FactExtractCloud

A Watson Explorer Content Analytics UIMA analysis engine for simplified information extraction.

Revision History

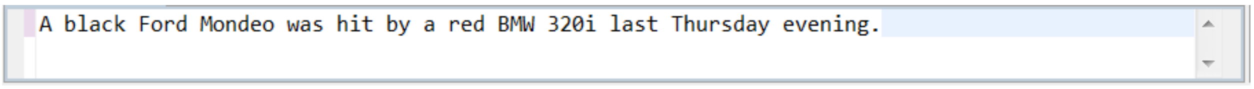
Date	Version	Status	Description	Author
02/06/2017	1.0	Release	Initial	Martin Saunders

Contents

1. Purpose

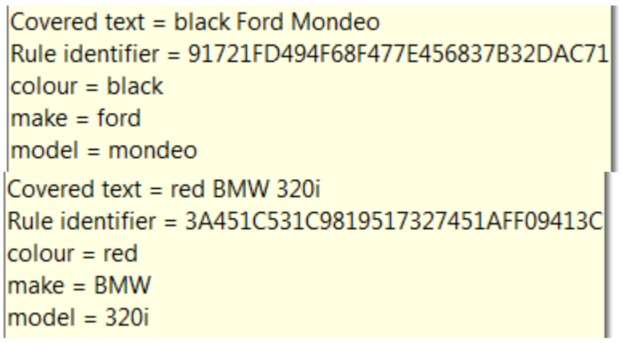
After text mining, information extraction projects are probably the most common usage for Watson Explorer Content Analytics. To do information extraction an annotator is configured in Content Analytics Studio to identify facts of interest and when this custom annotator is run in an Content Analytics server collection all the matching facts are extracted from the collection's corpus of documents and added to the index. These can then be exported to a relational database using standard features in the Content Analytics server. In effect Content Analytics has converted unstructured text to structured data which is typically consumed by 3rd party applications or other IBM tools such as Cognos, SPSS, i2 and DataStage. If your information extraction needs are for entities that are identified with a single value such as names, email addresses and credit card numbers these standard capabilities work very well.

In many instances the entities you want to extract are more complex and have multiples attributes that are also in the text; think of things like a vehicle, these may have a make, model and colour . These can be modeled in Content Analytics Studio using an annotation with multiple features. The instances of annotated text together with all the feature values can also all be stored in Content Analytics server indexes (and hence exported) but the text identifying the entity and each feature value are stored separately. Effectively the implied relationship between them is lost. For example, consider a document containing the following text



```
A black Ford Mondeo was hit by a red BMW 320i last Thursday evening.
```

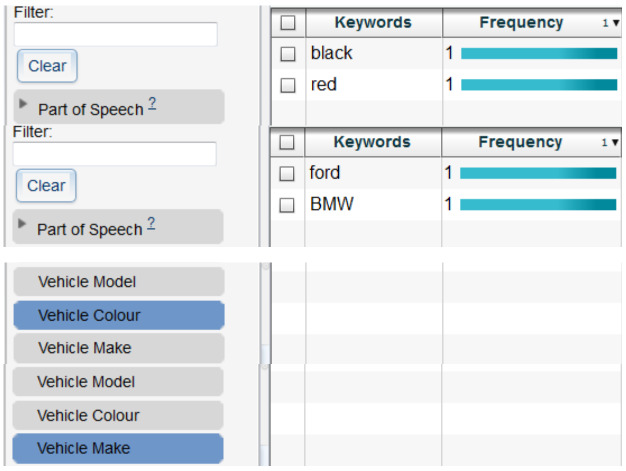
If your interest was in identifying vehicles you might well configure a com.ibm.Vehicle annotation in Content Analytics Studio that would create the following two instance with that text.



```
Covered text = black Ford Mondeo
Rule identifier = 91721FD494F68F477E456837B32DAC71
colour = black
make = ford
model = mondeo
```

```
Covered text = red BMW 320i
Rule identifier = 3A451C531C9819517327451AFF09413C
colour = red
make = BMW
model = 320i
```

