SOAP and the Web of Science: how to bulk download query results

Eric Orace Johnson

Recently, we had a faculty member ask to download thousands of citation records from Web of Science (WOS)[1]. With modern tasks such as data mining for research, this bulk "non-consumptive" use of textual data is increasing. Because of the size of the download, she needed the library's help. Described below are the steps and tools needed so you can repeat the process with your own search terms. This process is also applicable to other Web services that use the simple object access protocol (SOAP).

She started by going to the WOS Web site and crafting her search, selecting search terms, timespan and database indices. The Web interface allows downloads of up to 500 records at a time, but with more than 20,000 records returned, she wanted an easier method.

So, she asked Thomson Reuters (the owners of WOS) for application programming interface (API) access so that she could download more records at a time. While our library subscribes to WOS, large data pulls like that require additional permission. Thomson Reuters was helpful, providing access and information about their API. An API is a set of procedures and terms that other computer programs can use to access a software program, website or database.

It turns out that their API uses SOAP. This API is an eXtensible markup language (XML) that can send search requests and receive results over the Web.

While we could have written a program from scratch to extract the data, we decided to leverage some free tools to speed up the development process. SOAPUI is a tool designed to test SOAP Web interfaces[2]. It allows the user (you) to write a script that interacts with a SOAP website. There are both free and paid versions. The paid version makes some aspects of the script writing easier, but we used the free version.

To interact with WOS using SOAP, there are several steps. First, authenticate as a valid user, run a search, then retrieve and save the search results. In SOAPUI, we created projects to hold steps for Authentication and Searching. To the Searching project, we added a "test suite" which allows us to run several actions in sequence. We then pulled the authentication routines into the test suite.

By running the Authenticate script and Search script, we were able to test the connection and our search terms. Then we began adding elements needed to automate the process.

When we retrieve records, we are allowed to specify the sort order and which record(s) we want returned. One of the restrictions is that each "retrieve" step is allowed only 100 records. But, you are allowed to have multiple retrieve steps one right after the other.

We could have added 200 individual "retrieve" steps, but decided it would be easier to simply "loop" through the step 200 times using a "Conditional Goto" step in SOAPUI. As the retrieve request specifies which records to send, we also added a step in the "Groovy Script" language used by SOAPUI to increment the retrieve request parameter each time.

Then we hit another restriction. The rate of requests is limited to 2 per second. So, we added a delay of half a second to the loop.

We finished the process by adding scripts to initialize the variables and close the Internet session.

The procedure of downloading 21,602 records took 12 min and created 217 text files. We also ended up creating a set of scripts that can easily be adapted for other faculty queries.

Here is a detailed step-by-step guide for setting up the process:

(1) In the Web interface using the Advanced Search, determine your search criteria. Notice that words not surrounded by quotation marks are treated as if they are OR terms. For this example, select a topic search of "anhydrous composite" or "plastic resin" or "fiberglass" with a timespan of 1925 to 2015 (Figure 1).

(2) If you attempt to download more than 500 items, a message will pop up. Click on "Learn how" to get to a form that lets you request to download WOS data (http://ip-science.interest.thomsonreuters.com/researchdatausage) (Figure 2).

(3) Thomson Reuters will send you a link to a document with more information and specific codes for downloading data.

(4) Download and install the SOAPUI program from www.soapui.org/downloads/soapui/open-source.html

(5) Create two new projects – Authentication and Search. For each project, you will enter a Web Services Description Language (WSDL) URL string. The reason we created two projects is because it was an easy way to enter this WDSL information. The Search WSDL URL for WOS is http://search.webofknowledge.com/esti/wokmws/ws/WokSearchLite?wsdl And for authenticate, it is http://search.webofknowledge.com/esti/wokmws/ws/WOKMWSAuthenticate?wsdl[3] (Figure 3).

Figure 1. *WOS search*

(6) Be sure to select "Stores all file paths in project relatively to project file" so that you can find the output files later (Figure 4).
(7) Save the project (Figure 5).
(8) When creating one of the projects, also select "Create TestSuite". That is where we will put our scripts. Remember to select "Stores all file paths in project relatively to project file" also (Figure 6).
(9) Select "Single TestCase with one Request for each Operation" (Figure 7).

Figure 2. *Download more than 500 records*

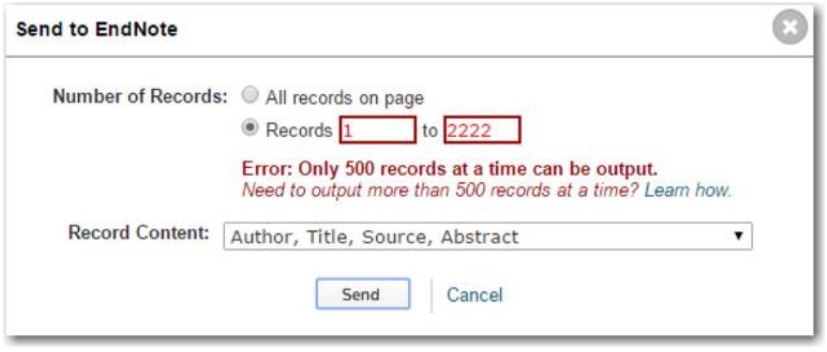


Figure 3. *Create new SOAP project*

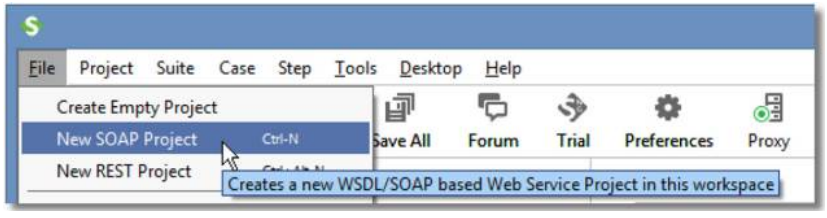
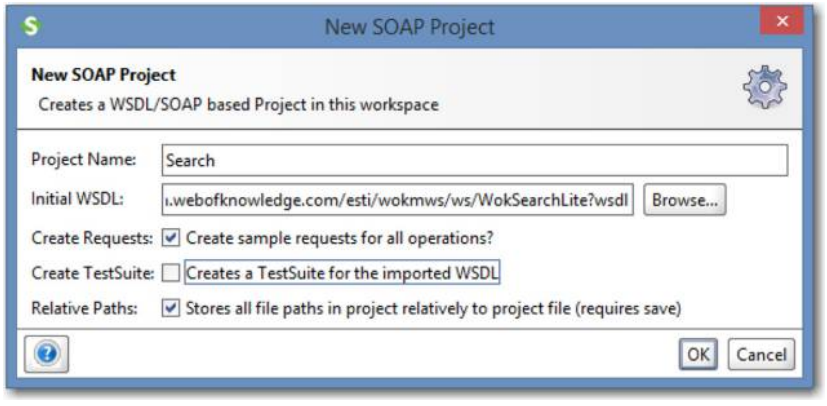


