



Technical Report

IT as a Service: Storage Automation Using NetApp OnCommand Workflow Automation 2.0 and HP Operations Orchestration 9.05

Dan Herington, Rick Ehrhart, NetApp
March 2013 | TR-4103

Abstract

NetApp® OnCommand® Workflow Automation (WFA) provides a mechanism for standardizing IT administration activities related to NetApp shared storage and publishing them as Web Services that are consumable by enterprise orchestration tools. This report describes a toolkit that can be imported into HP Operations Orchestration (HP OO) to enable users of HP OO to execute WFA workflows through the WFA Web Services API. This toolkit is available on the NetApp OnCommand Workflow Automation Community site:

https://communities.netapp.com/community/products_and_solutions/storage_management_software/workflow-automation.

TABLE OF CONTENTS

1	Executive Summary.....	4
1.1	Introduction	4
1.2	IT as a Service	4
1.3	The IT Automation Challenge	4
2	Software Components and Integration	5
2.1	Roles and Functionality.....	6
2.2	HP Operations Orchestration	6
2.3	NetApp OnCommand Workflow Automation	7
2.4	NetApp OnCommand Unified Manager	7
3	The HP OO Toolkit for NetApp OnCommand WFA	7
3.1	Importing and Configuring the Toolkit in HP OO	7
3.2	HP OO Subflows for WFA Web Service Calls.....	10
3.3	Utility Subflows.....	11
3.4	Example End-to-End OO Flows	12
4	Creating an OO Flow to Run Your WFA Workflow	13
4.1	Anatomy of an OO Flow for WFA.....	13
4.2	Customizing the Examples for Your WFA Workflow	14
4.3	Capturing Output Parameters	19
5	Conclusion	20
	References.....	21
	Acknowledgements	21

LIST OF FIGURES

Figure 1)	Keeping automation logic close to the infrastructure element.	5
Figure 2)	Software component and role interaction.	6
Figure 3)	Importing the toolkit repository into HP OO Studio.	8
Figure 4)	Toolkit configuration items.	9
Figure 5)	Updating NetAppWFAUser account credentials.....	9
Figure 6)	Updating NetAppWFAHost and NetAppWFAPort.	10
Figure 7)	Toolkit Web Services subflows.....	11
Figure 8)	Toolkit utility subflows.	12
Figure 9)	Toolkit example subflows.	12
Figure 10)	CreateQtreeAndNFSExport OO flow.....	13

Figure 11) CloneEnvironment example OO flow.	14
Figure 12) WFA workflow return parameters dialog box.	15
Figure 13) WFA workflow user inputs dialog box.	16
Figure 14) OO flow input parameters.	17
Figure 15) ProvisionOracleStorage OO flow inputs.	17
Figure 16) Update the workflowName input to the GetWFAWorkflowID step.	18
Figure 17) Modify the inputs to RunWFAWorkflow.	19
Figure 18) GetExport XPath statement.	20
Figure 19) GetExport subflow results.	20

1 Executive Summary

1.1 Introduction

NetApp OnCommand Workflow Automation (WFA) is a powerful framework for automating storage management tasks that enables agility, flexibility, efficiency, and repeatability of best practices in a NetApp shared storage infrastructure. WFA comes with predefined, supported base building blocks, including the ability to customize and extend the system to suit specific needs. It also features a Web Service API to allow WFA workflows to be triggered from virtually any enterprise orchestration application.

Hewlett-Packard Operations Orchestration (HP OO) software integrates enterprise management systems and automates standard operations as well as critical Information Technology Infrastructure Library (ITIL) processes. It provides out-of-the-box workflows for data centers and can also be extended and customized to suit particular deployment requirements.

