Preparing your manifest file

> For a description of the manifest file format, see the Manifest file section.

In order to prepare a manifest file for distribution you need to run the ManifestPrep utility on it which is found in the ***tools*** directory of the SDK.

ManifestPrep essentially does two things:

1. Hashes and records information about dependent files.
2. Signs the manifest.

Overview

ManifestPrep is a utility that prepares your package's manifest file so that your package can be used with a retail licensed (non-developer) WinGate. The tool generates information about your package, places it into the manifest and signs it. WinGate will then load the manifest file, check that it is signed and validate the information in it against the actual files themselves. If anything is deemed to be suspect then the package will not be loaded.

If you used the WinGate Module Wizard to create your package, the manifest file is created for you in the solution directory and you need only run ManifestPrep on it. If you did not use the WinGate Module Wizard you will need to create a manifest file by hand.

Documentation of the manifest file format is coming soon.

Running ManifestPrep

Run ManifestPrep specifying the path to the manifest file, the output manifest name and the name (or part thereof) of the certificate that you want to use to sign the manifest. e.g.

ManifestPrep.exe MyModule.manifest.xml MyModule.manifest "My Certificate"

By default ManifestPrep will look in the 'My' certificate store but you can specify the store to look in with the -S command line option. See below:

NOTE: When testing your package, if you used the WinGate Module Wizard to register your package with WinGate, then the output name of the manifest MUST be the same as the input name, minus the .XML suffix as demonstrated above. When you create your own installer for your package you can name the manifest anything you wish and place the path to the manifest file in the correct WinGate\Packages key. See below for more information on the Packages key.

%SourceRoot%

Inside the manifest file generated by the WinGate Module Wizard is a <signInfo> node for each file the package depends on. This node is there to assist manifest prep in finding the source file for that dependency. This is typically inside a developer's Release directory. The WinGate Module Wizard generates this node for you if you create your projects with it.

When ManifestPrep runs, it expands %SourceRoot% within the signInfo path and uses that to find the file. %SourceRoot% defaults to the current working directory.

Let's assume you have a directory structure like so: (ManifestPrep is in your path.)

C:\source\MyWinGatePackage\Release

Example 1:
You are in the c:\source\MyWinGatePackage directory. The signInfo node in the manifest.xml contains %SourceRoot%\Release.

ManifestPrep MyPackage.manifest.xml MyPackage.manifest "My Certificate"

Will sign and generate a retail manifest file by replacing %SourceRoot% with the current working directory and appending "Release".

Example 2:
You are in some other directory. The signInfo node in the manifest.xml contains %SourceRoot%\Release.

ManifestPrep -R c:\source\MyWinGatePackage MyPackage.manifest.xml MyPackage.manifest "My Certificate"

Will sign and generate a retail manifest file using "c:\source\MyWinGatePackage" as the base path and adding "Release" to that in order to find the correct file.

Deploying your package

Once manifest prep has been successfully run you need to package up the manifest file and the dependency files in your preferred format. Where they are installed is up to your installer and the end user.

NOTE: At present, when you install the files they need to all reside in the same directory as the manifest file; there is no support for sub-directory dependencies.

Getting WinGate to load your package

In order to get WinGate to load your package you need to add a registry key for the package under HKLM\Software\Qbik Software\WinGate\Packages. The name of the key must be the UUID of the package in your manifest file including any braces e.g {05b70dd3-55f0-4958-abc3-15a2aebe5ffe}. The WinGate Module Wizard generates this package UUID for you so you can just look in the raw manifest.xml file to find it.

NOTE: Do not use the module UUID(s), it must be the package UUID or WinGate will not load your package.

Now you have added the key, add a string value under it called Manifest, with the data being the complete path to your manifest file, including the name. e.g c:\program files\my package\MyPackage.manifest

When you next start WinGate, your package should load.

NOTE: The actual dependency files themselves do NOT need to be signed. The manifest file contains all the information it needs to validate the files and ensure they haven't been tampered with.

Other ManifestPrep Options

-L List the available certificates in all system stores
-LS List the available certificate system stores
-P Show progress
-D Show debug information
-S storename Use a specific store (For listing or signing. Default is 'My'
-R rootdirectory Specify the root directory for files usable with %SourceRoot%