



Micro Focus Visual COBOL 2.0 for Visual Studio 2010

A large, decorative graphic consisting of multiple overlapping, wavy blue lines that create a sense of motion and depth. The lines are in various shades of blue, from light to dark, and are arranged in a complex, swirling pattern that dominates the lower half of the page.

Release Notes

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

Copyright © 2009-2012 Micro Focus. All rights reserved.

MICRO FOCUS, the Micro Focus logo and Visual COBOL are trademarks or registered trademarks of Micro Focus IP Development Limited or its subsidiaries or affiliated companies in the United States, United Kingdom and other countries.

All other marks are the property of their respective owners.

2012-04-18

Contents

Micro Focus Visual COBOL 2.0 for Visual Studio 2010 Release Notes	5
System Requirements for Visual COBOL for Visual Studio	6
Installing Visual COBOL for Visual Studio	8
Licensing Information	10
What's New	11
New Features in Visual COBOL 2.0	11
Building Projects to Multiple Output Files	11
Compiler Directives	11
Debugging Enhancements	11
Library Routines	12
Managed COBOL Language Features	12
SQL Support	13
Run-Time Tunables	13
Samples	13
Vision Data File Searching	14
XML Support	14
Features Added in Visual COBOL 2010 R4 Update 2	14
Documentation for the Dialog System AddPack	14
OO COBOL Class Library Reference	14
Net Express Project Import Wizard	15
Features Added in Visual COBOL 2010 R4	15
ACUCOBOL-GT Compatibility	15
Creating Projects from Selected Files	16
Debugging Enhancements	16
Documentation	16
Embedded HTML	16
Improvements to the Implements Smart Tag	16
Language Improvements	16
Large Projects Support	16
New Compiler Directives	16
New Samples and Tutorial	17
Project Details Window	17
Project Properties Updates	17
RM/COBOL Compatibility	17
Smart Linkage	18
WCF Services and Service References	18
XML Extensions	18
Features Added in Visual COBOL 2010 R3	18
.NET COBOL Syntax Improvements	18
Creating Projects from Existing Code	21
Add Existing COBOL Items Wizard	22
Override Class Members Dialog	23
Smart Tag for Implementing Interfaces	24
Snippet for Implements	24
Navigate To	24
Find All References	25
Web Application Projects	25
Debugging	26
Samples Browser	27
ACUCOBOL-GT Compatibility and RM/COBOL	28

XML Support	29
Features Added in Visual COBOL 2010 R2	29
File Handler	29
Go To Procedure Division	29
OpenESQL Assistant	29
Samples	30
Snippets	30
SQL Support	30
XML Parse/Generate	30
Features Added in Visual COBOL 2010 R1	30
Visual Studio as the Core Integrated Development Environment	30
COBOL 2010	31
COBOL Language Extensions	31
Known Issues	32
Resolved Issues	35
Updates and SupportLine	44
Further Information and Product Support	44
Information We Need	44
Creating Debug Files	44
Disclaimer	46

Micro Focus Visual COBOL 2.0 for Visual Studio 2010 Release Notes

These release notes contain information that might not appear in the Help. Read them in their entirety before you install the product.



Note: This document contains a number of links to external Web sites. Micro Focus cannot be responsible for the contents of the Web site or for the contents of any site to which it might link. Web sites by their nature can change very rapidly and although we try to keep our links up-to-date, we cannot guarantee that they will always work as expected.



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

System Requirements for Visual COBOL for Visual Studio

Visual COBOL has the following requirements in addition to those of Visual Studio:

Hardware Requirements

Operating Systems

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

Disk Space

Approximately 42 Mb for the Sentinel RMS license server
Approximately 240 Mb for Micro Focus COBOL integration into Visual Studio. Note: This disk size includes the space needed to cache information locally so that you can modify the installation without the original source media.

Software Requirements

We recommend that you install this product using the Web Installer. The Web Installer will install any missing prerequisite software and the product on your machine.

If you choose to install using the full image, you need to install the following software beforehand:

- Microsoft Visual Studio 2010 (Premium, Professional, or Ultimate)
or
- Microsoft Visual Studio 2010 Integrated Shell

You can download the Visual Studio 2010 Integrated Shell from the [Microsoft Download Center](#).



Note: If you download the shell you need make sure that you run the installer to complete the installation. To do this run `vsintshell.enu.exe` from the location where you installed the download.



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

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

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

Installation Restrictions

- Visual COBOL and COBOL Server cannot co-exist on the same machine. This applies to Windows only.

Installing Visual COBOL for Visual Studio



Note:

- This version of the product is a full install.
- It is recommended to install this product using the Web Installer. The Web Installer helps you install any missing prerequisite software and the product on your machine.

Downloading the Product

1. Click on the product download link in your delivery notice.

This opens the Micro Focus Electronic Product Delivery page for your order.

2. Before downloading your product, check the **Legal Information** tab and read the End User License Agreement.
3. Check the **Documentation** tab and read the documentation supplied for your product.
4. Check the respective links to download the setup file or the Web installer for your product to your machine.

Installing

To use the Web Installer:

1. Double-click the `VisualCOBOLVisualStudio20_webinstaller.exe` file.
2. Click **Start** in the Web Installer dialog and follow the instructions to install the prerequisite software and the product.

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

1. Run the `VisualCOBOLVisualStudio20.exe` file and follow the wizard instructions to complete the installation.



Note:

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

Installing as an Upgrade

This release does not work concurrently with earlier releases and installing it uninstalls previous versions of the product.

- Visual COBOL and COBOL Server (or COBOL Server for Unit Testing) cannot co-exist on the same machine.

- If you are installing this as an update to an earlier version of the product, make sure that the product is not running on your machine when you start the installation.
- To install the product, you need to be logged in with a user-ID that has write access to the registry structure under HKEY_LOCAL_MACHINE, HKEY_CLASSES_ROOT, and HKEY_CURRENT_USER so the installation software can set the environment appropriately. You also need to be logged on with Administrator privileges.

After Installing

You are now ready to run Visual COBOL. From the Windows taskbar click **Start > All Programs > Micro Focus Visual COBOL > Visual COBOL for Visual Studio 2010**.

Please refer to the *Start Here* and *Product Information* sections in your product Help. Here, you will find information on getting started including tutorials and demonstration programs.

If you have problems trying to view the Micro Focus help, ensure that the Visual Studio Help Library is pointing to local help. From the Visual Studio menu click **Help > Manage Help Systems > Choose online or local help** and check the **I want to use local help** button.



Note: The first release of Visual Studio 2010 has a browser-based help system, Microsoft Help Viewer 1.0, which does not include an index for the locally-installed help. Navigation of the content is only available using the table of contents and Search and the help contents for the Help system does not expand and collapse in the same way as previous Help systems.

Visual Studio 2010 SP1 provides an upgrade of the help system, Microsoft Help Viewer 1.1, which provides a stand-alone help viewer with an index and a fully expandable table of contents.

If you do not wish to install Visual Studio 2010 SP1, you can install some third party tools that enable the index or the fully expanding table of contents. Read <http://kb.microfocus.com/display/4/kb/article.aspx?aid=31484> for more.

Licensing Information



Note:

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

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

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

The dialog box for the Micro Focus License System Administration Tool will be displayed.

2. Click the **Authorize** tab.
3. Enter your Authorization Code and click **Authorize**.
4. Close the Micro Focus License System Administration Tool.

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

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

License for the Integrated Development Environment for Visual Studio

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



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

License for COBOL Server for Unit Testing

You need to install this license to run and test your applications in the runtime environment. It is a timed license and expires after one year after the date of purchase. After this time you need to contact Micro Focus to request a renewal.

There is no evaluation period for this license and you need to use your Authorization Code to obtain a full license before you can test your applications. The Authorization Code can be found in the **Requested License Key** email that is supplied when you order your product. To obtain a full license use the code that is listed under **Visual COBOL - COBOL Server for Unit Testing**.

What's New

The following sections outline the new features that have been added in this release of Visual COBOL for Visual Studio 2010.

New Features in Visual COBOL 2.0

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.
- **ILUSING** - when set on a single file using the **SET** statement, `$set ilusing`, the directive only affects that file.

Debugging Enhancements

This release 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.

