# TABLE OF CONTENTS

1 | PURPOSE ................................................................................................................................... 3
2 | GENERAL ................................................................................................................................... 3  
   | 2.1 notes .................................................................................................................................. 3 
   | 2.2 Revisions ............................................................................................................................ 3 
3 | SECURITY REQUIREMENT ...................................................................................................... 4 
4 | WEB SERVICES ......................................................................................................................... 5  
   | 4.1 Workflow Service .............................................................................................................. 5 
   | 4.2 User Service ....................................................................................................................... 16 
5 | CREATING A WEB SERVICE PROXY WITH MICROSOFT POWERSHELL ...................... 18 
APPENDIX A – RETURN PARAMETERS – FEATURE DESCRIPTION ...................................... 21 
APPENDIX B – ERROR MESSAGES ............................................................................................. 22
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 workflows and users via web-services. Web services are supported using two common WSDL SOAP binding:

- Document/Literal encoding – Mostly used by Java applications
  Workflows: http://<IP address>/wfa-ws/WorkflowService_doc?wsdl

- RPC encoding – Mostly used by .Net, C# and PowerShell
  Workflows: http://<IP address>/wfa-ws/WorkflowService_rpc?wsdl
  Users: http://<IP address>/wfa-ws/UserService_rpc?wsdl

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


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
• 1.6:
  o Added web service – PreviewWorkflow
• 1.7:
  o Added web service – RejectWorkflow
  o Added web service – ResumeWorkflow
• 1.8:
  o Changed parameter of resume workflow to jobID
  o Adding the comment to the return value of getJobStatus
• 1.9:
  o Finalizing document for WFA 1.1 release
  o Adding The Error messages table
• 2.0:
  o Finalizing document for WFA 1.1.1 release
  o Added web service – WaitForCompletion – Supporting Synchronous calls to WFA Web services API
• 2.1:
  o Added web service – GetAllUsers
  o Added web service – ChangePassword
  o Add error message for GetAllWorkflowsByCategories

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 WORKFLOW SERVICE

4.2.04.1.1 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, Query or Number)
  - Possible values

Example of a workflow entry:

<table>
<thead>
<tr>
<th>description</th>
<th>Resize a selected volume by a certain size, without changing any additional parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>5</td>
</tr>
<tr>
<td>name</td>
<td>Resize existing volume</td>
</tr>
<tr>
<td>userInput</td>
<td>{ArrayIP, VolName, TargetSize}</td>
</tr>
</tbody>
</table>
Get ALL workflows BY CATEGORIES

Usage: GetAllWorkflowsByCategories(Categories List)

Description: Return workflows of a certain designated category.

In case the user is of “Operator” role only “Ready for production” workflows will be returned.

Input parameters:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categories</td>
<td>String</td>
<td>False</td>
<td>List of categories to retrieve</td>
</tr>
</tbody>
</table>

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 for execution:

```$workflows = $wfa. getAllWorkflowsByCategories("7 mode","C mode")```

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}```
**4.4.04.1.3 RUN Workflow**

**Usage:** RunWorkflow(WorkflowID, Input Parameters List, Execution Date and Time, Comment)

**Description:** This method can either execute a workflow immediately or schedule it for a future date.

**Input parameters:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow ID</td>
<td>Integer</td>
<td>true</td>
<td>The ID of the workflow as returned from GetAllWorkflows service</td>
</tr>
<tr>
<td>Input parameters</td>
<td>Array</td>
<td>false</td>
<td>Input parameters Array of strings - pairs in (name=value) format</td>
</tr>
<tr>
<td>Execution Date and Time</td>
<td>String</td>
<td>false</td>
<td>Execution date and time. For example: 9/23/11 2:59 PM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If left empty the job will be executed immediately</td>
</tr>
<tr>
<td>Comment</td>
<td>String</td>
<td>false</td>
<td>Executing first try</td>
</tr>
</tbody>
</table>

Example for execution:

```
$jobid = $wfa.runWorkflow(1, ("ArrayIP=10.68.66.214", "AggrName=EMBHNA_aggr1", "VolName=TestWS"), "9/23/11 2:59 PM", "Executing")
```

**Returned value:**

A JobID for the workflow (scheduled to be executed on executed immediatly) 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)
- If no date and time is provided the workflow will be executed immediately
4.5.04.1.4  RESUME Workflow

Usage:  resumeWorkflow(jobID, Comment, Execution Date and Time)

Description:  resume or reschedule a workflow, that was already scheduled for execution, with the given job id (as returned in the scheduleWorkflow call) and comment

Input parameters:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job ID</td>
<td>Integer</td>
<td>true</td>
<td>The job id of the workflow as returned from ScheduleWorkflow service</td>
</tr>
<tr>
<td>Comment</td>
<td>String</td>
<td>false</td>
<td>Comment for the execution</td>
</tr>
<tr>
<td>Execution Date and Time</td>
<td>String</td>
<td>True</td>
<td>Execution date and time. For example: 9/23/11 2:59 PM, or null, if rescheduling for immediate execution</td>
</tr>
</tbody>
</table>

Example for execution:

$wfa.resumeWorkflow(1, ("approved execution"), null)

Returned value:

There is no return value for this service.

Notes:

- If the user executing the service is an operator who is not allowed to execute this workflow, due to workflow authorization restrictions (the workflow is assigned to a category, this user is not assigned to) the following error will be returned: "current user 'user name' is not allowed to resume workflow 'workflowId'

- Only workflows with the status Paused, Scheduled, Aborted or Failed can be resumed. If trying to resume a workflow with other status the following error will be returned: "Could not resume workflow execution with id 'the workflow job id'. Resume is only allowed from status 'PAUSED'."

- If trying to resume a failed workflow, which has expired, the following error will be returned: "Resume of failed workflow execution with id 'the workflow job Id' is not allowed. Workflow execution is expired - more than 'number of expiration days' days have passed since failure"
4.6.04.1.5 **reject Workflow**

**Usage:** rejectWorkflow(WorkflowExecutionID, Comment String)

**Description:** reject a workflow, that was already scheduled for execution, with the given job id (as returned in the scheduleWorkflow call) and comment

**Input parameters:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow Execution ID</td>
<td>Integer</td>
<td>true</td>
<td>The job id of the workflow as returned from ScheduleWorkflow service</td>
</tr>
<tr>
<td>Comment</td>
<td>String</td>
<td>false</td>
<td>Comment for the execution</td>
</tr>
</tbody>
</table>

Example for execution:

```javascript
$wfa.rejectWorkflow(1, ("rejecting workflow execution"))
```

**Returned value:**

There is no return value for this service.

**Notes:**

- If the user executing the service is an operator who is not allowed to execute this workflow, due to workflow authorization restrictions (the workflow is assigned to a category, this user is not assigned to) the following error will be returned: "**current user 'user name' is not allowed to reject workflow 'workflowId'**"
- Only workflows with the status Paused, Scheduled, Pending or Running can be rejected. No error will be returned if a user will try to reject a workflow with a wrong status
- If an operator is trying to reject a paused workflow, which an operator is not allowed to reject the following error will be returned: "**Users with 'Operator' role are not allowed to reject workflow executions**"
Get Job Status

**Usage:** GetJobStatus(JobID)

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

**Input parameters:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job ID</td>
<td>Integer</td>
<td>true</td>
<td>The Job ID as returned from executeWorkflow or taken from the monitoring GUI</td>
</tr>
</tbody>
</table>

**Returned value:**

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

Example of returned values:

<table>
<thead>
<tr>
<th>endTime</th>
<th>Apr 27, 2011 9:22:47 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>errorMessage</td>
<td></td>
</tr>
<tr>
<td>jobStatus1</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>jobType</td>
<td>Workflow Execution - OVM RAC/No RAC Provisioning</td>
</tr>
<tr>
<td>plannedExecutionTime</td>
<td>Apr 27, 2011 9:17:49 AM</td>
</tr>
<tr>
<td>scheduleType</td>
<td>Immediate</td>
</tr>
<tr>
<td>startTime</td>
<td>Apr 27, 2011 9:17:52 AM</td>
</tr>
<tr>
<td>comment</td>
<td>Creating oracle SID</td>
</tr>
</tbody>
</table>

**Notes:**

- If job ID not found the service will return the following error message:
  "No job found for id: ‘10’"
# Get Return Parameters

**Usage:** GetReturnParameters(JobID, Parameters Name List)

**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:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobID</td>
<td>Integer</td>
<td>true</td>
<td>JobID as returned from ExecuteWorkflow or taken from the monitoring GUI</td>
</tr>
<tr>
<td>Parameters Name List</td>
<td>Array</td>
<td>False*</td>
<td>Names of specific parameters to be extracted as per execution</td>
</tr>
</tbody>
</table>

**Returned value:**

- In case “Parameters Name” was specified: an array of name=value pairs with the specified parameters
- In case “Parameter Name” was not specified: an array of name=value pairs of all the return parameters of the workflow

**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 <Current Job Status>, 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 ‘parameter name’ not found. It’s not defined as a return parameter"
Example for usage (Without specifying a parameter name):

