# Soapgate Q! 4 (4.1)

With Soapgate Q! 3.0 we introduced a new security feature which allows the server side custom code execution (Lotus Script, Formula or Agent) through the API to be limited to code that is known to the Soapgate Q! in form of Code Profiles.

Similar to the Database Access Profiles, Code Profiles provide the means to authorize code and make it available to external users using reader names fields.

In addition it is possible to refer to Lotus Script, Notes Formulae or Agents using the NoteID of the Code Profile (rather than submitting the code as a parameter to the respective web service operations of the API.

## New in 4.1

**In Soapgate Q! 4.1 this development has been completed in that ALL web service operations that allow for parameterised (server side) code execution can now simply send Code Profile references.**

As part of this new code security model, the previously in the Database Access Profile located "Server-side code execution" setting has moved into the global API Configuration screen. The options also changed slightly:

- Permit only Code Profile references (default)
- Permit server-side code execution

If set to *Permit server-side code execution* all Notes Formula and or existing Notes Agents will be executed. This setting is not recommended and should only be used in test or development environments.

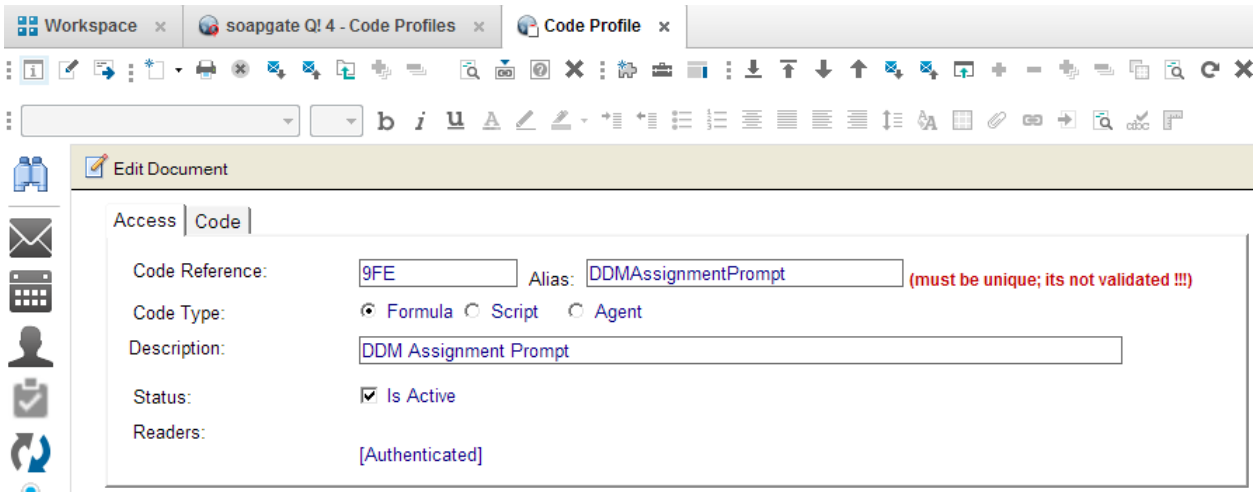
In production environments it is important not to allow the execution of arbitrary code sent to the server. Therefore once parameterised code is working, it should be placed in Code Profiles and the option *Permit only Code Profile references* should be selected. Code Profiles allow the use (execution) of Notes Formula, Lotus Script and any type of Notes Agent that has/have been sanctioned by the developers, administrators and or business owners. Code Profiles provide for Readernames access control.

Code Profiles have also no a Reference Alias. Rather then referring to a Code Profile using its Notes ID, one can now add and use a meaningful text (Alias) as the reference.

**The rest of this document is an in some details corrected version of the documentation found in the Soapgate Q! 3 release notes.**

