Atitit 文件搜索的方案

目录

[1.1. 实时遍历搜索区的所有文件 1](#_Toc30021)

[1.2. 读取USN日志 1](#_Toc4982)

[1.3. Everything 语法参数 2](#_Toc14823)

[1.4. 桌面搜索，是通过建立和维护一个索引数据库来实现的 6](#_Toc12667)

## 实时遍历搜索区的所有文件

XP非常的古老，是十四年前的系统。其搜索方式是当你搜索的时候，实时遍历搜索区的所有文件，所以搜索速度极其缓慢。

## 读取USN日志

不过everything这样的软件也存在非常明显的缺点，比如读取USN日志显然需要管理权限，这就存在安全问题，搜索也仅支持文件名或路径名，不能支持其他文件属性如创建时间，另外还只支持NTFS一种文件格式。

事实上，现代操作系统中的搜索通常并不采用这种方式。

Everything并没有全部逐一扫描我们硬盘上的文件，而是通过读取NTFS文件系统中的USN日志来完成的，只能检索文件名。  
　　USN是系统日志的一部分，是Update Service Number Journal or Change Journal的英文缩写，直译为“更新序列号”，是对NTFS卷里所修改过的信息进行相关记录的功能。当年微软发布Windows 2000时，建立NTFS 5.0的同时，加入了一些新功能和改进了旧版本的文件系统，，它可以在分区中设置监视更改的文件和目录的数量，记录下监视对象修改时间和修改内容。当这个功能启用时，对于每一个NTFS卷，当发生有关添加、删除和修改文件的信息时，NTFS都使用USN日志记录下来。  
　　USN日志的工作方式，相对来说很简单，所以非常的高效。它开始的时候是一个空文件，包括NTFS每个卷的信息。每当NTFS卷有改变的时候，所改变的信息会马上被添加到这个文件里。这其中，每条修改的记录都使用特定符号来标识为日志形式，也就是USN日志。每条日志，记录了包括文件名、文件信息做出的改变，日志里包括发生了什么变化（添加、删除或其他操作）。  
　　USN日志相当于一本书的索引，当然书里面内容发生添加、修改或删除的时候，USN日志会记录下来何时做了修改，并使用特定序列号来标识，但它并不会记录里面具体修改了什么东西，所以索引文件很小。而当你想查找某一篇文章时，你就不用一页一页去翻书，可以直接通过查找USN日志（也就是建立的索引）就知道这篇文章是否存在。  
综上： NTFS文件系统中的USN日志，是一项系统管理功能，能够记录卷上文件和文件夹的所有更改。“Everything”的搜索功能也是基于这个日志，只是在索引当中根据文件名过滤出符合条件的文件或文件夹故而十分迅速。随着版本的优化，其速度也到了目前秒开的水平，确实是Windows的文件名检索利器。

· **Everything 可在软件设置选项中，开启 http 服务器功能，他可以让你在手机上访问电脑的文件，被窝看片不是梦（虽然现在wifi和无限流量套餐都蛮普及的）。**

· **Everything 不仅可以搜索文件名，现在还可以搜索文档中的内容了。只需要在搜索时加上 content:文档内容，即可。**

## Everything 语法参数

操作符:

space 与 (AND)

| 或 (OR)

! 非 (NOT)

< > 分组

" " 搜索引号内的词组.

通配符:

\* 匹配 0 个或多个字符.

? 匹配 1 个字符.

\*. 匹配不包含扩展的文件名.

\*.\* 与 \* 相同

宏:

quot: 双引号 (")

apos: 单引号 (')

amp: 与号 (&)

lt: 小于 (<)

gt: 大于 (>)

#x: Unicode字符代码 x.

修饰符:

ascii: 启用快速 ASCII 大小写对比.

case: 区分大小写.

diacritics: 匹配变音标记.

file: 仅匹配文件.

folder: 仅匹配文件夹.

noascii: 禁用快速 ASCII 大小写对比.

nocase: 不区分大小写.

nodiacritics: 不匹配变音标记.

nofileonly: 仅不允许文件.

nofolderonly: 仅不允许文件夹.

nopath: 不匹配路径.

noregex: 禁用正则表达式.

nowfn: 不匹配完整文件名.

nowholefilename: 不匹配完整文件名.

nowholeword: 仅禁用全字匹配.

nowildcards: 禁用通配符.

noww: 仅禁用全字匹配.

path: 匹配路径和文件名.

regex: 启用正则表达式.

utf8: 禁用快速 ASCII 大小写对比.

wfn: 匹配完整文件名.

wholefilename: 匹配完整文件名.

wholeword: 仅匹配全字符.

wildcards: 启用通配符.

ww: 仅全字匹配.

