**Metadata and Reflection**

The reason for using metadata is simple. It allows the .NET runtime to know at run time what types will be allocated and what methods will be called. This enables the runtime to properly set up its environment to more efficiently run the application. The means by which this metadata is queried is called *reflection*. In fact, the .NET Framework class libraries provide an entire set of reflection methods that enable any application—not just the CLR—to query another application's metadata.

Tools such as Visual Studio.NET use these reflection methods to implement features such as IntelliSense. With IntelliSense, as you type in a method name, that method's arguments pop up in a list box on the screen. Visual Studio.NET takes that functionality even further, showing all the members of a type.

Another incredibly useful .NET tool that takes advantage of reflection is the Microsoft .NET Framework IL Disassembler (ILDASM). This powerful utility parses the target application's metadata and then displays information about the application in a treelike hierarchy.

