



**NetApp™**  
Go further, faster

NETAPP INTERNAL DOCUMENT

## WFA Web services primer

Yaron Haimsohn, NetApp  
October 2011 | DRAFT v 1.5

## TABLE OF CONTENTS

<b>1</b>	<b>PURPOSE</b>	<b>3</b>
<b>2</b>	<b>GENERAL</b>	<b>3</b>
2.1	notes	3
2.2	Revisions	3
<b>3</b>	<b>SECURITY REQUIREMENT</b>	<b>5</b>
<b>4</b>	<b>WEB SERVICES</b>	<b>6</b>
4.1	Get ALL workflows	6
4.2	Get ALL workflows BY CATEGORY	7
4.3	Execute Workflow	8
4.4	SCHEDULE WORKFLOW	9
4.5	Get Job Status	10
4.6	Get command arguments	11
4.7	Get Return Parameters	13
<b>5</b>	<b>CREATING A WEB SERVICE PROXY WITH MICROSOFT POWERSHELL</b>	<b>15</b>
	<b>APPENDIX – RETURN PARAMETERS – FEATURE DESCRIPTION</b>	<b>18</b>

## 1 PURPOSE

The purpose of this document is to describe the web services externalized by WFA and the way to utilize them.

## 2 GENERAL

WFA supports utilization and monitoring of its workflows via web-services. Web services are supported using two common WSDL SOAP binding:

- Document/Literal encoding – Mostly used by Java applications

[http://<IP address>/wfa-ws/WorkflowService\\_doc?wsdl](http://<IP address>/wfa-ws/WorkflowService_doc?wsdl)

- RPC encoding – Mostly used by .Net, C# and PowerShell

[http://<IP address>/wfa-ws/WorkflowService\\_rpc?wsdl](http://<IP address>/wfa-ws/WorkflowService_rpc?wsdl)

### 2.1 NOTES

For more information regarding web services, please refer to the following document:

<http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl/>

### 2.2 REVISIONS

- 1.1 :
  - Added encoding common usage
  - Added WS overview document reference
- 1.2 :
  - Added new web service
  - Added WS security requirement
- 1.3.1:
  - Updated definition of services
  - Inclusion of security requirements
- 1.4:
  - Added web service – *GetReturnParameters*
- 1.5
  - Added web service – *GetAllWorkflowsByCategory*
  - Added web service – *ScheduleWorkflow*
  - Added appendix – Return Parameter feature



### 3 SECURITY REQUIREMENT

Every web service request must be accompanied by a valid username and password. These properties of the request must be set in order for the request to be performed. Otherwise – An “**401 Unauthorized**” response will be issued.

Only users of “Operator” level and above are allowed to execute workflows either via web services or from the portal. The request “Username” and “Password” properties must be set accordingly.

## 4 WEB SERVICES

### 4.1 GET ALL WORKFLOWS

**Usage:** GetAllWorkflows()

**Description:** Return all workflows currently defined in WFA. The result set will include workflows regardless of their production release status.

**Input parameters:** none

**Returned value:**

The method returns an array of workflows. Every entry would include:

- Workflow ID
- Workflow name
- Workflow description
- Workflow category/categories
- Array of allowed user inputs (As applicable, if available):
  - Parameter name
  - Parameter description
  - Parameter default value
  - Parameter type (String, Enumeration or Query)
  - Possible values (in case of Enumeration and Query only)

Example of a workflow entry:

description	: Resize a selected volume by a certain size, without changing any additional parameters
id	: 5
name	: Resize existing volume
userInput	: {ArrayIP, VolName, TargetSize}

## 4.2 GET ALL WORKFLOWS BY CATEGORY

**Usage:** GetAllWorkflowsByCategory(Category)

**Description:** Return workflows of a certain designated category. The result set will include workflows regardless of their production release status.