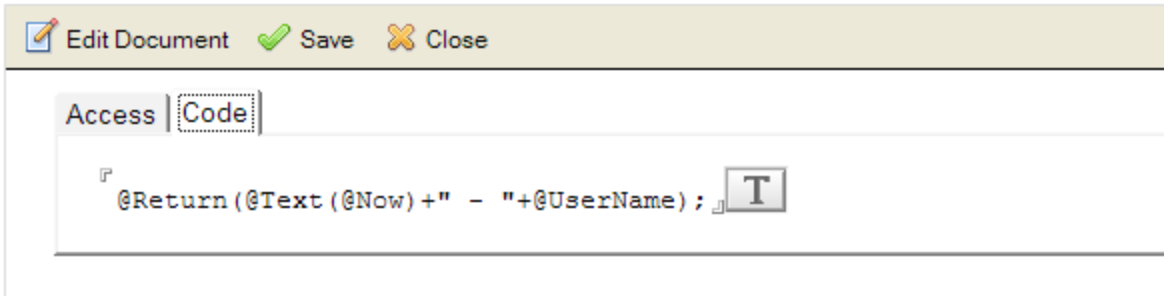
The Code Profile

In the Code Profile Access tab one defines the type of the authorized code (Formula, Script or Agent), and defines who will have access to it through the API. The Code Reference is computed and is basically the NoteID of the document.



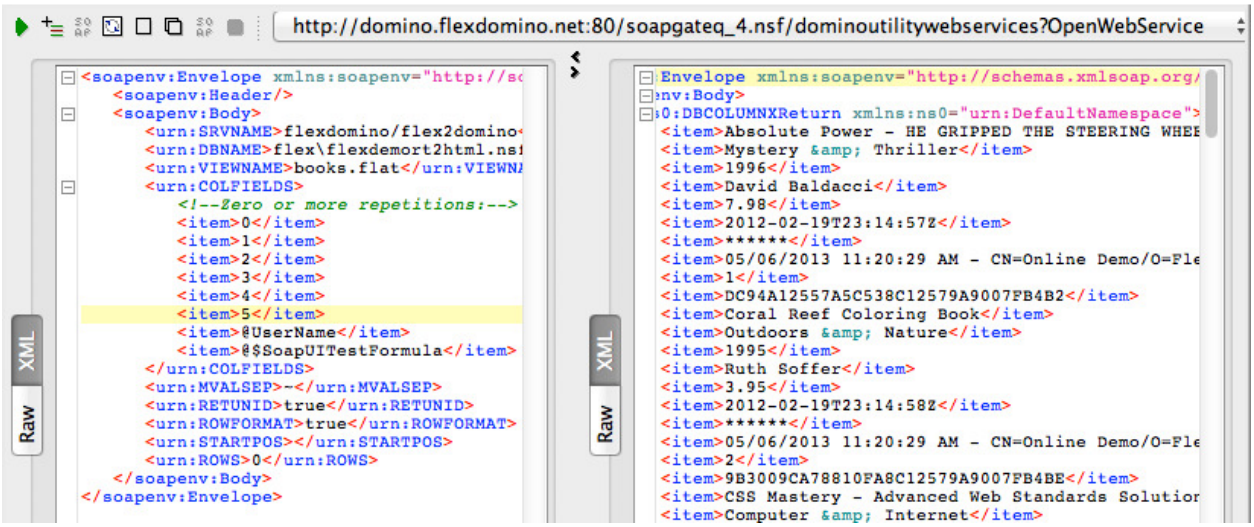

**Note: as indicated in the above screen shot, the new Alias must be unique. However, this is not validated**

The Code tab can be used to store either a Notes Formula, Lotus Script or the name of an Agent (this agent must be located in the source/target database which is accessed through the API, not the Soapgate Q! database.




Referencing a Code Profile

The code or agent name contained in the Code Profile can be referenced using the notation **$**CodeReference**Alias** (as set in the Code Profile). To test this in SOAP-UI this would look like this:



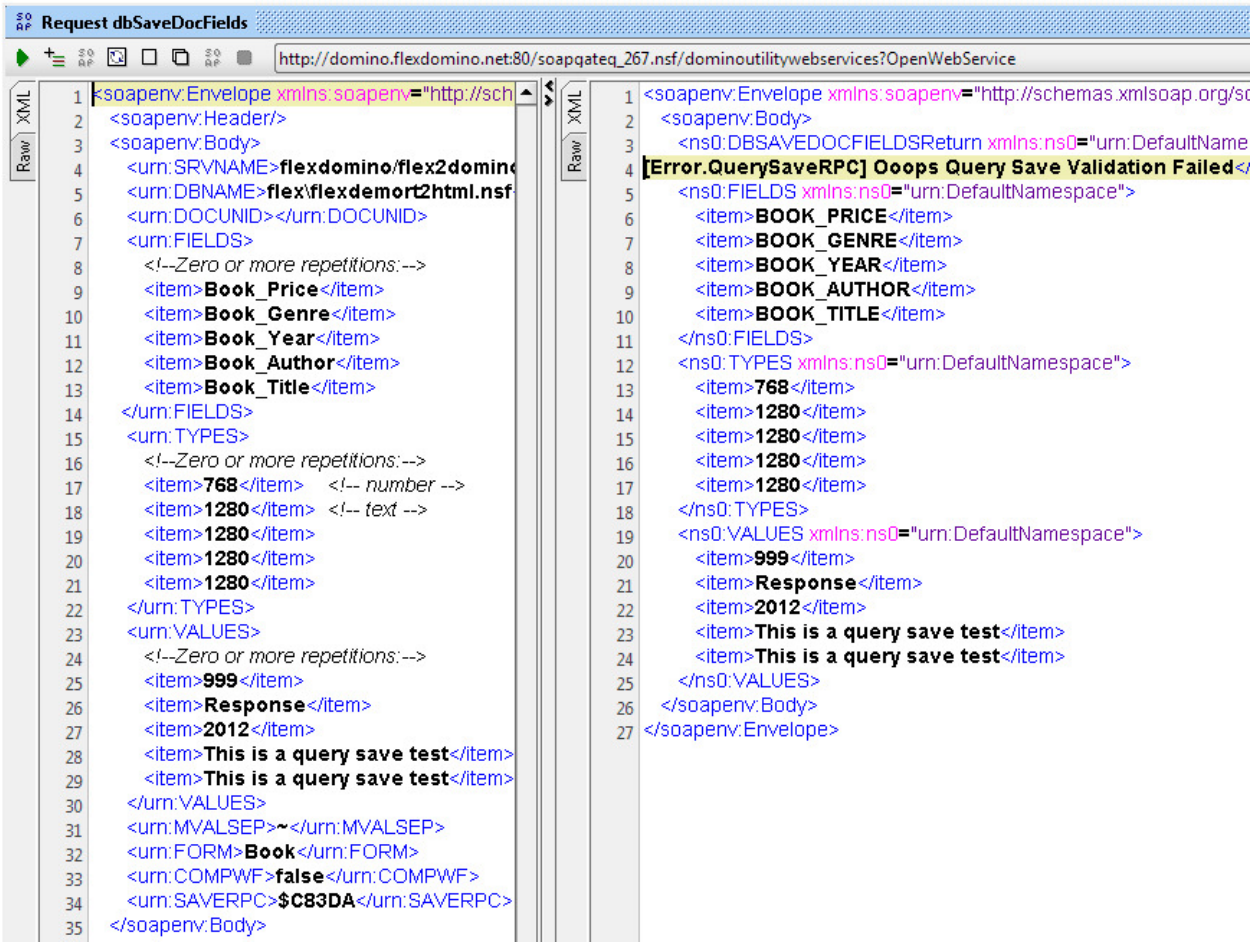

The above screen shot shows a dbColumnX() request. The last two items in the COLFIELDS list are a formula (@UserName) and a Code Profile reference (@$SoapUiTestFormula). As the code security is set to production environment, the formula fails (******), whilst the formula stored in the Code Profile is executed and returns a concatinated DateTime + Username string.

**Note**: the leading @ to indicate a computed field is still required, as $SoapUiTestFormula could be a custom $field on the Notes document. In the context of computed fields only Code Profiles of type Script or Formula are valid; Code Profiles of type Agent are ignored.

Query Save Agent (saveRPC) in dbSaveDocFields()

With Soapgate Q! 3.0 we also finally implemented the equivalent of the QuerySaveAgent for Notes web forms. The dbSaveDocFields() operation always provided the saveRPC parameter, which however was ignored in all previous versions.

It is now possible to either pass on a name of an Agent that must be located in the target database or to reference a Code Profile. In the context of the saveRPC, all Code Profile types are valid. However, whilst Lotus Script or a Script Agent can make modifications on the to be saved document, a Formula cannot; the latter can simply indicate through a return value whether or not the document should be saved or not.



