Together Search Server Administration Guide

Table of Contents

List of Figures

List of Tables

Chapter 1. 'Together Search Server' Administration/Indexing

Login

Accessing the administration for the first time displays the logging dialog. By default the username is “admin” and password is “enhydra”.

Administrative part of the application consists of several modules:

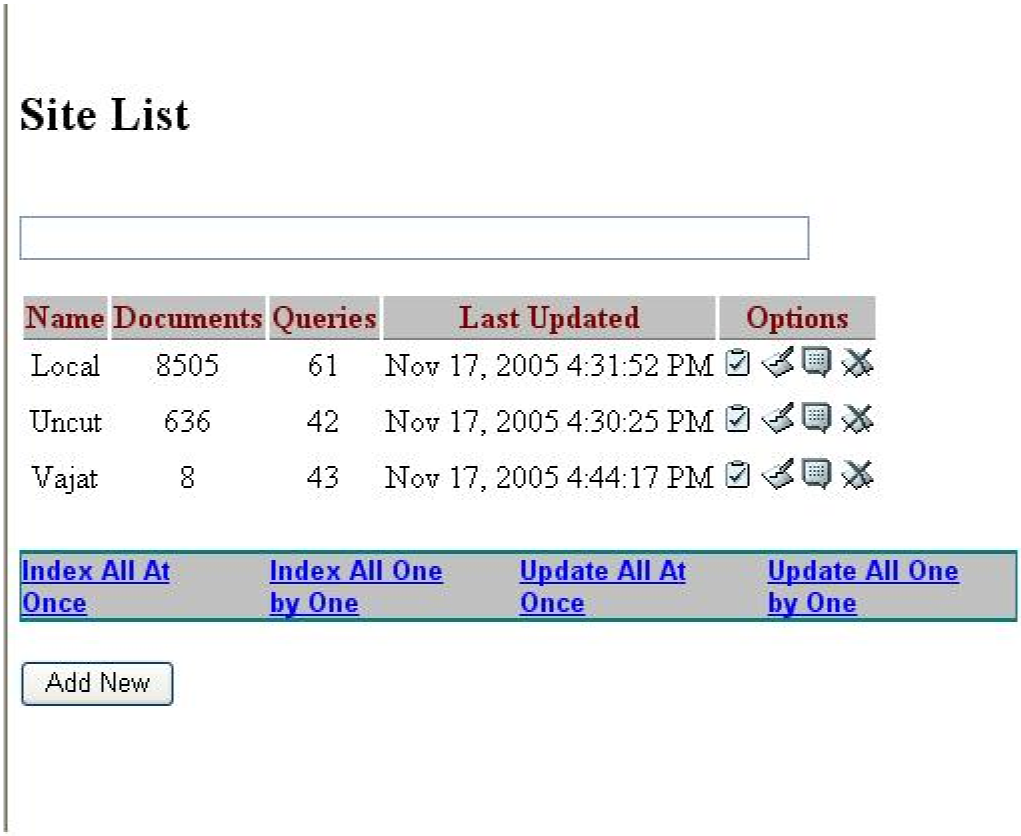
- *LIST* - site management, site settings, indexing, updating, optimizing.
- *PATHS* - site paths management
- *DOCUMENT GROUP - document groups management*
- *FILE TYPES* – file types management
- *STATUS* – site index/update/optimize status
- *HISTORY* – site index/update/optimize history

Site List Management

Selecting the LIST button in the Site bar displays the Site List page (module) in the bottom frame. Initially, the list of sites is displayed. The following Site management operations are available on this page:

- *list of sites*
- *add site*
- *remove site*
- *index*
- *update*
- *optimize*
- *index all*
- *update all*
- *optimize all*

Figure 1.1. List of sites with available operations



Site List

Name	Documents	Queries	Last Updated	Options
Local	8505	61	Nov 17, 2005 4:31:52 PM	☑ ✎ 🖵 ✕
Uncut	636	42	Nov 17, 2005 4:30:25 PM	☑ ✎ 🖵 ✕
Vajat	8	43	Nov 17, 2005 4:44:17 PM	☑ ✎ 🖵 ✕

The following information can be obtained from the List of Sites:

- *Name* - site name
- *Documents* - number of documents indexed in the site
- *Queries* - number of queries (searches) made to this site
- *Last Updated* - the time of last indexing/updating/optimizing of site index
- *Options* - site management operations

Operations that can be performed over an existing site are displayed in the Options column. Each operation is represented by a button:

Table 1.1. Buttons in Site List Management

☑	*Index*, site indexing
🧹	*Update*, update site’s index
💬	*Edit/View* site settings
✗	*Delete Site*, deleting site
🛠	*Optimize*, optimizing site's index

The Add New button is displayed on the bottom of the page:

Clicking the Index All button indexes all Sites. There is a choice between indexing all Sites at once, or one-by-one (after one is finished the next one is started).

