

Domino Full Text Indexing (FTI) Fundamentals

Contents :

Introduction

Chapter 1) What is a Full Text Index and why have one?

Chapter 2) Creating the Full Text Index.

Chapter 3) Maintaining the Full Text Index.

Chapter 4) Using the Full Text Index.

Chapter 5) Things to be aware of with a Full Text Index.

Chapter 6) Troubleshooting and debug options with a Full Text Index.

Chapter 7) Common Issues with Full Text Indexing and Search

Introduction

The aim of this article is

To explain what a Full Text Index is.

To show why you would want to use them.

To show how to use them.

To work through some troubleshooting options and solutions to common issues.

Glossary :

FTI Abbreviation of Full Text Index

.FT folder The Full Text Index is created in a folder with the same root name and location as the database it is created but has the .FT extension. When referring to the ".FT folder" this folder is being discussed.

Search Term The string or piece of text being searched for.

Chapter 1 - What is a Full Text Index and why have one?

A Full Text Index is a text index or list of the words in a database. It contains the text within the database and within attachments, if selected, and which documents contain that text.

It gives more accurate and faster results than a simple search, and lists the search results in order of relevance.

It allows for more complicated searches, like searching in specific fields or forms and for boolean options like And, Or, Not, etc. More on these options later.

The index itself is a number of files external to the database. It is generated in a sub-folder to the database with the database's root filename but the extension ".FT".

Generally the files within the .FT folder are not accessed directly and should not be directly edited.

Further information on the files that compose the Full Text Index can be found here :

Title : Overview of Full Text Index subdirectory and file layout

Doc # : 1220297

URL : <http://www.ibm.com/support/docview.wss?uid=swg21220297>

The FTI can be created for server based databases or locally, with a Notes Client, for databases stored on a hard drive and not accessed through a server.

Chapter 2 - Creating the Full Text Index.

By default a database does not have a Full Text Index so it needs to be created. There are several ways and options for doing this, these will be covered here.

Creating the FTI for a single database using a Notes Client

Creating the FTI for multiple databases using the Administrator Client

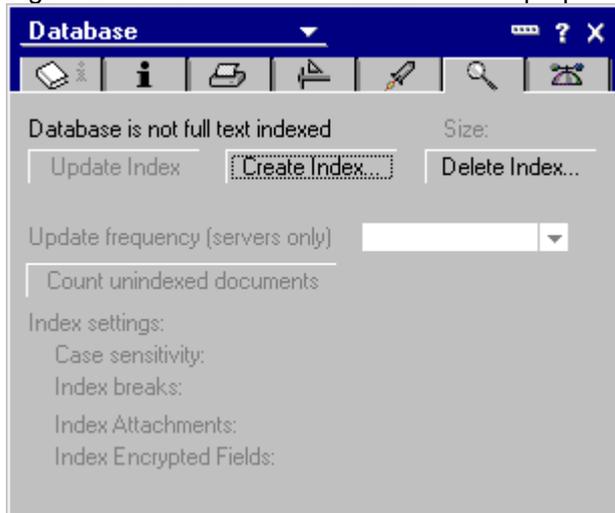
Creating the FTI for a single database using a Notes Client

From your Notes Client select or open the database that you want to create an FTI for.

Open the database properties, either on the workspace, by right clicking on its icon, or by the menu option "File\Application\Properties"

Select the 'Full Text Tab' (second from the far right) :

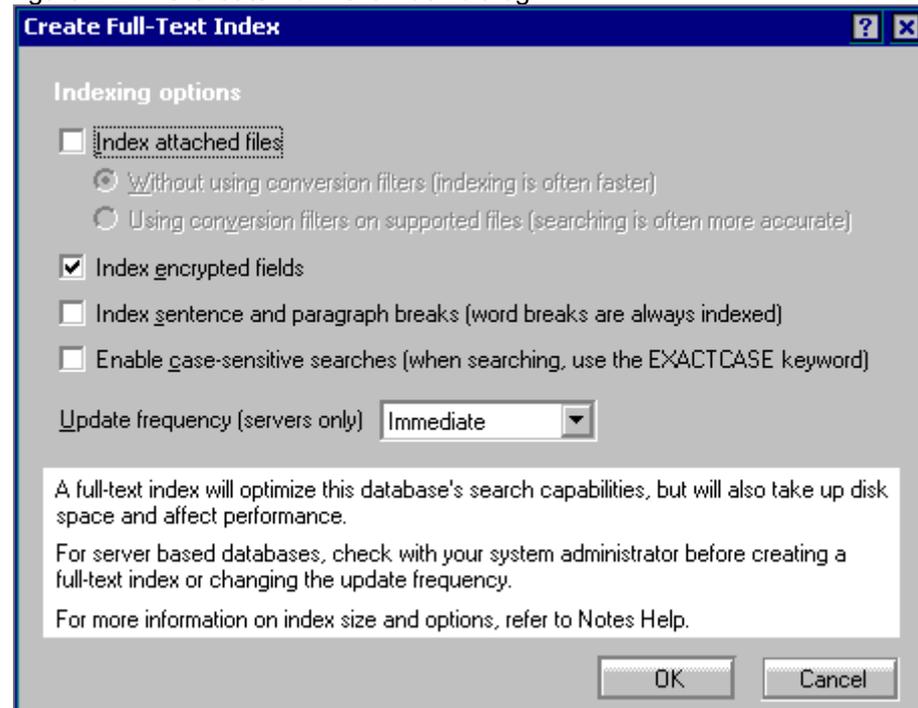
Figure 2-1. Full Text Index Tab in Database properties :



If the FTI does not exist many of the options will not be available. We will be looking at these later.

To create the FTI click on "Create Index..." to open the "Create Full-Text Index" dialog :

Figure 2-2 The Create Full-Text Index dialog :



Reviewing the options available :

Index attached files

This controls if text in attachments is included in the Full Text Index. This will make the index larger, but allows for more complete search of the text contained in the documents.

When active this gives two options :

Figure 2-3 : Index attached files options :



- ◆ "Without using conversion filters (Indexing is often faster)"
- ◆ "Using Conversion filters on supported files (searching is often more accurate)."

Conversion filters will attempt to parse only the text within the attachments that would normally be displayed by the attachment - so any programming or formatting data stored in the file will not be included in the index.

If the Conversion filters are not selected all text within the attachments, including all formatting or coding will be added to the index.

By not using the conversion filters the indexing process can be faster as the indexing task does not need to extract the text from the files. However, the resulting index is often larger and it can contain references to text that is not displayed as text from the attachment itself.

By using conversion filters the time and resources needed to create and maintain the index can increase, but the overall size of the index will be smaller as less text is added to it and searches will only find text that should be displayed in the attachment. Resulting searches can also be faster.

Several attachment types are not included in the FTI by default :

.au, .cca, .dbd, .dll, .exe, .gif, .img, .jpg, .mp3, .mpg, .mov, .nsf, .ntf, .p7m, .p7s, .pag, .sys, .tar, .tif, .wav, .wpl, .zip.

More on attachments further on

Index encrypted fields

This option sets if encrypted fields are included in the Index - provided the indexer (either the server ID or the User's ID if indexing a local database) has access to the text in the encrypted fields.

Index sentence and paragraph breaks (word breaks are always indexed)

This sets if these breaks will be included in the index when separating words.

Enable case-sensitive searches (when searching, use the EXACT CASE keyword)

Enables case sensitive searching. By default searches are not case sensitive.

Note : If case sensitive searches are required the index needs to be created with case sensitivity enabled. This will increase the size of the index.

Update frequency (servers only)

This selects how the index will be updated when it is hosted on a server, the options here are :

Figure 2-4 : Index update frequency options :



Daily - The index will be updated when the UPDALL task is run on the server overnight.

Scheduled - requires an UPDALL task to be run on the database through a program document on a schedule.

Hourly - The index will be updated hourly by the UPDATE task.

Immediate - The index will be updated when a change is made.

Note : Immediate could mean up to several minutes depending on how busy the server is.

We will be looking closer at the options to update and maintain the index later.

Creating the FTI for multiple databases using the Administrator Client

We've seen above how the FTI can be generated for a single database, if indexes are required for multiple databases this process would become tedious and very time consuming for the administrator.

To access this :

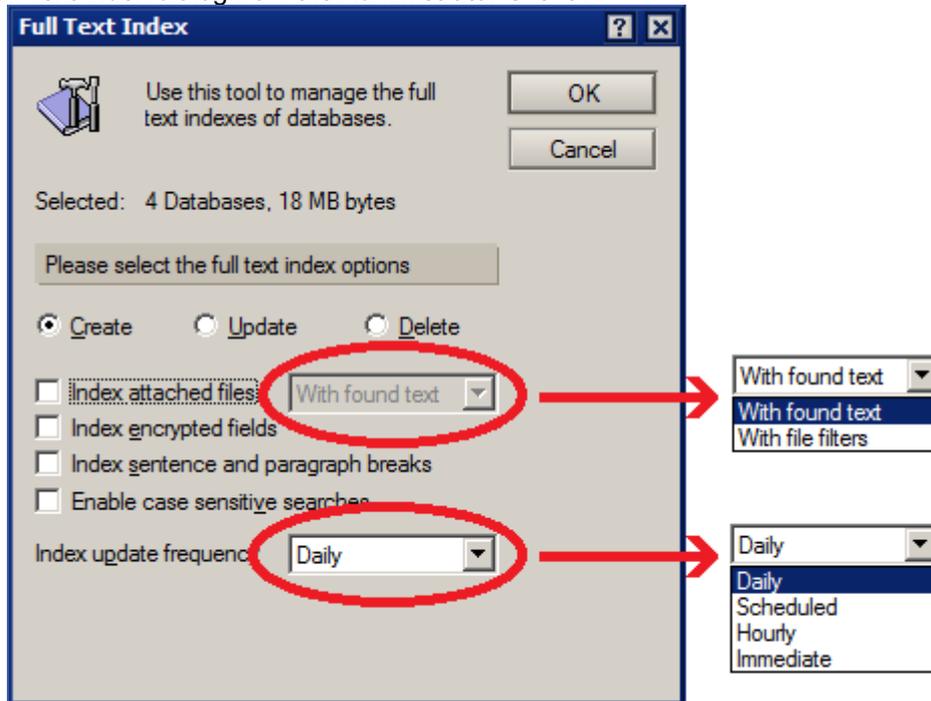
Open the Administrator Client.

Select the Files tab.

Highlight the databases that require an FTI. Multiple database can be selected either using the Shift key or the CTRL key and clicking on the individual databases.

Once selected select the Tools\Database\Full Text Index... option on the right hand side to open the Full Text Index dialog :

Figure 2-5 : Full Text Index dialog from the Administrator Client :



Here the same options for creating the FTIs are available.

Note : In a situation where a database previously was full text indexed and the index was removed through the operating system (not through a Notes Client) then it is possible to recreate the full text index on the server with the command :

```
load updall -f <database path and file name>
```

This is only possible if the Full Text Index was removed outside of a Notes Client.

We will be looking at further options to correctly remove a full text index later on.

Note : Manager rights are needed in the database's ACL to be able to create the Full Text Index. If the user creating the Full Text Index does not have manager rights to the database then they will not be able to create the full text index. The Create Index options will not be active.

Chapter 3 - Maintaining the Full Text Index.

With every new, deleted or edited document the Full Text Index becomes out of date and may give incorrect results.

