installing this release as an upgrade to Enterprise Developer 2.2 Update 1, after the upgrade you must rebuild any applications that are compiled using the –zp1 option.

  The behavior of the –zp1 option has been reverted to that of versions of Enterprise Developer earlier than 2.2 Update 1, with an additional correction relating to Char Varying data items. For a full description of the –zp1 option, refer to the Open PL/I User’s Guide in the product help

  This fix restores the behavior in Enterprise Developer versions earlier than 2.2 where, when compiling with –zp1, all parameters are treated as unaligned. (In Enterprise Developer 2.2 Update 1, the behavior when compiling with –zp1 was to not treat parameters as if unaligned).

  When using the –zp1 compiler option, all Character Varying data items are now treated as if unaligned. In previous versions of Open PL/I, for Character Varying data items, the –zp1 unaligned requirement was applied only to structure members and parameters.

  To illustrate the change, consider the following example:

  ```
  zptest: proc options(main);

     dcl 1 st1,
       2 c char,
       2 x(4) char(7) var init ('a', 'xx', 'yyy', 'zzzz');
  ```

```
   dcl y(4) char(7) var init ('a', 'xx', 'yyy', 'zzzz');

   dcl sub entry ((4) char(7) var);

   call sub (x);

   call sub (y);

end;

sub: proc (z);

   dcl z(4) char(7) var;

   dcl i fixed bin(31);

  do i = 1 to hbound(z);
      z(i)  = 'x';
  end;

end;
```

Where:

- For x and z, each `char (7) var` item is 7 plus 2 bytes which equals 9 and then multiplied by 4 equals 36.
- If y were aligned on half-word by default, each array element is half-word aligned and each equals 10 bytes (9 + 1 pad byte), and the total size equals 40 bytes.
- At `call sub (x)`, the calling argument and parameter are matched.
- At the `call sub (y)`, the y element size (10 bytes) is mismatched against the parameter z element size (9 bytes) due to –zp1. This is incorrect and causes unexpected program behavior.

Due to this correction of treating all Char Varying data items as if unaligned when using –zp1, the size of CHAR VARYING arrays now differs from previous versions of Open-PL/I. For example:

```
   dcl X(4) char(7) var;

   Put skip list (size(X))  /* size is 36 bytes vs. 40 bytes in previous
versions of Open-PL1 */
```

2789213 (1095636)

Open PL/I Debugger

- Previously, the SRC list in the debugger only included the source filename without the path to the file. If a source file did not exist in the current directory, the debugger could not find it using the filename as it always expects a fully qualified filename. The SRC list now contains the fully qualified file name which includes the path. Also, when changing the SRC or ENV values the source is updated correctly.

  2783734 (1094988)

Significant Changes in Enterprise Developer 2.2 Update 1

Enterprise Developer version 2.2 update 1 includes significant changes in the following areas:

- *.NET Compiler*
- *MTO - JCL ESMAC*
- *MTO - JCL MVS*

- *MTO - JCL System Catalog*
- *MTO - JCL Utils - IDCAMS*
- *Open PL/I Compiler*
- *Open PL/I Debugger*
- *Open PL/I RTS*
- *SQL: COBSQL*

.NET Compiler

- In member reference in managed COBOL syntax, you may now only use parentheses when referencing methods. You can no longer specify parentheses when referencing fields or properties, as this will produce a syntax error. For example:

  ```
  set intLength to testString::Length()
  ```

  must change to:

  ```
  set intLength to testString::Length
  ```

MTO - JCL ESMAC

- As part of an enhancement to improve performance of the spool display function, several new indices have been added to the `casspool.dat` file. If you are upgrading from a previous release, you must convert the `casspool.dat` file of any enterprise server created before Enterprise Developer2.2 Update 1 to the new format before starting any JES-enabled enterprise server and submitting jobs to it.

  Enterprise Developer version 2.2 update 1 provided the following scripts that convert the `casspool.dat` file to the new format:

  - On Windows: `fixcasspool.bat` in `%COBDIR%\bin` or `%COBDIR%\bin64`
  - On UNIX/Linux: `fixcasspool.sh` in `$COBDIR\bin`

  ,

  To convert the `casspool.dat` file, run the appropriate script from an Enterprise command prompt using the following syntax:

  - On Windows:

    ```
    fixcasspool.bat <old-location>casspool.dat <new-location><backup_filename>
    ```
  - On UNIX/Linux:

    ```
    fixcasspool.sh <old-location>casspool.dat <new-location><backup_filename>
    ```

  Where:

  - *<old-location>* - the fully or partially qualified location of the existing `casspool.dat` file, if not in the current directory. Optional.
  - `casspool.dat` - the spool file that will be updated. The file is located in the same directory as the enterprise server's system catalog.
  - *new-location* - the fully or partially qualified location where the backup file will be created, if not in the current directory. Optional.
  - *backup_filename* - the name of the file that will be a back-up of the old `casspool.dat` file. If the file name already exists, the script terminates without converting the `casspool.dat` file.

  If you do not convert `casspool.dat`, attempts to start the enterprise server fail with the following messages:

  ```
  JES000011E Batch Spool File Error; Status [39]
            JES000130E Note JES000011 Error - New casspool indexes may need
  generating. See error
            description in product docs
  ```

2654417 (1090421)

MTO - JCL MVS

- The final disposition processing of temporary datasets has been corrected to be in line with the mainframe processing, with a default value of PASS unless DELETE has been specified in the DD card. Previously, the default value was DELETE.

  2671175 (1091627)
- When running batch programs, receiving an "Out of Space" error requires operator intervention to continue processing.

  2642115 (1089055)

MTO - JCL System Catalog

- You can now use MGMTCLAS to provide an EXPIRE AFTER number of days. This is used to calculate an expiry date for datasets allocated using the MGMTCLAS parameter.

  2664639 (1090992)
- MFLEX*.DAT files were not being moved with spool files. Spool housekeeping now deletes LEX files when it deletes JESYSMSG. Also, to identify any remaining orphan LEX files, the MF_SPOOL_HK_LEX_SCAN environment variable sets spool housekeeping to scan the data directory for LEX files older than the max retain period.

  2659555 (1090759)
- Spool housekeeping has been rewritten to provide a number of new features. These will be described in the product documentation.

  (592452)

MTO - JCL Utils - IDCAMS

- The state of VSAM virgin files is now being correctly processed.

  2686149 (1093587)
- A REPRO COUNT of zero is now processed as zero records to be included in the REPRO operation.

  2665006 (1091034)

Open PL/I Compiler

- On Intel platforms, the -bitsltr Compiler option can be used to store bit strings left-to-right within each byte. This is similar to Big Endian bit ordering on Intel platforms. The Intel default is to store bit strings right-to-left within each byte similar to the Intel hardware bit ordering.

