

## Steps to hook and deploy customizations with the ProjectWise Custom Module Manager

The recommended workflow to create, test, and deploy ProjectWise Custom Modules to a ProjectWise client is the following:

1. Open the appropriate version of ProjectWise Custom Module Manager:
  - a. For 32-bit ProjectWise custom modules (More Common):
    - Start Menu > Programs > Bentley > ProjectWise V8i (SELECTseries X) > Administrative Tools > Custom Module Manager x86
    - Registry location:  
HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Bentley\ProjectWise\08.11\CustomModules
  - b. For 64-bit ProjectWise custom modules (Less Common):
    - Start Menu > Programs > Bentley > ProjectWise V8i (SELECTseries X) > Administrative Tools > Custom Module Manager x64
    - Registry location:  
HKEY\_LOCAL\_MACHINE\SOFTWARE\Bentley\ProjectWise\08.11\CustomModules
2. “Add” your custom module to the list of “Registered Modules”

NOTE: It is recommended to place your custom modules in the ..\ProjectWise\bin directory. Doing so provides ease of testing, deployment, and module loading.
3. Provide the appropriate information for these details about your customization:
  - a. Module name: // Base file name of your custom module
  - b. Library name: // Full path and file name of your custom module (.dll)
  - c. Function name: // Function or procedure entry point name
  - d. Load for applications: // Specific application list that your custom module will be used in. e.g. Typically: “ProjectWise Explorer”
  - e. User mask: // This value is automatically generated for you by selecting the applications in the list above
4. Once you have registered and have validated proper execution of your custom module open Microsoft Regedit and export out the necessary registry key location(s) referenced in Step 1a and/or 1b appropriate for your customization(s) saving as a .reg file.
5. Since you may have both 32-bit and 64-bit custom modules you will need to target the appropriate registry file to the appropriate platform(s). For instance, if you have both a 32-bit and 64-bit version of your custom module you may have a need to import both your 32-bit and 64-bit registry files for the respective target OS and programs for one environment, and only a need to import the 32-bit registry file for users running a 32-bit OS and 32-bit programs.

6. Bentley does not recommend one specific deployment vehicle over another since each users deployment needs and policies vary widely. Some deployments can be as simple as sending an email that your users can follow hyper links to custom 32-bit or 64-bit batch/command files. Another way can be executing a login script that pulls changes when the user logs in through a custom batch(bat) or command (cmd) file. Other large scale deployments may require custom install packages to be created, tested, approved, and rolled out through e.g. Microsoft Active Directory policies or SMS updates. If doing the former you can identify your target OS bitness using something similar to: if defined ProgramFiles(x86) (set WOW6432Node=Wow6432Node\&& set OSARCH=64) else (set OSARCH=32). There are also numerous techniques for querying and verifying installed software that can be addressed through any number of Microsoft technologies from batch/command files through custom installers or programming.