CHEAT SHEET

This cheat sheet is designed as a way for you to quickly study the key points of this chapter.

Reflection

- There are two ways to get a reference to the `Type` object, using `typeof()` or the `.GetType()` method on an object.
- You can use the `System.Reflection.Assembly` class to examine the types within an EXE file or DLL file.
- The `Assembly.Load` method loads an assembly into memory and enables you to execute code.
- The `Assembly.ReflectionOnlyLoad` method loads the assembly into memory, but you cannot execute any code.
- The `Assembly.CreateInstance` method creates an instance of a type.
- The `System.Type` class represents a class, interface, array, value type, enumeration, parameter, generic type definitions, and open or closed constructed generic types.
- The `Type.GetProperty` method returns a `PropertyInfo` object and enables you to set or get a property's value.

Attributes

- Attributes enable you to create metadata for a class, a property, or a method.
- Attributes are contained in square brackets `[]` just above the target.
- Custom attributes must inherit from the `System.Attribute` class.

Code Document Object Model (CodeDOM)

- The CodeDOM is a set of classes that enables you to create code generators.
- The `System.CodeDom.CodeCompileUnit` class is the top-level class; this is the container for all other object within the class you want to generate.
- The `System.CodeDom.CodeDOMProvider` class generates the class file in either C#, VB, or JScript.

Lambda expressions

- Lambda expressions are shorthand syntax for anonymous functions.
- A delegate is a type that references a method.
- Covariance enables you to have a method with a more derived return type than the delegate's return type.
- Contravariance permits parameter types that are less derived than the delegate type.
- The => in a lambda expression is referred to as "goes to."

REVIEW OF KEY TERMS

anonymous method Enables you to associate a block of code with a delegate without declaring the method signature.

assembly A compiled piece of code in a DLL or EXE file.

attribute Enables you to associate metadata with assemblies, types, methods, properties, and so on.

Code Document Object Model (CodeDOM) Enables the developer to generate code in multiple languages at run time based on a single code set.

context When loading an assembly using reflection, the context is where reflection searches for the assembly.

contravariance Permits parameter types that are less derived than the delegate's parameter types.

covariance Enables you to have a method with a more derived return type than the delegate's return type.

delegate A type that references a method.

expression lambda A lambda expression that contains only one statement for the body.

Expression Tree Code in a tree-like structure where each node is an expression.

field A variable defined in a class or struct.

lambda expression Shorthand syntax for an anonymous method that can be associated with a delegate or expressions tree.

load context When loading an assembly using reflection, this context contains the assemblies found by probing.

load-from context When loading an assembly using reflection, this context contains the assemblies located in the pat passed into the `LoadFrom` method.

module A file that composes an assembly. Typically this is the DLL or EXE file.

probing The process of looking in the GAC, the host assembly store, the folder of the executing assembly, or the private bin folder of the executing assembly to find an assembly.

reflection Provides classes that can be used to read metadata or dynamically invoke behavior from a type.

reflection-only context When loading an assembly using reflection, this is the context that contains the assemblies loaded with the `ReflectionOnlyLoad` and `ReflectionOnlyLoadFrom` methods.

statement lambda A lambda expression with more than one statement in the body of the expression.

target The class, property, or method that contain metadata defined by an attribute.

type Any class, interface, array, value type, enumeration, parameter, generic type definition, and open or closed constructed generic type.

EXAM TIPS AND TRICKS

The Review of Key Terms and the Cheat Sheet for this chapter can be printed off to help you study. You can find these files in the ZIP file for this chapter at `www.wrox.com/remtitle.cgi?isbn=1118612094` on the Download Code tab.