

SolrUIMA

Solr UIMA integration

⚠ UIMA Module was removed in Solr 7.5 (SOLR-11694)

⚠ Solr3.1

- [Solr UIMA integration](#)
 - [SolrUIMA UpdateRequestProcessor](#)
 - [Installation](#)
 - [Configuration](#)
 - [UIMA components used](#)
 - [Using other UIMA components](#)
 - [Import the component jar](#)
 - [Use a different UIMA descriptor](#)
 - [Adjust AE configuration \(optional\)](#)
 - [Change the types and features' mapping](#)
 - [Deploy new jars inside one of the lib directories](#)
 - [Solrcas](#)

Solr UIMA contrib enables enhancing of Solr documents using the Unstructured Information Management Architecture ([UIMA](#)). UIMA lets you define custom pipelines of Analysis Engines which incrementally add metadata to the document via annotations.

SolrUIMA [UpdateRequestProcessor](#)

The SolrUIMA [UpdateRequestProcessor](#) is a custom [UpdateRequestProcessor](#) that takes document(s) being indexed, sends them to a UIMA pipeline and then returns the document(s) enriched with the specified metadata.

Installation

1. Go to `dev/solr/contrib/uima` and run 'ant clean dist' 2. get the package `apache-solr-uima-4.0-SNAPSHOT.jar` together with the jars under the `dev/solr/contrib/uima/lib` directory and paste everything inside one of the lib directories of your Solr instance (defined inside the `solrconfig.xml`). You may need to create the lib directory for a specific core.

```
mkdir solr/example/solr/collection1/lib
cp solr/dist/apache-solr-uima*.jar solr/example/solr/collection1/lib
cp solr/contrib/uima/lib/*.jar solr/example/solr/collection1/lib/
cp solr/build/contrib/solr-uima/lucene-libs/lucene-analyzers-uima-4.0-SNAPSHOT.jar solr/example/solr/collection1/lib/
```

3. modify your Solr instance config files as described in the [solr/contrib/solr-uima/README.txt](#) 4. run your Solr instance and enjoy UIMA enriching documents being indexed

Configuration

All the SolrUIMA configuration is placed inside a `<uimaConfig>` element inside the `solrconfig.xml`.

```
<uimaConfig>
  <runtimeParameters>
    <!-- here go parameters defined in the AE which override parameters in the delegate AEs -->
    ...
  </runtimeParameters>
  <analysisEngine><!-- here goes the AE path in the classpath --></analysisEngine>
  <analyzeFields merge="true"><!-- comma separated list of fields of the original document to analyze --></analyzeFields>
  <fieldMapping>
    <!-- here goes the mapping between features of UIMA FeatureStructures to Solr fields -->
    <type name="org.apache.uima.something.Annotation">
      <map feature="oneFeature" field="destination_field"/>
    </type>
    ...
  </fieldMapping>
</uimaConfig>
```

The `analysisEngine` element holds the classpath to the UIMA Analysis Engine descriptor that describes which analysis block should be executed. The analysis engine referenced can be primitive or aggregate.

The `analyzeFields` element lists the name of fields (comma separated) which will be analyzed by the UIMA pipeline. If the attribute `merge` is false the field specified will be analyzed separately while if `merge` is true the listed fields contents will be merged and analyzed only once.

see [SOLR-2129](#)

UIMA components used

UIMA supports the use of existing analysis engines (see [here](#) and [here](#)) as long as the creation of custom components.

The current contrib/uima module uses a predefined set of components :

1. [WhitespaceTokenizer](#)
2. [HMMTagger](#)
3. [OpenCalaisAnnotator](#)
4. [AlchemyAPIAnnotator](#)

These components are arranged in a pipeline inside the [OverridingParamsExtServicesAE](#) Analysis Engine descriptor. As you can see looking at the descriptor fragment;

```
<node>AggregateSentenceAE</node>
<node>OpenCalaisAnnotator</node>
<node>TextKeywordExtractionAEDescriptor</node>
<node>TextLanguageDetectionAEDescriptor</node>
<node>TextCategorizationAEDescriptor</node>
<node>TextConceptTaggingAEDescriptor</node>
<node>TextRankedEntityExtractionAEDescriptor</node>
```

the first node represent an aggregate Analysis Engine which includes the Whitespace Tokenizer and HMM Tagger (recognizing sentences), the second node uses the Open Calais Annotator to extract named entities, the following nodes use different Alchemy API Annotator services to detect keywords, language, document category, discovered concepts and named entities.

Using other UIMA components

To use different UIMA components inside the contrib/uima module you need to:

1. import the component jar
2. use the new component Analysis Engine descriptor inside `config/uimaConfig/analysisEngine` element of `solrconfig.xml`
3. adjust Analysis Engine configuration (optional)
4. change the types and features' mapping inside `config/uimaConfig/fieldMapping` element of `solrconfig.xml`
5. deploy new `apache-solr-uima.jar` and component inside one of the lib directories

Import the component jar

If you're using Ant you only need put the component jar inside the `solr/contrib/uima/lib` directory.

If you're using Maven you need to declare the component you want to use inside the `<dependencies>` element in the generated `pom.xml`.

For example if you want to use UIMA Dictionary Annotator 2.3.1-SNAPSHOT you can either get it from [snapshot repo](#) and paste it in `solr/contrib/uima/lib` and run 'ant clean dist' or paste the following in the generated `pom.xml` (as child of the `<dependencies>` tag) and run 'mvn clean package'.

```
<dependency>
  <groupId>org.apache.uima</groupId>
  <artifactId>DictionaryAnnotator</artifactId>
  <version>2.3.1-SNAPSHOT</version>
</dependency>
```

Use a different UIMA descriptor

Change the descriptor to be used by this module inside `config/uimaConfig/analysisEngine` of the `solrconfig.xml` of your Solr instance.

One can use the default one bundled inside the component or create a new one.

For example to use one of the default Dictionary Annotator Analysis Engine descriptors use the following (which runs Whitespace Tokenizer and then Dictionary Annotator):

```

<config>
  ...
  <uimaConfig>
    ...
    <analysisEngine>/AggregateAE.xml</analysisEngine>
    ...
  </uimaConfig>
  ...
</config>

```

Adjust AE configuration (optional)

Sometimes Analysis Engines require custom parameters to be set inside their descriptor or custom resources to be imported. The easiest way to do so is to get a copy of such a descriptor, modify parameters/resources as needed and put them inside a directory which gets included in the final jar (i.e.: solr/contrib/uima/src/main/resources/org/apache/uima)

Change the types and features' mapping

Inside the solrconfig.xml go to config/uimaConfig/fieldMapping element and change <type> element according to the annotations extracted by the used component.

For example if you're using the Dictionary Annotator and you want to put the dictionary entry annotations found inside a 'lemmas' field you should configure the fieldMapping element as following:

```

<config>
  ...
  <uimaConfig>
    ...
    <fieldMapping>
      <type name="org.apache.uima.DictionaryEntry">
        <map feature="coveredText" field="lemmas"/>
      </type>
    </fieldMapping>
    ...
  </uimaConfig>
  ...
</config>

```

Deploy new jars inside one of the lib directories

Run 'ant clean dist' (or 'mvn clean package') from the solr/contrib/uima path.

Get the generated apache-solr-uima*.jar from the build directory along with the used components' jars and paste both inside one of the <lib> directories defined inside the solrconfig.xml.

You can now restart the Solr-UIMA instance to test it.

Solrcas

This is a UIMA component, see [SVN](#) and [documentation](#)

For a deepest dive into UIMA please take a look at the [documentation](#)