Brief Overview

This technical note provides the step-by-step instructions necessary to integrate the xqDoc web service on to your XQuery development workbench. We view this project as beneficial since it will help keep your XQuery code and xqDoc documentation synchronized. For example, imagine changing one of your XQuery modules and then have the xqDoc xml for that module automatically generated behind the scenes. This was the fundamental motivation for this project.

This project leverages the MarkLogic Trigger feature and introduces a general purpose core library module for accessing web services via SOAP. Since the current MarkLogic Trigger feature is 'lightly' documented, this project (by offering numerous trigger examples) could help those developers attempting to leverage this new and powerful capability. Similarly, since the SOAP library module is very generic, it could be utilized as a starting point for accessing other web services (beyond xqDoc).

Dependencies and Assumptions

1. The following instructions require that MarkLogic CQ (version 2.2 or 3.0), available from xqZone, be configured for the MarkLogic installation. The subsequent installation steps make the following assumptions concerning the CQ installation.

    - The MarkLogic CQ http server root is 'cq'. If the root is not 'cq', then change the $CQ-ROOT variable in the install.xqy and uninstall.xqy files.
    - The MarkLogic CQ http server has been configured with 'application-level' authentication and the default user has 'admin' privileges.

2. MarkLogic 3.0+ (required for the Trigger functionality)

3. Internet connection (required for accessing the xqDoc web service)

Installation Steps

1. Unzip the contents of the xqDoc-ws.zip into the root directory for the CQ http server. The root directory would be the 'cq' directory under the MarkLogic installation.

2. Execute the install script accessible from the following url. This assumes that CQ is available on localhost and port 8007. If this is not the case, use the appropriate machine and port.

    http://localhost:8007/xqDoc-ws/install.xqy

    You should receive the following message:

    "The xqDoc web service modules and triggers have been successfully installed."

    To learn more about the install script, review the documentation contained in the install.xqy main module.

3. Create a database (and forest) to store the generated xqDoc xml returned from the xqDoc web service. Use all of the default options when creating the database (and forest). In the remaining steps, we will assume this database has been named 'xqDoc'.

4. Create an http server for the xqDoc xml database ('xqDoc') that was created in step (3). In the http server configuration screen, specify the following:

    - 'xqDoc' for the root
    - Specify an available port (i.e. '8000')
    - 'Modules' for the modules
    - 'xqDoc' for the database
    - 'application-level' for authentication
    - a user with admin privileges as the default user

Using the xqDoc Web Service

If you are already storing your XQuery main modules and library modules into a *modules* database (instead of the *filesystem*) the remaining setup to use the xqDoc web service is trivial. Simply go to the configuration page for the *modules* database associated with your XQuery modules and specify 'Triggers' for the triggers database. Then as you change the code contained in your *modules* database, the xqDoc xml will be automatically generated.

If you are currently storing your XQuery main modules and library modules in a *fileystem*, all is not lost. You can still generate the xqDoc xml by performing the following step.

> Create a webdav server for the XQuery module database ('Modules'). In the webdav server configuration screen specify the following:
>
> - '/' for the root
> - 'Modules' for the database
> - 'application-level' for authentication
> - a user with admin privileges as the default user

After configuring the webdav server in Windows (see pg. 37 of the MarkLogic Administrator's Guide), you can then copy your XQuery main modules and library modules to this webdav server and the xqDoc xml will be automatically generated. If you change one of the XQuery module(s) contained in the *filesystem*, you will then need to copy the XQuery module(s) to the webdav server to generate the updated xqDoc xml.

Regardless of the approach taken, the xqDoc information for your XQuery modules can then be viewed from the following url. If you did not use port '8000' or localhost, then use the port you specified and the appropriate machine.

> http://localhost:8000/default.xqy

Experimenting

You might notice that I didn't bother to include the xqDoc documentation for the library and main modules that were developed for this project. The xqDoc documentation can be quickly (and easily) generated by copying the .xqy files contained in the xqDoc-ws.zip to the webdav server created in the previous section, 'Using the xqDoc web service'. Replace any existing files.

Uninstall

To uninstall the supporting main modules, library modules, and triggers that were installed as part of the installation process for xqDoc, simply execute the uninstall script accessible from the following url. This assumes that CQ is available on localhost and port 8007. If this is not the case, use the appropriate machine and port.

> http://localhost:8007/xqDoc-ws/uninstall.xqy

You should receive the following message:

> "The xqDoc web service modules and triggers have been successfully uninstalled."

To learn more about the uninstall script, review the documentation contained in the uninstall.xqy main module. It should be noted that the xqDoc database and forest (created in step 3 of the installation) and the xqDoc http server (created in step 4 of the installation) will not be removed.

Futures

In the near future, we will distribute the xqDoc web service as a .war file to permit a local installation of the xqDoc web service. This would then eliminate the need for being connected to the Internet to access the xqDoc web service available from xqdoc.org. Plus, this would alleviate any potential security concerns around XQuery code being sent to a 3rd party web service for the generation of the xqDoc xml.

Have Fun.

Notes

xqDoc xml will only be generated for a 'valid' XQuery library or main module. A 'valid' module implies the module can be successfully 'parsed' by the xqDoc grammar. As a general rule of thumb, if the module can be processed by the MarkLogic grammar, then it is almost always a 'valid' xqDoc module. If you notice inconsistencies (other than those listed below) between the MarkLogic grammar and the xqDoc grammar, please drop us a note.

The following short list documents the known differences between the MarkLogic grammar and the xqDoc grammar. When appropriate, a workaround is offered to enable a XQuery module to be successfully processed by both the MarkLogic and xqDoc grammars. In all of the following instances, the xqDoc grammar (we believe) more accurately reflects the XQuery language specification.

1. The xqDoc grammar does not handle the MarkLogic proprietary extension of a semi-colon to designate multiple XQuery transactions within a single main module. There is no workaround for this scenario.

2. The xqDoc grammar uses a strict interpretation of the pre-defined entities that are permitted (per the XQuery specification) for a string-literal. Hence, if a string-literal contains " " it will pass the MarkLogic grammar but will fail the xqDoc grammar. The workaround in this scenario is to follow the XQuery specification and utilize the appropriate character reference for the illegal entity. For the case of ' ' one would use " ".