GetDiskFreeSpaceEx

The **GetDiskFreeSpaceEx** function retrieves information about the amount of space available on a disk volume: the total amount of space, the total amount of free space, and the total amount of free space available to the user associated with the calling thread.

```
BOOL GetDiskFreeSpaceEx(
  LPCTSTR lpDirectoryName,
  PULARGE_INTEGER lpFreeBytesAvailable,
  PULARGE_INTEGER lpTotalNumberOfBytes,
  PULARGE_INTEGER lpTotalNumberOfFreeBytes
);
```

Parameters

lpDirectoryName
   [in] Pointer to a null-terminated string that specifies a directory on the disk of interest. If this parameter is NULL, the function uses the root of the current disk. If this parameter is a UNC name, it must include a trailing backslash (for example, \\MyServer\MyShare\).

   Note that this parameter does not have to specify the root directory on a disk. The function accepts any directory on the disk.

lpFreeBytesAvailable
   [out] Pointer to a variable that receives the total number of free bytes on the disk that are available to the user associated with the calling thread. This parameter can be NULL.

   If per-user quotas are in use, this value may be less than the total number of free bytes on the disk.

lpTotalNumberOfBytes
   [out] Pointer to a variable that receives the total number of bytes on the disk that are available to the user associated with the calling thread. This parameter can be NULL.

   If per-user quotas are in use, this value may be less than the total number of bytes on the disk.

lpTotalNumberOfFreeBytes
   [out] Pointer to a variable that receives the total number of free bytes on the disk. This parameter can be NULL.

Return Values

If the function succeeds, the return value is nonzero.

If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Remarks

Note that the values obtained by this function are of type **ULARGE_INTEGER**. Be careful not to truncate these values to 32 bits.

Windows 95 OSR2 and later: The **GetDiskFreeSpaceEx** function is available beginning with Windows 95 OEM Service Release 2 (OSR2). To determine whether **GetDiskFreeSpaceEx** is available, call **GetModuleHandle** to get the handle to Kernel32.dll. Then you can call **GetProcAddress**.

The following code fragment shows one way to do this:

```
pGetDiskFreeSpaceEx = GetProcAddress( GetModuleHandle("kernel32.dll"),
                        "GetDiskFreeSpaceExA");

if (pGetDiskFreeSpaceEx)
{
   fResult = pGetDiskFreeSpaceEx (pszDrive,
             (PULARGE_INTEGER)&i64FreeBytesToCaller,
             (PULARGE_INTEGER)&i64TotalBytes,
             (PULARGE_INTEGER)&i64FreeBytes);

// Process GetDiskFreeSpaceEx results.
}

else
{
   fResult = GetDiskFreeSpace (pszDrive,
             &dwSectPerClust,
             &dwBytesPerSect,
             &dwFreeClusters,
             &dwTotalClusters)

// Process GetDiskFreeSpace results.
```

```
        }
```

It is not necessary to call **LoadLibrary** on Kernel32.dll because it is already loaded into every process address space.

Windows NT Server 3.51 and earlier, Windows 95: Include an additional header file called NewAPIs.h to make **GetDiskFreeSpaceEx** available on these operating systems. The function is not implemented natively, but by a wrapper that utilizes other native functions on these systems. See the header file for details of the use of preprocessor directives that make the function available. If you do not have this header file, it can be obtained by downloading the most recent Windows SDK from the SDK Update Site.

Windows Me/98/95 OSR2: GetDiskFreeSpaceExW is supported by the Microsoft Layer for Unicode. To use this, you must add certain files to your application, as outlined in Microsoft Layer for Unicode on Windows 95/98/Me Systems.

Requirements

Client: Included in Windows XP, Windows 2000 Professional, Windows NT Workstation 4.0, Windows Me, Windows 98, and Windows 95 OSR2 and later.
Server: Included in Windows Server 2003, Windows 2000 Server, and Windows NT Server 4.0.
Unicode: Implemented as Unicode and ANSI versions. Note that Unicode support on Windows Me/98/95 requires Microsoft Layer for Unicode.
Header: Declared in Winbase.h; include Windows.h.
Library: Use Kernel32.lib.

See Also

Disk Management Functions, **GetDiskFreeSpace,**, **GetModuleHandle**, **GetProcAddress**

Platform SDK Release: February 2003

What did you think of this topic?   Order a Platform SDK CD