OCR and Fileindexing

Constellio

Constellio search in Tempus Serva installations

Activate the search servlet in your installation

The search servlet is deactivated by default.

1. Edit the <tomcat>/webapps/<Tempus Serva>/WEB-INF/web.xml
2. Remove comments from the search servlet
3. Remove comments from the search filter

If you are using web container security please remove it from the search servlet: The servlet filter will handle authentication of crawler robots using a specialized form of basic authentication (normal users will be redirected to the main servlet instead).

Option: Create a user for crawling

You will need at least 1 user for crawling the content in Tempus Serva, possibly more if content restrictions apply to different search user groups.

The following applies to crawling users

- All group and policies will be respected through the indexing
- No codeunits will be activated
- No log entries will be created

You can test what the crawler will se by logging in with an extra parameter:

```
/login?SearchIndexing=true
```

Prepare Constellio

Install Constellio

1. Download the 1.3 installer
2. Run the installer by doubleclicking the .jar file
3. Install to MySQL database
4. Run the **Start constellio**

Setting up a connector

Before setting up a connector create or choose a valid search scope

1. Choose connector type: **auth-http-connector**
2. Ensure that **Use security** is checked
3. Set start URL to: **http://<server name>/TempusServa/search**
4. Include the same URL in include patterns
5. Enter username for the crawler user (a valid TS user)
6. Enter password for the crawler user (a valid TS user)
7. After submitting the new connector, crawling/indexing will start by itself

No further actions are needed:

The search servlet will automatically redirect real users after they click on a search result.

Option: Tweak search results

The search servlet will automatically deliver content in a crude form, without any extra html such as wrappers. It will also provide the crawler with information about when it was last updated, and document Title will be se to current records Resume value.

You might consider excluding the **command=list** pages for better (less redundant) search results.

Elastic search

Adding OCR capability

OCR components must be installed on the same server as TS file indexing service.

Only GhostScript and Terrasect are required to proces PDF files.

TEMPORARY FIX: <tomcat>\catalina\catalina.properties add java.io.tmpdir=c:/Temp

Install: ImageMagick binaries

Download and unpack "portable" version (recommended c:\ImageMagick)

```
https://www.imagemagick.org/script/binary-releases.php
```

Register the location of the **convert** executeable in web.xml

```
<context-param>
   <param-name>ExecutableImageMagick</param-name>
   <param-value>c:\ImageMagick\convert</param-value>
</context-param>
```

Leaving the entry empty will prevent OCR handling of image files: png, jpg, jpeg

Install: Ghostscript binaries

Download and run installer

```
http://www.ghostscript.com/download/gsdnld.html
```

Note: You are not required to buy a license

Register the location of the **gswin64c** executeable in web.xml

```
<context-param>
   <param-name>ExecutableGhostscript</param-name>
   <param-value>c:\Program Files\gs\gs9.20\bin\gswin64c.exe</param-value>
</context-param>
```

Leaving the entry empty will prevent OCR handling of PDF files

Install: Tesseract binaries

For linux just use install from repository using

```
sudo yum install tesseract-ocr
```

If you are using Amazon linux please use this instead (thanks for help).

```
sudo yum --enablerepo=epel --disablerepo=amzn-main install libwebp
sudo yum --enablerepo=epel --disablerepo=amzn-main install tesseract
```

For Windows download installer or zip archieve

```
https://sourceforge.net/projects/tesseract-ocr-alt/files/
```

Register the location of the **tesseract** executeable in web.xml

```
<context-param>
   <param-name>ExecutableTerrasect</param-name>
   <param-value>c:\tesseract\tesseract</param-value>
</context-param>
```

Install

In order to index records and files you will need to complete these steps

1. Install standalone Elastic search server
2. Install and configure Tempus Serva file indexing
3. Configure the Tempus Serva installation

Finally you may want to install optional components to handle OCR (scanned PDF's and images)

Install Elastic search

Java 8 / Elastic search 6

This is the recommended version but requires Java 8.

Follow these steps:

```
sudo rpm --import https://packages.elastic.co/GPG-KEY-elasticsearch
sudo sh -c 'curl https://gist.githubusercontent.com/nl5887/b4a56bfd84501c2b2afb/raw/elasticsearch.repo >>
/etc/yum.repos.d/elasticsearch.repo'
sudo yum update -y
sudo yum install -y elasticsearch
sudo chkconfig elasticsearch on
```

The service runner configurations should have updated RAM allowance by adding an extra line in the file

```
sudo nano /etc/sysctl.conf
vm.max_map_count=262144
```

Restart service and validate settings were updated

```
sudo sysctl --system
sysctl vm.max_map_count
```

Run the daemon

```
sudo service elasticsearch start
```

Java 7 / Elastic search 1.7

This version is an alternate version.