**Input parameters:**

<b>Name</b>	<b>Type</b>	<b>Mandatory</b>	<b>Description</b>
Categories	String	False	List of categories to retrieve

**Returned value:**

The method returns an array of workflows (Same as GetAllWorkflows). Every entry would include:

- Workflow ID
- Workflow name
- Workflow description
- Workflow category/categories
- Array of allowed user inputs (As applicable, if available):
  - Parameter name
  - Parameter description
  - Parameter default value
  - Parameter type (String, Enumeration or Query)
  - Possible values (in case of Enumeration and Query only)

Example of a workflow entry:

description	: Resize a selected volume by a certain size, without changing any additional parameters
id	: 5
name	: Resize existing volume
userInput	: {ArrayIP, VolName, TargetSize}

### 4.3 EXECUTE WORKFLOW

**Usage:** ExecuteWorkflow(WorkflowID, Parameter String)

**Description:** Execute a designated workflow with the given parameters

**Input parameters:**

<u>Name</u>	<u>Type</u>	<u>Mandatory</u>	<u>Description</u>
Workflow ID	Integer	true	The ID of the workflow as returned from GetAllWorkflows service
Input parameters	String	false	Input parameters A string of pairs in (name=value) format

Example for execution:

```
$jobid = $wfa.executeWorkflow(1, ("ArrayIP=10.68.66.214", "AggrName=EMBHNA_aggr1", "VolName=TestWS"))
```

**Returned value:**

A JobID for the executed workflow is returned

**Notes:**

- If the format of user inputs pair string is incorrect the service will return the following error message:  
***"Invalid user-input value 'some\_input\_with\_value', should be name=value"***
- Default values are used in case a user input value not provided (If applicable)



#### 4.4 SCHEDULE WORKFLOW

**Usage:** ScheduleWorkflow(WorkflowID, Parameter String, Execution Date and Time)

**Description:** Schedule a workflow for execution with the given parameters

**Input parameters:**

<u>Name</u>	<u>Type</u>	<u>Mandatory</u>	<u>Description</u>
Workflow ID	Integer	true	The ID of the workflow as returned from GetAllWorkflows service
Input parameters	String	false	Input parameters A string of pairs in (name=value) format
Execution Date and Time	String	True	Execution date and time. For example: 9/7/11 2:59 PM

Example for execution:

```
$jobid = $wfa.scheduleWorkflow(1, ("ArrayIP=10.68.66.214", "AggrName=EMBHNA_aggr1", "VolName=TestWS"), "9/7/11 2:59 PM")
```

**Returned value:**

A JobID for the workflow (scheduled to be executed) is returned

**Notes:**

- If the format of user inputs pair string is incorrect the service will return the following error message:  
*"Invalid user-input value 'some\_input\_with\_value', should be name=value"*
- Default values are used in case a user input value not provided (If applicable)

#### 4.5 GET JOB STATUS

**Usage:** GetJobStatus(JobID)

**Description:** Return execution status for a certain JobID (Not limited to jobs executed via Web services).

**Input parameters:**

<b>Name</b>	<b>Type</b>	<b>Mandatory</b>	<b>Description</b>
JobID	Integer	true	JobID as returned from ExecuteWorkflow or taken from the monitoring GUI

**Returned value:**

- Job status – One of the following:
  - SCHEDULED
  - PENDING
  - RUNNING
  - COMPLETED
  - FAILED
  - ABORTING
  - CANCELED
  - OBSOLETE
  - ABORTED
- Job type – Cache / Workflow execution
- Schedule type – Delayed / Immediate / Recurring
- Start time
- End time
- Planned execution time
- Error message

Example of returned values:

endTime	: Apr 27, 2011 9:22:47 AM
errorMessage	:
jobStatus1	: COMPLETED
jobType	: Workflow Execution - OVM RAC/No RAC Provisioning
plannedExecutionTime	: Apr 27, 2011 9:17:49 AM
scheduleType	: Immediate
startTime	: Apr 27, 2011 9:17:52 AM

**Notes:**

- If job ID not found the service will return the following error message:  
***“No job found for id: ‘10”***

#### 4.6 GET COMMAND ARGUMENTS

**Usage:** GetCommandArguments(JobID, CommandDisplayName, RowNumber, parameterName)

**Description:** Returns command execution arguments as stored in monitoring preview table (the values of the cells). This WS has two modes:

- Retrieve whole row of specific command
- Retrieve specific parameter in a row of specific command

