# 这一节我们来学习磁盘驱动器属性，

# 学习大纲如下

|  |
| --- |
|  |

## GetDriveType函数的语法如下

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 语法 UINT GetDriveTypeA(  [in, optional] LPCSTR lpRootPathName  ); 参数 [in, optional] lpRootPathName  驱动器的根目录。  尾部反斜杠是必需的。如果此参数为 **NULL，**则函数使用 root 的目录。 返回值 返回值指定驱动器的类型，可以是以下值之一。  展开表   |  |  | | --- | --- | | **返回代码/值** | **描述** | | **DRIVE\_UNKNOWN**  0 | 无法确定驱动器类型。 | | **DRIVE\_NO\_ROOT\_DIR**  1 | 根路径无效;例如，在指定路径上没有挂载的卷。 | | **DRIVE\_REMOVABLE**  2 | 驱动器具有可移动媒体;例如，软盘驱动器、拇指驱动器或闪存卡读卡器。 | | **DRIVE\_FIXED**  3 | 驱动器具有固定介质;例如，硬盘驱动器或闪存驱动器。 | | **DRIVE\_REMOTE**  4 | 驱动器是远程 （网络） 驱动器。 | | **DRIVE\_CDROM**  5 | 驱动器是 CD-ROM 驱动器。 | | **DRIVE\_RAMDISK**  6 | 驱动器是 RAM 磁盘。 |  言论 |