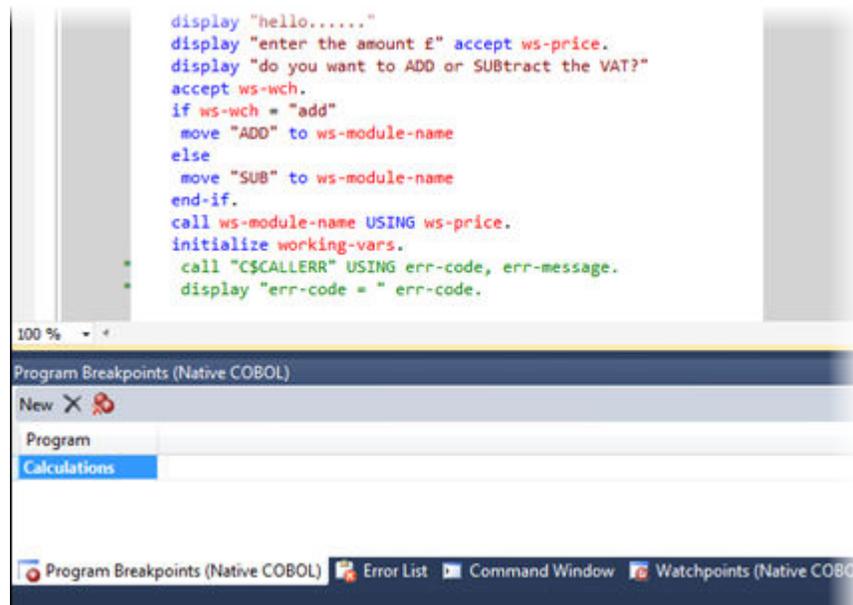
Debugging native COBOL Link Library projects You can now debug native COBOL link library projects that get built to a single .dll file.

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)**.



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

Visual COBOL 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.

Library Routines

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.

SQL Support

This release provides the following enhancements:

- Improved IDE integration with SQL directives - now supports handling of deprecated and removed directives. Also supports filtering of the choices offered to the user by product type, project type, and platform.
- OpenESQL has been enhanced and it now:
 - defaults to optimal performance
 - supports 64bit ODBC across all platforms

Run-Time Tunables

This release 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

This release provides the following new ACUCOBOL-GT compatible environment variables to help search for Vision data files at run time:

```
APPLY_FILE_PATH
FILE_CASE
FILE_PREFIX
FILE_SUFFIX
```

XML Support

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

Features Added in Visual COBOL 2010 R4 Update 2

Documentation for the Dialog System AddPack

Documentation for the Dialog System AddPack is now available and is integrated with the Visual COBOL documentation.

This documentation describes the AddPack, which enables you to modernize Dialog System applications within Visual COBOL. You migrate an application to Visual COBOL 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` or as `.`. See the Managed Customer sample `ManagedCustomer.sln`.



Note: The Dialog System AddPack is not part of Visual COBOL or COBOL Server. It is separately installable and available from Micro Focus SupportLine

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

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

Net Express Project Import Wizard

Visual COBOL 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 Visual COBOL 2010 R4

ACUCOBOL-GT Compatibility

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

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

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

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

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

Creating Projects from Selected Files

A new option, Create Project From Selection, is now available for your projects in Solution Explorer. You can select a number of COBOL files and copybooks in your project and opt to create a new project from them in the same solution.

Debugging Enhancements

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

Documentation

If you are using Visual Studio 2010 Service Pack 1, the help is displayed in a stand-alone help viewer with an index and a fully expandable table of contents.

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.

Improvements to the Implements Smart Tag

The implements smart tag now supports value-types in addition to classes.

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.

Large Projects Support

Visual COBOL has been optimized to work with bigger, more complex applications. This includes faster processing of multiple files and various IDE features that facilitate the process of developing large-scale project.

You can quickly move existing COBOL code into Visual Studio with the help of various wizards and windows such as the Create Project from Existing Code wizard and the Create Project from Selection wizard. The IDE is preconfigured so that during the file import it automatically scans the files and sets Compiler directives on them as appropriate.

New Compiler Directives

The following new Compiler directives are provided:

- ILCUTPREFIX - removes a specified prefix from the names of the COBOL data items in your source code.
- ILSMARTLINKAGE - exposes the Linkage Section and entry points to managed code by creating types.
- RUNTIME-ENCODING - determines the runtime encoding.
- SOURCE-ENCODING - passes the encoding of the source program to the Compiler.

New Samples and Tutorial

New samples and a new tutorial showing how to create WCF services in COBOL are available.

Project Details Window

A new window, Project Details, is available for your COBOL projects and solutions showing a complete list of the files in a project or a solution and various file details. You can open the window from the context menu for a project or a solution in Solution Explorer.

Project Properties Updates

The project properties pages have been restructured to make setting options more intuitive.

RM/COBOL Compatibility

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

- The following RM/COBOL standard library routines can now be used with Visual COBOL in native code:
 - C\$Century
 - C\$ConvertAnsiToOem
 - C\$ConvertOemToAnsi
 - C\$DARG
 - C\$Delay
 - C\$GetEnv
 - C\$GetNativeCharset
 - C\$LogicalAnd
 - C\$LogicalComplement
 - C\$LogicalOr
 - C\$LogicalShiftLeft
 - C\$LogicalShiftRight
 - C\$LogicalXor
 - C\$NARG
 - C\$SetEnv
 - C\$RERR
 - DELETE
 - RENAME
- The RM/COBOL file handler can now be used with Visual COBOL, enabled by using the CALLFH(ACUFH) Compiler directive, and then configuring an add-on to the Vision file handler.

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

Smart Linkage

Exposing COBOL group items as managed types

You can expose COBOL Linkage sections to other managed languages by using the ILSMARTLINKAGE directive. Smart Linkage saves the need to edit your original COBOL code or write wrapper classes.

WCF Services and Service References

Support is now available for adding WCF services as service references to your COBOL projects.

Note: WCF is not supported in the Visual Studio Shell but adding service references for client applications is supported.

XML Extensions



Note: This functionality is supported in native COBOL only.

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

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

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

Features Added in Visual COBOL 2010 R3

.NET COBOL Syntax Improvements

Quoteless syntax

Quotes are not needed when defining types, classes or methods, or when invoking classes and methods.

Construct improvements

The structure of class-id, method-id, enum-id, delegate-id, interface-id, valuetype-id has been improved.

Environment division, Configuration section, Repository

The Environment division, Configuration section and the Repository are no longer needed.

Static, Factory and Object blocks

The Static, Factory and Object blocks are no longer needed.

Attributes, Custom-Attribute and Class-Attributes

CUSTOM-ATTRIBUTE is now replaced by the ATTRIBUTES phrase. You no longer need to define class-attributes. Instead, specify the class custom attributes in the class definition.



Tip:

- Visual COBOL supports the older syntax, so projects that are using it will still compile. However, it is recommended to create applications using the new syntax and adhere to the *.NET COBOL Best Practices*.

- It is recommended to use the COBOL project and file templates, snippets and Intellisense as they use the new syntax. To see the new syntax in action, check the Visual COBOL samples.

The following is a more detailed overview of the changes in the syntax with examples:

Quoteless Syntax

Quotes are no longer needed when you define types, classes or methods, or when you invoke classes and methods. For example:

New Syntax	Old Syntax
01 01 type MyClass	01 01 type "MyClass"
type MyClass::New	type "MyClass"::"New"
set o to new MyClass	set o to new "MyClass"
set class::Property to value	set "class"::"Property" to value
set return-value to class::Method(param1)	set return-value to "class"::"Method"(param1)
invoke class::Method(param1)	invoke "class"::"Method"(param1)
<pre> method-id MyMethod public. local-storage section. procedure division. goback. end method. </pre>	<pre> method-id. "MyMethod" public. local-storage section. procedure division. goback. end method "MyMethod". </pre>

Construct of class-id, method-id, enum-id, delegate-id, interface-id, valuetype-id

The construct of class-id, method-id, enum-id, delegate-id, interface-id, valuetype-id has been improved as follows:

- You do not have to type a period after the declaration (for example, method-id *MethodName*).
- Quotes are no longer required around names.
- You do not need to use the name in the end marker.

New Syntax	Old Syntax
<pre> class-id Namespace.MyClass. object-storage section. method-id InstanceMethod. local-storage section. procedure division. goback. end method. method-id StaticMethod public static. local-storage section. procedure division. goback. </pre>	<pre> class-id. MyClass as "Namespace.MyClass". environment division. configuration section. repository. static. working-storage section. end static. object. working-storage section. method-id. "InstanceMethod". </pre>

New Syntax	Old Syntax
<pre>end method. end class.</pre>	<pre>local-storage section. procedure division. goback. end method "InstanceMethod". end object. end class MyClass.</pre>

Environment Division, Configuration Section, Repository

You no longer need to use an Environment division, a Configuration section or a Repository paragraph. For example:

New Syntax	Old Syntax
<pre>program-id. Program1 as "MyProject.Program1". data division. working-storage section. procedure division. goback. end program Program1.</pre>	<pre>program-id. Program1 as "MyProject.Program1". environment division. configuration section. repository. data division. working-storage section. procedure division. goback. end program Program1.</pre>

Static, Factory and Object Blocks

The Static and Object blocks are no longer used. With the new syntax you need only one working-storage section for items that were defined in a static or object block under the old syntax.

To define a static method, use the STATIC word.