This document describes a toolkit that can be imported into HP OO 9.05 that provides OO subflows to execute the WFA Web Services that are used to execute and manage the WFA storage automation workflows. This document contains:

- An overview of how HP OO and OnCommand WFA can be integrated
- An overview of the WFA Web Services API and the HP OO operations available in the integration toolkit
- A description of how to modify the example HP OO flows in the toolkit to execute any other WFA workflow

1.2 IT as a Service

Many IT organizations are being asked to explore and/or implement cloud computing technologies. Although the primary benefit of cloud computing was originally believed to be cost savings, the reality is that over time the most compelling advantage is a significant reduction in the time required to implement new applications and solutions. One of the primary reasons for this reduction is that much of the infrastructure lifecycle of infrastructure is automated, resulting in:

- Reduced implementation time
- Fewer errors compared to implementing custom solutions for individual applications
- Lower maintenance costs because the automation implements standard configurations that are easier to troubleshoot

1.3 The IT Automation Challenge

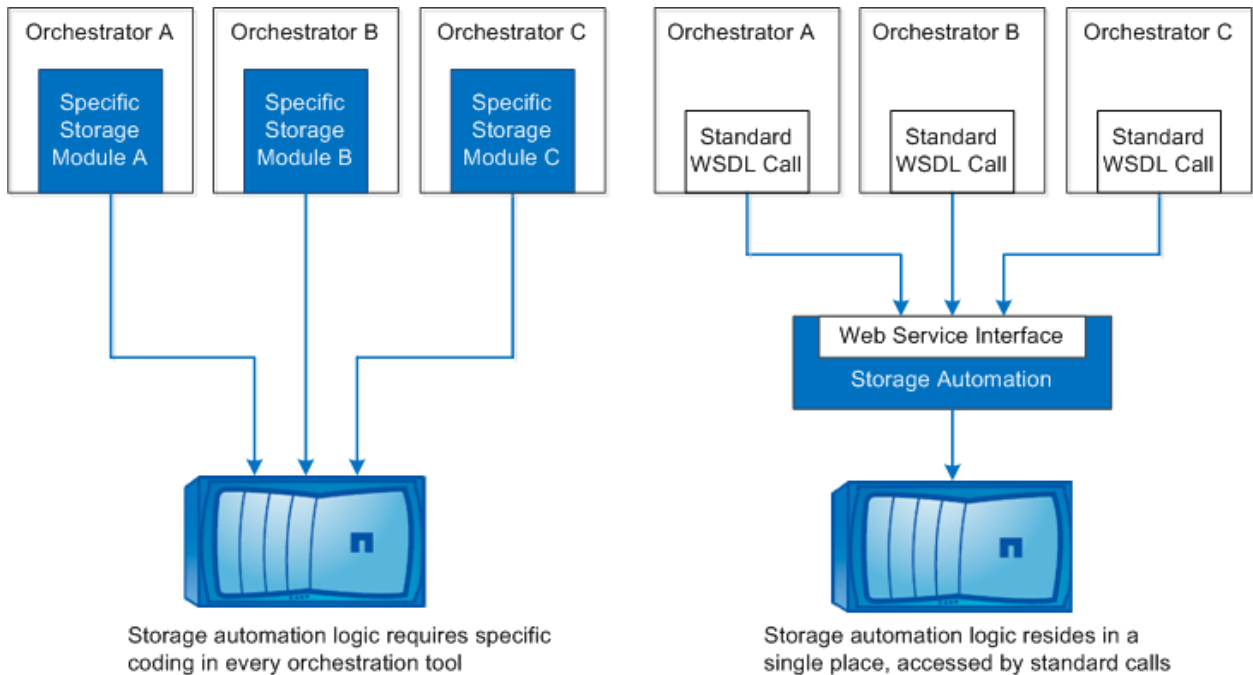
To realize the benefits of automation, it is necessary to design, implement, and test the automation itself. Clearly this would not be cost effective unless the process being automated is expected to be run several times. Therefore it is necessary to build a flexible infrastructure that is based on standard building blocks.

Another challenge to automation is that it typically requires a deep level of understanding of the underlying technology components as well as of the application programming interfaces to those components.

Although many enterprise automation and orchestration tools are available in the market, it is often challenging to integrate and maintain a single enterprise management tool that can directly manage all aspects of the IT infrastructure. It is therefore desirable to modularize the automation of individual infrastructure elements (such as servers, storage, and so on) and to have the enterprise tool provide overall coordination according to well-defined interfaces. This means that for any change in the enterprise orchestration tool or other aspect of the IT infrastructure, there is no need to recreate those administrative workflows to minimize the effect on the automation of any specific element.

Figure 1 illustrates an example of the complexity of implementing the automation of storage from multiple orchestration products. By using an automation tool that is designed to handle the underlying hardware platform, the complexity of the automation in the orchestration tool is significantly reduced.

