

DOCUMENTATION

Version 1.0 BETA, February 17, 2010

soapgate 
domino exposed
...the missing link



© 2009 FlexDomino.net

PREFACE

We thank you for being a user of SoapGateQ! the universal web service gateway for Lotus Notes Domino™.

This manual is designed to assist Domino administrators in the installation of SoapGateQ! as well as developers in the use of the web service API provided by SoapGateQ!

These materials are copyrighted and the intellectual property rights are vested in FlexDomino.net.

Copying through any means is unauthorized without the express written permission of an officer of FlexDomino.net.

Every reasonable attempt has been made to ensure the accuracy of this user manual and that it reflects the operations of the product, however users of SoapGateQ! are responsible for ensuring that the product and its documentation are suitable for the needs of that organization. No warranties in respect of this user manual are being made or can be assumed.

© *FlexDomino.net*, 2010

Lotus, Lotus Notes, Domino and their respective logos as well as the IBM Business Partner logo are all registered trademarks of IBM Corporation.

Flex, Flex Builder, Flash and Flash Player as well as their respective logos are all trademarks of Adobe Corp. All other trademarks are *hereby duly acknowledged as the property of their respective owners.*

INDEX

Table of Contents

Preface.....	2
Index.....	3
Introduction.....	5
Installation and system requirements.....	6
Installation.....	6
API Reference.....	7
Overview.....	7
dbColumn.....	8
dbColumnX.....	9
dbRowX.....	10
dbLookup.....	11
dbLookupX.....	13
dbGetFormFields.....	15
dbGetFieldTypes.....	16
dbReadDocFields.....	17
dbSaveDocFields.....	19
dbRenderDoc.....	21
dbDeleteDoc.....	22
dbGetUserInfo.....	23
dbSendDocument.....	24
dbViews.....	25
dbSearch.....	26
dbViewFTSearch.....	28

Documentation for SoapGateQ!

dbFTSearch.....	30
dbCallAgent.....	32
Appendix.....	33
Notes field and data types:.....	33
Data Serialization.....	34

INTRODUCTION

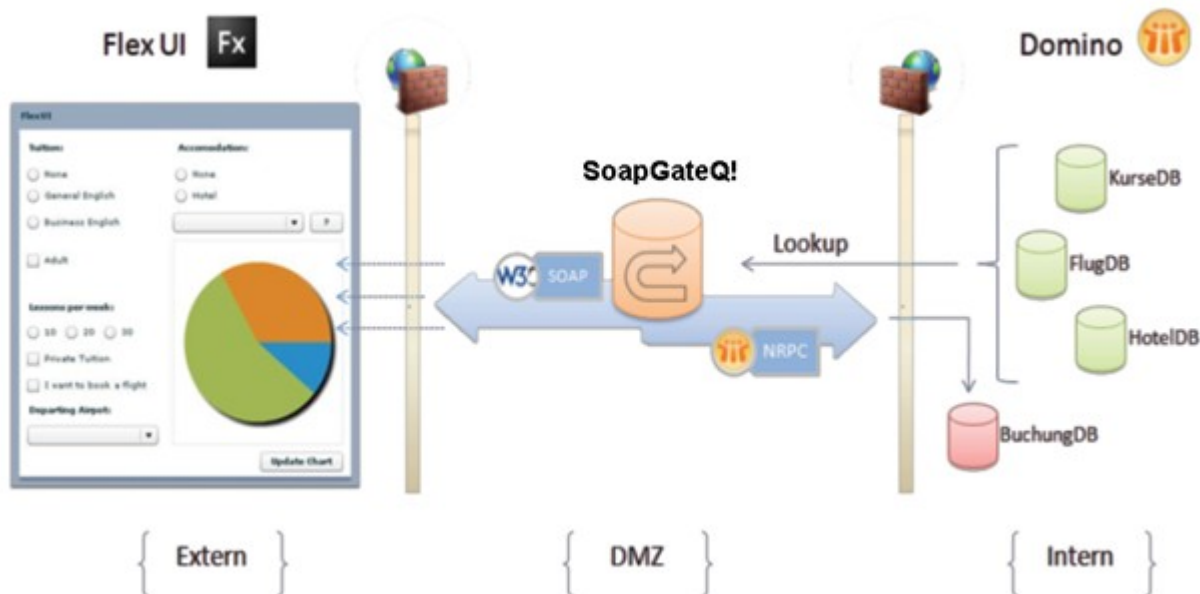
SoapGateQ! is a universal gateway to Domino based on a web service API. The web service API is contained in a single web service stored in the SoapGateQ! (Notes) database.