Install and unpack files

```
sudo wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.6.tar.gz
tar -xvf elasticsearch-1.7.6.tar.gz
sudo rm elasticsearch-1.7.6.tar.gz
```

Run as a daemon

```
elasticsearch-1.7.6/bin/elasticsearch -d
```

Test that the service is running

```
curl 'http://localhost:9200/?pretty'
```

Handling crashes

ElasticSearch normally requires 1GB of memory, which is in the default memory configuation

```
sudo nano /etc/elasticsearch/jvm.options
```

Set the maximum memory entry to a lower value

```
-Xmx256m
```

Then restart the service

```
sudo service elasticsearch restart
```

Fixing: "curl: (7) Failed to connect to localhost port 9200: Connection refused"

In some cases the firewall needs to be configured

```
sudo iptables -I INPUT -p tcp --dport 9200 --syn -j ACCEPT
sudo iptables -I INPUT -p udp --dport 9200 -j ACCEPT
sudo iptables-save
```

Install using yum installer

```
sudo rpm --import https://packages.elastic.co/GPG-KEY-elasticsearch
sudo sh -c 'curl https://gist.githubusercontent.com/nl5887/b4a56bfd84501c2b2afb/raw/elasticsearch.repo >>
/etc/yum.repos.d/elasticsearch.repo'
sudo yum install -y elasticsearch
```

Alternative: Install with RPM

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.9.1-x86_64.rpm
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.9.1-x86_64.rpm.sha512
shasum -a 512 -c elasticsearch-8.9.1-x86_64.rpm.sha512
sudo rpm --install elasticsearch-8.9.1-x86_64.rpm
sudo rpm -e elasticsearch-8.9.1-x86_64.rpm
```

Install TS indexing service

Install war file

```
cd /usr/share/tomcat7/webapps/
sudo wget https://www.tempusserva.dk/install/tsFileIndexingService.war
```

A couple of seconds later you can configure he data connection and paths for OCR librarys

```
sudo nano /usr/share/tomcat7/conf/Catalina/localhost/tsFileIndexingService.xml
```

(or depending on Linux distribution)

```
sudo nano /etc/tomcat7/Catalina/localhost/tsFileIndexingService.xml
```

Example configurations can be seen below

Restart server after changes

```
tstomcatrestart
```

Windows example configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/tsFileIndexingService">

  <Resource name="jdbc/TempusServaLive" auth="Container" type="javax.sql.DataSource"
            maxActive="80" maxIdle="30" maxWait="2000"
            removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
            validationQuery="SELECT 1" validationInterval="30000" testOnBorrow="true"
            username="root" password="TempusServaFTW!" driverClassName="com.mysql.jdbc.Driver"
            url="jdbc:mysql://localhost:3306/tslive?autoReconnect=true"
  />
  <Parameter name="ExecutableImageMagick" value="c:\ImageMagick\convert"/>
  <Parameter name="ExecutableGhostscript" value="c:\Program Files\gs\gs9.20\bin\gswin64c.exe"/>
  <Parameter name="ExecutableTesseract" value="c:\Program Files (x86)\Tesseract-OCR\tesseract"/>
  <Parameter name="LanguagesTesseract" value="eng+dan"/>
  <Parameter name="ElasticServerAddress" value="localhost"/>
</Context>
```

Linux example configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/tsFileIndexingService">

 <Resource name="jdbc/TempusServaLive" auth="Container" type="javax.sql.DataSource"
           maxActive="80" maxIdle="30" maxWait="2000"
           removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true"
           validationQuery="SELECT 1" validationInterval="30000" testOnBorrow="true"
           username="root" password="TempusServaFTW!" driverClassName="com.mysql.jdbc.Driver"
           url="jdbc:mysql://localhost:3306/tslive?autoReconnect=true"
 />
 <Parameter name="ExecutableImageMagick" value="/usr/bin/convert"/>
 <Parameter name="ExecutableGhostscript" value="/usr/bin/ghostscript"/>
 <Parameter name="ExecutableTesseract" value="/usr/bin/tesseract"/>
 <Parameter name="LanguagesTesseract" value="eng+dan"/>
 <Parameter name="ElasticServerAddress" value="localhost"/>
</Context>
```

Enable and test indexing in Tempus Serva

Set the following configurations to true

- fulltextIndexData
- fulltextIndexFile

Also add port 8080 to the following URL

- fulltextFileHandlerURL

Update any record in the TS installation

Tjeck the index is created and that there is a mapping for the solution

