TopQuadrant™

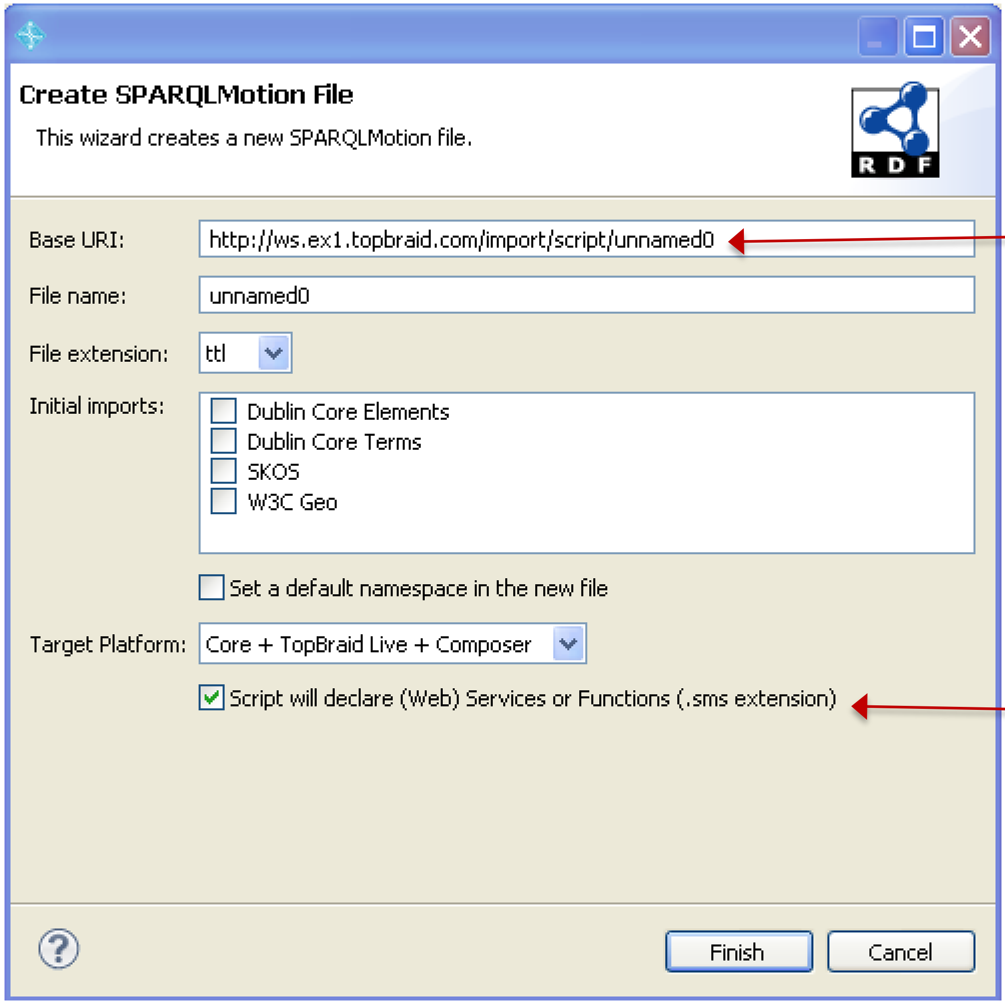
Creating a Semantic Web Service in 5 Easy Steps

Using SPARQLMotion in TopBraid Composer Maestro Edition

TopQuadrant™

Step 1: Create a SPARQLMotion file

In the Navigator View, select project or project folder where the file will be located. Right click and select New. Then select RDF/SPARQLMotion File (marked by a yellow star icon).



Create SPARQLMotion File

This wizard creates a new SPARQLMotion file.

