DOCUMENTS 4

DOCUMENTS
IFilterReader

Table of Contents

1. Introduction....4

1.1 Installation requirements 4
1.2 Installed filters....5
1.3 Filter tools from Windows SDK 6
1.4 Standard Windows filters 6
1.5 Other filter packs available from Microsoft:....6
1.6 Filter packs available from Adobe:....7
1.7 Document properties 7

2. IFilterReader....9

3. Table of Figures....10

1. Introduction

A variety of Windows products allow formulating search queries that extend beyond searching the file system only for the occurrence of a specific file. In fact, search can also be performed within the files themselves, where a currently held search index enables quick retrieval of the searched-for file.

So, from Windows 2000 the "*Windows Indexing Service*" as a basic service helps responding to search queries in which not only file *properties* such as the time when the file was created, the time the file was last edited and the file name are searched for, but in which the content of the files itself can be searched (*full text search*).

In doing so, the text or a textual representation of the file content is determined and indexed while indexing, depending on the file type.

The filter component through which the text is filtered from the file is dependent on the respective file type. Thus, for instance, a "*Plain Text Filter*" for reading "`*.txt`" files is available which can also be used for "`*.bat`" batch files. The responsible filter or *IFilter* is stored in the registry; it is then determined for the respective file type on indexing via a defined mechanism.

Filters for basic formats such as "`*.txt`" are installed as part of the operating system. Special filters for Microsoft formats such as "`*.doc`" or "`*.xls`" are installed along with the Microsoft Office installation; however, they can also be obtained separately from Microsoft.

Frequently the manufacturer of a specific software product also installs the suitable filter representing implementation of the *IFilter* interface specified by Microsoft. This also allows capturing their documents by the indexing service.

IFilter components are used in the Indexing Service, in SharePoint Portal Server, Exchange, SQL Server and Windows Desktop Search as well as all other products using Windows search; they are integrated into Vista and Windows7.

1.1 Installation requirements

IFilterReader requires completed installation of "*Windows Desktop Search*" and its activation or, optionally, activation of the Windows search service or indexing service, because these activate the use of *IFilter* system-wide. Depending on the operating system, these services only need to be activated because they have already been installed with the operating system.

The directories in which Windows performs indexing can then be defined in such a manner that only minimum system load is performed by the search service, e.g. by only indexing the start menu. The server load will then be insignificant on running the search service.

For Windows Server 2008, MSDN provides the corresponding tutorial under the keyword "*Windows Search Service*".

http://support.microsoft.com/kb/947036/en-us

For installing *TIFF IFilters* on Windows Server 2008 R2, installation notes are available under the keyword "*Windows TIFF IFilter Overview*" and the URL:

http://technet.microsoft.com/en-us/library/dd834685.aspx

1.2 Installed filters

An overview of the installed filters can be found under "*Control Panel-> Indexing Options->Advanced->File Types*":

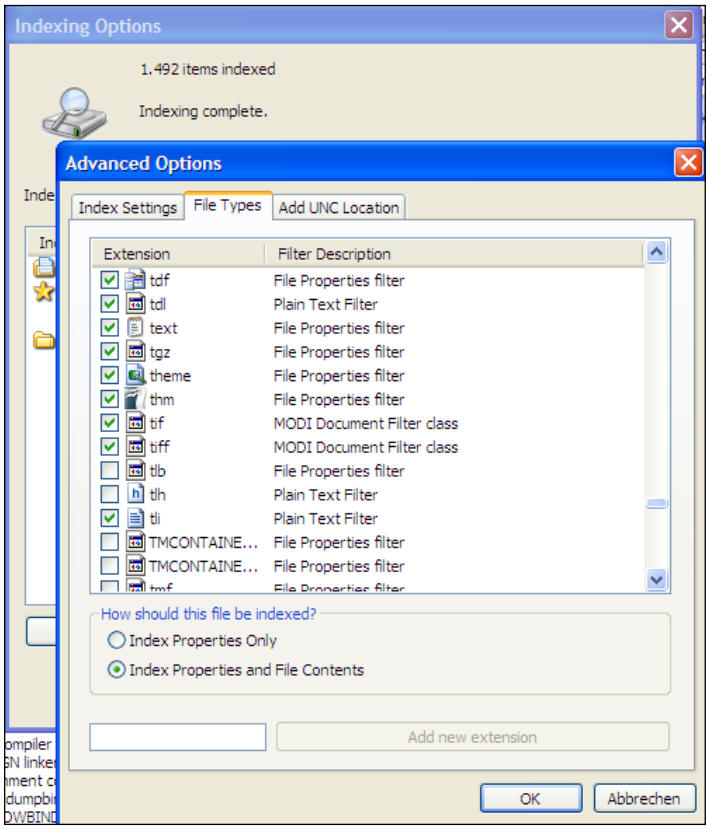

Fig. 1: Registered filters for various file types

The `filtreg.exe` tool (see below) from the Windows SDK provides a more detailed overview.