```
curl 'http://localhost:9200/tempusserva/?pretty'
```

Next validate that records are found when searched for (replace * with a valid string)

```
curl 'http://localhost:9200/tempusserva/_search?pretty&q=*'
```

Finally validate that the Tempus Serva wrapper also works

```
http://<server>/TempusServa/fulltextsearch?subtype=4&term=*
```

Optional OCR components

Some libraries must be installed (ghostscript is probably already installed)

```
sudo yum install ImageMagick
sudo yum install ghostscript
```

Also install tesseract

CentOS/Fedora

```
sudo yum install tesseract-ocr
```

Amazon linux

```
sudo yum --enablerepo=epel --disablerepo=amzn-main install libwebp
sudo yum --enablerepo=epel --disablerepo=amzn-main install tesseract
```

Afterwards change the configurations in the file indexer

```
sudo nano /usr/share/tomcat7/conf/Catalina/localhost/tsFileIndexingService.xml
```

The values should be

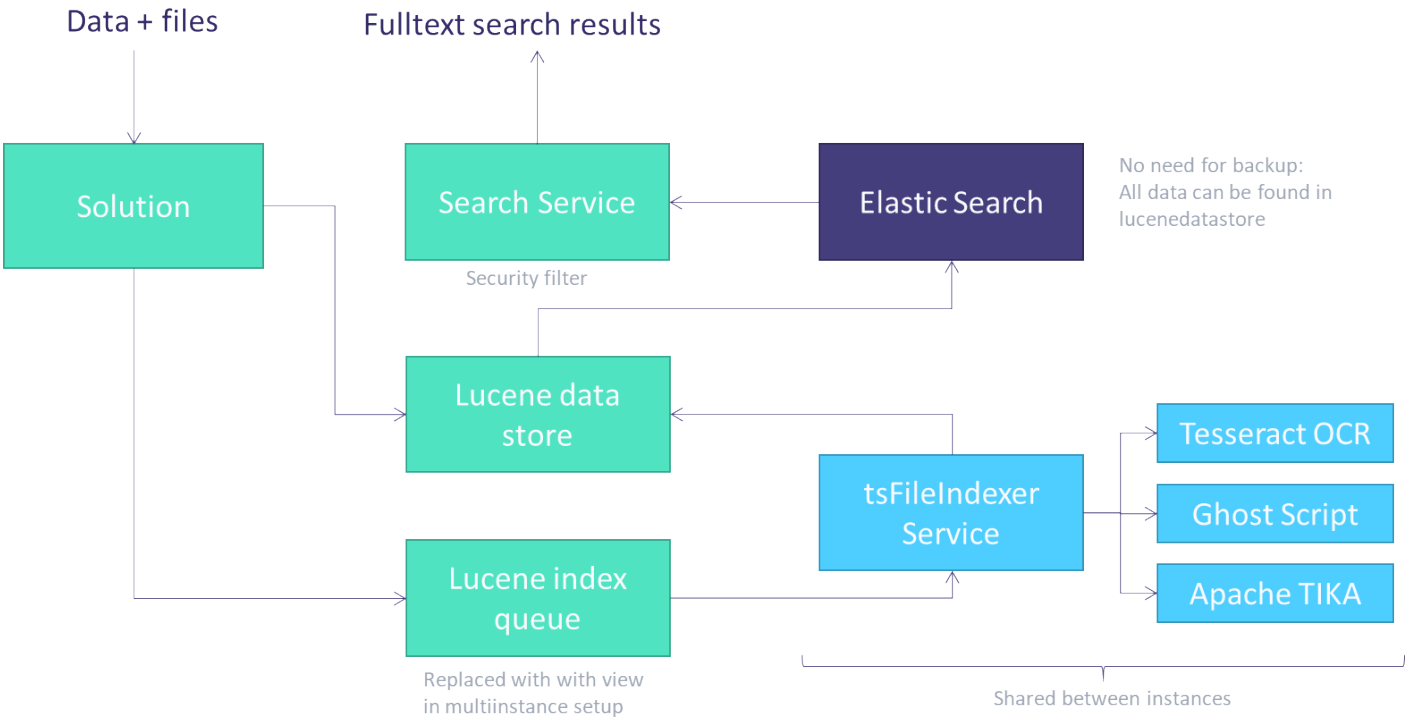
- /usr/bin/tesseract
- /usr/bin/convert
- /usr/bin/ghostscript

After changing the values restart the server.

Introduction

Adding Elastic search to you existing TS installation, will provide you with freetext searches in data and files.

Files are indexed together with the data in the records, so a record can be found by either their record values (name, phone etc.) or by search hits in files attached to those records. Results are filtered realtime according to the current security model, so no indexing is needed if settings change.



Lucene data store and services

Lucene data store will contain lines for each record and record file in the system.

All data in **Lucene data store** will be sent to Elastic search. Every time a record is updated an entry is made in **Lucene data store**, and by default the data is sent synchroniously to ElasticSearch (fulltextBatchProcessBlobs).

Files are instead put in the **Lucene file queue**. By default the indexer is notified imidiately, a will start executing the **Lucene file queue**. When finished each translated text is written back into the **Lucene data store** and deleted from the **Lucene file queue**. Data is by default be sent to Elastic search right away (fulltextBatchProcessFiles).

TS contains 2 services

- Data index builder
- File index builder

These services sends data from **Lucene data store** to ElasticSearch. As mentioned this will normally be carried out automatically / synchroniously, unless some kind of error occurs - like Elastic being offline etc. In that case unprocessed items queue up: In the datastore, file queue or both.

Running the services will handle everything in the queues.

Consider having **Data index builder** running every day (1440 mins) to clean up the queue now and then.

Reindexing

Reindex files

Before reindexing starts may clean up the index (this is optional)