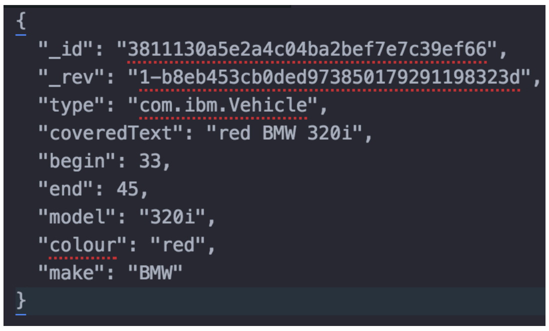
When this is deployed in an Content Analytics collection with a document containing the same text the text miner facets show:



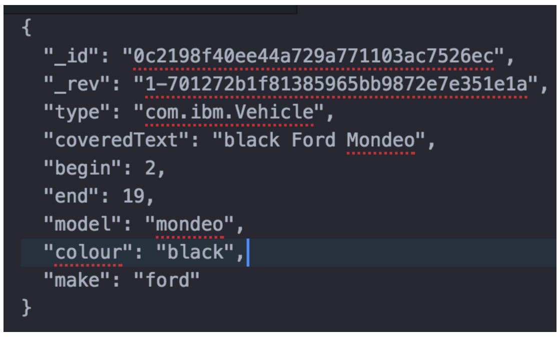
Keywords	Frequency
black	1
red	1

Keywords	Frequency
ford	1
BMW	1

Which vehicle is black? You cannot drill-down on ***black*** in the **Vehicle Colour** facet and expect to see one **Vehicle Make** as the answer; you'll see both, because both **Vehicles Makes** are in are in the same document as the **Vehicle Colour *black***. If we export these facts we'll see we've got a BMW, a Ford, a black vehicle and a red vehicle, and we'll also know which document they occur in. But the only way to answer the question is to read the text. Using the FactExtractCloud overcomes this limitation as facts with multiple features can be written as a single json document to a database. Using FactExtractCloud the above text could produce the following two documents.



```
{
  "_id": "3811130a5e2a4c04ba2bef7e7c39ef66",
  "_rev": "1-b8eb453cb0ded973850179291198323d",
  "type": "com.ibm.Vehicle",
  "coveredText": "red BMW 320i",
  "begin": 33,
  "end": 45,
  "model": "320i",
  "colour": "red",
  "make": "BMW"
}
```



```
{
  "_id": "0c2198f40ee44a729a771103ac7526ec",
  "_rev": "1-701272b1f81385965bb9872e7e351e1a",
  "type": "com.ibm.Vehicle",
  "coveredText": "black Ford Mondeo",
  "begin": 2,
  "end": 19,
  "model": "mondeo",
  "colour": "black",
  "make": "ford"
}
```

Creating records like this allows the relationships between features and their associated annotated text be maintained and allows for easy integration with downstream applications.