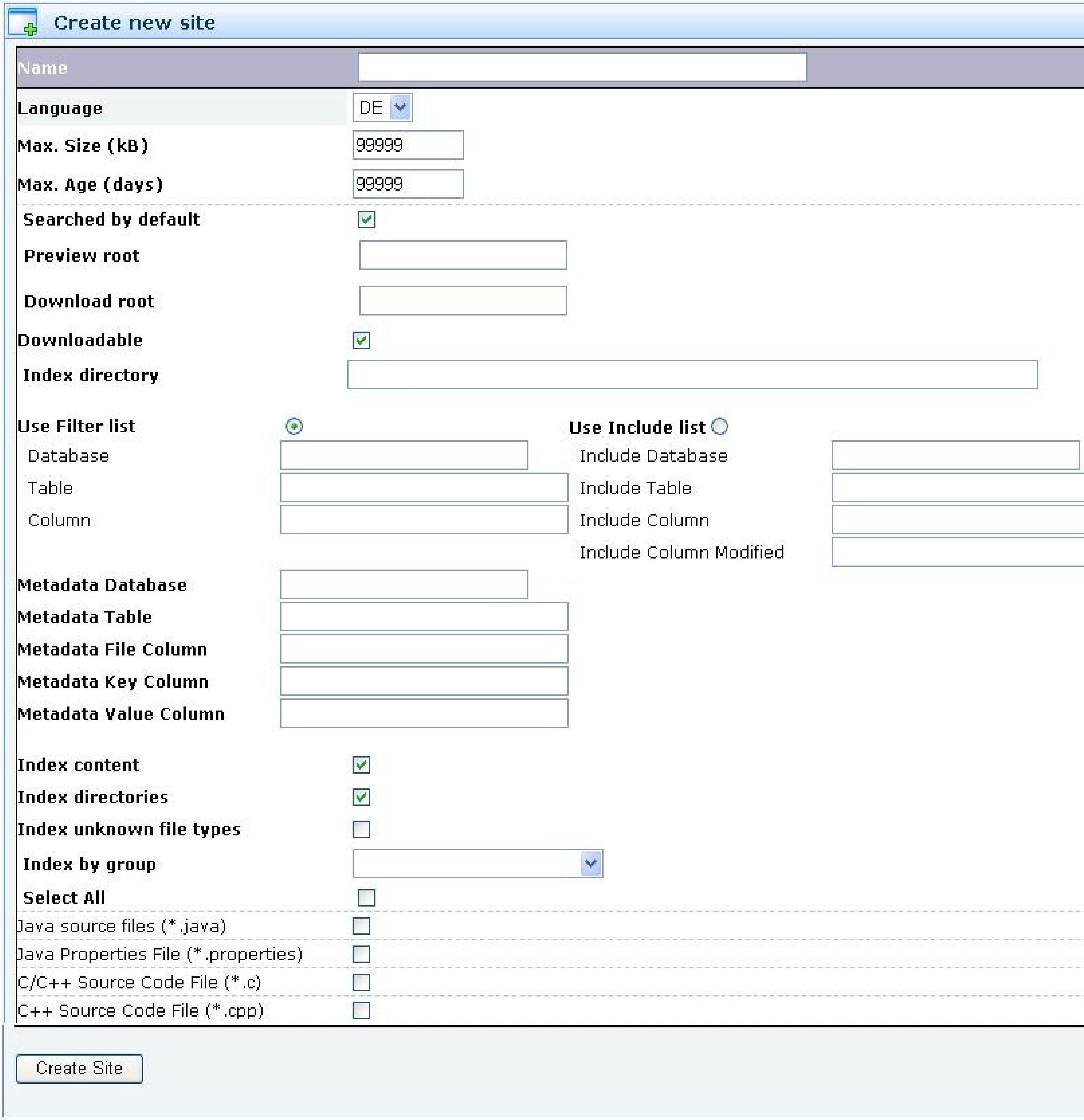
Clicking the Update All button updates all Sites. There is a choice between updating all Sites at once, or one-by-one (after one is finished the next one is started).

Clicking the Optimize All button optimizing all Sites. There is a choice between optimizing all Sites at once, or one-by-one (after one is finished the next one is started).

Adding a site

Clicking the Add New button displays the Create New Site (Site Creation) page.

Figure 1.2. Site creation page



Create new site

Name

Language DE

Max. Size (kB) 99999

Max. Age (days) 99999

Searched by default ☑

Preview root

Download root

Downloadable ☑

Index directory

Use Filter list ●

Database

Table

Column

Use Include list ○

Include Database

Include Table

Include Column

Include Column Modified

Metadata Database

Metadata Table

Metadata File Column

Metadata Key Column

Metadata Value Column

Index content ☑

Index directories ☑

Index unknown file types ☐

Index by group

Select All ☐

Java source files (*.java) ☐

Java Properties File (*.properties) ☐

C/C++ Source Code File (*.c) ☐

C++ Source Code File (*.cpp) ☐

Site Attributes

The following attributes are listed on the Site Creation page. The following fields are mandatory in order to properly setup a new site:

- *Name* – Site name. Must be unique (name per site).
- *Language* – Site language. Used for analyzing and indexing documents
- *Max. Size (kb)* – Maximal size of a file to be indexed in thousands of bytes. If a file is larger than the value specified in this filed, it won’t be indexed.
- *Max. Age (days)* - Maximal age of a file to be indexed in days. If a file is older than the value specified in this filed, it won’t be indexed.
- Index directory – location of index directory

The rest fields are not mandatory.