The web service API is a collection of native Notes Formula and Lotus Script classes (or methods respectively), such as @dbccolumn(), @dblookup(), NotesDatabase.DBSearch(), NotesView.FTSearch etc. In addition to the native Notes features the API also contains a few web service operations that provide extended functionality, such as dbcolumX() and dblookupX(), which unlike their native counterparts return more than one column or field in a single call.

The concept of SoapGateQ! is to provide an easy way to expose your Domino data to none Notes systems or none Notes clients/applications, such as Oracle, .Net, Flex/Flash or mobile devices (e.g. Blackberry, iPhone). Basically to any framework that can consume web services.

This without requiring changes to any of your existing Domino applications (databases), and equally important without exposing your data directly to the world. The SoapGateQ! database would typically reside on the public Domino server (located in the DMZ), accessing databases on the Domino server(s) in the private (fire-walled) network.

The following chart illustrates the above:



The SoapGateQ! web service runs as a Domino web user and hence supports all Domino authentication requirements to the respective databases that are accessed via the web service.

INSTALLATION AND SYSTEM REQUIREMENTS

SoapGateQ! is developed for Lotus Domino 8.5.1. The web service is configured to support the RPC doc/literal SOAP message format.

We will outline further down how to create the web service for a Lotus Domino 8 environment and how to provide a web service that uses a different SOAP message format.

On request only we can provide SoapGateQ! for Lotus Domino 7.

Installation

1. SoapGateQ! does not come with an installer as there are only a few simple steps to “plug and play”
2. Copy the SoapGateQ! database into the data directory of your public Domino server.
3. Open the Notes Administrator client, connect to the respective Domino server and sign the SoapGateQ! database with the server's ID file.
4. Configure the SoapGateQ! ACL accordingly. Recommended are following ACL entries:
 - -Default- (Unspecified) = No Access
 - Anonymous (Unspecified) = Editor (for public sites)
 - Anonymous (Unspecified) = No Access (for extranet scenarios)
 - <Your Server> (Server) = Manager
 - Username(s) (Person) = Editor (for extranet scenarios)
 - Groupname(s) (Person group) = Editor (for extranet scenarios)
5. The server that is to host SoapGateQ! must run the Domino HTTP task. So if Domino is not the public HTTP server, but for instance IIS or an Apache etc., the Domino server has to be configured to run its HTTP task on a port other than port 80. This type of configuration of course effects firewall configuration requirements as well as the external systems accessing the SoapGateQ! web service operations. For the latter the port has to be part of the calling URL.

API REFERENCE

Overview

dbColumn

Web service implementation of the Notes Formula function @dbcolum().

Parameters:

cache	Boolean	if false Domino will update its cache from the data on file
srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
viewname	String	Name or alias of the view, which data is to be returned
colnumber	Integer	Column number to be returned starting from 1 for the first column in the view
mvalsep	String	Multi-value separator

Return value:

Array:Any

Serialization:

Because of the required data serialization for multi-value columns, such values will be converted to type string, each value separated by the character passed on in *mvalsep*. Otherwise data types remain intact.

See *Appendix* for more information.

dbColumnX

Web service implementation of the Notes Formula function @dbcolum() with two extended feature:

- returns one or more columns
- returns columns and or fields

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
viewname	String	Name or alias of the view, which data is to be returned
colfields	String	List of column numbers and or fieldnames in a single string separated with the character passed on in <i>mvalsep</i>
mvalsep	String	Multi-value separator
addUNID	Boolean	If True an additional column is returned containing the document universal Ids
addNoteID	Boolean	If True (and addUNID = False) an additional column is returned containing the document notes Ids
rowformat	Boolean	If true data is returned by row rather than by column (see <i>Return value</i>)

Return value:

Array: Any

Serialization:

Due to data serialization the returned value is a single dimensioned array. The *rowformat* parameter determines the result of the serialization.