**Input parameters:**

<u>Name</u>	<u>Type</u>	<u>Mandatory</u>	<u>Description</u>
JobID	Integer	true	JobID as returned from ExecuteWorkflow or taken from the monitoring GUI
CommandDisplayName	String	true	Name of the command in the workflow (Allows distinction if the same command is used twice)
RowNumber	Integer	true	Row in the workflow for which values will be returned
ParameterName	String	false	Request of a specific parameter value

**Returned value:**

- In case ParameterName was specified: A single value=name pair
- In case ParameterName was not specified: an array of name=value pairs

**Notes:**

- **This service is replaced by the newer GetReturnParameters service (But is still available for backward compatibility).**
- The service will returns data only if job status is in final state. It must be in one of the following statuses: COMPLETED, FAILED or ABORTED
- If status of the job in question is not one of above then the service will return the following error message:  
***“The job’s status is RUNNING , data can be retrieved only in the following statuses: COMPLETED, FAILED, ABORTED”***
- If the service can’t find execution results according to provided job id the the service will return the following error message:  
***“Failed to find workflow execution result by provided job id ‘10”***
- If the requested row in the workflow cannot be found the service will return the following error message:  
***“Row number ‘10’ not found”***

- If the workflow cannot find the requested command as per the CommandDisplayName provided, the service will return the following error message:  
***“Command display name ‘no such command’ not found”***
- If workflow cannot resolve the requested parameter name the service will return the following error message:  
***“Parameter name ‘no such parameter’ not found in command ‘Create Volume”***

Example for usage (Without specifying a parameter name):

```
$params = $wfa.getCommandArguments(54,"Create Volume",1,"")
```

Example of return values:

name	value
----	----
Size	200
Filer	10.68.66.215
SpaceReserve	none
VolumeName	n_data_0000000Yaron_0
AggregateName	emvlna_aggr1
SnapshotReservePercentage	50
AutoDeleteOptions	state on,commitment try,delete_order oldest_first
SnapshotScheduleOptions	0 7
Options	nosnap on,create_ucose on,try_first snap_delete
Create	true

#### 4.7 GET RETURN PARAMETERS

**Usage:** GetReturnParameters(JobID, @(comma separated list of parameter names))

**Description:** Returns values of parameters that were set for extraction in a certain workflow as they were calculated in a certain execution of said workflow. More information on this feature can be found in this document appendix. This WS has two modes:

- Retrieve all designated parameters and their values
- Retrieve specific parameters and their values

**Input parameters:**

<u>Name</u>	<u>Type</u>	<u>Mandatory</u>	<u>Description</u>
JobID	Integer	true	JobID as returned from ExecuteWorkflow or taken from the monitoring GUI
ParameterName	Array	False*	Names of specific parameters to be extracted as per execution

**Returned value:**

- In case ParameterName was specified: A single value=name pair
- In case ParameterName was not specified: an array of name=value pairs

**Notes:**

- Array of parameter names or empty array is mandatory. In case all the parameters are requested, use '@()' (in case of PowerShell WS command line client)
- The service will returns data only if job status is in final state. It must be in one of the following statuses: COMPLETED, FAILED or ABORTED
- **Values are returned as per planning phase.** Failure of workflow will not result in any change to the returned values.
- If status of the job in question is not one of above then the service will return the following error message:  
***“The job’s status is RUNNING , data can be retrieved only in the following statuses: COMPLETED, FAILED, ABORTED”***
- If the service can’t find execution results according to provided job id the the service will return the following error message:  
***“Failed to find workflow execution result by provided job id ‘10”***
- If workflow cannot resolve the requested parameter name the service will return the following error message:  
***“Parameter ‘qtree\_name’ not found. It’s not defined as a return parameter”***

Example for usage (Without specifying a parameter name):

```
$returnParams = $wfa. $wfa.getReturnParameters(39, @())
```

Example of return values:

```
$ returnParams | Format-table -auto
```

name	value
------	-------

----	-----
------	-------

vol_name	WS_test
----------	---------

Path	WS_test/test_qt
------	-----------------

## 5 CREATING A WEB SERVICE PROXY WITH MICROSOFT POWERSHELL

- 1) Open Microsoft Powershell Shell by "Start -> Run"
- 2) Setup the password and credentials for meeting the security requirements