Figure 4. *Use relative paths*



(10) The default TestSuite name is okay (Figure 8).
(11) Save the project to the same directory (Figure 9).
(12) Save the projects periodically by pressing Ctrl-Alt-S, selecting the "Save All Projects" menu item under the File menu or clicking the "Save All" icon (Figure 10).
(13) If this is the first time the project has been saved, you will be presented with a "Save Project" message. This means that during project creation, you missed the checkbox to select relative file paths. Delete the projects and start over (Figure 11).
(14) This is how SOAPUI navigator window will look with the trees partly expanded (Figure 12).
(15) Double click on the Test Suite (which has a checkmark) to open it (Figure 13).
(16) Expand the projects tree in the "search" Project section and drag the "search Request 1" into the test suite below the other steps. You can also drag it to the TestSuite in the project tree (Figure 14).
(17) Then accept adding the request to the test case (Figure 15).
(18) The "Add Request to TestCase" message will pop up saying that it is missing required interfaces. Respond "Yes", and it will bring in the proper WSDL information for us (Figure 16).
(19) The defaults are sufficient for this project (Figure 17).
(20) Drag the "search – Request 1" line and drop it between the "authenticate" and "close session" lines.
(21) The Test Suite now has a search step (Figure 18).
(22) In the Test Suite, click on the gear icon and make sure "Maintain HTTP session" is checked (Figure 19).
(23) Checking "Maintain HTTP session" keeps the session ID active after authentication so that subsequent SOAP requests

Figure 5. *Save project "search"*

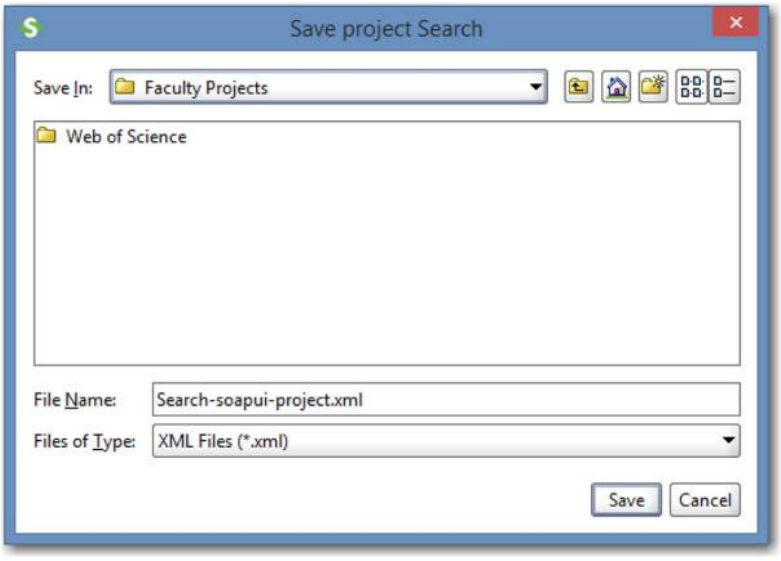


Figure 6. *Create TestSuite*

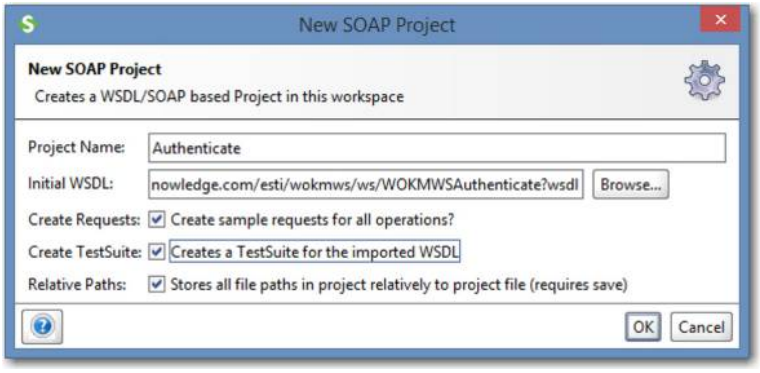


Figure 7. *Use "Single TestCase" mode*

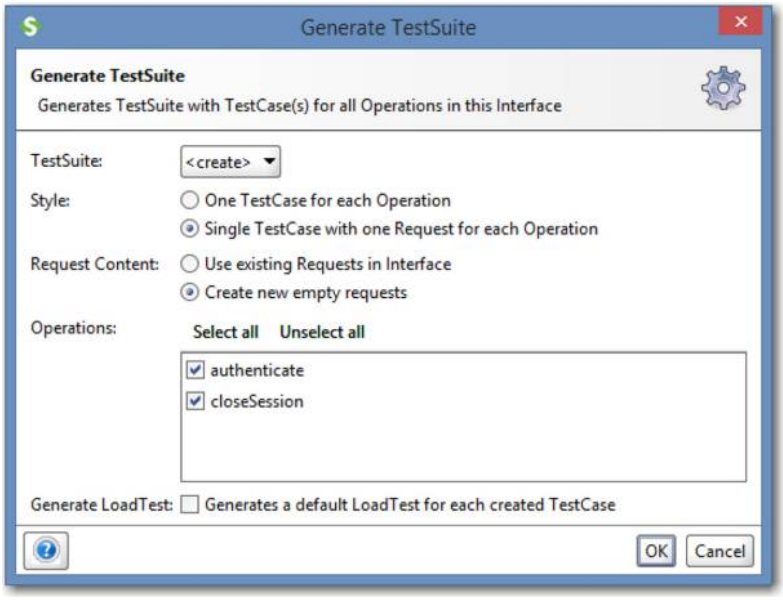
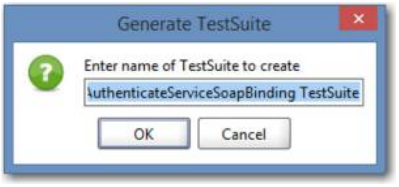


Figure 8. *Use the default TestSuite name*



in the test case run are accepted (Figure 20).

