

JSONTopicMap Installation Manual

Latest: 20140219

Background

Installation entails the following steps:

- Download the github project¹
- Assemble the required library *jar* files by downloading them from github², and adding the json-topicmap-xxx.jar as illustrated in Figure 1
- Deal with ElasticSearch.
 - Remote Installation:
 - Download the tested version, elasticsearch.xxx.zip from the same github Library link as the jar files
 - Install it per its own instructions
 - Boot it -- it's ready to run
 - See Configuration section below if Remote is used
 - Embedded Installation
 - No instance of the full ElasticSearch server is needed; this mode will use the existing ElasticSearch jar files and create an embedded version of it in the same JVM in which the topic map is running. There are performance penalties for doing this, but it's an easy way to get started.
 - See Configuration section below if Embedded is used
- Boot the topic map using the run.bat (or shell script)

¹ JSONTopicMap: <https://github.com/SolrSherlock/JSONTopicMap>

² Libraries: <https://github.com/SolrSherlock/SolrSherlock/tree/master/masterLib>

Installation Directory

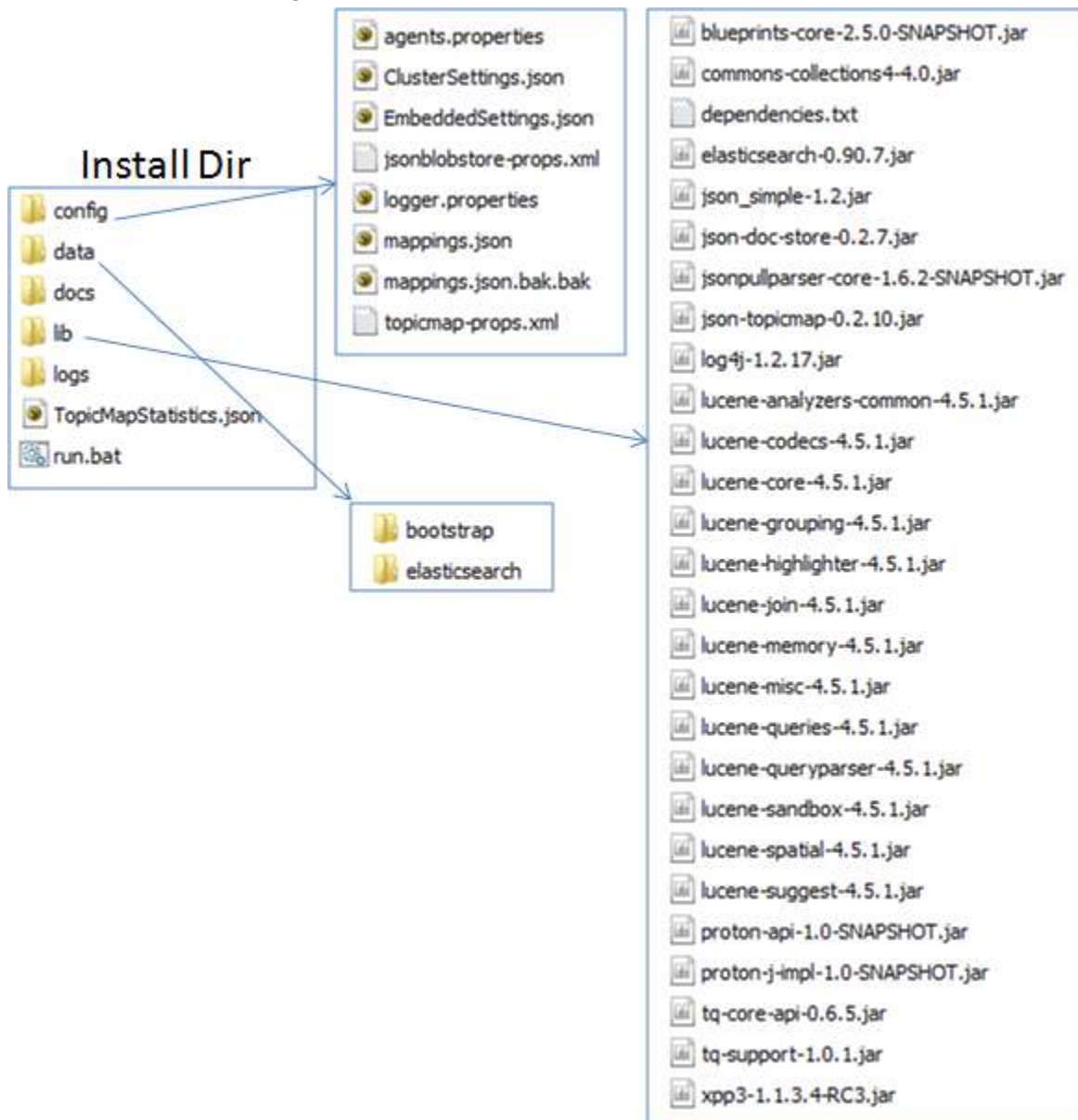


Figure 1 Installation Directories

Configuration

JSTopicMap requires a certain amount of configuration. First, it runs against the ElasticSearch index server; it can use that server in one of two different modes, depending on configuration settings:

- Remote, which means a single server, either remotely or even *localhost*, or a cluster of servers
- Embedded, which means the topic map will create an embedded index using the ElasticSearch libraries it includes. Embedded also means that indexing will be performed by the same Java virtual machine (JVM) that the topic map is using, which will slow down its performance.

Refer to the *config* directory image in Figure 1, and observe the *jsonblobstore-props.xml* file. Here are the relevant sections:

