Micro Focus Visual COBOL 2.1 Update 1 for Visual Studio

Release Notes

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

2013-03-27

Contents

Micro Focus Visual COBOL 2.1 Update 1 for Visual Studio 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.

Product Description

Visual COBOL enables you to develop COBOL applications within Microsoft Visual Studio. You use the integrated development environment (IDE) in Visual Studio to edit, compile and debug COBOL applications. The IDE provides all the functionality to manage projects and build applications.

The product is available in two variants - Visual COBOL for Visual Studio 2010 , and Visual COBOL for Visual Studio 2012.

Compiling for COBOL Server or for COBOL Server 2012

Visual COBOL for Visual Studio 2010 and Visual COBOL for Visual Studio 2012 each target a different version of the COBOL run-time system - COBOL Server and COBOL Server 2012, respectively.

Ensure that after you've edited the project in the preferred version of Visual Studio, you compile your source code in that version of Visual COBOL which targets the version of the COBOL run-time in which you want your applications to run.

For example, if you build your application using Visual COBOL for Visual Studio 2010 you cannot deploy the target files to COBOL Server 2012; you must use COBOL Server.

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

Hardware Requirements

Visual COBOL has the following requirements in addition to the requirements of Microsoft Visual Studio. See the Visual Studio documentation for details of the Microsoft requirements.

The disk space requirements are:

- 58MB for the Sentinel RMS license server
- 200MB for Micro Focus COBOL integration into Visual Studio

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

Operating Systems Supported

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

The following platforms are supported:

32-bit Windows Platforms	• Windows Server 2008 SP2 • Windows Vista • Windows XP Professional SP3 or later • Windows 7 • Windows 8
64-bit Windows Platforms	• Windows Server 2008 SP2 • Windows Server 2008 R2 • Windows Vista • Windows XP Professional SP2 or later • Windows 7 • Windows 8
Citrix and Terminal Server	• Windows Server 2008 SP2 • Windows Server 2008 R2 • Windows 7 • Windows 8 • Windows Server 2012

Note:

- Visual COBOL for Visual Studio 2012 is not supported on versions of Windows earlier than Windows 7.

Software Requirements

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

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

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

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*.

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

- A full version of Microsoft Visual Studio 2010 or Microsoft Visual Studio 2012 (Premium, Professional, or Ultimate), or the the respective version of the Microsoft Visual Studio Integrated Shell.

  Note:

  - Microsoft Visual Studio Express Edition is not supported.
  - You can download the Visual Studio Integrated Shell from the *Microsoft Download Center*. If you choose to install the Shell, ensure you run the installer to complete the installation - run `vsintshell.enu.exe` from the location where you installed the download.

The following functionality, tools and features have additional requirements:

Visual Studio IDE

- Microsoft Windows SDK is required if you are using Visual Studio Shell. See the *Microsoft Download Center* and search for Windows SDK.
- Microsoft .NET Framework 4.0. This is included with the above versions of Visual Studio.
- Microsoft .NET Framework 4.5. This is included with Visual Studio 2012.
- Visual Studio 2010 Service Pack 1 is required if you are using Visual COBOL for Visual Studio 2010 and Visual COBOL for Visual Studio 2012 on the same machine. If you are using the Web installer to install the product, it downloads and installs Visual Studio 2010 Service Pack 1 automatically. If you do not have Internet access, you need to download Visual Studio 2010 Service Pack 1 on a machine that has Internet connection, and then copy the installer to your machine.

  In addition, when building a native COBOL project that contains resources you must replace the `cvtres.exe` in the `bin` and `bin64` directories of your Visual COBOL installation with the version of the file which gets installed with Visual Studio 2010 SP1. You can download Service Pack 1 for Visual Studio 2010 from the *Microsoft Download Center*.

SQL CLR Integration for Visual Studio 2012

The following software is required to use the SQL CLR integration feature, which is specifically for the development and deployment of COBOL stored procedures under Microsoft SQL Server.

Projects based on the SQL Server Database Project template require:

- Either of the following:
  - Visual Studio Shell 2012 and Microsoft SQL Server Tools (SSDT SDK)
  - Visual Studio Professional 2012, Premium 2012, or Ultimate 2012

- Any of the following:
  - SQL Server 2008 R2 targeting .NET CLR v2.0 frameworks (2.0, 3.0, 3.5)
  - SQL Server 2012 targeting .NET CLR v4.0 framework (4.0), or .NET CLR v2.0 frameworks (2.0, 3.0, 3.5)
  - SQL Server Azure targeting .NET CLR v4.0 framework (4.0) and also .NET CLR v2.0 frameworks (2.0, 3.0, 3.5)

⚠ **Important:** The SQL CLR Database project template is not supported with Visual Studio 2012. Projects based on the SQL CLR Database project template are automatically upgraded to use the SQL Server Database project template when opened in Visual COBOL for Visual Studio 2010.

Windows Forms

- Microsoft Internet Information Service (IIS) is also required for generating Windows Forms test clients.

XML Extensions

- XML Extensions has the same requirements as RM/COBOL version 12 for 32-bit Windows. (See the *RM/COBOL User's Guide, Second Edition* or later.) Additionally, XML Extensions may be used in conjunction with Terminal Server.

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.

Installation restrictions and requirements

Before starting the installation you should consider the following:

- Visual COBOL and COBOL Server cannot coexist on the same machine.
- Visual COBOL and Enterprise Developer cannot coexist on the same machine.
- If, when you install Visual COBOL for Visual Studio 2010, the machine does not have Microsoft Visual C++ 2010 Redistributable Runtime already installed, it is installed as required by Visual COBOL. The installation of Microsoft Visual C++ 2010 Redistributable Runtime adds a number of `.dll` files, without digital signatures, into the `winsxs` directory.
- If, when you install Visual COBOL for Visual Studio 2012, the machine does not have Microsoft Visual C++ 2012 Redistributable Runtime already installed, it is installed as required by Visual COBOL. The installation of Microsoft Visual C++ 2012 Redistributable Runtime adds a number of `.dll` files, without digital signatures, into the `winsxs` directory.
- If you are installing this as an upgrade, make sure that none of the product files are in use when you start the installation.
- 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.

Downloading the Product

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

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

Installing

Note: If you do not have Visual Studio 2010 Service Pack 1 installed on your machine, the Web Installer will download and install it. You need Visual Studio 2010 Service Pack 1 if you are using Visual COBOL for Visual Studio 2010 and Visual COBOL for Visual Studio 2012 on the same machine. If you wish to install Visual Studio 2010 Service Pack 1 later, or you do not have Internet access, you need to download Visual Studio 2010 Service Pack 1 on a machine that has Internet connection, and then copy the installer to your machine.

To use the Web Installer:

1. Double-click the `visualcobolvisualstudio21_update1_2010_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 `visualcobolvisualstudio21_update1_2010.exe` file and follow the wizard instructions to complete the installation.

Note: If you are installing Visual COBOL for Visual Studio 2012 the files are `VisualCOBOLVisualStudio21_update1_2012.exe` and `VisualCOBOLVisualStudio21_update1_2012_webinstaller.exe`, respectively.

