WHITE PAPER

Intel® Software Guard Extensions (Intel® SGX)

Microsoft* Visual Studio* 2015 and Microsoft* Universal C Run Time with Intel® Software Guard Extensions (Intel® SGX) Applications

Scope

This paper explains how to use the Microsoft* Universal C Run Time (CRT) library with Intel SGX-enabled applications and Microsoft* Visual Studio* 2015, including building with and deploying the right CRT libraries. This paper is provided as background information so developers can plan deployment of their Intel SGX enabled applications to ensure the Microsoft Universal CRT is available when their applications run. General information on Intel SGX is provided on the Intel SGX portal at: https://software.intel.com/en-us/sgx.

Introduction

Some of the new features included in the latest version of Visual Studio 2015 include:

- C++ standard library — C++11.0 features support
- C Run Time library — Universal CRT library support
- Features targeting Windows 10

Before diving into details of the Universal C Run Time library, let's walk through a short introduction on C Run Time libraries.

The C Run Time library is automatically compiled in for any C program to be executed. The version of library is used depends upon the computer, platform, debugging option, and multi-threaded option in the development environment.

In Visual Studio 2015, major architectural changes were done with respect to C Run Time library support. The updated C Run Time library is divided into two parts:

- VCRuntime
- Stable part

The "VCRuntime" part, contained the compiler support functionality required for process start-up and exception handling. This part still exists in the same form without changes from earlier Visual Studio versions.

The "Stable" part contained all of the purely library parts of the CRT, in the form of two libraries: the AppCRT and the DesktopCRT. These libraries have been combined into a single library, starting with Visual Studio 2015, named Universal CRT. The new DLLs for this library

are named `ucrtbase.dll` (release) and `ucrtbased.dll` (debug) and they do not include a version number.

The Universal CRT is a component of the Windows 10 OS.

Building software using the Universal CRT

In Visual Studio versions earlier than 2015, CRT libraries, sources, and headers were distributed as part of the Visual C++ SDK. The CRT was installed in the VC subdirectory of the Visual Studio installation, in the following path:

```
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC
```

With Visual Studio 2015, however, only the "VCRuntime" part comes along with the Visual C++ SDK. The headers, sources, and libraries are now distributed as part of a *separate* Universal CRT SDK. This SDK is included with Visual Studio and is installed by default to:

```
C:\Program Files (x86)\Windows Kits\10.
```

By default, the Visual C++ MSBuild properties and targets files are updated to add the new Universal CRT directories to the `include` and `library` paths.

If a new project in Visual Studio 2015 is created or an existing project is upgraded to Visual Studio 2015, it should generally pick up these new directories automatically.

However, for an upgraded project that does not use the Visual C++ MSBuild properties and targets files, or that does not inherit the default `include` and `library` paths from those props and targets files, the project must be updated manually to include the new directories.

The following MSBuild properties are used to find the Universal CRT SDK files:

```
$(UniversalCRT_IncludePath)
$(UinversalCRT_LibraryPath_x86)
$(UniversalCRT_LibraryPath_x64)
$(UniversalCRT_LibraryPath_arm)
```

If a Visual studio project is *not* linked with the `/nodefaultlib` option, all of the correct library files will be found and linked when the project is linked.

If a Visual studio project *is* linked with the `/nodefaultlib` option, several extra libraries must be identified and linked. For example, when an earlier version of a Visual Studio project that was linked with `msvcrt.lib` to use the CRT DLL is upgraded to Visual Studio 2015, the project must be linked with `vcruntime.lib` and `ucrt.lib`. Table 1 lists the VS2015 build modes and their corresponding library requirements.

Table 1. VS2015 Build Modes/Required Libraries

Build Mode	Required Libraries		
Release DLLs (/MD)	msvcrt.lib	vcruntime.lib	ucrt.lib
Debug DLLs (/MDd)	msvcrtd.lib	vcruntimed.lib	ucrtd.lib

Deployment in Visual C++

If the Visual C++ application is be deployed to other computers, both the application and the library it depends on must be installed on those systems. For example, if the target system has an earlier version of the Windows OS than the development environment, the versions of libraries on the target system will be older. So, a proper deployment mechanism is needed to install the necessary libraries for each different target environment. The following subsections discuss Visual C++ and the Universal CRT separately.

Visual C++ deployment methods

1. **Central Deployment:**
   Visual C++ files are installed in the `%windir%\system32\` directory. One of the following sub-methods can be used.

   - **Redistributable package file**
     This is a stand-alone command-line executable that can be used to install the Visual C++ redistributable libraries.

   - **Redistributable merge modules**
     These are used to deploy specific libraries and are included in the application's window installer (.msi) file.

2. **Local Deployment:**
   Library files are installed in the application folder together with the executable file.

3. **Static Linking:**
   Library files are statically linked with the executable. So there is no need to deploy Visual C++ libraries separately.

Universal CRT deployment methods

Deploying the Visual C++ runtime does not deploy the Universal CRT runtime.

The Universal CRT is a Windows 10 component. The Universal CRT is distributed through Windows updates for Window versions prior to Windows 10. There are Windows Update MSU packages for Windows Vista through Windows 8.1.

The Visual Studio 2015 VCRedist will both install the Visual C++ libraries and Universal CRT libraries. Therefore, this is the recommended deployment mechanism.

> **Note:** Prior to Visual Studio 2015, the Intel SGX PSW Installer included the VS Runtime Merge Modules, which were installed by the PSW `.msi`. This deployment mechanism covered Windows 7.x, 8.x, and 10.x. But since Visual Studio 2015 Merge Modules do not include the Universal CRT, the Intel® SGX PSW installer started requiring that it already be present.

Summary

With Visual Studio 2015, Microsoft changed the way that the CRT is presented (libraries) and made available (CRT SDK). Developers need to understand how to make sure the Universal CRT libraries are included in their Visual Studio projects.

Developers also need to understand the various target systems their application will run on and implement a flow that makes sure the right Universal CRT libraries are/get installed on the target systems to support their application.

The recommended deployment method for the Universal CRT on Windows 7.x-8.x is to have it installed by VCRedist*.exe.

Another option is to have the Universal CRT installed by MSU Updates for Windows 7.x-8.x.

The Universal CRT library is a Microsoft Windows 10 component, so no updating is needed.

References

1. “Deployment in Visual C++” — MSDN
2. Introducing the Universal CRT – Visual C++ Team Blog — 2015 James McNelis
3. “Intel SGX SDK Developer Reference for Windows OS” — 2016 Intel Corporation