## 这里的API：GetVolumeInformation就可以用这个字符串得到相关信息。

## 下面就是这个函数的语法

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 语法 BOOL GetVolumeInformationA(  [in, optional] LPCSTR lpRootPathName,  [out, optional] LPSTR lpVolumeNameBuffer,  [in] DWORD nVolumeNameSize,  [out, optional] LPDWORD lpVolumeSerialNumber,  [out, optional] LPDWORD lpMaximumComponentLength,  [out, optional] LPDWORD lpFileSystemFlags,  [out, optional] LPSTR lpFileSystemNameBuffer,  [in] DWORD nFileSystemNameSize  ); 参数 [in, optional] lpRootPathName  指向包含要描述的卷的根目录的字符串的指针。  如果该参数为 **NULL，**则使用当前目录的根目录。尾部反斜杠是必需的。例如，将 \\MyServer\MyShare 指定为 “\\MyServer\MyShare\”，或将 C 盘指定为 “C：\”。  [out, optional] lpVolumeNameBuffer  指向接收指定卷名称的缓冲区的指针。缓冲区大小由 nVolumeNameSize 参数指定。  [in] nVolumeNameSize  卷名称缓冲区的长度，以 **TCHAR** 为单位。最大缓冲区大小为 **MAX\_PATH**+1。  如果未提供卷名称缓冲区，则忽略此参数。  [out, optional] lpVolumeSerialNumber  指向接收卷序列号的变量的指针。  如果不需要序列号，则此参数可以为 **NULL**。  此函数返回操作系统在格式化硬盘时分配的卷序列号。若要以编程方式获取制造商分配的硬盘序列号，请使用 Windows Management Instrumentation （WMI） [Win32\_PhysicalMedia](https://learn.microsoft.com/en-us/previous-versions/windows/desktop/cimwin32a/win32-physicalmedia)属性 **SerialNumber**。  [out, optional] lpMaximumComponentLength (文件名的最大长度)  一个指向变量的指针，该变量接收指定文件系统支持的文件名组件的最大长度（以 **TCHAR** 为单位）。  文件名组件是文件名中介于反斜杠之间的部分。  存储在 \*lpMaximumComponentLength 指向的变量中的值用于指示指定的文件系统支持长名称。例如，对于支持长名称的 FAT 文件系统，该函数存储值 255，而不是以前的 8.3 指示符。使用 NTFS 文件系统的系统也可以支持长名称。  [out, optional] lpFileSystemFlags  指向一个变量的指针，该变量接收与指定文件系统关联的标志。  此参数可以是以下一个或多个标志。但是，**FILE\_FILE\_COMPRESSION** 和 **FILE\_VOL\_IS\_COMPRESSED** 是互斥的。  展开表   | **价值** | **意义** | | --- | --- | | **FILE\_CASE\_SENSITIVE\_SEARCH** 0x00000001 | 指定的卷支持区分大小写的文件名。 | | **FILE\_CASE\_PRESERVED\_NAMES** 0x00000002 | 指定的卷在将名称放在磁盘上时支持文件名的保留大小写。 | | **FILE\_UNICODE\_ON\_DISK** 0x00000004 | 指定的卷在文件名中支持 Unicode，因为它们在磁盘上显示的名称。 | | **FILE\_PERSISTENT\_ACLS** 0x00000008 | 指定的卷保留并强制实施访问控制列表 （ACL）。例如，NTFS 文件系统保留并强制实施 ACL，而 FAT 文件系统则不保留。 | | **FILE\_FILE\_COMPRESSION** 0x00000010 | 指定的卷支持基于文件的压缩。 | | **FILE\_VOLUME\_QUOTAS** 0x00000020 | 指定的卷支持磁盘配额。 | | **FILE\_SUPPORTS\_SPARSE\_FILES** 0x00000040 | 指定的卷支持稀疏文件。 | | **FILE\_SUPPORTS\_REPARSE\_POINTS** 0x00000080 | 指定的卷支持重新分析点。  **ReFS：**ReFS 支持重新分析点，但不为它们编制索引，因此 [FindFirstVolumeMountPoint](https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-findfirstvolumemountpointa) 和 [FindNextVolumeMountPoint](https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-findnextvolumemountpointa) 将无法按预期运行。 | | **FILE\_SUPPORTS\_REMOTE\_STORAGE** 0x00000100 | 文件系统支持远程存储。 | | **FILE\_RETURNS\_CLEANUP\_RESULT\_INFO** 0x00000200 | 成功执行清理操作后，文件系统将返回描述清理期间执行的其他操作（例如删除文件）的信息。文件系统筛选器可以在其清理后回调中检查此信息。 | | **FILE\_SUPPORTS\_POSIX\_UNLINK\_RENAME** 0x00000400 | 文件系统支持 POSIX 样式的删除和重命名操作。 | | **FILE\_VOLUME\_IS\_COMPRESSED** 0x00008000 | 指定的卷是压缩卷，例如 DoubleSpace 卷。 | | **FILE\_SUPPORTS\_OBJECT\_IDS** 0x00010000 | 指定的卷支持对象标识符。 | | **FILE\_SUPPORTS\_ENCRYPTION** 0x00020000 | 指定的卷支持加密文件系统 （EFS）。有关更多信息，请参阅[文件加密](https://learn.microsoft.com/en-us/windows/win32/FileIO/file-encryption)。 | | **FILE\_NAMED\_STREAMS** 0x00040000 | 指定的卷支持命名流。 | | **FILE\_READ\_ONLY\_VOLUME** 0x00080000 | 指定的卷是只读的。 | | **FILE\_SEQUENTIAL\_WRITE\_ONCE** 0x00100000 | 指定的卷支持单个顺序写入。 | | **FILE\_SUPPORTS\_TRANSACTIONS** 0x00200000 | 指定的卷支持事务。有关更多信息，请参阅[关于 KTM](https://learn.microsoft.com/en-us/windows/win32/Ktm/about-ktm)。 | | **FILE\_SUPPORTS\_HARD\_LINKS** 0x00400000 | 指定的卷支持硬链接。有关更多信息，请参阅[硬链接和交汇点](https://learn.microsoft.com/en-us/windows/win32/FileIO/hard-links-and-junctions)。  **Windows Server 2008、Windows Vista、Windows Server 2003 和 Windows XP：**在 Windows Server 2008 R2 和 Windows 7 之前，不支持此值。 | | **FILE\_SUPPORTS\_EXTENDED\_ATTRIBUTES** 0x00800000 | 指定的卷支持扩展属性。扩展属性是一段特定于应用程序的元数据，应用程序可以将其与文件相关联，而不是文件数据的一部分。  **Windows Server 2008、Windows Vista、Windows Server 2003 和 Windows XP：**在 Windows Server 2008 R2 和 Windows 7 之前，不支持此值。 | | **FILE\_SUPPORTS\_OPEN\_BY\_FILE\_ID** 0x01000000 | 文件系统支持按 FileID 打开。有关更多信息，请参阅 [FILE\_ID\_BOTH\_DIR\_INFO](https://learn.microsoft.com/en-us/windows/win32/api/winbase/ns-winbase-file_id_both_dir_info)。  **Windows Server 2008、Windows Vista、Windows Server 2003 和 Windows XP：**在 Windows Server 2008 R2 和 Windows 7 之前，不支持此值。 | | **FILE\_SUPPORTS\_USN\_JOURNAL** 0x02000000 | 指定的卷支持更新序列号 （USN） 日志。有关更多信息，请参阅[更改日记记录](https://learn.microsoft.com/en-us/windows/win32/FileIO/change-journal-records)。  **Windows Server 2008、Windows Vista、Windows Server 2003 和 Windows XP：**在 Windows Server 2008 R2 和 Windows 7 之前，不支持此值。 | | **FILE\_SUPPORTS\_INTEGRITY\_STREAMS** 0x04000000 | 文件系统支持[完整性流](https://learn.microsoft.com/en-us/windows-server/storage/refs/integrity-streams)。 | | **FILE\_SUPPORTS\_BLOCK\_REFCOUNTING** 0x08000000 | 指定的卷支持在同一卷上的文件之间共享逻辑集群。文件系统在写入共享集群时重新分配。指示 **FSCTL\_DUPLICATE\_EXTENTS\_TO\_FILE** 是支持的操作。 | | **FILE\_SUPPORTS\_SPARSE\_VDL** 0x10000000 | 文件系统跟踪文件的每个簇是包含有效数据（来自显式文件写入或自动归零）还是无效数据（尚未写入或归零）。使用稀疏有效数据长度 （VDL） 的文件系统不存储有效数据长度，也不要求有效数据在文件中是连续的。 | | **FILE\_DAX\_VOLUME** 0x20000000 | 指定的卷是直接访问 （DAX） 卷。  **注意：** 此标志是在 Windows 10 版本 1607 中引入的。 | | **FILE\_SUPPORTS\_GHOSTING** 0x40000000 | 文件系统支持重影。 |   [out, optional] lpFileSystemNameBuffer  指向接收文件系统名称的缓冲区的指针，例如 FAT 文件系统或 NTFS 文件系统。缓冲区大小由 nFileSystemNameSize 参数指定。  [in] nFileSystemNameSize  文件系统名称缓冲区的长度，以 **TCHAR** 为单位。最大缓冲区大小为 **MAX\_PATH**+1。  如果未提供文件系统名称缓冲区，则忽略此参数。 返回值 如果检索到所有请求的信息，则返回值为非零。  如果未检索到所有请求的信息，则返回值为零。要获取扩展错误信息，请调用 [GetLastError](https://learn.microsoft.com/en-us/windows/win32/api/errhandlingapi/nf-errhandlingapi-getlasterror)。 言论 当用户尝试获取有关没有软盘的软盘驱动器或没有光盘的 CD-ROM 驱动器的信息时，系统会显示一个消息框，供用户分别插入软盘或光盘。若要防止系统显示此消息框，请使用 **SEM\_FAILCRITICALERRORS**调用 [SetErrorMode](https://learn.microsoft.com/en-us/windows/win32/api/errhandlingapi/nf-errhandlingapi-seterrormode) 函数。  **FILE\_VOL\_IS\_COMPRESSED** 标志是基于卷的压缩的唯一指示器。文件系统名称不会更改以指示压缩，例如，在 DoubleSpace 卷上返回设置此标志。当压缩是基于卷的时，整个卷将被压缩或不压缩。  **FILE\_FILE\_COMPRESSION**标志指示文件系统是否支持基于文件的压缩。当压缩是基于文件时，可以压缩或不压缩单个文件。  **FILE\_FILE\_COMPRESSION** 和 **FILE\_VOL\_IS\_COMPRESSED** 标志是互斥的。不能返回两个 bit set。  存储在 lpMaximumComponentLength 中的最大组件长度值是卷支持比正常 FAT 文件系统 （或其他文件系统） 文件名更长的唯一指示器。文件系统名称不会更改以表示支持长文件名。  [GetCompressedFileSize](https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-getcompressedfilesizea) 函数获取文件的压缩大小。[GetFileAttributes](https://learn.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-getfileattributesa) 函数可以确定是否压缩了单个文件。 |