See *Appendix* for more information.

dbRowX

Returns column and or field values for a single view row.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
viewname	String	Name or alias of the view, which data is to be returned
colfields	String	List of column numbers and or fieldnames in a single string separated with the character passed on in <i>mvalsep</i>
mvalsep	String	Multi-value separator
addUNID	Boolean	If True an additional column is returned containing the document universal Ids
addNoteID	Boolean	If True (and addUNID = False) an additional column is returned containing the document notes Ids
docUNID	String	Universal ID of the document representing the single view row to be returned

Return value:

Array: Any

Serialization:

Because of the required data serialization for multi-value column entries, such values will be converted to type string, each value separated by the character passed on in *mvalsep*. Otherwise data types remain intact.

See *Appendix* for more information.

dbLookup

Web service implementation of the Notes Formula function @dblookup().

Parameters:

cache	Boolean	if false Domino will update its cache from the data on file
srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
key	String	A value to be matched against the first sorted column in the view
viewname	String	Name or alias of the view, which data is to be returned
colnumber	Integer	Number of a column in the view starting at 1. If a column number is used, the return values are taken from this column in the view
fieldname	String	Name of an item in the documents in the view. If an item name is used, the return values are taken from this item in the documents from which the view data is drawn
mvaluep	String	Multi-value separator
failsilent	Boolean	returns "" (null string) instead of an error if the key cannot be found
partialmatch	Boolean	returns a match if the key matches the beginning characters of the column value
returnUNID	Boolean	returns the UNID of the document instead of a field or column value

Return value:

Array:Any

Serialization:

Because of the required data serialization for multi-value column or fields, such values will be converted to type string, each value separated by the character passed on in *mvaluep*. Otherwise data types remain intact.

See *Appendix* for more information.

dbLookupX

Web service implementation of the Notes Formula function @dblookup() with two extended features:

- returns one or more columns
- returns columns and or fields

Parameters:

cache	Boolean	if false Domino will update its cache from the data on file
srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
key	String	A value to be matched against the first sorted column in the view
viewname	String	Name or alias of the view, which data is to be returned
colfields	String	List of column numbers and or fieldnames in a single string separated with the character passed on in <i>mvalsep</i>
mvalsep	String	Multi-value separator
failsilent	Boolean	returns "" (null string) instead of an error if the key cannot be found
partialmatch	Boolean	returns a match if the key matches the beginning characters of the column value
returnUNID	Boolean	returns the UNID of the document in addition to the requested fields and or columns
returnNoteID	Boolean	returns the UNID of the document in addition to the requested fields and or columns (works only if <i>returnUNID</i> = False)
rowformat	Boolean	If true data is returned by row rather than by column (see <i>Return value</i>)

Return value:

Array:Any

Serialization:

Due to data serialization the returned value is a single dimensioned array. The *rowformat* parameter determines the result of the serialization.

See *Appendix* for more information.

dbGetFormFields

Returns the list of fieldnames and -types contained in a specified form.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
form	String	Name or alias of the form, which field list is to be returned
includeLSF	String	If True includes Lotus \$ (system) fields

Return value:

Array:String and Integer

fieldname1
fieldname2
fieldname3
...
datatype1
datatype2
datatype3
...

Serialization:

Due to data serialization the returned value is a single dimensioned array. The array starts with the list of field names and continues with the list of data types.

See *Appendix* for more information.

Data types:

See *Appendix*

dbGetFieldTypes

Returns the data types for a given list of fields contained in a given form.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
form	String	Name or alias of the form, which field list is to be returned
fields	String	List of fieldnames in a single string separated with the character passed on in <i>mvalsep</i>
mvalsep	String	Multi-value separator

Return value:

Array:Integer

Data types:

See *Appendix*

dbReadDocFields

Returns the values and data types for the list of given fields of a given document.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
docUNID	String	Universal ID of the document which field values are to be returned
fields	String	List of fields in a single string separated by the character passed on in <i>mvaluep</i> , which values are to be returned
mvaluep	String	Multi-value separator

Return value:

Array:String and Integer

```
fieldvalue1  
fieldvalue2  
fieldvalue3  
...  
datatype1  
datatype2  
datatype3  
...
```

Serialization:

Due to data serialization the returned value is a single dimensioned array. The array starts with the list of field values and continues with the list of data types. For those field values that represent a Notes multi-value field, the multi-value content is serialized using the *mvaluep* character. The web service consumer must convert these back to arrays.