Figure 1) Keeping automation logic close to the infrastructure element.



The OnCommand Workflow Automation tool was implemented to provide a deep understanding of the underlying architecture of the storage system and the best practices for deploying storage. Because it also provides a standards-based Web Services interface, WFA allows the higher-level orchestration products to leverage that inherent knowledge with minimum added complexity.

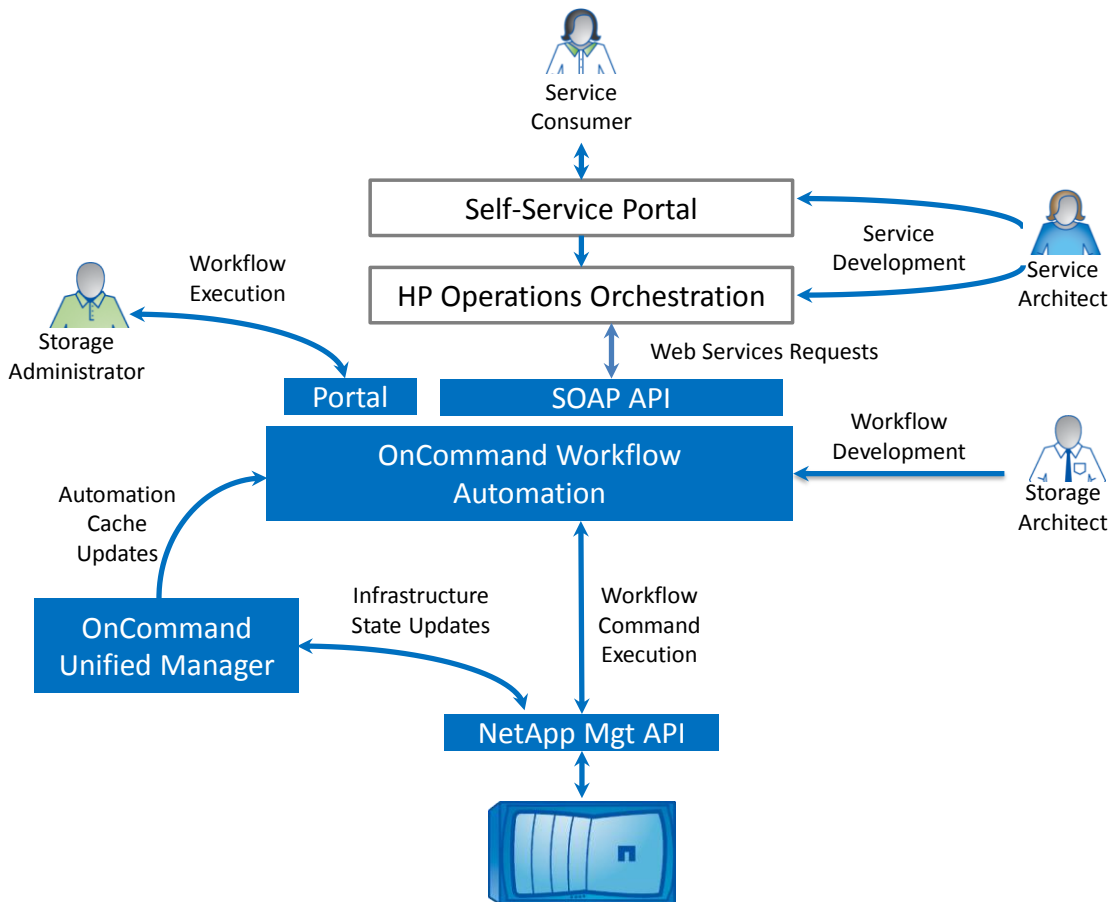
2 Software Components and Integration

This section describes the interaction of the individual software components used to implement IT service automation that uses HP OO while leveraging WFA for storage automation. The four main components are:

- Self-service portal
- HP Operations Orchestration (HP OO)
- OnCommand Workflow Automation (WFA)
- NetApp OnCommand Unified Manager (OCUM)

Figure 2 shows the basic interactions between these components and the roles that interact with them.

Figure 2) Software component and role interaction.