Base URI:	http://ws.ex1.topbraid.com/import/script/unnamed0
File name:	unnamed0
File extension:	ttl
Initial imports:	☐ Dublin Core Elements ☐ Dublin Core Terms ☐ SKOS ☐ W3C Geo
	☐ Set a default namespace in the new file
Target Platform:	Core + TopBraid Live + Composer
	☑ Script will declare (Web) Services or Functions (.sms extension)

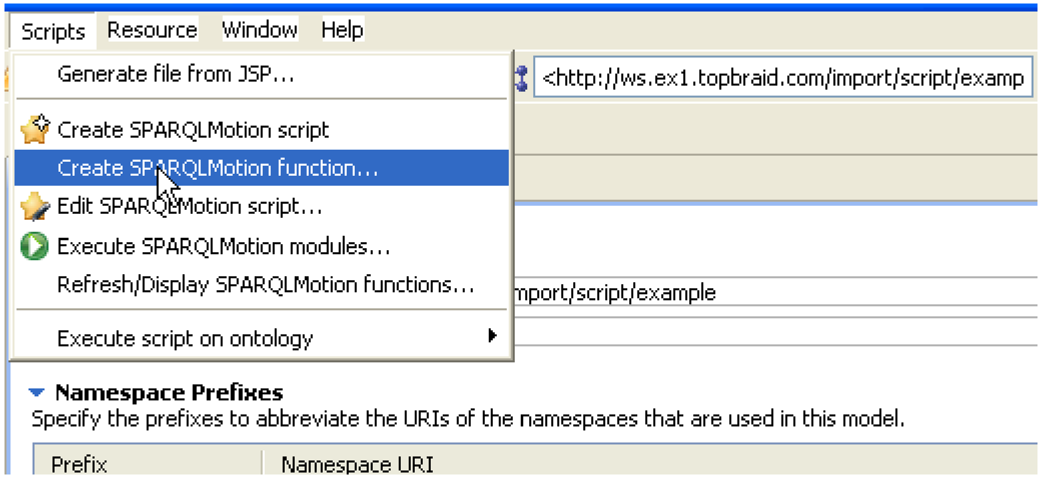
Name your script file by changing the Base URI in the Create wizard.

Make sure to check this box

This is the option that enables you to create web services

Step 2: Create a SPARQLMotion function

With the newly created SPARQLMotion file opened, go to Scripts menu and select Create SPARQLMotion function…