The recommended (default) multi-value separator is “~” (tilde). Assuming the returned second field value is of type multi-value, the returned array content looks like this...

Documentation for SoapGateQ!

fieldvalue1 fieldvalue2 fieldvalue3 ...	fieldvalue1 fieldvalue2_a- fieldvalue2_b-... fieldvalue3 ...
--------------------------------------------------	-----------------------------------------------------------------------

All field values are converted to string. This complies with the way the Notes Client UI document work. Whilst edited all fields in a Notes document are edited as text. Default and custom validations control the users input.

Similar to this concept the web service consumer, for instance a Flex form must have input formatters and validations implemented.

Alternatively the consumer must convert the returned string data in accordance with the also provided data type information into whatever is required.

The correlating web service operation *dbSaveDocFields* converts the string data back into the respective data types required in the used Notes form for the document being modified.

See *Appendix* for more information.

Data types:

See *Appendix*

dbSaveDocFields

Updates field values in a given document or saves values in a new document.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
docUNID	String	Universal ID of the document which field values are to be returned
fields	String	List of fields in a single string separated by the character passed on in <i>mvalsep</i> , which values are to be returned
types	String	List of corresponding data types converted to a single string separated by the character passed on in <i>mvalsep</i> . Please refer to the <i>Appendix</i> for the Notes field data types
values	String	List of field values in a single string separated by the fixed character "-". This is due to the fact that individual values might be themselves a serialized multi-value string using the <i>mvalsep</i> character for separation. This also means that "-" cannot be used for <i>mvalsep</i> . FlexGateQ! the wrapper classes for Flex will do this conversion automatically. For which reason in Flex the developer deals with arrays only, data serialization is transparent
mvalsep	String	Multi-value separator
form	String	Name of form used for new documents
compWF	Boolean	If True runs the NotesDocument.ComputeWithForm() method on the server before the document is saved to file
saveRPC	String	Runs a Notes agent on the server before the document is saved file. This is the equivalent to the WebQuerySave event when submitting Notes

Documentation for SoapGateQ!

	form
--	------

Return value:

String:UNID

Universal ID (UNID) of the document updated or created.

dbRenderDoc

This feature is not yet implemented.

Renders a Notes document into HTML or MIME

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
docUNID	String	Universal ID of the document which is to be rendered
noteID	String	Alternatively the NoteID
form	String	Name of form used for rendering the document
noCache	Boolean	True or false. Not yet supported parameter

Return value:

String:HTML/MIME

dbDeleteDoc

Deletes a document from file.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
docUNID	String	Universal ID of the document which field values are to be returned
force	Boolean	IF True the document will be deleted even if it is currently edited by another user

Return value:

String: "true", "false"

dbGetUserInfo

Web service implementation of the Domino class method `Notesdirectory.Getmailinfo()`.

Parameters:

username	String	Name of the registered user
----------	--------	-----------------------------

Return value:

Array:String

Elements returned are as follows:

Mail Server	Home mail server for the specified person
BuildNumber	If <code>getver</code> is true, a string representation of the build number of the specified person's mail server, for example, "303". If <code>getver</code> is false, ""
DominoVersion	If <code>getver</code> is true, a string representation of the Domino version of the specified person's mail server, for example, "Build V80_07042006NP". If <code>getver</code> is false, ""
MailFile	Mail file for the specified person
ShortName	Short form of the specified person's name
MailDomain	Notes Domain of the specified person's mail address
User Name	First entry in the list of user names honored for the specified person
InternetMailAddress	Internet mail address for the specified person
OutOfOffice	Out of Office service type. "1" indicates Agent, "2" indicates Service

dbSendDocument

Web service implementation of the Domino class method `NotesDocument.Send()`.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
docUNID	String	Universal ID of the document which field values are to be returned
mvalsep	String	Multi-value separator
attachForm	Boolean	Attach form used for the send document (as defined in the <i>Form</i> field)
recipients	String	List of recipients in a single string separated by the character passed on in <i>mvalsep</i>