The following example shows how to define static methods with the new syntax and how to avoid using an object block:

New Syntax	Old Syntax
<pre>class-id Namespace.MyClass. working-storage section. 01 my-object-data pic x. 01 my-static-data pic x static. method-id InstanceMethod. local-storage section. procedure division. goback. end method. method-id StaticMethod public static. local-storage section. procedure division.</pre>	<pre>class-id. MyClass as "Namespace.MyClass". environment division. configuration section. repository. static. working-storage section. 01 my-static-data pic x. end static. object. working-storage section. 01 my-object-data pic x. method-id. "InstanceMethod".</pre>

New Syntax	Old Syntax
<pre> goback. end method. end class. </pre>	<pre> local-storage section. procedure division. goback. end method "InstanceMethod". end object. end class MyClass. </pre>

Attributes, Custom-Attribute and Class-Attributes

These are the changes for CUSTOM-ATTRIBUTE and class-attributes:

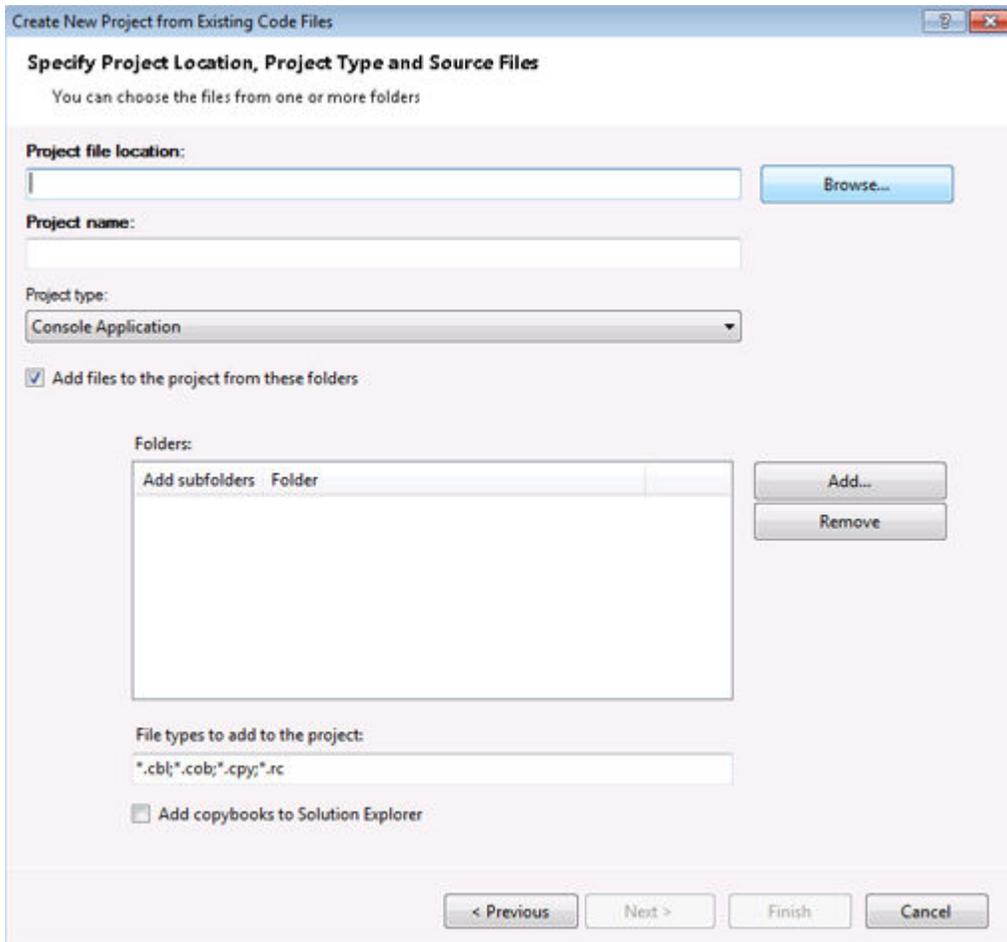
- The CUSTOM-ATTRIBUTE phrase is replaced by the ATTRIBUTE phrase.
- You no longer have to define `class-attributes`. Instead, specify the class custom-attributes in the class definition using the ATTRIBUTE phrase.
- Quotes are no longer needed around the name of the attribute and you can omit the word "Attribute" from the name.

For example:

New Syntax	Old Syntax
<pre> class-id MyNamespace.MyClass. attribute Serializable. working-storage section. ... end class. </pre>	<pre> class-id. MyClass as "MyNamespace.MyClass". class-attributes. custom-attribute is type "SerializableAttribute". object. working-storage section. ... end object. end class MyClass. </pre>

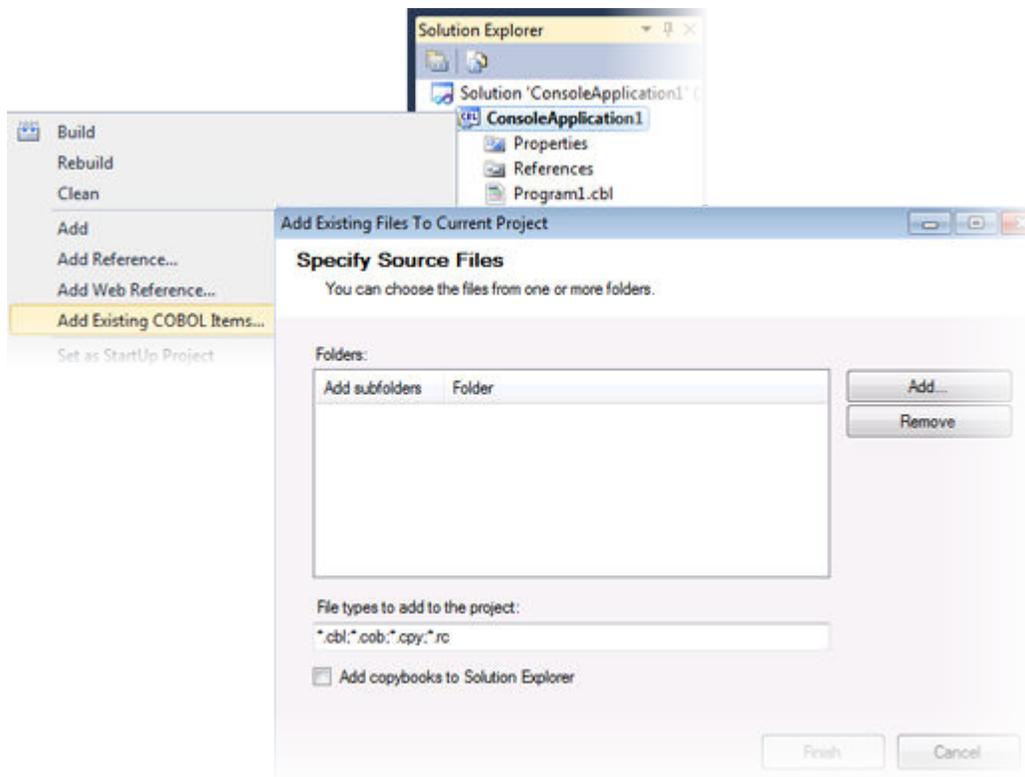
Creating Projects from Existing Code

Now you can create Visual Studio COBOL projects from existing applications using the Create New Project from Existing Code Files wizard. The wizard will create a new COBOL project and automatically add files to it from the specified directories. It will perform an automatic file scanning to identify which files are programs and copybooks, so that they can be correctly added to the project.