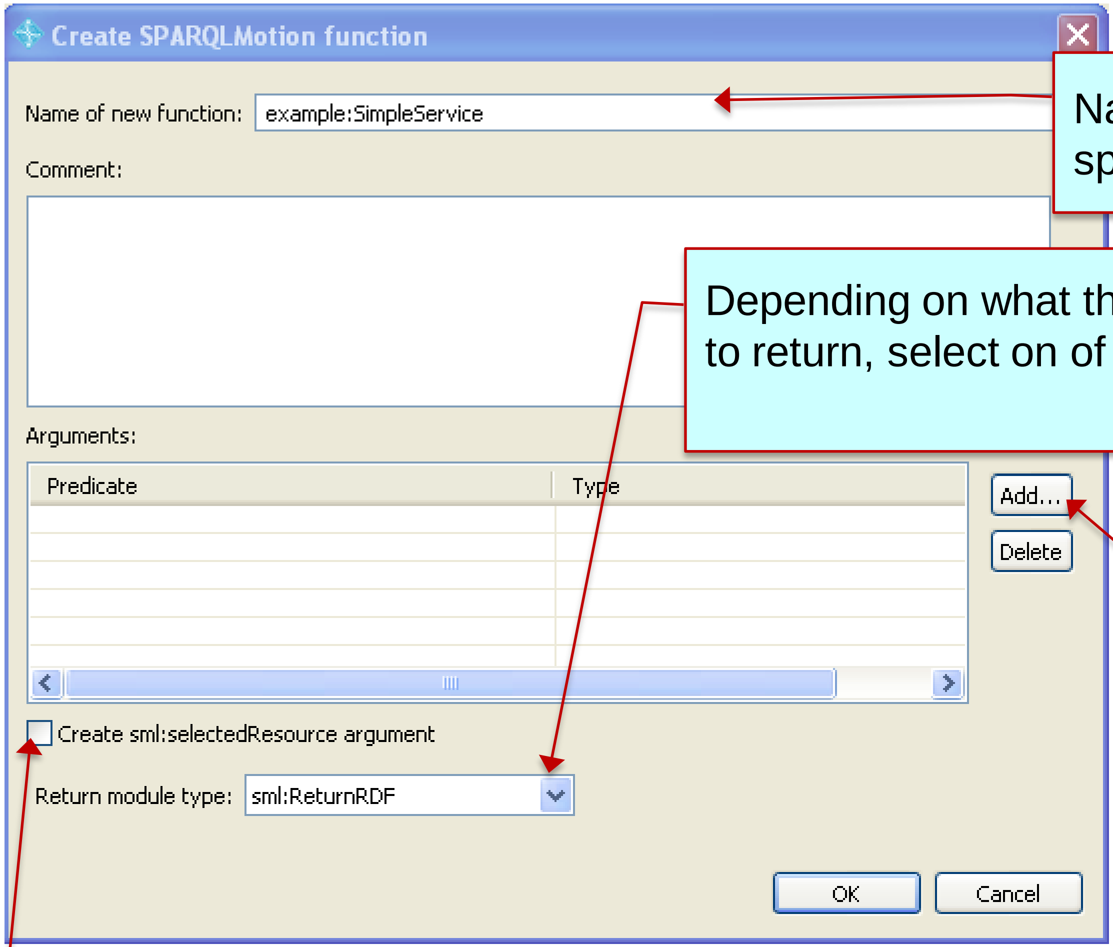

Scripts
Resource
Window
Help
Generate file from JSP...
Create SPARQLMotion script
Create SPARQLMotion function...
Edit SPARQLMotion script...
Execute SPARQLMotion modules...
Refresh/Display SPARQLMotion functions...
Execute script on ontology
<http://ws.ex1.topbraid.com/import/script/examp
mport/script/example
Namespace Prefixes
Specify the prefixes to abbreviate the URIs of the namespaces that are used in this model.
Prefix
Namespace URI

SPARQLMotion functions are web services. Each file can contain multiple functions.

Note: *You will also see Create SPARQLMotion script option. This option simply opens a canvas where you can start assembling modules. However, the resulting assembly is not a web service. It can be executed in TBC, but can not be called using REST interface.*

Step 3: Name the service (function) and identify what it will return

Name the function. Select the return module type. If the service needs to have any input arguments, click on the Add button to add them.



Create SPARQLMotion function

Name of new function: example:SimpleService

Comment:

Arguments:

Predicate	Type

☐ Create sml:selectedResource argument

Return module type: sml:ReturnRDF

Name your function. Remember to specify prefix.