1.3 Filter tools from Windows SDK

Windows SDK provides a number of tools that are used to determine and test the existing filters. The MSDN article entitled "*Testing Filter Handlers*" provides an overview of this:

http://msdn.microsoft.com/en-us/library/dd940434(VS.85).aspxs

- `filtdump.exe`: Shows the output of the filter created for a specific document.
- `filtreg.exe`: Provides an overview of all *IFilters* installed on the system in addition to the `dll` in which these were implemented.
- `ifilttst.exe`: Tool to test filters with detailed login information.

1.4 Standard Windows filters

An overview of the filters installed along with Windows Search can be found in MSDN under the keyword "*Filter Handlers that Ship with Windows*"

http://msdn.microsoft.com/en-us/library/dd940431(VS.85).aspx

1.5 Other filter packs available from Microsoft:

Microsoft provides other, separately available filter packs that extend the range of files from which information can be filtered. You need to consider the respectively listed installation requirements and restrictions for installation:

- 2007 Office System Converter: Microsoft Filter Pack: http://www.microsoft.com/downloads/details.aspx?FamilyId=60C92A37-719C-4077-B5C6-CAC34F4227CC&displaylang=en
- Visio IFilter 2003 Add-In: Text Search in Visio Files: http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=dcee9e09-448b-4386-b901-efea29cac80
- Visio IFilter 2002: http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=0d585a21-dd90-447f-b145-ded2bc21cb5c
- Windows Desktop Search: Add-in for Outlook saved mail (.msg file) indexing: http://www.microsoft.com/downloads/details.aspx?familyid=134ECBB0-C162-4D07-BEF3-0B602C4A79DD&displaylang=en
- Windows Desktop Search: Add-in for Lotus Notes: http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=ac768e36-be57-4306-966c-5089b0c4d50e
- Windows TIFF IFilter Installation and Operations Guide: http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=d220d961-5130-4279-b913-28b5f4be7a57
- Office 2003: Microsoft Office Document Imaging Visual Basic Reference (MODI):

http://www.microsoft.com/downloads/details.aspx?familyid=8F93E445-B1CF-4477-A373-E17417D616BC&displaylang=en

1.6 Filter packs available from Adobe:

- Adobe PDF IFilter 9 for 64-bit platforms: http://www.adobe.com/support/downloads/detail.jsp?ftpID=4025
- Adobe PDF IFilter v6.0: http://www.adobe.com/support/downloads/detail.jsp?ftpID=2611

1.7 Document properties

Besides the text, *IFilter* can also collect numerous document properties and make them available to the search engine. Which properties these exactly are depends on the respective implementation. An overview can be found under "*Schema - The schema documents the values and properties that the index uses to store data for indexing or sorting*":

http://msdn.microsoft.com/en-us/library/aa965725(VS.85).aspx

Name	Description
DocComments	Comments
Create	Date and time the file was created
DisplayFolder	User-friendly folder for item
DocAuthor	Author of the document
DocCategory	Category
DocKeywords	Keywords
DocTitlePrefix	Prefix of subject (Re:, Fw:, etc.)
DocTitle	Title
FileExtDesc	User-friendly description of the file type from the registry
FolderName	Name of the parent folder
IsAttachment	Indicates item is an attachment
IsDeleted	Indicates item is marked for deletion (Recycle bin, deleted items, etc.)
LastViewed	Date the item was last viewed by user
People	People involved with this item
PerceivedType	PerceivedType of the object NOTE: This is only for retrieval.
PerceivedTypeName	Display Name of the PerceivedType.

`PrimaryDate`	Most interesting date (Last write time for files, date received for e-mail)
`Size`	Size of a file.
…	…

Fig. 2: Excerpt from available document properties

2. IFilterReader

IFilterReader determines the responsible registered *IFilter* for a specific data file with a defined file extension, opens it and outputs the information returned by the filter in `UTF-8` string format.

The `IFilterReade.exe` file is run via the command line. Outputs come via the console's default output. The output can also be displayed in a *MessageBox* and be performed in a data file on request. Other options also allow defining a timeout and restricting maximum memory consumption.

```
IFilterReader.exe /dump:[show|file] /help /timeout:n Integer /file:path
/ext:fileextension (.jpg)
Options:
/help   shows this message
/dump   dumps the text into  a file named 'filename.extension.dump' in the
same directory as the source file
/dump:file      dumps the text into  a file 'filename.extension.dump' in
the same directory
/dump:show      shows the first 1000 characters of the extracted text in a
messagebox
/dump:showfile  dumps and shows the text
/timeout:n      kills the process after n minutes
/maxBuffer:nnnn maximal allowed Text buffer. The value has to be at least
8192 bytes (8K) long and should be a multiple of 8K
/file:filename   full path and filename ('C:\\tmp\my
documents\theDocument')  use the quote (')if path or filename contains any
whitespace.
/ext:abc        the file extension (.jpg, exe). This option will override
any filextension given in the file option.
/metadata       shows the metadata of the file, i.e. DocumentTitle, author,
file, size
```

Fig. 3: Command line call and options

Command line call and options

Output is in `UTF-8` format, which is why the console must be set to `Codepage 65001`:

```
chcp 65001
```

Next, you need to select a font that supports the corresponding representation of characters (e.g. Arial Unicode MS).

3. Table of Figures

Fig. 1: Registered filters for various file types 5
Fig. 2: Excerpt from available document properties 8
Fig. 3: Command line call and options 9