2.1 Roles and Functionality

There are four main roles in this environment:

- **Service consumer.** The ultimate consumer of services offered by the self-service portal.
- **Service architect.** Designs the services offered in the self-service portal and defines the HP OO service flows, combining resources such as those offered by OnCommand Workflow Automation and others provided by compute systems, applications, and so on. The service architect develops HP OO flows to interact with defined WFA storage automation workflows.
- **Storage architect.** Implements storage-management policies by creating the storage workflows in WFA, which are in turn offered as resources for the creation of services in the self-service portal and HP OO.
- **Storage administrator.** Manages the storage infrastructure and can directly access the WFA workflows to perform storage administration tasks according to defined policies. These workflows may or may not be accessible to HP OO as storage resource operations.

2.2 HP Operations Orchestration

The focus of HP OO is to automate IT processes and reduce manual activity. It orchestrates across the functions of multiple technology management frameworks to accommodate the execution of the end-to-end workflow. Adapters provide an interface between HP OO workflows, IT components, and management frameworks. An HP OO workflow might leverage workflows of different automation frameworks.

HP OO is accessed, possibly indirectly, by a service consumer when the consumer requests, for example, a predefined service. Service templates can be created by a service architect based on a packaged combination of available resources from storage, compute, and application elements for easy consumption by service consumers.

2.3 NetApp OnCommand Workflow Automation

OnCommand Workflow Automation is a software solution that offers “one-click” automation for key storage processes. You can automate storage management tasks such as storage provisioning, migration, decommissioning, and cloning. You can use WFA to build workflows that do exactly what your processes require. It provides a robust workflow design interface and a rich set of predefined content to simplify the workflow development process. There is also a significant amount of content available for download from the [NetApp OnCommand WFA Community](#) site, which can be customized to meet your requirements.

Workflows are made available through a Web Services interface for use by HP OO by using Simple Object Access Protocol (SOAP) requests. The toolkit described in this document provides HP OO subflows for all of the WFA SOAP requests as well as some examples of how to use the subflows to execute example WFA workflows.

WFA relies on OnCommand Unified Manager (OCUM) to maintain status information about the storage infrastructure—available storage controllers, the presence and state of aggregates, volumes, and so on—and the allocation of storage into resource pools. WFA collects this data from OCUM and stores it in a local cache on a regular basis. WFA can also collect and make use of data from VMware® vCenter™ (not shown in Figure 2).

2.4 NetApp OnCommand Unified Manager

NetApp OnCommand Unified Manager is the primary monitoring and reporting tool for NetApp storage systems. It provides both a human interface and an application programming interface (API) for integration with other management software. The API uses well-defined XML objects for communication, which provides transparency and interoperability.

Data collected by OCUM is queried by WFA to provide a subset of relevant information for use with automation workflows. Multiple OCUM systems can feed into a single WFA installation. The storage administrator can directly access OCUM to create storage management reports and perform monitoring activities.

3 The HP OO Toolkit for NetApp OnCommand WFA

This section is an overview of the HP OO toolkit for OnCommand Workflow Automation. This toolkit is available for download from the [NetApp OnCommand WFA Community](#) site and can be imported into HP OO to allow Operations Orchestration flow developers to leverage WFA workflows in their automation flows.

The HP OO toolkit for OnCommand WFA contains subflows that execute each of the Web Services calls available for WFA, in addition to some utility and example OO flows. This section describes how to download and import the toolkit into HP OO. It also describes the contents of the toolkit.