2. Installation

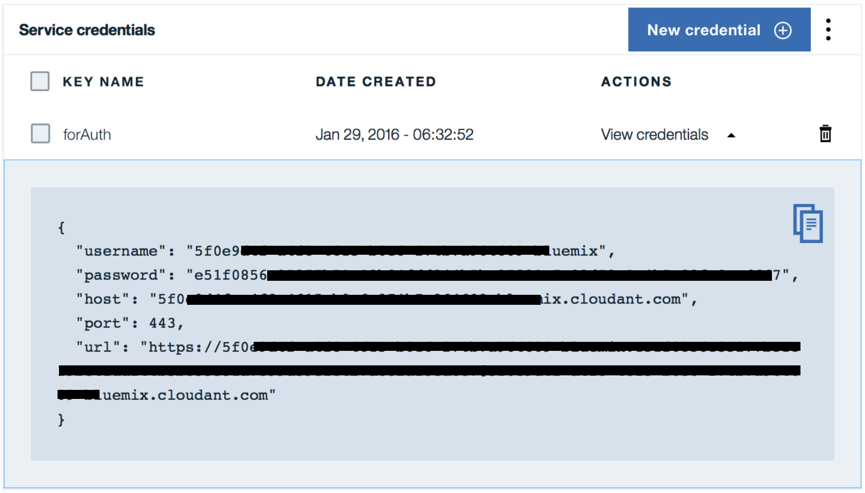
2.1 Pre-Requisites

- Installation of Watson Explorer v11 or above with access to the administrative application via esadmin user.
- Installation of Content Analytics Studio version 11 or above
- An instance of the Cloudant service on Bluemix or equivalent
  - Access to the service credentials.
- FactExtractCloud files:
  - Core files (included in distribution):
    - FactExtractCloud-ae.xml
    - FactExtractCloud-n-n-n.jar
    - ICAUIMAUtils-n-n-n.jar
  - Cloudant and library support (included in distribution):
    - cloudant-client-2.9.0.jar
    - cloudant-http-2.9.0.jar
    - commons-codec-1.6.jar
    - commons-io-2.4.jar
    - gson-2.7.jar

2.2 Obtain service credentials

2.2.1 Login to Bluemix and create or find your existing Cloudant service instance.

2.2.2 Open the credentials for that instance. They'll look something like this



KEY NAME	DATE CREATED	ACTIONS
forAuth	Jan 29, 2016 - 06:32:52	View credentials

```
{
  "username": "5f0e9[redacted]uemix",
  "password": "e51f0856[redacted]7",
  "host": "5f0[redacted]ix.cloudant.com",
  "port": 443,
  "url": "https://5f0[redacted]uemix.cloudant.com"
}
```

2.2.3 Make a note of the **username** and **password**

2.2.4 Make a note of the **host** parameter excluding the ".cloudant.com" suffix. This is your Cloudant **account** name.

2.3 Content Analytics Studio

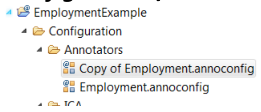
2.3.1 Create a project and configure your annotations to identify relevant facts.

2.3.2 In this project create a folder to hold the FactExtract resources. This can be anywhere in the workspace but something like ***Resources/Custom/FactExtractCloud*** would be a good choice.

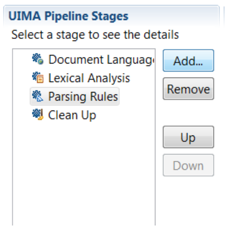
2.3.3 Copy the annotation engine configuration file (***FactExtractCloud-ae.xml***) and all the jar files that make up FactExtractCloud into this new folder.

2.3.4 In the relevant UIMA pipeline configuration file in your project add a custom stage as the penultimate stage of the pipeline. To do this:

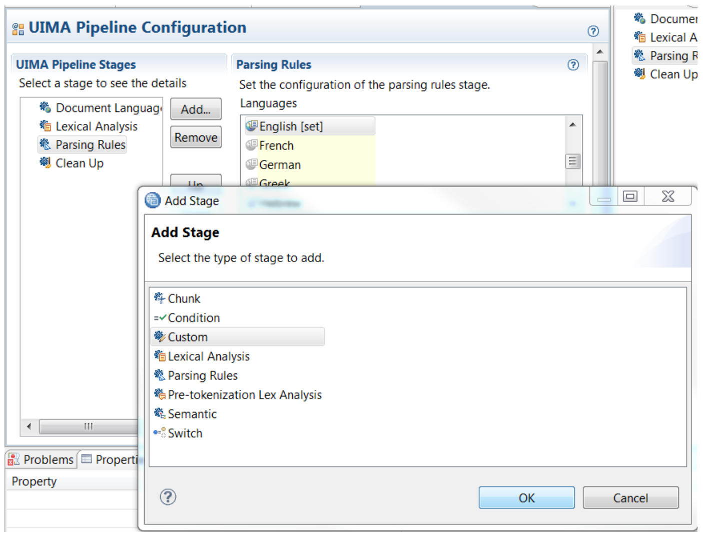
- Double click on the relevant pipeline configuration file under your ***Configuration/Annotators*** folder to open and edit it.