Note:

- If you are installing onto a machine that has an existing Micro Focus product that uses an older Sentinel RMS License Manager, you might be prompted to remove it and install the Micro Focus License Manager. By doing this you maintain the existing Sentinel RMS license files while adding the Micro Focus License Manager. If you are unsure about existing licenses on your computer or removing the Sentinel RMS License Manager, consult your System Administrator. If you want to proceed, remove Sentinel RMS License Manager by using Windows **Add or Remove Programs** and rerun the installation file.
- Trial licenses cannot be used with remote desktop services. If you want to use your product in this way, please contact Micro Focus SupportLine to obtain a relevant license.
- We recommend that you install any updates for 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 will update existing installations of Visual COBOL 2.1 for Visual Studio.

Before installing, check *Installation Restrictions and Requirements*.

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**.

Note: The Start menu is not available on Windows 8. You use the Start screen to invoke programs.

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.

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.
- 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.
- 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.

- To view the help in Visual Studio 2012, ensure that the Visual Studio Help Library is pointing to local help. From the Visual Studio menu click **Help > Set Help Preferences > Launch in Help Browser**.
- For full details of the Visual Studio 2012 Help system, see the locally installed Microsoft Help Viewer 2.0 Help, which is available from Help menu in the IDE.

Repairing

If any product files, registry settings or shortcuts are accidentally removed at any point, you can perform a repair on the installation to replace them.

To repair your installation on versions of Windows Vista or later:

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

To repair your installation on older versions of Windows, such as Windows XP:

1. Click **Start Menu > Control Panel > Add/Remove Programs**.
2. Click your Micro Focus product in the list of installed programs.
3. Click **Click here for support information**.
4. Click **Repair**.

Uninstalling

Windows

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

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

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

Note: The installer creates separate installations for Micro Focus Visual COBOL and Micro Focus License Manager. Uninstalling only Visual COBOL does not automatically uninstall the Micro Focus License Manager or any of the prerequisite software.

To completely remove the product you must uninstall the Micro Focus License Manager as well.

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

Some registry entries are not removed by the uninstallation process and you need to manually delete them.

The following folders might not be removed:

- The `Micro Focus` *`Product Name`* folder in the Start menu - you can delete it manually.
- `%systemdrive%\Users\Public\Documents\Micro Focus` - includes the binaries and the log files of the samples which you have built.

- `%ProgramData%\Micro Focus` - includes some data files used by the Micro Focus licensing system.
- `%Program Files%\Micro Focus` - you can delete it manually.

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
```

Visual COBOL Editions and Managing Licenses

Visual COBOL comes in the following variants:

Visual COBOL Visual COBOL is for customers looking for off-mainframe COBOL development and deployment tools to modernize business-critical enterprise applications. This option uses either the Visual Studio or Eclipse-based IDE and includes development and test tools for all platforms currently supported by Micro Focus.

Visual COBOL Personal Edition Visual COBOL Personal Edition is a full-featured version of Visual COBOL for student and non-commercial usage. The product is restricted only by a limited numbers of source lines that can be compiled for a given program (approximately 2200).

You can use Visual COBOL Personal Edition for a period of 365 days, after which you will not be able to use it. A trial license of Visual COBOL lasts for 30 days, after which, if you have not authorized it with your authorization code, you will return to Personal Edition functionality.

After activation, you can see how many days your trial license has remaining by selecting **Help > Micro Focus > Product Licensing**, or by using the Micro Focus Licensing Administration tool.

To manage your product licenses you need to use the Micro Focus Licensing Administration tool. The tool allows you to authorize, view and revoke licenses. You can set up your license locally or request a license from, if your site is using one, a central license server.

You can also apply your authorization code directly within the product from the **Micro Focus *Product Name* Licensing** dialog.

For more on the Micro Focus Licensing Administration Tool, see *Licensing* in the Visual COBOL help.

To activate Visual COBOL Personal Edition

Note: Having activated Visual COBOL Personal Edition, your use of it is limited to 365 days. After this period you will need to enter an authorization code in order to continue using it, either for a 30 day trial or full license of Visual COBOL.

1. Start Visual COBOL.

   If you have not installed a license for Visual COBOL, starting the IDE and trying to create a COBOL project opens the **Micro Focus Visual COBOL Product Licensing** dialog box. If you cancel this dialog box, you can invoke it again from **Help** > **Micro Focus Product Help** > **Product Licensing**.
2. Click **I want to activate the free product**.
3. Ensure that the email address used to register the product is in the **Email address** text entry field. If you haven't registered your email address yet, click **registration page** and follow the instructions on that page.
4. Select one of the following options:

   Automatic Use this if you have Internet access.

   1. Click **Activate License** to activate your copy of Visual COBOL Personal Edition.

   Manual Use this if you are not connected to the Internet.

   1. Click **Send email**.

This opens your default mail client and creates a new email filled in with the details to send to Micro Focus about activating your copy of the free Personal Edition of the product.

Note: If there is no mail client installed on your machine, you will be presented with a template that includes the email address, the subject and the contents of an email to send with a mail client of your choice. Do not modify the details.

You will receive a response email with details about your authorization request.

2. Paste the contents of the response email in the designated field in the **Micro Focus Visual COBOL Product Licensing** dialog box.
3. Click **Activate License**.

   This displays a message confirming your request for authorization has been successful.
4. Click **Close**.

To request and activate a 30-days trial license for Visual COBOL

1. In the IDE, click **Help** > **Micro Focus Product Help** > **Product Licensing**.

   This opens the **Micro Focus Visual COBOL Product Licensing** dialog box.
2. Type your email address in the **Email address** text entry field.

   You need to use the same email address you used for the registration.
3. Select one of the following options:

Automatic Use this if you have Internet access.

1. Click **Activate License** to activate the trial license for Visual COBOL.

   Important: If, after you click **Finish**, you click the **Cancel** button in the progress dialog box to stop the activation process, the trial license might have registered on Micro Focus servers but not yet be registered on your machine. If you request a trial again then you can get a message saying that a trial license has already been used. If this happens, you should contact a Micro Focus Sales representative to obtain a new license.

   You can do this by sending an email to VCBuyNow@microfocus.com, or selecting **Help > Micro Focus > Buy Now** and using the contact options in the dialog box.

Manual Use this if you are not connected to the Internet.

1. Click **Send email**.

   This opens your default mail client and creates a new email filled in with the details to send to Micro Focus about activating the trial license for Visual COBOL.

   Note: If there is no mail client installed on your machine, you will be presented with a template that includes the email address, the subject and the contents of an email to send with a mail client of your choice. Do not modify the details.
2. Send the email.

   You will receive a response email with details about your authorization request.
3. Paste the contents of the response email in the designated field in the **Micro Focus Enterprise Developer Product Licensing** dialog box.
4. Click **Activate License**.

This displays a message confirming your request for authorization has been successful.

5. Click **Close**.

After activation, you can see how many days your trial license has remaining by selecting **Help > Micro Focus > Product Licensing**, or by using the Micro Focus Licensing Administration tool.

To buy and activate a full unlimited license

Note: You can only activate a full license from the IDE if you haven't yet activated Personal Edition. To activate a full version at any other time you must use the Micro Focus Licensing Administration tool.

For instructions on using the Micro Focus Licensing Administration Tool, see *Licensing* in the Visual COBOL help.

1. In the IDE, click **Help** > **Micro Focus Product Help** > **Product Licensing**.

   This opens the **Micro Focus Buy Now** dialog box.
2. Ensure that the email address you used to register the product is in the **Email address** text entry field.
3. Click **Send email**.

   This opens your default mail client and creates a new email filled in with the details to contact Micro Focus. Send the email.

   If there is no mail client installed on your machine, you will be presented with a template that includes the email address and subject line for an email to send to Micro Focus using a mail client of your choice.

   You will be contacted by a Micro Focus Sales representative.

Activating the license from the IDE

When you have bought your license you are given an authorization code with which to activate the product.

1. Start Visual COBOL.

   If you have not installed any license for Visual COBOL, starting the IDE and trying to create a COBOL project opens the **Micro Focus Visual COBOL Product Licensing** dialog box. If you cancel this dialog box, you can invoke it again from **Help** > **Micro Focus Product Help** > **Product Licensing**.
2. Click **I have a full Visual COBOL license**, paste the code in the **Enter authorization code** field, and then click **Authorize**.
3. Click **Activate License**.

   You should receive a message that the activation has been successful.
4. Click **Close**.

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 2.1 Update 1

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

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.

Features Added in Visual COBOL 2.1

ACUCOBOL-GT Data Types in Managed Code

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

ACUCOBOL-GT Library Routines in Managed Code

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

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 Visual COBOL projects that compile to `.int` and `.gnt` code.

Managed COBOL Enhancements

Delegates and Events 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

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

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

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

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

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

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

SQL Server We now support Microsoft SQL Server 2012.

Features Added 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.

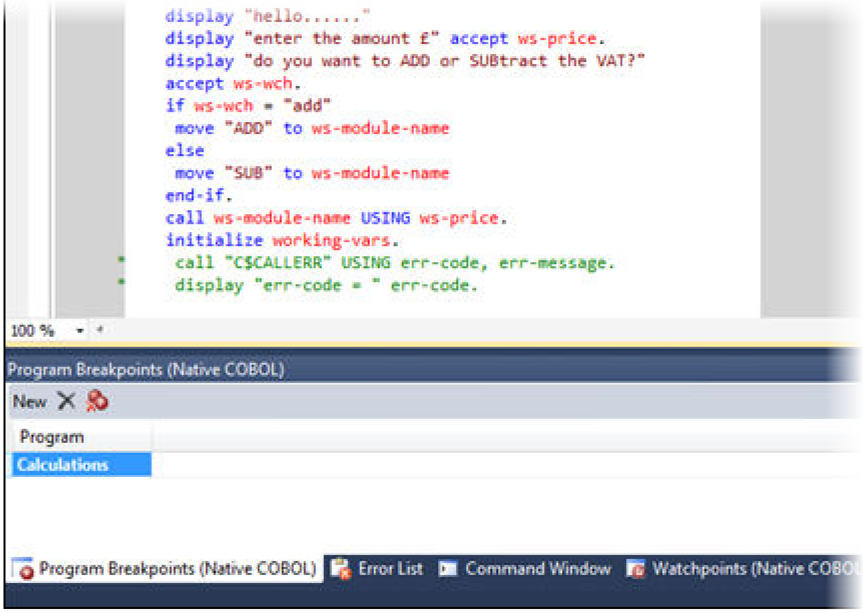
Program Breakpoints

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)**.



