Appendix: Windows DDK

Registry Enhancements

As with Windows 2000, the registry plays a key role in the configuration and control of Windows XP. The registry, which resides on the disk as multiple files called *hives*, was originally designed as a repository for system configuration data. Although most people think of the registry as static data stored on the hard disk, it is also a window into various in-memory structures maintained by the Windows XP executive and kernel.

The registry code is redesigned for Windows XP, providing enhanced performance while remaining transparent to applications by using existing registry programming interfaces. Windows XP registry enhancements provide performance improvements, including the following areas:

- Converting a delayed close table to a Least Recently Used (LRU) list.
- Reducing Kernel Control Block (KCB) lock contention with do not lock registry exclusive and do not touch volatile information.
- Providing a security cache to eliminates duplicate security descriptors.

The new registry implementation delivers two key benefits:

- Larger registries
- Faster queries

Larger Registries

Windows XP supports larger registries than previous versions of the kernel, which were effectively limited to about 80 percent of the total size of paged pool. The new implementation is limited only by available system disk space.

A tendency to use the registry more like a database developed among registry consumers, which increased demands on registry size. The original design of the registry kept all of the registry files in the paged pool, which, in the 32-bit kernel, is effectively limited at approximately 160 MB because of the layout of the kernel virtual address space. A problem arose because, as larger registry consumers such as Terminal Services and COM appeared, a considerable amount of paged pool was used for the registry alone, potentially leaving too little memory for other kernel-mode components.

Windows XP solves this problem by moving the registry out of paged pool and using the cache manager to do an in-house management of mapped views of the registry files. The mapped views are mapped in 256-KB chunks into system cache space instead of paged pool.

Faster Queries

Another issue that affected registry performance in earlier versions is the locality problem. Related cells are spread through the entire registry files. Accessing certain information, such as attributes of a key, could degenerate into page-faults, which lowers performance.

The Windows XP registry uses an improved algorithm for allocating new cells that keeps related cells in closer proximity — such as keeping cells on the same page or nearby pages, which solves the locality problem and reduces the page faults incurred when accessing related cells. A new hive structure member tracks freed cells instead of relying on linked freed cells. When future cells are allocated, the freed cell list and a vicinity argument are used to ensure the allocation is in the same bin as the hive.

Windows XP improves the way the registry handles big data. In versions before Windows XP, if an inefficient application constantly increased a value with a small increment, it created a sparse and wasteful registry file. Windows XP solves this problem with a big cell implementation where cells larger than 16 KB are split into increments of 16-KB chunks. This reduces fragmentation when the data length of a value is increased within a certain threshold.

Built on Wednesday, July 24, 2002