  2675860 (1092098)
- The libraries `libmf.so`, `libespli.so` and `libespliz.so` are no longer delivered. This is transparent if linking PL/I objects using mfplx or ldpli, and no changes are needed to your build process.

  However, if linking PL/I objects using either the UNIX ld command or the Enterprise Developer cob command, you must include the libraries `libmfpli.so` when linking stand-alone Open PL/I objects (executed outside of Enterprise Server), or `libmfpliz.so` when linking PL/I objects executed under the control of Enterprise Server.

  To do this, use the option -lmfpli for stand-alone Open PL/I objects or -lmfpliz, respectively, for PL/I executed under Enterprise Server. These options replace the –lmf option.
- Open PL/I applications must be recompiled and re-linked in order to use them with this release.

Open PL/I Debugger

- The debugger now stops before an ON UNIT triggers. This lets you perform debugging on the ON UNIT itself. You can configure the ON UNITS the debugger stops on.

  2674783 (1092319)

Open PL/I RTS

- Writing to SYSPRINT when running outside of Enterprise Server now drives output directly to the terminal and does not buffer the I/O.

  2477031 (1076203)

SQL: COBSQL

- COBSQL now displays appropriate COBOL syntax errors after encountering EXEC SQL statement errors.

  2673619 (1093197)

Significant changes in Enterprise Developer 2.2

Enterprise Developer version 2.2 includeD significant changes in the following areas:

- *.NET Compiler*
- *ASM Compiler: HLL Macro Assembler*
- *CCI Session Layer Code*
- *Compiler*
- *File Handling - Sort / JCL Sort*
- *Interface Mapping Toolkit*
- *JCL System Catalog - Dataset Allocation*
- *MF Directory Server*
- *MF Server Administrator (GUI)*
- *MFBSI*
- *MTO - IMS Other*
- *MTO - JCL MVS*
- *MTO - JCL System Catalog*
- *Open PL/I Compiler*
- *Open PL/I Debugger*
- *Open PL/I RTS*
- *Request Handler*
- *Samples Browser*
- *Visual Studio IDE*

.NET Compiler

- The use of ILSMARTLINKAGE in sub-programs is now working correctly. In previous versions, if it was specified in a sub-program, a dynamic call of that program from another program could cause execution failure.

  2608825 (1087435)
- In managed code, the statement DISPLAY UPON SYSERR now behaves as expected. Previously, it was behaving the same as DISPLAY UPON CONSOLE.

  2589053 (1085576)
- When the PROPERTY keyword is used on a numeric edited field, the SET method now works correctly when the program specifies DECIMAL POINT IS COMMA.

2588508 (1085560)

ASM Compiler: HLL Macro Assembler

- A "." (dot) specification is no longer ignored in the SYSIN environment variable or in the mf370ctl.cfg file when used to specify an output directory, and indicates the current directory. This is a change in behavior from Mainframe Express, where the "." (dot) character in the SYSIN environment variable or in the mf370ctl.cfg file is still ignored and assembler output files are placed in the directory specified by the SYSIN environment variable.
- Assembler Build Output will now be sent to the location specified in the COBOL output path project setting. Previously, all Assembler build output went to the project directory.

CCI Session Layer Code

- A new option, use_global_namespace, is available for the cci.ini file in the Windows %SystemRoot% folder. If use_global_namespace is set, all the ccishared memory objects are created in a system-wide address space, and the applications hosted by different users, including system services, can communicate. To use this facility, edit the cci.ini file and ensure use_global_namespace is set to "yes". [ccismem-base] # Allow interaction between users on a # single system. Using this option reduces security as # all users will have access to the same name space. use_global_namespace=yes If the value of this option is anything other than "yes", or if the option is missing, no change is made to the existing behavior.

  2195519 (1062800)

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.

File Handling - Sort / JCL Sort

- When used with MFJSSTRICTSORT=ON, MFJSORT now errors (return code 16) if the input RECFM is different to a SORTOUT RECFM. It also errors if the input RECFM is different to an OUTFIL RECFM when FTOV or VTOF/CONVERT has not been specified.

  2643384 (1088993)

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.

- The default for previous releases was to create a catalog entry with no physical file until data was actually written to the catalog. Now, a PC file is allocated for non-VSAM files by default. You can emulate the previous behavior if you set the environment variable MFALLOC_PCFILE to N.

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)

MFBSI

- casout now correctly sets return-code to the job status for the sysout retrieval when the file is less than 65K. Also, MFBSI now considers return codes of 0, 7 or 8 to be successful for JESYSMSG retrieval where, previously, it only considered 0 as successful.

  2655482 (1090444)

MTO - IMS Other

- The DBDCPY, MFSCPY, and PSBCPY environment variables are no longer required and have been deprecated. Use the IMSCPY environment variable to specify copybook locations.

MTO - JCL MVS

- Empty values for the USER or NOTIFY parameters on the JCL JOB card no longer cause an error when parsing the JCL.

  2646770 (1089380)
- The disposition processing of PASSed datasets has been revised to be consistent with mainframe processing. Particularly, received datasets are removed at the end-of-step processing, unless they are to be passed on to another step in the job. Any passed datasets that were not received at the end of the job may be removed if they conform to the mainframe conditions for this.

  2610362 (1087601)

MTO - JCL System Catalog

- The value for the JCL DD UNIT=device parameter is now stored in the system catalog entry for the dataset.

  2619050 (1089095)

Open PL/I Compiler

- Open PL/I applications must be recompiled and re-linked in order to use them with this release.

  2675860 (1092098)
- If using the -cics, -ims, or -mvs options when compiling on a little endian platform, the possibility for errors existed. Starting with this release, using the -cics, -ims, or -mvs options automatically uses -bigendian, thereby eliminating the possibility for these errors. Evaluate any applications that were built with a prior release to determine any impact on any data files being used, or where programs compiled with the -mvs option pass data to sub-programs compiled without it. The -bigendian option can be suppressed by adding the -native option after the -cics, -ims, or -mvs options.

Open PL/I Debugger

- CodeWatch no longer attempts to use an incorrect directory when starting. Also, CodeWatch is now the default debugger when no other debugger is specified, and is the only debugger available when using -optexec plitest.

  2655906 (1090193)
- Support for the evaluation of X, NX, Z, A, and E constants nas been added to CodeWatch.

Open PL/I RTS

- A problem with possible truncation or trailing zeroes within PICTURE assignments has been fixed. As the fix depends on changes to both the Compiler and the run-time, you must perform a complete rebuild of all your source code (recompile and relink) in order to apply it.

  2662261 (1090768)
- Previously, when taking a PLIDUMP, if the .PDB file needed for a PLIDUMP was not in the same directory as the .DLL and the .STB files, and they were not in the first CAS_BATCH_PATH subdirectory, then the PLIDUMP was not successfully generated and an S0C4 abend could occur.