(24) To examine the Test Suite steps, double click on "authenticate" (Figure 21).

(25) If you click on the green arrow, it will run the process and return a session ID from the WOS, displaying the results in the right side window (Figure 22).

(26) There is nothing we need to do with the authenticate file so close it by clicking the boxed × in the upper right corner of the window (Figure 23).

(27) Double click the "search – Request 1" step (Figure 24).

(28) The question marks are places we need to add information (or delete the line). They are:
- databaseID;
- userQuery;
- collection;
- edition;
- symbolic TimeSpan;
- begin;
- end;
- query Language;
- first Record;
- count;
- name; and
- sort.

(29) The database ID and collection are both "WOS" for Web of Science[4].

(30) The user Query is based on what we entered in the Web interface. For example, if we searched for "anhydrous composite" or "plastic resin" or "fiberglass" in topics, we need to rewrite it a bit adding the Topic field indicator of TS (or TO). *TS = "anhydrous composite" or TS = "plastic resin" or TS = "fiberglass"*[5].

Figure 9. *Save project "Authenticate"*

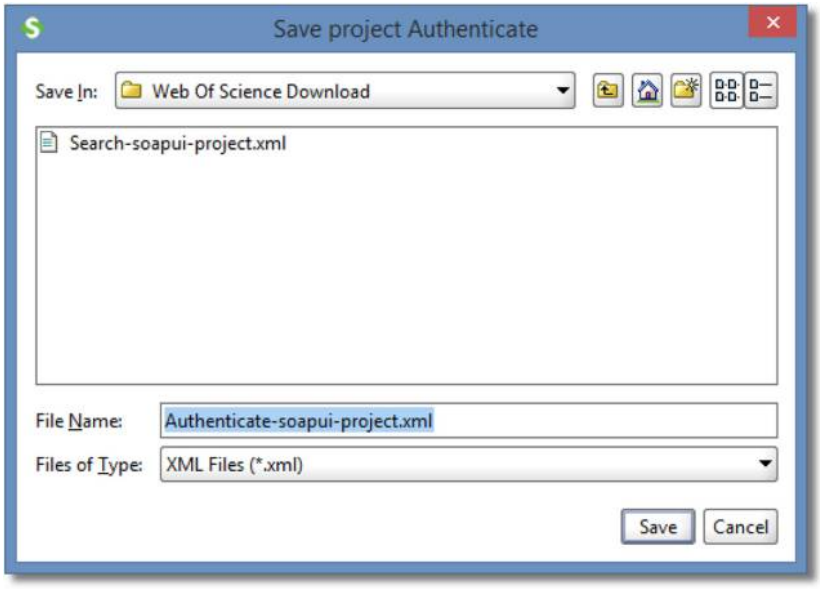


Figure 10. *Save all projects*

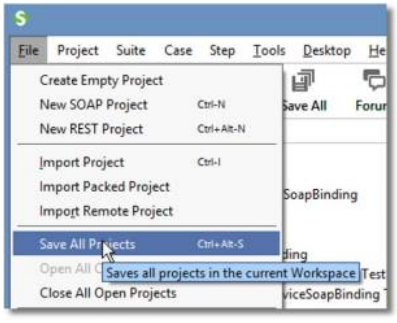
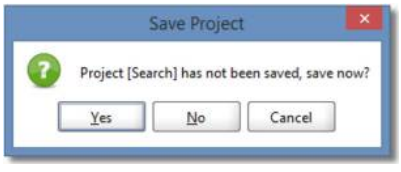


Figure 11. *Save project – relative paths error*



(31) Edition is the index used[4]. If blank, all indexes will be searched.
(32) Symbolic Time Span is a range of dates when the information was loaded into the database. We deleted this optional element.
(33) Begin, End – This is the range of publication dates, formatted as YYYY-MM-DD.
(34) Query Language is always "en".
(35) First record, count. The search request can immediately return some records if you want. We used first record and count just to verify that our query was returning what we expected.
(36) Name – we used "AU" to sort by author[6]. Not all of the search fields are available for sorting.
(37) Sort – and we used "A" to sort ascending. Use "D" for descending.
(38) Here is a resulting XML search command:

```
<soapenv:Envelope xmlns:soapenv="http://
    schemas.xmlsoap.org/soap/envelope/"
xmlns:wok="http://wokSearchLite.v3.
wokmws.thomsonreuters.com">
  <soapenv:Header/>
  <soapenv:Body>
    <wok:search>
      <queryParameters>
        <databaseId>WOS</databaseId>
```