### 可以把前面学习过的GetLogicalDriveStrings获取到的结果作为该函数的一个参数，用来获取磁盘信息

# 演练

## 1.新建一个常规的空项目，叫做：Lesson13-drive-info，然后添加一个cpp文件，取名：drive-info.cpp,添加初始内容如下

|  |
| --- |
| #include<Windows.h>  #include<stdio.h>  #include<stdlib.h>  #define BUFSIZE 1024  BOOL DisplayDriveInfo(LPSTR szDrive)  {  UINT uDriverType;  DWORD dwVolumeSerialNumber;  DWORD dwMaximumComponentlength;  DWORD dwFileSystemFlags;  CHAR szFileSystemNameBuffer[BUFSIZE];  CHAR szDriveName[MAX\_PATH];  //显示盘符  printf("\n%s\n",szDrive);  //1.判断磁盘类型  uDriverType=GetDriveType(szDrive);  switch(uDriverType)  {  case DRIVE\_UNKNOWN:  printf("Unknown drive.\n");  break;  case DRIVE\_NO\_ROOT\_DIR:  printf("The root directory does not exist.\n");  break;  case DRIVE\_REMOVABLE:  printf("Removable Disk Drive.\n");  break;  case DRIVE\_FIXED:  printf("Hard Drive.\n");  break;  case DRIVE\_REMOTE:  printf("Remote Drive.\n");  break;  case DRIVE\_CDROM:  printf("CDRom drive.\n");  break;  case DRIVE\_RAMDISK:  printf("Ram Disk Drive\n");  break;  }  //2.获取逻辑驱动器的信息  if(!GetVolumeInformation(  szDrive,  szDriveName,  MAX\_PATH,  &dwVolumeSerialNumber,  &dwMaximumComponentlength,  &dwFileSystemFlags,  szFileSystemNameBuffer,  BUFSIZ))  {  return FALSE;  }  //3.显示信息  if(0!=lstrlen(szDriveName))  {  printf("\nDrive Name is %s.\n",szDriveName);  }  printf("\nVolume Serial is %u.",dwVolumeSerialNumber );  printf("\nMaximum Component Length is %u.",dwMaximumComponentlength);  printf("\nSystem Type is %s.\n",szFileSystemNameBuffer);  if (dwFileSystemFlags & FILE\_VOLUME\_QUOTAS)  {  printf("The file system supports disk Quotas.\n");  }  if (dwFileSystemFlags & FILE\_SUPPORTS\_REPARSE\_POINTS)  {  printf("The file system does not support volume mount points.\n");  }  if (dwFileSystemFlags & FILE\_CASE\_SENSITIVE\_SEARCH)  {  printf("The file system supports case-sentitive file name.\n");  }  printf("...\n");  return TRUE;  }  int main()  {  CHAR szLogicalStr[BUFSIZE];  PCHAR szDrive;  ZeroMemory(szLogicalStr,BUFSIZE);  GetLogicalDriveStrings(BUFSIZE-1,szLogicalStr);  szDrive = (PCHAR)szLogicalStr;  do  {  if(!DisplayDriveInfo(szDrive))  {  printf("Get Drive Info Failed...");  }  szDrive +=(lstrlen(szDrive)+1);  } while(lstrlen(szDrive)!='\x00');  } |

### 效果：

|  |
| --- |
| C:\  Hard Drive.  Volume Serial is 3363692264.  Maximum Component Length is 255.  System Type is NTFS.  The file system supports disk Quotas.  The file system does not support volume mount points.  The file system supports case-sentitive file name.  ...  D:\  Hard Drive.  Drive Name is 数据盘.  Volume Serial is 1550604159.  Maximum Component Length is 255.  System Type is NTFS.  The file system supports disk Quotas.  The file system does not support volume mount points.  The file system supports case-sentitive file name.  ...  E:\  Hard Drive.  Drive Name is 新加卷.  Volume Serial is 542590885.  Maximum Component Length is 255.  System Type is NTFS.  The file system supports disk Quotas.  The file system does not support volume mount points.  The file system supports case-sentitive file name.  ...  F:\  Hard Drive.  Drive Name is Document and software.  Volume Serial is 3639735162.  Maximum Component Length is 255.  System Type is NTFS.  The file system supports disk Quotas.  The file system does not support volume mount points.  The file system supports case-sentitive file name.  ...  G:\  Hard Drive.  Drive Name is learning.  Volume Serial is 3629183472.  Maximum Component Length is 255.  System Type is NTFS.  The file system supports disk Quotas.  The file system does not support volume mount points.  The file system supports case-sentitive file name.  ... |