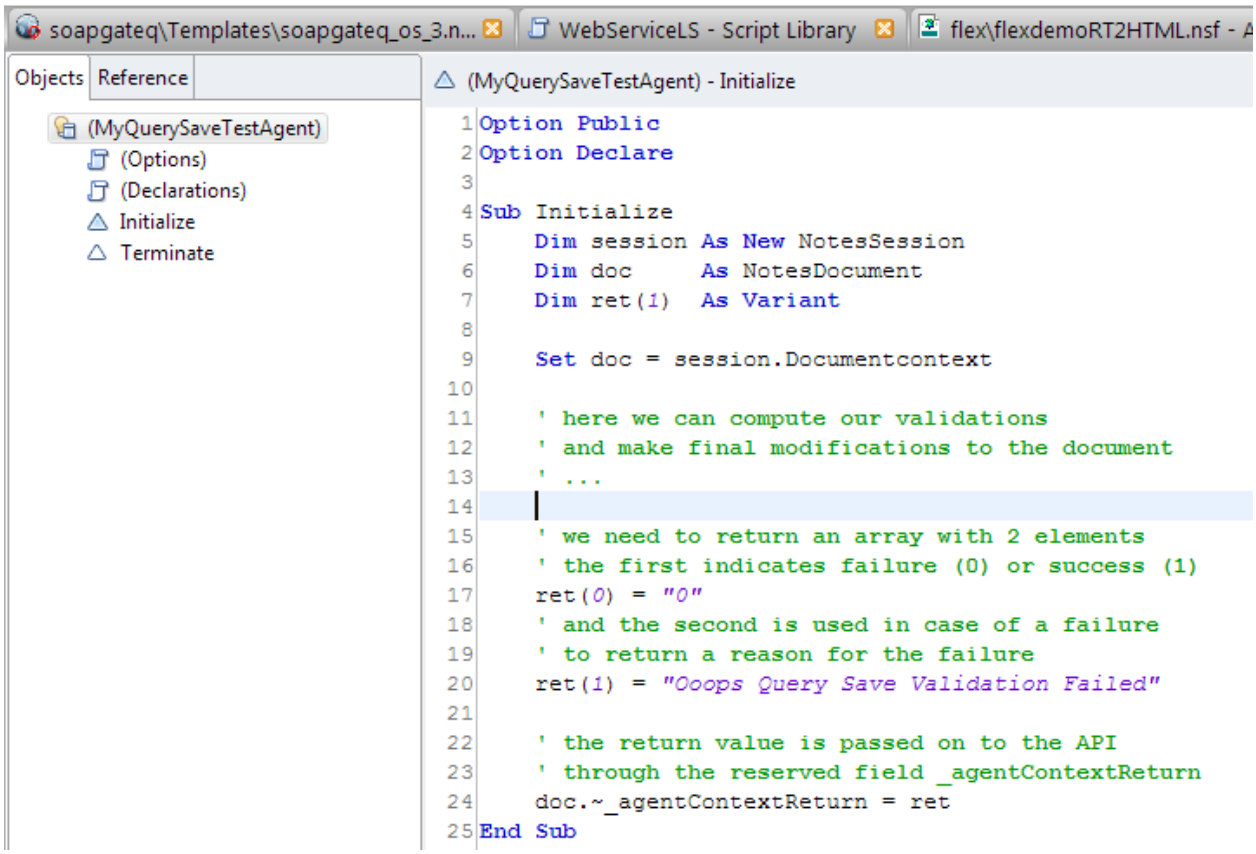

The above screen shot shows a Code Profile reference $C83DA in the SAVERPC parameter. In the example the code returns a failure code causing the document not to be saved.

Code Profile of type Agent

Code Profiles can reference any type of agent, however, Lotus Script or Java agents are recommended for most web service operations provided through this API.

