
Exchanger XML Editor - Using XQuery

Copyright © 2005 Cladonia Ltd

Table of Contents

Query	2
Execute XQuery	2
Database Access	2

Exchanger XML Editor - Using XQuery

Query

Exchanger XML Editor currently supports the XQuery language, as implemented in release 8.4 of Saxon (see <http://saxon.sourceforge.net>).

Execute XQuery

The XQuery project contains a sample input file `projects/XQuery/input/books.xml` containing a list of books with their titles and authors and also a sample query file `projects/XQuery/query/books.xq`. The query in this case sorts the books in order of the length of their titles. Select **Transform->Execute XQuery** and enter the appropriate input and query files. The output of the transformation can be sent to a new window or saved to file. Notice that in the resultant document, the books have now been sorted with the shortest title first.

Database Access

The latest Saxon XSL processor provides extensions that allow data to be retrieved from database tables. Database drivers must be added manually by adding the jars in the **System** tab of **File->Preferences** - this requires a restart of the application to take affect.

The built-in sample is for the MySQL database and performs a simple SQL Select-From-Where query. To test the database access, install the appropriate jar, edit the existing Scenario by selecting **Transform->Manage Scenario**, choose "Simple Database Access" and press the **Change** button. Click on the **Parameters...** button in the XSLT section of the dialog to see the current settings. The default driver details are for MySQL so, if necessary, modify this parameter to match the driver for your database. Now change the mandatory database name, user, password and table parameters to suit your setup. Other optional parameters include "column" which is a comma separated list of the columns that should be returned by the query - by default all (*) will be returned. Also a "where" condition can be specified to narrow the search criteria.

Save the changes you have made to the Scenario and then execute it using **Transform->Execute Scenario**. If the connection is successful, the query will return data in a new window - otherwise, check the Output tab for information on possible error conditions.

Note that the XSLT 2.0 version of the Saxon processor must be used to avail of this functionality (see the **Processor...** section of the Scenario dialog). More information on using the Saxon sql extensions can be found in the `sql-extension.html` file in the doc directory.