Depending on what the service needs to return, select on of the module type

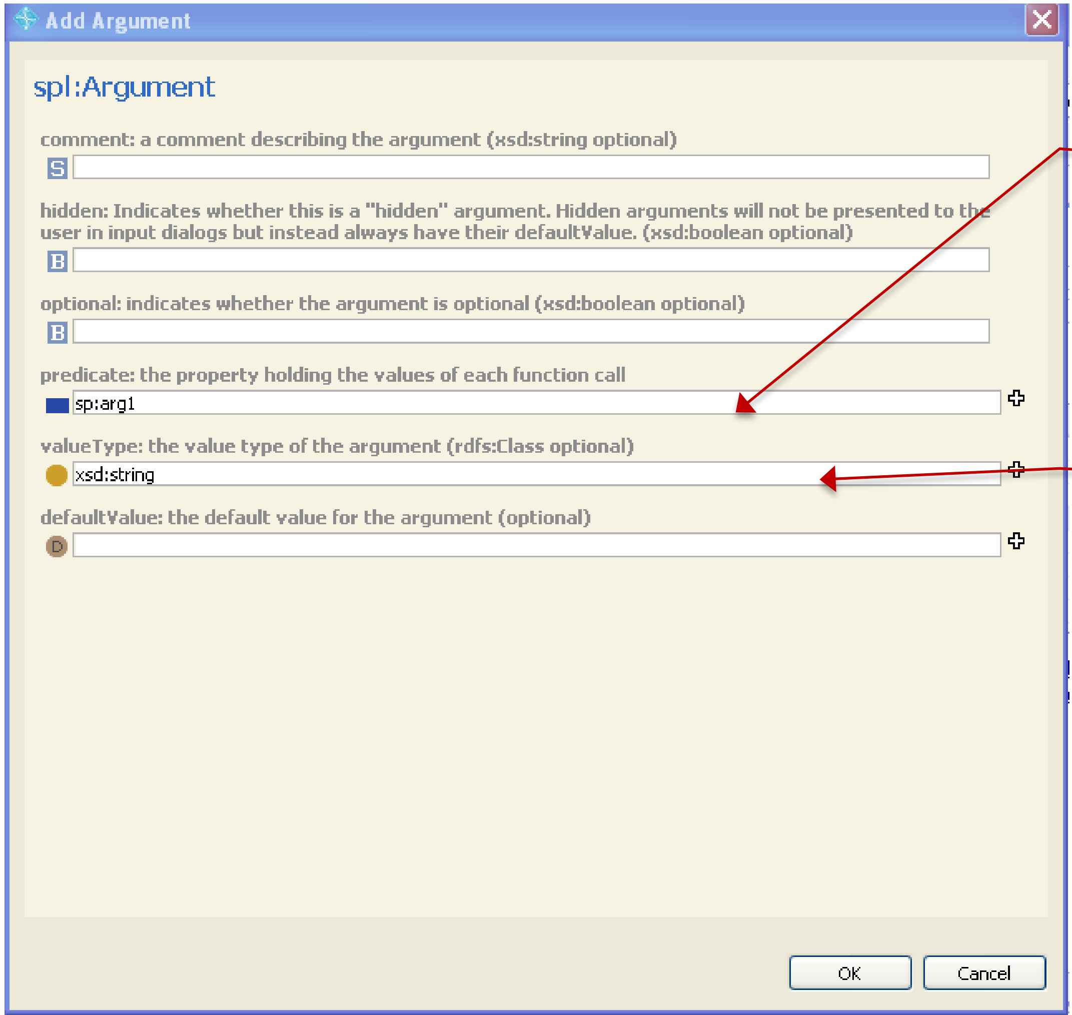
Next step (step 4) will be to specify arguments

This option is used only for functions executed in TBC or called from TopBraid Ensemble and Flex SDK applications

Step 4: Add input arguments

To add an argument, identify a predicate and (optionally) the value type.

You can type in the field or use a select dialog invoked by a "+" icon. Any property can be used as a predicate including built-in generic sp:arg1, sp:arg2, etc.



spl:Argument

comment: a comment describing the argument (xsd:string optional)

hidden: Indicates whether this is a "hidden" argument. Hidden arguments will not be presented to the user in input dialogs but instead always have their defaultValue. (xsd:boolean optional)

optional: indicates whether the argument is optional (xsd:boolean optional)

predicate: the property holding the values of each function call
sp:arg1

valueType: the value type of the argument (rdfs:Class optional)
xsd:string

defaultValue: the default value for the argument (optional)

Specify a property. If you want to use a custom property, it should already be created.
You can cancel function creation and return later if you need to add new properties.

Type or select value type of the argument.

Click OK and return to the previous dialog. Add as many arguments as you need.

Then press OK to complete function creation.

Step 5: Add processing modules and connect everything together

You are now in the main SPARQLMotion canvas where you can add processing modules that will do the work of your web service. Callouts below point out some of the important UI features.