```
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.
```

Program
Calculations

100 %
Program Breakpoints (Native COBOL)
New
Program Breakpoints (Native COBOL)
Error List
Command Window
Watchpoints (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.

Data Access

This release provides the following enhancements:

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

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 Compatibility AddPack for Visual COBOL 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 `nxrclr.chm`, which is installed in the Help folder of your installation. The default location is`%ProgramFiles(x86)%\Micro Focus\Visual COBOL\Help`.

To open the help, double-click `nxrclr.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$CALLEDBY
  - C$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.

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 o1 type MyClass`	`01 o1 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)`

`       method-id MyMethod public.`
 	
`       local-storage section.`
 	
`       procedure division.`
 	
`           goback.`
`       end method.`
	`       method-id. "MyMethod" public.`
	`                local-storage section.`
	
	`                procedure division.`
	`                    goback.`
	`                end method "MyMethod".`

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.

<table>
<tr><th>New Syntax</th><th>Old Syntax</th></tr>
<tr><td><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.
       end method.

       end class.
</pre></td><td><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".
       local-storage section.
       procedure division.

           goback.
       end method "InstanceMethod".

       end object.
       end class MyClass.
</pre></td></tr>
</table>

Environment Division, Configuration Section, Repository

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

<table>
<tr><th>New Syntax</th><th>Old Syntax</th></tr>
<tr><td><pre>
       program-id. Program1 as "MyProject.Program1".

       data division.
       working-storage section.
       procedure division.

           goback.
       end program Program1.
</pre></td><td><pre>
       program-id. Program1 as "MyProject.Program1".