  2615412 (1090483)
- The Open PL/I debugger no longer loses its place and behaves erraticly when debugging code containing the same PL/I macro multiple times within the same source routine, even if the -nodebuginfo and -pp options are not specified.

  2612530 (1087809)

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.

Samples Browser

- If you have installed the version of this product for both Visual Studio 2010 and Visual Studio 2012, the Samples Browser now provides links for you to open the samples in either one of them.

Visual Studio IDE

- Performance when loading and building COBOL projects that consist of a large number of files has been improved.

  2657121 (1090316)
- When building a project outside of the IDE, adding a new file to a project no longer results in a full rebuild of the project.

  2617003 (1089623)
- You can now change the display format of values on individual rows in the Watch window using modifiers after the names of variables and expressions. To do this, click a row, press F2, and type a modifier after the name of the variable or the expression as follows - type VariableName,h or VariableName,x to always display the variable or the expression in hexadecimal format; type VariableName,d to always display numeric variables in decimal format and strings - as text.

  2614182 (1087959)
- When you perform a search in the IDE for copybooks that your programs depend upon, the COPYEXT directive you set from the Additional directives field on the COBOL tab in your project's properties is now used in preference to the list of copybook extensions specified in Tools > Options > Text Editor > Micro Focus COBOL > Advanced, and in Copybook extensions. This change does not affect the project build and background parsing that already use COPYEXT.

  2612053 (1087773)
- Attempting to restart an enterprise server failed if the process took longer than fifteen seconds.

  2607051 (1087259)
- When you debug native code and query a data item which contains null bytes, the value displayed in the Watch window is no longer truncated at the first null byte.

  2604749 (1087235)
- A new setting, "Define DEBUG constant", is now available on the COBOL page in the project properties of managed projects for the Debug configuration. You can select it to add "constant"DEBUG(1)"" to the build directives.

  2600567 (1086629)
- The Output window now displays a notification when the compilation of the IMS files has been successful.

  2600137 (1086667)
- A problem with the value of "Link with objs" setting being duplicated after you reload the COBOL Link properties page has been resolved.

  2595408 (1086091)
- IntelliSense now displays data names longer than 32 characters correctly.

  2594901 (1086053)
- Previously, when you upgraded COBOL projects with signed assemblies from Visual Studio 2003 format to Visual Studio 2010 format, the signed assembly property was lost.

  2585458 (1085258)
- There is an improvement in the performance of the cursor in the text editor when working with larger files and projects.

2585450 (1085255)

- Using linked files in projects has been improved as follows:
  - Folders that only contain linked files no longer disappear from the project when you reload it.
  - The location of linked files is preserved in the project structure when you reload the project.
  - Moving a linked file within a project no longer moves the actual file on the disk but only moves the link inside the project structure.
  - It is no longer possible to copy a linked file into the same project where, previously, this created a copy of the actual file on the disk.

2574102 (1084208)

- When you convert Net Express projects that compile to .int or .gnt files to Enterprise Developer for Visual Studio, the application environment variables are now imported successfully.

2569777 (1083953)

- It is now possible to convert Net Express projects that do not contain any COBOL source files to Visual Studio projects.

2568638 (1083900)

- You can now set environment variables used by the native COBOL Run-Time system during initialization from the project properties - click Environment on the Application tab.

2539447 (1081251)

- Pressing Enter to create a new line now preserves the indent in the COBOL editor in Smart edit mode.

2496411 (1077769)

- The "Edit IMS Database" command was disabled for DBD files in your project if the ACB file was not in the project directory. You can now optionally define a location for the ACB files different than the project directory by setting the ES_ACBLIB environment variable.

2496391 (1077864)

- Pressing End inside the writing area of the COBOL editor now positions the cursor on the first non-blank character, if columns 12-72 contain any text. If the line is blank, the cursor is positioned on column 73.

2496351 (1077767)

- The COBOL editor now supports a "Smart edit mode" (see Tools > Options > Text Editor > Micro Focus COBOL > Margins) which is enabled by default. In Smart edit mode, when you press ENTER in the code (A/B) area the comments move to the next line and stay in the right-hand area.

2496350 (1077768)

- You now receive a suitable error message when you try to use "Start Without Debugging" with Mainframe Subsystem Application projects.

2495298 (1078343)

- You can now use Solution folders to group projects in your solution.

2259310 (1065760)

- If, when you start debugging, the debug target does not exist you now receive an error message that correctly states the missing filename.
- Any leading or trailing spaces you added to the name specified in the "Output name" field on the Application tab in the properties of a native project are now ignored.
- Class view no longer contains outdated information from previous background syntax checking after classes have been renamed.
- Class view no longer contains outdated information from previous background syntax checking when the files defining types have been removed.
- A problem with navigating to member definitions from the Class View has been resolved.
- A problem with navigating to member definitions from the Class View has been resolved.
- An issue where canceling some of the dialogs in Tools > Options resulted in the changes being saved anyway, or the pages not reverting when next loaded, has been resolved.

- You can now start and stop servers from the Server Explorer when using the local machine's IP address or DNS name.
- It is now possible to debug a project that has its "Compile for debugging" setting enabled in the project properties for the Release configuration.
- Selecting a project in Class View now triggers background syntax checking, if it has not been performed for that project yet.
- If you have .NET Framework v. 4.5 installed, creating a COBOL Web application with Visual Studio 2010 Shell was failing with a Signal 114 error and was causing the IDE to crash. This issue has been resolved for the IDE. To work around the issue when you compile COBOL Web Applications from the command line, you need to manually set the environment variable VisualStudioVersion to 10.0 before compiling.
- Using the tab character in comments no longer causes issues with the colorization of the code.
- The Visual Studio IDE no longer crashes if you try to close it while directives scan is in progress.
- Adding files to a project when directives scan is disabled was taking a long time to complete.
- Friend assemblies (as identified by the InternalsVisibleToAttribute) are now correctly handled during syntax checking, and types are correctly found within friend assemblies when they are present.
- An issue with adding new platforms or configurations to a project or solution when the project contains pre- or post-build events has been resolved.
- A problem with navigating to member definitions from the Class View, when the definitions occurred within copybooks used within partial classes, has been resolved.
- A problem where trying to navigate to the source of an item from the Class View positioned you on the wrong line, has been resolved.
- An issue where the Implements smart tag was present for interfaces that were fully implemented (when some of the methods from that interface were implemented explicitly; and in addition to another interface which was not fully implemented) has been fixed.
- An issue where method signatures were rendered incorrectly if the member had a 78 level item in its local storage has been resolved.

Significant Changes in Enterprise Developer 2.1 Update 1

Enterprise Developer version 2.1 update 1 includes significant changes in the following areas:

- *Documentation*
- *JCL Support*
- *PL/I Support*

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.