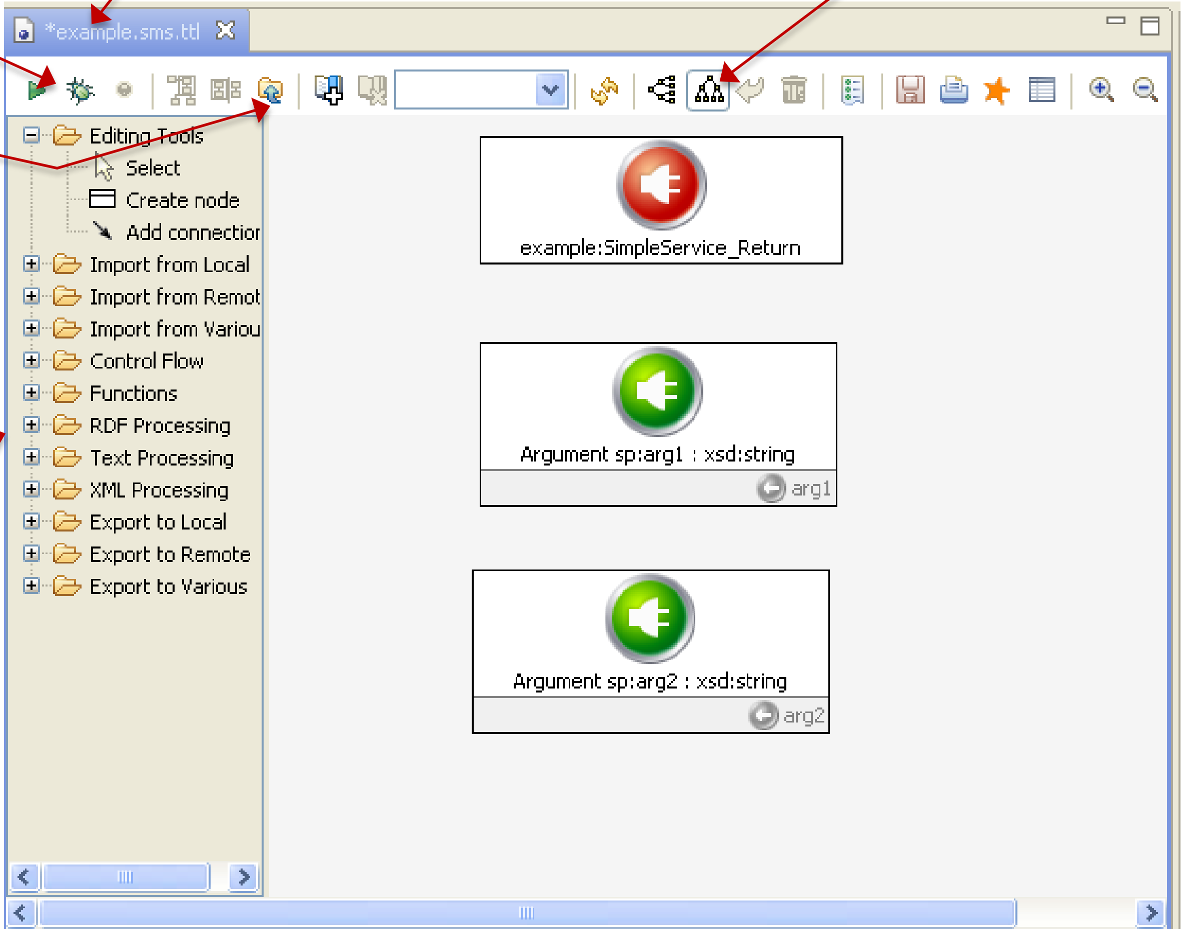
Double click to maximize/minimize

Click to switch to a vertical layout

Run and debug buttons

A button to Navigate to the function. Needed if, for example, you want to later add another argument.