- Search by default – checked automatically includes the site in search
- Preview root – alternative path to preview application
- Download root– alternative path to download application
- Downloadable – checked allows for downloading of sites files
- Filter database – database where filtered files are stored
- Filter table - table where filtered files are stored
- Filter column - column where filtered files are stored
- Include database – database where included files are stored
- Include table - table where included files are stored
- Include column - column where included files are stored
- Include column modified – document modification time column
- Index content – should file content be indexed or just index file name and modification time.
- Index directories – directrories will be index.
- *Index unknown file types* – all other file types. These files are not parsed for contents but rather indexed only using their file name and modification time.
- Index by group – Choose predefined file group

Filtering files

If the optional attributes Filter database, Filter table, and Filter column were entered for a Site, during indexing and updating files to be indexed will be filtered according to the criteria (file paths) entered in the Filter column.

Filter database is a logical database name. Filter table is a table within the database. Filter column is a column within the table. This is the column where the actual files (file paths) to be filtered are stored.

In order to use the database, a user/administrator should configure ("filterDB”) in

<together-application-server-runtime-7.0 root>/multiserver/webapps/tssAdmin/WEB-INF/ web.xml

<together-application-server-runtime-7.0 root>/multiserver/conf/catalina/localhost/tssAdmin.xml

Include list

If the optional attributes Include database, Include table, Include Column and Include Column Modified were entered for a Site, during indexing and updating files to be indexed will be indexed according to an include list in an external database.

Include database is a logical database name. *Include table* is a table within the database. *Include column* is a column within the table containing full file path. This is the column where the actual files (file paths) to be indexed are stored. *Metadata key column* is a column within the table where the file property keys are stored. *Metadata value column* is a column within the table where the file property values are stored.

In order to use the Include List feature of 'Together Search Server' the following steps are to be performed.

Create Include database

The first step is to create an include list database. In the database a table should be created with following columns in the same order as written here. The column names are *arbitrary* and are given here only as an example:

OID – int, object ID

FILENAME – Varchar(254), file path

MODIFIED, Timestamp, file modification date

Insert data

After the successful database/table creation, the include list should be filled out the file paths to the files to be indexed along with their modification dates.

An example:

Table 1.2. Insert data example

Attribute	FILENAME	MODIFIED
1	'C:\myfile.xls'	'2005-01-01 00:00:00.0'

Setting up 'Together Search Server Admin' application

The next step is to configure the 'Together Search Server Admin' application to have access to our Include List.

The datasource for the database should be declared in the tssAdmin.xml. Here is an example of a datasource declaration:

```
<Resource name="includeDB" type="javax.sql.DataSource"
maxWait="5000" maxActive="40" password="" maxIdle="10" username="sa"
    driverClassName="org.hsqldb.jdbcDriver"
    url="jdbc:hsqldb:${catalina.base}/hsql/ includeDB / includeDB" />
```

Parameters:

`Resource name` is the datasource name for the database that will be used throughout the application.

`driverClassName` is the name of the driver class used

`url` is the url to the database

Setting up a Site and Include list

An Include List for a Site is defined in configuretion file ('siteConf.xml'). The information about the Include list should be entered:

NAME: is the resource name defined in tssAdmin.xml

INCLUDETABLE: table where the include list data are contained

INCLUDECOLUMN: column in the table where the file paths are contained

INCLUDECOLUMNMODIFIED: column in the table where the file modification dates are contained.

Configuring Include list on a different database server

In order to run the include list on a database server other than HSQL, the following modification should be performed:

In the **tssAdmin.xml** the proper driver and url should be entered Supported values: HypersonicSQL, PostgreSQL, DB2, QED, MySQL, Sybase, Oracle, MSQL.

Example for MSQL:

```
<Resource name="includeDB" type="javax.sql.DataSource"
     maxWait="5000" maxActive="40" password="" maxIdle="10" username="sa"
          driverClassName="com.microsoft.jdbc.sqlserver.SQLServerDriver"
     url="jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=includeDB;Select
Method=cursor" />
```

Metadata

If the optional attributes Metadata database, Metadata table, Metadata File Column, Metadata Key Column and Metadata Value Column were entered for a Site, during indexing and updating files addition information about each file whose reference exists in Metadata database will be indexed, and at the end of the index/update process if the predefined file type 'NULL' is selected to be indexed, Meta Data addition information will be indexed in case that file path of the file from 'Metadata File Column' don`t exists on file system and not have been indexed/updated yet.

If the parameter LogicalNameFromDatabase was set to “1” in the application configuration file, the indexing/update mechanism will try to locate the DocumentLogicalName parameter (also defined as a string value in the configuration file) in the metadata’s key column for the file being indexed. If found the document title is obtained from metadata’s value column corresponding to that key. This value will be then indexed along with other attributes.

If the LogicalNameFromDatabase parameter was set to “0” or no DocumentLogicalName was found among keys in the metadata database, the default mechanism for extracting document title is used.

If parameter 'DocumentUpdate' in the application configuration file was found among keys in the metadata database, modification time is obtained from metadata’s value column corresponding to that key. If the key is not found current time will be used as modification time.

Parameter 'DocumentUpdatePattern' in the application configuration file is used to describe the pattern of the modification time in the metadata’s value column.

Metadata database is a logical database name. *Metadata table* is a table within the database. *Metadata file column* is a column within the table. This is the column where the actual files (file paths) to be indexed are stored.

In order to use the Metadata feature of 'Together Search Server' the following steps are to be performed.

Create Metadata database

The first step is to create a metadata database. In the database a table should be created with following columns in the same order as written here. The column names are *arbitrary* and are given here only as an example:

OID - int, object ID

FILENAME - Varchar(254), file path

KEY - Varchar(254), metadata key name

VALUE - Varchar(254), metadata value for the given key

Insert data

After the successful database/table creation, the metadata table should be filled out with file paths to the files to be indexed along with their key and value pairs.