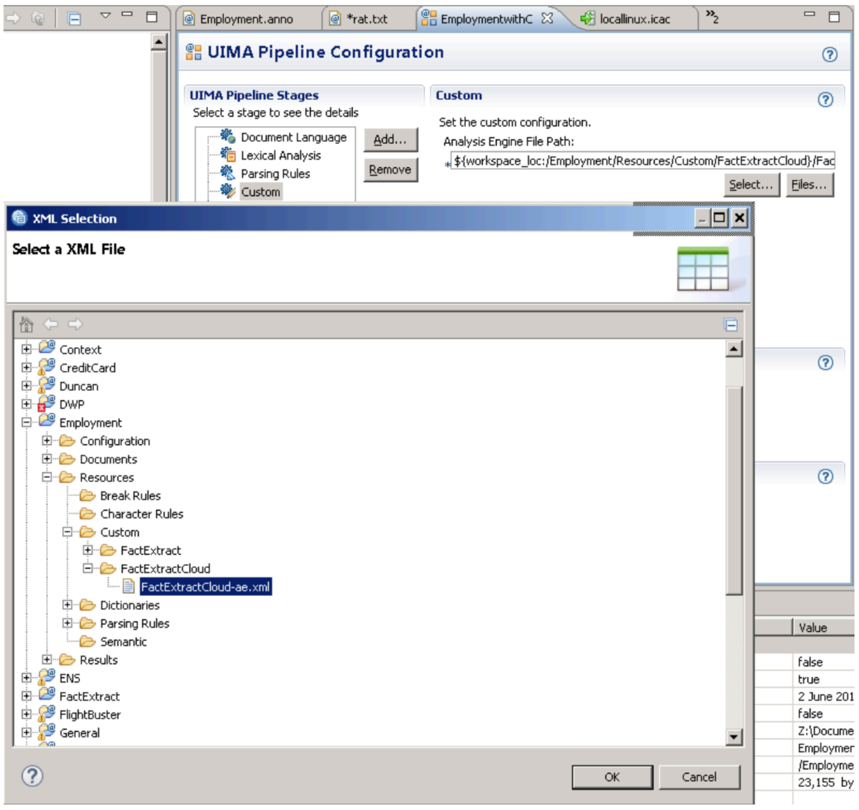
- select the existing penultimate stage
  click the “Add” button



- select “Custom” in the popup window
  click “OK”



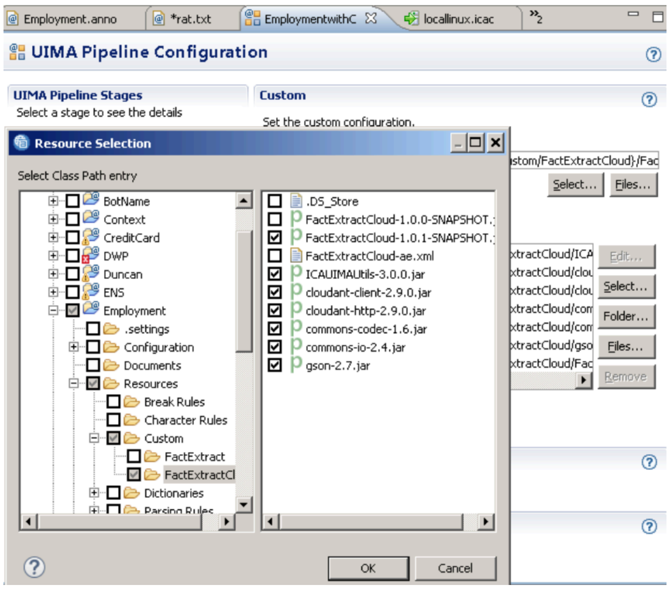
- Click in the text box of the Custom panel under where it says ***Analysis Engine File Path:*** and click the ***Select*** button. Navigate to the folder created in step 2.3.2 and select the ***FactExtractCloud-ae.xml*** annotation engine configuration file.