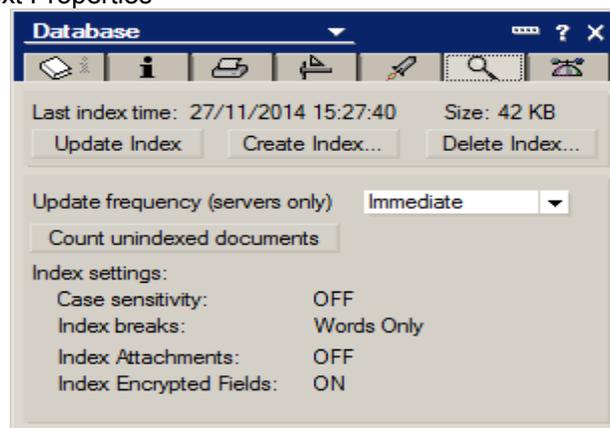
A method is then needed to update the Full Text Index to incorporate the changes allowing it to be used successfully again.

Manually updating the Full Text Index

Using a Notes Client

As when creating a Full Text Index select the database and open the database properties. Select the 'Full Text Tab' (second from the far right) :

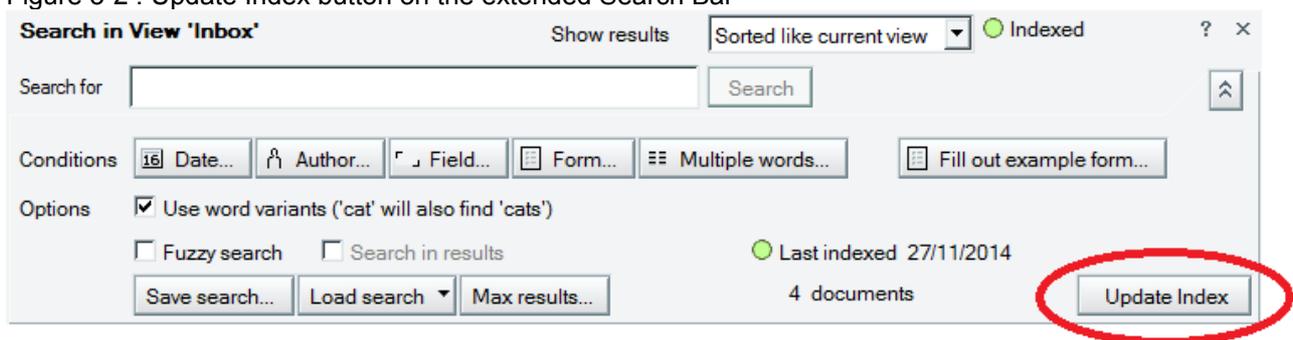
Figure 3-1 Database Full Text Properties



By clicking on the "Update Index" button the index will be updated with any changes that were made to the database.

Another option to update the index using the Notes Client is displayed in the Search bar (more on the Search Bar when we go through using a Full Text Index). When the extra options are accessed in this bar there is a button available - Update Index. This will update the index when activated :

Figure 3-2 : Update Index button on the extended Search Bar

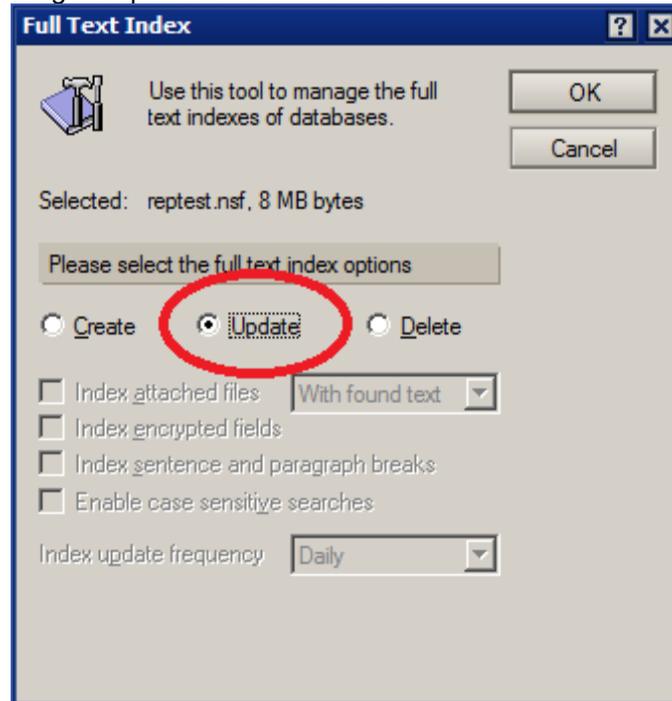


Using the Administrator client.

For server based databases the Administrator client can be used to update one or more indexes. As when creating multiple Full Text Index:

- Open the Administrator Client.
- Select the Files tab.
- Highlight the database or databases whose indexes need to be updated.
- From the Tools menu on the right select Tools\Database\Full Text Index...
- Select the Update option and Ok to update the indexes for the selected databases.

Figure 3-3 : Manually updating multiple Full Text Indexes :



Using the Server Console.

The "Updall" command on the server console can be used to both update the View Indexes in a database and the Full Text Indexes.

The command line to only update the Full Text Index for a database is :

```
load updall -f <database path and file name>
```

e.g.

```
> load updall -f ftitest.nsf
[3964:0002-0724] 27/11/2014 16:00:10   Index update process started:  -f
ftitest.nsf
[3964:0002-0724] 27/11/2014 16:00:10   Updating views in
C:\IBM\Domino\data\ftitest.nsf
[3964:0002-0724] 27/11/2014 16:00:10   Index update process shutdown
>
```

This can be run on individual databases, all databases in a folder or all databases on the server.

There are several other options with the Updall command that will be reviewed below.

All of these options will update the full text index allowing for accurate searches of the current data, however these are user driven, it is not desirable to need to update the index manually before each search. Because of this there is an automated option to maintain the Full Text Index.

Automatically Updating the Full Text Index

When creating the Full Text Index we saw that there were 4 options for the update frequency of the index on a server:

Daily
Scheduled

Hourly
Immediate

Daily and Scheduled options are carried out with the Updall task and Hourly and Immediate are handled by the UPDATE task.

Daily

By default the Updall task will run on all databases in a server at 2am. This performs several tasks including updating the View Index, purging old deletion stubs and updating the Full Text Index for databases that have an index.

This is configured in the server's NOTES.INI file in the line :

```
ServerTasksAt2=UpdAll
```

Scheduled

The update of the full text index of a database can be configured to run at any time through the use of a Program document :

- Open the Administrator client.
- Select the Configuration tab.
- In the Views on the left select Server\Programs
- Click on the Add Program or open and edit an existing Program document.
- Fill in these options as described :

In the Basics column

Program Name: Updall

Command Line: -f <database path and file name> (The '-f' option will only update the Full Text Index).

Server to run on: <the name of the server that is hosting the database>

In the Schedule column

Enabled/disabled: Enabled

Run at times: <Enter the time that you require the update to run. Leave blank if you wish to use an interval time>

Repeat interval of: <Enter the interval between updates. If zero (0) then the update will only happen once>

Days of week: <Select the days that you require the update to run>

Note : In the "Server to run on" field the full hierarchical name of the server should be used or an asterisk (*) to run the program document on all servers in the domain.

Figure 3-4 : Program document for scheduling Updall :

The screenshot shows a configuration window titled "Program: Updall". It has two tabs: "Basics" (selected) and "Administration". The "Basics" tab is divided into two sections: "Basics" and "Schedule".

Basics		Schedule	
Program name:	Updall	Enabled/disabled:	Enabled
Command line:	-f FTITEST.NSF	Run at times:	each day
Server to run on:	TestServer1/TST	Repeat interval of:	15 minutes
Comments:		Days of week:	Sun, Mon, Tue, Wed, Thu, Fri, Sat

Hourly and Immediate

Hourly and Immediate updates on the server are handled by the Update task (not to be confused with the Updall Command). The Update task's primary use is to maintain the View Indexes for the databases.

The Update task runs constantly on the server. It is loaded on the server when it starts in the ServerTasks line of the NOTES.INI :

```
ServerTasks=Update,Replica,Router,AMgr,AdminP,CalConn,Sched,HTTP,LDAP,RnRMgr
```

The Update task works continuously from a queue called the \$UpdateQueue. When a change has been made in a database, such as deletions, additions and edits, a corresponding request is entered into the Update queue. Update checks the queue every 5 seconds for any new requests that have been deposited, plucking the requests from the queue on a first-come first-served basis.

Although Update checks the queue every 5 seconds, it does not refresh indexes at the same interval. Instead, it uses what is known as the Update Suppression Time. With Suppression Time, Update waits for multiple, similar requests to be deposited in the queue and then batches them. In this way, Update processes all changes to a database at the same time. Since the Indexer is the most CPU-intensive Domino server task batching requests reduces the performance impact on a server significantly. It is only after the Suppression Time has passed that Update forces the update of the view collection and the full text indexes as requested. By default, the Suppression Time is 15 minutes; however, this can be overridden with the following NOTES.INI parameter if required :

```
Update_Suppression_Time=minutes
```

After updating view indexes in a database, it then updates all databases that have full-text search indexes set for immediate or hourly updates.

When Update encounters a corrupted view index or full-text index, it rebuilds the view index or full-text index in an attempt to correct the problem. This means it deletes the view index or full-text index and rebuilds it. However, some corruption will not be discovered by the Update task requiring a manual rebuild.

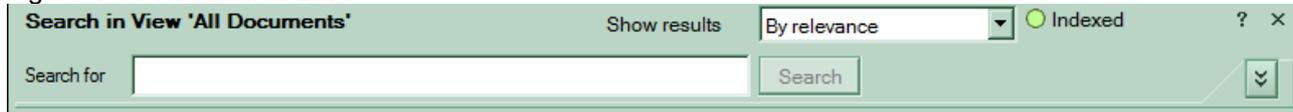
Chapter 4 - Using the Full Text Index.

So, you have created the Full Text Index - now you want to use it.

Open the database you want to search through.

With the database open you may see the search bar at the top of the screen, if not select it through the menu option View\Search this view :

Figure 4-1 : The Search Bar :



Individual words or phrases can be entered.

Restricted words :

Certain words are reserved for use by the Full Text search engine.

TOPIC, AND, NOT, OR, CONTAINS, NEAR, ACCRUE, EXACTCASE, TERMWEIGHT, PARAGRAPH, FIELD, and SENTENCE

To search for these words you need to place the words in “quotes”. If the words are part of a phrase the whole phrase should be within the quotation marks.

FTI supports Boolean searches using 'and', 'or', etc. This allows for the search of multiple words or phrases that may not be together in the document.

e.g. The search term : this and that

will find documents with both the words “this” and “that” but not only one of the words.

The search term : this or that

will find any document containing either the words “this” or “that” even if the other words in the search are not in the document.

These options can be extended to longer phrases.

The search term : “This text to be found” and “find these words” will find documents with both of those phrases as listed.

Further search operators are available, these can be reviewed in the Administrator Help document : Refining a search query using operators.

http://www-01.ibm.com/support/knowledgecenter/SSKTWP_9.0.0/com.ibm.notes900.help.doc/sch_refine_query_r.html

Note : To get the best results from using the Full Text Index it is important to understand that this is a WORD search engine – the search will try and match the Search Term to words not to a binary match.