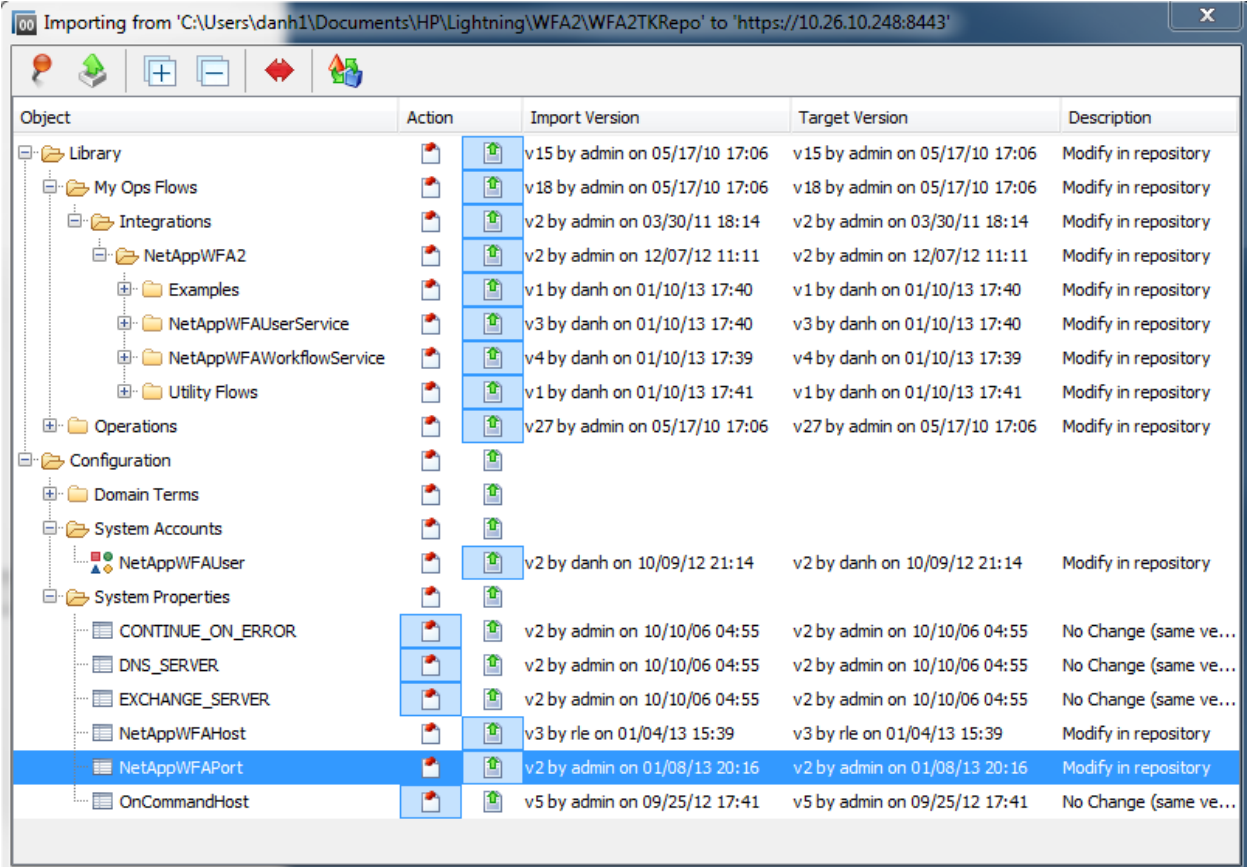
3.1 Importing and Configuring the Toolkit in HP OO

To import and configure the toolkit, complete the following steps.

1. Download the HP OO toolkit for OnCommand WFA from https://communities.netapp.com/community/products_and_solutions/storage_management_software/workflow-automation. Access to the NetApp OnCommand WFA Community site requires registration.

2. In the NetApp OnCommand WFA Community site, click the Content tab and then filter the results for HP OO. You should download the toolkit to the system on which you run HP OO Studio.
Note: The HP OO Toolkit for OnCommand WFA was implemented using HP OO 9.05 and OnCommand Workflow Automation 2.0.
3. After downloading the toolkit to your system, extract the files from the compressed zip file.
4. Open HP OO Studio and select Repository > Import Repository.
5. Locate the directory where you extracted the zip file and click Open.
HPOO Studio examines the content of the toolkit repository and compares it to your active repository. Any new or updated content is highlighted for import, as shown by the example output in Figure 3.

Figure 3) Importing the toolkit repository into HP OO Studio.



Object	Action	Import Version	Target Version	Description
Library				
My Ops Flows				
Integrations				
NetAppWFA2				
Examples				
NetAppWFAUserService				
NetAppWFAWorkflowService				
Utility Flows				
Operations				
Configuration				
Domain Terms				
System Accounts				
NetAppWFAUser				
System Properties				
CONTINUE_ON_ERROR				
DNS_SERVER				
EXCHANGE_SERVER				
NetAppWFAHost				
NetAppWFAPort				
OnCommandHost				