JCL Support

- Input files are no longer optional so you can receive errors if you catalog a file without creating the physical file. To help avoid failures of JES job steps because of a missing input file, you can use one or both of the following environment variables: 1) MFALLOC_PCFILE=Y - results in the creation of a catalog file if DSORG=PS is set in the DCB parameters. 2) MFJ_INPUTDS_ERROR=N - results in IEBDG, IEBGENER and IEBCOMPR utilities to treat missing input files as though they were present but empty files.

PL/I Support

- When running the macro preprocessor as part of an Open PL/I compile, if there are missing %INCLUDE files, we now flag them as SEVERE errors which are visible in the Visual Studio Output Tab even with the minimal level of MSBuild information enabled. We terminate after the macro phase, allowing the customer to correct the situation before continuing.

Significant Changes in Enterprise Developer 2.1

Enterprise Developer version 2.1 includes significant changes in the following areas:

- *CICS Support*
- *Compiler*
- *IDE*
- *Run-Time System*
- *Vision File System*

CICS Support

- If external security is configured (MLDAP) and the default ESMAC user is not allowed to login, the region does not startup.

  2550800 (1082337)

Compiler

- The use of extended ACCEPT or DISPLAY statements in Windows applications (ilsubsystem"2") now produces error COBCH1634: Extended ACCEPT/DISPLAY not allowed with a Windows GUI executable (ILSUBSYSTEM"2"). Remove the syntax or change the output type of your application to 'Console Application'.

  2575702 (1084365)

IDE

- A problem with the item templates in the Add New Items dialog box not available for PL/I projects has been resolved.

  2581187 (1084876)
- There is now a build summary report for projects that compile to multiple output files.

  2578006 (1084506)
- Previously, Visual Studio used to always rebuild the entire project after you changed the project properties or added files when there were build error. This has been changed so that during subsequent builds Visual Studio only rebuilds the files that are out-of-date.

  2576400 (1084393)
- A new setting has been added to the COBOL tab in the project properties, "Generate listing option", which produces an .lst file in a Listing subfolder in the project directory. The "Generate directives file" setting now generates the directive file in the project output directory.

  2494484 (1077662)
- Command line arguments are now passed to the program when started via the 'Start without Debugging' option in the Visual Studio IDE.
- Previously, if you started debugging using CBL_DEUGBREAK and there was a COBOL source file open in the editor, when you stopped debugging its dictionary file (.idy) was not being released. This resulted in any subsequent builds and rebuilds failing. This problem has been resolved.
- The native COBOL projects in Visual Studio have been enhanced so that you can now build them to multiple executables - each COBOL program in the project producing a separate executable.

- You no longer receive an error message when adding an event handler to an object in a COBOL WPF project if the .xaml code behind file includes a class that does not have methods in it.
- The IDE no longer ignores certain Run-Time System errors from the Compiler which previously caused the build to fail without an explanation.
- Compile extensionless COBOL files with the IDE no longer fails.
- Using the UI to add COBOL switches or environment variables to a COBOL Web Application (right-click the configuration file in Solution Explorer and select Edit) no longer causes a configuration error when you run the application.

  2558133 (1082871)

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 Enterprise Developer 2.0

Enterprise Developer version 2.0 includes significant changes in the following areas:

- *Compiler*
- *HCO for DB2 LUW*
- *File Handling*
- *IDE*
- *OpenESQL*
- *Run-Time System*

Compiler

- An attempt to serialize a class containing COBOL data such as PIC X or group items could lead to a serialization exception.

  2555177 (1082730)
- Dependent assemblies (ones that are required by other assemblies) that fail to load now issue a COBCH1581 warning "Could not load dependent assembly AssemblyName". Previously, they generated a COBCH0942 error. If an assembly generates both COBCH0942 and COBCH1581 errors, the error report lists only the COBCH0942 error.

  2507380 (1078845)
- 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.

HCO for DB2 LUW

- 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)

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

  2525670 (1080092)

IDE

- Using File > New > Solution From Net Express Project to convert a Net Express project to a Enterprise Developer solution on a machine with a language locale other than English or Japanese produced an empty solution.

  2548551 (1082073)
- The COBOL tab in the project properties of a new WPF Application sometimes was not showing any settings.

  2535705 (1080950)
- In some menus and dialogs in the IDE copybooks were referred to as copy files. They are now consistently referred to as copybooks.
- The project properties have been enhanced. The SQL directives are no longer shown in Build Settings on the COBOL page where they sometimes appeared as out-of-date. The SQL directives now appear only on the SQL page.
- Adding a watch on an item of the form buffer(1:size) caused the IDE to hang when "size" was not initialized.
- ActiveX components in the Windows Forms of a managed project could cause the programs to crash at run-time. Previously, to work around this problem, you had to check the ActiveX references in the project and set their CopyLocal reference property to True. This is now done automatically when the references are added.

  2536071 (1080978)
- Adding an existing Windows Form to another project as a linked file did not function correctly. When you reopened the project, the designer and the resx files were missing from all Windows Forms in the project.

  2524313 (1079984)

OpenESQL

- OpenESQL did not always return a consistent error message or it returned an incorrect error message when a data conversion exception occurred within the ADO.NET run-time system.

Run-Time System

- On Windows 7, building 64-bit native COBOL applications always rebuilt the entire project. This was caused by the Microsoft FileTracker tool, used by the COBOL projects to track dependencies, not working because of Windows 7 security updates. This product now includes a fix to work around this issue.

Unsupported or Deprecated Functionality

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

Unsupported or Deprecated at Enterprise Developer 4.0

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

Unsupported or Deprecated at Enterprise Developer 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 Enterprise Developer for Visual Studio. The run-time components are installed as part of Enterprise 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 Enterprise Developer 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 Enterprise Developer 2.3 Update 1

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

Windows Azure versions

Back to Top

- Starting with Enterprise Developer 2.3 Update 1, versions of the Microsoft Azure SDK earlier than version 2.8 are no longer supported.
- Support for the Microsoft Azure SDK is no longer provided in Enterprise Developer for Visual Studio 2012. If you have COBOL Azure projects created with earlier versions of Enterprise Developer for Visual Studio 2012, to maintain them, use Enterprise Developer for Visual Studio 2013 or 2015.

Unsupported or Deprecated at Enterprise Developer 2010

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

These Net Express compiler directives are not supported by Enterprise Developer :

01SHUFFLE
64KPARA
64KSECT
AUXOPT
CHIP
COBIDY
DATALIT
EANIM
EDITOR

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

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

BASENAME
FILENAME
PATH
TARGETDIR

Known Errors and Restrictions

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

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

The following errors and restrictions are known to apply to Enterprise Developer version 4.0:

- *Azure support*
- *CICS*
- *Commuter licenses*
- *Debugging*
- *Docker*
- *Enterprise Server*
- *Installation*
- *Micro Focus Rumba*
- *OpenSSL*
- *PL/I Support*
- *Resource Adapters*
- *REST/JSON IMTK implementation and the same-origin policy (SOP)*
- *Visual Studio*

Azure support

- When compiling COBOL Azure Role projects in Enterprise Developer for Visual Studio 2013 you receive warnings at build time and the code fails to execute when deployed to Azure.