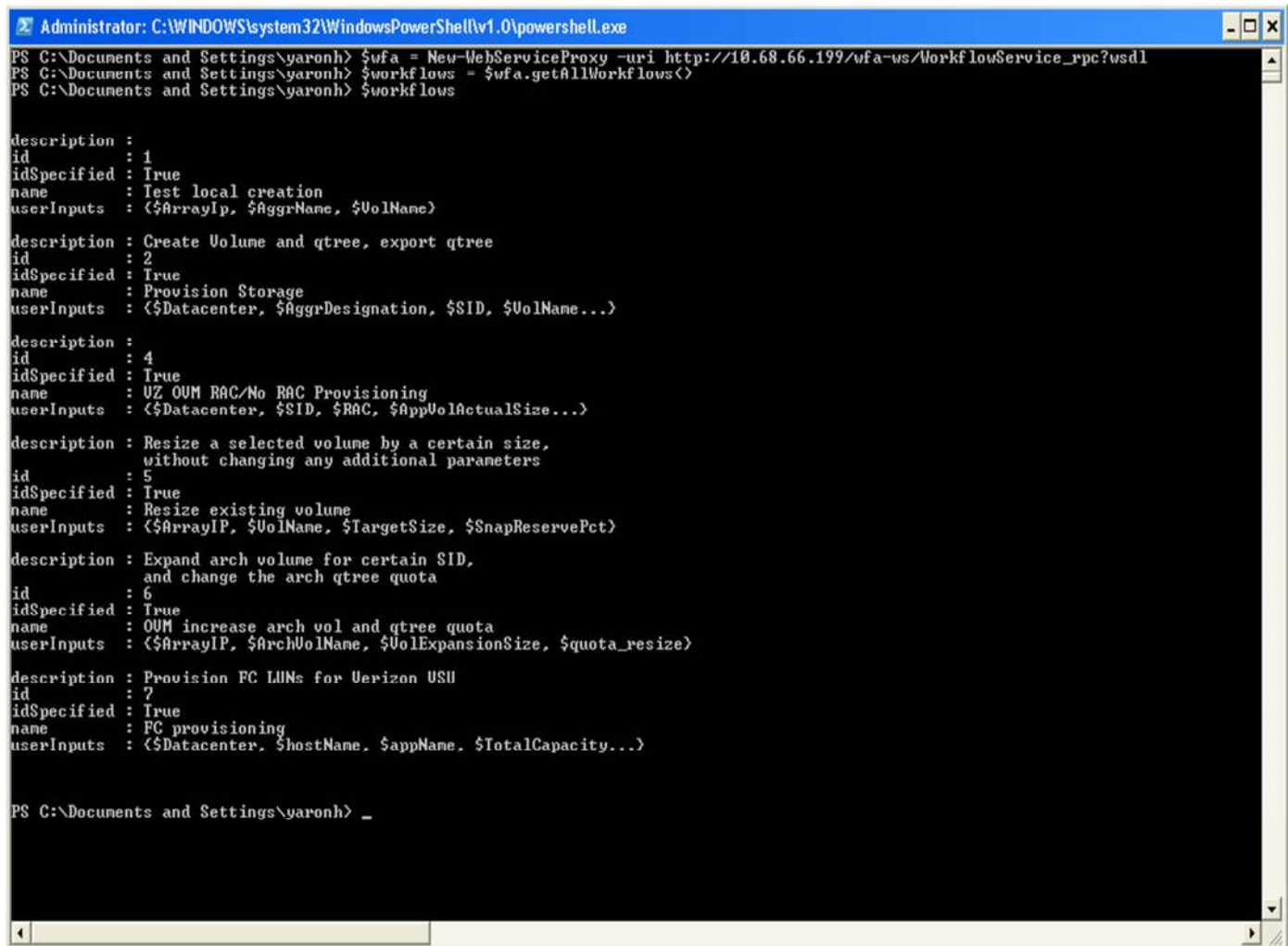
```
$password = ConvertTo-SecureString "admin" -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential ("admin", $password)
```