       environment division.
       configuration section.
       repository.

       data division.
       working-storage section.
       procedure division.

           goback.
       end program Program1.
</pre></td></tr>
</table>

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
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. goback. end method. end class.	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". local-storage section. procedure division. goback. end method "InstanceMethod". end object. end class MyClass.

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.

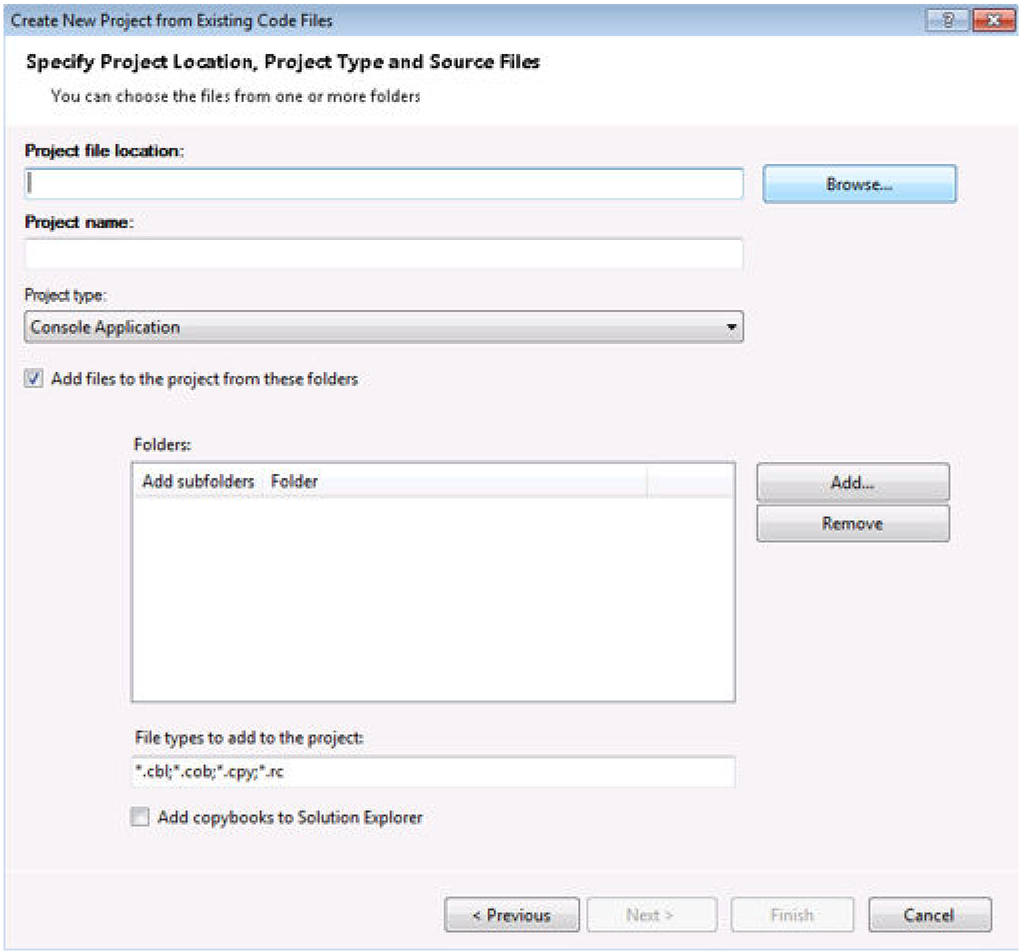
For example:

New Syntax	Old Syntax

class-id MyNamespace.MyClass.
    attribute Serializable.
 	
working-storage section.
...
 	
end class.
	class-id. MyClass as "MyNamespace.MyClass".
	
	class-attributes.
	custom-attribute is type "SerializableAttribute".
	
	object.
	
	working-storage section.
	   ...
	
	end object.
	
	end class MyClass.

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.



Create New Project from Existing Code Files

Specify Project Location, Project Type and Source Files

You can choose the files from one or more folders

Project type: Console Application

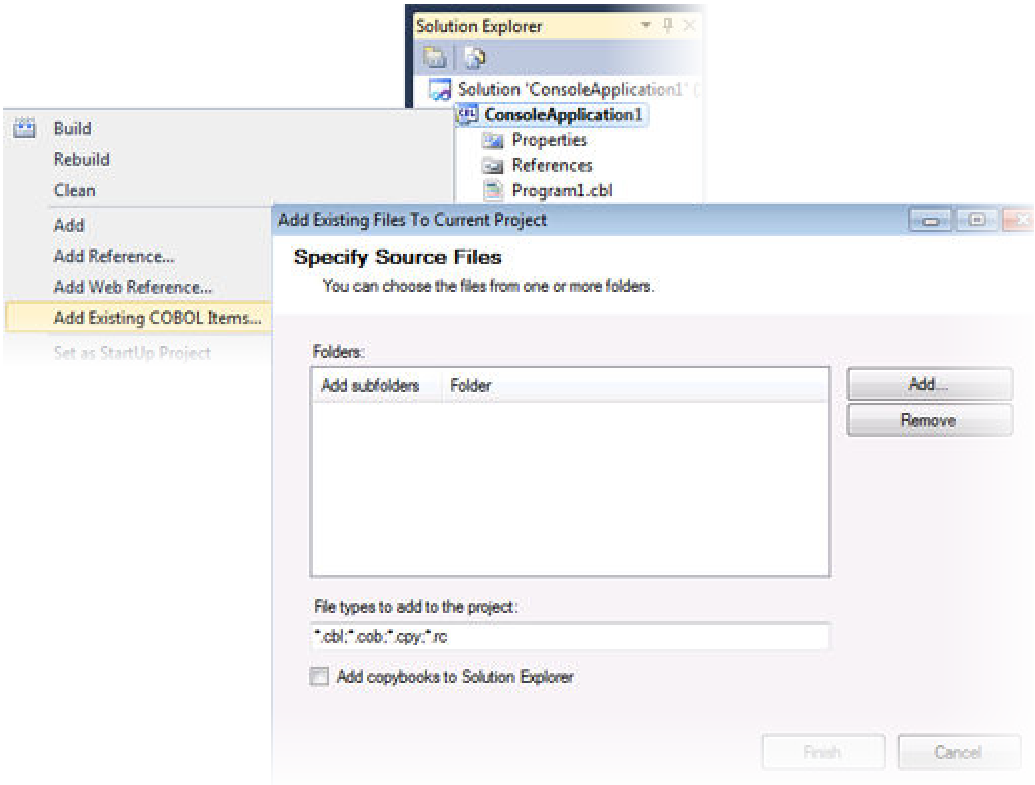
☑ Add files to the project from these folders

File types to add to the project: *.cbl;*.cob;*.cpy;*.rc

□ Add copybooks to Solution Explorer

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.



Specify Source Files

You can choose the files from one or more folders.

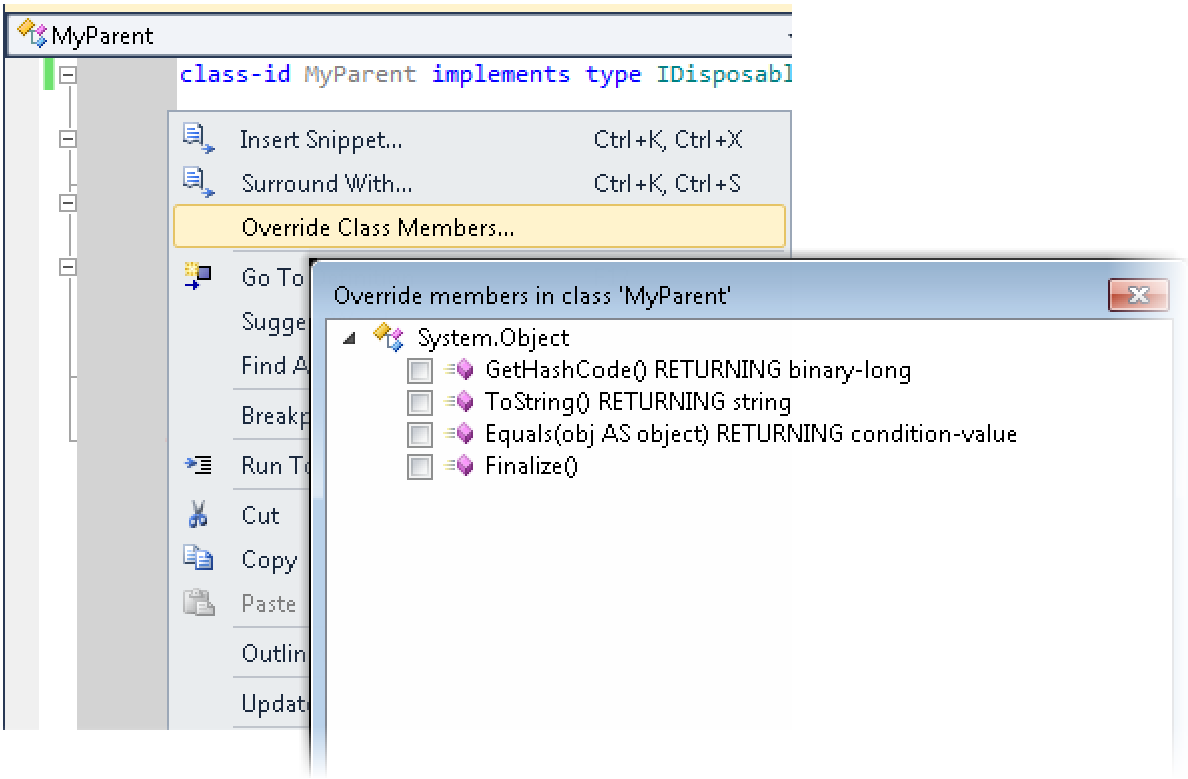
File types to add to the project: *.cbl;*.cob;*.cpy;*.rc

☐ Add copybooks to Solution Explorer

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.



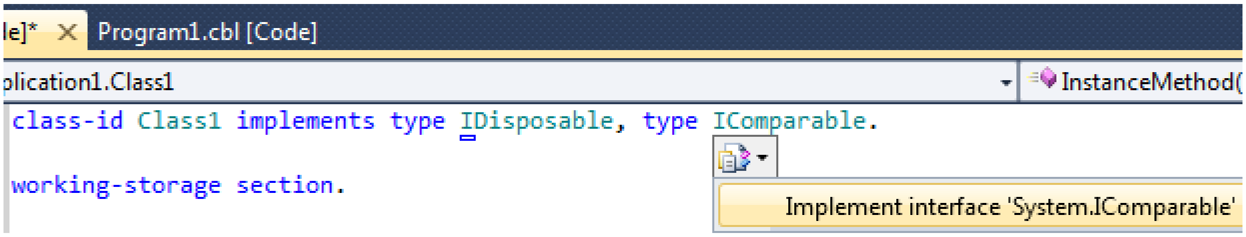
```
class-id MyParent implements type IDisposabl
```

MyParent
Insert Snippet... Ctrl+K, Ctrl+X
Surround With... Ctrl+K, Ctrl+S
Override Class Members...
Go To
Sugge
Find A
Breakp
Run To
Cut
Copy
Paste
Outlin
Updat
Override members in class 'MyParent'
System.Object
□ GetHashCode() RETURNING binary-long
□ ToString() RETURNING string
□ Equals(obj AS object) RETURNING condition-value
□ Finalize()

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.