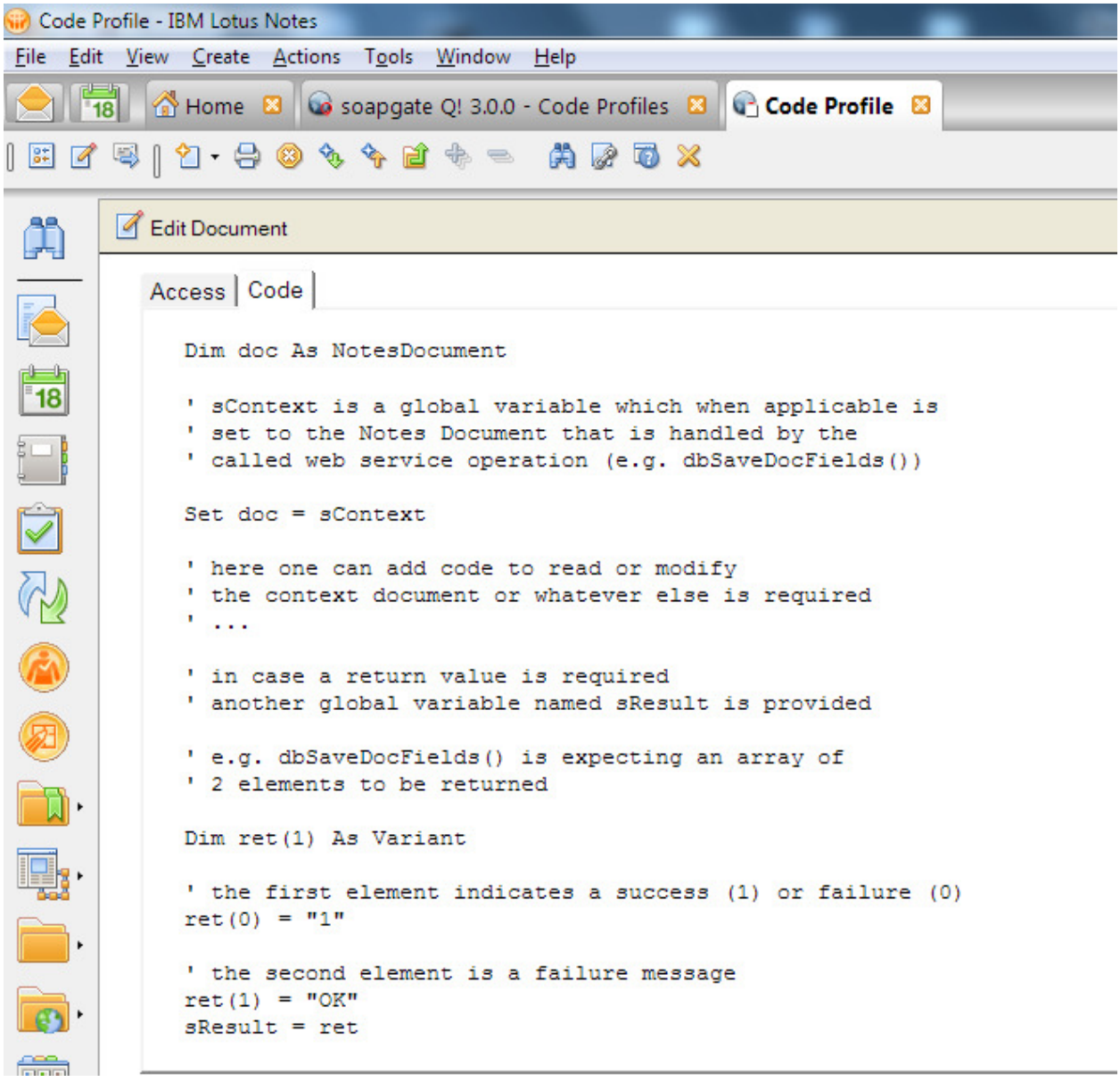
Where applicable agents are executed through Lotus Script using the NotesAgent.Runwithdocumentcontext method. This will allow the agent to reference or modify the in-memory document.

The latter is required for agents called through the saveRPC parameter using dbSaveDocFields(). This as this web service operation is expecting a return parameter in a reserved document field named '_agentContextReturn'. The field will be deleted from the document before being saved.