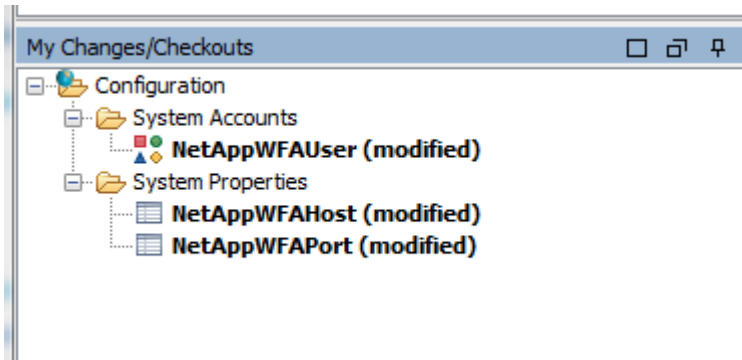
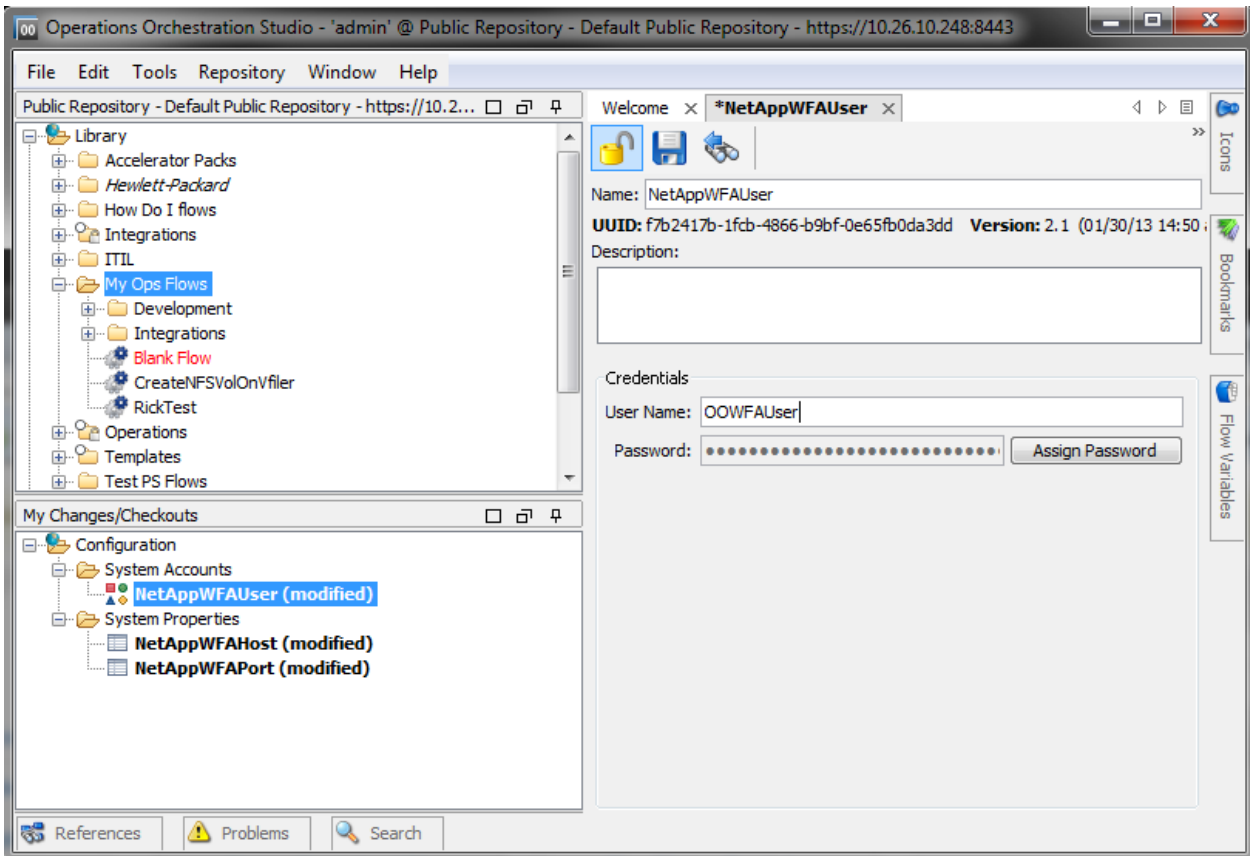
6. Click the Apply button  to import the content.
The changes appear in the My Changes/Checkouts area. By default, the toolkit places its content under the My Ops Flows/Integrations/NetAppWFA2 folder. You can change this location before checking in the content. Place the content in the required location and then check in the content of the Library tree.
7. There are three items under the Configuration folder that you must modify before you check in the content. These configuration items, shown in Figure 4, are used by all of the other subflows in the toolkit.