```
DELETE FROM lucenedatastore WHERE FieldID > 0;
```

To reindex execute the statement below using the following parameters

- schema of the database (example: "tslive")
- file table of the solution (example: "data_solution_file")

```
INSERT INTO lucenefilequeue (application,tablename,FileID)
SELECT 'tslive', 'data_solution_file', f.ID as FileID FROM data_solution_file as f WHERE f.IsDeleted = 0;
```

After executing the statement execute the indexing service and wait patiently

Rewrite index

I case your Elastic Search is lost or corrupted it is quite easy to add the whole database to Elastic search

```
UPDATE lucenedatastore SET IsProcessed = 0;
```

Note this will just add all data once more - there is no indexing or OCR being carried out.

Reindexing existing data

Make sure that the application name is correct in the **applicationName**

Data can be reindexed using

```
Backend > Admin services > Rebuild artifacts > Index blobs
```

Optionally include the files too

```
Backend > Admin services > Rebuild artifacts > Index files
```

Setting up basic search service

Note that the Elastic search server can be installed on a seperate server (neither TS file indexing or the application server is required).

Install: Elastic search server

Elastic search server (version 5) will run standalone and will require Java 8 or higher

1. Download Elastic search zip archieve
2. Unpack files to suitable location
3. Start elastic.bat in /bin folder

```
https://www.elastic.co/downloads/elasticsearch
```

For Linux you can follow the guide in Install with tar

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.9.1-x86_64.rpm
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.9.1-x86_64.rpm.sha512
shasum -a 512 -c elasticsearch-8.9.1-x86_64.rpm.sha512
sudo rpm --install elasticsearch-8.9.1-x86_64.rpm
sudo rpm -e elasticsearch-8.9.1-x86_64.rpm
```

Alternatively use this script

```
sudo rpm --import https://packages.elastic.co/GPG-KEY-elasticsearch
sudo sh -c 'curl https://gist.githubusercontent.com/nl5887/b4a56bfd84501c2b2afb/raw/elasticsearch.repo
>> /etc/yum.repos.d/elasticsearch.repo'
sudo yum install -y elasticsearch
sudo chkconfig elasticsearch on
```

```
sudo nano /etc/elasticsearch/jvm.options
-Xms256m
-Xmx256m
```

```
sudo service elasticsearch start
```

```
curl 'http://localhost:9200/app/_count?pretty&q='y
```

Install: TS file indexing service (TSFIS)

For TSFIS to run yo will need a servlet container (Tomcat,JBoss,Oracle AS).