```
<!-- Model created for a local, embedded index  
  
<parameter name="Model"  
value="org.topicquests.persist.json.es.ElasticSearchEmbeddedModel" /> -->  
  
<!-- Model created for remote and cluster servers -->  
  
<parameter name="Model"  
value="org.topicquests.persist.json.es.ElasticSearchClusterModel" />
```

There are two configurations for JSONDocStore, the indexing database application on which JSONTopicMap runs. One is for Embedded, and one is for Clusters. In the illustration above, embedded is commented out.

To use an embedded database, be sure to comment out the cluster model (remove the "-->" following the comment, and move it to after the parameter field, as is now illustrated for the embedded model, and make sure that the embedded model is not commented out by moving its "-->" back to the comment field above the parameter field.

To use a remote database, make sure, as mentioned above, that the cluster model is not commented out and the embedded model is commented out. In the remote model, we need to tell the system the location of its server(s). For that, a different section of the props file is used:

```
<list name="Clusters">  
  
    <parameter name="localhost" value="9300" />  
  
</list>
```

In this section, we have the opportunity to list one or more ElasticSearch servers. The illustration is for a lone server, installed *locally* (on the same computer as the topic map), so that is its declaration.

Suppose that we install two servers on different computers. Using made-up IP numbers, here is how that might look:

```
<list name="Clusters">  
    <parameter name="10.1.10.92" value="9300" />  
    <parameter name="10.1.10.94" value="9300" />  
</list>
```

At the time of this writing, that summarizes the only critical configuration needed.

Running

Running simply means firing up a batch or script file.