  To work around this issue, you need to manually add the code below to the configuration section of the `Web.config` file in your project for Web Roles or in the `app.config` file for Worker Roles:

```
<runtime>
      <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
          <dependentAssembly>
              <assemblyIdentity name="Microsoft.Data.Services.Client"
publicKeyToken="31bf3856ad364e35" culture="neutral" />
              <bindingRedirect oldVersion="0.0.0.0-5.7.0.0"
newVersion="5.7.0.0" />
          </dependentAssembly>
          <dependentAssembly>
              <assemblyIdentity name="Microsoft.Data.OData"
publicKeyToken="31bf3856ad364e35" culture="neutral" />
              <bindingRedirect oldVersion="0.0.0.0-5.7.0.0"
newVersion="5.7.0.0" />
          </dependentAssembly>
          <dependentAssembly>
              <assemblyIdentity name="Microsoft.Data.Edm"
publicKeyToken="31bf3856ad364e35" culture="neutral" />
              <bindingRedirect oldVersion="0.0.0.0-5.7.0.0"
newVersion="5.7.0.0" />
          </dependentAssembly>
          <dependentAssembly>
              <assemblyIdentity name="Newtonsoft.Json"
publicKeyToken="30ad4fe6b2a6aeed" culture="neutral" />
              <bindingRedirect oldVersion="0.0.0.0-8.0.0.0"
newVersion="8.0.0.0" />
          </dependentAssembly>
      </assemblyBinding>
</runtime>
```

This information is added automatically when you create COBOL Azure projects in Visual Studio 2015 or later.

CICS

EXEC CICS VERIFY PASSWORD has a failure scenario that doesn't exist on the mainframe. It is currently not possible to distinguish this failure from an incorrect password on the command itself via the returned EIBRESP and EIBRESP2.

Under Enterprise Server, the following options mean that an admin LIST request to ESF is necessary to acquire the appropriate information:

CHANGETIME
DAYSLEFT
EXPIRYTIME
INVALIDCOUNT
LASTUSETIME

If no users are signed on to CICS, then this admin request will be made as the default CICSUSER which will be rejected.

The VERIFY PASSWORD command will return NOTAUTH 2, "The supplied password is wrong".

This is misleading because it implies that the value for PASSWORD on the command itself is incorrect. Note that in that case, CICS returns NOTAUTH 2 as per the mainframe.

The aggregated return code from ESF to CICS for the admin request is identical in both cases and so CICS has no way of distinguishing between these two failure scenarios. We therefore cannot return a new EIBRESP or EIBRESP2 for this particular case.

To allow CICSUSER to make admin LIST requests, the anonymous list option needs to be enable in the region's security manager configuration (not the Security Manager):

```
[Admin]
allow-list=yes
```

Commuter licenses

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

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

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

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

Debugging

- When debugging `.gnt` code on early builds of Windows 10 and Windows Server 2016, execution fails to stop at COBOL breakpoints. To resolve this, you should install the most recent Windows updates for your operating system, as this problem is only known to be present with the earlier versions.

Docker

- Using the Micro Focus SupportLine Support Scan Utility, `MFSupportInfo`, in a container is not supported on Windows.
- When using the Docker support provided in the Enterprise Developer Build Tools or Visual COBOL Build Tools, on Windows Server version 1803 or Windows 10 version 1803, the following errors might be encountered:
  - Windows Server version 1803: container "xxxx" encountered an error during CreateContainer: failure in a Windows system call: The operating system of the container does not match the operating system of the host. (0xc0370101)
  - Windows 10 version 1803: container "xxxx" encountered an error during Start: failure in a Windows system call: The compute system exited unexpectedly. (0xc0370106)

  The problem is encountered only when creating containers which are intended to build COBOL applications. The creating of containers for application execution is not affected.

  Please, contact Micro Focus SupportLine for further information.

Enterprise Server

- The Historical Statistics Facility may generate incorrect records for SSTM-enabled enterprise servers.
- On Windows 10, if you are using Microsoft's EDGE browser to access the Enterprise Server Administration GUI, issues with EDGE can cause the automatic refresh feature to display a dialog asking whether you want to resubmit a form. To work around this issue, cancel the resubmit request and then refresh the server list page or the Home page of Enterprise Server Administration. You can also turn off the automatic refresh by setting the **Auto-refresh interval** setting on the Home page of Enterprise Server Administration to 0.
- In Enterprise Developer 2.3, some job scheduling tools from third-party vendors might stop communicating with Enterprise Server. This might have an impact on our external partners who integrate with Enterprise Server such as job schedulers which do not use MFBSI. The issue might occur with job schedulers such as Zena, CA, ASG or UC4, if these are run outside of a COBOL environment with the path set. *Click here* for more information about the issue and how to work around it.

Installation

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

Micro Focus Rumba

- On versions of Windows Vista and later, Enterprise Server listens only on the IPv4 loopback address (127.0.0.1). As a result, an attempt to connect to localhost with a TN3270 emulator such as Micro Focus Rumba may fail. To work around this issue, in your emulator's configuration use 127.0.0.1 in preference to localhost or your host machine's name.

OpenSSL

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

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

PL/I Support

- The Micro Focus PL/I Macro Preprocessor supports the majority of the IBM PP(MACRO) functions. The only support for PP(PLX) is for the deprecated KEYS option. There are no plans to further extend the existing support for PLX.

Resource Adapters

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

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

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

Visual Studio

- The Visual Studio 2017 option **Lightweight Solution Load** is not supported for COBOL. It must not be used with solutions that contain COBOL projects as it could result in problems with building the projects properly.
- In certain installation scenarios, after you install Enterprise Developer for Visual Studio 2015, the COBOL language service can be uninstalled. This results in the following features missing from the COBOL editor: colorization, IntelliSense for COBOL and any other COBOL language support. This is an issue with Visual Studio's MEF component cache becoming corrupted.

  To resolve this issue and restore COBOL language support in the editor:

  1. From a command prompt, navigate to `C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE`.
  2. Execute the following commands:

     ```
     devenv /updateconfiguration
     ```

     And then:

     ```
     devenv /clearcache
     ```

Upgrading from Mainframe Express to Enterprise Developer

An introduction to the process of upgrading your COBOL applications

Enterprise Developer enables you to develop applications that run on the mainframe or offload applications from the mainframe and develop them locally on Windows. If you are upgrading from Mainframe Express, you can move your existing applications to Enterprise Developer and continue to develop them in the new environment.

Once you have familiarized yourself with the basics of the Enterprise Developer IDE, you can import your existing code in Enterprise Developer.

See the *Upgrading to Enterprise Developer* guide for full details on upgrading your applications from Mainframe Express to Enterprise Developer, the changes between the two products and what technologies are not supported or have been superseded.