Figure 4) Toolkit configuration items.



8. Double-click the `NetAppWFAUser` system account, as shown in Figure 5.

Figure 5) Updating `NetAppWFAUser` account credentials.



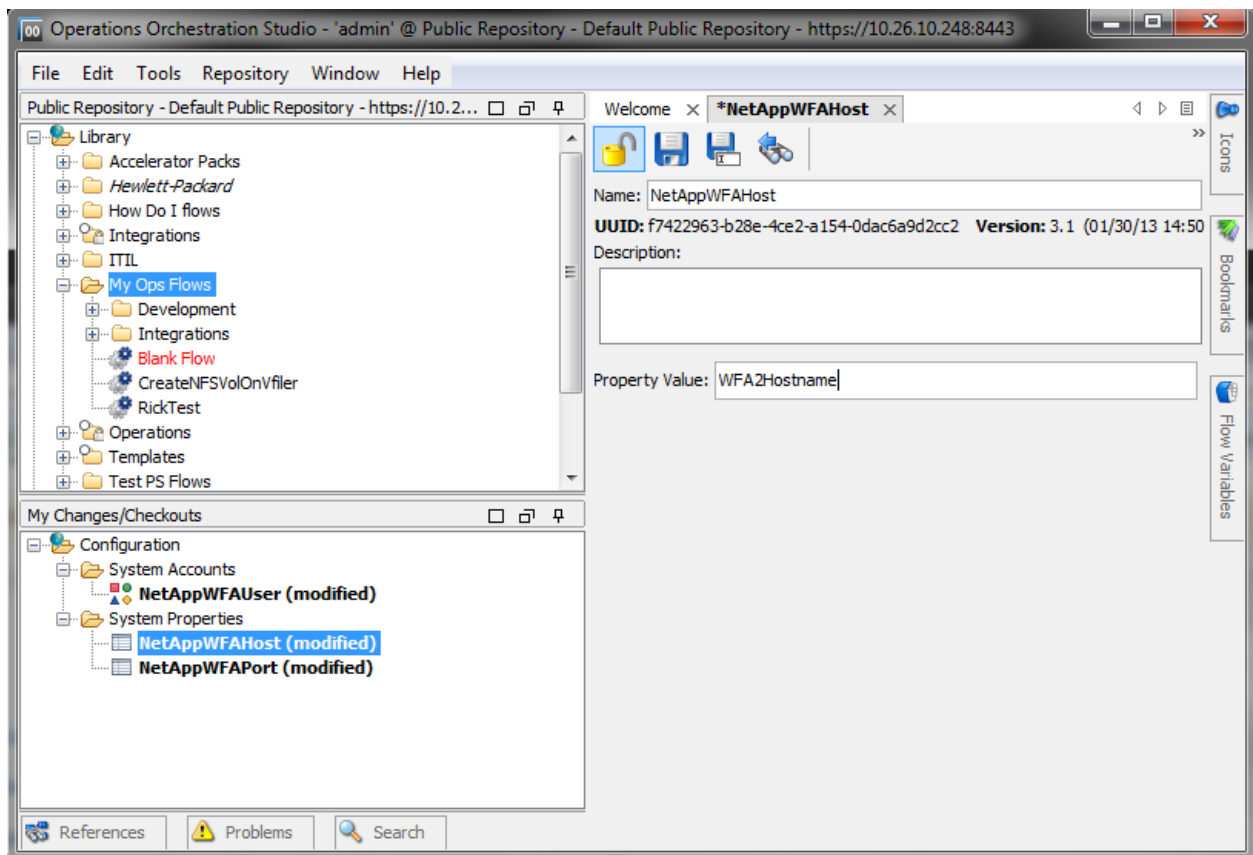
9. Enter the login credentials for the WFA user that will be used by HP OO to execute WFA workflows through the Web Service interface.

Note: For security and auditing reasons, you may want to create a new WFA user specifically for this purpose.