```
class-id Class1 implements type IDisposable, type IComparable.

working-storage section.
```

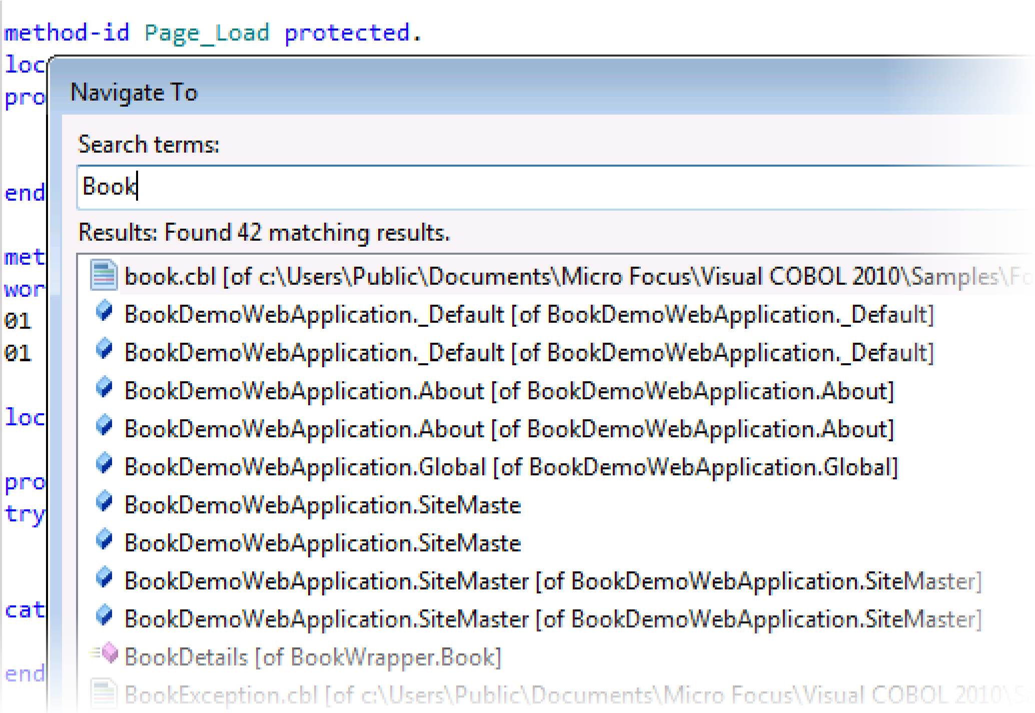
Snippet for Implements

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.



```
method-id Page_Load protected.
loc
pro

end

met
wor
01
01

loc

pro
try

cat

end
```

Search terms:

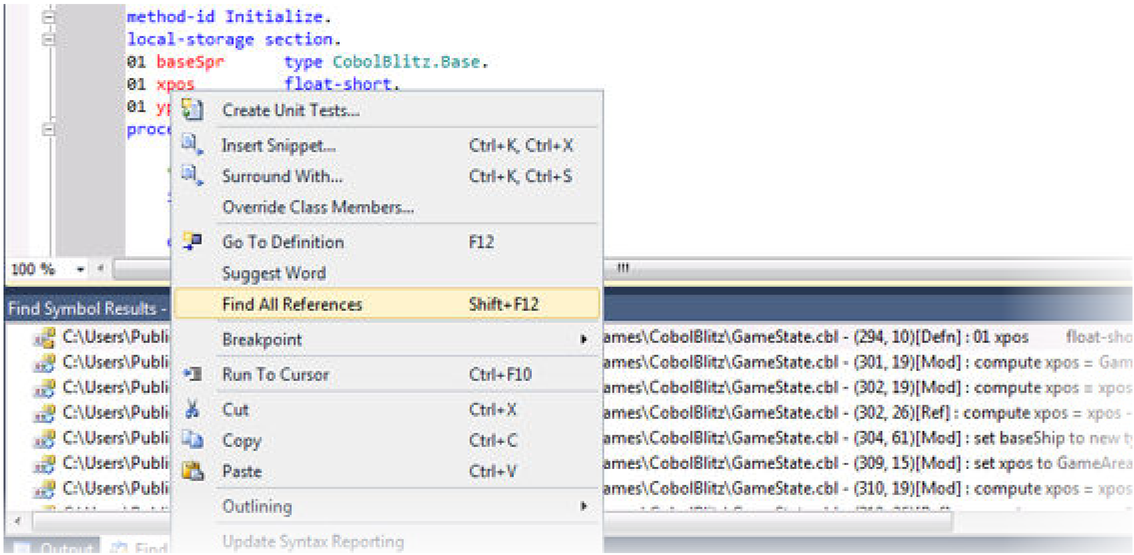
Book

Results: Found 42 matching results.

book.cbl [of c:\Users\Public\Documents\Micro Focus\Visual COBOL 2010\Samples\Fo
BookDemoWebApplication._Default [of BookDemoWebApplication._Default]
BookDemoWebApplication._Default [of BookDemoWebApplication._Default]
BookDemoWebApplication.About [of BookDemoWebApplication.About]
BookDemoWebApplication.About [of BookDemoWebApplication.About]
BookDemoWebApplication.Global [of BookDemoWebApplication.Global]
BookDemoWebApplication.SiteMaste
BookDemoWebApplication.SiteMaste
BookDemoWebApplication.SiteMaster [of BookDemoWebApplication.SiteMaster]
BookDemoWebApplication.SiteMaster [of BookDemoWebApplication.SiteMaster]
BookDetails [of BookWrapper.Book]
BookException.cbl [of c:\Users\Public\Documents\Micro Focus\Visual COBOL 2010\S

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.



