



Micro Focus Enterprise Developer 6.0 Build Tools for Windows

Release Notes

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

© Copyright 2020 Micro Focus or one of its affiliates.

MICRO FOCUS, the Micro Focus logo and Enterprise Developer are trademarks or registered trademarks of Micro Focus or one of its affiliates.

All other marks are the property of their respective owners.

2020-06-02

Contents

Enterprise Developer Build Tools for Windows Release Notes	4
Installation	5
Prerequisites for Using Enterprise Developer Build Tools for Windows	5
Installing Enterprise Developer Build Tools for Windows	5
Microsoft Package Dependencies	6
Managing the Microsoft Build Tools and Windows SDK Packages	7
The Microsoft Build Tools and Windows SDK Configuration Utility	7
Restrictions	10
Licensing Considerations	11
Updates and SupportLine	12
Further Information and Product Support	12
Information We Need	12
Creating Debug Files	13
Copyright and Disclaimer	14

Enterprise Developer Build Tools for Windows Release Notes

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

Product Overview

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

Installing Enterprise Developer Build Tools for Windows does not install any documentation. Instead, full information on how to use Enterprise Developer Build Tools for Windows is included in the documentation for Enterprise Developer. The documentation that describes how to use Enterprise Developer Build Tools for Windows includes links to the sections of documentation for Enterprise Developer that are relevant to Enterprise Developer Build Tools for Windows.

Installation

Prerequisites for Using Enterprise Developer Build Tools for Windows

This section lists any additional software that you need to be able to use Enterprise Developer Build Tools for Windows.

Apache Ant

Apache Ant version 1.9.x or later is required if you will be using Enterprise Developer Build Tools for Windows to perform build operations using Eclipse project files. You can download Apache Ant from the *Apache Ant project's Binary Distributions page*.

Existing Microsoft Build Tools packages

If you already have Microsoft Build Tools for Visual Studio 2017 installed, you must ensure that it contains the Visual C++ build tools workload (or at least the individual components contained in that workload) before you run this installation.

Installing Enterprise Developer Build Tools for Windows

To install Enterprise Developer Build Tools for Windows:

1. Read the following sections of these Release Notes to minimize the chances of encountering issues during or after the installation process:
 - *Licensing Considerations*
 - *Prerequisites for using Enterprise Developer Build Tools for Windows*
 - *Restrictions when using Enterprise Developer Build Tools for Windows*
2. Run the `edbt_60.exe` file and follow the wizard instructions to complete the installation.



Note:

`edbt_60.exe` can take a number of parameters, enabling you to specify a number of different types of installation. Examples are as follows:

- Non-standard Installation

Format:

```
edbt_60.exe skipmstools=1
```

Summary:

Full installation, minus the Microsoft package dependencies. A standard installation downloads and installs the latest Microsoft Build Tools (for Visual Studio 2017) and Windows SDK packages. Refer to *Microsoft Package Dependencies* to see if you will depend on these packages, and if not, use the option above to skip installation of the packages, thus reducing the footprint and duration of the installation process.

- Offline Installation (preparation)

Format:

```
start /wait edbt_60.exe /q downloadmstools=<directory>
```

Summary:

This type of installation should be run by an administrator when preparing an installation that will be run in an offline environment.

The <directory> specified as part of the `downloadmstools` option should be a fully-qualified path name to a directory that is accessible to the offline clients. It is to this location that the required Microsoft packages are placed. For more information on these packages, see *Microsoft Package Dependencies*.

- Offline Installation (client install)

Format:

```
edbt_60.exe installmstools=<directory>
```

Summary:

This type of installation should be run for client installs where internet access is not available during the installation. It must be run after an administrator has run the offline installation preparation (above).

The <directory> specified as part of the `installmstools` option should be the fully-qualified path name to the directory containing the Microsoft packages that were downloaded during the offline installation preparation.

- Silent Installation

Format: `edbt_60.exe /quiet`

Summary: Full installation, but the components are installed non-interactively with no user interface, using default options and directories.

- Modified Silent Installation

Format: `edbt_60.exe /quiet InstallFolder=d:\otherdirectory`

Summary: Full installation, but the components are installed non-interactively with no user interface, and Enterprise Developer Build Tools for Windows is installed to `d:\otherdirectory`.

To see all of the available parameters, execute the following from the command line:

```
edbt_60.exe /?
```

Microsoft Package Dependencies

Various actions and operations within your COBOL development environment depend on certain files that Microsoft distributes in the following packages: the Windows SDK package and the Microsoft Build Tools package.