Add Existing COBOL Items Wizard

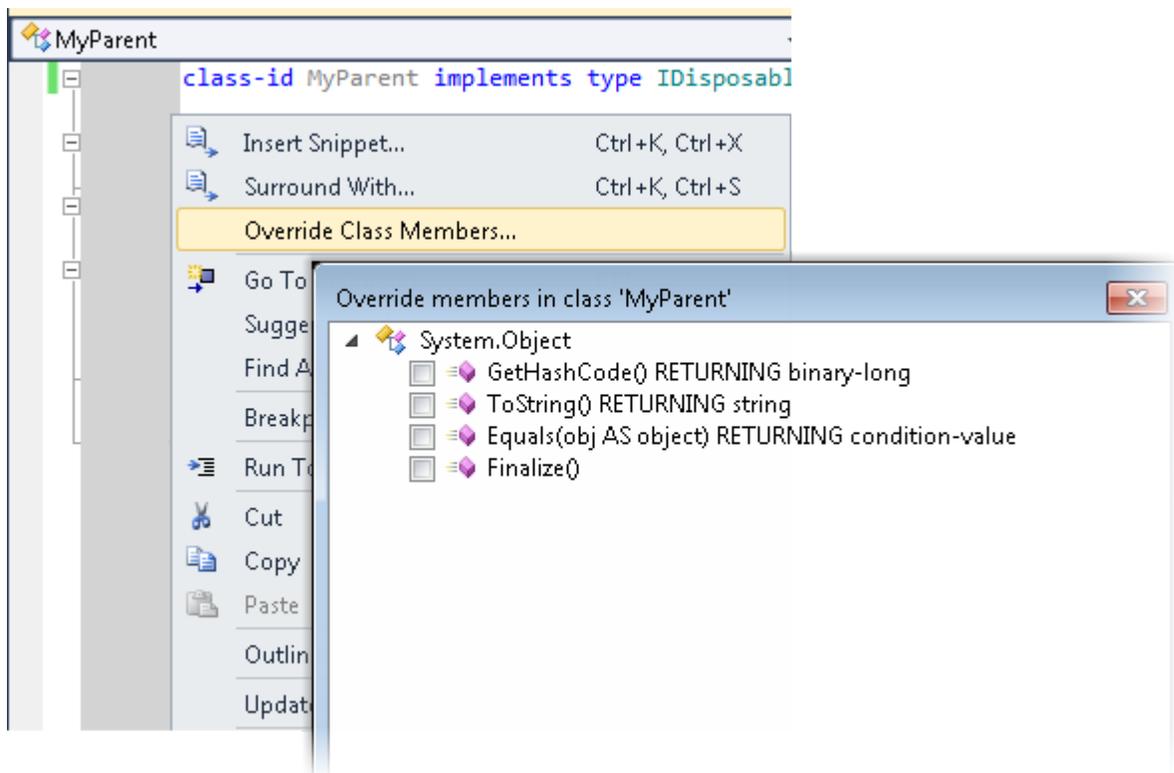
You can add existing COBOL files to your Visual Studio project using the new Add Existing COBOL Items wizard available from the context menu of the project in Solution Explorer. The COBOL files will be scanned to determine which ones are programs or copybooks, and then they will be added to the project.



Override Class Members Dialog

 **Note:** This feature works with .NET managed code only.

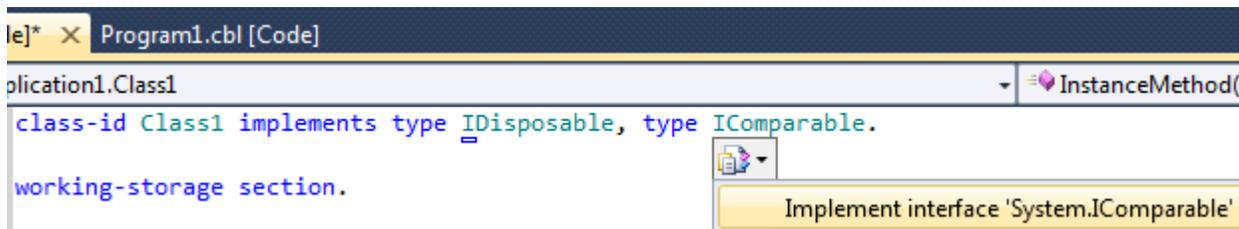
The new Override Class Members dialog available in the editor enables you to override the members of inherited classes. The dialog helps you see the base classes from which a class inherits, select the members to override and add the construct of the overriding methods to the class.



Smart Tag for Implementing Interfaces

 **Note:** This feature works with .NET managed code only.

You can now easily implement interfaces with the help of a Smart Tag. The tag appears underneath at the beginning of the declaration of any interface that is not fully implemented. To implement the interface, you simply need to click the tag.



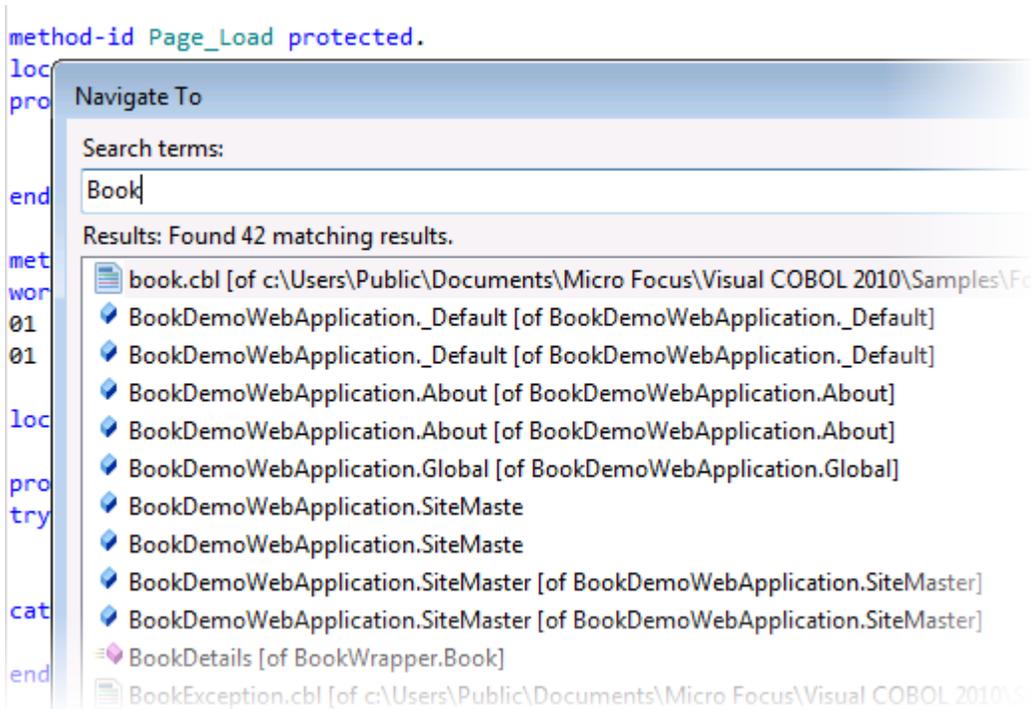
Snippet for Implements

 **Note:** This feature works with .NET managed code only.

The snippet for implements has been improved. It now automatically implements the members from an interface and has improved support for more complex method signatures.

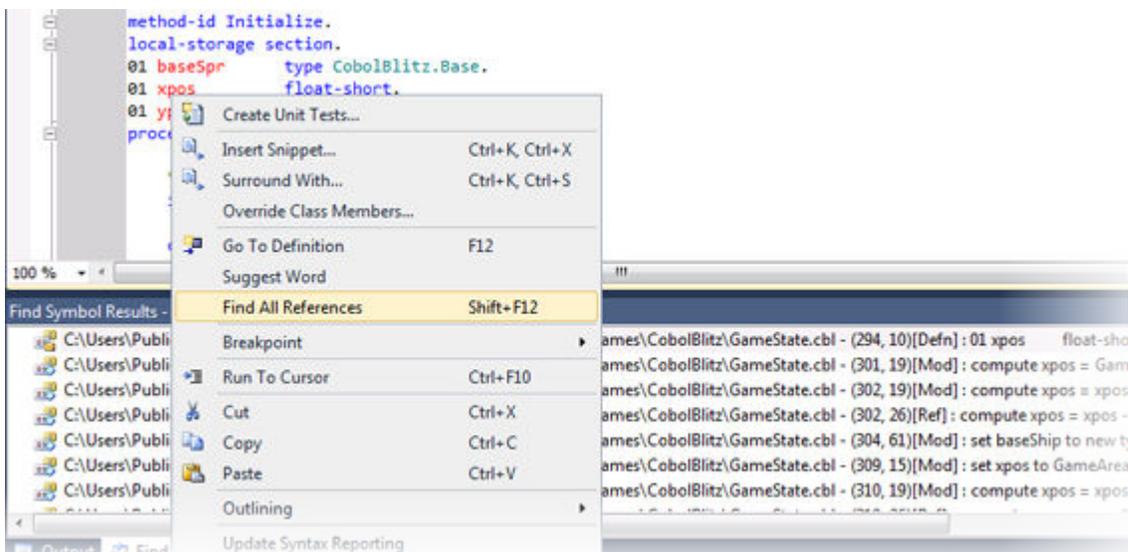
Navigate To

Use the Navigate To option in the Edit menu to search for files, variables and section names in all projects and files in your solution.



Find All References

The Find All References option available from the editor enables you to search for references of COBOL data items, section or paragraph names in your solution.



Web Application Projects

 **Note:** This feature works with .NET managed code only.

This release offers Web Application Project templates for creating COBOL Web applications and Web sites and applications. The benefits of using a Web Application project include:

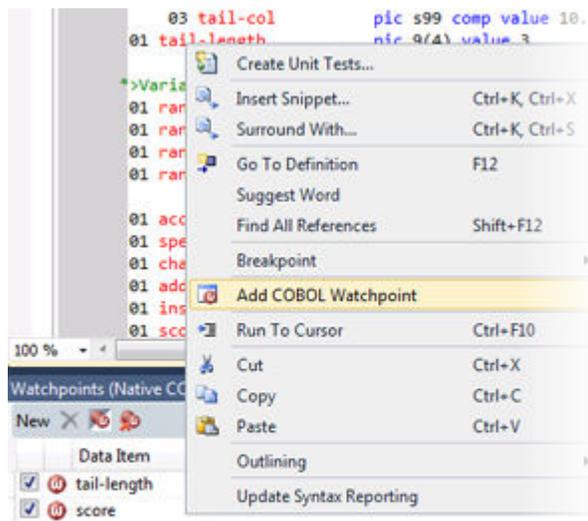
- A Web Application project includes a project file which enables you to specify what files are part of the project and should be compiled.
- It adds namespaces for all items of the project.