How to create an application created with Mainframe Express to Enterprise Developer

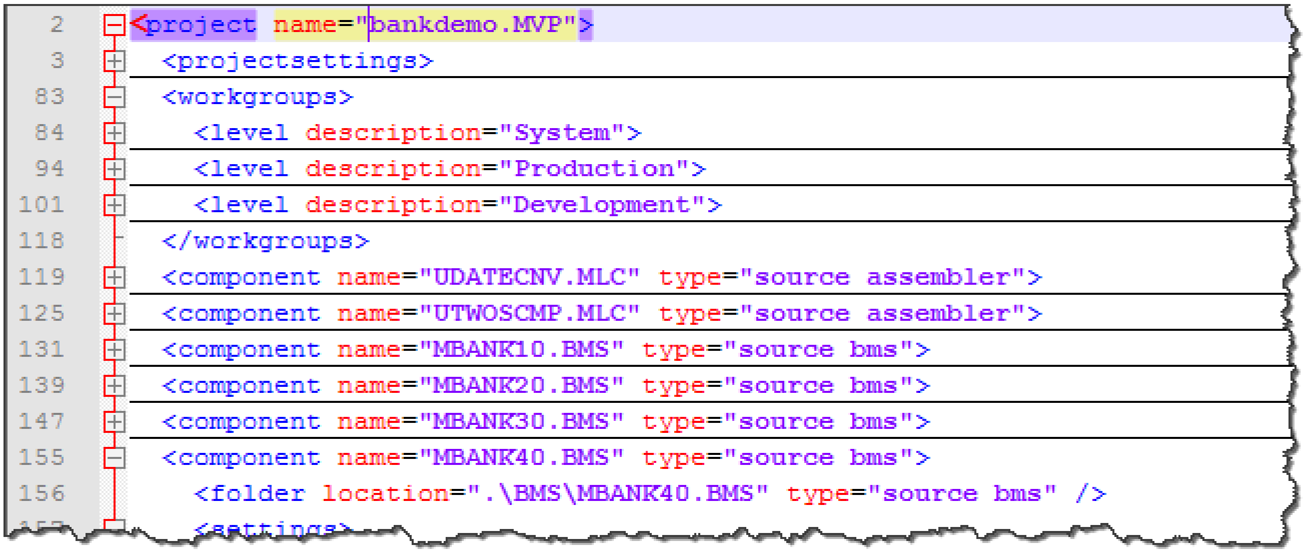
The following provides an outline of the incremental approach to upgrading a Mainframe Express application to Enterprise Developer where you use both development products to maintain the application while gradually moving the sources to the new IDE.

1. Review your Mainframe Express application:

    - Create an inventory of the sources it uses, the copybook dependencies, data files, SQL, JCL and IMS files, third party utilities, integrated preprocessors.
    - Check what workgroups are used and what the workflow processes are, what repositories are used to store the sources, what the build scripts and command line parameters are, how the application accesses the mainframe.

      Note: Use the MFECL command line utility of Mainframe Express to export the project details to an .xml file which will include all project details at a glance - files in use, workgroups, output and dependency folders, Compiler directives, project and file options. The command to use is:

      ```
      mfecl /exportxml project-name.mvp [ xml-filename.xml]
      ```
    - Review the XML file to see what information it includes:

      ```
      <project name="bankdemo.MVP">
        <projectsettings>
        <workgroups>
          <level description="System">
          <level description="Production">
          <level description="Development">
        </workgroups>
        <component name="UDATECNV.MLC" type="source assembler">
        <component name="UTWOSCMP.MLC" type="source assembler">
        <component name="MBANK10.BMS" type="source bms">
        <component name="MBANK20.BMS" type="source bms">
        <component name="MBANK30.BMS" type="source bms">
        <component name="MBANK40.BMS" type="source bms">
          <folder location=".\BMS\MBANK40.BMS" type="source bms" />
          <settings>
      ```

      The file provides details about the project and file settings, as well as of the workgroup levels. You are going to use this information when you are configuring the new projects in Enterprise Developer.

- Analyze the strengths and weaknesses of your application in the Mainframe Express environment and to what extent it meets the business requirements. The application might be a good candidate for being modernized by upgrading it to Enterprise Developer.

2. Define a small selection of the source files of your Mainframe Express application that you will move across to Enterprise Developer as a start – this could be the files responsible for a single operation of the application.

   See *Considerations* for whether the files need to be recompiled in the new environment.
3. Prepare your original application:
   - Replace any hard-coded paths in your Mainframe Express project with concatenated ones – to do this, in Mainframe Express, click **Project > Convert Paths to Relative**.
   - Backup the sources.
4. Create mainframe subsystem projects in Enterprise Developer.

   In Visual Studio:

   a. Click **File > New**.
   b. Under COBOL, click **Native > Mainframe Subsystem Application**.
   c. Click **OK**.

   To emulate workgrouping behavior in Visual Studio, you can create different projects in your solution that will represent each individual level in the workgroup. For each project, you need to specify the copybook paths that are for the respective workgroup level.
5. Configure the projects in Enterprise Developer.

   In Visual Studio, first turn off the automatic directive determination as you will be setting the directives manually:

   a. Click **Tools > Options > Micro Focus > Directives > COBOL**.
   b. Disable **Determine directives of files** and click **OK**.

   Configure the project properties:

   a. Click **Project > *YourProject* Properties** and click the individual tabs to specify folder for the generated files and any other COBOL, IMS, JCL and CICS properties.

      You need to set exactly the same project properties as the ones set for the MFE project as taken from the `*xml-filename*.xml` file.
6. Import the source files and establish a product baseline – use linking to add the files to the new projects in order to store the files in their original location:

   In Visual Studio:

   a. Right-click your project in Solution Explorer, and click **Add > Existing Item**.
   b. Browse to the directory which stores the sources of your Mainframe Express application.
   c. Select the files to add, and click **Add As Link**.

      This links the files to the project from within the original location and does not create copies of the files in the Enterprise Developer project directory.
7. Set individual file properties:
   - Right-click a file in your project, and click **Properties**.
   - Set individual file properties as taken from the `*xml-filename*.xml` file.
8. Compile the sources in Enterprise Developer.

   In Visual Studio:

   - Click **Build > Build Solution**. Check the **Output** window for the progress and the result of the build.
9. Create and configure an enterprise server instance for the projects.

   Check the `*xml-filename*.xml` file to see what regions the original application uses, and what its settings are.

10. Run the application.

    - Run JCL applications by submitting the JCL file – right-click the JCL file in your project, and click **Submit JCL** (in Visual Studio) or **Submit to Enterprise Server** (Eclipse).