There is a dependency on the Microsoft Windows SDK if you:

- Produce COBOL `.exe` or `.dll` files.
- Call Windows APIs from your COBOL `.exe` or `.dll` files.
- Link a C object file into a COBOL `.exe` or `.dll` file.
- Call Windows APIs from a C object file linked into a COBOL `.exe` or `.dll` file.
- Create MIDL files for OLE RPC.

There is a dependency on the Microsoft Build Tools if you:

- Produce COBOL `.exe` or `.dll` files.
- Manage `.lib` files.
- Create PDB files, for debugging in Visual Studio.
- Convert resource files to object modules.

A standard installation downloads and installs the latest available versions of the Windows 10 SDK and the Microsoft Build Tools for Visual Studio 2017 to a default location.

If you distribute a COBOL product installation to clients that do not have access to the internet, then you need to have downloaded these packages and placed them in an accessible place on the network before clients can run an 'offline installation'. The installer includes options to achieve this: `downloadmstools` enables you to download the required packages to a specific location; and `installmstools` enables an offline client to install the packages from a local network location.

If you do not use any of the features described above, you can use an install parameter (`skipmstools=1`) that will skip the download and installation of these dependencies.

All of these installation options are discussed in more detail in the *Installation Options* topic for your product.

One of the final steps of the installation process configures the downloaded and installed package versions for use in your COBOL environment. You can view these versions using the Microsoft Build Tools and Windows SDK Configuration Utility - see *Managing the Microsoft Build Tools and Windows SDK Packages*. This utility also enables you to specify alternative versions should you need to depend on a specific package other than the one downloaded, or one stored in a non-default location.

Managing the Microsoft Build Tools and Windows SDK Packages

There are a number of COBOL development features that depend on the Microsoft Build Tools and Windows SDK packages. By default, the product installer downloads and installs these packages to their default locations. The installer then sets your COBOL environment to depend on those downloaded versions.

You can use the Microsoft Build Tools and Windows SDK Packages configuration utility - `cb1ms` - to view which versions are in use, and also configure your COBOL environment if you require something other than the standard usage.

For example, some project types (for example, .NET Core) require a different version of the Microsoft Build Tools (the Visual Studio 2019 version) than the one installed (the Visual Studio 2017 version). If you have already installed the correct version for .NET Core, run `cb1ms` after product installation, to update your environment to use the correct package.

There is an option within the product installer to skip the installation of the default packages (`skipmstools=1`); however, this does skip both packages, so make sure that you do already have the package(s) that you require before setting this - see *Microsoft Package Dependencies* for details on the package(s) you may require.

The Microsoft Build Tools and Windows SDK Configuration Utility

Use the `cb1ms` command line utility to view and set the package versions in effect for your COBOL environment.



Important: If you are updating or clearing package versions, you must run this utility from a command prompt with administrative privileges.

Syntax:

```
cb1ms <option>
```

Parameters:



Note: *<option>* settings are case insensitive.

<option> can be one of:

-U

Updates the COBOL environment to use the latest versions of the Microsoft Build Tools and SDK packages that are located in the default folders. Any open IDE instance or command prompt must be restarted before these changes take effect.

-U<ms-pkg>

Updates the COBOL environment to use the latest version of the particular package type that is located in the default folder. Any open IDE instance or command prompt must be restarted before these changes take effect.

-U<ms-pkg>:<version>

Updates the COBOL environment to use a specific version of the particular package type that is located in the default folder. Any open IDE instance or command prompt must be restarted before these changes take effect.

-U<ms-pkg>:<path>

Updates the COBOL environment to use the latest version of the particular package type that is located in the (non-default) folder specified. Any open IDE instance or command prompt must be restarted before these changes take effect.

-U<ms-pkg>:<path>?<version>

Updates the COBOL environment to use a specific version of the particular package type that is located in the (non-default) folder specified. Any open IDE instance or command prompt must be restarted before these changes take effect.

-L

Lists all versions of the Microsoft Build Tools and SDK packages that are located in the default folders.

-L<ms-pkg>

Lists all versions of a particular package type that are located in the default folder.

-L<ms-pkg>:<path>

Lists all versions of a particular package type that are located in the (non-default) folder specified.

-Q

Displays the versions currently in use by the COBOL environment.

-Q<ms-pkg>

Displays the version of the selected package type that is currently in use by the COBOL environment.

-R

Clears all version information. (Only use this option when instructed to by SupportLine.)