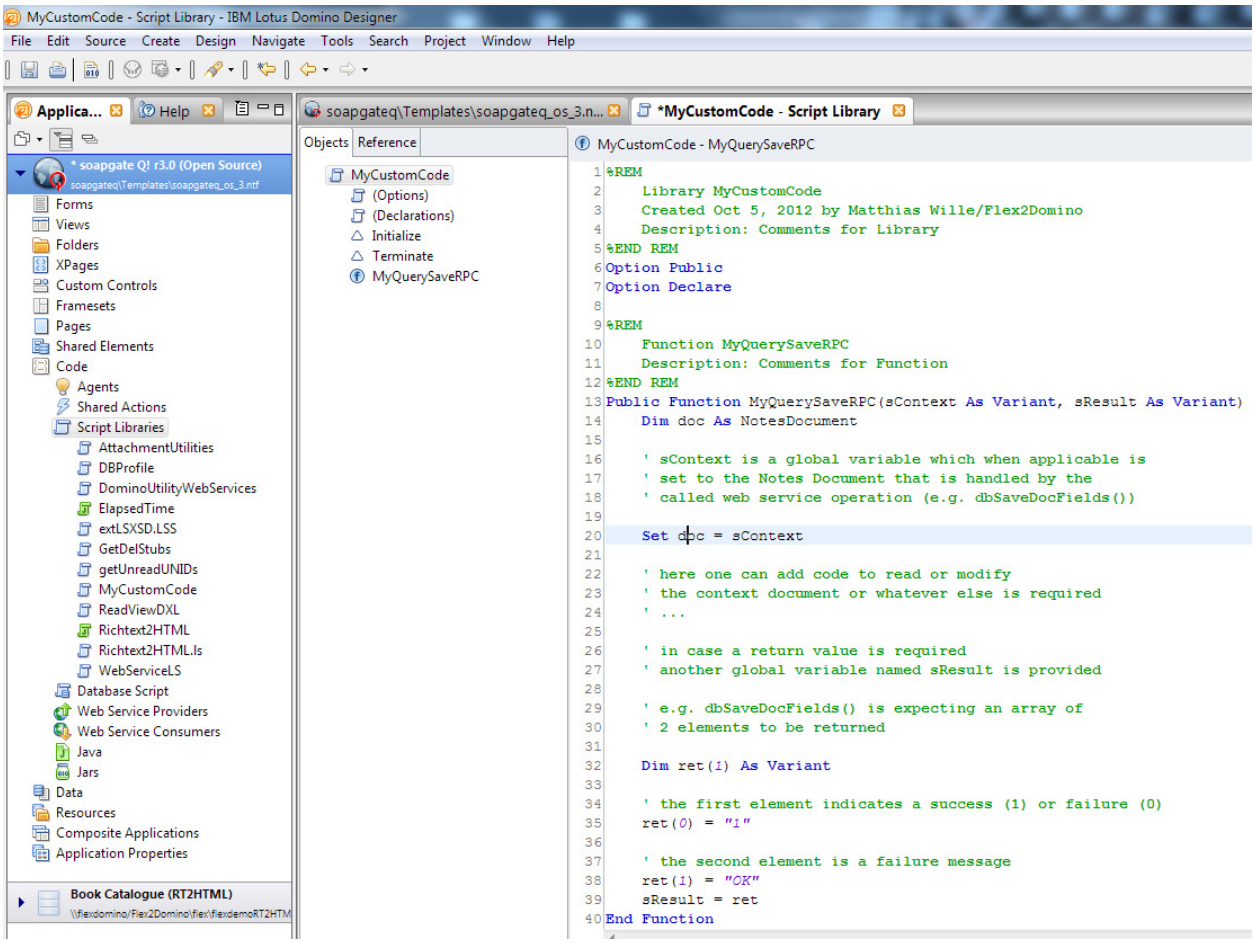


Code Profiles of type Lotus Script

For Lotus Script Code Profiles, the script is entered into the Code field of the profile. To communicate with the API, two global variables are provided: sContext and sResult (see screen shot remarks).



**Tip:** it is possible to place Lotus Script code into Script Libraries for easier coding and maintenance. The script library must be created in the Soapgate Q! database. Taking the above example, a script library could look like this:




The code in the Code Profile must be replaced with the following two lines:

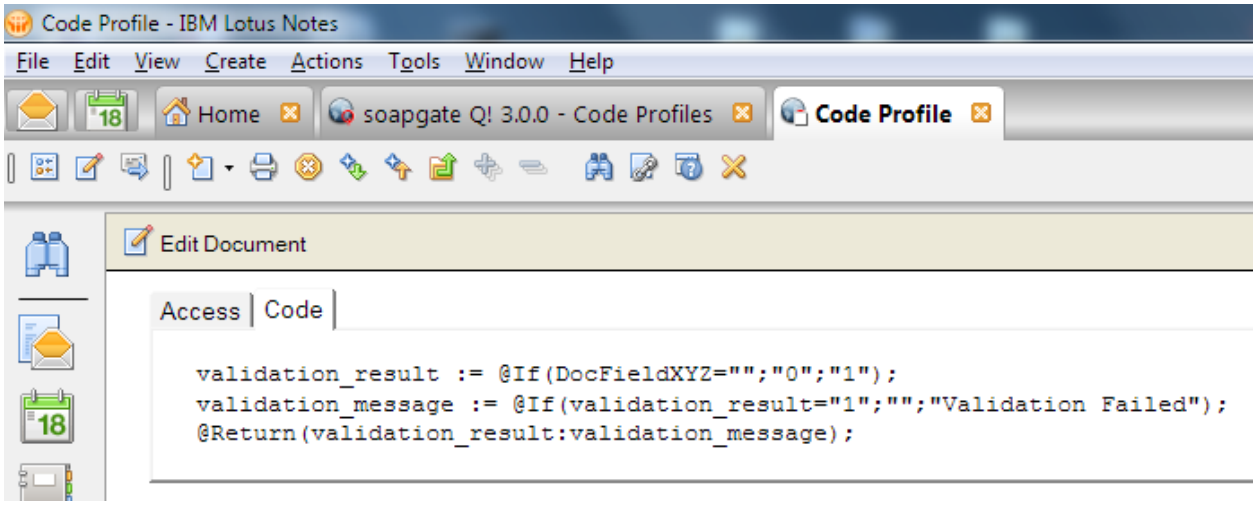
Use "MyCustomCode"

Call MyQuerySaveRPC(sContext,sResult)

Code Profiles of type Formula

For Formula Code Profiles, the Notes Formula is entered into the Code field of the profile. The formula is evaluated in the context of the document that is being processed. The formula therefore can directly reference fields on the document. However, READ ONLY!

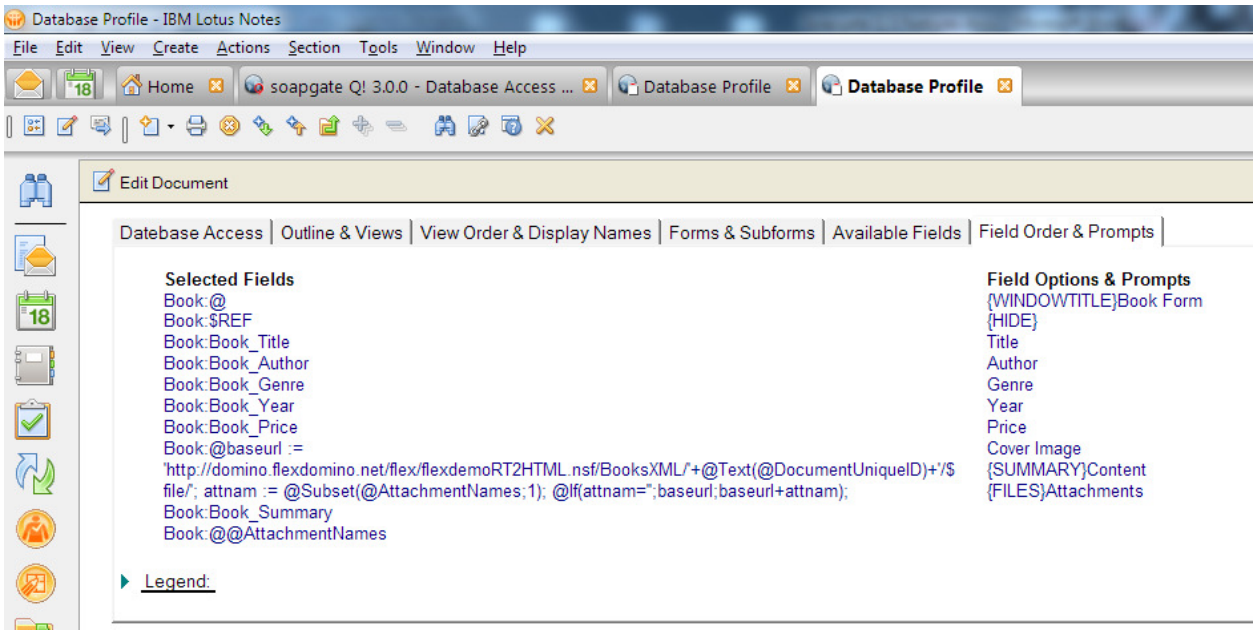
Using formula, a return value (where required) can be passed on to the API using the @Return() function.



