

has been a major constraint to keep in mind when designing for COM+, but in COM+ 1.5, it will be possible to configure the needed isolation level.

## **Applications as Windows Services**

Some of the component services, such as queued components, need a started COM+ application to work. In COM+ 1.5 it is possible to run COM+ applications as Windows services. This also comes in handy if you like to run the application as the local system account, to make it possible to act with the privileges from that user. This is also a solution to the short delay that might otherwise be apparent when the first request is made to a COM+ server application after system boot (or after the application has been shut down).

## **Memory Gates**

The Memory Gates feature prevents the creation of objects when the amount of free memory falls below a certain level. The reason for this functionality is to avoid adding objects when there's not enough memory to support their operations. Strange errors occur when a system runs out of memory, and it is hard to trap for errors in components in this environment. In fact, most components aren't tested for low memory conditions, so it is better that they don't get called at all if the system has run out of memory.

## **Pause/Disable Applications**

In COM+ 1.0, you can't pause a COM+ application. Instead, you have to delete the component if, for any reason, you don't want anybody to use it for a certain amount of time. In COM+ 1.5, you can both pause and disable applications.

## **Process Dumping**

To make debugging easier in COM+ 1.5, it will be possible to take a snapshot of the process without having to stop the process. This snapshot can then be used with a debugger to examine the exact state when the problem occurred.

## **Moving and Copying Components**

Often, it is useful to be able to let the same component appear several times, in several applications. That is especially the case for helper components. Assume that one helper component is used by five different COM+ applications. In COM+ 1.0, all five applications have to be upgraded at the same time to use the new version of the helper. I generally prefer to upgrade step-by-step instead. Therefore, it's common that helpers are reused in source-code form instead of as binary components. In COM+ 1.5, it will be possible to configure the same physical component several times and in different ways.

## **Public/Private Components**

In COM+ 1.0, all components are public. In COM+ 1.5, it will be possible to set a component to be private to the current application. The less you expose externally, the easier it is for you to make changes in the future to the application.

## **Application Partitions**

The purpose of application partitions is to partition a machine so that applications don't disturb each other. A typical situation when application partitions are of great use is in data centers. Consider the possibility of running several instances of SQL Server 2000 in one single machine.

## **COM+ Applications Can Be Exposed as XML Web Services**

If you want to expose some functionality of a COM+ application as XML Web services, COM+ 1.5 makes that easy. You can achieve that automatically with the help of the Component Services Explorer.

### **Note**

Now that I have sounded like a salesperson for a while, I'd like to say that I was told by Microsoft that there are no plans to add the functionality of COM+ 1.5 to Windows 2000. Only time will tell.

## **Visual Basic .NET**

I, for one, have been longing for Visual Basic .NET for a long time. Why, you ask?