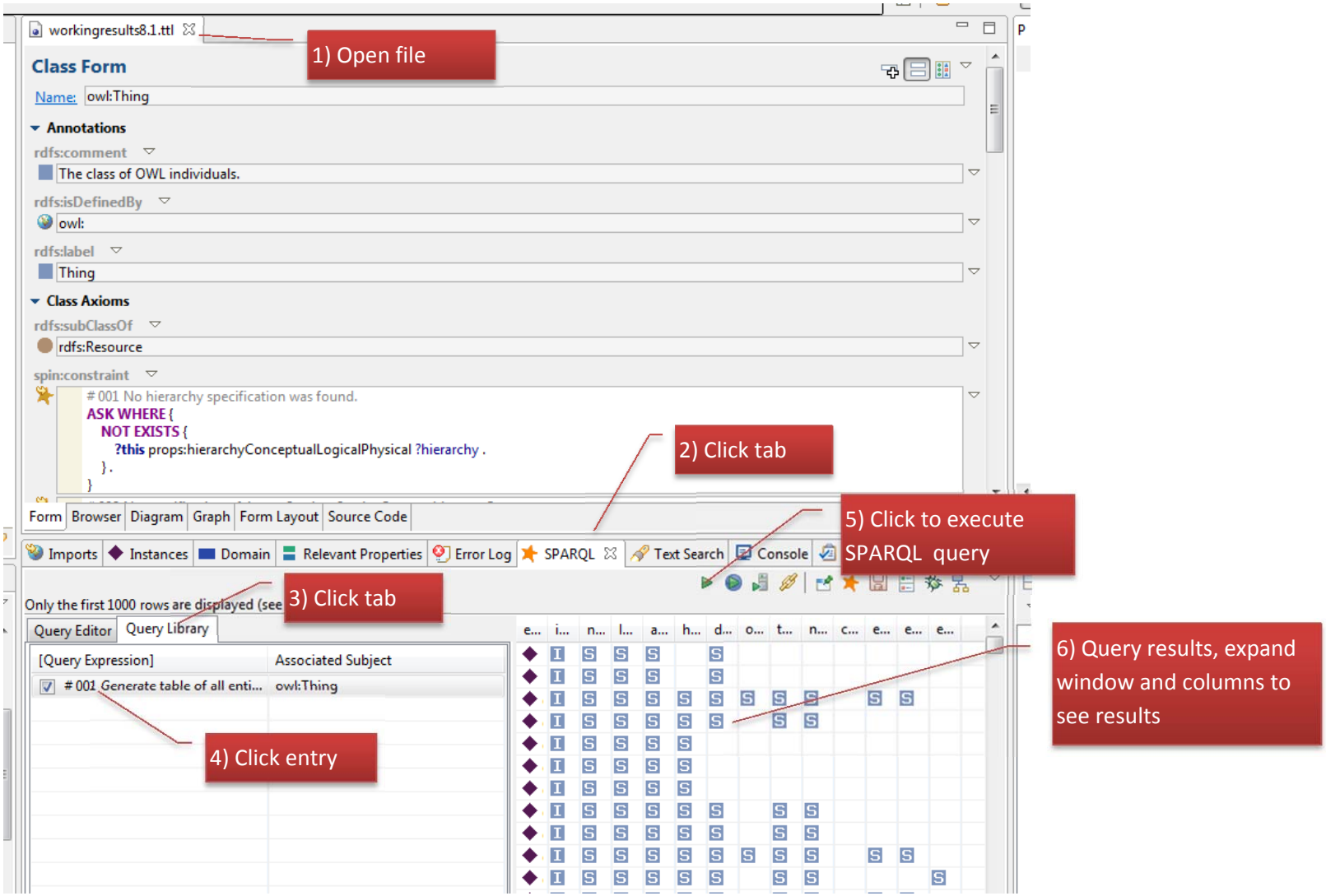
TBC-ME - Export to txt file (tab deliminited) – updated 03Aug2015

Script prepared by Steve Ray. Once the latest ADWP CAVES project GitHub private files are downloaded/cloned to your workstation, open e.g. 8.0/Data/workingresults8.1.ttl and follow the steps listed in Steve's email to ADWP participant team 15Jul2015 to generate the query, see screen snippet below.

