

# Testing your module

If you have everything set up correctly and have successfully built your module(s) then running the appropriate application, either the WinGate service or the WinGate Management Console should load your module(s). You can check the state of your module(s) from the Modules panel in the WinGate Management Console.

## Running in the debugger

### To run an engine (WinGate service) module in the debugger

Set up your development environment to run WinGate.exe with the command line argument -force9x. In Visual Studio 2010 these settings can be found Project Properties -> Configuration Properties -> Debugging. Set **Command** to the path to WinGate.exe which is typically c:\Program Files (x86)\WinGate\WinGate.exe and set **Command Arguments** to -force9x.

By default WinGate and the WinGate Management Console will load the Release build of your module. If you wish to load the Debug version of your plugin from within the IDE then specify -debug as an additional command argument.

### To run a UI (WinGate Management Console) module in the debugger

Set the **Command** option to the GateKeeper executable (e.g c:\Program Files (x86)\WinGate\GateKeeper.exe) and specify -debug if you wish to run the debug version of your module.

### -debug

The configuration specific paths to your module files are specified in the manifest file attributes **debugPath** and **releasePath** which are automatically set up by the WinGate Module Wizard to point to the **Debug** and **Release** sub folders of the module you are developing. By default WinGate and the WinGate Management Console will load your module from the **releasePath** folder. If you wish to load your module from the **debugPath** folder, specify the -debug flag when running WinGate and/or the WinGate Management Console.

**NOTE:** WinGate and the WinGate Management Console work separately, so each has to have their own flags set independently.