- The source code is compiled into a single assembly on your local machine and is then deployed to the IIS server. You don't have to deploy the code behind.
- A Web Application project includes a "Publish" option for deploying the compiled assembly to an IIS server directly from the IDE using the automated tools of Visual Studio.
- Supports the Visual Studio Code Analysis feature.

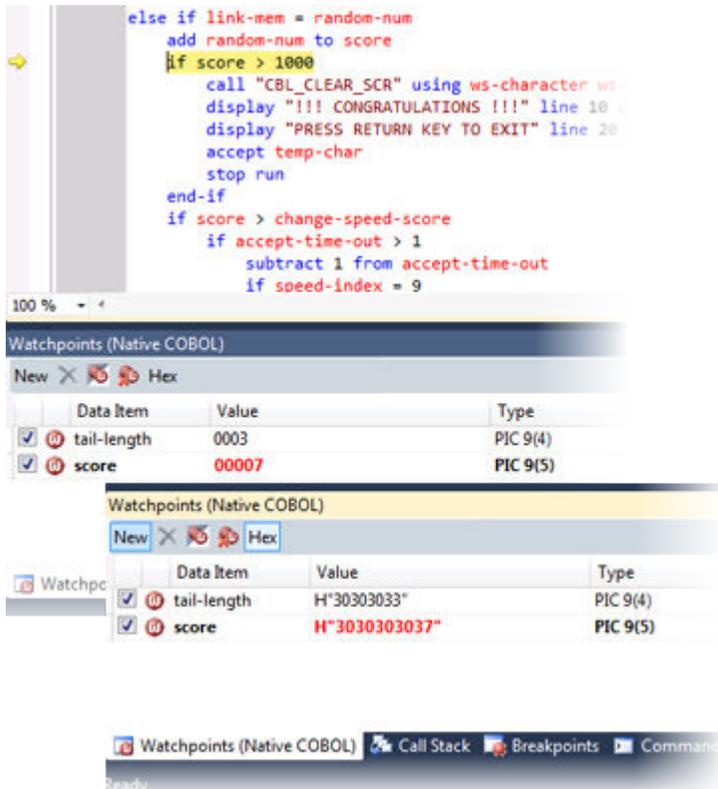
Debugging

The following debugging enhancements have been made:

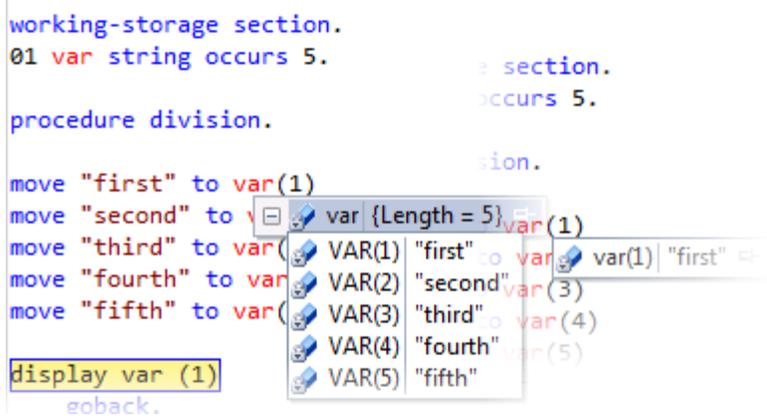
- COBOL watchpoints and break on data change - you can set COBOL watchpoints on individual data items in native COBOL. COBOL watchpoints enable you to watch the area of memory associated with the particular data item and help track memory corruption. When the memory changes, debugging stops on the line that immediately follows the line on which the data has changed. This feature works with native code only.



- **Watchpoints (Native COBOL)** window - enables you to manage the COBOL watchpoints you add to your applications and view the contents of the memory associated with each watchpoint. This feature works with native code only.

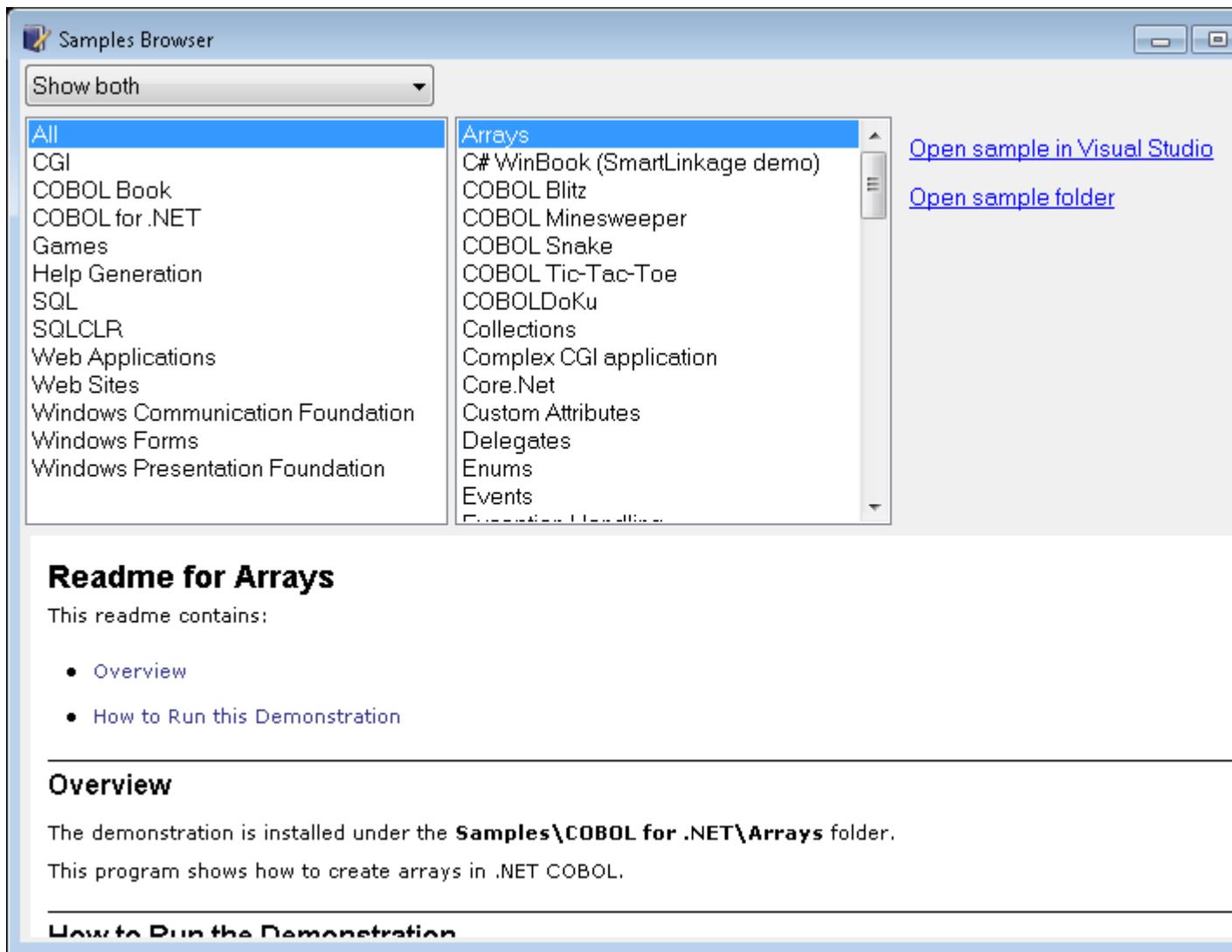


- Simplified remote debugging - a simplified process for setting up remote debugging is provided.
- Attach to 64-bit process and debug - provides the ability to attach to and debug 64-bit COBOL processes.
- Debug tooltip for OCCURS items - you can now specify whether the debug tooltips for OCCURS items should display all items in an array or the value of an expression.