An example:

Table 1.3. Insert data example

OID	FILENAME	KEY	VALUE
1	'C:\myfile.xls'	‘myKey’	'myValue'

Setting up 'Together Search Server Admin' application

The next step is to configure the 'Together Search Server Admin' application to have access to Metadata database.

The datasource for the database should be declared in the **tssAdmin.xml**. Here is an example of a datasource declaration:

```
<Resource name="metaDB" type="javax.sql.DataSource"
  maxWait="5000" maxActive="40" password="" maxIdle="10" username="sa"
  driverClassName="org.hsqldb.jdbcDriver"
  url="jdbc:hsqldb:${catalina.base}/hsql/ metaDB / metaDB" />
```

Parameters:

`Resource name` is the datasource name for the database that will be used throughout the application.

`driverClassName` is the name of the driver class used

`url` is the url to the database

Setting up a Site and Metadata database

A Metadata database for a Site is defined in configuretion file ('siteConf.xml'). The information about the MetaData list should be entered:

NAME: is the resource name defined in tssAdmin.xml

METATABLE: table where the metadata are stored

METAFILECOLUMN: column in the table where the file paths are stored

METAKEYCOLUMN: column in the table where property key are stored

METAVALUECOLUMN: column in the table where property values are stored

Configuring Metadata list on a different database server

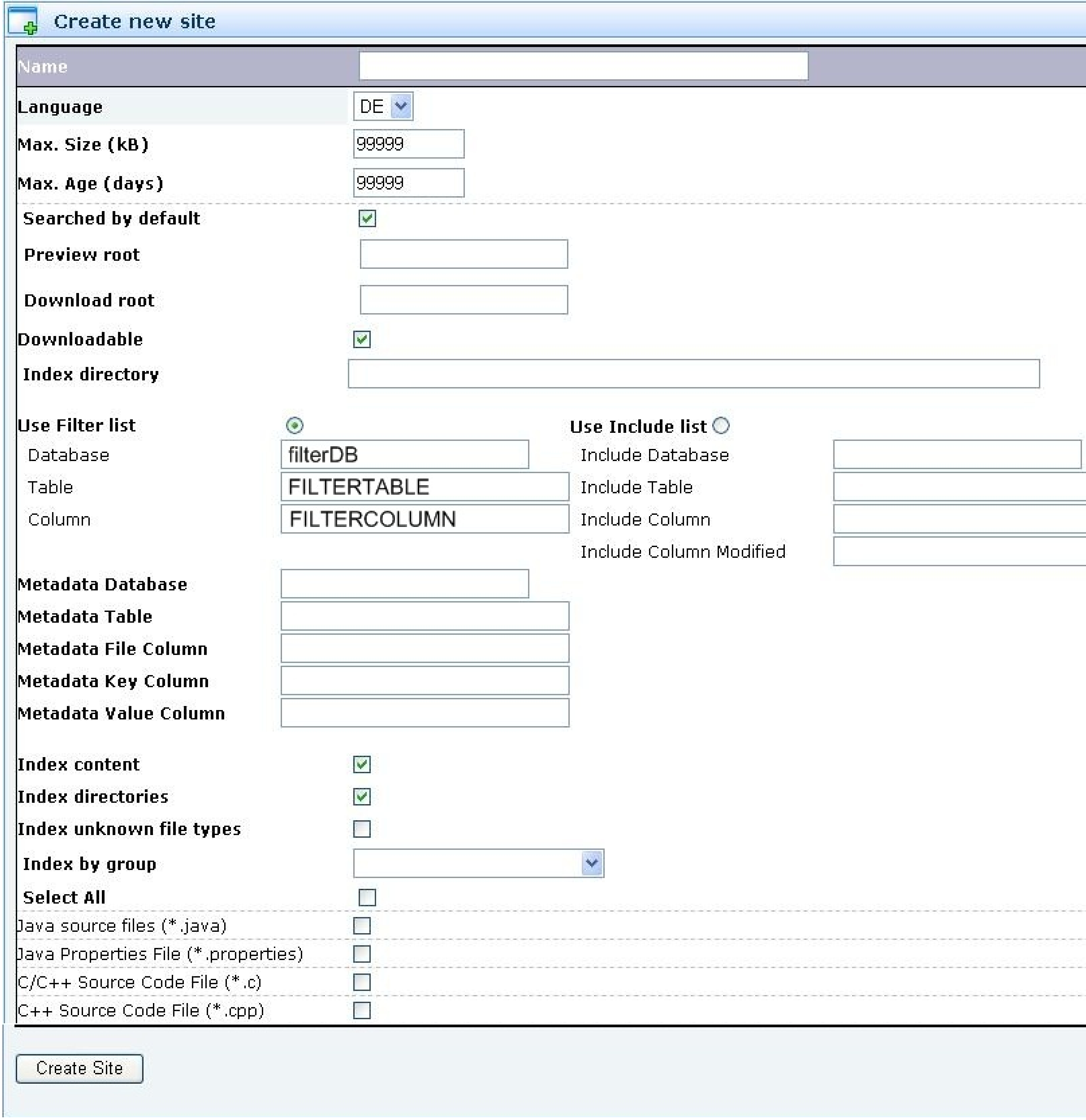
In order to run the metadata list on a database server other than HSQL, the following modification should be performed:

In the **tssAdmin.xml** the proper driver and url should be entered Supported values: HypersonicSQL, PostgreSQL, DB2, QED, MySQL, Sybase, Oracle, MSQL.

