

## Windows® 10 IoT Enterprise: Multi-Project / Multi-Platform Support in SIM

By Sean D. Liming and John R. Malin

Annabooks – [www.annabooks.com](http://www.annabooks.com)

July 2018

Windows Embedded Standard 7 (WES7) was a break from the thousands of components and component creation that was in Windows XP Embedded. The build tool, Image Configuration Editor (ICE), was a derivative of System Image Manager (SIM), and ICE built images from a single distribution share, where OEMs could support multiple platforms and multiple projects. Windows Embedded 8 Standard (WE8S) kept the same concept going, but starting with Windows Embedded 8.1 Industry, Microsoft moved away from the componentized Embedded operating system and custom tools. WE8.1, and now Windows 10, use System Image Manager (SIM) to help create custom OS installers. For Windows 10, distribution shares can be placed anywhere in the file system and not fixed to a single location as with WES7. SIM implementation allows OEMs to set up a distribution share for each project, but for those who were used to a single distribution share for all projects and platforms, the SIM implementation appears to be a step backward. The advantage of a single distribution share was that it was easy to back up all projects and share them with other developers.

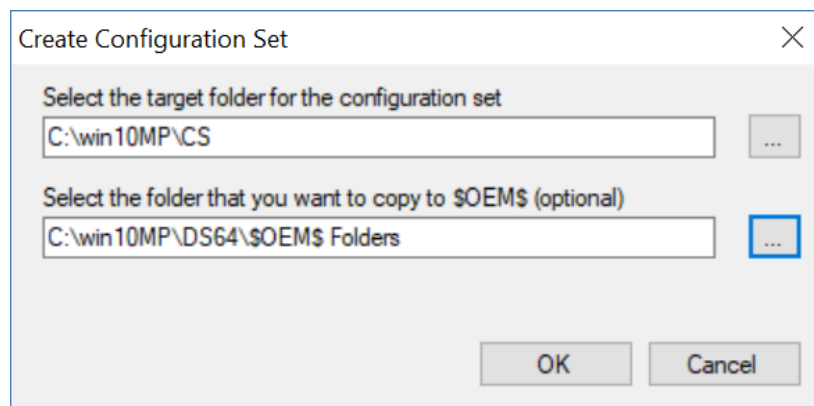
Multiple platforms are not an issue. Multiple drivers can be grouped in individual platform folders under the Out-of-box drivers folder just like WES7. The difference is in the *\$OEM\$ Folders* tree where custom applications and other installers are placed. Based on [System Image Manager documentation](#) the *\$OEM\$ Folders* didn't support multiple layers. In WES7's ICE, the Windows folder can be set up as follows:

```
DSSP1\%OEM$ Folders\\%OEM%\$$
```

\$\$ is the shorthand tag for the Windows folder. This folder setup allowed for different projects based on the project name. The *\$OEM\$* was a flag for the installer to install the files under the folder in the specified locations based on the folders and shorthand tags. According to the SIM documentation, the distribution share setup for the Windows directory is as follows:

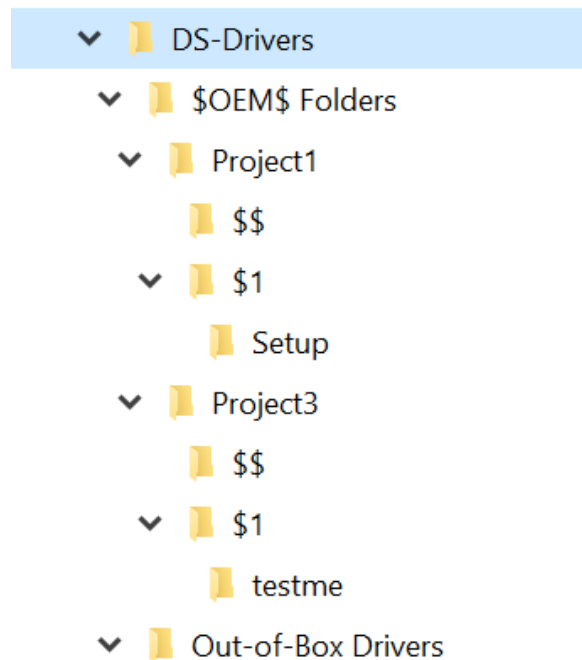
```
DS\%OEM$ Folders\$$
```

As you can see, there is no facility specified for multiple projects. There are other major differences between ICE and SIM. In ICE, you can point to the project and add it to the answer file so there is a link between the answer file and the *%OEM% Folders*. There is no link between the answer file and the *\$OEM\$ Folders* in SIM. Instead, when the build operation to generate the configuration set is started, that is when you point to the *\$OEM\$ Folders*. All the path information shows the absolute path to the *\$OEM\$ Folders*.

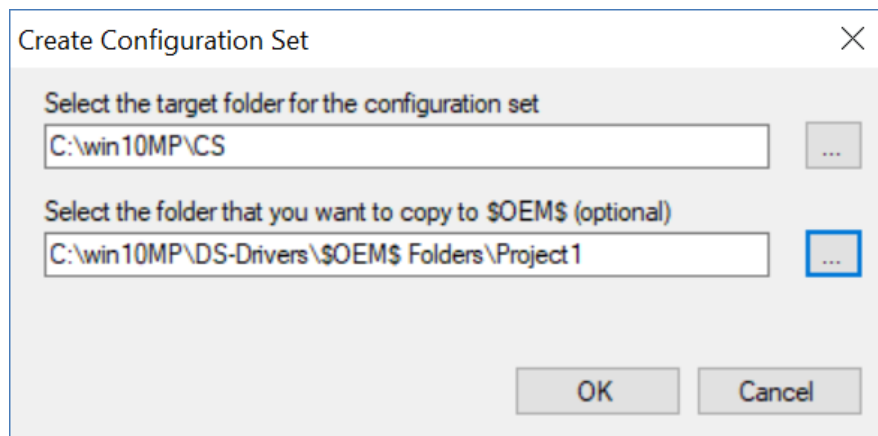


When WE8.1I first came out, I tried to just copy over a WES7 project and use it with SIM. I copied the project folder and contents to the *\$OEM\$ Folders* directory. It was a logical approach since most clients moving from WES7 to WE8.1I will just want to re-use the files from their WES7 build as is. Following the documentation, I simply pointed SIM to the *\$OEM\$ Folders* directory to build the configuration set. In the end, the files in the *\$OEM\$ Folders* directory tree failed to be installed, thus it appeared that multi-project builds weren't possible. Several of our clients were used to a single distribution share with multiple projects and multiple platforms. Not having multiple projects was not an easy message to deliver.

There had to be a way to solve this problem, and I let the idea noodle around in the back of my mind for a while. The solution came by throwing out what the SIM documentation says. I set up the *\$OEM\$ Folders* directory within the project folders using the recommended shorthand tags for Windows (\$\$) and drive C: (\$1). No extra *\$OEM\$* folder directories were created. Text files were placed in the various subfolders.



An answer file was set up, but the configuration set build was a little different. This time rather than stopping at *\$OEM\$ Folders*, I opened the project folder (Project 1) under the *\$OEM\$ Folders*.



The result was a success. The files in the `$OEM$ Folders\Project 1` folder were installed as expected. A repeat of the same steps for Project 3 proved success, confirming that this was, in fact, the solution to this problem. The *Project 1* in the example can be any project name, there can be multiple project folders under the `$OEM$ Folders` path, and each project can be independently built by calling that project out (`$OEM$ Folders\Project`) in the “Select the folder that you want to copy to `$OEM$`” option of the Create Configuration Set dialog.

Windows is a registered trademark of Microsoft Corporation  
All other copyrighted, registered, and trademarked material remains the property of the respective owners.