```
method-id Initialize.
local-storage section.
01 baseSpr         type CobolBlitz.Base.
01 xpos            float-short.
01 y
proc
```

```
C:\Users\Publi ames\CobolBlitz\GameState.cbl - (294, 10)[Defn] : 01 xpos      float-sho
C:\Users\Publi ames\CobolBlitz\GameState.cbl - (301, 19)[Mod] : compute xpos = Gam
C:\Users\Publi ames\CobolBlitz\GameState.cbl - (302, 19)[Mod] : compute xpos = xpos
C:\Users\Publi ames\CobolBlitz\GameState.cbl - (302, 26)[Ref] : compute xpos = xpos -
C:\Users\Publi ames\CobolBlitz\GameState.cbl - (304, 61)[Mod] : set baseShip to new t
C:\Users\Publi ames\CobolBlitz\GameState.cbl - (309, 15)[Mod] : set xpos to GameArea
C:\Users\Publi ames\CobolBlitz\GameState.cbl - (310, 19)[Mod] : compute xpos = xpos
```

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:

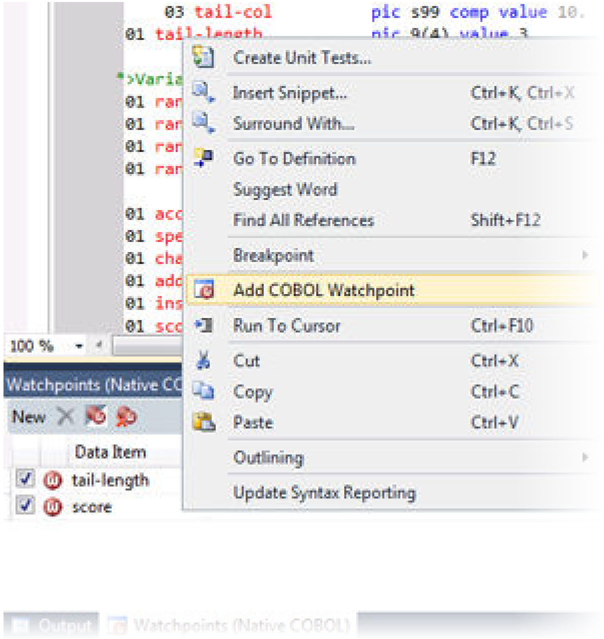
- A Web Application project includes a project file which enables you to specify what files are part of the project and should be compiled.

- 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

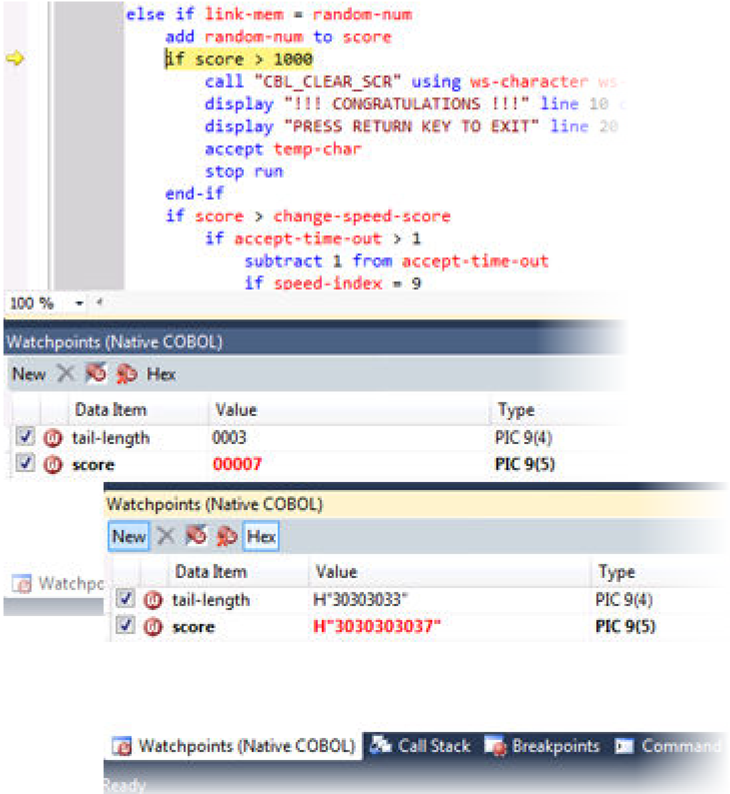
The following debugging enhancements have been made:

- 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.



```
    03 tail-col          pic s99 comp value 10.
01 tail-length           pic 9(4) value 3
```

- **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.

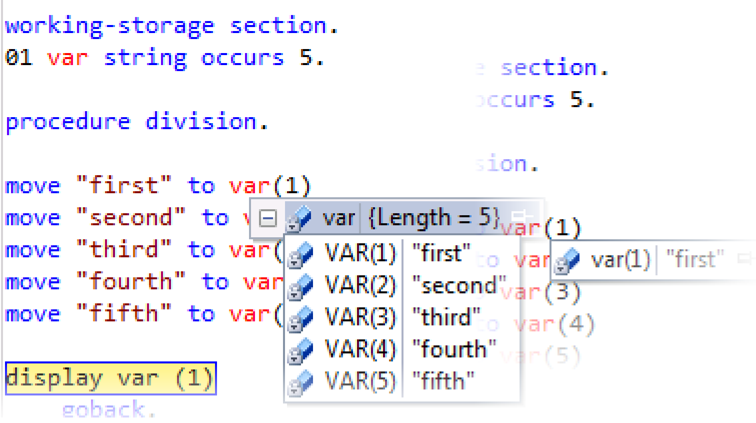