Example for MSQL:

```
<Resource name="metaDB" type="javax.sql.DataSource"
   maxWait="5000" maxActive="40" password="" maxIdle="10" username="sa"
   driverClassName="com.microsoft.jdbc.sqlserver.SQLServerDriver"
   url="jdbc:microsoft:sqlserver://localhost:1433;DatabaseName= metaDB;SelectMethod=cursor"/>
```

An example of selecting values on this screen for



Create new site

Name
Language: DE
Max. Size (kB): 99999
Max. Age (days): 99999
Searched by default ☑
Preview root
Download root
Downloadable ☑
Index directory

Use Filter list ●
Database: filterDB
Table: FILTERTABLE
Column: FILTERCOLUMN

Use Include list ○
Include Database
Include Table
Include Column
Include Column Modified

Metadata Database
Metadata Table
Metadata File Column
Metadata Key Column
Metadata Value Column

Index content ☑
Index directories ☑
Index unknown file types ☐
Index by group
Select All ☐
Java source files (*.java) ☐
Java Properties File (*.properties) ☐
C/C++ Source Code File (*.c) ☐
C++ Source Code File (*.cpp) ☐

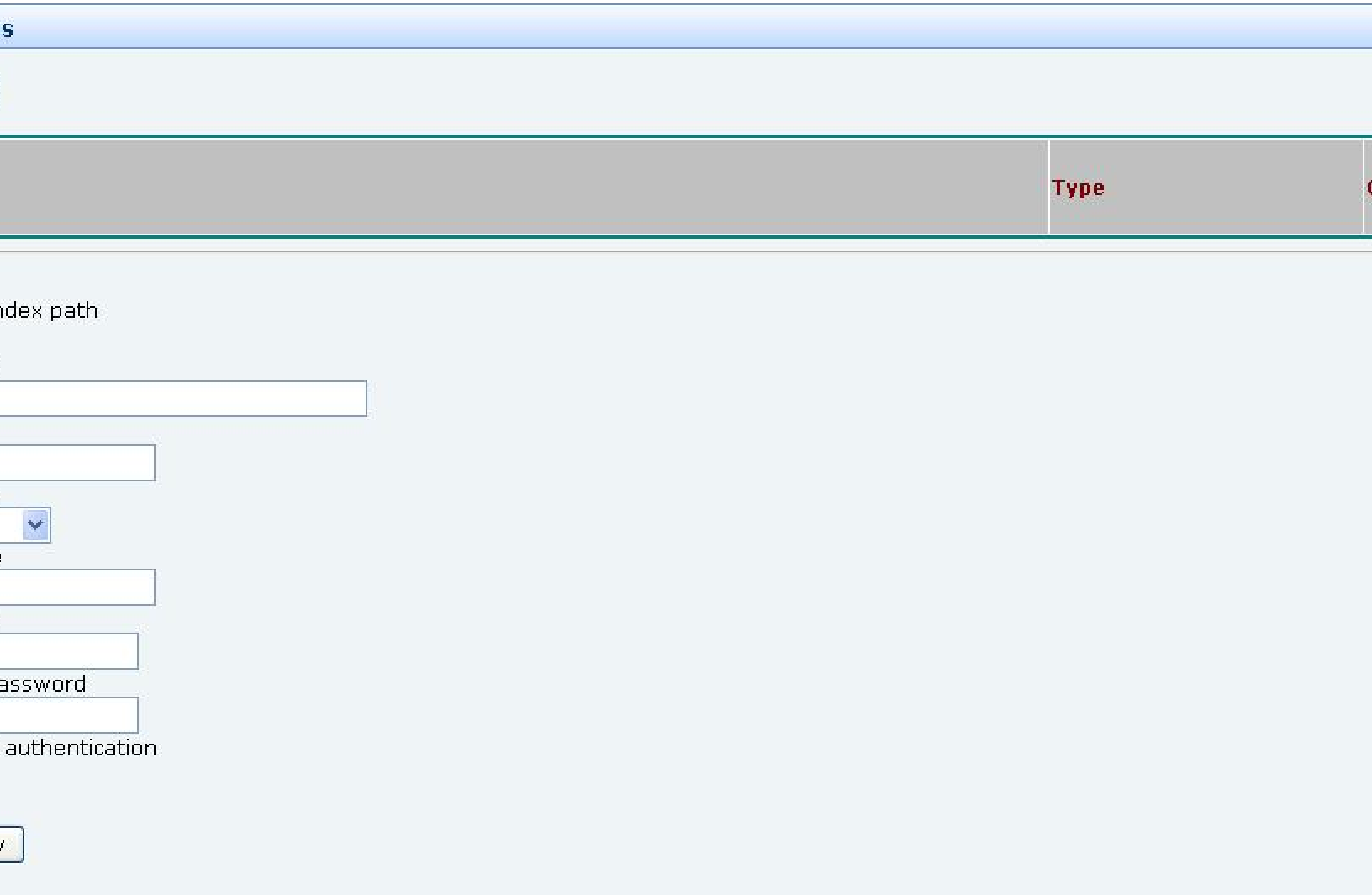
After the fields have been filled and file types selected, a new site can be created by clicking the Create New button -

If a site was successfully created the Site List screen is displayed.