```powershell
$returnParams = $wfa.getReturnParameters(39, (vol_name,Path))
```

Example of return values:

```powershell
$returnParams | Format-table -auto
name value
---- -----
vol_name WS_test
Path WS_test/test_qt
```

**4.8.04.1.8 Preview Workflow**

**Usage:** `previewWorkflow(WorkflowID, Input Parameters List)`

**Description:** Preview a designated workflow with the given input parameters

**Input parameters:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow ID</td>
<td>Integer</td>
<td>true</td>
<td>The ID of the workflow as returned from GetAllWorkflows service</td>
</tr>
<tr>
<td>Input parameters</td>
<td>Array</td>
<td>false</td>
<td>Input parameters Array of strings - pairs in (name=value) format</td>
</tr>
</tbody>
</table>

**Returned value:**

- Retrieve all designated return parameters and their values in case of success
- Retrieve exception error message in case of fail.

**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)
- If Workflow ID not found the service will return the following error message:
  "No workflow found for id: ‘10”

Example for usage (Without specifying a parameter name):

```powershell
$returnParams = $wfa.previewWorkflow(1, (“ArrayIP=10.68.66.214”, “AggrName=EMBHNA_aggr1”, “VolName=TestWS”))
```
Example of return values:

```
$ returnParams | Format-table -auto

name          value
-----          ----- 
volume1.name   TestWS
volume1.array.ip  10.68.66.214
```
**4.9.04.1.9 get WorkFlow Return Parameters**

**Usage:** GetWorkflowReturnParameters(WorkflowID)

**Description:** Returns all the return parameters that were defined for a given workflow.

**Input parameters:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow ID</td>
<td>Integer</td>
<td>true</td>
<td>The workflow ID as returned from getAllWorkflows</td>
</tr>
</tbody>
</table>

**Returned value:**

- Retrieve all the return parameters that were defined for a workflow: ‘description’ (if exist), ‘Id’, ‘name’ (null if not exist) and ‘value’

**Notes:**

- If the requested workflow cannot be found, the service will return the following error message: ‘No workflow found for id: ‘14”

**Example for usage:**