Samples Browser

You can preview the samples and access them more easily with the help of the Samples Browser which is now available from the Start menu. Samples Browser lists the samples by category



ACUCOBOL-GT Compatibility and RM/COBOL

The Compiler and run-time now include initial support for ACUCOBOL-GT. This support is enabled by several new Compiler directives. 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:

- Vision indexed file system and utilities (vutil, vio, and logutil) support. Vision support is enabled by the new CALLFH(ACUFH) option.
- ACUCOBOL-GT compiler options. By using the new ACUOPT directive you can specify the same options available in ACUCOBOL-GT.
- ACUCOBOL-GT and RM/COBOL data type support. This enables you to mix ACUCOBOL-GT and Micro Focus Visual COBOL applications via data files or calls.
- Initial ACUCOBOL-GT syntax support. The Compiler now supports some of the ACUCOBOL-GT extensions.
- Interoperability between ACUCOBOL-GT and Visual COBOL components. You can begin to build applications that combine ACUCOBOL-GT with Visual COBOL features.

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

XML Support

Enhancements have been made to XML Parse/Generate to provide compatibility with IBM® Enterprise COBOL for z/OS® v4.2.

New in XMLGENERATE:

- ATTRIBUTES phrase
- NAMESPACE and NAMESPACE-PREFIX phrases
- XML-DECLARATION phrase

New in XMLPARSE:

There are now two modes of XMLPARSE support using the XMLPARSE() compiler directive:

- XMLPARSE(COMPAT) – provides compatibility with IBM Enterprise COBOL for z/OS v4.1 and earlier.
- XMLPARSE(XMLSS) – provides compatibility with IBM Enterprise COBOL for z/OS v4.2.

XMLPARSE(XMLSS) provides:

- ENCODING phrase
- RETURNING NATIONAL phrase
- VALIDATING phrase
- New special registers - XML-NAMESPACE, XML-NNAMESPACE, XML-NAMEPSACE-PREFIX and XML-NNAMESPACE-PREFIX.
- New behaviors – for example, different return codes, different output registers depending on the EVENT.

Note: The ability to parse XML documents one segment at a time with the help of the END-OF-INPUT XML event is not supported yet.

Features Added in Visual COBOL 2010 R2

File Handler

The Micro Focus File Handler is now provided as both verifiable and non-verifiable versions. Compiling your application with the ILVERIFY directive will automatically reference the verifiable File Handler assembly.

Go To Procedure Division

The Go To Procedure Division button is now available on the Go To Location toolbar. Clicking the button positions the cursor on a Procedure Division depending on the current context of the code.

OpenESQL Assistant

Support for the OpenESQL Assistant has been added. The OpenESQL Assistant is an interactive tool that enables you to easily design and build SQL queries and embed those queries into your COBOL code.

Features include:

- Prototype SQL SELECT statements and test them against a database
- Design SQL INSERT, UPDATE, and DELETE statements
- Insert SQL queries into the COBOL code
- Create and insert auxiliary code into your COBOL code

Samples

The following games have been added to the samples:

- COBOL Blitz - A shooter game in which the players use a laser cannon to defend themselves against the invasion of aliens troops. The goal is to destroy the troops and prevent them from reaching the bottom of the screen.

Special Features:

- 2D graphics
- Audio effects
- Snake - An arcade game in which the player navigates a long chain of symbols across the screen and scores by collecting numbers. Numbers add to the overall length and the speed of the snake. The player needs to avoid hitting the borders of the screen or touching the snake's body as this terminates the game.
- Tic-Tac-Toe - The player competes with the PC to place three identical marks in a horizontal, vertical, or diagonal row on the 3x3 board.

Snippets

This release provides new snippets for Attribute, DateTime, Implements and for static methods.

SQL Support

The SQL technology that was present in Net Express is now seamlessly integrated within the Visual Studio 2010 development environment. When you develop COBOL SQL applications in Visual Studio, you can use the same development environment to extend and modernize your COBOL assets.

Features include:

- OpenESQL technology that supports embedded SQL in your COBOL applications
- OpenESQL Assistant wizard that automatically generates embedded SQL in a COBOL program template given basic database information
- DB/2 ECM technology that uses embedded SQL to work with DB2 LUW
- COBSQL processor that provides native DBMS SQL support for Oracle's Pro*COBOL and other vendors

XML Parse/Generate

Visual COBOL now supports the IBM-style XML syntax and enables your applications to process XML data. Support for the XML PARSE and XML GENERATE statements is provided in the Visual COBOL compiler.

Features Added in Visual COBOL 2010 R1

Visual Studio as the Core Integrated Development Environment

The Visual Studio editor has been extended in a number of ways to enhance its support for COBOL, including Standard Visual Studio 2010 features for program navigation are exploited for COBOL applications.

- Fully integrated COBOL development environment delivers high programmer productivity by exploiting Visual Studio tools and providing instant feedback.
- Enhanced COBOL syntax for .NET programmers makes it easier for COBOL programmers to use .NET services or for programmers with .NET experience in other programming languages to be productive with COBOL.
- Visual COBOL supports the development and deployment of both "managed" .NET (with multi-targeting for .NET Framework V4 and earlier versions) and "unmanaged", native code applications.
- Visual COBOL is a part of the Visual Studio 2010 product portfolio from Micro Focus which also includes testing and developer productivity tools.
- COBOL Margins - visual indication of COBOL margins which are sensitive to the COBOL margin directive currently selected for the program – if the setting is changed via an embedded "\$SET SOURCEFORMAT" directive, then the display is immediately updated.
- COBOL sensitivity is extended to support COBOL methods and data items in IntelliSense and preconfigured "code snippets" reduce the effort required to complete code and avoid errors being introduced.
- Background parsing continuously ensures that the code being worked on will compile cleanly.

COBOL 2010

Visual COBOL supports the development of both "managed" code which is fully interoperable with other .NET languages and "native code". It is built on a new Micro Focus COBOL platform "COBOL 2010".

A standalone COBOL Server is available for deploying applications developed within Visual COBOL.

Visual COBOL provides a test license version of the COBOL Server to allow system testing before deployment into production.

COBOL Language Extensions

Historically, COBOL has been case-insensitive which makes interoperation with .NET methods more difficult than it should be. For example, method or member names had to be enclosed within quotation marks and declare synonyms to refer to external types. With Visual COBOL these restrictions have been removed and the code is more ".NET-like" while still retaining COBOL's traditional ease of understanding. Unnecessary COBOL elements such as "repository" have been made optional which greatly reduces the size and complexity of a COBOL .NET program. The language changes improve readability and simplify the learning process for existing C# or VB programmers who can easily work on the COBOL code. With this flexibility, teams can be more agile and thus reduce development and maintenance costs.

Known Issues

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

In addition, please note the following:

ASP.NET

It is not possible to run ASP.NET Web Sites or Web Services on a production machine (one that has Micro Focus COBOL Server installed) without an additional setup. This is because the production machine does not contain development tools such as the COBOL Compiler. To work around this, you need to do the following steps:

1. Precompile the site before you deploy it using the **Publish Web Site** command in Visual Studio.
2. Edit the `.asmx` file of the Web service project or the `.aspx` file of the Web site and delete the `Language="COBOL"` statement.
3. Edit the `Web.config` file with a text editor and delete the line which contains: `<compiler language="COBOL" . . .`
4. Ensure that a .NET Server license is installed using Aptrack.

-1080101

Co-existing with Earlier Micro Focus Products

Run-time system error due to COBCONFIG A run-time system error occurs if either the COBCONFIG or COBCONFIG_ environment variable is set when you run a Visual COBOL application or when you use Visual COBOL to edit or create projects and the configuration file it refers to contains entries that are not valid for Visual COBOL.

For example, this might happen if you have Net Express or Studio Enterprise Edition installed and either COBCONFIG or COBCONFIG_ is set for it.

To work around this issue, ensure that Visual COBOL is not running and then modify the configuration file by doing one of the following:

- If the invalid tunable is not needed by another application, remove it from the run-time configuration file.
- Add the following as the first line in the configuration file:

```
set cobconfig_error_report=false
```
- Unset COBCONFIG (or COBCONFIG_) or set it to another configuration file that does not contain the invalid tunable for the particular session you are running in.

Creating COBOL Projects from Selection

The documentation on creating COBOL projects from selection specifies that the newly created project has the same properties and references as the original project. You should also note that the new project has the default build configuration for the selected project type.

File Handling

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

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

Fileshare

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

Installation

- A bug in 64-bit Windows 7 may cause the display of the Compatibility Assistant dialog box during the installation of the product showing incorrectly that `as1mpclocate.exe` and `init2as1m.exe` are not compatible. The two utilities run properly and the installation is successful. To avoid receiving this notification, run Windows update and install update KB978637 before installing this product.
- If, when you start Visual Studio, you receive a message box about a Micro Focus package load failures, check that the Packages folder is on the PATH environment variable. If it is not, click No to disable loading the packages.

If you accidentally click Yes, you need to re-enable loading the packages, as follows:

1. Open a Visual Studio command prompt from the **Start** menu.
2. Enter the command:

```
devenv /ResetSkipPkgs
```
3. Reboot the machine.

This should resolve the problem. The PATH is set up correctly and the packages are found.