Figure 12. *SOAPUI navigator*

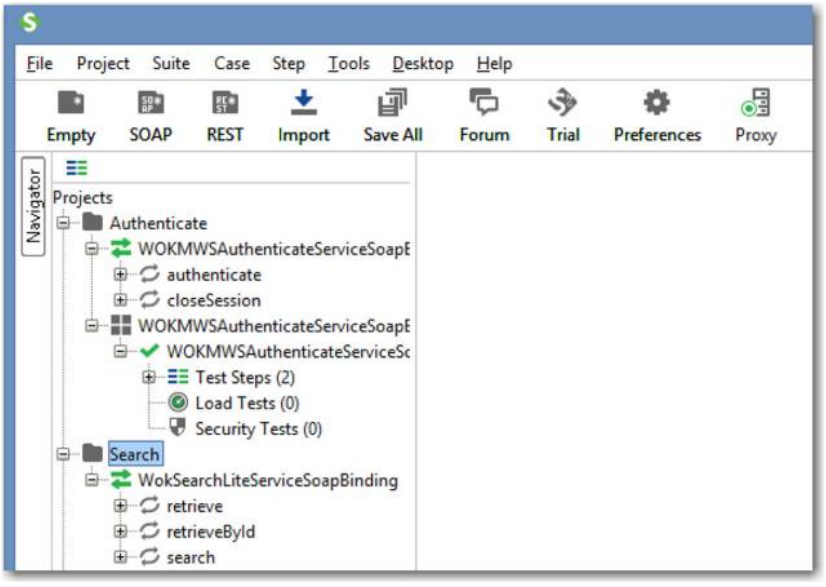


Figure 13. *Open TestSuite*

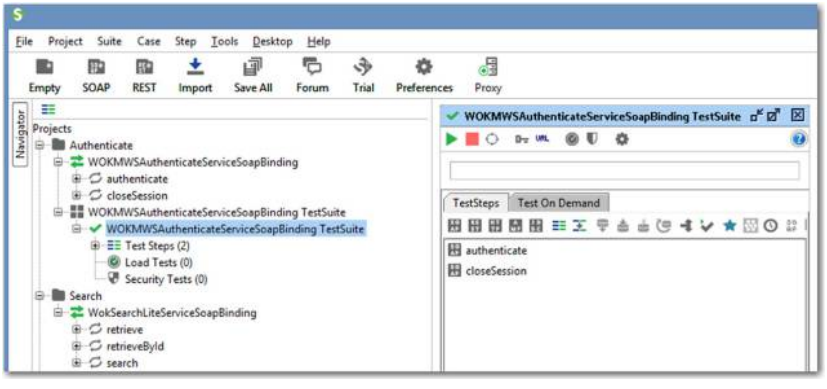


Figure 14. *Drag "search Request 1" to the test suite*

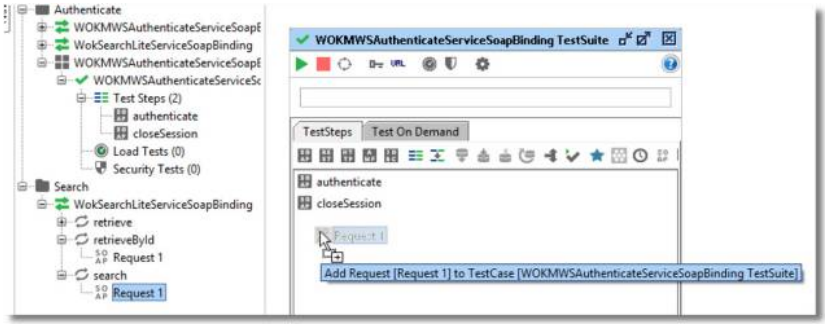


Figure 15. *Add request to TestCase*

Figure 16. *Automatically add WDSL information*

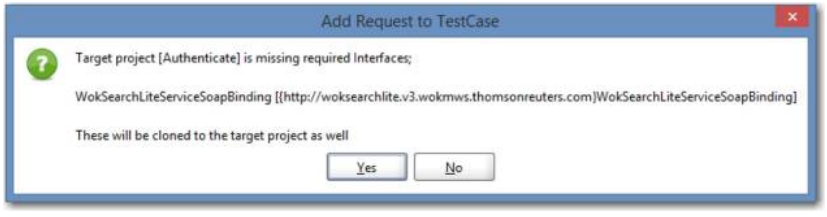
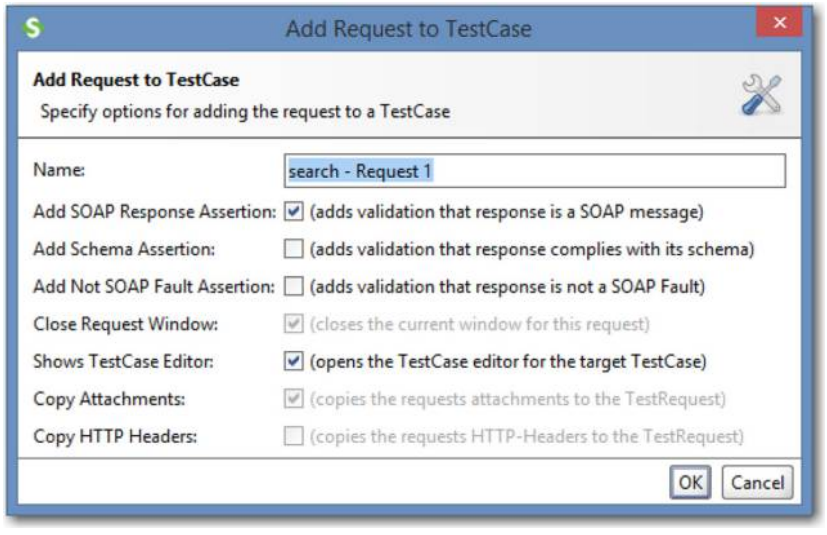


Figure 17. *Use defaults for adding request to the test case*




```
<userQuery>TS="anhydrous
  composite" OR TS="PLASTIC
  RESIN" OR TS="fiberglass"
  </userQuery>
<timeSpan>
  <begin>1925-01-01</begin>
    <end>2015-12-31</end>
  </timeSpan>
  <queryLanguage>EN</queryLanguage>
</queryParameters>
<retrieveParameters>
  <firstRecord>1</firstRecord>
```

Figure 18. *Position "search – Request 1" between authenticate and closeSession*

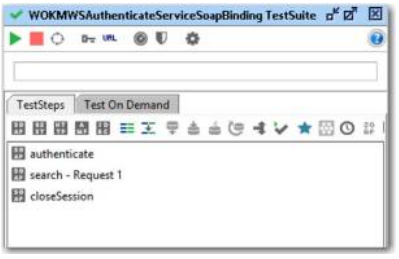
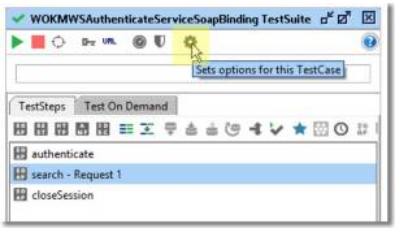


Figure 19. *The test suite "gear" icon*




```
    <count>2</count>
  </retrieveParameters>
 </wok:search>
</soapenv:Body>
</soapenv:Envelope>
```

(39) Clicking on the green arrow at the top of the "search – Request 1" screen will return an error in the right hand panel. "Session identifier cookie cannot be null [. . .]".

(40) So, go to the Test Suite window and click on its green arrow to run the authentication step along with the search step.

(41) If you still get a "Session identifier cookie [. . .]" error, you need to click on the gear icon and make sure "Maintain HTTP session" is selected.