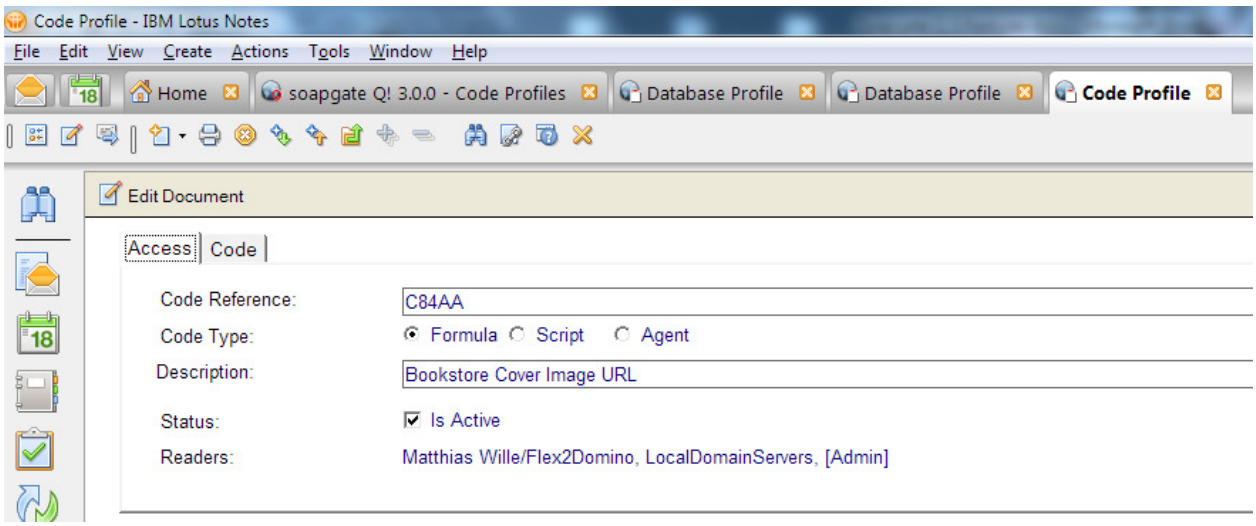
Referencing Code Profiles in Database Access Profiles

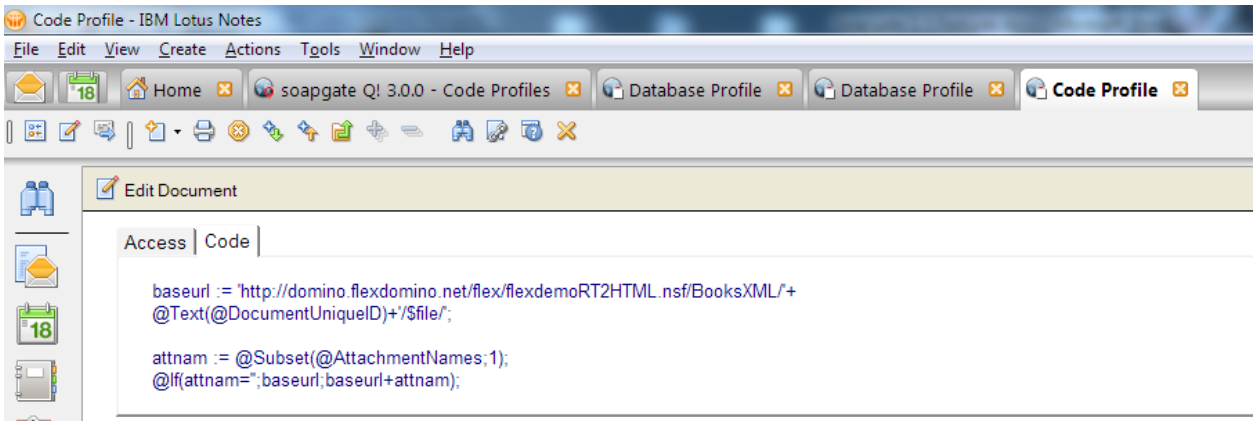
Though not required from a security point of view, Code Profiles also can be used in the Database Access Profiles. It easier to maintain short $xxx references then long formulae. And of course it is now possible to use Lotus Script to compute fields and prompts.

Taking our Bookstore sample database, the Database Access Profile used to look like this:




Note the long formula to compute the URL for the book cover image. It spans over several lines, making it difficult to match the Field List with the Prompts List. With Soapgate Q! we can now create a Code Profile for this:






...and simplify the Database Access Profile...