**Creating Assemblies**

**.NET Framework 4.5**

You can create single-file or multifile assemblies using an IDE, such as Visual Studio 2005, or the compilers and tools provided by the Windows Software Development Kit (SDK). The simplest assembly is a single file that has a simple name and is loaded into a single application domain. This assembly cannot be referenced by other assemblies outside the application directory and does not undergo version checking. To uninstall the application made up of the assembly, you simply delete the directory where it resides. For many developers, an assembly with these features is all that is needed to deploy an application.

You can create a multifile assembly from several code modules and resource files. You can also create an assembly that can be shared by multiple applications. A shared assembly must have a strong name and can be deployed in the global assembly cache.

You have several options when grouping code modules and resources into assemblies, depending on the following factors:

* Versioning

Group modules that should have the same version information.

* Deployment

Group code modules and resources that support your model of deployment.

* Reuse

Group modules if they can be logically used together for some purpose. For example, an assembly consisting of types and classes used infrequently for program maintenance can be put in the same assembly. In addition, types that you intend to share with multiple applications should be grouped into an assembly and the assembly should be signed with a strong name.

* Security

Group modules containing types that require the same security permissions.

* Scoping

Group modules containing types whose visibility should be restricted to the same assembly.

Special considerations must be made when making common language runtime assemblies available to unmanaged COM applications.