If everything worked, you should have results like this in the window for the search step:

```
<soap:Envelope xmlns:soap="http://
   schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
  <ns2:searchResponse xmlns:ns2=
      "http://woksearchlite.v3.wokmws.
      thomsonreuters.com">
   <return>
    <queryId>1</queryId>
    <recordsFound>3057</recordsFound>
    <recordsSearched>60167018
      </recordsSearched>
    <records>
    <uid>WOS:000317246400088
      </uid>
     <Title>
      <label>Title</label>
      <value>Determination of Peroxide
       Value in the Plastic
       Resin</value>
```

Figure 20. *Maintain HTTP session*

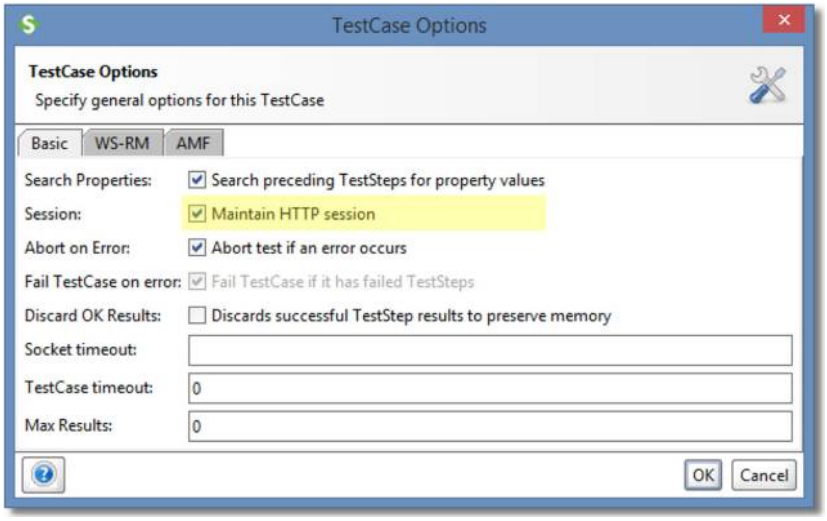


Figure 21. *Authenticate routine before running*

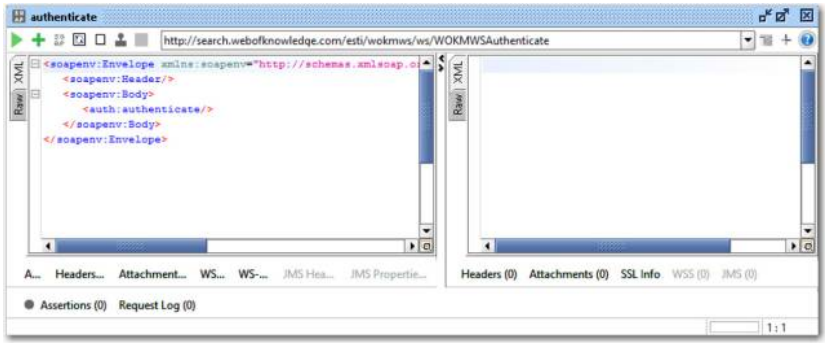


Figure 22. *Authenticate routine after running*


```
</Title>
  <label>Title</label>
  <value>Determination of Peroxide
    Value. . . .
```

This search has a queryid of 1 and found 3,057 matching records.

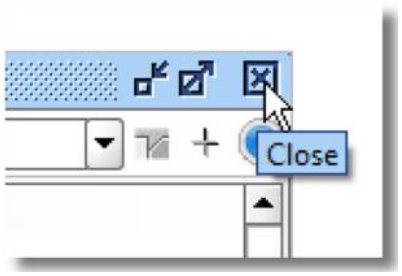
(42) Now, we need to begin retrieving records. Expand the Projects tree and drag the "retrieve – Request 1" line into the search suite (Figures 26 and 27).

(43) Then drag the retrieve step into position after the "search – Request 1" step (Figure 28).

(44) Double clicking on the retrieve step will open its window. There are question marks where we need to add parameter values:

- queryID;
- first Record;
- count;
- name; and
- sort.

Figure 23. *Close the authenticate routine*



(45) The query ID will be 1, as we have only one query. It can be seen in the results of the search step.

(46) We can set the first Record variable to 1 and the count to 100 to retrieve the first 100 records. To retrieve the second 100 records, the "first Record" variable will need to be increased to 101. For testing, we can set the count to 2 records.

(47) Name is AU for the author field and sort is A for ascending.

(48) After replacing the question marks in the retrieve step, you can click on the green arrow in the *Test Suite* window to see the results in the *retrieve* window.

(49) If you left a question mark, there will be an "Unmarshalling" error. Correct it and try again (Figure 29).

(50) To save the information that is returned, with the *retrieve* window highlighted, look at the lower left of the main SOAPUI window to where it says "Test Request Properties". Scroll most of the way down to find the property named "Dump File". In the Value cell to the right, enter a file name. Each time the retrieve step is run, it will save the output to this file, replacing anything that may have been in the file before (Figure 30).

(51) Put a sample name like "testSOAPUI.txt" in this area, run the test suite and then find the file on your computer. As you selected relative paths for the project, it will be saved to the same directory where you saved the TestSuite project. Open the file to see that it is a copy of the retrieve step results (Figure 31).

(52) As a general note, be sure to save your project often during this process (Control-Alt-S, under the file or project menu or the "Save All" icon) so that your work is not lost.

(53) To add the automation, we need to loop through the retrieve step many times, changing the firstRecord entry and dump file name each time.

(54) We need some parameters to keep track of the download progress. Add a step to the test suite to initialize them. In the TestSuite window, click on the search test step, then click the Star to add a "Groovy Script" step. Name the step "Initialize". In the Initialize step window that pops up, add the code:

Figure 24. *Search routine before running*

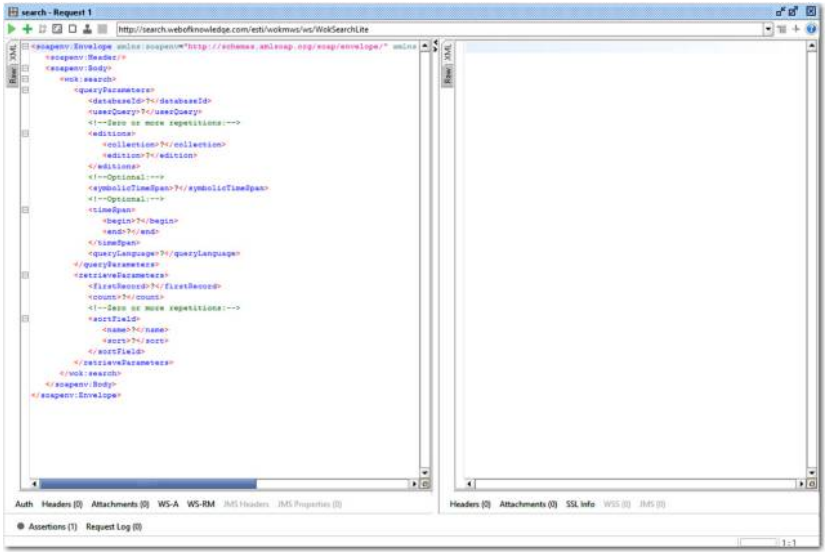


Figure 25. *Run Authenticate and search routines*


```
// set initial values
testRunner.testCase.setProperty
   Value("RetrieveStart","1")
testRunner.testCase.setProperty
   Value("BatchSize","2")
testRunner.testCase.setProperty
   Value("NumberOfRecords","5")
```

We are starting with a small batch size and number of records to test the script. After it runs well, we will increase the batch size to 100 and the number of records to the "recordsFound" value returned by the search request.