- 3) Create the web services proxy using the WFA installation

```
$wfa = New-WebServiceProxy -Uri http://<WFA IP>/wfa-ws/WorkflowService_rpc?wsdl -
Credential $credential
```

- 4) List all workflows by using:

```
$workflows = $wfa.getAllWorkflows()
and print it by typing:
$workflows
```



```
Administrator: C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe
PS C:\Documents and Settings\yaronh> $wfa = New-WebServiceProxy -uri http://10.68.66.199/wfa-ws/WorkflowService_rpc?wsdl
PS C:\Documents and Settings\yaronh> $workflows = $wfa.getAllWorkflows()
PS C:\Documents and Settings\yaronh> $workflows

description :
id           : 1
idSpecified : True
name        : Test local creation
userInputs  : (<Array>IP, $AggrName, $VolName)

description : Create Volume and qtree, export qtree
id           : 2
idSpecified : True
name        : Provision Storage
userInputs  : (<Datacenter, $AggrDesignation, $SID, $VolName...>)

description :
id           : 4
idSpecified : True
name        : UZ OUM RAC/No RAC Provisioning
userInputs  : (<Datacenter, $SID, $RAC, $AppVolActualSize...>)

description : Resize a selected volume by a certain size,
              without changing any additional parameters
id           : 5
idSpecified : True
name        : Resize existing volume
userInputs  : (<Array>IP, $VolName, $TargetSize, $SnapReservePct)

description : Expand arch volume for certain SID,
              and change the arch qtree quota
id           : 6
idSpecified : True
name        : OUM increase arch vol and qtree quota
userInputs  : (<Array>IP, $ArchVolName, $VolExpansionSize, $quota_resize)

description : Provision FC LUNs for Verizon USU
id           : 7
idSpecified : True
name        : FC provisioning
userInputs  : (<Datacenter, $HostName, $appName, $TotalCapacity...>)

PS C:\Documents and Settings\yaronh> _
```

5) List the parameters of a workflow (Count workflows, starting with 0:

```
$workflows[3].userInput|Format-Table -auto
```

6) Execute a workflow (No need to pass values for variables whose defaults are accepted)

```
$jobid = $wfa.executeWorkflow(1, ("ArrayIP=10.68.66.214", "AggrName=EMBHNA_aggr1",  
"VolName=TestWS"))
```

7) Monitor status of the returned JobID:

```
$status = $wfa.getJobStatus(54)  
$status  
  
endTime       : Apr 27, 2011 9:22:47 AM  
errorMessage  :  
jobStatus1    : COMPLETED  
jobType       : Workflow Execution - OVM RAC/No RAC Provisioning  
plannedExecutionTime : Apr 27, 2011 9:17:49 AM  
scheduleType  : Immediate  
startTime     : Apr 27, 2011 9:17:52 AM
```

8) Retrieve parameter value from execution:

```
$Arg = $wfa.getCommandArguments  
      (54, "Create Volume",1," AggregateName")  
  
and print it by typing:  
$Arg  
  
The result would be:  
name      value  
----      -  
AggregateName      emvlna_aggr0
```



In the same manner, without selecting a specific parameter :

```
$Arg = $wfa.getCommandArguments(54, "Create Volume",1,"")
```

```
$Arg
```

name	value
----	----
Size	111
SpaceReserve	none
Filer	10.68.66.215
SnapshotScheduleOptions	0 7
VolumeName	n_yaron_000000123456_0
Create	true
AggregateName	emvlna_aggr0
SnapshotReservePercentage	10

9) Retrieve return parameters from workflow:

To retrieve all parameters designated for extraction use

```
$Ret = $wfa.getReturnParameters(39, @())
```

and print it by typing:

```
$Ret
```

The result would be ( for "\$Ret | Format-list")

```
name : vol_name
```

```
value : WS_test
```

```
name : Path
```

```
value : WS_test/test_qt
```

To retrieve a single specific parameter set for extraction use

```
$Ret = $wfa.getReturnParameters(39, @('vol_name'))
```