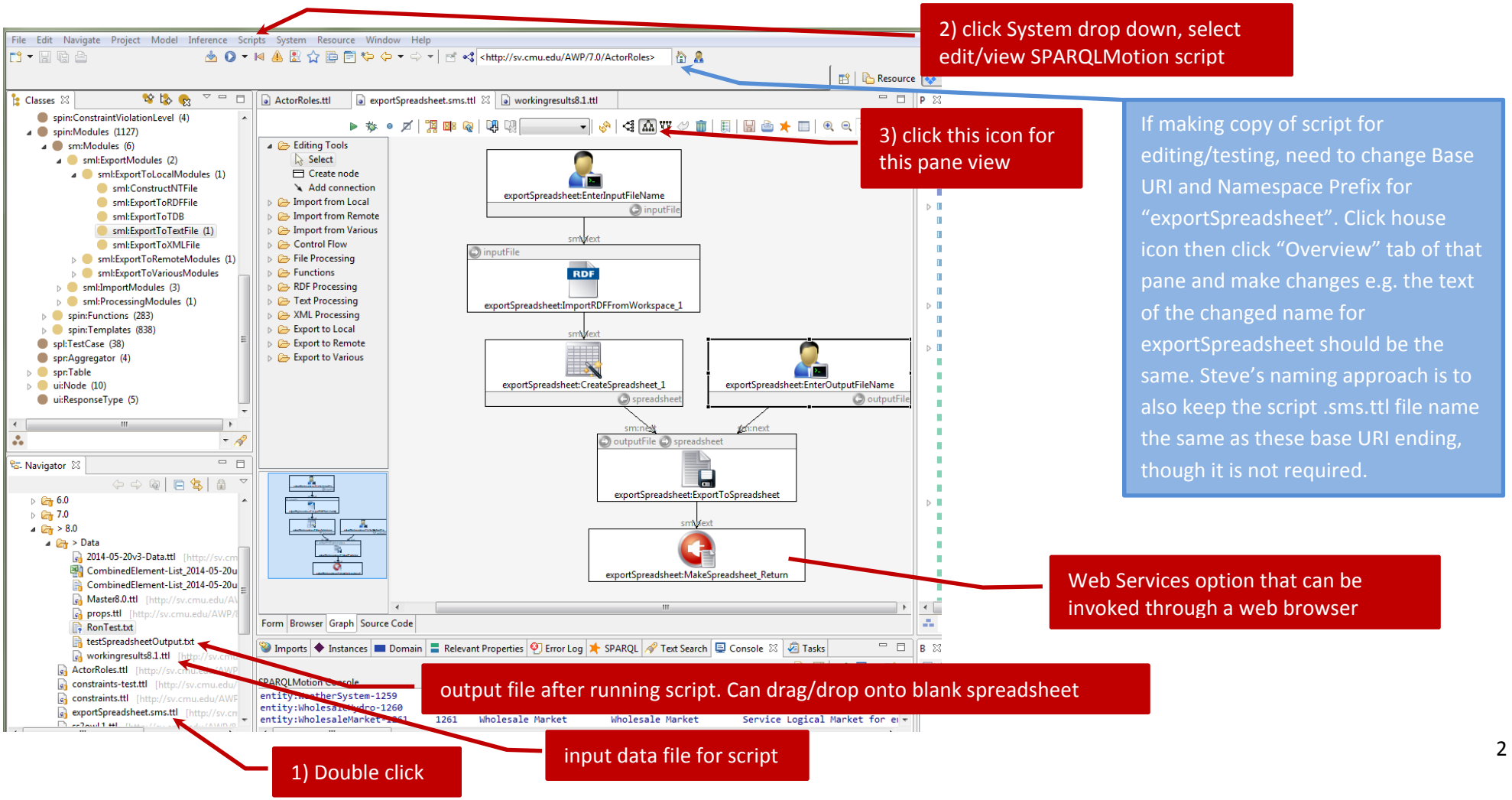


To look at the query, double click on the "OWL: Thing", note this differs from what Steve's 15Jul2015 email states. Steve later clarified he moved the constraints up to owl:Thing because he decided we should check constraints on the instances that were properly classified as well as the ones that weren't. Note that the