Return value:

String: "true", "false"

dbViews

Web service implementation of the Domino class property `NotesDatabase.Views()`.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
viewonly	Boolean	If true returns views only
folderonly	Boolean	If true returns folders only
includehidden	Boolean	If true returns also hidden views and folders

Return value:

String:XML

<views>

<node label="By Severity" data="vwEventsBySeverity1"/>

<node label="By Date" data="vwEventsByDate1"/>

<node label="By Type" data="vwEventsByType1"/>

<node label="By Database" data="vwEventsByDatabase1"/>

<node label="By Server" data="vwEventsByServer1"/>

<node label="By Assignment" data="vwEventsByAssignment1"/>

</views>

The *label* attribut contains the view name, the *data* attribut contains the alias name of the view or if not alias is available the view name.

dbSearch

Web service implementation of the Domino class method `NotesDatabase.Search()` with extended features:

- returns one or more columns
- returns columns and or fields
- returns document collection in form of list of UNIDs/NotelDs

Parameters:

<code>srvname</code>	String	Name of the Domino server on which the source database resides
<code>dbname</code>	String	Pathname of the database (relative to the root data directory of the server)
<code>formula</code>	String	A Notes @function formula that defines the selection criteria
<code>datetime</code>	String	A cutoff date. The method searches only documents created or modified since the cutoff date. Specify "" (null string) to indicate no cutoff date.
<code>maxdocs</code>	Integer	Maximum number of documents returned
<code>colfields</code>	String	List of column numbers and or fieldnames in a single string separated with the character passed on in <i>mvalsep</i>
<code>mvalsep</code>	String	Multi-value separator
<code>returnUNID</code>	Boolean	returns the UNID of the document in addition to the requested fields and or columns
<code>returnNotelD</code>	Boolean	returns the UNID of the document in addition to the requested fields and or columns (works only if <i>returnUNID</i> = False)
<code>rowformat</code>	Boolean	If true data is returned by row rather than by column (see <i>Return value</i>)

Return value:

Array:Any

Serialization:

Due to data serialization the returned value is a single dimensioned array. The *rowformat* parameter determines the result of the serialization.

See *Appendix* for more information.

dbViewFTSearch

Web service implementation of the Domino class method `NotesView.FTSearch()` with extended features:

- returns one or more columns
- returns columns and or fields
- returns document collection in form of list of UNIDs/NoteIDs

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
viewname	String	Name or alias of the view, which data is to be returned
query	String	The full-text query. Please refer to the Lotus Notes Help for syntax information
maxdocs	Integer	The maximum number of documents you want returned from the query. Set this parameter to 0 to retrieve all documents (up to 5,000 - see the Usage section for more details) that meet the search criteria
colfields	String	List of column numbers and or fieldnames in a single string separated with the character passed on in <i>mvalsep</i>
mvalsep	String	Multi-value separator
returnUNID	Boolean	returns the UNID of the document in addition to the requested fields and or columns
returnNoteID	Boolean	returns the UNID of the document in addition to the requested fields and or columns (works only if <i>returnUNID</i> = False)
rowformat	Boolean	If true data is returned by row rather than by column (see <i>Return value</i>)

Return value:

Array:Any

Serialization:

Due to data serialization the returned value is a single dimensioned array. The *rowformat* parameter determines the result of the serialization.

See *Appendix* for more information.

dbFTSearch

Web service implementation of the Domino class method NotesDatabase.FTSearch() with extended features:

- returns one or more columns
- returns columns and or fields
- returns document collection in form of list of UNIDs/NoteIDs

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
query	String	The full-text query. Please refer to the Lotus Notes Help for syntax information
maxdocs	Integer	The maximum number of documents you want returned from the query. Set this parameter to 0 to retrieve all documents (up to 5,000 - see the Usage section for more details) that meet the search criteria
sortoption	Integer	Optional. Sorting options. Choose one (64) sorts by document date in ascending order. (32) sorts by document date in descending order. (1543) sorts by document creation date in ascending order. (1542) sorts by document creation date in descending order. (8) sorts by relevance score (default)
otheroptions	Integer	Optional. Search options. Combine options by adding (8192) includes Domino databases. (16384) searches for related words. Need not be an exact match. (4096) includes files that are not Domino