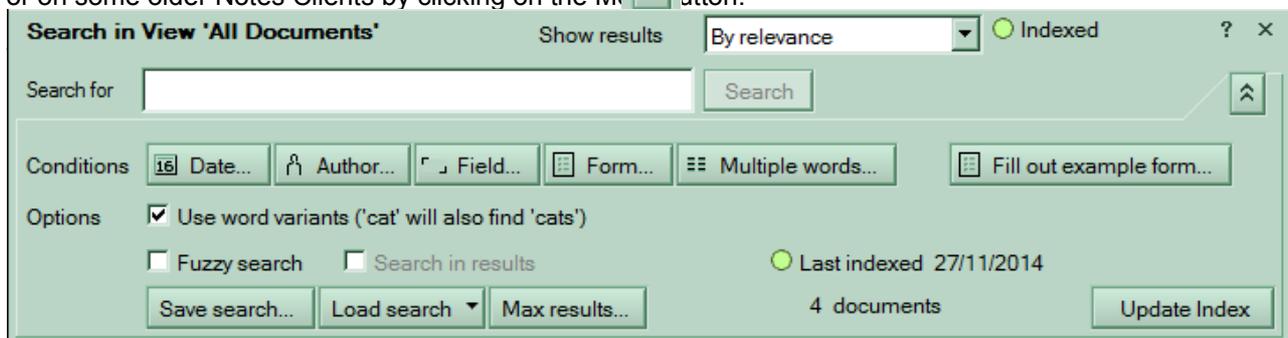
e.g. Searching for the term “fun” will not find the word “functionality” within a document even though the letters “fun” are a part of the work “functionality”.

To be able to find the word “functionality” with the search term “Fun” a wildcard asterisk needs to be used. With the search term “fun*” being used then the results will find “functionality”.

Advanced Search Options

Further search options can be accessed by clicking on the arrow down button :

or on some older Notes Clients by clicking on the  button.



- Show results
- By relevance
- By Date (last modified)
- By Date (first modified)
- Sorted like current view
- Within all documents

This allows the results to be sorted as needed.

Date...

Clicking on the 'Date...' button will open the Add Condition dialog, with several options, focusing on the "By Date" option here. The other options will be discussed individually.

Figure 4.3 : Add Condition, By date

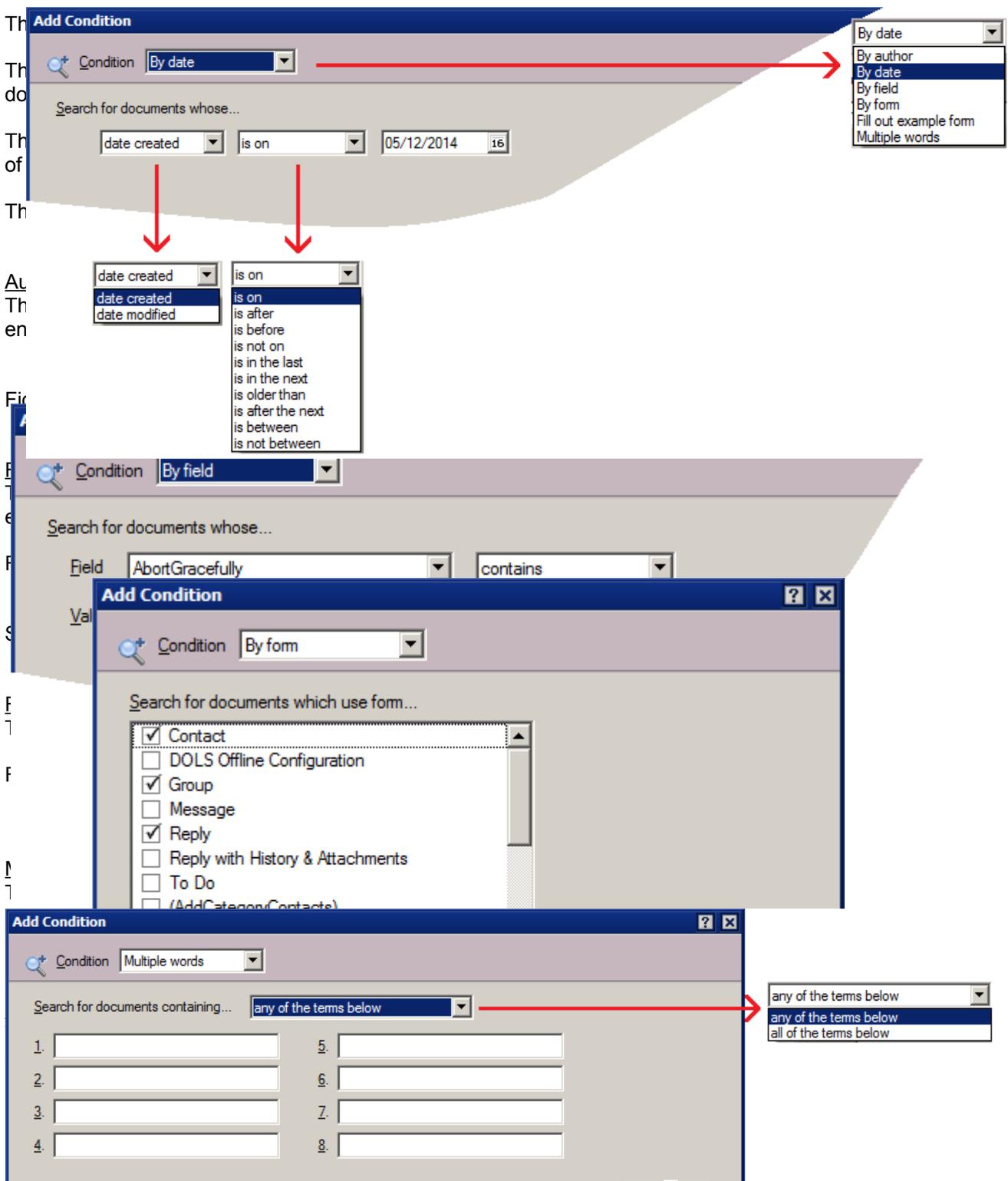


Figure 4-8 : Add Condition, Fill out example form

Condition: Fill out example form

Form: Contact

Contact Name: []

Company Name: [] Job Title: []

E-mail:

Business: []

Personal: []

Assistant: []

Business 2: []

Personal 2: []

Addresses:

Business: []

Street: []

City: []

State/Province: []

ZIP/Postal code: []

Country/Region: []

Notes will find documents similar to the example you provide. Use single words or short phrases and fill out only those fields which are relevant to your search.

Add Cancel

This option is available after a search. It allows a further search where only the current result documents are searched through.

Save Search

As seen above there are many options available for a search, these can become very complex and time consuming to enter. To make this more usable search options can be saved allowing them to be re-used without having to enter all the variables again.

Figure 4-9 : Save Search

Save Search

Name: []

Share this search with other users

OK Cancel

The Save Search option allows the user to save the search to whatever name is desired. The search can also be made available to other users.

Load search

Clicking on this button will display the available saved searches and allow the deletion of any saved search.

Max results...

This configuration limits the maximum number of results being displayed.

Figure 4-10 : Maximum Results

Maximum Results

Maximum number of matches: 65535

The value zero uses the default, server based setting; or the Notes preference for local databases.

OK Cancel

Note : Lowering this value will not speed up the search as the search results are first all found, then displayed as configured (By relevance, date etc). So, the functionality is that all results are found, then sorted and the maximum number of them are then listed in order.

Chapter 5 - Things to be aware of with a Full Text Index.

There are many different assets for the Full Text Index, here we will be looking at :

- 5.1) Updall - all the FTI options
- 5.2) Removing the Full text index
- 5.3) Moving the FTI from one server to another
- 5.4) Attachments
- 5.5) Running a separate thread for Full Text Indexing - UPDATE_FULLTEXT_THREAD
- 5.6) Stopping the Full Text Indexer - UPDATE_NO_FULLTEXT=1
- 5.7) Option for storing FTIs in other locations : FTBasePath=
- 5.8) Highlighting
- 5.9) Reader fields - are the results found and displayed?
- 5.10) FTI does not replicate
- 5.11) The FTI is different to the Domain Index
- 5.12) Show Schedule - to view the updall schedule
- 5.13) How to confirm if a database is full text indexed
- 5.14) Other types of searches :
 - Simple Search
 - View Search
 - Domain Index Search

5.1) Updall - all the FTI options

As discussed in Chapter 3 - Maintaining the Full Text Index the Updall command can be used to maintain the Full Text Index. It can be run manually by entering the command in the server console, through a program document and it is used in the NOTES.INI to run overnight.

If run without any options Updall will first update the View Indexes in a database and then the Full Text Index.

Updall can be used on all databases by not entering a database path and filename, on an entire folder or on an individual database. It will not accept wildcard characters.

There are several switches available for the Updall command that don't affect the Full Text Index these can be reviewed either in the Administrator Help or by entering the command :

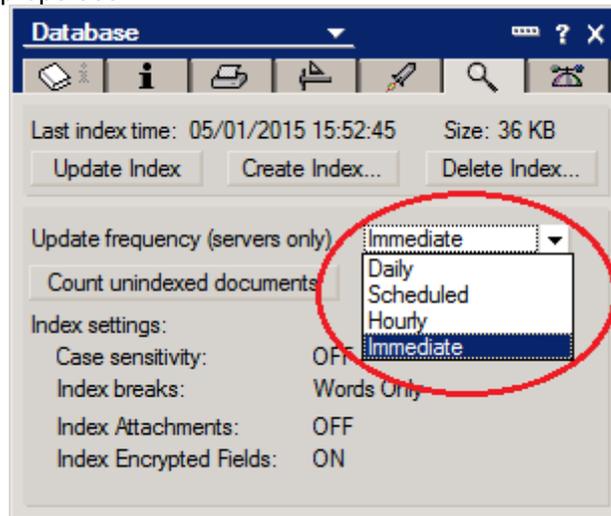
load updall /?

The Full Text Index related switches are :

- f Update only full text indexes, regardless of their update frequencies.
 - h Update only full text indexes of 'immediate' or 'hourly' frequency.
 - m or -s Update only full text indexes of 'immediate', 'hourly' or 'scheduled' frequency.
 - l Update only full text indexes of 'immediate', 'hourly' or 'daily' or 'scheduled' frequency.
 - x Rebuild all full text indexes.
- y or -Y Recurse through subfolders (-Y, default) or do not recurse (-y)

“-f”, “-h”, “-m”, “-s” and “-l” all refer to the Update frequency setting in the Database properties for the Full Text Index :

Figure 5-1 : FTI scheduling properties



The remaining Update options are mostly related to View Indexing and not to Full Text Indexing :

- t "name" Update single view named "viewname" only.
- T+ "name" Special view named "viewname" is a critical view.
- t- "name" Special view named "viewname" is not a critical view.
- c Build unused view indexes.
- v Update existing and already built view indexes only.
- r Rebuild all already built view indexes.
- p Obsolete, on by default.
- a Incremental full text index update of search site databases (faster).
- b Complete full text index update of search site databases (slower).
- g Remove build-on-first-use collations when rebuilding views
- nodbmt Do not perform the dbmaintool nightly tasks

5.2) Removing the Full text index

There are times when removing the Full Text Index is required. This can be due to disk space requirements, the Full Text Index no longer being required or to reduce the load on the Update Task.

The safest option to remove the FTI is to select the database, open the database properties, select the FTI tab and click on "Delete Index...", a confirmation will be requested and then, once approved, the index will be removed by the server.