Analysis Engine File Path: ${workspace_loc:/Employment/Resources/Custom/FactExtractCloud}/Fac

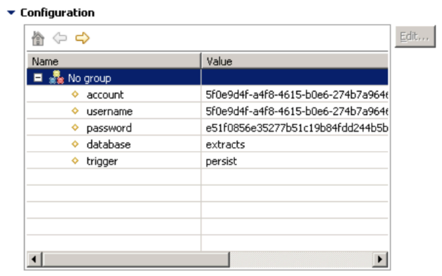
Value
false
true
2 June 201
false
Z:\Docume
Employmer
/Employme
23,155 by

- Back in the ***Custom*** panel click the small right arrow to open the ***Class Path*** panel and click ***Select***. Again navigate to the new folder and this time click the check boxes to select the jar files.
  Click ***OK***



- [ ] .DS_Store
- [ ] FactExtractCloud-1.0.0-SNAPSHOT.j
- [x] FactExtractCloud-1.0.1-SNAPSHOT.j
- [ ] FactExtractCloud-ae.xml
- [x] ICAUIMAUtils-3.0.0.jar
- [x] cloudant-client-2.9.0.jar
- [x] cloudant-http-2.9.0.jar
- [x] commons-codec-1.6.jar
- [x] commons-io-2.4.jar
- [x] gson-2.7.jar

- Back in the ***Custom*** panel click the small right arrow to open the ***Configuration*** panel. then click the small plus sign to open the ***No group*** Configuration group.



Name	Value
No group	
account	5f0e9d4f-a4f8-4615-b0e6-274b7a964
username	5f0e9d4f-a4f8-4615-b0e6-274b7a964
password	e51f0856e35277b51c19b84fdd244b5b
database	extracts
trigger	persist

Set the parameters appropriately according to the guidance in the table below.

Parameter	Default value	Note
account	none	* Cloudant service account from 2.2.4
username	none	* Cloudant service username from 2.2.3
password	none	* Cloudant service password from 2.2.3
database	extracts	The Cloudant database to write to. It will be created if it does not exist
trigger	persist	The name of the feature in annotations that will be used to signal annotations of this type should be saved.

* At a minimum these parameters should be changed.

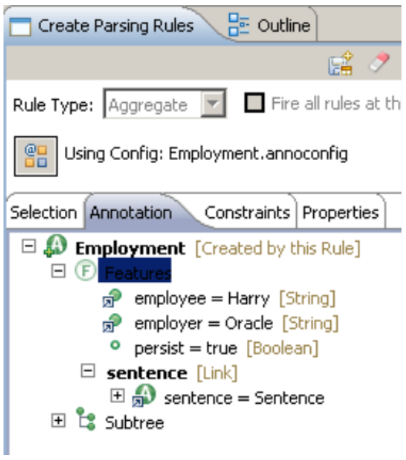
2.4 Content Analytics Server

Once you have defined, developed and tested your project that includes FactExtractCloud, you must export it to the Content Analytics server so that it can be used to analyze documents. This is done by simply deploying your Studio project that contains the FactExtractCloud custom stage to the server just like you would with any other project.