```
else if link-mem = random-num
    add random-num to score
    if score > 1000
        call "CBL_CLEAR_SCR" using ws-character ws-
        display "!!! CONGRATULATIONS !!!" line 10 c
        display "PRESS RETURN KEY TO EXIT" line 20
        accept temp-char
        stop run
    end-if
    if score > change-speed-score
        if accept-time-out > 1
            subtract 1 from accept-time-out
            if speed-index = 9
```

	Data Item	Value	Type
☑	tail-length	0003	PIC 9(4)
☑	score	00007	PIC 9(5)

	Data Item	Value	Type
☑	tail-length	H"30303033"	PIC 9(4)
☑	score	H"3030303037"	PIC 9(5)

- 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.



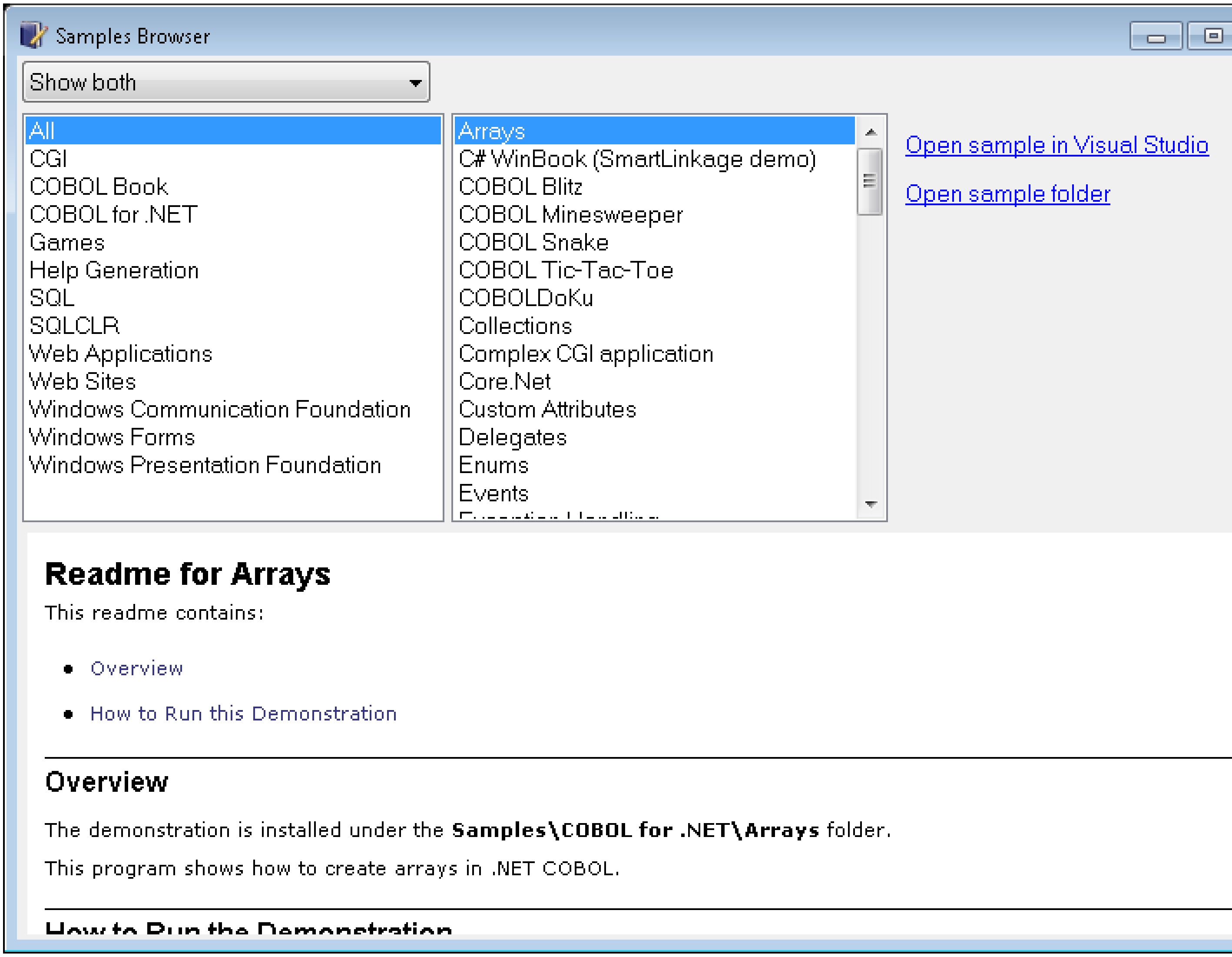
```
working-storage section.
01 var string occurs 5.

procedure division.

move "first" to var(1)
move "second" to v
move "third" to var(
move "fourth" to var
move "fifth" to var(

display var (1)
    goback.
```

```
var {Length = 5}
VAR(1) "first"
VAR(2) "second"
VAR(3) "third"
VAR(4) "fourth"
VAR(5) "fifth"
```

