```
//
// ClusterSize.cpp : Determine the cluster size for a given volume
//
// Usage:       Clustersize [volume]
//
// volume can be any of the drive letters(A-Z) with a root directory path (e.g. C:\)
//
// Examples:
//              clustersize      (returns cluster size of C:\)
//              clustersize d:\  (returns cluster size of D:\)
//

#include <stdio.h>
#include <windows.h>

// Local functions
DWORD GetClusterSize(LPCWSTR pszVolume, LPDWORD pClusterSize, LPDWORD pSectorsPerCluster, LPDWORD pBytesPerSector);

int wmain(int argc, WCHAR* argv[])
{
    DWORD rc = ERROR_SUCCESS;
    DWORD ClusterSize = 0, SectorsPerCluster, BytesPerSector;
    LPCWSTR pVolume = L"C:\\";

    // if the user has picked another drive besides C:, just use that
    if(argc == 2)
    {
        // might be useful to do some parameter checking on the arguments
        // but for now we assume that we have something good
        pVolume = argv[1];
    }

    // be sure not to put up a popup for failed accesses
    SetErrorMode(SEM_FAILCRITICALERRORS);

    wprintf(L"Determining cluster size for volume %s\n", pVolume);

    // Get the actual cluster size
    rc = GetClusterSize(pVolume, &ClusterSize, &SectorsPerCluster, &BytesPerSector);

    // print out more information about the volume if possible
    switch(rc)
    {
        case ERROR_SUCCESS:
            wprintf(L"Volume(%s) ClusterSize(%u) SectorsPerCluster(%u) BytesPerSector(%u)\n",
                    pVolume, ClusterSize, SectorsPerCluster, BytesPerSector);
            break;

        case ERROR_PATH_NOT_FOUND:
            wprintf(L"The system cannot find the path specified.\n");
            break;
```

```
        case ERROR_NOT_READY:
            wprintf(L"The device is not ready.\n");
            break;

        case ERROR_INVALID_PARAMETER:
            wprintf(L"The parameter is incorrect.\n");
            break;

        default:
            wprintf(L"Volume(%s) Error(%u)\n", pVolume, rc);
            break;
    }

    return ClusterSize;
}

/*
 * GetClusterSize
 *
 * Determine the cluster size for a given volume
 *
 * Parameters:
 *      LPCWSTR pszVolume       - name of volume (example C:\)
 *      LPDWORD pClusterSize    - returned size of cluster (in bytes)
 *
 * Return:
 *      DWORD rc - Win32 error code (SUCCESS - ERROR_SUCCESS, ERROR - everything else)
 */
DWORD GetClusterSize(LPCWSTR pszVolume, LPDWORD pClusterSize, LPDWORD pSectorsPerCluster, LPDWORD pBytesPerSector)
{
    DWORD rc = ERROR_SUCCESS;

    if((pClusterSize != NULL) && (pSectorsPerCluster != NULL) && (pBytesPerSector != NULL))
    {
        DWORD FreeClusters, TotalClusters;
        BOOL  bRet;

        // Use GetDiskFreeSpace to determine the cluster size.
        //
        // Do not trust the FreeClusters and TotalClusters since they
        // cannot go above 2GB in size
        //
        bRet = GetDiskFreeSpace(pszVolume, pSectorsPerCluster, pBytesPerSector, &FreeClusters,&TotalClusters);

        if(bRet == 0)
        {
            // something obviously went wrong with GetDiskFreeSpace
            rc = GetLastError();
        }
        else
```

```
        {
            // if the actual cluster size is requested, deliver it now
            if(pClusterSize != NULL)
            {
                // ClusterSize is going to be fairly small so we do not need to worry about multiplication overflow
                *pClusterSize = *pSectorsPerCluster * *pBytesPerSector;
            }
        }
    }
    else
    {
        rc = ERROR_INVALID_PARAMETER;
    }

    return(rc);
}
```