Documentation for SoapGateQ!

		databases. (512) uses stem words as the basis of the search. (1024) uses thesaurus synonyms
colfields	String	List of column numbers and or fieldnames in a single string separated with the character passed on in <i>mvalsep</i>
mvalsep	String	Multi-value separator
returnUNID	Boolean	returns the UNID of the document in addition to the requested fields and or columns
returnNotelD	Boolean	returns the UNID of the document in addition to the requested fields and or columns (works only if <i>returnUNID</i> = False)
rowformat	Boolean	If true data is returned by row rather than by column (see <i>Return value</i>)

Return value:

Array:Any

Serialization:

Due to data serialization the returned value is a single dimensioned array. The *rowformat* parameter determines the result of the serialization.

See *Appendix* for more information.

dbCallAgent

Web service implementation of the Domino class property `NotesAgent.Run()`.

Parameters:

srvname	String	Name of the Domino server on which the source database resides
dbname	String	Pathname of the database (relative to the root data directory of the server)
agent	String	Name of agent to be executed
paramXML	String:XML	Any XML formatted string which will be attached to an agent context document in a pre-defined field called <i>agentcontextXML</i>
noteID	String	Note ID to an agent context document

Return value:

String:XML

The return value is either "" (null string) or taken from the agent context document from the pre-defined field *agentcontextXML*.

APPENDIX

Notes field and data types:

ATTACHMENT (1084)	means file attachment
AUTHORS (1076)	means authors
DATETIMES (1024)	means date-time value or range of date-time values
EMBEDDEDOBJECT (1090)	means embedded object
HTML (21)	means HTML source text
MIME_PART (25)	means MIME support
NAMES (1074)	means names
NOTELINKS (7)	means link to a database, view, or document
NOTEREFS (4)	means reference to the parent document
NUMBERS (768)	means number or number list
READERS (1075)	means readers
RFC822Text (1282)	means RFC822 Internet mail text
RICHTEXT (1)	means rich text
SIGNATURE (8)	means signature
TEXT (1280)	means text or text list

There are more data types. Please refer to the Lotus Notes Designer documentation.

Data Serialization

At the current stage of development most SoapGateQ! web service operations return a single dimensional array, regardless of whether the operations does collect the requested data in a 2-dimensional array. The Domino 8+ server serializes a 2-dimensional array to a single dimensional array.

As a result an array of values collected by e.g. `dbcolumnX` for the first 3 columns of a view looking like this...

Column1 Row1	Column2 Row1	Column3 Row1
Column1 Row2	Column2 Row2	Column3 Row2
Column1 Row3	Column2 Row3	Column3 Row3

...is converted to a single dimensional array looking like this...

Column1 Row1
Column1 Row2
Column1 Row3
Column2 Row1
Column2 Row2
Column2 Row3
Column3 Row1
Column3 Row2
Column3 Row3

For some of the offered web service operations the parameter *rowformat* determines the way the data serialization works:

Documentation for SoapGateQ!

rowformat = True	rowformat = False
row1_column1	row1_column1
row1_column2	row2_column1
row1_column3	row3_column1
...	...
row2_column1	row1_column2
row2_column2	row2_column2
...	...

Data serialization also effects multi-value column entries or fields. Multi-value column and field values will be converted to type string and each value separated by the character passed on in the *mvalsep* parameter available to all effected web service operations. The data types for single value columns and fields remain intact.

The consumer of the SoapGateQ! web service operation must reverse the serialization and create a 2-dimensional array or array collection from the returned single dimensional array. It also needs to revert back serialized multi-values.

The wrapper classes of FlexGateQ! (for Flex Builder 3) are doing this for all web service operations of SoapGateQ! where the above applies.

Multi-value separator considerations:

The character used in *mvalsep* (default "~" tilde) cannot be part of any of the data columns and or fields received from or send to the server. The same applies for the second (fixed) serialization character "-" used by the system.