

Microsoft Visual C++ 2015 fails to install when C++ 2017 is already installed

Last Modified on 01/12/2022 9:34 am EST

Symptom

OnGuard installation prerequisite Microsoft Visual C++ 2015 fails to install because Microsoft Visual C++ 2017 is already installed.

Resolution

In previous versions of Visual Studio, binary compatibility between object files (OBJs), static libraries (LIBs), dynamic libraries (DLLs), and executables (EXEs) built using different versions of the compiler toolset and runtime libraries was not guaranteed. This has changed in Visual Studio 2017. In Visual Studio 2015 and Visual Studio 2017, the C++ toolset major number is 14 (v140 for Visual Studio 2015 and v141 for Visual Studio 2017). This reflects that both the runtime libraries and the applications compiled with either version of the compiler are, for the most part, binary compatible. Therefore, for example, if there is a DLL in Visual Studio 2015, we don't have to recompile it in order to consume it from an application that was built with Visual Studio 2017.

Source: <https://docs.microsoft.com/en-us/cpp/porting/binary-compat-2015-2017?view=vs-2017>

What this means is that Visual Studio 2017 should be compatible with OnGuard installations that need 2015 or later C++ libraries. Our installer is currently coded to check for Visual Studio 2015 and install it if not found. If Visual Studio 2017 exists, it will fail to install for the reason above (newer version installed) as expected.

If a customer already has Visual Studio 2017 installed on the workstation already, the current workaround is to note the C++ version installed, uninstall it, install Visual Studio 2015, and then get the installer from Microsoft for the Visual Studio 2017 version and re-install it.

The installer can be found here:

<https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads>

Applies To

OnGuard (all versions that use C++ 2015)

Additional Information