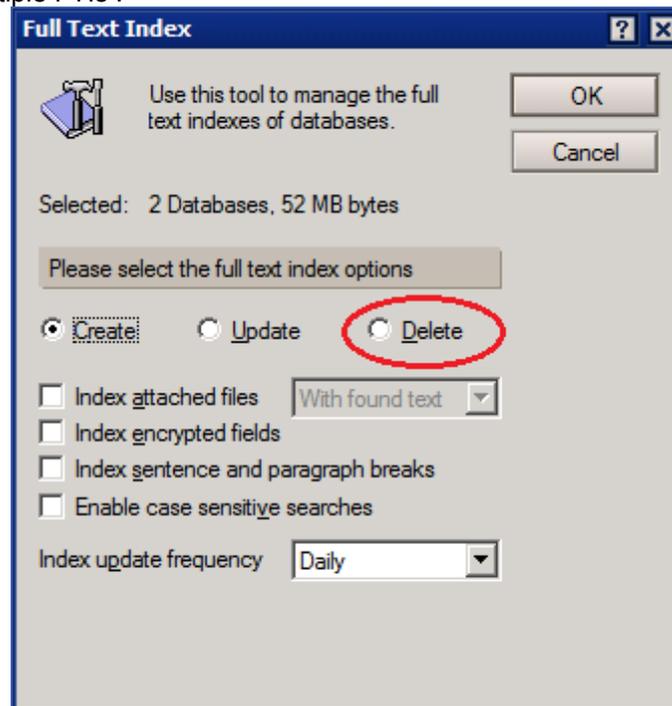
If the server does not have deletion rights to the location for the FTI then this may fail.

Also, if the Index is being updated at the time of the deletion request the deletion can fail.

Multiple FTIs can be deleted using the Files tab of the Administrator client, select the databases required to have their FTIs deleted and click on Tools\Database\Full Text Index... to open the Full Text Index dialog box.

From here the option to 'Delete' is available, once selected click on OK and confirm the request :

Figure 5-2 : Deleting multiple FTIs :



If the above options to not remove the Full Text Index it is also possible to simply delete the .FT folder that contains the Full Text Index. However, this can affect the functionality of the database in that it will still be configured as having a Full text index but the functionality will not be available.

5.3) Moving the FTI from one server to another

Moving a database from one server to another will not normally move the Full Text Index. In some cases it is possible to have the target server recreate the Full Text Index when the database has been moved but this is a new Index and not the original index.

Some users have needed to move the Full Text Index itself and this is possible by simply copying the .FT folder through the operating system.

The target database must be configured as having been full text indexed, with its own index deleted at the operating system level and the new index replacing it in the matching location for the database, but the index may not be present.

The most common need for this is where the database is extremely large and the server very busy leading to the resources for updating the FTI not being available or very slow. In this case the index can be created on another server and copied.

The entire structure of the FT folder will need to be copied.

This is not normally recommended as the documents in the database the index is created from, even though it is a replica, may not match the documents database being searched at the time of the search. Also, maintaining the index can be a problem if the index is being built on another server.

5.4) Attachments

Documents can contain attachments with text in them. The Full Text Indexer is able to parse text from these and a search will return the documents the attachments are in.

Text within attached graphics (BMP, JPG, GIF, etc.) cannot be added to the index – so if there is text listed in a screen shot or other picture this cannot be added to the index.

As listed above the option to include attachments in a Full Text Index can be selected when creating the index. The further option is then available whether to use conversion filters or not.

Conversion filters will attempt to parse only the text within the attachments that would be displayed by the attachment normally - so any programming or formatting data stored in the file will not be included in the index.

By not using the conversion filters the indexing process can be faster as the indexing task does not need to extract the text from the files. However, the resulting index is often larger as more text is added to it and it can contain text for information that is not displayed from the attachment itself possibly leading to incorrect search results.

By using conversion filters the time and resources needed to create and maintain the index can increase, but the overall size of the index will be smaller as less text is added to it, searches will only find text that should be displayed in the attachment.

Several attachment types are not included in the FTI by default :

.au, .cca, .dbd, .dll, .exe, .gif, .img, .jpg, .mp3, .mpg, .mov, .nsf, .ntf, .p7m, .p7s, .pag, .sys, .tar, .tif, .wav, .wpl, .zip.

Further attachment types can be configured to not be included in the FTI using the NOTES.INI parameter :

`FT_INDEX_ATTACHMENTS=value`

To specify more than one type separate the values with commas.

When the search text is found within an attachment on opening the link to the resulting document the file name of the attachment will be highlighted. However, when the attachment is opened the search term will not be highlighted within the opened file. More on highlighting in section 5.8 – Highlighting.

At a database level the indexing of attachments can be enabled and disabled through the database properties. However, it is also possible to enable or disable the indexing of attachments for the entire server. This can be controlled with the NOTES.INI parameter :

`FT_Index_Attachments=<value>`

If this is set to '1' then attachments will be indexed.

If this is set to '0' then attachments will not be indexed.

5.5) Running a separate thread for Full Text Indexing - UPDATE_FULLTEXT_THREAD

By default the UPDATE task will maintain Full Text Indexes that are set to Immediate on the server. This task will first review the View Indexes of a database and then, once this is completed, it will update the Full Text Index as required.

By adding “UPDATE_FULLTEXT_THREAD=1” to the NOTES.INI of the server and restarting the server it splits the Full Text Indexing into its own thread on the server. This will require further resources from the server but can result in the Full Text Indexes being maintained more promptly.

When this is enabled the new task will be used on the server, the “Show tasks” command will show this as :

```
[93DC:0008-2AD4] Full Text Indexer  Idle
```

5.6) Stopping the Full Text Indexer - UPDATE_NO_FULLTEXT=1

Enabling the NOTES.INI parameter “UPDATE_NO_FULLTEXT=1” on a server will stop the server from creating new FTIs and updating existing FTIs.

Existing Full Text Indexes will still function normally but will not be updated.

This can be used to prevent a server from using resources maintaining any Full Text Index except when required but the indexes may quickly be out of date compared to the documents in the databases.

5.7) Option for storing FTIs in other locations : FTBasePath=

The use of the NOTES.INI parameter “FTBasePath=<folder>” will allow the storage and creation of Full Text Index in a location on a drive other than where the databases are stored.

This can be used where the server disks are low on space or where DISK I/O needs to be optimized.

This can be set to a different drive outside of the DATA folder.

It can only be set at a server level for all databases not for individual databases and only one location can be configured for the server.

The Full Text Indexes will be recreated in the new location matching the folder structure of the databases in the DATA folder.

e.g.

If the FTBasePath is set to “X:\FTI_path” and the databases are in the data folder like this :

```
DATA\MAIL\User1.NSF
DATA\MAIL\User2.NSF
DATA\MAIL\User3.NSF
DATA\APPLICATION_DBS\APP_DB1.NSF
DATA\APPLICATION_DBS\APP_DB2.NSF
DATA\APPLICATION_DBS\APP_DB3.NSF
```

Then the Full Text Indexes as created will be :

```
X:\FTI_path\MAIL\User1.FT
X:\FTI_path\MAIL\User2.FT
X:\FTI_path\MAIL\User3.FT
X:\FTI_path\APPLICATION_DBS\APP_DB1.FT
X:\FTI_path\APPLICATION_DBS\APP_DB2.FT
X:\FTI_path\APPLICATION_DBS\APP_DB3.FT
```

5.8) Highlighting

When viewing the results of a search using a Notes Client the matching text in the document will be highlighted when viewed.

As mentioned in the Attachment section the attachment file name will be highlighted if the string is found in the attachment.

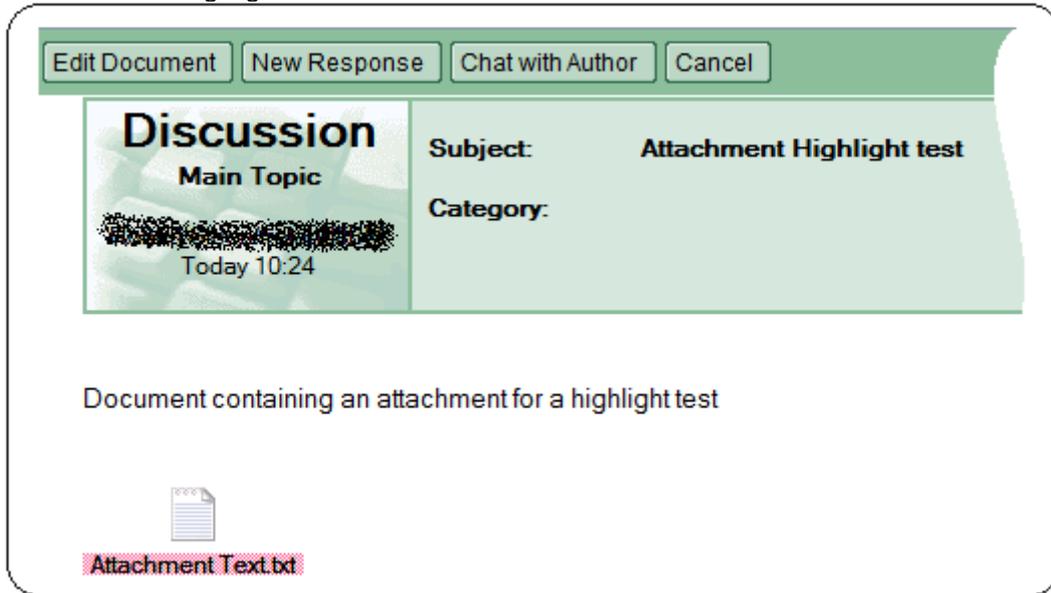
e.g. Searching for the term “search results” found one document and highlighted the text :

Figure 5-3 : Highlighted text :



e.g. Searching for text in an attachment. In this example the text being searched for is “Text In”, and it was found in an attached text file called “Attachment Text” :

Figure 5-4 : Attachment Highlight :



Here the attachment file name is highlighted when the search term is found in the attachment. Opening the attachment, even opening the attachment in a Notes Client will not then highlight the search text in the file.

Highlighting gone wrong

The highlighter will attempt to highlight the matching word to the search string. Although in most cases this can be simple, it also will attempt to highlight variants and fuzzy matches if enabled – this can lead to confusion on which piece of text is highlighted. This can occur when there are a large number of symbols or other characters in a document. So, this can lead to the wrong words being highlighted.

Search results slow to open

There are times when opening a document from the search results is very slow. Opening the document normally when not in search results has no problems. This can be caused by the highlighter parsing through attachments in the document to see if the attachment should be highlighted.

This can be stopped with the NOTES.INI parameter :
FT_LIMIT_HIGHLIGHT_FILTER=1

This setting prevents the highlighter from parsing through documents that would not contain the search text.

Note : Search results are not highlighted when searching using an Internet Browser.

5.9) Reader fields - are the results found and displayed?

When a document contains a Reader field that the user is not included in then that user should not be able to access that document, however in a search on a server the Server ID is used to perform the search and the Server ID may have access to the resulting document.

In this situation the document with the Reader field is still not displayed in the search results. Although the document is found by the search it is then filtered out from the display.

By enabling the debug option "DEBUG_FTV_SEARCH=1" (more on this in Chapter 6) we see what the server has found :

```
IN FTGSearch
[9164:0081-7EA0] option = 0x400219
[9164:0081-7EA0] Query: (search AND text)
[9164:0081-7EA0] Engine Query: ("search"%STEM * "text"%STEM)
GTR query performed in 6 ms. 2 documents found
[9164:0081-7EA0] 0 documents disualified by deletion
```

```
[9164:0081-7EA0] 0 documents disqualified by ACL
[9164:0081-7EA0] 0 documents disqualified by IDTable
[9164:0081-7EA0] 1 documents disqualified by NIF
[9164:0081-7EA0] Results marshalled in 1 ms. 1 documents left
[9164:0081-7EA0]   OUT FTGSearch error = 0
[9164:0081-7EA0] FTGSearch: found=1, returned=1, start=0, count=0, limit=0
[9164:0081-7EA0] Total search time 8 ms.
```

Here the debug output is showing that 2 documents have been found in the search, it is then showing that 1 document was disqualified by NIF – this is the Notes Indexing Facility. This is showing that the document was removed from being displayed due to the Reader field.