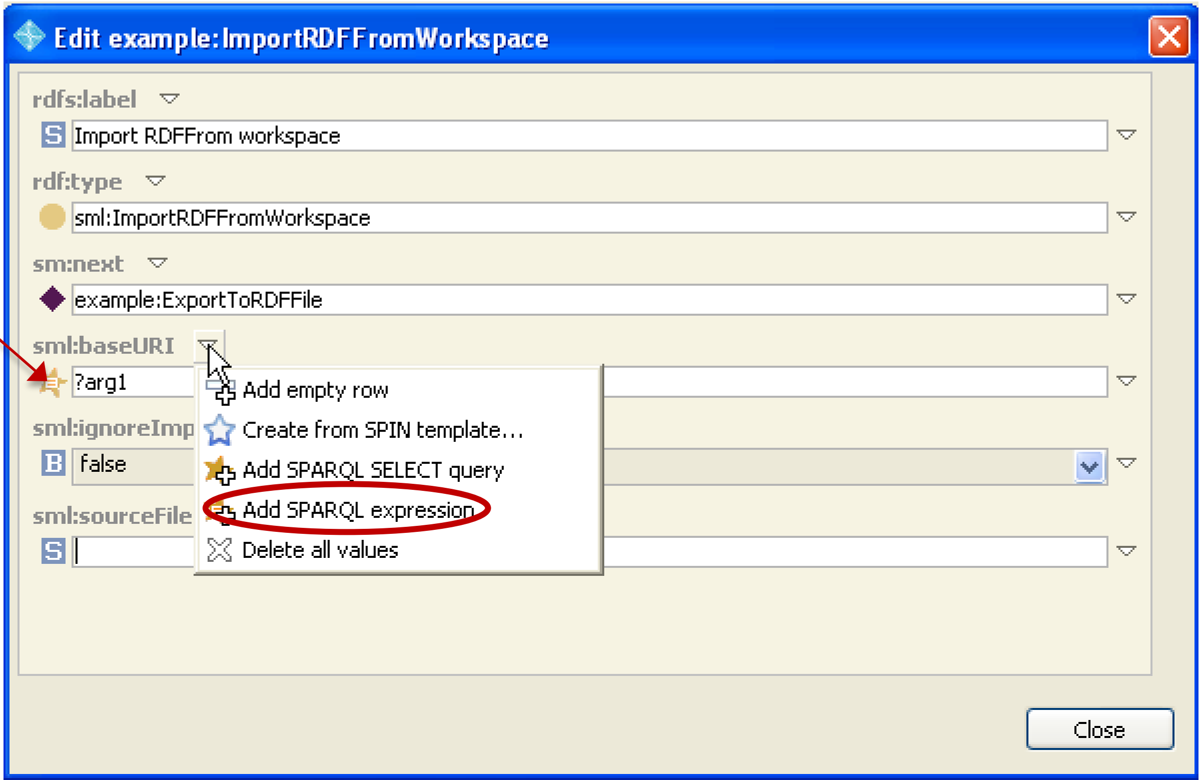
Expand folders to see and select modules as required.



SPARQLMotion offers well over 100 pre-built modules

- Modules are selected from the palette and connected using ‘next’ property
- Double clicking on a module opens a form
  - Forms show information appropriate to the module type
- RDF flows in and out of the modules
  - If a module receives RDF stream as an input and outputs RDF, it has sml:replace property. When set to ‘true’, only the output is passed to the next module.