      Use the Micro Focus Rumba emulator supplied with Enterprise Developer to connect to the enterprise server instance you have created. Run the application in Rumba.
11. Debug and perform thorough testing.
12. Configure MFA and the **Mainframe Explorer** window or the **Remote Systems** view.
13. Plan for rolling out Enterprise Developer project and servers to your development and QA teams. Also, organize relevant training for them.

    a. As a first step, ensure your project is configured as desired and that it builds successfully.

    b. Distribute the project:

    You can use Visual Studio to create customized templates with the desired set of options and the required source programs, and share these within your organization. Then, each user would be able to import the templates from a shared network location into their Visual Studio environments. See the Visual Studio help for more information on how to create, configure and distribute custom project templates.

    Enterprise Server templates:

    Create your own templates using the following command at an Enterprise Developer command prompt:

    ```
    mfds -x [repository type] [repository address] [server name] <options>
    <user id> <password>
    ```

    Distribute the server .xml files across your development teams. The file should be saved in the *VSInstallDir*`\Common7\IDE\Extensions\Micro Focus\Visual COBOL SOA\1.0\Etc\ServerTemplates` subfolder in the local Enterprise Developer installation folders.

Using Enterprise Developer for Visual Studio

Understanding the structure of Enterprise Developer solutions

On creating a new project, the following files are created in the file system with the following structure:

```
...
   |_Location
     |_Solution
     | |_Name
     |   |_bin
     |   | |_x86
     |   | | |_Debug
     |   | | |_Release
     |   | |
     |   | |_x64
     |   |   |_Debug
     |   |   |_Release
     |   |
     |   |_obj
     |   | |_x86
     |   | | |_Debug
     |   | | |_Release
     |   | |
     |   | |_x64
     |   |   |_Debug
     |   |   |_Release
     |   |
     |   |_Properties
     |
```

```
    |_Solution.sln
    |_Name.cblproj
    |_Name.dep
    |_Program1.cbl
```

If you select the Create Directory for Solution option when creating a solution, the structure is slightly different.

In the *Solution* folder:

- **Solution.sln** - a description of the solution and what it contains.
- ***Name*.cblproj** - the project file that is opened in Visual Studio, which holds the description of the project and all its related configuration and directives information.
- COBOL source files - when you create a project, a skeleton COBOL source file `Program1.cbl` is added for most of the project templates.

In the *`Name`* folder:

- `..\bin` - this is the default location of build artefacts. With this folder are the subfolders `x86\Debug` that contains the executables or libraries, and `.idy` file for each of the project's COBOL source files. The `.idy` files contain information required for debugging your application. When you use the Release build configuration, build output goes to a subfolder `x86\Release` and no `.idy` files are created.

  Debug and Release are standard build configurations that you launch from the Visual Studio task bar. They use a different set of compiler directives as well as outputting different files. You can create your own build configurations by clicking **Build > Configuration Manager** and choosing **New** from the **Active solution configurations** drop-down list.

  The `x86` folder exists because the default output platform is 32-bit. If you change this to be 64-bit, you will instead find your ouptut in an `x64` folder.
- `..\obj` - this also has `x86\Debug` subfolders, and contains an `.obj` file for each source file, used in intermediate build stages. The `obj` folder also holds supporting information such as logs and file lists.

Note: The project file `.cblproj` is an msbuild file, much like a makefile but consisting of XML that you can extend and modify to customise your builds. You can use this directly from command line, as it uses the same build environment as the IDE, and behavior is identical. This means you can have a single source of configuration information that makes your build process easier to maintain.

If you open a command prompt and change to the *Location* folder, you can execute the `msbuild` command, without needing to specify the `.cblproj` file.

Finding your way around the IDE's features

- **Solutions and projects**

  A solution is a container holding one or more projects that work together to create an application. The solution has the extension `.sln`. A COBOL project has the extension `.cblproj` and a C# project has the extension `.csproj`.

  Solution Explorer shows the solution that is open and the projects therein.

  You can use the project's properties pages to display a list of the files in your solution with file details like output file type and location, COBOL dialect, and the number of errors generated by the file. To display the properties, click **Project** > ***Name* Properties**.
- **COBOL editor**

  The COBOL editor provides help such as column cut and paste, and background syntax checking, which underlines errors with red wavy lines (also known as "squigglies"), which you can then hover over to display details of the syntax error.

  When you are editing, you can insert code snippets and navigate forward and backward quickly, and the **Find All References** option enables you to search for references of any COBOL data items, section and paragraph names in the solution.

You can customize the editor to display line numbers, adjust colorization, tabs, and margins, from the **Text Editor** > **Micro Focus COBOL** > **Advanced** page in **Tools** > **Options**.

When developing code, the editor provides IntelliSense that helps you write syntactically correct code and, in managed code, helps when you need to type more complicated constructs, such as the code to override the members that a class inherits from a base class or the code for implementing an interface.

The Smart TagLight Bulbs feature for implementing an interface helps complete incomplete interface declarations. A Smart TagLight Bulbs appears at the beginning of the declaration: click it and choose the missing member(s) of the inheriting interface.

When you encounter a COPY statement, or data item that is defined in a copybook, if you put your cursor on that code and press F12 the appropriate copybook opens in the editor at the relevant line. You can also do this by right-clicking the line and selecting **Show copybook name**.

- **Setting Compiler directives**

  Many Compiler directives are set automatically by certain configuration options in the IDE, but you can explicitly add directives to your project. Right-click on the project in Solution Explorer and choose **Properties**. In the COBOL tab, you can see directives that are set by the IDE in the **Build Settings** text box. Enter others in the **Additional Directives** text box as a space-separated list.

  If you use a separate text file to manage your directives, you can reference this instead by entering the USE"*directives file*" directive. You should enter a relative path.
- **Build Tools, the Output Pane and the Error List**

  Build configurations define how to build a project or solution. There are default configurations of Debug and Release for each project type, and you can create your own specific configurations.

  The Output window shows the results of your build together with errors. You can double-click an error and navigate directly to the appropriate line in the source code. You can do the same from the Error List.
- **Debugging**

  When you debug the application, you can step through the code, hover over a data item to see its value, and watch data item values in a variety of ways. You can specify breakpoints on a range of conditions, such as when an expression is true or changes, or when a line is hit a specified number of times.

  In native code, you can set COBOL watchpoints on data items and watch for changes in the area of memory associated with the watchpoints. When the memory changes, the debugger breaks on the line that follows the line on which the data change occurred.

  Also in native code, you can use the Memory window to watch the contents of the memory that is associated with data items or expressions.

Change the Defaults to Replicate Your Existing Project Structure

Change the location of source files

To add an existing COBOL source file to your project, right-click the project in Solution Explorer and choose **Add > Existing item**. You can then browse to the sources you want to add.

- If you click **Add**, Enterprise Developer makes a copy of the file, which it saves in the project folder. Any edits you make to this file do not get applied to the original.
- If you click **Add As Link**, a reference to the original file, rather than a copy of it, is added to the project in Solution Explorer. If you then open the file in Visual Studio, any edits are applied to the file in its original location.