The final number of search results that are displayed to the user is shown as “Results marshalled in 1 ms. 1 documents left”.

The number of search results displayed in the Notes Client status bar will only show the displayed count.

If the reader field is then disabled the debug output becomes :

```
IN FTGSearch
[9164:0081-351C] option = 0x400219
[9164:0081-351C] Query: (search AND text)
[9164:0081-351C] Engine Query: ("search"%STEM * "text"%STEM)
[9164:0081-351C] GTR query performed in 6 ms. 2 documents found
[9164:0081-351C] 0 documents disualified by deletion
[9164:0081-351C] 0 documents disqualified by ACL
[9164:0081-351C] 0 documents disqualified by IDTable
[9164:0081-351C] 0 documents disqualified by NIF
[9164:0081-351C] Results marshalled in 1 ms. 2 documents left
[9164:0081-351C]   OUT FTGSearch error = 0
[9164:0081-351C] FTGSearch: found=2, returned=2, start=0, count=0, limit=0
[9164:0081-351C] Total search time 7 ms.
```

Showing that the document is not being removed from the results.

5.10) FTI does not replicate

The Full text Index for a database does not replicate – under any circumstances.

The Index file can be copied from one server to another, but not replicated.

When creating a new replica of a database it is possible to have the Full Text Index created on the target system – but this is not a replica of the original Full Text Index it is a new Full Text Index.

Figure 5-5a : Create a new FTI when creating a new replica of a database :

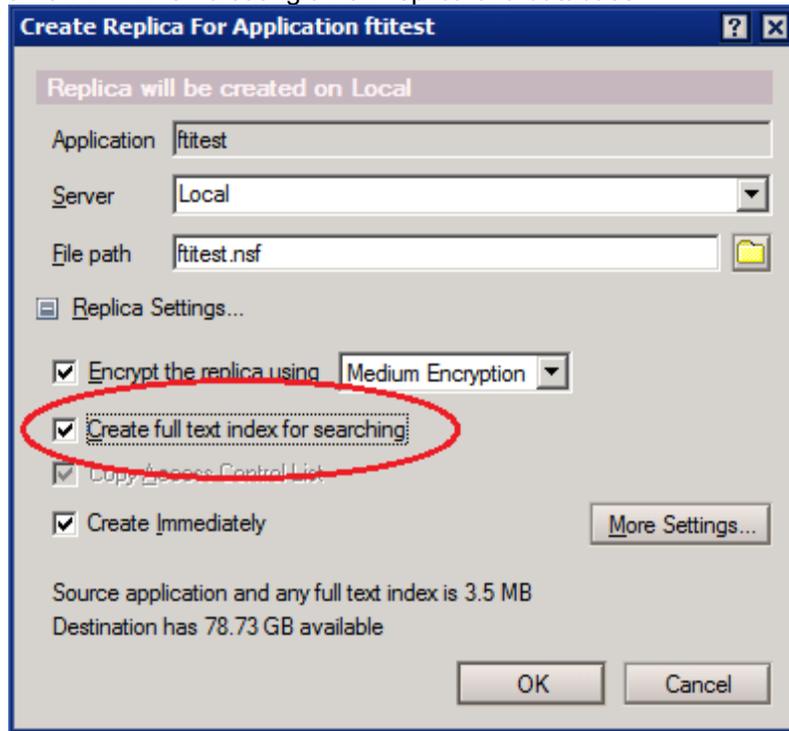
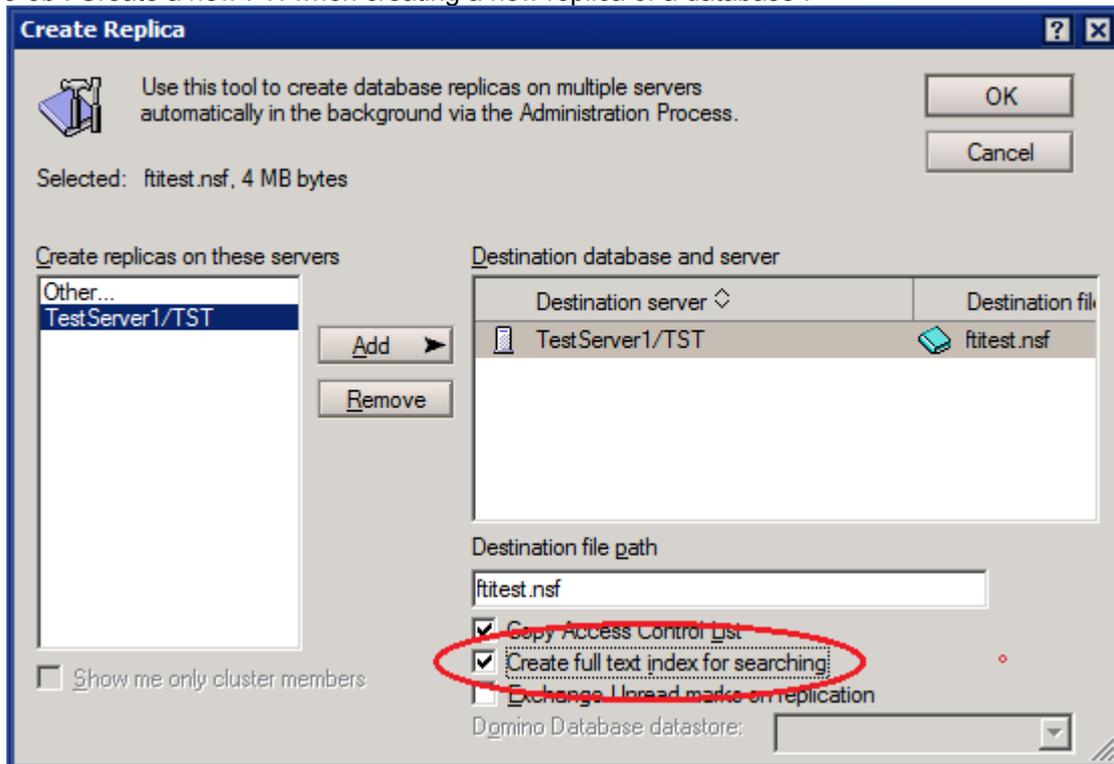


Figure 5-5b : Create a new FTI when creating a new replica of a database :



As the Full Text Index does not replicate this can lead to differences in results between the same searches performed on different replicas of the same database. This can be caused by the Indexes not being updated at the same time and not reflecting changes made to the documents or by documents themselves not replicating for any reason.

5.11) The FTI is different to Domain Index or Domain Search

The Full Text Index is an index created for a single database. It is specific to that database in one location – either a server or locally on a Notes Client.

The Domain Index is an Index that can cover multiple databases on one or more servers. It does not have all the search functions of the Full Text Index search and will not be maintained either by the Update or the Updall functions. Databases can have both a Full Text Index and be included in the Domain Index.

Further information can be found in the Administrator Help : Domain Search

[http://www-](http://www-12.lotus.com/ldd/doc/domino_notes/9.0/help9_admin.nsf/b3266a3c17f9bb7085256b870069c0a9/1b521d873a37f0cc85257b19005b4174?OpenDocument)

[12.lotus.com/ldd/doc/domino_notes/9.0/help9_admin.nsf/b3266a3c17f9bb7085256b870069c0a9/1b521d873a37f0cc85257b19005b4174?OpenDocument](http://www-12.lotus.com/ldd/doc/domino_notes/9.0/help9_admin.nsf/b3266a3c17f9bb7085256b870069c0a9/1b521d873a37f0cc85257b19005b4174?OpenDocument)

5.12) Show Schedule - to view the updall schedule

The server command "Show Schedule" (or "sh sched") will display a list of all replication, routing and program events scheduled to run next on the server.

This includes all program documents and can be used to confirm when a scheduled update to a Full Text Index should next happen.

e.g :

```
> sh sched
[9164:0008-8740] Scheduled          Type          Next schedule

[9164:0008-8740] TestServer1/Tst    Mail Routing   06/01/2015 13:30:00
[9164:0008-8740] TestServer1/Tst    Replication    06/01/2015 13:30:41
[9164:0008-8740] Updall            Run Program    06/01/2015 13:18:32
```

Here the next events are displayed including a program document what will run Updall.

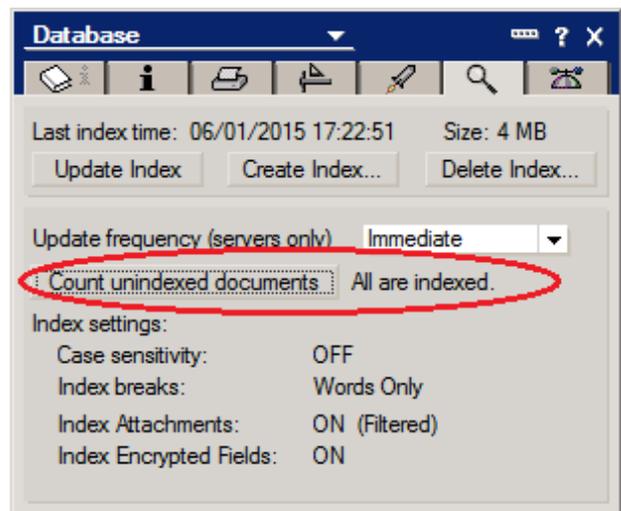
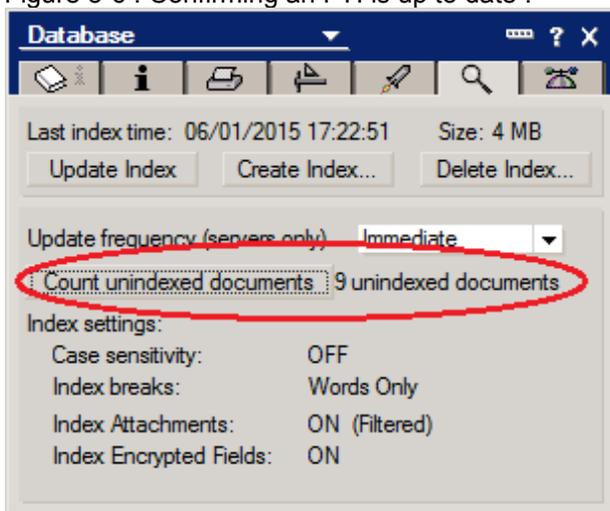
5.13) How to confirm if a database is full text indexed

Every time a document is edited, added or deleted from a database then the Full Text Index becomes out of date.

To check if the Full Text Index is up to date :

- 1) Using a Notes Client select the database to be tested.
- 2) Open the database properties and select the Full Text tab.
- 3) Click on the button "Count unindexed documents" to display if there are documents to be indexed.

Figure 5-6 : Confirming an FTI is up to date :



5.14) Other types of searches :

- Simple Search
- View Search
- Domain Index Search
- Search in a document

There are other searches available with Domino Databases – these have different functionality and properties.

