Appendix D

Working with .NET Serviced Components

In order for serviced components to fire the necessary instrumentation events, it is necessary to define a public interface for the component and to set certain attributes.

The following code sample illustrates the needed attribute settings and interface.

Code Prerequisites:

```
// Here we define the public interface for our Example class.
[ ComVisible( true ) ]
public interface IExample
{
      int Method1(int Parm1);
}

// The class interface type MUST be set to AutoDual
[ ClassInterface(ClassInterfaceType.AutoDual) ]
[ ComVisible( true ) ]

// The following attribute is optional, but if it isn't set in the
// code, then the "Component supports events and statistics" checkbox
// on the component's Activation properties page must be checked
// manually in the ComponentServices management console.
[ EventTrackingEnabled ]

// JIT activation defaults to off for components which are configured
// in COM+, but is enabled automatically if automatic transactions are
// requested.
//
// Here we'll set it to on in our example
[ JustInTimeActivationAttribute ]

public class Example : ServicedComponent, IExample
{
      public int Method1(int Parm1)
      {

            return (Parm1);

      }
}
```

Once built, the ServicedComponent application would then be deployed as usual, while ensuring that it has been added to the COM+ catalog by either using the COM+ management console or the RegSvcs.exe utility.