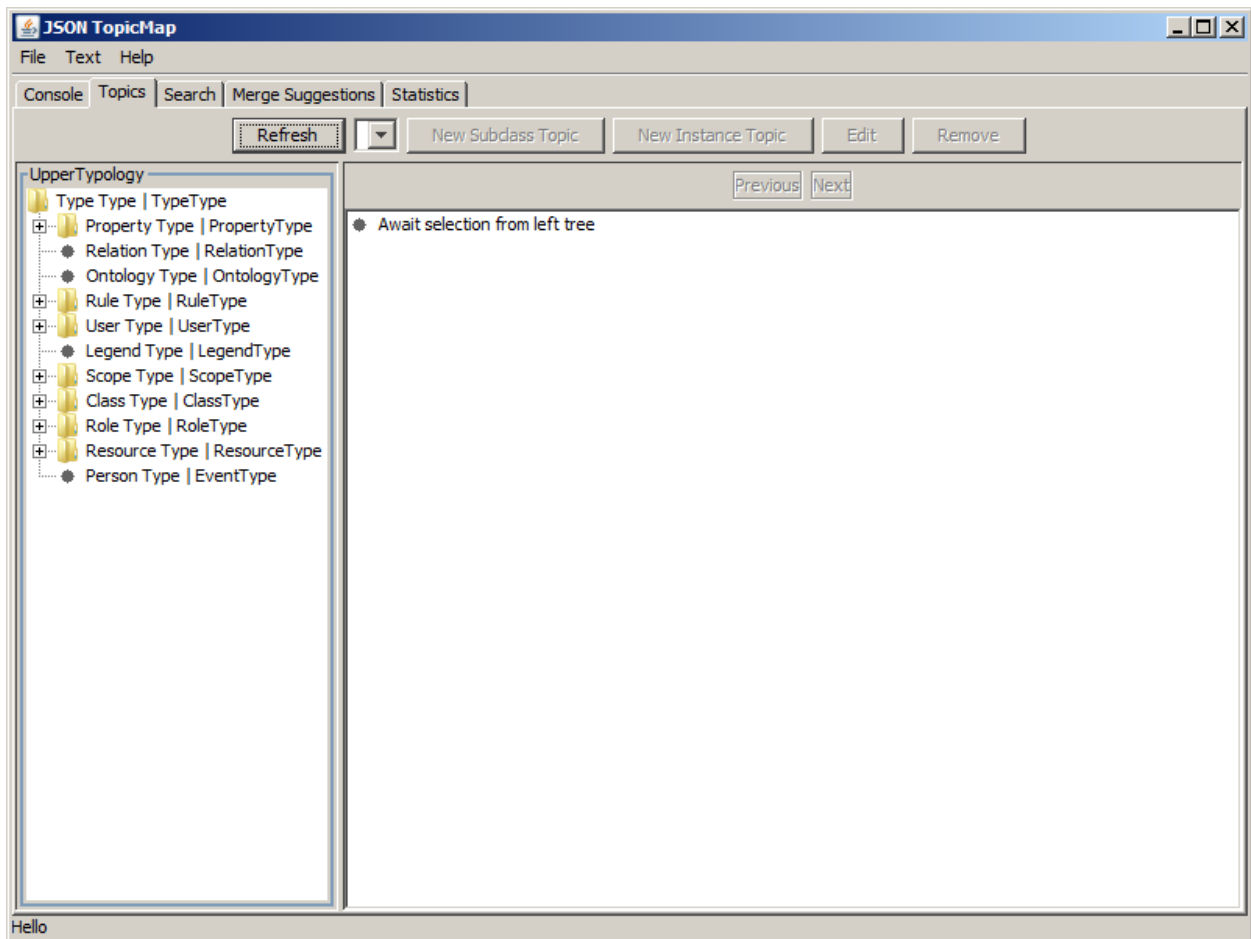


Figure 2. Basic Topic Navigator View

Figure 2 shows a tree view of what is in the index for this topic map. On first boot of the topic map, it will *bootstrap* (load) a collection of topics, each of which is intended to form a *typology* in which to capture topics. The highest 3 levels of that typology are shown in the left pane.

To see deeper, double click on any node in the left tree, and that node plus deeper levels are then displayed in the right tree, as shown in Figure 3. Double click on any node in the right tree, and that node replaces the root node in the right tree. There are *Next* and *Previous* buttons to allow for scrolling along long lists of nodes.