```powershell
$returnParams = $wfa.getWorkFlowReturnParameters(3)
```

**Example of return values:**

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

<table>
<thead>
<tr>
<th>description</th>
<th>id</th>
<th>name</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>1</td>
<td>volume</td>
<td>$VolumeName</td>
</tr>
<tr>
<td>Aggregate</td>
<td>2</td>
<td>agg</td>
<td>$AggregateName</td>
</tr>
<tr>
<td>Array</td>
<td>3</td>
<td>array</td>
<td>$ArrayIP</td>
</tr>
</tbody>
</table>

---

**4.10.04.1.10 wait for completion**

**Usage:** WaitForCompletion(JobID)

**Description:** Synchronously waits till a workflow execution reaches one of the following statuses: COMPLETED, FAILED, PAUSED (waiting for approval)

**Input parameters:**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobID</td>
<td>Integer</td>
<td>true</td>
<td>JobID as returned from ExecuteWorkflow or taken from the monitoring GUI</td>
</tr>
</tbody>
</table>

**Returned value:**

Same as GetJobStatus

**Notes:**

- If the job matching the requested jobId is not in one of these statuses: SCHEDULED, PENDING, RUNNING, the service will return the following error message: ‘*Cannot wait on job with status: FAILED*’
- If the job matching the requested jobId is in status scheduled but is not scheduled to run during the next 24 hours, the service will return the following error message: ‘*Cannot wait for executions that are not scheduled to run in the next 24 hours*’
4.114 USER SERVICE

4.114.2 Get All Users

Usage: GetAllUsers()

Description: Returns all users currently defined in WFA.

Input parameters: none

Returned value:

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

- Username
- User role type (e.g. “Admin”)
- Array of category names authorized for the user
- Whether the user is Ldap user or not

Notes:

- Service is available just for “Admin” users. If an unauthorized user is trying to execute this service, it will return the following error message:
  “User ‘operator1’ is not allowed to retrieve all users”.

Example of a user entry:

<table>
<thead>
<tr>
<th>Username</th>
<th>operator1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role type</td>
<td>Operator</td>
</tr>
<tr>
<td>Categories</td>
<td>{CategoryA, CategoryB}</td>
</tr>
<tr>
<td>isLdap</td>
<td>false</td>
</tr>
</tbody>
</table>
## 4.11.24.2.2 Change Password

**Usage:**
```
changePassword(oldPassword, newPassword)
```

**Description:** Changes the password for the current user (the user that is executing the service).

**Input parameters:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old password</td>
<td>String</td>
<td>true</td>
<td>Current password</td>
</tr>
<tr>
<td>New password</td>
<td>String</td>
<td>true</td>
<td>New password</td>
</tr>
</tbody>
</table>

**Returned value:** none

**Notes:**

- Service is available for all user roles.
- Service is unavailable for LDAP users.
- If current password is empty or null, the service will return the following error message: *"Unable to change password because old password is empty"*.
- If new password is empty or null, the service will return the following error message: *"Unable to change password because new password is empty"*.
- If current password is incorrect, the service will return the following error message: *"Unable to change password because old password does not match existing password"*.
- If the current user is an LDAP user, the service will return the following error message: *"LDAP users are not allowed to change their passwords"*.

**Example for execution:**
```
$wfa.changePassword("oldpass1234", "newpass5678")
```
5 CREATING A WEB SERVICE PROXY WITH MICROSOFT POWERSHELL

Note: The following steps describe the process of creating a web service proxy for “WorkflowService_rpc” WSDL. Similar steps should be taken for creating a web service proxy for “UserService_rpc” WSDL.

1) Open Microsoft Powershell Shell by “Start -> Run”

2) Setup the password and credentials for meeting the security requirements

```powershell
$password = ConvertTo-SecureString "admin" -AsPlainText –Force
```

3) Create the web services proxy using the WFA installation

```powershell
```

4) List all workflows by using:

```powershell
$workflows = $wfa.getAllWorkflows()
and print it by typing:
$workflows
```
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 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 A – 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.
## APPENDIX B – ERROR MESSAGES

<table>
<thead>
<tr>
<th>#</th>
<th>Relevant methods</th>
<th>Error Message</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Workflow Service</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>GetAllWorkflowsByCategories</td>
<td>Category name does not exist</td>
<td>The provided category name does not exist.</td>
<td>Category name: examples does not exist</td>
</tr>
<tr>
<td>2</td>
<td>GetAllWorkflowsByCategories</td>
<td>Please provide at least one category name</td>
<td>In case no category is provided to this method</td>
<td>Please provide at least one category name</td>
</tr>
<tr>
<td>3</td>
<td>GetAllWorkflowsByCategories</td>
<td>Current user is not authorized for any of the given categories</td>
<td>The user is an operator and is not authorized for any of the categories given as parameter</td>
<td>Current user is not authorized for any of the given categories</td>
</tr>
<tr>
<td>4</td>
<td>runWorkflow</td>
<td>No workflow found for id:</td>
<td>The provided workflow id does not exist.</td>
<td>No workflow found for id: -1</td>
</tr>
<tr>
<td>5</td>
<td>runWorkflow</td>
<td>Current user is not allowed to execute workflow</td>
<td>The user is an operator and not assigned to the workflow category</td>
<td>Current user David is not allowed to execute workflow 6</td>
</tr>
<tr>
<td>6</td>
<td>runWorkflow</td>
<td>Current user is not allowed to execute workflow since it is not ready for production yet</td>
<td>The user is an operator and this workflow is not “ready for production”.</td>
<td>Current user David is not allowed to execute workflow 6 since it is not ready for production yet</td>
</tr>
<tr>
<td>7</td>
<td>runWorkflow</td>
<td>Got incorrect date and time format input</td>
<td>The date format is not valid</td>
<td>Got incorrect date and time format ‘234’. Correct format ‘M/d/yy h:mm a’, for example: 2/14/12 3:28 PM</td>
</tr>
<tr>
<td>8</td>
<td>runWorkflow</td>
<td>Cannot schedule a workflow in the past</td>
<td>The date for the workflow is in the past.</td>
<td>Cannot schedule a workflow in the past</td>
</tr>
<tr>
<td>9</td>
<td>runWorkflow, previewWorkflow</td>
<td>Invalid user-input and value, should be: name=value</td>
<td>The user input value was not provided in the right format</td>
<td>Invalid user-input and value volumeName, should be: name=value</td>
</tr>
<tr>
<td>10</td>
<td>runWorkflow, previewWorkflow</td>
<td>The values for have to fit</td>
<td>When a user input is defined as a locked query and the value does not match the query</td>
<td>The values for array_ip have to fit select array_ip from wfa.array</td>
</tr>
<tr>
<td></td>
<td>runWorkflow, previewWorkflow</td>
<td>The values for have to be within</td>
<td>When a user input is defined as a locked Enum and the value is not one of the its values</td>
<td>The values for size have to be within 30,40,50</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------</td>
<td>----------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>12</td>
<td>runWorkflow, previewWorkflow</td>
<td>The values for have to be between and</td>
<td>When a user input is defined as a Number and the value is not in the defined range</td>
<td>The values for age have to be between 10 and 50</td>
</tr>
<tr>
<td>13</td>
<td>runWorkflow, previewWorkflow</td>
<td>The values for must match the regular expression:</td>
<td>When a user input is defined as a String and the value does not match the regular expression</td>
<td>The values for ip must match the regular expression: [^]</td>
</tr>
<tr>
<td>14</td>
<td>runWorkflow, previewWorkflow</td>
<td>User input is not defined in workflow</td>
<td>User input name is not defined in workflow Create Vfiler</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>resumeWorkflow, rejectWorkflow</td>
<td>Workflow execution Id was not found</td>
<td>The provided job Id was not found</td>
<td>Workflow execution Id 12 was not found</td>
</tr>
<tr>
<td>16</td>
<td>resumeWorkflow</td>
<td>current user is not allowed to resume workflow</td>
<td>The user is an operator and not assigned to the workflow category</td>
<td>current user David is not allowed to resume workflow 6</td>
</tr>
<tr>
<td>17</td>
<td>resumeWorkflow</td>
<td>Could not resume workflow execution with id. Resume is only allowed from statuses: PAUSED CANCELED FAILED SCHEDULED</td>
<td>Resume is only possible for jobs that are in statuses: PAUSED CANCELED FAILED SCHEDULED</td>
<td>Could not resume workflow execution with id 213. Resume is only allowed from statuses: PAUSED CANCELED FAILED SCHEDULED</td>
</tr>
<tr>
<td>18</td>
<td>resumeWorkflow</td>
<td>Resume of workflow execution with id is not allowed. Workflow execution is expired - more than days have passed since it stopped running.</td>
<td>This job cannot be resumed since its reservation is already expired and the results are not predictable.</td>
<td>Resume of workflow execution with id 105 is not allowed. Workflow execution is expired - more than 2 days have passed since it stopped running.</td>
</tr>
<tr>
<td></td>
<td>Action</td>
<td>Message</td>
<td>User Access</td>
<td>Reason</td>
</tr>
<tr>
