本文档介绍MyEverything的使用。

程序是字符界面的，因为仅仅是为了学习。程序实现了如下功能：

* 能够在很短的时间内扫描你指定分区上的所有文件并把文件信息存储在内存数据库中。
* 根据文件名搜索文件，速度很快。
* 可以监视一个已经扫描过的分区，该分区上所有文件的创建、删除、改名操作都会实时反映到搜索结果中。

Everything软件的实现技术用到了NTFS 5.0中加入的Change Journal特性，操作该特性的win32 api（DeviceIoControl）为我们提供了两大功能：

* 快速遍历分区上的文件和文件夹
* 监视分区上文件和文件夹的变化

于是便有了像Everything这样只搜索名字但是速度奇快的硬盘搜索软件。

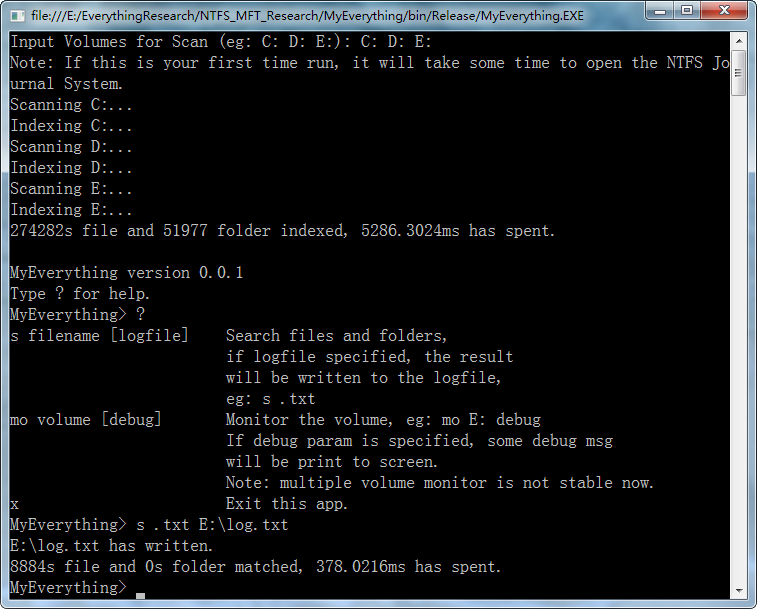


图1

图1展示的是最基本的使用方法：输入你想扫描的分区，图中输入的是**C: D: E:**，注意，第一次运行本程序可能会花点时间，因为NTFS的JournalSystem默认是不打开的，打开需要一定的时间，第二次运行程序由于 NTFS Journal已经被打开，时间将会大幅度缩短。

输入？可以打开帮助，有三条命令可以使用，分别是搜索、监视和退出程序。输入s filename可以搜索文件名中带filename字样的文件和文件夹，当然前提是这些文件存在于之前扫描过的分区上，尚不支持多条件搜索，即输入的文件名不能带有空格。结果将会打印在屏幕上，如果预料到输出的结果会很多，可以加上一个输出文件名参数，这样结果会写入输出文件而不打印到屏幕上，例如：**s filename E:\log.txt**