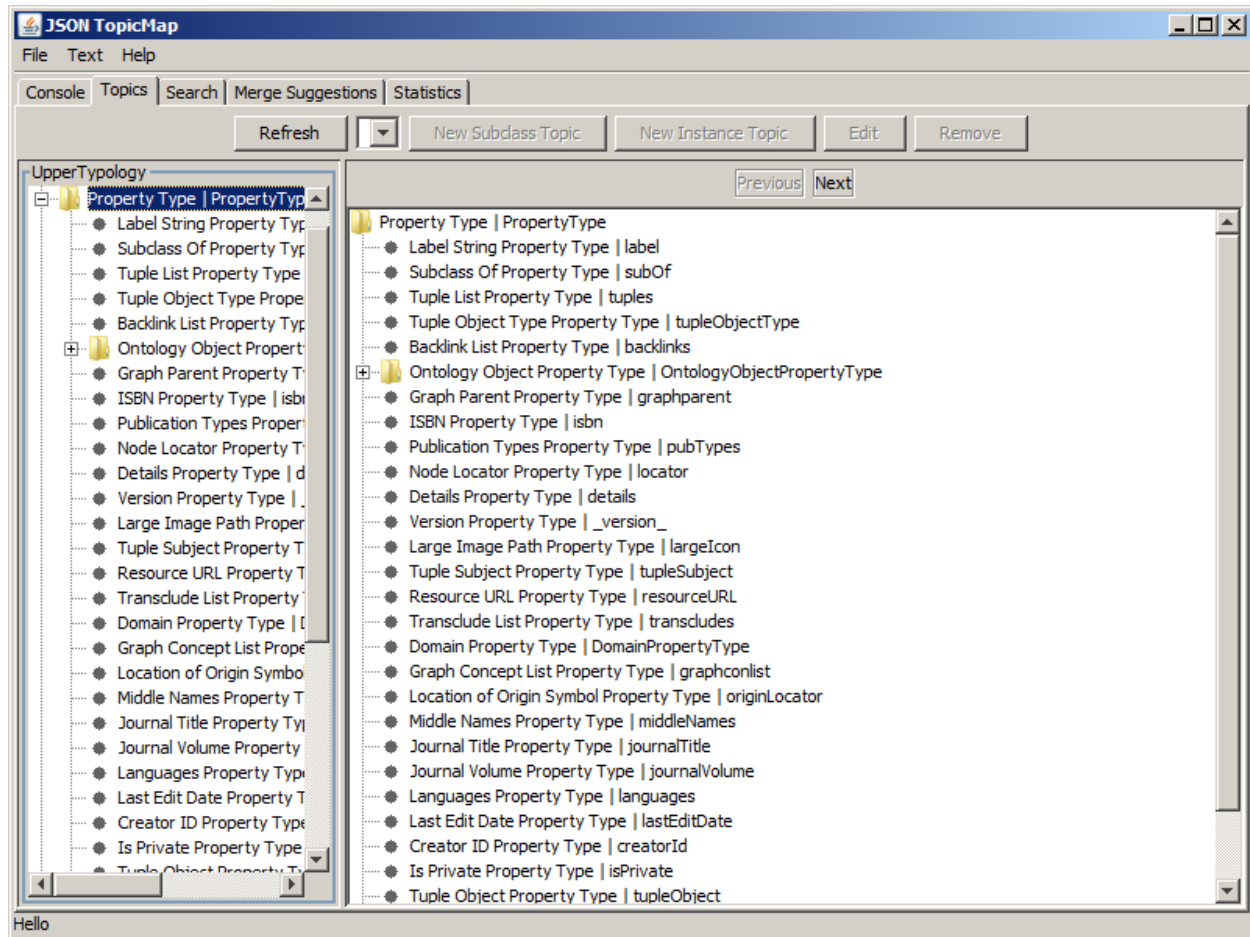


Figure 3. Drilling into the Topics

Select any node with a single click, and the *Edit* button will appear. Click the Edit button and an XML representation of the node will appear, as shown in Figure 4. At this point, it is possible to change values in the topic, then click *Save*. Typical changes are those of fixing typos in *label* field values or in *details* field values. Changing any other values should be considered dangerous.