<td>---</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>19</td>
<td>resumeWorkflow</td>
<td>Users with role are not allowed to workflow executions</td>
<td>When an operator tries to resume a job which is in status paused and the configuration forbids it</td>
<td>Users with role Operator are not allowed approve to workflow executions</td>
</tr>
<tr>
<td>20</td>
<td>resumeWorkflow</td>
<td>Workflow execution with id cannot be resumed because it was canceled before it completed its planning</td>
<td>If the workflow was canceled before the planning was done then the workflow cannot be resumed</td>
<td>Workflow execution with id 213 cannot be resumed because it was canceled before it completed its planning</td>
</tr>
<tr>
<td>21</td>
<td>rejectWorkflow</td>
<td>current user is not allowed to reject workflow</td>
<td>The user is an operator and not assigned to the workflow category</td>
<td>current user David is not allowed to reject workflow 6</td>
</tr>
<tr>
<td>22</td>
<td>rejectWorkflow</td>
<td>Users with role are not allowed to workflow executions</td>
<td>When an operator tries to reject a job which is in status paused and the configuration forbids it</td>
<td>Users with role Operator are not allowed to reject workflow executions</td>
</tr>
<tr>
<td>23</td>
<td>getJobStatus</td>
<td>No job found for id:</td>
<td>The job id does not exist</td>
<td>No job found for id: 16</td>
</tr>
<tr>
<td>24</td>
<td>getReturnParameters</td>
<td>Failed to find workflow execution result by provided job id</td>
<td>The job id does not exist</td>
<td>Failed to find workflow execution result by provided job id 143</td>
</tr>
<tr>
<td>25</td>
<td>getReturnParameters</td>
<td>The job status is data can be retrieved only in the following statuses: COMPLETED, FAILED, CANCELED</td>
<td>Return parameters can be viewed only after the job finished running which is in: COMPLETED, FAILED, CANCELED</td>
<td>The job status is RUNNING, data can be retrieved only in the following statuses: COMPLETED, FAILED, CANCELED</td>
</tr>
<tr>
<td>26</td>
<td>getReturnParameters</td>
<td>Parameter was not yet calculated, please check execution result</td>
<td>One of the parameters was not yet calculated (should not happen)</td>
<td>Parameter volumeName was not yet calculated, please check execution result</td>
</tr>
<tr>
<td>Line</td>
<td>Method</td>
<td>Issue</td>
<td>Solution</td>
<td>User Service</td>
</tr>
<tr>
<td>------</td>
<td>--------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>27</td>
<td>getReturnParameters</td>
<td>Parameter not found. It's not defined as a return parameter</td>
<td>The given return parameter was not defined for this workflow</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Parameter volumeName not found. It's not defined as a return parameter</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>previewWorkflow</td>
<td>Cannot preview workflow, workflow with id: does not exist</td>
<td>The given workflow id does not exist</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cannot preview workflow, workflow with id: 13 does not exist</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>waitForCompletion</td>
<td>No job found for id:</td>
<td>The job id does not exist</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No job found for id: 16</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>waitForCompletion</td>
<td>Cannot wait on job with status:</td>
<td>The job is currently not in a status that allows waiting for it to complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Cannot wait on job with status: COMPLETED</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>waitForCompletion</td>
<td>Cannot wait for executions that are not scheduled to run in the next 24 hours</td>
<td>The job is scheduled to run in the near future (24 hours)</td>
<td>Executing waitForCompletion with a job id that is scheduled to run in a week from today, will return the following error: ‘Cannot wait for executions that are not scheduled to run in the next 24 hours’</td>
</tr>
<tr>
<td>1</td>
<td>getAllUsers</td>
<td>Current user is not allowed to retrieve all users</td>
<td>The current user is not of role ‘Admin’ and therefore not allowed to retrieve all users</td>
<td>User ‘operator’ is not allowed to retrieve all users</td>
</tr>
<tr>
<td>2</td>
<td>changePassword</td>
<td>Unable to change password because old password is empty</td>
<td>Old password given as parameter is either null or an empty string</td>
<td>Unable to change password because old password is empty</td>
</tr>
<tr>
<td>3</td>
<td>changePassword</td>
<td>Unable to change password because new password is empty</td>
<td>New password given as parameter is either null or an empty string</td>
<td>Unable to change password because new password is empty</td>
</tr>
<tr>
<td>4</td>
<td>changePassword</td>
<td>Unable to change password because old password does not match existing password</td>
<td>Old password given as parameter does not match the password in the database</td>
<td>Unable to change password because old password does not match existing password</td>
</tr>
<tr>
<td>5</td>
<td>changePassword</td>
<td>LDAP users are not</td>
<td>The current user</td>
<td>LDAP users are not allowed to</td>
</tr>
<tr>
<td></td>
<td>allowed to change their passwords</td>
<td>that is trying to change his password is an LDAP user</td>
<td>change their passwords</td>
<td></td>
</tr>
</tbody>
</table>