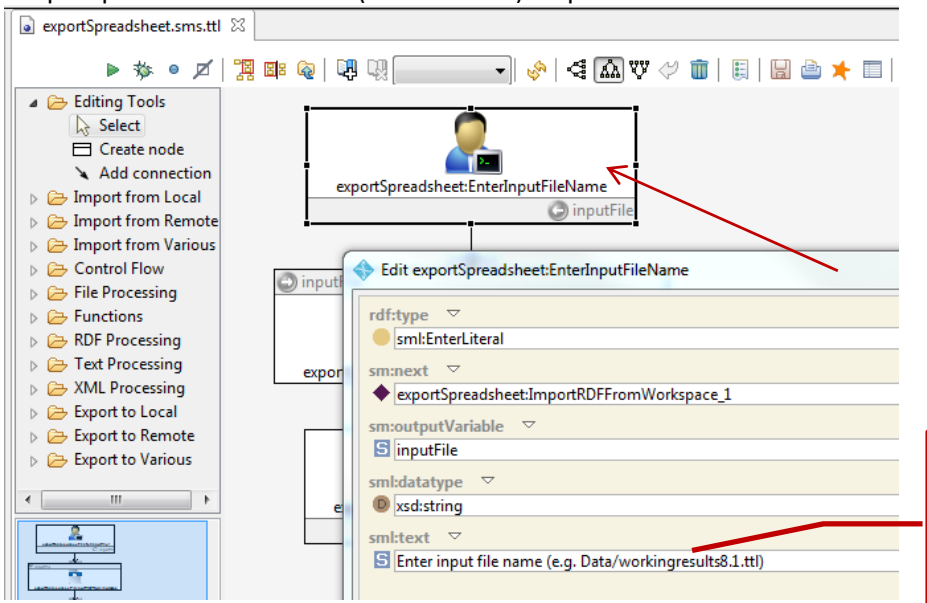
constraints are stored as values of spin:constraint. The query isn't – it's stored as the value of spin:query. In the spin query section. Steve also clarified that you don't need to do the query described above if you just want to use the SPARQLMotion script. The query for that is stored separately in the script .sms file.

Export Query Table results to Text Tab Delimited File:

The following are authors documenting of Steve's how to use the script to export the above query results to a Text Tab Delimined File via SPARQLMotion tool in TBC-ME. 1st, double click the "exportSpreadsheet.sms.ttl" file in the Navigator pane, then click on "Scripts" dropdown from top of window then click e.g. "Scripts > Edit/View SPARQLMotion script". The following screen should show up. Note: if you plan to edit the script, then make a copy of this script and rename it, e.g. with the target file not in edit mode, in Navigator pane, right-click on "exportSpreadsheet.sms.ttl", select "copy", then "paste" with a file name different from filename being copied. With that new script .ttl file open in TBC edit mode (alternately performing the necessary name changes within text edit mode can be performed prior to opening that new named file and helping avoid Base URI and other Namespace collisions), click the house icon at top of window, click the "Overview" tab in top center pane, then change the Base URI (Location) and Namespace Prefix entries per the blue shaded callout note below. Remember to press enter key to record the change (an Eclipse task).