- Installing this release as an upgrade to a previous version of the product might take longer compared to installing the product for the first time.

Visual Studio IDE

- In your project properties, the Application page currently allows you to select any of the static methods in the application as a Startup object. This is incorrect. You should always set only the first static method or the program name as a Startup object in order for the project to build.
- In Visual Studio, the controls on the COBOL page in the project properties are not displayed for a WPF project. To workaround this issue, open the Application page in the project properties and from the drop-down list under Output type choose the same output type that was already selected. Close the project property pages and save the project. The next time you open the project properties, the COBOL page is properly displayed.
- When compiling a native COBOL application in Visual COBOL for Visual Studio which contains resource files, you may receive compiler error "cannot open include file 'pshpack2.h'". To workaround this problem, ensure you have the Windows SDK installed on your machine, add the include folder in the Windows SDK installation (by default, `%ProgramFiles%\Microsoft SDKs\Windows\v7.0A\include` on Windows 7) to your include path, and recompile. Note: The Windows SDK is available as a free download from Microsoft's Web site.

Visual Studio Shell

The following restrictions apply if you are using the Visual Studio Shell:

- On Windows XP, when building a managed COBOL application in Visual Studio, you may receive a dialog box with error "The application failed to initialize properly (0xc0000005)". If you are running Symantec antivirus runtime protection then this might be the cause for this issue. To resolve it, [click here](#) to download a fix from Symantec.
- WCF is not supported so the WCF demonstration programs do not work and the project templates for WCF are not installed.

- There is no integrated designer for icon and bitmap files.
- The Windows SDK must be installed.
- Examples and demonstration programs that use languages other than COBOL do not work because Visual Studio Shell doesn't support any other languages. This applies to the following demonstrations:
 - CursorDemo
 - InterfacingWithStdCOBOL
 - LobDemo
 - OrderStatusDemo

Resolved Issues

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

- [.NET Compiler](#)
- [.NET ESQL Support](#)
- [.NET RTS](#)
- [.NET WinForm/WebForm](#)
- [Compiler](#)
- [Compiler ECM Support](#)
- [Documentation](#)
- [File Handling - External File Handler](#)
- [File Handling - Fileshare](#)
- [File Handling - Rebuild](#)
- [File Handling - Sort](#)
- [MFSupportInfo](#)
- [OLE Domain Support](#)
- [RTS](#)
- [Setup Issues](#)
- [SQL Support \(non-Compiler\)](#)
- [SQL: DB2 ECM](#)
- [SQL: OpenESQL](#)
- [Vision File System](#)
- [Visual Studio IDE](#)
- [XML Extensions](#)

.NET Compiler

- COBOL programs no longer give compilation errors when there are multiple periods at the end of PROCEDURE DIVISION.
1078852 (2509767)
- 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.
1078845 (2507380)
- Attributes that use the TYPE OF operand produced invalid MSIL and generated an exception when accessed through reflection APIs.
1079810 (2522328)
- Certain structures of the SCREEN SECTION records could lead to generation of invalid managed code and caused the build to fail with error "COBCH0857: System error - failure during ILASM phase".
1080096 (2524955)
- 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.

579979 ()

- .NET COBOL projects which included concatenation of level 78 numeric literals could not be built and caused the IDE to crash.

1080853 (2534541)

- The error message "COBCH0857 : System error - failure during ILASM phase" was sometimes produced for class programs containing multiple methods with declaratives and/or PERFORM statements.

1080854 (2534542)

- Compiling OO programs with matching methods where the method parameters had generic items sometimes failed with error COBCH0956, "Method 'MethodName' not found in inherited class", during the conformance checking phase.

1081338 (2538683)

- An incorrect decimal alignment occurred for arithmetic expressions involving a COMP-2 operand and a PIC S9(12)V99 operand in .NET COBOL.

1081334 (2540303)

- In .NET managed COBOL, SCREEN SECTION fields that use a numeric edited with decimal points displayed the number incorrectly, not taking into consideration the decimal point.

1081393 (2540560)

- When compiling for managed code, an INSPECT statement of the form "INSPECT data-item TALLYING n1 FOR TRAILING SPACE" produced an incorrect answer when data-item contained all spaces.

1082345 (2551352)

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

1082730 (2555177)

- In managed COBOL, a level 78 item appearing as the first item under the file description (FD) for an EXTERNAL file could result in a bad code generation.

1082841 (2557615)

- In some cases, when adding the result of a division operation to a numeric item with an implied decimal point, the final value could be incorrectly rounded.

1083008 (2559639)

.NET ESQL Support

- The ODBC wrapper interface to the native SQL runtime was calling a method incorrectly. As a result you could have received an exception when running managed UI applications calling the native SQL runtime system.

1083167 (2561658)

- In some cases OpenESQL did not mark an open cursor as non-purgable from the prepared statement cache and this could result in unexpected errors on subsequent fetches from the cursor.

1081741 (2543910)

- 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.

586534 ()

.NET RTS

- Loading COB32API by .NET managed COBOL code more than once caused a memory leak.

586885 ()

- CBL_EXEC_RUN_UNIT_CALL was not working correctly in managed code when the flag was set to create a new console.

1082055 (2548658)

- Managed ADIS applications were corrupting the screen if you run them using the Microsoft Telnet client.

1081681 (2544298)

- The .NET console processed numeric keys incorrectly when Caps Lock was enabled.

1080669 (2532347)

- In COBOL for .NET, MOVE of a BigDecimal field to 01 level PIC SV999 COMP-3 item was giving an incorrect result when truncation was involved.

1081198 (2536415)

.NET WinForm/WebForm

- 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.

1079984 (2524313)

- 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.

1080978 (2536071)

Compiler

- In constructs of type "STRING item1 item2 INTO identifier1", if the identifier uses a reference modification, the length from the picture clause definition was incorrectly used when determining if item1 and item2 fit into identifier1.

586313 ()

- Invalid syntax errors are no longer produced when compiling screen section items.

1082549 (2549214)

- ACUCOBOL-GT reporting has been enhanced and now produces a Compiler error for the following syntax which is not yet supported: ACCEPT... CURSOR <data-item>.

1081729 (2544855)

- An XML GENERATE statement on a record containing an ambiguous name whose case is distinct from the other declarations of that name now executes as expected.

1081691 (2543133)

- The ACUCOBOL-GT syntax ACCEPT ... DISPLAY OMITTED previously caused an internal Compiler error.

1081423 (2538935)

- Compiling a program which contains an XML GENERATE statement to exe no longer produces unresolved external symbol errors.

1078019 (2498657)

- Illegal ODO items which reference many levels of qualification now display a syntax error.

1080616 (2530541)

- Executing a statement using an intrinsic function with a reference modified argument when compiled with SSRANGE(2) no longer produces spurious run-time errors.

1080275 (2526877)

- The RECORD-POSITION operator returned a value that was one less than the correct value.

586672 ()

Compiler ECM Support

- A declaration of SQL TYPE ROWID was sometimes not processed correctly.

1080477 (2529596)

Documentation

- The following run-time system tunable is available: subsystem_cancel_mode. Specifies whether logical or physical cancels are used for CBL_SUBSYSTEM cancels. Syntax:

```
>>-----set subsystem_cancel_mode---.-0-.----->< +-1-+
```

Parameters: 0 Physical cancels: any programs canceled with a CBL_SUBSYSTEM cancel will be freed from memory, making more memory available. 1 Logical cancels: any programs canceled with a CBL_SUBSYSTEM cancel will not be freed from memory. The programs are available to be reused and hence the performance of CALL/CANCEL statements is greatly increased. As far as your program is concerned, the behavior of logical and physical cancels is identical. Properties: Default: 0 on Net Express; 1 on Server Express

1080440 (2529528)

- This release requires version 10000.2.660 or later of the Micro Focus licensing software. For local servers, you do not need to install it separately, as the setup file will install a new client (Visual COBOL) and a new licensing server on the same machine. If you have a network server, you must update the license server before installing the product as the client is not able to communicate with license servers of versions older than 10000.2.660. You can check the version of your license server by clicking **Help > About** in the Micro Focus Licensing System Administration tool.

1082175 (2549745)

File Handling - External File Handler

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

1079491 (2518330)

- Dataset concatenations ending with DD DUMMY (or DSN=NULLFILE) no longer cause an abend when the file is closed.

1078960 (2511384)

- When opening a file for I-O under mainframe emulation, the file does not have to exist.

1078362 (2501743)

- A series of concatenated files will now all be opened with the same opcode so it is treated as a single file.

1078617 (2503276)

- Missing indexed files in the USE statement incorrectly resulted in a 114 error. A 9/013 I/O error is now correctly displayed.

583179 ()

- SORT did not display correctly the report fields (such as sections, headers, trailers) when no OUTREC was specified.

583227 ()

- An invalid input detected in the SORT FIELDS statement caused the application to crash. A 114 error message was incorrectly displayed. This has been changed and an error "Command statement error(s) detected" is now displayed.