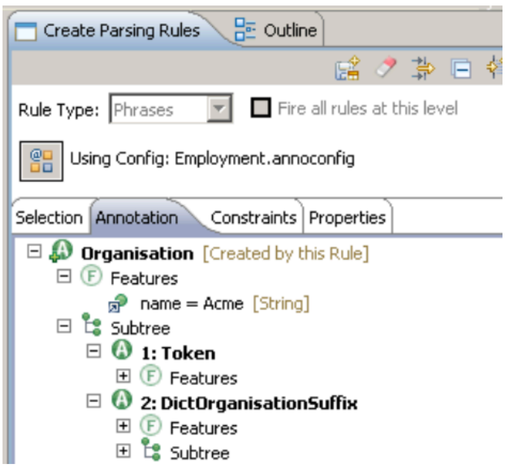
3. Configuring the extraction

The FactExtractCloud can be run in pipelines in Content Analytics Studio and in pipelines deployed to the Content Analytics Server. It's often easiest to first run it in Studio to test the extractions and then run it deployed into your server to extract facts from your corpus of documents in a collection.

FactExtractCloud uses the presence of a named feature in your annotations to decide if an annotation is going to be persisted or not. The name of this feature is configured in the custom stage (see 2.3), the default name is "persist". So given the two annotation definitions below the first one will be persisted and the second one won't.



```
Rule Type: Aggregate   Fire all rules at th
Using Config: Employment.annoconfig
Employment [Created by this Rule]
  Features
    employee = Harry [String]
    employer = Oracle [String]
    persist = true [Boolean]
  sentence [Link]
    sentence = Sentence
  Subtree
```

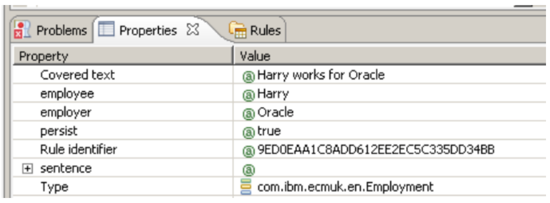


```
Rule Type: Phrases   Fire all rules at this level
Using Config: Employment.annoconfig
Organisation [Created by this Rule]
  Features
    name = Acme [String]
  Subtree
    1: Token
      Features
    2: DictOrganisationSuffix
      Features
      Subtree
```

The value of the feature is irrelevant, the mere presence is enough to trigger the save.

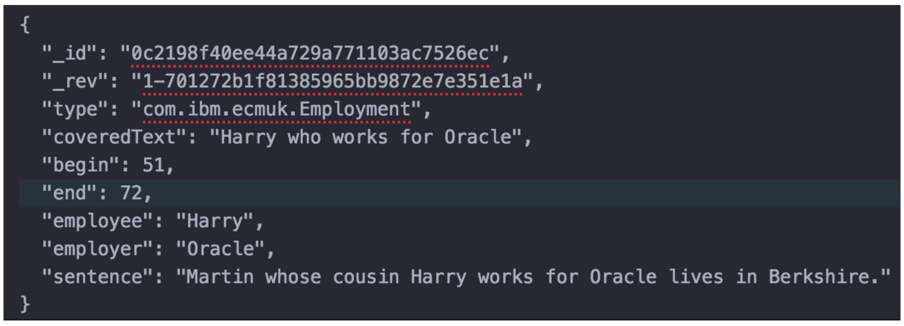
3.1 What gets persisted

All primitive features with the exception of the rule identifier get persisted, as does the covered text, the character offsets of the annotation and the annotation type name. If you've associated either the sentence or paragraph with the annotation (as in the above Employment example) then the text of that sentence or paragraph will get persisted too. Here's an instance of that example Employment annotation in Studio:



Property	Value
Covered text	Harry works for Oracle
employee	Harry
employer	Oracle
persist	true
Rule identifier	9ED0EAA1C8ADD612EE2EC5C335DD34BB
sentence	
Type	com.ibm.ecmuk.en.Employment

and here's the final document in Cloudant:



```
{
  "_id": "0c2198f40ee44a729a771103ac7526ec",
  "_rev": "1-701272b1f81385965bb9872e7e351e1a",
  "type": "com.ibm.ecmuk.Employment",
  "coveredText": "Harry who works for Oracle",
  "begin": 51,
  "end": 72,
  "employee": "Harry",
  "employer": "Oracle",
  "sentence": "Martin whose cousin Harry works for Oracle lives in Berkshire."
}
```

There is currently no support in the annotator for updates or deletes. So re-running the annotator will create more documents in Cloudant each with a unique **_id** and **_rev**.