(55) Click on the green arrow in the "Groovy Script – Initialize" window to run the script. It will create variables and set the beginning values. Click on the Properties section of the TestSuite window to see those parameters (Figure 32).

(56) Next, we need to change the values after each retrieve step. To do this, we will use another Groovy Script that adds the batch size to the first record to retrieve

Figure 26. *Add Request to TestCase options*

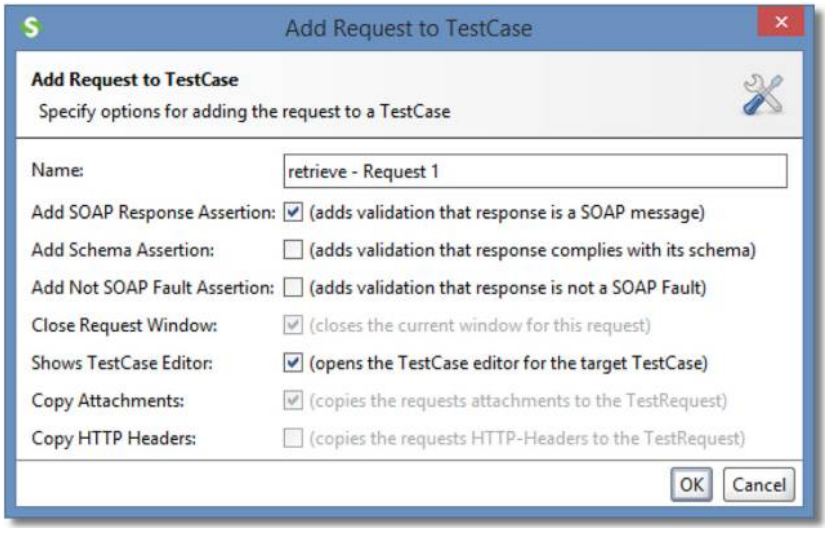


Figure 27. *Retrieve – Request 1 when first placed in the test suite*

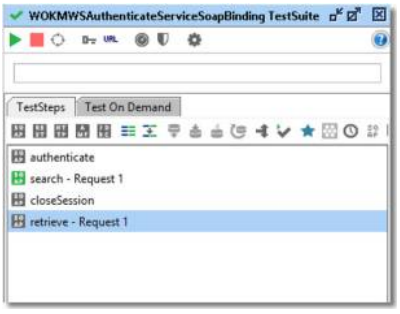
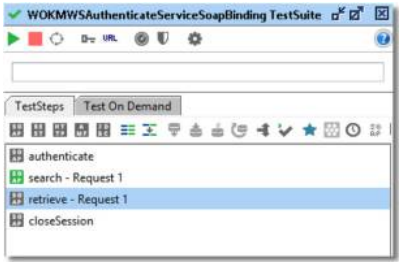


Figure 28. *Retrieve – Request 1 moved into position*



value. Click the Star again to create a Groovy Script named "Increment" and move it to after the retrieve step.

(57) In the increment script, add:

```
// retrieve start variable,
   convert string to value so
   we can increment it
int b = testRunner.testCase.
   getPropertyValue
   ("RetrieveStart").toInteger()
// increment start by Batch
   Size (Set in test case
   properties)
b = b + testRunner.testCase.get
   PropertyValue("BatchSize").
   toInteger()
/// need to convert back to
   a string to save the value
// save new value
testRunner.testCase.setProperty
   Value("RetrieveStart",
   b.toString())
```

(58) Each time you click on the green arrow in the Increment step to test it, the Retrieve Start property will be increased by the BatchSize amount as you can see in the properties window (Figure 33).

(59) You can reset the values by running the Initialize step.

(60) If it works correctly, click "Save All" to save your scripts.

