Visual Studio Analyzer is a tool for collecting analysis information about distributed applications. Visual Studio Analyzer comes with both of the Enterprise editions of VS .NET 2003: VSEA and VSED. It is a high-level tool that works at the component level, effectively helping you isolate the components of your application that potentially need optimizing. Most profiling tools, such as code profilers, generally work at the method level, profiling the code in your methods. You'll use those kinds of profiling tools once you've identified the components of your distributed application that need to be optimized by using Visual Studio Analyzer.

In this chapter, you'll see how you can use Visual Studio Analyzer for analyzing your distributed application's performance. Visual Studio Analyzer complements Microsoft Application Center Test (ACT), which you can find more information about in Chapter 13. Although this is a short survey chapter, we're confident that you'll get to know the basics of Visual Studio Analyzer, so you'll know when and for what purpose to use it.

**Visual Studio Analyzer Architecture**

Visual Studio Analyzer is really nothing more than a client/server application. The server is the computer that hosts the components to be analyzed, and the client is the computer that collects results from the analysis performed on the server. In practice, the server and client can be the same computer.

The Visual Studio Analyzer client is responsible for collecting the analysis results from the Visual Studio Analyzer server, and also possibly displaying those results. The Visual Studio Analyzer server collects results from the components running on the computer on which it is installed. Figure 14-1 shows a very simple overview of a Visual Studio Analyzer setup, with two servers.
Now, although this simple setup is correct, there is obviously more to it than that. The event collection and analysis works like this:

1. On each server, a Local Event Concentrator (LEC) is responsible for collecting events, such as writing to disk or communicating with a database server, from an event source. An event source is any component that has been activated by the LEC.

2. The component/event source generates events using an In-process Event Creator (IEC) that the LEC collects and filters. This filter is set by you using the Visual Studio Analyzer interface.

3. The events that are filtered are passed back to the Public Event Collector (PEC) on the client that initiated the analysis.

4. The PEC saves the events to an event log.

Figure 14-2 illustrates the Visual Studio Analyzer event collection and analysis process. You can see how the PEC sends filters to the LECs, which in turn activate the event sources on the local computer, based on the filter received. The event sources use the IEC to generate events that are then collected by the LEC, which sends them back to the PEC.