Defining paths for a site

Selecting the PATHS button in the Site Management bar displays the Path List page (module) in the central frame. Paths for a selected (combo box) site are listed on this screen, as well as fields for defining a new path for a selected site.

Figure 1.4. List of paths and new path creation



The following *Path* management operations are available on this page:

- *Site combo box* – selected a site and list its defined paths

- *Add New* – add a new path for a selected site
- *Delete* – delete a path

Adding a new path

Paths can be defined either on the local File System or UNC storage.

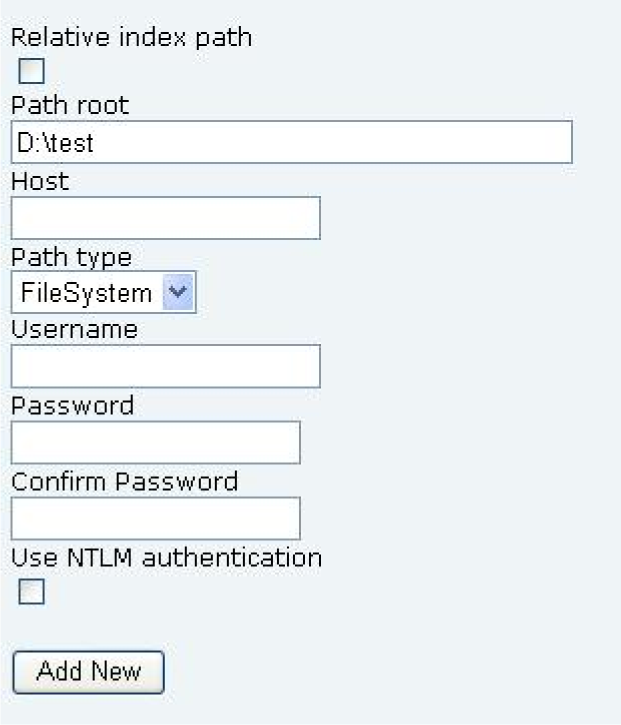
Adding a new path – File System

An example of adding a new path linked to the File System *of the computer on which the application runs*. We are going to add new path for our *Demo* site:

1. Select the *Demo* site from the combo box
2. In the *Path Root* field we are going to type in the location where the files we want to index are located. In this example this is: “d:\test”
3. In the *Path Type* combo box select *File System*

The screen should look something like this:

Figure 1.5. Defining a new path – File System



Relative index path
☐
Path root
D:\test
Host
Path type
FileSystem
Username
Password
Confirm Password
Use NTLM authentication
☐

Adding a new path – UNC

In an example of adding a new path linked to the *UNC* we are going to add new path for our *Demo* site:

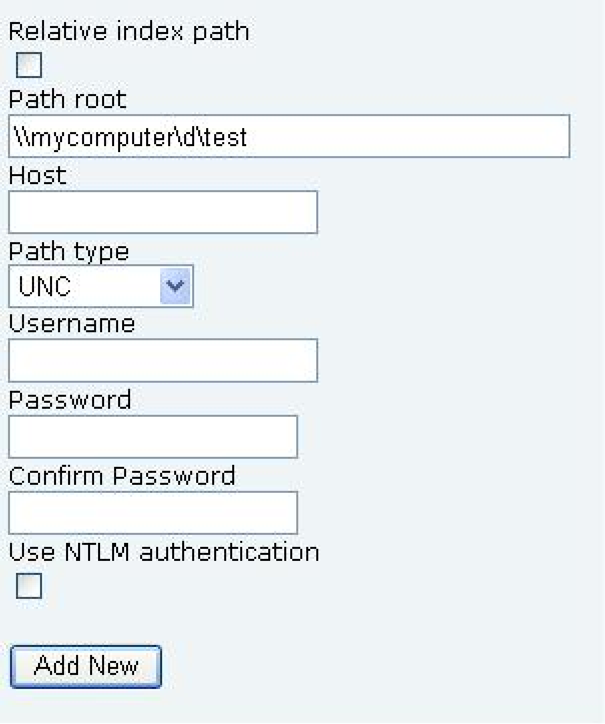
1. Select the *Demo* site from the combo box

2. In the *Path Root* field we are going to type in the location where the files we want to index are located. In this example this is: “\\mycomputer\d\test”

3. In the *Path Type* combo box select *UNC*

The screen should look something like this:

Figure 1.6. Defining a new path – UNC



Relative index path
☐
Path root
\\mycomputer\d\test
Host

Path type
UNC
Username

Password

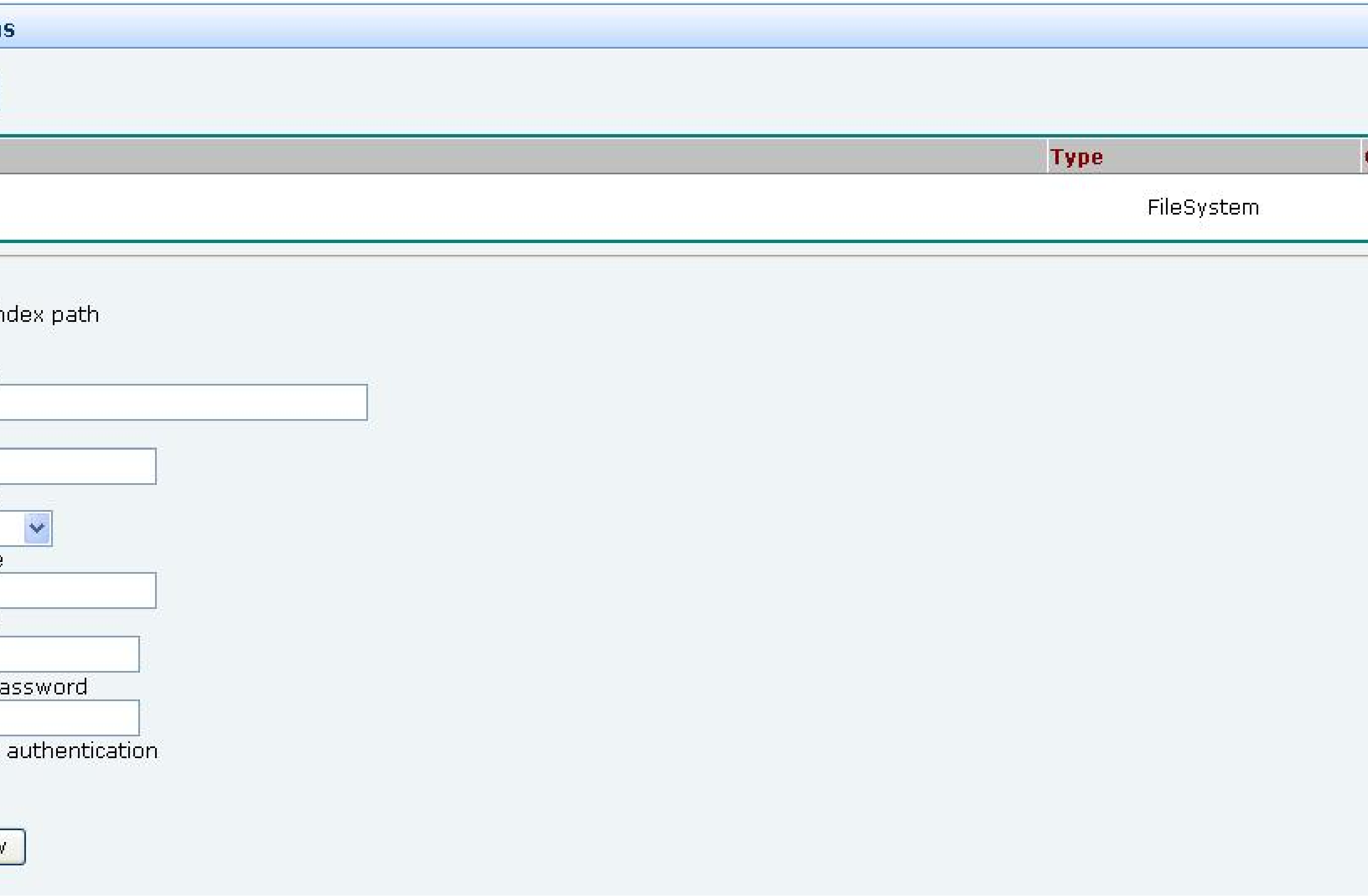
Confirm Password

Use NTLM authentication
☐

Add path

Clicking the *Add New* button adds the new button to the site. After the path has been added it appears in the Site’s path list:

Figure 1.7. List of paths for a selected site



Deleting a path

Clicking the Delete button

associated with the path on the Path List screen, deletes that path from the Site selected.

Site Indexing

Two mode of indexing are present. Normal and Re-Index mode.

In Normal mode every click on index button will start index process from the beginning.

The main functionality of Re-Index mode is to continue with index process from last indexed file. According to this, index process will create ‘txt’ file (‘include.txt’ or ‘path.txt’, that is placed in index directory of the ‘Site’, together with index files) where will be logged last indexed ‘directory/file’.

After index process is finished, the ‘txt’ file will be deleted. According to this new index process will be started from beginning in Re-Index Mode.

Mode of Index, that application will start is defined in ‘web.xml’ file. Parameter is ‘Indexer.ReIndexMode’ and default value is ‘true’.

Clicking the Indexing button

Site Index Updating

Clicking the Update button

on the Site List screen starts the updating of a selected site.

Site Index Optimizing

Clicking the Optimize button

on the Site List screen starts the optimizing of a selected site.

Index/Update/Optimize all

Clicking the *Index All* button indexes all Sites. There is a choice between indexing all Sites at once, or one-by-one (after one is finished the next one is started).

Clicking the *Update All* button updates all Sites. There is a choice between updating all Sites at once, or one-by-one (after one is finished the next one is started).

Clicking the *Optimize All* button optimizes all Sites. There is a choice between optimizing all Sites at once, or one-by-one (after one is finished the next one is started).