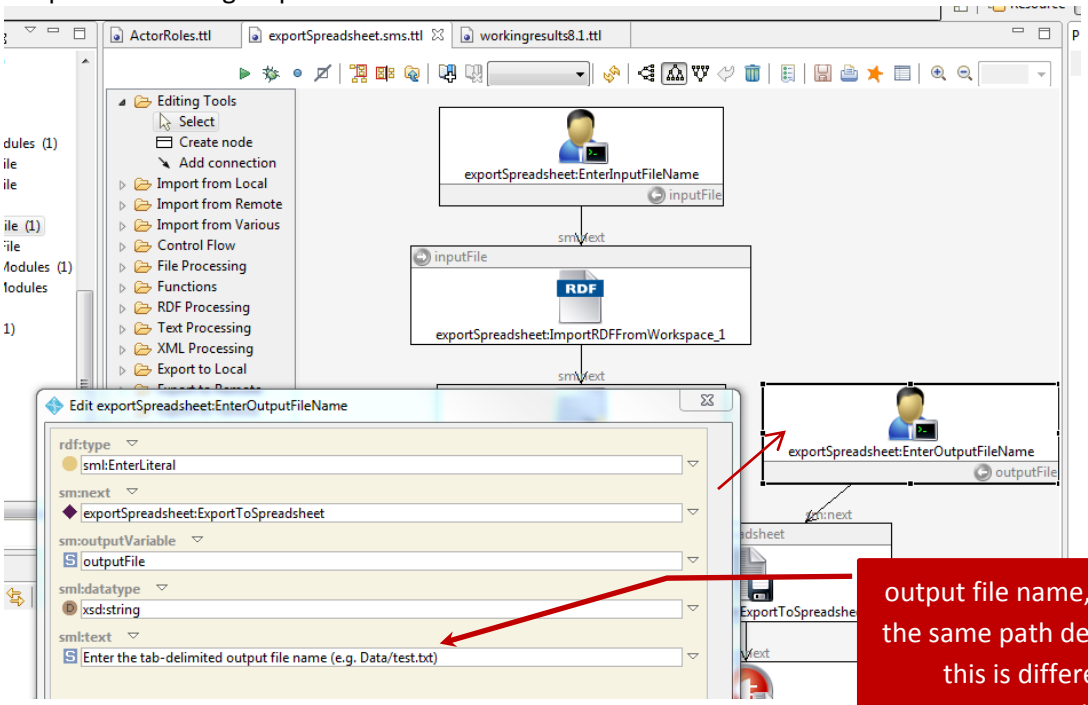


Script Input file identification (user defined) step



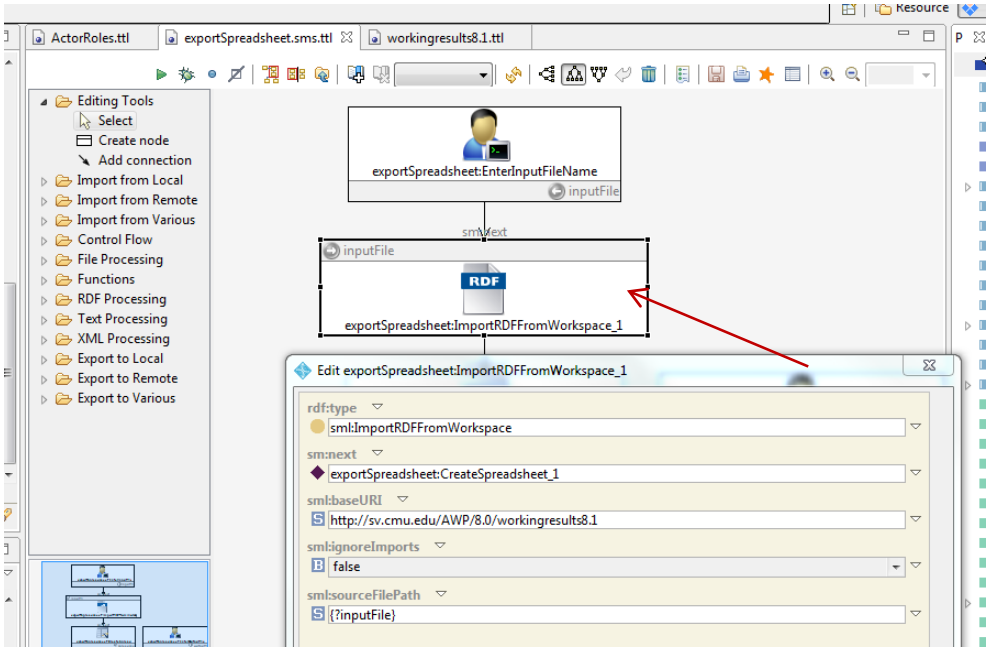
Source file input, text as shown in prompt. Must enter name in the same path depth as the example given in the prompt. Remember to press enter key, then click Next