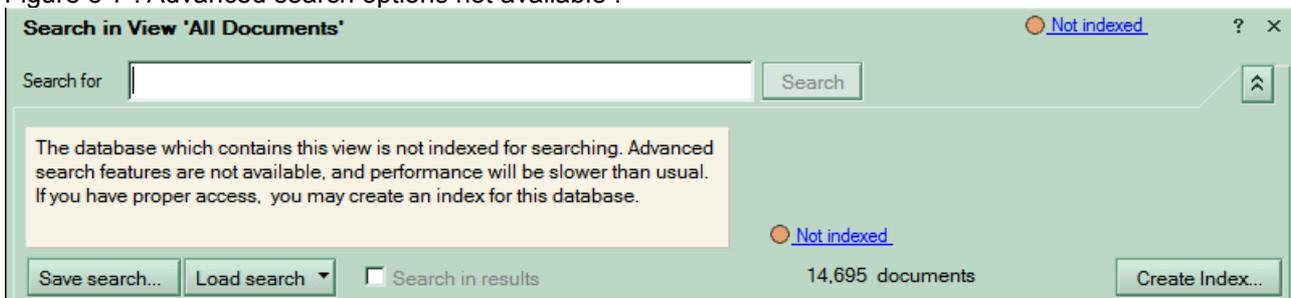
Simple Search

This refers to a search through the documents for a database where there is no Full Text Index.

Searches are still possible when there is no Full Text Index however the search results will be slower and a lot of the advanced search options will not be available.

When accessing the advanced features for a non-Full Text Indexed database this is displayed :

Figure 5-7 : Advanced search options not available :

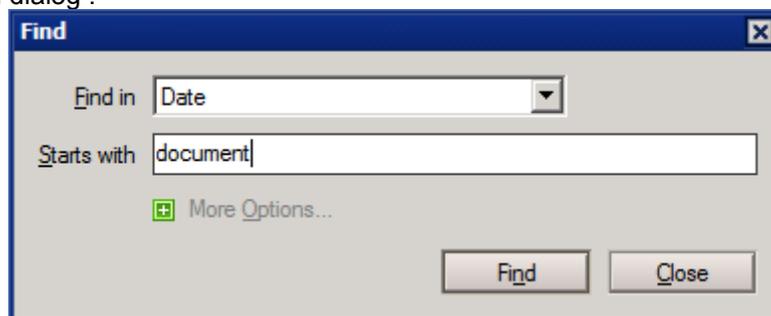


Text within attachments will not be searched either.

View Search

When looking at a list of documents in a view and selecting the menu option "Edit\Find Next..." the "Find" dialog is displayed. This allows for searches in the text displayed within the view, in the columns. It will not find text within the documents, only text displayed by the view to sort the documents.

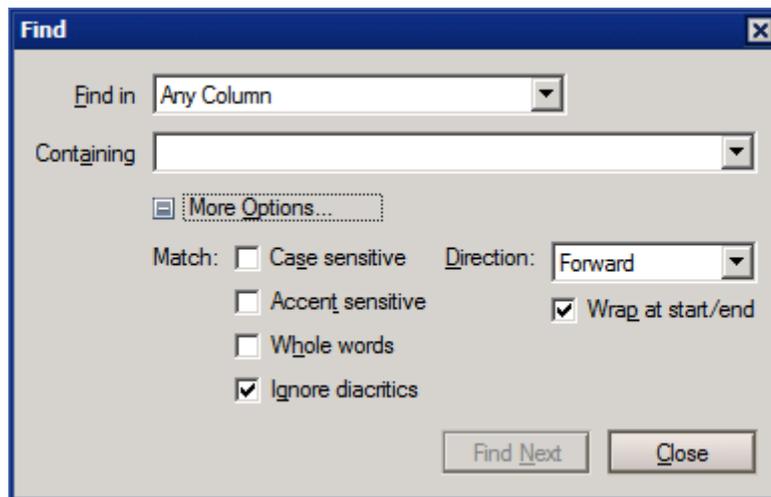
Figure 5-8 : The Find dialog :



The "Find in" field allows the selection of which available column from the View to search through.

Clicking on "More Options" will increase the search options to :

Figure 5-9 : The Find dialog – displaying More Options :



This method of searching places the current focus on only the first document that meets the criteria; it does not return an entire collection of documents that match the search term.

This can be very useful in selecting an individual document from a view.

A quick option of activating this search is to click on documents in the view and start typing the search term, the 'Find' dialog will be activated and the search enabled.

Domain Index Search

As mentioned in section 5.11 the Domain Index is an Index that can cover multiple databases or one or more servers. It does not have all the search functions of the Full Text Index search and will not be maintained either by the Update or the Updall functions.

It is enabled and configured through the Domino Administrator. Which servers and databases are to be included in the index can be configured.

The search is initiated and the results viewed through the CATALOG.NSF database :

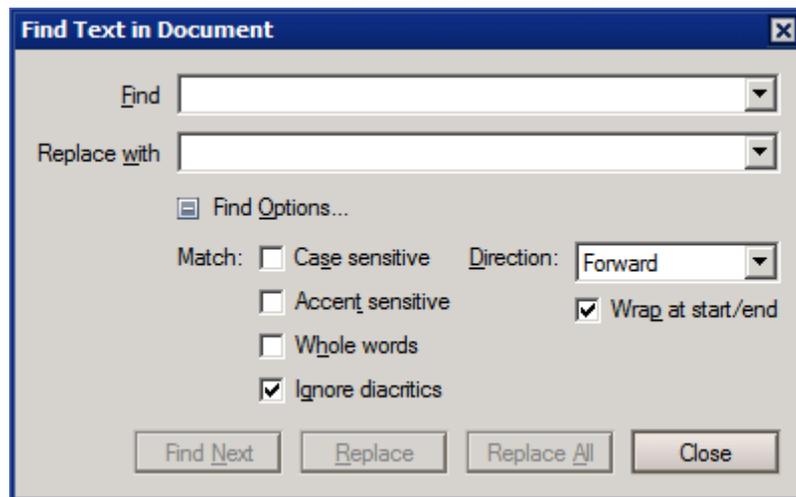
The Domain Index can be very large and very resource intensive to use and maintain. The selection of databases and update frequency for the index should be carefully planned.

Further information on Domain Indexing can be found in the Administrator Help document "Domain Search".
http://www-12.lotus.com/ldd/doc/domino_notes/9.0/help9_admin.nsf/b3266a3c17f9bb7085256b870069c0a9/1b521d873a37f0cc85257b19005b4174?OpenDocument

Search in a document

It is possible to search through the displayed text in an open document. This can be started either by clicking on the menu option "Edit\Find/Replace..." or by using Ctrl+F. This will open the "Find Text in Document" dialog.

Figure 5-10 ; The Find Text in Document dialog with full options displayed :



This allows the search for terms within the document. This is not a word search, so partial words can be searched for and variants or fuzzy matches will not be found.

Chapter 6 - Troubleshooting and debug options with a Full Text Index.

In this chapter we will be reviewing some general troubleshooting, Debug options to see more of what is happening and reviewing some more specific issues and areas.

General Troubleshooting:

For many issues with the Full Text Index the best solution and the first step should be :

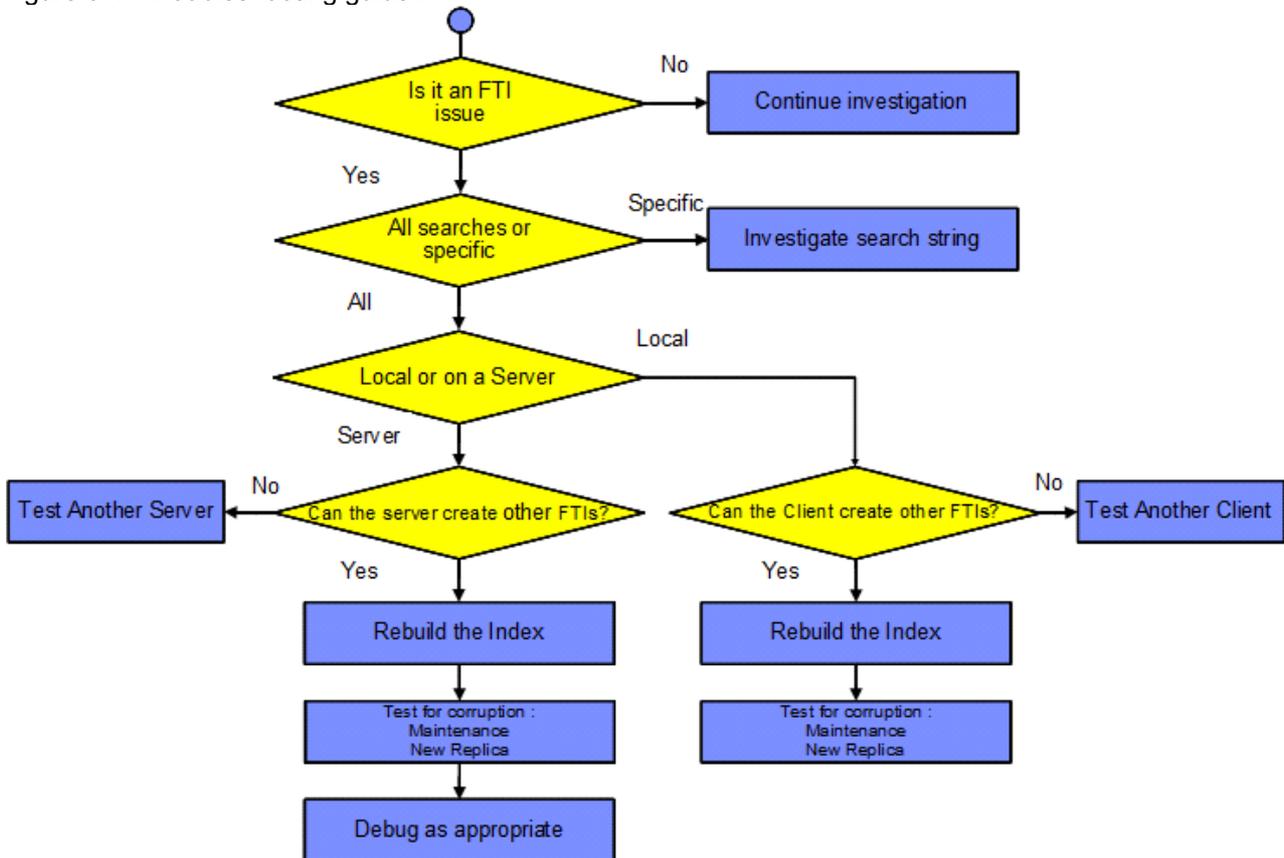
- 1) Delete the current Full Text Index – preferably through the database properties.
Confirm that the complete index with the .FT folder have been removed.
- 2) Re-create the Full Text Index.

This will ensure that the current Full Text Index is completely up-to-date with the latest document edits and that if the original index was corrupt that the corruption is replaced.

This will not resolve all Full Text Indexing issues but is a very good first step more most problems.

Further troubleshooting can follow this guide to help localize where the issue may be. Although this doesn't cover all possible issues it is a good guide :

Figure 6-1 : Troubleshooting guide :



Here the steps are :

Is it an FTI issue this may sound straight forward but often the issue may be with a View Index, or database corruption (more on this later). Confirm if the problem is with the FTI. If it is continue as here, if the problem is not with the index then the investigation should continue as required.

All searches or specific Are all searches affected by this issue or just specific searches? If only certain searches are affected and all other searches of this database are normal then the search term itself should be reviewed, there may be an issue with an operator command being used, or a quotation not being closed.

Local or on a Server Confirm if the database being searched is on a server or being accessed locally through a Notes Client, this information can determine what steps may follow.

If local

Can the Client create other FTIs Confirm if the Notes Client can create Full Text Indexes for other databases. This will confirm if the problem is with the Client itself.