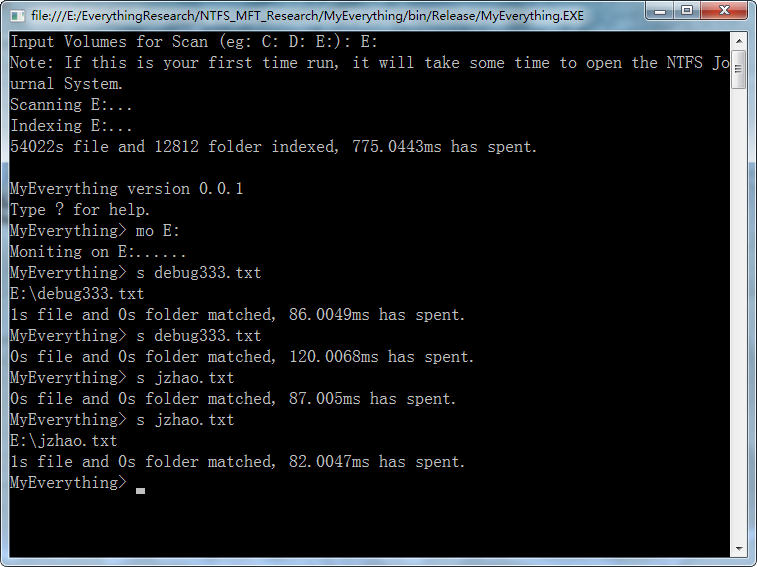


图2

图2展示的是监视分区的功能，监视某个分区意味着该分区上所有文件的变化都会被程序感知，MyEverything需要处理的文件变化类型是文件的新建、 删除和重命名，因为这些操作会影响到搜索结果。截图中首先输入**mo E:**，表示监视E:分区，然后搜索文件debug333.txt，发现在E:根目录存在该文件。然后我删除了E:\debug333.txt，再次搜索，发现文件搜索不到了，换个文件名搜索jzhao.txt，发现文件不存在，在E:根目录创建文件jzhao.txt后再次搜索，文件被搜索出，这就是监视的效果。

图3展示的是在监视分区过程中输出debug信息的功能，只要为mo命令添加第二个参数debug即可，例如：**mo E: debug**

输入监视命令后，我在E:根目录创建文件“实验.txt”，屏幕上立即输出了3条记录，分别是“New file E:\实验.txt added.”、“File E:\实验.txt deleted”和“New file E:\实验.txt added.”，最后反映到内存数据库的结果是文件被创建了。

之后我将“实验.txt”删除至回收站，于是我们看到了一条重命名记录，仔细观察会发现这其实是将文件从E:根目录移动到了回收站中。然后我在桌面上单击鼠标右键选择“撤销 删除”，屏幕上又出现了一条重命名记录，仔细观察，其实是文件从回收站中移回了E:根目录。最后我对文件实施shift + delete，这次只出现了“File E:\实验.txt deleted”一条记录。如果你通过源代码建立工程并以debug模式运行程序，可以在ide的debug窗口中看到更多更详细的记录，这些记录反应的是操作系统对NTFS格式分区底层的实际操作。

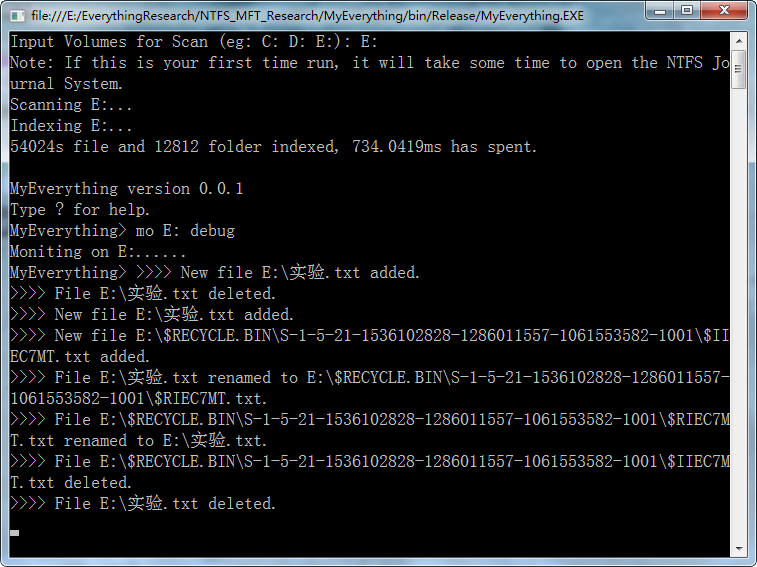


图3

程序中没有的功能：

* 程序每次打开都要重新扫描磁盘，因为程序仅用来做演示，程序关闭后没有保存上次扫描后的结果。
* 尚不知对文件夹的新建、重命名、删除操作的监视效果如何。
* 和Everything一样，本程序不能搜索文件的内容。