Figure 29. *Unmarshalling error*

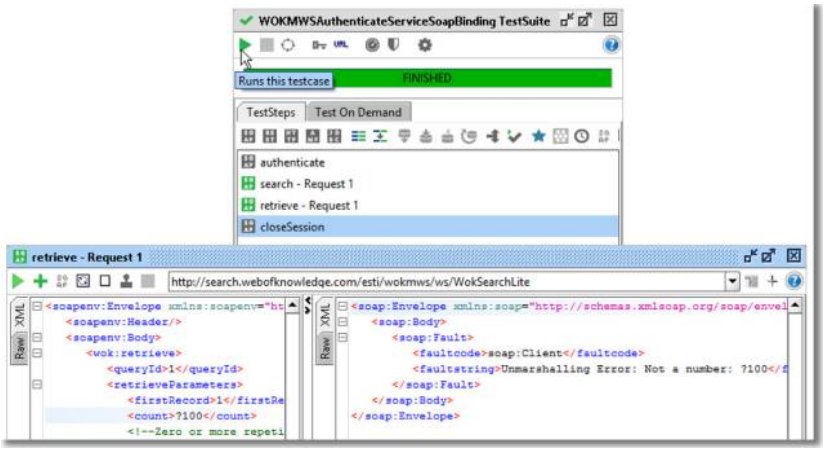


Figure 30. *Dump file name location*

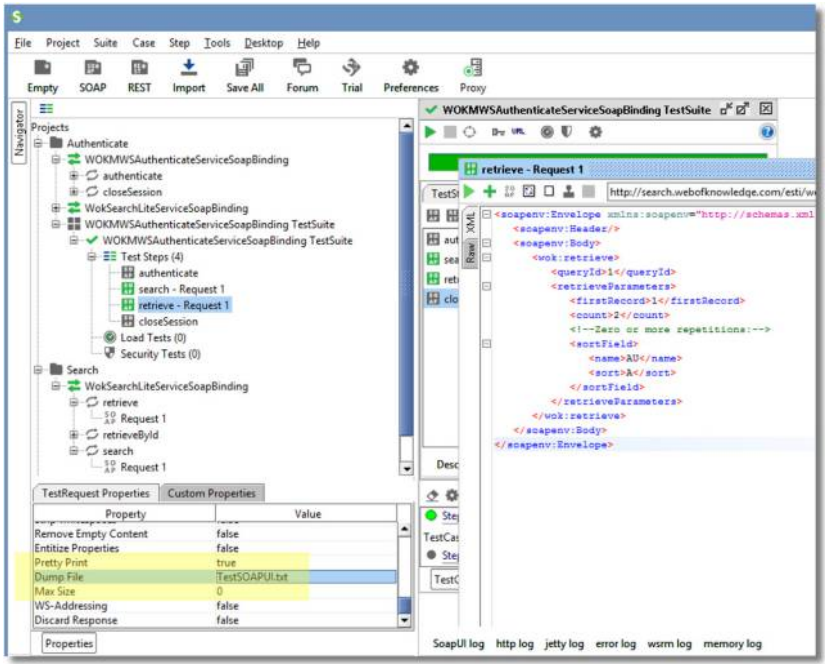


Figure 31. *Dump file in file structure*

(61) We need to add a loop so that the retrieve step is repeated. In the TestSuite, click on the icon to the left of the check mark to add a Conditional Goto TestStep. Name it Goto and move it to just above "closeSession" (Figure 34).

(62) In the Conditional Goto window, click on the green plus to add a condition. Name it whatever you want. In the "Condition XPath Expression" section of the Conditional Goto step, add:

```
${#TestCase#RetrieveStart}
  < (${#TestCase#NumberOfRecords}
  + ${#TestCase#BatchSize})
```

(63) In the "Target step" section, select the retrieve step (Figure 35).

(64) This will compare the current firstRecord value to the total number of records and batch size. If we have not retrieved them all, it will do the retrieve step again.

(65) The retrieve step needs to be written to use these parameters. Open the retrieve step and find the line that sets the first record. Change it to:

```
<firstRecord>${=(${#TestCase#
   RetrieveStart}).toInteger()}
   </firstRecord>
```

(66) Change the line for count to:

```
<count>${=(${#TestCase#
   BatchSize}).toInteger()}
   </count>
```

This will cause the retrieve process to use parameter values which change each time.

(67) We also need to change the dump file name. Select the retrieve step, then look at the "TestRequest Properties" in the lower left of the SOAPUI window. Scroll down and find the Property "Dump File". Change the value to:

```
Output-${#TestCase#
   RetrieveStart}.txt
```

This will create a file name that changes each time the retrieve step runs. You can change the words "Output-" and ".txt" to whatever you want (Figure 36).

(68) Test the file name by running the whole test suite (click on the green arrow in TestSuite window) and verify that new files of those names were created. Delete these output files before your final run, as they are not needed (Figure 37).

Figure 32. *TestSuite properties*

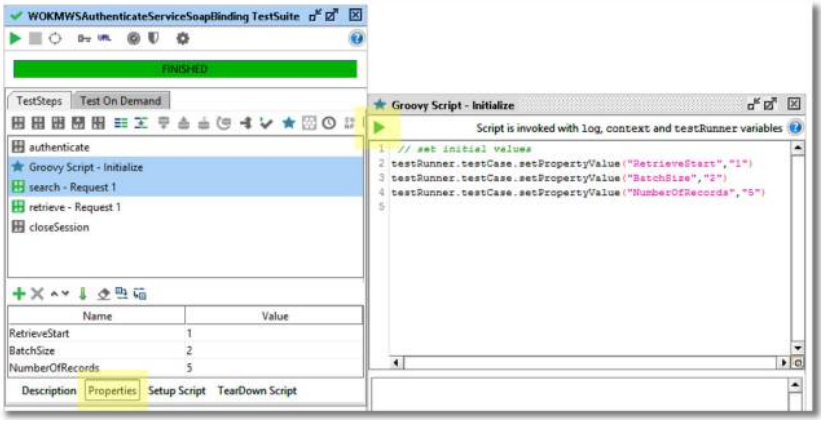


Figure 33. *Properties change with each run*

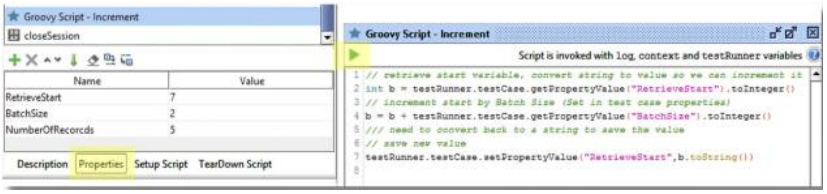
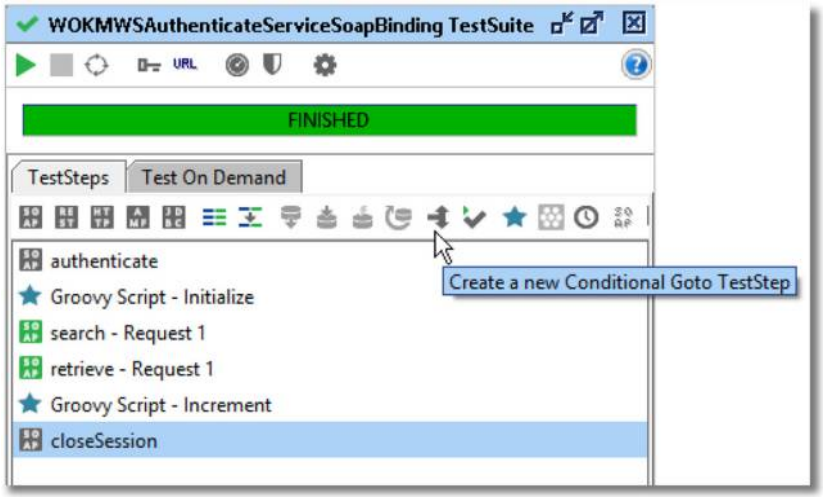


Figure 34. *Create conditional GoTo statement*



(69) One of the problems is that we are limited to two retrieves per second. Depending on the speed of your computer and network, if you look at the results in the retrieve window of the last file saved, you may find details about exceeding the speed limit. As our script can run faster than is polite to others, we need to slow it down. Even if the speed problem did not happen on your test run, we do not want it leaving glitches in the full run. Add a delay step by selecting "Conditional Goto" in the TestSuite and clicking on the Clock icon to create a new Delay TestStep (Figure 38).