If the Client is unable to create the index then the database should be tested on another Notes Client, it is possible that the install of the Notes Client is corrupt or something else is preventing the normal FTI functionality. The Notes Client may need to be re-installed.

If the Client can create fully working Full Text Indexes for other databases then the issue may be specific to the database or this Index.

The Index should be recreated and tested ensuring it was fully deleted.

Also, the database may be corrupt – running maintenance on the database or replacing it with a replica may resolve the issue.

Debug options (more on these later) may also be added to the NOTES.INI of the Notes Client and may help find the source of the issue.

If on a Server

Can the Server create other FTIs Confirm if the Domino Server can create Full Text Indexes for other databases. This will confirm if the problem is with the Server itself.

If the Server is unable to create the index then the database should be tested on another Server, it is possible that there is an issue with the server – there may be a corrupt file or setting preventing the index being created.

If the Server can create fully working Full Text Indexes for other databases then the issue may be specific to the database or this Index.

The Index should be recreated and tested ensuring it was fully deleted.

Also, the database may be corrupt – running maintenance on the database or replacing it with a replica may resolve the issue.

Debug options (more on these later) may also be enabled on the server and may help find the source of the issue.

FTI Debug options:

There are several debug parameters available that when enabled can increase the logging when creating, updating or searching with the Full Text Index.

The most used and useful parameters are :

DEBUG_THREADID
Debug_FTV_Search
Debug_FTV_Index
Debug_FT_Stream

Note : With the exception of DEBUG_THREADID debug options should be disabled once the required data has been generated, leaving them running can impact the servers performance and resources.

DEBUG_THREADID

Although not specific to Full Text Indexing this is a very important debug option.

This displays each thread and task on the server with a separate ID in the CONSOLE.LOG. With this enabled it is possible to follow each task through the console.log by the leading numbers on each line. This is very useful when there are multiple tasks being displayed on the console, even multiple indexing events.

Since R8.0 DEBUG_THREADID is enabled by default on all servers - however some administrators have disabled it by adding :

```
DEBUG_THREADID=0
```

into the NOTES.INI of the server.

Before starting any of the logging or debugging options DEBUG_THREADID=1 should be enabled on the test server, either by adding "DEBUG_THREADID=1" to the NOTES.INI and restarting the server or by running the command "set config DEBUG_THREADID=1".

This option does not impact the performance of the server, but will slightly increase the size of the CONSOLE.LOG.

If the server cannot be restarted it is possible to enable this DEBUG option using the server command :

```
set config DEBUG_THREADID=1
```

This will enable the debug until it is disabled by a user or until the server is restarted.

Adding "DEBUG_THREADID=1" to the NOTES.INI will ensure it is enabled when the server starts.

Debug_FTV_Search

Setting this to '1' will send information about searches to the server console. This will include the actual search string being used and some information on the results of the search.

This can be helpful in situations where a search term is not working, or if the results are not as expected.

This can be enabled on a running server without having to restart the server with the command :

```
set config DEBUG_FTV_SEARCH=1
```

or it can be added to the NOTES.INI of the server and the server restarted.

Setting this to '0' will disable the debug.

Example output :

```
IN FTGSearch
[2F04:007C-3518] option = 0x400219
[2F04:007C-3518] Query: (test)
[2F04:007C-3518] Engine Query: ("test"%STEM)
[2F04:007C-3518] GTR query performed in 9 ms. 5 documents found
[2F04:007C-3518] 0 documents disqualified by deletion
[2F04:007C-3518] 0 documents disqualified by ACL
[2F04:007C-3518] 0 documents disqualified by IDTable
[2F04:007C-3518] 1 documents disqualified by NIF
[2F04:007C-3518] Results marshalled in 5 ms. 4 documents left
[2F04:007C-3518] OUT FTGSearch error = 0
[2F04:007C-3518] FTGSearch: found=4, returned=4, start=0, count=0, limit=0
[2F04:007C-3518] Total search time 17 ms.
```

From this we can see :

The search term exactly as used by the search engine : test

The length of time the search took : 9 ms

The number of documents in the database matching the search : 5

Any documents that were removed from the results : 1 (due to a reader field in this test)

The time it took to order the results depending on the settings : 5 ms

If there were any errors reported during the search.

The total number of results being displayed : 4

The total search time : 17 ms

The search term will be displayed depending on how it is entered, in the above example the search term was just "test" (without quotes in the actual search). To show how a more complicated search term would be displayed :

Complete search term : "Test search term1" and "Test search term2" or "Test search term3" not "search term4"

```
IN FTGSearch
[2F04:007C-33B8] option = 0x400219
Query: ("Test search term1" AND "Test search term2" OR "Test search term3" AND
NOT "search term4")
[2F04:007C-33B8] Engine Query: ("Test search term1"%STEM * "Test search
term2"%STEM + "Test search term3"%STEM * ! "search term4"%STEM)
[2F04:007C-33B8] OUT FTGSearch error = F22
[2F04:007C-33B8] FTGSearch: found=0, returned=0, start=0, count=0, limit=0
```

Here we can see that the search term is split into the separate phrases being searched for and then the operands added.

Debug_FTV_Index

This parameter will output extra data when a Full Text Index is created or updated.

This can be enabled on a running server without having to restart the server with the command :

```
set config DEBUG_FTV_INDEX=<value>
```

or it can be added to the NOTES.INI of the server and the server restarted.

Setting this to '0' in the NOTES.INI will disable the debug once the server is restarted.

The output will vary depending on how it is configured.

```
Debug_FTV_Index=1
```

Example output :

```
[2184:0002-18F4] FTGIndex: After call to FTGIndexStart 2 ms for
[C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] FTGIndex: Get modified NoteID table 0 ms for
[C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] FTGIndex: Before calling IDEnumerate 242 ms for
[C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] FTGIndex: Finished: 15 ms. for [C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] 3 documents added, 3 updated, 3 deleted: 1630 text bytes; 144
numeric bytes for [C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] FTGIndex: All Done: 162 ms for [C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] OUT FTGIndex rc = 0 (No error) - for
[C:\IBM\Domino\data\ftitest.ft]
```

Here we can see :

The start of the indexing process.

The useful information displayed is :

The name of the Full Text Index – and from this the name of the database.

The time indexing took.

The number of documents added to the index, removed from the index and updated.

Any listed errors.

This setting with this debug will not have a significant affect on server performance but would not be recommended to be left enabled on the server.

```
Debug_FTV_Index=15
```

Setting this parameter will display data on each note in the database being indexed. This can be very long and will affect server performance, so it would be recommended to disable this once any testing is completed.

Example output :

```
[2184:0002-18F4] FTGIndex: After call to FTGIndexStart 1 ms for
[C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] FTGIndex: Get modified NoteID table 0 ms for
[C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] FTGIndex: Before calling IDEnumerate 2 ms for
[C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] Start indexing document 0xEEF6 (0 text bytes indexed)
[2184:0002-18F4] This create date is 01/08/2015 11:28:30
[2184:0002-18F4] This revision date is 01/08/2015 11:28:39
[2184:0002-18F4] Close document 0xEEF6 (262 text bytes indexed)
[2184:0002-18F4] Start indexing document 0xF4E6 (262 text bytes indexed)
[2184:0002-18F4] This create date is 01/08/2015 15:13:32
[2184:0002-18F4] This revision date is 01/08/2015 15:13:32
[2184:0002-18F4] Close document 0xF4E6 (524 text bytes indexed)
[2184:0002-18F4] FTGIndex: Finished: 79 ms. for [C:\IBM\Domino\data\ftitest.ft]

[2184:0002-18F4] 2 documents added, 0 updated, 0 deleted: 524 text bytes; 48
numeric bytes for [C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] FTGIndex: All Done: 129 ms for [C:\IBM\Domino\data\ftitest.ft]
[2184:0002-18F4] OUT FTGIndex rc = 0 (No error) - for
[C:\IBM\Domino\data\ftitest.ft]
```

If the index is being updated then only new or updated documents will be listed in the output, this could be useful to determine if a document update is being correctly added to the index.

If the index is being built for the first time then each document will be added to the output as it is indexed.

Debug_FT_Stream

As with DEBUG_FTV_INDEX this parameter will output extra data to the server console when the index is being updated or rebuilt.

To enable this parameter it must be added to the NOTES.INI of the server and the server restarted. The parameter cannot be enabled using the 'set config' command.

The output will vary depending on how it is configured.

DEBUG_FT_STREAM=1

This debug level will just show each document that needs to be indexed being opened and closed. It can be used if there is a suspected corruption within the database affecting the Full Text Index.

Example output :

```
[0420:0002-284C] FTGetDocStream: INIT: Opened NoteID EEF6 in DB
C:\IBM\Domino\data\ftitest.nsf
[0420:0002-284C] FTGetDocStream: TERM: Finished NoteID EEF6 in DB
C:\IBM\Domino\data\ftitest.nsf
[0420:0002-284C] FTGetDocStream: INIT: Opened NoteID F4E6 in DB
C:\IBM\Domino\data\ftitest.nsf
[0420:0002-284C] FTGetDocStream: TERM: Finished NoteID F4E6 in DB
C:\IBM\Domino\data\ftitest.nsf
```

This output will only show documents that are new or have been updated.

DEBUG_FT_STREAM=3

As with level '1' this will output information for each document being indexed – it will then give further regarding each document.

Example output for a single document when creating an index :

```
[1D5C:0002-3A40] FTGetDocStream: INIT: Opened NoteID F4E6 in DB
C:\IBM\Domino\data\ftitest.nsf
[1D5C:0002-3A40] FTGetDocStream: Start item 'Form'; Datatype 500; UNK # 244
[1D5C:0002-3A40] FTGetDocStream: Start item 'From'; Datatype 500; UNK # 88
[1D5C:0002-3A40] FTGetDocStream: Start item 'AbbreviateFrom'; Datatype 500; UNK#
90
[1D5C:0002-3A40] FTGetDocStream: Start item 'AltFrom'; Datatype 500; UNK # 92
[1D5C:0002-3A40] FTGetDocStream: Start item 'AltLang'; Datatype 500; UNK # 77
[1D5C:0002-3A40] FTGetDocStream: Start item 'ThreadId'; Datatype 500; UNK # 74
[1D5C:0002-3A40] FTGetDocStream: Start item 'Remote_User'; Datatype 500; UNK #
127
[1D5C:0002-3A40] FTGetDocStream: Start item 'MainID'; Datatype 500; UNK # 89
[1D5C:0002-3A40] FTGetDocStream: Start item 'AbrFrom'; Datatype 500; UNK # 91
[1D5C:0002-3A40] FTGetDocStream: Start item 'WebCategories'; Datatype 500; UNK #
142
[1D5C:0002-3A40] FTGetDocStream: Start item 'Body'; Datatype 1; UNK # 82
[1D5C:0002-3A40] FTGetDocStream: Start item 'readers'; Datatype 500; UNK # 70
[1D5C:0002-3A40] FTGetDocStream: Start item 'NewsLetterSubject'; Datatype 500;
UNK # 75
[1D5C:0002-3A40] FTGetDocStream: Start item 'Path_Info'; Datatype 500; UNK # 141

[1D5C:0002-3A40] FTGetDocStream: Start item 'Subject'; Datatype 500; UNK # 79
[1D5C:0002-3A40] FTGetDocStream: Start item 'Categories'; Datatype 500; UNK # 81

[1D5C:0002-3A40] FTGetDocStream: Start item 'Abstract'; Datatype 500; UNK # 78
[1D5C:0002-3A40] FTGetDocStream: Start item '$UpdatedBy'; Datatype 501; UNK # 3
[1D5C:0002-3A40] FTGetDocStream: TERM: Finished NoteID F4E6 in DB
C:\IBM\Domino\data\ftitest.nsf
```

Here, displayed, is the NoteID of the document and the fields within the document being indexed.