You can also drag files from Windows Explorer and drop them into your project in Solution Explorer. This also makes a copy of the file and leaves the original in place.

To remove a file from your project, but not delete the file on disk (whether added as a link or not), right-click the file in Solution Explorer and choose **Exclude From Project**.

Change the location of built files

By default, built artefacts for the Debug configuration are created in the `..\`*`Location`*`\`*`Solution`*`\`*`Name`*`\bin\x86\Debug` folder.

You might want to change this, so that several developers can save built items in the same folder for example. To do this, right-click the project in Solution Explorer and choose **Properties**. In the COBOL tab, change the value of the **Output path** field to the preferred folder. (We recommend you always use relative paths when entering this value.) When the project builds, the output files will be saved in this folder, and the folder created if it doesn't already exist.

To change the output path for the Release configuration, select **Release Configuration** in the COBOL property page and change the value of the output path.

Change the type of built files

The default output and target types when you create a project depend on the project type. You can change these settings on the project **Properties** page. Use the following table to show the default output and target types for each project and the possible changes once the project has been created :

Project type		Output type	Target type	Possible output types	Possible target type
Native	Console application	`.exe`	single	`.dll`, `.exe`	single, multi
	Windows application	`.exe`	single	`.dll`, `.exe`	single, multi
	Link library	`.dll`	single	`.dll`, `.exe`	single, multi
	Enterprise Server application	`.dll`	multi	`.dll`, `.exe`	single, multi
	INT/GNT application	`.int`	multi	`.int`,`.gnt`	multi
Managed	Console application	`.exe`	single	`.dll`, `.exe`	single
	Windows application	`.exe`	single	`.dll`, `.exe`	single
	Link library	`.dll`	single	`.dll`, `.exe`	single
	Procedural multi-output project	`.dll`	multi	`.dll`, `.exe`	multi

Best Practice in Enterprise Developer Development

Break down large projects

Projects with a large number of source files and build artefacts can be hard to navigate and slow to build. If you find this the case. we recommend that you review the contents of large projects and split them into separate projects (and possible separate solutions) in which you group items that are logically related. These projects can still be built in the same output folder if required.

For example:

- If you have different versions of a product for different customers, keep common source in one project and a separate project for each customer. You could also have a master solution into which you add projects from other solutions by right-clicking a solution and selecting **Add > Existing Project**.
- If you have core code that is rarely changed or recompiled, keep that in one project and have separate projects for those areas that change regularly.

Referencing common sources

To avoid repetition and reduce maintenance effort, you should consider keeping all your Compiler directive settings in a directives file and reference this file in each project. Similarly you should keep copybooks in a single project and add this project as a dependency to your COBOL projects.

If using managed code and multiple projects, use project references rather than file references.

Create templates

After creating and configuring a project, you can save the settings as a template that can be reused and distributed to other users. It can be added to the list of project types available when clicking **File > New > Project > COBOL**.

To create a template of the open project, click **File > Export Template** and follow the steps explained in the Export Template Wizard.

Use relative paths

Keep source relative to a base path and avoid full paths so that code is portable and easy to use with source control systems. You should also avoid using network shares or drives.

Modernize Your Applications and Processes

Following industry standard development practises

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

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

Interface modernization

Enterprise Developer enables you to use Visual Studio's built-in design tools to create more intuitive user interfaces. By wrapping existing procedural COBOL in an wrapper class you can integrate your code into Windows Forms (WinForms) and Windows Presentation Foundation (WPF) technology, and WebForms for ASP.NET browser-based applications.

Multi-user applications

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

Developing Web-based applications

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

You can do this by creating an Enterprise Server application

Developing .NET applications

Both new and existing COBOL can be compiled as .NET managed code. This enables you to:

- Reuse existing COBOL business logic and data access across the .NET environment
- Access .NET Framework classes and features from COBOL applications including Windows Forms and Web Forms
- Create and extend composite applications consisting of COBOL, C#, VB.NET, C++ and ASP.NET

- Reuse and extend Open ESQL applications

Both procedural and OO COBOL are supported within the .NET framework. OO COBOL classes can inherit classes written in other Microsoft .NET languages and vice versa.

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

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

Modernizing Dialog System applications

Enterprise Developer provides the following support for Dialog System applications:

- Dialog System run-time system and run-time components.
- Panels V2.
- Dialog System painter.
- GUI class library and OLE class library. These libraries are needed if you migrate an existing Dialog System application that was extended using those libraries.

  Projects for building the GUI and OLE class libraries from source are also supplied. Additionally, a project file for the Base class library was added in Enterprise Developer 2.0.
- Visual Studio plug-in to associate screensets in Visual Studio with Dialog System. Double-clicking a screenset in Solution Explorer in Enterprise Developer starts the Dialog System painter.
- Sample applications demonstrating a range of modernization techniques.
- Supporting documentation in this Help explaining the significant elements of the sample code.

You can modernize Dialog System applications within Enterprise Developer. You migrate an application to Enterprise Developer and from there you can run the application without change, or modernize it over time.

Modernization techniques include:

- A Windows Forms form replacing a Dialog System dialog, where the form can contain .NET controls. See the Customer + .NET WinForm sample `CustomerWinForm.sln`.
- A Windows Forms control wrapped as an ActiveX control and used on a Dialog System dialog. See the Customer + .NET GridView User Control sample `custgrid.sln`.
- A WPF user control hosted by a Windows Forms user control, which is then exposed as ActiveX ready for use by Dialog System. See the Customer + .NET WPF GridView User Control sample `CustGridWPF.sln`
- A .NET managed code application interacting with Dialog System as native COBOL `.dll`. See the Managed Customer sample `ManagedCustomer.sln`.

Data File Tools

Enterprise Developer comes with two versions of the Data File Tools utility: Data File Tools and Classic Data File Tools.

The Classic Data File Tools is the utility that was previously available in Net Express. It includes the Data File Converter, Data File Editor, IMS Database Editor, Segment Layout Editor and the Record Layout Editor. This utility is only available on Windows.

Data File Tools is a new version of the utility and comprises the Data File Editor and the Structure File Editor. This utility is available on both Windows and UNIX.

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: `.il` for the .NET framework`.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 ilgen 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 (`.il`)(JVM byte code `.class`/`.jar`).

COBOL and all other .NET languages (for example C# and VB)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 Visual Studio there are native COBOL project templates.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 intermediate language (`.il` files) can be deployed to a Windows platform running Microsoft's Common Language Runtime (CLR) for execution.The JVM byte code (`.class`/`.jar` files) can be deployed to a JVM for execution.

All programs written for the .NET Framework are executed by the Microsoft's Common Language Runtime (CLR) which makes mixed language application programming seamless.

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

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 Enterprise Server.

.NET COBOL applications are deployed to Windows platforms running Microsoft's CLR.

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