Output File Naming step

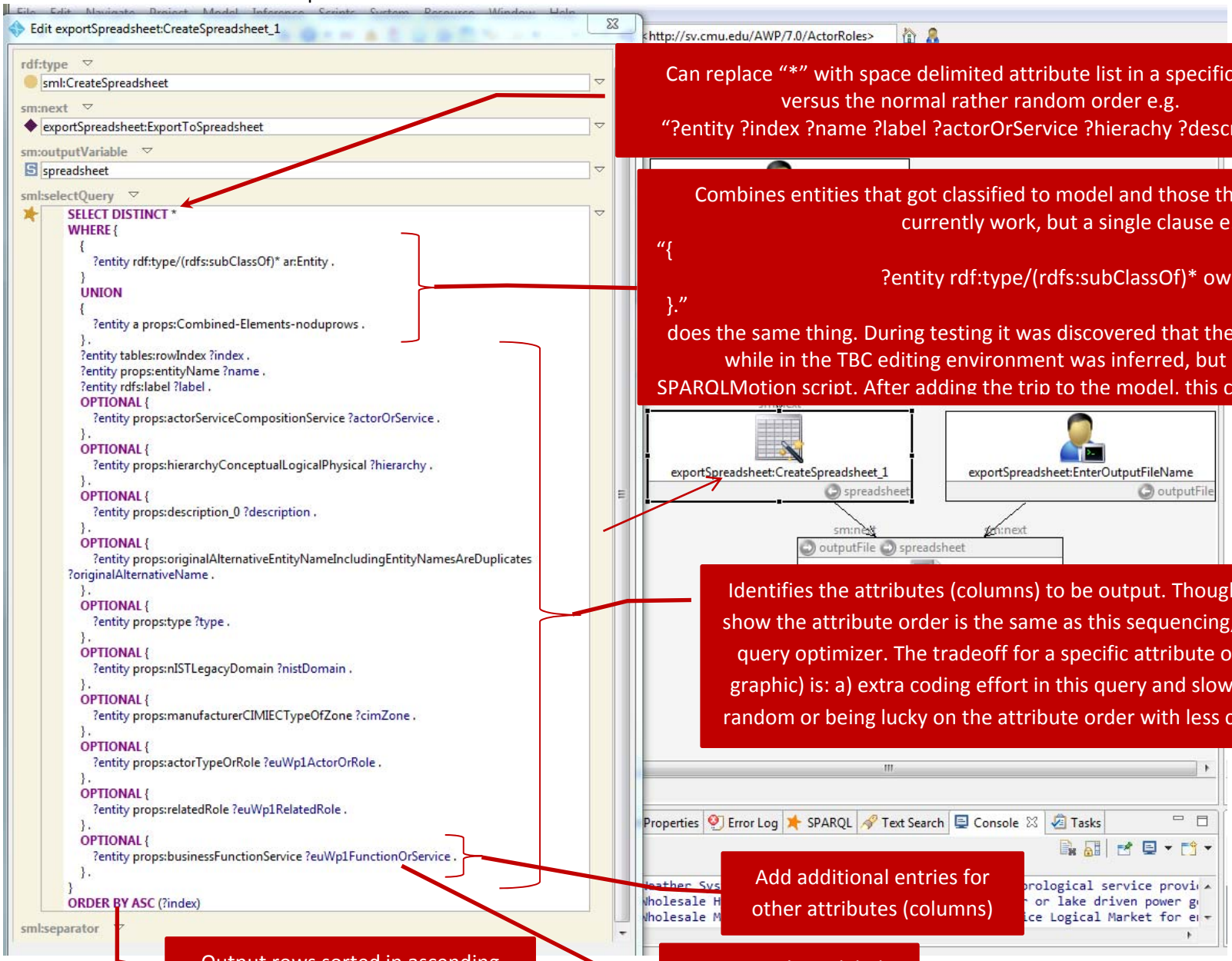


output file name, text as shown in prompt. Must enter name in the same path depth as the example given in the prompt. Note, this is different from the input file source path depth. Remember to press enter key, then click Next

RDF From Workspace Import step



Create table view of RFD data step



Can replace "*" with space delimited attribute list in a specific order, versus the normal rather random order e.g.
"?entity ?index ?name ?label ?actorOrService ?hierachy ?description"

Combines entities that got classified to model and those that did not. These clauses currently work, but a single clause e.g.

"{

?entity rdf:type/(rdfs:subClassOf)* owl:Thing .

}."

does the same thing. During testing it was discovered that there was a missing triple, that while in the TBC editing environment was inferred, but not while running the SPARQLMotion script. After adding the trip to the model, this clause produces same results.

Identifies the attributes (columns) to be output. Though initial TBC-ME script tests show the attribute order is the same as this sequencing, that order is subject to the query optimizer. The tradeoff for a specific attribute order (ref 1st callout on this graphic) is: a) extra coding effort in this query and slower execution speed, versus random or being lucky on the attribute order with less coding and faster execution.

Add additional entries for other attributes (columns)

Output column label

Output rows sorted in ascending order of the variable ?index values