4. Running Content Analytics Collections for Information Extraction Only.

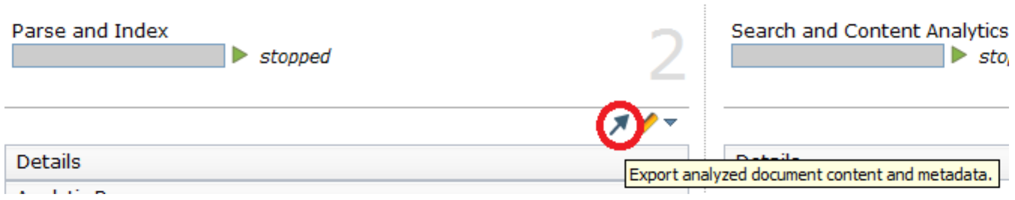
This is an optional configuration step.

Having successfully followed the steps described in sections 1 – 3 you will have a collection with a custom analysis step included in its processing pipeline. When the collection crawler is configured and started, the source system will be crawled, and the data analyzed using the custom annotator you designed and that annotator will have been will configured using the FactExtract to automatically write relevant annotations to a database. Finally, the server will also write, at a minimum, the standard lexical analysis annotations to the collection index (things like language, parts of speech, sentence, and paragraph annotations) to be used in either the Text Miner or Search applications. In many cases where we are writing extracted data to a database there is no requirement for these applications, and hence no use for an index. Constructing and maintaining these indexes is time consuming and wasteful of resources. In these cases it is possible to configure the pipeline to run the analysis stages only and not build an index for the collection.

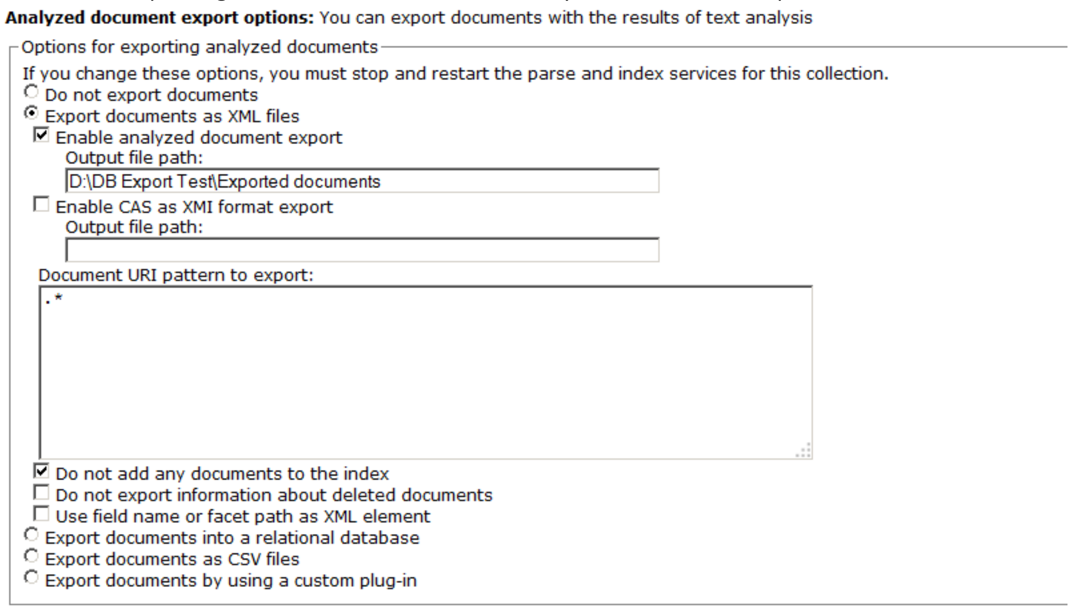
The procedure for doing this is not obvious as the relevant options are included in the collection export configuration screens.