(70) You may accept the default of 1,000 ms (1 s) or double click on the test step to set it to a different delay.

(71) Press "Save All", then test the setup by clicking the green arrow in the TestSuite window. If everything works, it is time to run the final process.

(72) Double click the Search step to locate the number of records found. Double click the initialize step and set Number of Records to the value from the search step. Set BatchSize to 100 (or how many records you want in each output file) (Figure 39).

Figure 35. *In the GoTo, select the retrieve step*

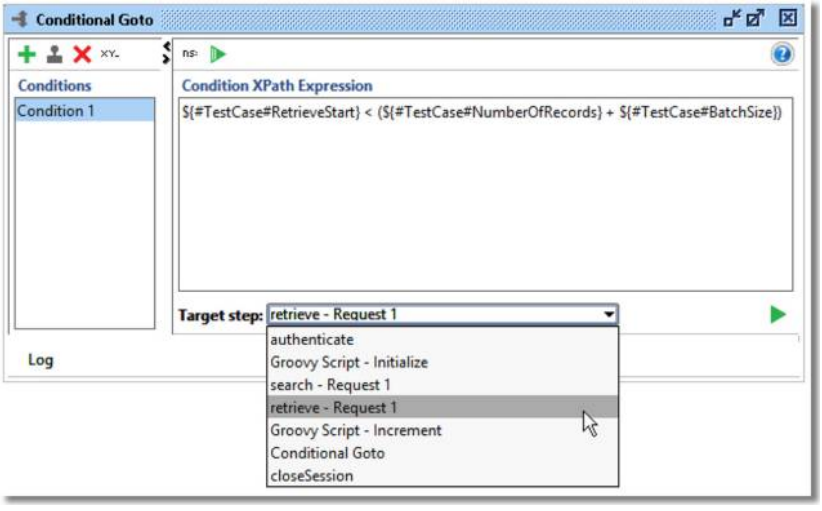


Figure 36. *Automatic dump file naming*

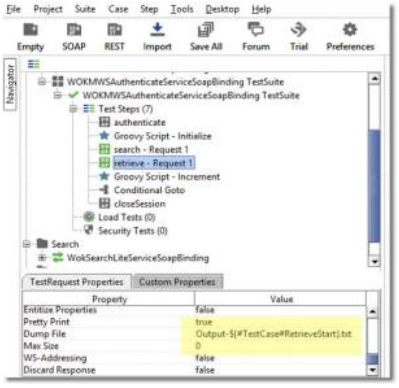
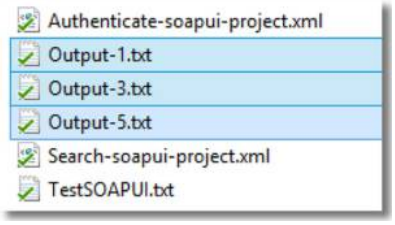


Figure 37. *Automatically named dump files in the file structure*



(73) Click on the green arrow in the TestSuite window.

(74) The resulting text files will be placed in the same directory as your project.

(75) The last file will have an error message, "..input is invalid [RetrieveParameter firstRecord: 3,101 exceeds recordsFound: 3,057 [. . .]". This indicates that we downloaded all the records available and that the last file can be deleted.

This process can be used for other searches in WOS by changing the search terms. Test new search terms on the Web site to be sure they will return the desired type of results. You can also download data from other repositories that use SOAP. Just change the WSDL URLs and read the messages returned by the Web site to adapt the scripts as needed.

NOTES

1. "Web of Science [v.5.18] – All Databases Home", available at: http://apps.webofknowledge.com/UA_GeneralSearch_input.do?product=UA&search_mode=Gener alSearch&SID=1AEhWdEwxk79DemQHTt&preferencesSaved= (accessed 10 August 2015).

Figure 38. *Create delay step*

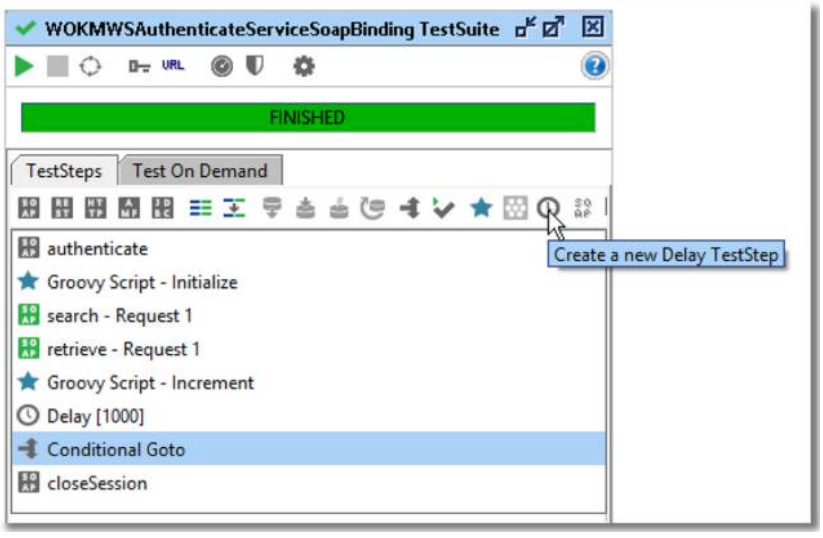


Figure 39. *Set number of records*

2. "SoapUI – The Home of Functional Testing", available at: www.soapui.org/ (accessed 10 August 2015).
3. "ISI Web of Knowledge Web Services, version 2.0", available at: http://science.thomsonreuters.com/tutorials/wsp_docs/soap/Guide/overview.html (accessed 10 August 2015).
4. "ISI Web of Knowledge Web Services, version 2.0", available at: http://science.thomsonreuters.com/tutorials/wsp_docs/soap/WokSearch/db_editions.html (accessed 10 August 2015).
5. "ISI Web of Knowledge Web Services, version 2.0", available at: http://science.thomsonreuters.com/tutorials/wsp_docs/soap/WokSearch/userquery/wos_userquery.html (accessed 10 August 2015).
6. "Web of science help", available at: http://images.webofknowledge.com/WOK46/help/WOS/h_fieldtags.html (accessed 10 August 2015).

Eric Orace Johnson (*johnsoeo@miamioh.edu*) is Numeric and Spatial Data Services Librarian at Miami University, Oxford, Ohio, USA.