-H

Displays the available command line options.

Legend:

<ms-pkg> *<ms-pkg>* can be one of **S** or **SDK** for the Microsoft SDK, or **B** or **BT** for the Microsoft Build Tools.

<path> *<path>* is the path in which to locate packages that are not installed to the default location for their package type.

<version> *<version>* is the package version with which to update when the latest version is not the one required. When specifying *version*, it must be in the form:

- *n.n.n.n* for a Windows SDK
- *n.n.n* for Build Tools
- *n* for list ID - the list ID is displayed using `cblms -L`.

The following example updates the COBOL environment to use version 10.0.17134.0 of the Windows 10 SDK, installed in `C:\WinSDK\10`.

```
cblms -USDK:C:\WinSDK\10?10.0.17134.0
```

Restrictions when Using Enterprise Developer Build Tools for Windows

The following list describes elements of functionality of Enterprise Developer Build Tools for Windows that you should be aware of before installing it:

- You cannot install Enterprise Developer Build Tools for Windows on the same machine as Enterprise Developer, Visual COBOL, or Visual COBOL Build Tools for Windows.
- Installing Enterprise Developer Build Tools for Windows does not install any documentation. Instead, full information on how to use Enterprise Developer Build Tools for Windows is included in the documentation for Enterprise Developer. The documentation that describes how to use Enterprise Developer Build Tools for Windows includes links to the sections of documentation for Enterprise Developer that are relevant to Enterprise Developer Build Tools for Windows.
- To debug a COBOL application that is running in Enterprise Developer Build Tools for Windows you need to use remote debugging.

Information on using remote debugging with Enterprise Developer Build Tools for Windows is available in your Enterprise Developer documentation. See *Remote Debugging of Native COBOL Applications* if you are using Enterprise Developer for Visual Studio or *Debugging using a remote machine* if you are using Enterprise Developer for Eclipse.

- Remote debugging is not available for .NET COBOL applications, so you cannot use Enterprise Developer Build Tools for Windows to debug .NET COBOL applications (although you can use Enterprise Developer Build Tools for Windows to build and run them).

Licensing Considerations

Enterprise Developer Build Tools for Windows requires the same license as Enterprise Developer so you need to ensure that you have enough licenses to cover your use of both Enterprise Developer Build Tools for Windows and Enterprise Developer.

For example, if you have purchased six development licenses and install one instance of Enterprise Developer Build Tools for Windows you can install Enterprise Developer (or Enterprise Developer Build Tools for Windows) on a maximum of five other machines.

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 *Product Updates* section of the Micro Focus SupportLine Web site, where you can download fixes and documentation updates. Go to [Micro Focus Product Updates](#)
- The *Examples and Utilities* section of the Micro Focus SupportLine Web site, including demos and additional product documentation. Go to <https://supportline.microfocus.com/examplesandutilities/index.aspx>.
- The *Support Resources* section of the Micro Focus SupportLine Web site, that includes troubleshooting guides and information about how to raise an incident. Go to <https://supportline.microfocus.com/supportresources.aspx>

To connect, enter <https://www.microfocus.com/en-us/home/> in your browser to go to the Micro Focus home page, then click **Support & Services** > **Support**. Type or select the product you require from the product selection dropdown, and then click **Support Login**.



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, <https://www.microfocus.com/support-and-services/contact-support/>. 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.

Also, visit:

- The Micro Focus Community Web site, where you can browse the Knowledge Base, read articles and blogs, find demonstration programs and examples, and discuss this product with other users and Micro Focus specialists. See <https://community.microfocus.com>.
- The Micro Focus YouTube channel for videos related to your product. See [Micro Focus Channel on YouTube](#).

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 .

Creating Debug Files

If you encounter an error when compiling a program that requires you to contact Micro Focus SupportLine, 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.

Copyright and Disclaimer

© Copyright 2020 Micro Focus or one of its affiliates.

The only warranties for this product and any associated updates or services are those that may be described in express warranty statements accompanying the product or in an applicable license agreement you have entered into. Nothing in this document should be construed as creating any warranty for a product, updates, or services. The information contained in this document is subject to change without notice and is provided "AS IS" without any express or implied warranties or conditions. Micro Focus shall not be liable for any technical or other errors or omissions in this document. Please see the product's applicable end user license agreement for details regarding the license terms and conditions, warranties, and limitations of liability.

Any links to third-party websites take you outside Micro Focus websites, and Micro Focus has no control over and is not responsible for information on third party sites.