4.1 Turning off document indexing

- From the Content Analytics Administration Console, expand the collection and click on the export icon in Parse and Index section.


Parse and Index
stopped
2
Search and Content Analytics
sto
Details
Export analyzed document content and metadata.

- On the Export configuration page, you will want to configure an export for Analyzed Documents, as the documents must be analyzed in order for custom annotator to be applied to the documents. There are several options to choose from, each of which is explained in more detail in the on-line help. If you simply want to avoid the creation of an index then exporting the documents as xml files may be a worthwhile option:

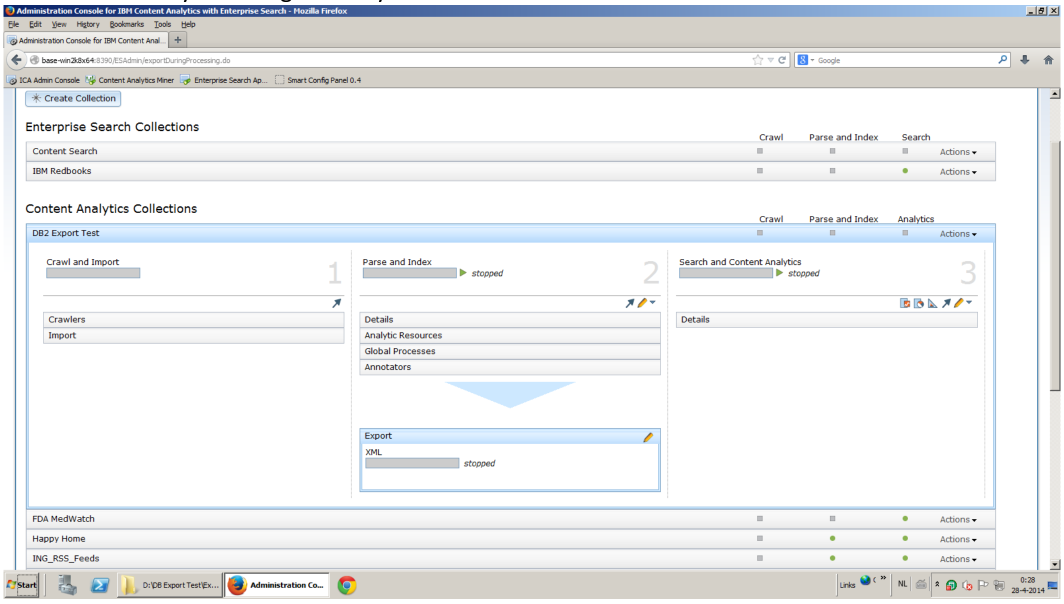

Analyzed document export options: You can export documents with the results of text analysis

Options for exporting analyzed documents

If you change these options, you must stop and restart the parse and index services for this collection.

- ○ Do not export documents
- ● Export documents as XML files
  - ☑ Enable analyzed document export
    - Output file path: D:\DB Export Test\Exported documents
  - ☐ Enable CAS as XMI format export
    - Output file path:
  - Document URI pattern to export: .*
  - ☑ Do not add any documents to the index
  - ☐ Do not export information about deleted documents
  - ☐ Use field name or facet path as XML element
- ○ Export documents into a relational database
- ○ Export documents as CSV files
- ○ Export documents by using a custom plug-in

- As the above figure shows, the two important parameters are the file location in which the xml files will be written, and the option "Do not add any documents to the index". This ensures that no index is created for the collection.
- One additional point to make here is that the system must write the documents somewhere. If you were to deselect the "Enable analyzed document export" option, you would get an error.
- Having completed the export configuration, you can simply import data or crawl a data sources. the documents will be written to the target you defined and the annotations to the database as you configured in your annotator.



- You can manually delete the xml files, or write a simple utility to do this automatically.