583425 ()

- Setting STRIPSPACE to OFF now correctly honors the record length and allows trailing spaces for LSEQ files.

573921 ()

- Support has been added to allow trailing spaces in LSEQ files when used with the variable STRIPSPACE=OFF, when printing records via stdout or piping to a file.

585740 ()

- Sequential files on a 64-bit machine are now correctly padded to fill the LRECL.

586461 ()

- A memory leak could occur when performing file I-O operations inside a Web service.

1081398 (2540611)

File Handling - Fileshare

- An fsview close file now correctly closes the file.

1079724 (2508855)

- A CICS close on a file no longer fails if a transaction has started and updates are made to any file.

1079738 (2473753)

- Using a READ statement after a CLOSE REEL through Fileshare could result in a 9/077 error.

587474 ()

File Handling - Rebuild

- If the checkpoint LAST is set and no checkpoint exists, BATCH BACKOUT now backs out of the entire log.

1081493 (2541220)

- Rebuild will no longer try to rebuild an indexed file without any keys, and will report the failure. Rebuild now also reports that there are no keys when performing file validation or file information on the file.

1078000 (2499296)

File Handling - Sort

- The NUM function has been added to SORT.

1079136 (2512764)

- The TOOLMSG output "Number of records resulting from criteria:" now displays the number of records correctly.

1079280 (2513886)

- Users can now use the DATEn+r field constants as well as comparing DATE constants within the INCLUDE/OMIT statements.

1078273 (2501816)

- NODETAIL now correctly prevents data records from being displayed in the OUTFIL. SORT will also correctly default to displaying the entire input record if neither NODETAIL or an OUTREC is specified.

1074166 (2443710)

- COUNT will now work correctly, representing a count of the number of data records in that section. The value of SUBMAX will now be output when defined in the OUTFIL.

1080528 (2443710)

- SORT will now regard the next line as a continued comment (and will therefore ignore everything on that line) if the previous line is a completed statement and has a comment in column 72.

1080262 (2527579)

- Using TOTAL or SUBTOTAL will now display the correct value instead of truncating the result.
584223 ()
- 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.
1080092 (2525670)
- An RTS 252 error no longer occurs if RTS memory checking is enabled and outfil processing is used in a sort card.
1081337 (2540109)
- Setting SORTSPACE to the same value as SORTTEMPSPACE no longer causes an RTS 252 to occur.
1081277 (2539633)
- SORTCOMPRESS=5 will no longer cause the records output from SORT to differ from the records that are taken in.
1080006 (2518730)
- MFSORT will now accept the following constants in the outrec, header, and footer: * DATE1+n * DATE1-n * DATE1(c)+n * DATE1(c)-n * DATE2+n * DATE2-n * DATE2(c)+n * DATE2(c)-n * DATE3+n * DATE3-n * DATE3(c)+n * DATE3(c)-n * DATE1P+n * DATE1P-n * DATE2P+n * DATE2P-n * DATE3P+n * DATE3P-n where n represents the number of days for DATE1 and DATE3, and number of months for DATE2, and where c represents the date separator.
1079771 (2521778)
- The keywords CENTWIN and Y2PAST are now both accepted under both DFSORT and SYNC SORT emulation.
585369 ()
- Sort now closes down correctly after an RTS 163 error, and is now able to carry on and process further jobs.
1082133 (2544148)
- When combining OUTFIL and SORTOUT files where headers and trailers were used, SORTOUT sometimes contained mainframe printer control characters.
1082048 (2548488)
- When writing to EBCDIC files, DATE constants will now correctly be converted into EBCDIC format.
586917 ()
- A SORT performed from REXX no longer causes a COBRTS 114 error.
1081942 (2546794)
- SORT will now correctly compare INCLUDE/OMIT parameters when SUM FIELDS=NONE.
1083014 (2558088)
- Setting MFJSSTRICTSORT=ON will now cause records that are shorter than the output file's LRECL to be padded with low values instead of spaces.
1081919 (2547118)
- If a SUBSYS is specified and does not exist, SORT reverts to using EXTFH and emits a warning message.
1081570 (2542992)
- PARM RC16 is ignored but the comment line reports a warning.
1081203 (2537395)

MFSupportInfo

- MFSupportInfo now supports machines with greater than 4Gb of memory installed.
1076718 (2483333)

OLE Domain Support

- In-process COM objects no longer leak memory when used within Microsoft's Windows Communication Foundation (WCF) framework.
1078908 (2509742)

RTS

- The Micro Focus run-time system is no longer unloaded prematurely when using IBM TXSeries, and does not produce a COBRT 173 error.
1079371 (2516765)
- The run-time text window no longer returns incorrect data when content is copied from it using Edit/Mark.
1078975 (2511669)
- On Japanese Windows, the IME now consistently appears at the current cursor position when ANSI ACCEPTs are done in the COBOL Text Window.
1079925 (2523680)
- Use of PC_PRINTER_DRAW_RECTANGLE, PC_PRINTER_GET_BMP_INFO, PC_PRINTER_LINE_FROM_TO, PC_PRINTER_SET_BACKGROUND_MIX_MODE, PC_PRINTER_SET_PEN or PC_PRINTER_SET_POSITION no longer give an unresolved symbol error when linking as a 32-bit application.
1072571 (2418196)
- 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.
588615 ()

Setup Issues

- The Visual COBOL R4 Web installer did not detect an existing installation of Visual COBOL R3 in a custom directory. It removed Visual COBOL R3 and installed Visual COBOL R4 in the default location without a prompt.
1080182 (2524759)

SQL Support (non-Compiler)

- The ODBC pre-compiler has been modified to properly support UTF-8 coded host variable names in EXEC SQL statements.
1079019 (2512255)

SQL: DB2 ECM

- The DB2 ECM has been updated to resolve run-time errors returned when compiling against mainframe databases in 64-bit mode.
1082441 (2549058)
- The Host Compatibility Option was updated to honor database connection criteria specified on the command line.

1081585 (2537570)

SQL: OpenESQL

- OpenESQL has been updated to support 64-bit operation for INSERT and FETCH statements.

1079765 (2499575)

Vision File System

- There is a new configuration option - RMFM_RUN_INDEX_FILES_PREVENT_WRITE_STARVATION. The default value of this option is set to NO, but when set to YES, it enables the new additional locking scheme that prevents write starvation when there are many active readers of the same file.

589086 ()

- ACUFH incorrectly returned status 00 on OPEN INPUT of a non-existent OPTIONAL file. This has been fixed so that ACUFH returns the correct status 05.

1082400 (2552367)

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

1079788 (2521519)

Visual Studio IDE

- In some menus and dialogs in the IDE copybooks were referred to as copy files. They are now consistently referred to as copybooks.

583180 ()

- If you specify copybook files without an extension, you no longer get an infinite recursion of including copybooks. A message with file and copybook name is shown in the Output Window and logged if the Logger is activated.

1081461 (2541871)

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

1080950 (2535705)

- Adding a new role to an Azure project resulted in a DISP_E_MEMBERNOTFOUND exception.

587104 ()

- Debugging, editing and rebuilding COBOL code sometimes produced build errors with an I-O error on IDY files.

587108 ()

- Native COBOL projects failed to compile if they included files with names containing spaces.

587349 ()

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

1082073 (2548551)

- When debugging native COBOL using the Just-In-Time debugger, the IDE was not releasing IDY files thus causing subsequent builds to fail.

587053 ()

- 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.

587365 ()

- The Net Express Project Import utility did not work correctly when you tried to convert Net Express projects located in network directories specified with a UNC directory name.

587710 ()

- Adding a watch on an item of the form buffer(1:size) caused the IDE to hang when "size" was not initialized.

587977 ()

XML Extensions

- XML EXPORT failed to output some occurrences from an OCCURS data item nested in another OCCURS data item.

1080752 (2533412)

Updates and SupportLine

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

Further Information and Product Support

Additional technical information or advice is available from several sources.

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

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

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



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

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

Information We Need

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

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

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

Alternatively, you might be asked to provide a log file created by the Consolidated Tracing Facility (CTF) - a tracing infrastructure that enables you to quickly and easily produce diagnostic information detailing the operation of a number of Micro Focus software components.

Creating Debug Files

If you encounter an error when compiling a program that requires you to contact Micro Focus technical support, your support representative might request that you provide additional debug files (as well as

source and data files) to help us determine the cause of the problem. If so, they will advise you how to create them.

Disclaimer

This software is provided "as is" without warranty of any kind. Micro Focus disclaims all warranties, either express or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall Micro Focus or its suppliers be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if Micro Focus or its suppliers have been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of liability for consequential or incidental damages so the foregoing limitation may not apply.

Micro Focus is a registered trademark.

Copyright © Micro Focus 1984-2012. All rights reserved.