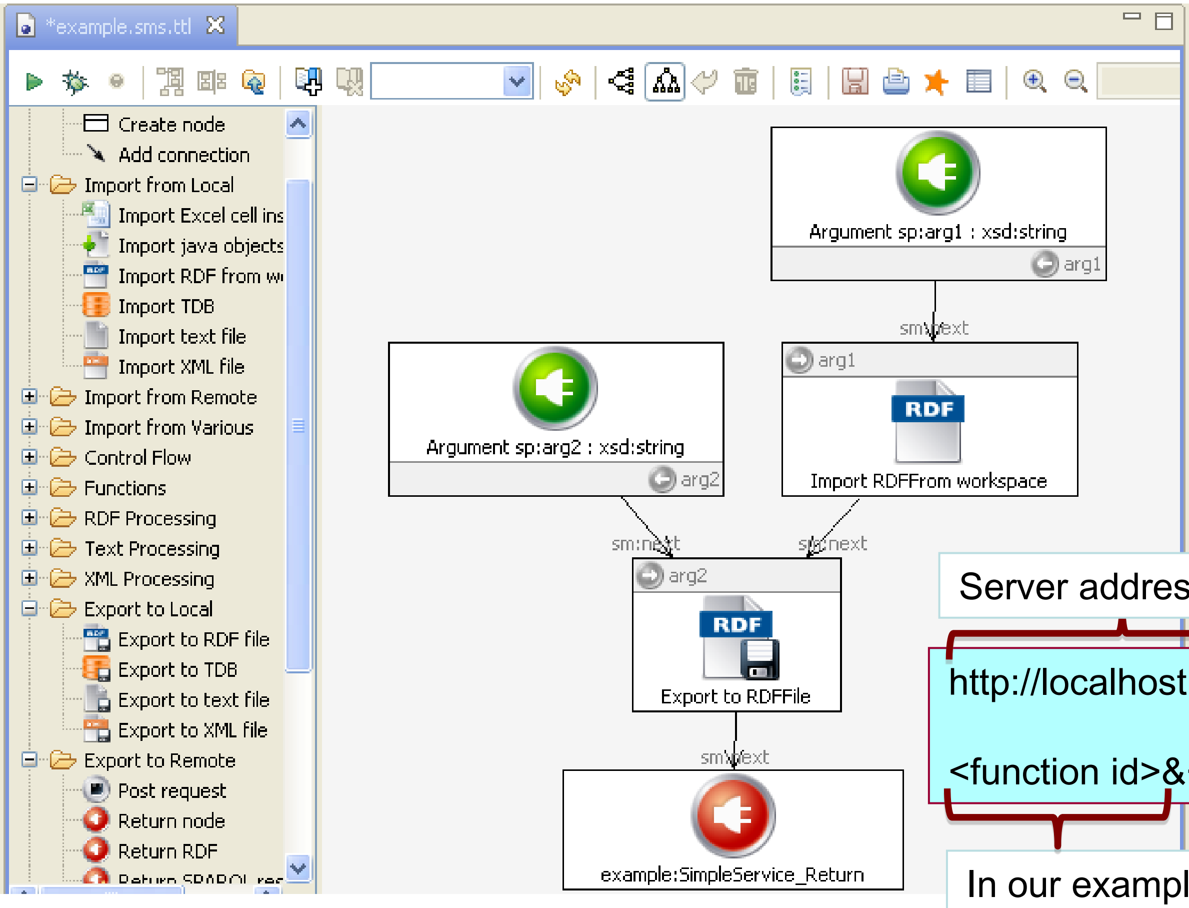
Here is how we are passing an input argument to the SPARQL module. It is done SPARQL style using ? And the variable name


Edit example:ImportRDFFromWorkspace

rdfs:label	Import RDFFrom workspace
rdf:type	sml:ImportRDFFromWorkspace
sm:next	example:ExportToRDFFile
sml:baseURI	?arg1
sml:ignoreImp	false
sml:sourceFile	

You are done!

You can test the new service directly in TBC. Or you can run it in the browser.



A simple service shown here imports RDF file in one serialization and exports it in another serialization.

Serialization is determined by the file extension specified in the targetFilePath.

To run a service use the following URL format:

Server address

http://localhost:8083/tbl/actions?action=sparqlmotion&id=

<function id>&<argument id>=<argument value>

In our example, it is example:SimpleService

Before running the service for the first time (and after changing it later), go to Scripts menu and select Refresh/Display SPARQLMotion functions to make sure that the new definition of the service is registered with the system.