```
var(1) "first"
```

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



Readme for Arrays

This readme contains:

- Overview
- How to Run this Demonstration

Overview

The demonstration is installed under the **Samples\COBOL for .NET\Arrays** folder.

This program shows how to create arrays in .NET COBOL.

How to Run the Demonstration

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.

Main Features of Visual COBOL 2.1 for Visual Studio 2012

This version of Visual COBOL includes the following main features:

Integration with Visual Studio 2012

Visual COBOL provides support for the new features of Visual Studio 2012 and the Visual Studio 2012 Integrated Shell, including:

Compatibility (Project Round-Tripping)

The project round-tripping features enables you to use Visual Studio 2012 to open and edit a project created with Visual Studio 2010 without upgrading or changing anything. After closing the project, you can still open it in Visual Studio 2010 to make further changes.

These are the requirements and the restrictions for using project round-tripping with COBOL projects:

Note:

- The version of Visual Studio 2010 being used to create and edit projects must be Service Pack 1.
- For managed COBOL, the feature only works for applications that target versions of the .NET Framework 2 to 4.
- There is a new format for COBOL SQL CLR projects in Visual Studio 2012. Such projects created with Visual Studio 2010 will be upgraded when you open them in Visual Studio 2012.

.NET Framework 4.5 Compatibility Provides support for creating managed COBOL applications that target version 4.5 of the .NET Framework.

Reference Manager Supports the new Add Reference dialog box which provides a faster way for adding references to your projects.

Browse Code in Solution Explorer Hub Supports the enhanced Solution Explorer Hub without the in-built class/object browser features.

Previewing Files in the Code Editor Supports code preview for COBOL files which enables you to explore the files in your project without opening them in the editor.

Loading Projects Asynchronously Enables you to load your source code in the IDE and start working on it faster even when you have large applications.

Search Supports the enhanced search features across the IDE for COBOL projects.

Visual Studio 2012 Themes Visual COBOL conforms to the new look and feel of Visual Studio 2012.

Microsoft Help Viewer 2.0 Provides support for the Micro Focus Help in Microsoft Help Viewer 2.0.

New SQL CLR Project Templates

The COBOL SQL CLR project template has a new format. COBOL SQL CLR projects created with Visual Studio 2010 will be upgraded when you open them in Visual Studio 2012.

Windows 8

Visual COBOL supports Windows 8.

Coexistence with Visual COBOL for Visual Studio 2010

Visual COBOL 2.1 for Visual Studio 2012 and Visual COBOL 2.1 for Visual Studio 2010 can coexist on the same machine. You can use either one of these versions to edit COBOL projects that are supported by the project round-tripping feature in Visual Studio 2012.

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 workaround 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 Apptrack.

COBOL Watchpoints

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

Co-existing with Earlier Micro Focus Products

Run-time system error due to COBCONFIG

A run-time system error occurs if either the COBCONFIG or COBCONFIG_ environment variable is set when you run aVisual 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.

Documentation

- In Visual Studio 2012, clicking **Help** > **Micro Focus Product Help** > **Product Documentation** results in the message "Cannot find requested topic on your computer". If you see this message, click **Micro Focus Visual COBOL 2.1 Update 1 for Visual Studio 2012** in the **Contents** tab to display the documentation as expected.
- If you install Visual COBOL for Visual Studio 2012 and its documentation is not available in the Microsoft Help Viewer, perform the following steps:

  1. In Visual Studio 2012, click **Help** > **Add and Remove Help Content**.
  2. Choose the **Manage Content** tab.
  3. Check the **Disk** radio button, then navigate to the folder containing the Visual COBOL for Visual Studio 2012 documentation. By default, this is `C:\Program Files (x86)\Micro Focus\Visual COBOL for Visual Studio 2012\help`.
  4. Select `helpcontentsetup.msha` and click **Open**.
  5. In the content list, click **Add** next to the Visual COBOL 2.1 entry.
  6. Click **Update**.
  7. Click **Yes** on the User Account Control and Microsoft Help Viewer 2.0 dialog boxes to enable the update to continue.

     The documentation for Visual COBOL for Visual Studio 2012 is added to the Microsoft Help Viewer.

  Note:

  This problem only occurs if you install Visual COBOL for Visual Studio 2012 while documentation for Visual Studio 2012 is being downloaded or updated, which typically happens when you install Visual Studio 2012 or run it for the first time.

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).
- If you are accessing Vision files through the Vision file handler (as opposed to the default Micro Focus File Handler) and you are upgrading from Visual COBOL version 2.0 to version 2.1, you must first uninstall version 2.0, otherwise the Vision file handler may not function correctly.

Fileshare

A write of a record with an alternate key where the alternate key is the first value for that key may could 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 `aslmpclocate.exe` and `init2aslm.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 accidently 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.

Native COBOL

On Windows XP, when building a native COBOL application in Visual Studio, a dialog may pop up with error "Unhandled exception at 0xc0006866 in cobol.exe: 0xC0000005: Access violation reading location 0xc0006866)". This is caused by Symantec antivirus runtime protection. To resolve this, you need to apply the following fix from Symantec: *http://www.symantec.com/business/support/index?page=content&id=TECH97280&locale=en_US*.

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*
- *BIS Service Engine*
- *Compiler*
- *Data Tools Converter*
- *Data Tools Vrecgen (Character)*
- *Documentation*
- *File Handling - Sort*
- *MF Communications Server*
- *MF Directory Server*
- *MVS REXX Emulation*
- *NCG*
- *RTS*
- *SQL: COBSQL*
- *SQL: OpenESQL*
- *Visual Studio IDE*
- *XML syntax support runtime*

.NET Compiler

Back to List

- A COMPUTE statement where the target field had fewer significant digits than some of the operands of the arithmetic expression no longer results in invalid arithmetic.

  1087987 (2614566)

.NET ESQL Support

Back to List

- SQLWARN4 flag was not being set when more than one row was returned for a singleton SELECT statement. The OpenESQL run-time now correctly sets SQLWARN flags when a singleton SELECT statement returns multiple rows.

  1085945 (2593798)
- The OpenESQL pre-compiler sometimes flagged object host variables as invalid when multiple programs were compiled into a single .exe or a .dll file.

  1086906 (2603712)

.NET RTS

Back to List

- A performance issue with raising an expression to a fractional power has been fixed.

1086614 (2594740)

BIS Service Engine

Back to List

- Visual COBOL BIS supports and can operate with either one of the following products - Visual COBOL for Visual Studio 2010, Visual COBOL for Visual Studio 2012, or COBOL Server. If you have installed both Visual COBOL for Visual Studio 2010 and Visual COBOL for Visual Studio 2012 on the same machine, BIS uses Visual COBOL for Visual Studio 2010, by default. To use the version for Visual Studio 2012, you need to edit the following registry entries and set the "vcob_key" value to "Visual COBOL 2012":

  1) HKEY_LOCAL_MACHINE\SOFTWARE\Micro Focus\Xcentrisity BIS\2.1\Visual COBOL\Config

  2) On 64-bit versions of Windows, HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Micro Focus\Xcentrisity BIS\2.1\Visual COBOL\Config

  594902 ( )
- A problem with running BIS programs that open files on AIX which was caused by an stack size that was too small has been resolved. Set COBMAINSTACK in /etc/sysconfig/xbis to set the desired size.

  595039 ( )
- The BIS interface functions called by COBOL programs under Xcentrisity BIS now begin with "B_" instead of "B$". This change is required to make Xcentrisity BIS for Windows source-compatible with Xcentrisity BIS for Linux and AIX.

  594736 ( )

Compiler

Back to List

- Specify an environment variable with the USE and DIRECTIVES Compiler directives (e.g. USE"$myDirs") to locate a directives file.

  1086063 (2592063)

Data Tools Converter

Back to List

- The dfconv replacement input and output filenames are no longer truncated to the filename lengths used within the profile file.

  1086569 (2599741)

Data Tools Vrecgen (Character)

Back to List

- The source files for the VRECGEN and VRECGEN2 utilities are now stored in the .\src folder in the product installation.

  1087209 (2603963)

Documentation

Back to List

- The SSRANGE Compiler directive is provided for emulation of the IBM mainframe compiler of the same name; because of this, its scope is limited to the syntax permissible in a mainframe dialect.

  1087083 (2605408)

- 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.

  593437 ( )
- All COBOL CICS programs that call user exits need to be compiled with the NOAMODE directive.

  593780 ( )
- The documentation now clarifies the EZACICM.MOD location.

  1086693 (2599949)
- The documentation has been updated to provide more information about the MFJAMS LISTCAT command.

  594069 ( )

File Handling - Sort

Back to List

- A SORT RETURN statement now returns a 9/230 error for the return past EOF.

  1087358 (2606867)
- SORT now terminates with return code 16 and throws a 9/013 error when the catalogued input file is not physically present.

  1087529 (2607690)
- SORT now terminates with return code 16 and displays error message "SORT103E Invalid operator .JOINKEYS." when the JCL contains the JOINKEYS parameter.

  1087311 (2607369)

MVS REXX Emulation

Back to List

- Execs residing in temporary datasets allocated to SYSEXEC or SYSPROC no longer sometimes fail to load.

  1086478 (2579785)
- The bpxwunix() function is now supported by the REXX engine.

  1085190 (2584045)

NCG

Back to List

- Display statements of the type "display a(1:i*c) at 0101" could cause the generator to fail when in debug mode.

  1088048 (2607368)

SQL: COBSQL

Back to List

- The COBSQL preprocessor could not process options longer than 65 characters.

  1085597 (2589416)

SQL: OpenESQL

Back to List

- The OpenESQL preprocessor sometimes generated incorrect query lengths for EXEC SQL PREPARE INTO FROM statements, resulting in the SQL queries being truncated.

  1087324 (2607761)
- The ODBC pre-compiler now allows you to define host variables after the DECLARE CURSOR SQL statement if they are not defined in the PROCEDURE DIVISION.

  1086501 (2599123)
- When using the SQL Server Native Client ODBC driver, the OpenESQL Run-Time System now correctly processes SQL Server data defined as VARCHAR(MAX).

  1086665 (2601192)

Visual Studio IDE

Back to List

- The Error List window now shows the details about the "Illegal command line" error when it is a result of setting invalid SQL directives.

  1085251 (2585426)
- 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.

  1087235 (2604749)
- Adding files to a project when directives scan is disabled was taking a long time to complete.

  593378 ( )
- An issue where you could not open copybooks from the context menu in the COBOL editor when the filename was specified with its extension and without surrounding quotes has been resolved.

  1087031 (2604709)
- There is an improvement in the performance of the cursor in the text editor when working with larger files and projects.

  1085255 (2585450)
- A problem with the value of "Link with objs" setting being duplicated after you reload the COBOL Link properties page has been resolved.

  1086091 (2595408)
- 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.

  1085258 (2585458)
- There is no longer a crash when adding a reference path to a managed COBOL project.

  1087055 (2604844)

XML syntax support runtime

Back to List

- The XML preprocessor now generates correct output for the COUNT IN clause.

  1086285 (2596137)

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.