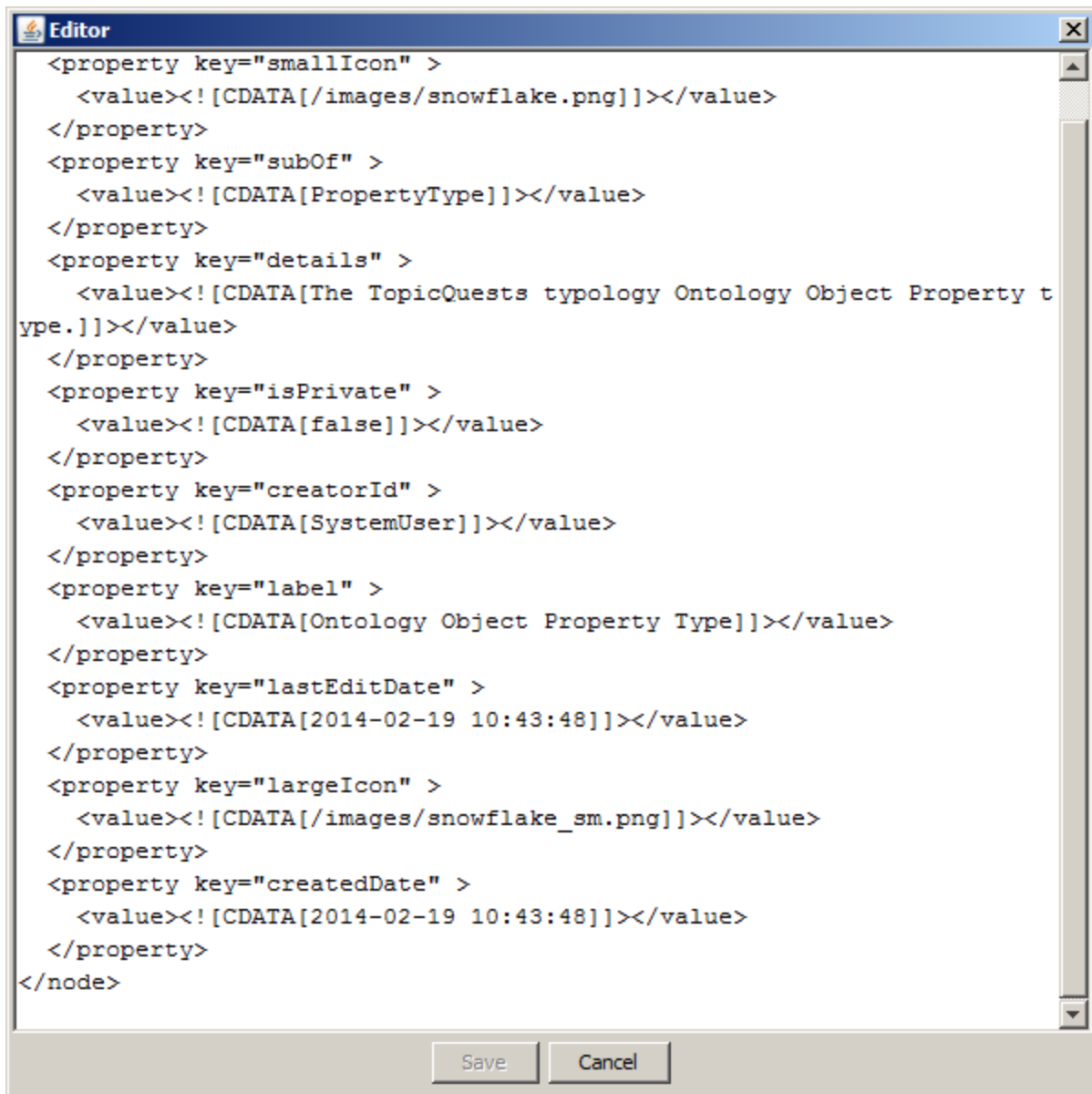


Figure 4. A Topic Ready To Edit