Edit a Site

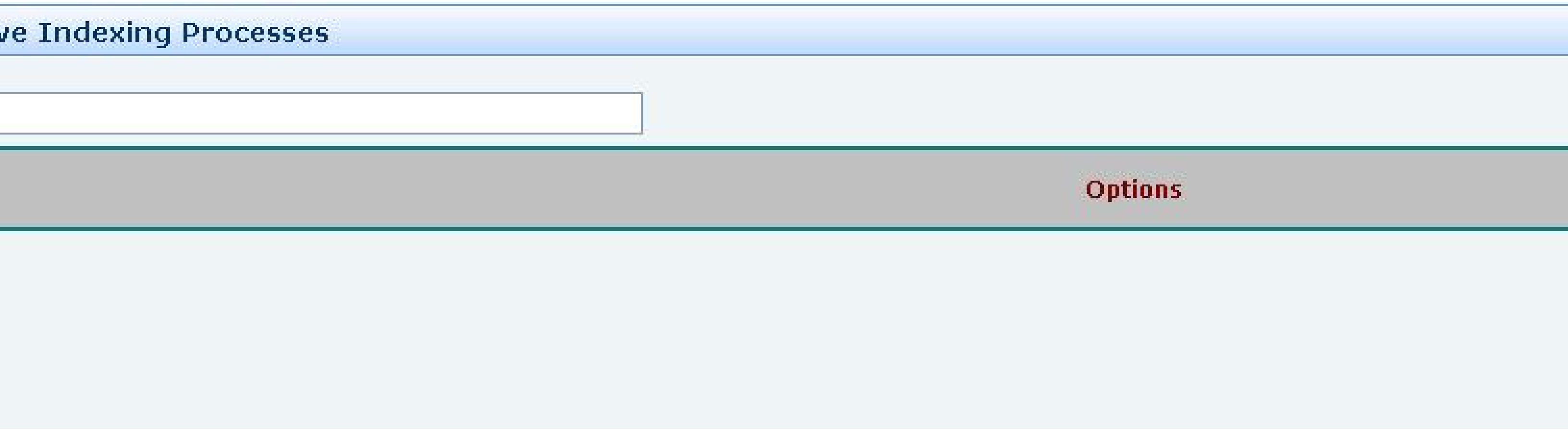
Clicking the Edit Site button

on the Site List screen displays the Site page. This page resembles the Site page. All attributes are explained in the Create Site section.

Indexing Status

Selecting the Status option on the Administration menu displays the active/started indexing processes. The screen looks like this:

Figure 1.8. Indexing Status



Indexing History

Clicking the History menu displays the history of sites indexed with the following information:

Site – site name

Document No. – number of documents in the index

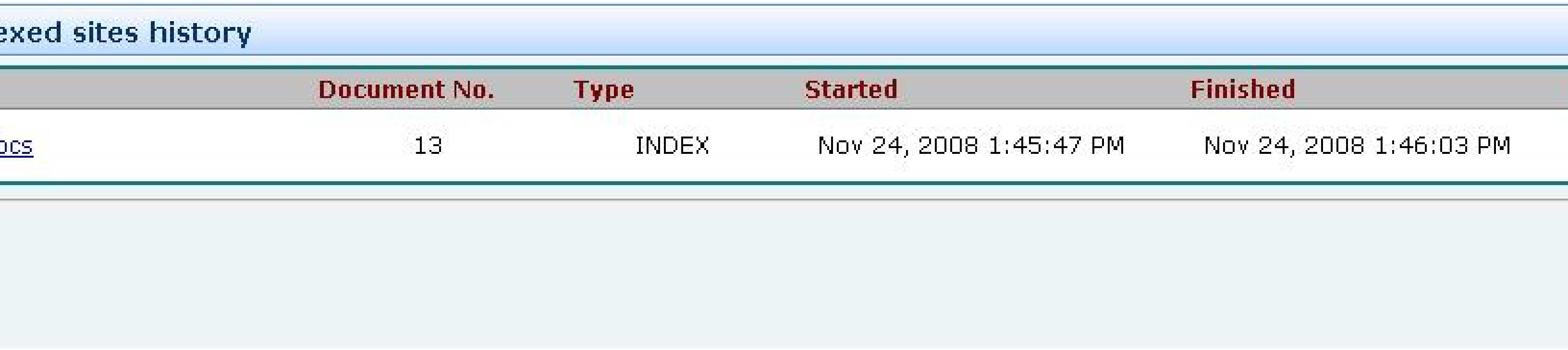
Type – operation type: indexing or update

Started – operation start time

Finished – operation finish time

Length – length of the operation

Figure 1.9. Sites indexing history



	Document No.	Type	Started	Finished
ocs	13	INDEX	Nov 24, 2008 1:45:47 PM	Nov 24, 2008 1:46:03 PM

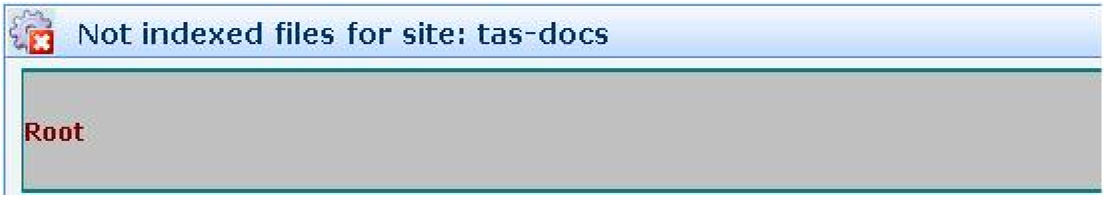
Clicking the Clear button clears the history.

Clicking the Site name displays the page showing all not indexed files during the operation (if any).

Not indexed files

In case a file was not successfully indexed, it appears on this page displaying its full path with the possibility to download the file.

Figure 1.10. Not Indexed Files



Not indexed files for site: tas-docs

Root