The result would be ( for "\$Ret | Format-list")

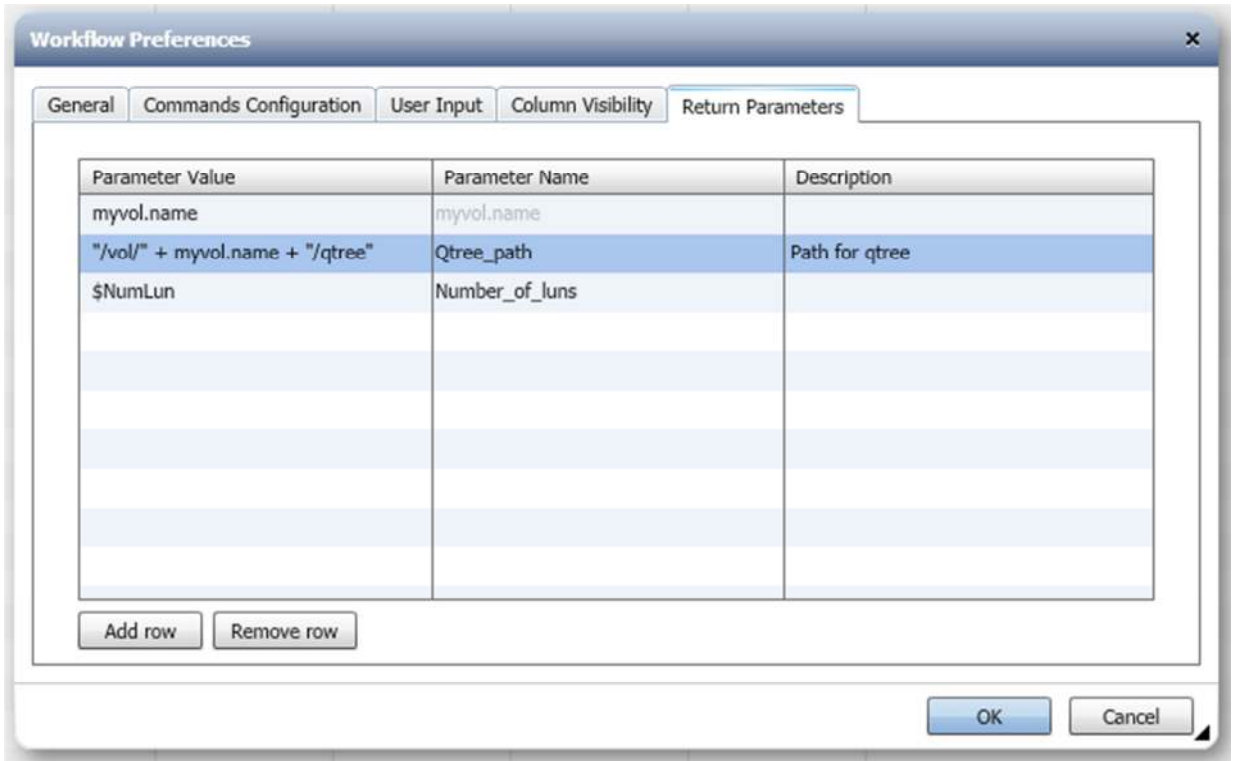
```
name : vol_name
```

```
value : WS_test
```

## APPENDIX – RETURN PARAMETERS – FEATURE DESCRIPTION

This feature allows designating a set of parameters (Variable attributes, expressions, user input values etc.) in a workflow, and to retrieve the values for said parameters upon request.

The designation is done in the “Return parameters” tab in workflow preferences:



Start by adding a row and determining which value you wish to get in return. It would receive a label automatically, identical to the parameter value column, which would appear in light gray. You may change it to match any label you may like and even add a short description.

**Note:** When executing the workflow, the values would be populated as soon as the planning phase is completed and execution commences. Given that, it is crucial to test the workflow execution status and confirm its completion before addressing the values of the return parameters.

These values are set per execution. If another parameter was added to be returned after several executions, that parameter value would be available from that point onwards only, and not in any execution pre-dating its addition.