This can be used to investigate if text in a field is being added to the index.

This output can be verbose and can affect the server performance, it would be recommended that this be disabled once testing is complete.

DEBUG_FT_STREAM=13

This is the most verbose of the debug options listed. Along with listing each document and field being indexed it will also give information on the amount of data per field being indexed and the text being added to the index..

Example output :

```
[33F4:0002-3528] FTGetDocStream: INIT: Opened NoteID F4E6 in DB
C:\IBM\Domino\data\ftitest.nsf
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: UNKNOWN: 261
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: UNKNOWN: 261
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: UNKNOWN: 263
[33F4:0002-3528] FTGetDocStream: Start item 'Form'; Datatype 500; UNK # 244
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 9 bytes
[33F4:0002-3528] 'MainTopic'
[33F4:0002-3528] FTGetDocStream: Start item 'From'; Datatype 500; UNK # 88
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 32 bytes
[33F4:0002-3528] 'CN=Test User/OU=Test/O=Test'
[33F4:0002-3528] FTGetDocStream: Start item 'AbbreviateFrom'; Datatype 500; UNK#
90
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 24 bytes
[33F4:0002-3528] 'Test User/Test/Test'
[33F4:0002-3528] FTGetDocStream: Start item 'AltFrom'; Datatype 500; UNK # 92
```

```

[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 32 bytes
[33F4:0002-3528] 'CN=Test User/OU=Test/O=Test'
[33F4:0002-3528] FTGetDocStream: Start item 'AltLang'; Datatype 500; UNK # 77
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 0 bytes''
[33F4:0002-3528] FTGetDocStream: Start item 'ThreadId'; Datatype 500; UNK # 74
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 11 bytes
[33F4:0002-3528] 'RSTN-9SKFGD'
[33F4:0002-3528] FTGetDocStream: Start item 'Remote_User'; Datatype 500; UNK #
127
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 0 bytes''
[33F4:0002-3528] FTGetDocStream: Start item 'MainID'; Datatype 500; UNK # 89
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 32 bytes
[33F4:0002-3528] 'E16F7C523AF9493880257DC7003F08FE'
[33F4:0002-3528] FTGetDocStream: Start item 'AbrFrom'; Datatype 500; UNK # 91
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 26 bytes
[33F4:0002-3528] 'TEST_USER__TEST__TEST'
[33F4:0002-3528] FTGetDocStream: Start item 'WebCategories'; Datatype 500; UNK #
142
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 0 bytes''
[33F4:0002-3528] FTGetDocStream: Start item 'Body'; Datatype 1; UNK # 82
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 13 bytes
[33F4:0002-3528] 'test document'
[33F4:0002-3528] FTGetDocStream: Start item 'readers'; Datatype 500; UNK # 70
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 0 bytes''
[33F4:0002-3528] FTGetDocStream: Start item 'NewsLetterSubject'; Datatype 500;
UNK # 75
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 13 bytes
[33F4:0002-3528] 'test document'
[33F4:0002-3528] FTGetDocStream: Start item 'Path_Info'; Datatype 500; UNK # 141

[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 0 bytes''
[33F4:0002-3528] FTGetDocStream: Start item 'Subject'; Datatype 500; UNK # 79
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 13 bytes
[33F4:0002-3528] 'test document'
[33F4:0002-3528] FTGetDocStream: Start item 'Categories'; Datatype 500; UNK # 81

[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 0 bytes''
[33F4:0002-3528] FTGetDocStream: Start item 'Abstract'; Datatype 500; UNK # 78
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 13 bytes
[33F4:0002-3528] 'test document'
[33F4:0002-3528] FTGetDocStream: Start item '$UpdatedBy'; Datatype 501; UNK # 3
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: FIELD SEP
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: TEXT: 32 bytes
[33F4:0002-3528] 'CN=Test User /OU=Test/O=Test'
[33F4:0002-3528] FTGetDocStream: GETCHAR returns: EOF
[33F4:0002-3528] FTGetDocStream: TERM: Finished NoteID F4E6 in DB
C:\IBM\Domino\data\ftitest.nsf

```

From this it can be seen that the text 'test document' is being added to the index.

Chapter 7 - Common Issues with Full Text Indexing and Search

Here are some of the common issues seen when using a Full Text Index with possible solutions.

Issues discussed :

- 7.1) Full Text Index not being created
- 7.2) Full Text Index is present – but database is showing as Not Indexed
- 7.3 Full Text Index is not present – but database is showing as Indexed
- 7.4) Full Text Index is not being updated
- 7.5) Search term not found
- 7.6) Text in attachments not being found
- 7.7) Result document slow to open

7.1) Full Text Index not being created

In this situation there are a number of possible causes, the first things to check are :

- Can the server or Notes Client create Full Text Indexes for other databases?

If it can then the issue is likely just with the single database.

- a) Check that the ID (either the user's ID or the server's ID has Manager rights defined in the ACL for the database.
- b) Take a backup of the database and then run maintenance on it (Fixup, Updall & Compact), monitor if there are any errors displayed that would show corruption in the database.
- c) If there is a replica of the database available check to see if it can be Full Text Indexed, if it can then replace the problem database from the replica – keeping a backup.
- d) Confirm if there are any documents in the database. If there are no documents present then the Full Text Index will not be generated.
- e) Enable the “DEBUG_FTV_INDEX=15” debug, this may highlight if the index build is stopping on one problem document that may need to be investigated.

If it can't then the issue may be :

- a) “UPDATE_NO_FULLTEXT=1” is enabled – this will prevent the creation of any new Full Text Indexes and the update of any existing ones. This should be set to '0' in the NOTES.INI and the Server or Notes Client restarted.
- b) The Update task was not loaded on the Server. This task is required to be loaded in the ServerTasks line of the NOTES.INI for the Indexer to be available, this should be added to the NOTES.INI and the server restarted.
- c) The server itself may have some damaged files, re-installing the server could resolve the issue, full backups of all critical data should be taken before trying this.

7.2) Full Text Index is present – but database is showing as Not Indexed

In most cases like this removing the Full Text Index by deleting the .FT folder for the database and then rebuilding it resolves the situation.

7.3 Full Text Index is not present – but database is showing as Indexed

This can happen if the Full Text Index is deleted by a process outside of the Server or a Notes Client. The correct way that an Index should be deleted is by accessing the Database properties, selecting the Full Text tab and clicking on the Delete Index button.

Rebuilding the index or running the command “load updall -x <database path and file name>” will recreate the Full Text Index for use.

If the Full Text Index is not actually wanted, the Full Text Index needs to be recreated and then removed correctly. This will set the Database to 'Not Indexed’.

7.4) Full Text Index is not being updated

In this situation there is an existing Full Text Index – but it is out of date.

To resolve this try :

- a) Run "load updall -x <database path and file name>" on the database.
- b) confirm if the Full Text Index for other databases on the server are being updated, if not then review the NOTES.INI for "Update_No_Fulltext=1" and check to see the access level for the server in the ACL of the database.
- c) Delete and rebuild the Full Text Index.
- d) Enable the "DEBUG_FTV_INDEX=15" debug, this may highlight if the index build is stopping on one problem document that may need to be investigated.
- e) Run maintenance on the database (Fixup, Compact and Updall).

Note : When the Full Text Index is set to update "Immediately" it may still be some time before the Index is updated. The update requests are all queued together and then the Update task runs through them on a first-come-first-served order. This is to prevent the server being overloaded by Index update requests every time a document is created, deleted or updated.

7.5) Search term not found

If the search term is not being found :

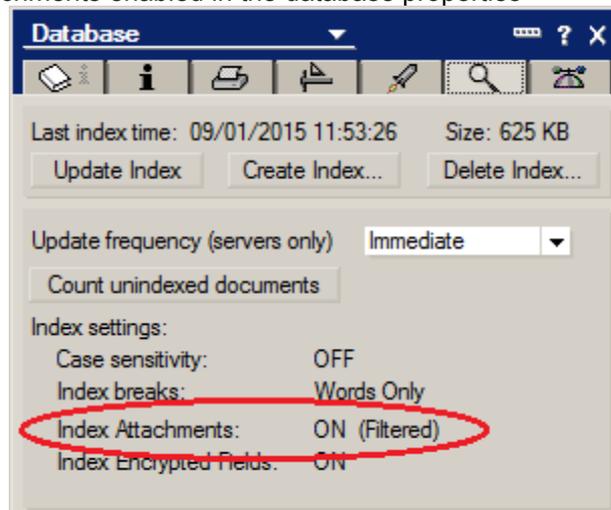
- a) Confirm that the search term is actually in a document and that there are no documents not included in the index.
- b) Rebuild the Full Text Index
- c) Copy text from a document in the database and search for it – to confirm that the index is actually working.
- d) Confirm that the search term is correctly spelled.
- e) Confirm if the full word is being searched for, not a partial of the word without a wildcard.
- f) Enable DEBUG_FTV_SEARCH=1 to confirm if the correct search term is being used by the search utility and if any of the results are being removed – for example they may not be being displayed due to Reader fields.

7.6) Text in attachments not being found

In this situation text which is known to be in attachments is not being found by the search. To work through this scenario :

- a) Reviewing the database properties on the Full Text tab confirm that attachments are included in the index either filtered or not :

Figure 7-1 : Indexing of attachments enabled in the database properties



- b) Confirm that the search term is correct and present in the attachment. Try searching for other complete words from the attachment.
- c) Rebuild the Full Text Index.
- d) Confirm that the text in the attachment is actually text. If words are included in a graphic then the indexer is not able to parse the text from them. A test for this that works in most, but not all cases, if you can open the attachment, highlight the text, copy it and then paste it into NotePad then it should be in the index.
- e) Confirm that the attachment is not one of the type automatically skipped over by the indexer :

.au, .cca, .dbd, .dll, .exe, .gif, .img, .jpg, .mp3, .mpg, .mov, .nsf, .ntf, .p7m, .p7s, .pag, .sys, .tar, .tif, .wav, .wpl, .zip.

or that it is excluded by adding it to the parameter :

FT_INDEX_ATTACHMENTS=value

f) Confirm that the indexing of attachments is not disabled for the server with the setting :

FT_Index_Attachments=0

7.7) Result document slow to open

As mentioned in the section above (5.8-Highlighting) there are situations where search results can be slow to open. Opening the document normally, when not in search results, is not slow. This can be caused by the highlighter parsing through attachments in the document to see if the attachment should be highlighted. To prevent this enable the parameter in the Server's NOTES.INI and restart the server :

FT_LIMIT_HIGHLIGHT_FILTER=1

This setting prevents the highlighter from parsing through documents that would not contain the search text. It will result in the attachment not being highlighted when the document is opened even if the search term is present and found in it.

About the author :

Robert Steen joined IBM in 1998, and in 2003 he joined the Customer Support team, since which time he has been working with the Server Core team, assisting numerous customers from all over the world.

He was the author for : Domino Replication Fundamentals and Troubleshooting

<http://www-01.ibm.com/support/docview.wss?uid=swg21693949&aid=1>

He was a co-author for : Archiving and recovering IBM Lotus DAOS-enabled databases

http://www-10.lotus.com/ldd/dominowiki.nsf/dx/Archiving_and_recovering_IBM_Lotus_DAOS-enabled_databases