函数:

attrib:<attributes> 搜索指定的文件属性的文件和文件夹.

attributes:<attributes> 搜索指定的文件属性的文件和文件夹.

child:<filename> 搜索包含匹配文件名文件的文件夹.

childcount:<count> 搜索包含有指定数目子文件夹或文件的文件夹.

childfilecount:<count> 搜索包含有指定数目文件的文件夹.

childfoldercount:<cnt> 搜索包含有指定数目子文件的文件夹.

dateaccessed:<date> 搜索指定访问时间的文件和文件夹.

datecreated:<date> 搜索指定创建日期的文件和文件夹.

datemodified:<date> 搜索指定修改日期的文件和文件夹.

daterun:<date> 搜索指定打开时间的文件和文件夹.

da:<date> 搜索指定访问时间的文件和文件夹.

dc:<date> 搜索指定创建日期的文件和文件夹.

dm:<date> 搜索指定修改日期的文件和文件夹.

dr:<date> 搜索指定打开时间的文件和文件夹.

dupe: 搜索重复的文件名.

empty: 搜索空文件夹.

endwith:<text> 搜索以指定文本结尾的文件 (包含扩展名).

ext:<list> 搜索和列表中指定的扩展名匹配的文件 (扩展名以分号分隔).

filelist:<fn1|fn2|...> 搜索文件名列表中的文件.

filelistfilename:<name> 搜索文件名列表中的文件和文件夹.

fsi:<index> 搜索指定盘符索引中文件或文件夹 (索引 0 表示 C 盘, 以此类推).

infolder:<path> 搜索指定路径下的文件和文件夹 (不包含子文件夹).

len:<length> 搜索和指定的文件名长度相匹配的文件和文件夹.

parent:<path> 搜索指定路径下的文件和文件夹 (不包含子文件夹).

parents:<count> 搜索有指定数目父文件夹的文件和文件夹.

rc:<date> 搜索指定最近修改日期的文件和文件夹.

recentchange:<date> 搜索指定最近修改日期的文件和文件夹.

root: 搜索没有父文件夹的文件和文件夹.

runcount:<count> 搜索指定打开次数的文件和文件夹.

size:<size> 搜索指定大小的文件 (以字节为单位).

sizedupe: 搜索大小重复的文件.

startwith:<text> 搜索指定文本开头的文件.

type:<type> 搜索指定的文件类型的文件和文件夹.

函数语法:

function:value 等于某设定值.

function:<=value 小于等于某设定值.

function:<value 小于某设定值.

function:=value 等于某设定值.

function:>value 大于某设定值.

function:>=value 大于等于某设定值.

function:start..end 在起始值和终止值的范围内.

function:start-end 在起始值和终止值的范围内.

大小语法:

size[kb|mb|gb]

大小常数:

empty

tiny 0 KB < 大小 <= 10 KB

small 10 KB < 大小 <= 100 KB

medium 100 KB < 大小 <= 1 MB

large 1 MB < 大小 <= 16 MB

huge 16 MB < 大小 <= 128 MB

gigantic 大小 > 128 MB

unknown

日期语法:

year

month/year 或者 year/month 取决于本地设置

day/month/year, month/day/year 或者 year/month/day 取决于本地设置

日期常数:

today

yesterday

tomorrow

<last|past|prev|current|this|coming|next><year|month|week>

<last|past|prev|coming|next><x><years|months|weeks|hours|minutes|mins|seconds|secs>

january|february|march|april|may|june|july|august|september|october|november|december

jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec

sunday|monday|tuesday|wednesday|thursday|friday|saturday

sun|mon|tue|wed|thu|fri|sat

unknown

属性常数:

A 存档

C 压缩

D 目录

E 加密

H 隐藏

I 未索引的内容

L 重解析点

N 一般

O 离线

P 稀疏文件

R 只读

S 系统

T 临时

V 设备

正则表达式语法:

a|b 匹配 a 或 b

gr(a|e)y 匹配 gray 或 grey

. 匹配任一字符

[abc] 匹配任一字符: a 或 b 或 c

[^abc] 匹配任一字符, 但不包括 a、b、c

[a-z] 匹配从 a 到 z 之间的任一字符

[a-zA-Z] 匹配从 a 到 z, 及从 A 到 Z 之间的任一字符

^ 匹配文件名的头部

$ 匹配文件名的尾部

\* 匹配前一项内容 0 或多次

? 匹配前一项内容 0 或 1 次

+ 匹配前一项内容 1 或多次

{x} 匹配前一项内容 x 次

{x,} 匹配前一项内容 x 或多次

{x,y} 匹配前一项内容次数介于 x 和 y 之间

## 桌面搜索，是通过建立和维护一个索引数据库来实现的

从VISTA开始，windows引入的全新的桌面搜索，是通过建立和维护一个索引数据库来实现的。这也是目前搜索GB以上数据时要达到合理性能的唯一方法。软件在初次运行时会花费时间来遍历所有文件生成一个数据库，之后只需要实时维护数据库即可。优点在于不仅可以提供文件名搜索，还可以搜索元数据（如标题，作者，注释等），甚至文件内容

作者：Jeff  
链接：https://www.zhihu.com/question/30038398/answer/46580084  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。