1. Download tsFileIndexingService.war
2. Dump to webapplication folder on application server
3. Change settings in web.xml
   - Database connection strings: If on same server just copy the seeting from your main application
   - ExecutableGhostscript: Path to Ghostscript (see above)
   - ExecutableTerrasect: Path to Terrasect OCR module (see above)
   - ElasticServerAddress: IP or servername where ElastisSearch is installed (see above)
4. Restart server (to reload DB credentials)
5. Test application at: <server>/tsFileIndexingService/execute

Network configuration

In the event that Elastic search or the file indexer is not on the same server you will need to ensure that

- Open port 3306 fra **TS file indexing service** to **MySQL database** (normally the application server)
- Open port 2100 fra **TS file indexing service** to **ElasticSearch** server
- Open port 2100 fra **Tempus Serva application** to **ElasticSearch** server

Also remember to update configrations for server names

- Elastic search: elasticsearch.yml file > network.host (add IP or servername)
  - https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-network.html
- TS file indexing service: web.xml > ElasticServerAddress (elastic search server)
- Tempus Serva application: Server policies
  - fulltextFileHandlerURL (file indexing server)
  - fulltextElasticBaseURL (elastic search server)

Multi application setup

1. Setup a shared table for **lucenefilequeue** using views
   - Delete the lucenefilequeue table in all slave databases
   - Create a view of lucenefilequeue pointing to the master database

2. **TS file indexing service** must have a user with access to all TS databases

Multiple instances will have a shard each in the Elastic index

Trouble shooting

Status on the file indexing

The file indexer has a stus page that will display information about the state of the indexer

```
https://<server>/tsFileIndexingService/execute
```

The page also constains a goodword "HEALTHY" taht is displayed if the process has not exceeded the specified timeouts.

Controlling timeouts

Timeouts are specified in seconds and should be tuned to CPU size and quality of documents

```
<Parameter name="TimeoutTesseract" value="600"/>
<Parameter name="TimeoutGhostscript" value="60"/>
```

Poor quality documents on virtualized environments can easily consume about a minute per page.

Debugging OCR proces

By default output from the external components are written to logfiles, which can be disabled by adding this option

```
<Parameter name="SuppressCommandOutput" value="0"/>
```

Note that there is a switch in configuration file (context.xml) which can disable file deletion on the server

```
<Parameter name="DisableFileCleanup" value=""/>
```

Understanding integrated search

The integrated fulltext serach using Elatic search is a internal/active approach to indexing the content. Content will be added to a indexing queue every time it is updated - ensuring allways updated content, but consuming CPU ressources on the indexing server.

Beacuase file indexing is very CPU intensive, the file indexing functionality i seperated into a service that can run on a server seperated from te main application server. Anyway the fileindexer will run from a database queue, so in most cases seperation is not strictly required.

The basic search service requires

- TS file indexing service (queue handler)
- Elastic search server (search engine)

For multitenant setups a single TS file indexing service can service multiple instances, as long as they write requests to the same queue (using DB views). The Elastic search server can also handle multiple applications.

If PDF OCR functionality is needed the following components needs installation too

- Ghostscript (PDF to TIFF conversion)
- Tesseract (OCR library)

The above components for OCR must be installed on the file indexing server.

Behind the scenes

Indexes in Tempus Serva is stored in intermediate tables in the database. ElasticSearch indexes can be dropped an regenerated from the intermediate storage.

Data indexing

Data is mainly indexed in one large text blob.

> Submit data > Stored in **lucenedatastore** > Transfer to ElasticSearch

For solutions using version history data will be reused, in order to minimize the overhead.

File indexing

File indexes points to the record, not the file itself. Likewise permission checks will rely on read access to a record.

```
Upload file > Stored in lucenefilequeue > tsFileIndexingService > Transfer to ElasticSearch
```

The indexing service handles files in various conversion processes

- tsFileIndexingService > Apache Tika (most files)
- tsFileIndexingService > terrasect (tif images)
- tsFileIndexingService > GhostScript > terrasect (PDF images)

For multi application installation **lucenefilequeue** is made for sharing between applications.

ElasticSearch structure

Multiple applications can share the same ElasticSearch server

```
/ APPLICATION / SOLUTION / RECORD ID
```

Records contains the must general information

- Title
- Content (large text blob)
- SagID
- DataID
- FieldID (in case of subrecords)
- ModifiedAt
- ModifiedBy (UserID)

Serahc results are filtered against the Tempus Serva permission engine **on record level**.

Data and subrecords (such as files) are stored in the same area, with slight adjustment to their record ID: DataID + "f" + FileID

```
/tempusserva/crm/6541
/tempusserva/crm/6541f45
```