soapgate Q! 4.5

FIXES

Multi-threading issue related to NotesViewEntries.ColumnValues()

A multi-threading bug was reported to us, causing a Domino Server 8.5.3 (64bit Windows) to crash reproducibly when load testing (multiple concurrent calls of) the dbLookupX() and dbColumnX() web service operations. This is apparently an issue of the NotesViewEntry.ColumnValues() method that of course is used in both of the above web service operation. The issue is under investigation by IBM.

For the time being we have made a minor code adjustment that will allow the save use of the two web service operations in a Domino 8.5.3 environment. NotesViewEntry.ColumnValues() will only be accessed if the web service call has parameters referring to columns. If only fields/items (of documents shown in the view) are requested, NotesViewEntry.ColumnValues() will no longer be accessed at all. Of course this work around comes at a performance loss.

We are too still investigating on our end if this is a generic issue with NotesViewEntry.ColumnValues() or is somehow related to the view design or content shown. And whether this issue has been fixed in LND 9.0.

Subscript out of range in dbFTSearch()

Another issue reported was a subscript out of range error when calling dbFTSearch() where the search criteria would result in no documents. This issue has been fixed.

Changes & New Features

We have been working on performance on the new version and as part of this we restructured the web services. Instead of a single web service containing all web service operations, we have split and grouped them. We now have following web services:

DominoUtilityWebServices	This is the same as before containing all web service operations, except for the Admin Process related ones added in version 4.
DominoAdminPWebServices	All web service operations related to the NotesAdministrationProcess class.
DominoDocumentsWebServices	All Notes document related web service operations.
DominoViewsFoldersWebServices	All Notes view and folder related web service operations, except for dbLookupNames() and dbLookupAllNames().

	The latter are now located in DominoAccessNABWebServices.
DominoAccessNABWebServices	All user access, ACL and NAB (names and address book) related web service operations.
DominoWinCAPIWebServices	All web service operations that require Win C-API calls and are therefore only usable in Domino for Windows environments.

Please note: some of the operations in the new web services have additional parameters. However, the operations in the old DominoUtilitiesWebservices are not affected. We kept it backwards compatible.

Following operations in DominoViewsAndFoldersWebServices have additional and or removed parameters:

dbColumnX()

dbLookupX()

dbFTSearch()

dbViewFTSearch()

dbSearch()

The Cache parameter of all the above operations addresses now a new feature in Release 4.5, which is data caching. If the parameter is set to true, the computed return value (array) is cached in a Notes document. The caching follows following logik:

1. If a previous query with the same parameters has been found in the cache, the data stored in the cache document is returned if it has not been expired yet (5 minutes; until release it is a fixed value).
2. If no cache document for the same query has been found or it has been found, but was expired, a new cache document is created or the found one updated with the data of the current query.

Note: the cache document is only accessable for the authenticated user, who created it. For multi-user scenarios the cache will not work or will have very little impact respectively. The cache feature is useful in a very limited set of scenarios only. A real world scenario where this feature has already been implemented successfully (though still in its Beta phase) is that of barcode scanning devices (all) synching the same Notes data at periodic schedules. The first synching device causes the data being read from the Notes databases and creates the cache document. All other devices synch (update) from the cache. The devices are using a technical user for authentication and hence have all access to the cache document. Whilst the data sync process for the first device takes about 60 seconds (for about 10,000 Notes documents with an average of 15 fields, totalling to about 1MB of data), all subsequent

sync processes of the other devices are shortened to 30-40 seconds. These timings include storing the data in local SQLight tables on the device.

Following operations in DominoDocumentsWebServices have additional parameters:

dbSaveDocFields()

dbSaveProfileFields()

Both operations support now a two step operation: saving and reading. After completing the save operation, the dbReadDocFields() or dbReadProfileFields() operations are called respectively. To allow control on the list of returned fields, both operations have an additional parameters readFlds (Array). If the array is empty, dbSaveDocfields() and dbSaveProfileFields() return the UNID or NoteID of the document / profile created or amended (this is the previous behavior). If readFlds is an array, only fields listed will be returned. **Note**: NoteID and or UNID will not be automatically added to the list of returned fields. To receive either one of them or both, submit $$NOTEID or $$UNID as one of the entries in readFlds.

Paging for dbLookupX() and dbViewFTSearch()

Whilst dbColumnX() does support paging for some time now, both dbLookup() and dbViewFTSearch() did not. Both these operations finally support paging.

Improved Notes Richtext support

In previous release we used a Java URL request to get the Domino Server to render a Notes Richtext field to HTML. This required some Domino (security) settings in the server document to work, despite not being the best performing method. However, it delivered the so far best results from a fidelity point of view.

In release 4.5 we managed to use the NotesDocument.ConvertToMime method. This is much faster and touch less from a configuration point of view. We had of course to add some tweaks to support inline images and attachments.

Changes from a rendering point of view:

- added support for attachments
- improved support for inline images
- tabbed tables are serialized (all tabs showing below each other)
- sections are serialized

We hopefully will manage to improve on the loss in fidelity related to tabbed tables and sections in future releases.

soapgateQ! database info

We added a RESTFul service to the soapgateQ! database to provide some XML formatted information about the database itself and the environment it runs in. By default the service is available to public users. So no authentication is required.

http://domino.flexdomino.net/soapgateq_4.nsf/database%20info

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
	<soapgateq>
		<product>soapgate Q!</product>
		<version>4.5</version>
		<published format="yyyymmdd">20130902</published>
		<copyright>flexdomino.net</copyright>
		<license>GNU Affero General Public License Version3</license>
		<licenseurl>http://www.gnu.org/licenses/</licenseurl>
		<sponsor>Qkom GmbH</sponsor>
		<sponsorurl>http://www.qkom.de</sponsorurl>
	</soapgateq>
	<environment>
		<server>CN=flexdomino/O=Flex2Domino</server>
		<dominoversion>392</dominoversion>
		<database>soapgateq_4.nsf</database>
		<sessionuser>Anonymous</sessionuser>
	</environment>
	<configuration>
		<debuglevel>0</debuglevel>
		<defaultaccess>1</defaultaccess>
		<serversidecode>1</serversidecode>
		<base64encoding>2</base64encoding>
		<htmloptions></htmloptions>
		<htmluser></htmluser>
		<forcehttps></forcehttps>
		<datetimeformat>ISO8601</datetimeformat>
	</configuration>
</root>
```

soapgateQ! database info

We added a RESTFul service to the soapgateQ! database to provide some XML formatted information about the database itself and the environment it runs in. By default the service is available to public users. So no authentication is required.