10. Select and update the values for the `NetAppWFAHost` and `NetAppWFAPort` system properties, as shown in Figure 6.

The `NetAppWFAHost` is the hostname or IP address of the host that is running WFA. The default port for WFA is 80. Use the default port, unless you modified the port when you installed WFA.

Figure 6) Updating `NetAppWFAHost` and `NetAppWFAPort`.



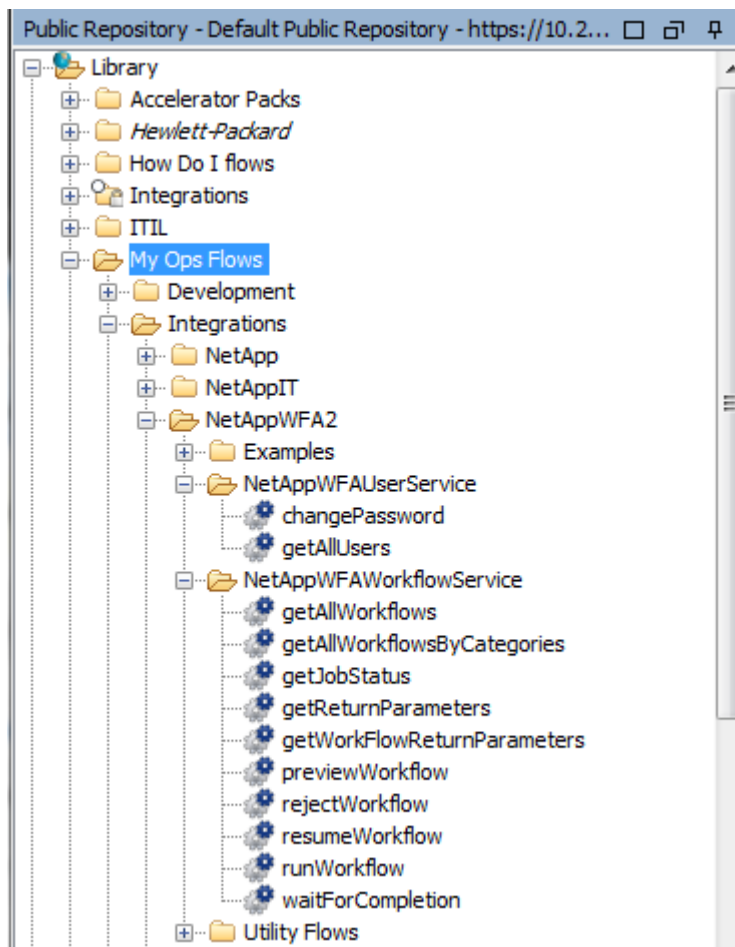
11. After making these configuration changes, you can check in the `Configuration` tree. The toolkit is now ready for use.

3.2 HP OO Subflows for WFA Web Service Calls

This section examines the contents of the toolkit.

There are two Web Services interfaces for WFA — a User Service and a Workflow Service. As shown in Figure 7, the toolkit contains HP OO subflows to execute all supported methods in these two interfaces.

Figure 7) Toolkit Web Services subflows.



The WFA includes a rudimentary portal interface where users can execute workflows for which they have been given access. The user service provides functionality that is helpful for managing those users.

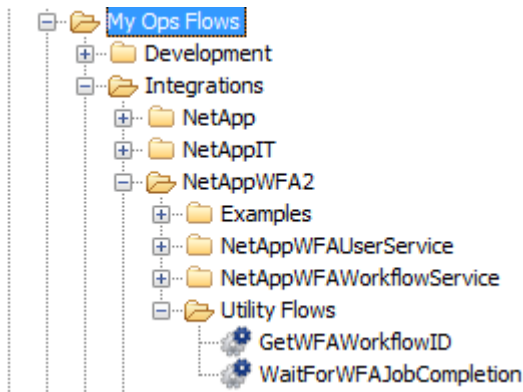
The Workflow Service section provides access to the methods required to run and manage the execution of WFA workflows.

For details about the functionality of each of these Web Services calls, see the [WFA Web Services primer](#).

3.3 Utility Subflows

To simplify the development of HP OO flows based on the underlying WFA Web Services calls, two utility subflows (`GetWFAWorkflowID` and `WaitForWFAJobCompletion`) were developed as part of the toolkit (see Figure 8).

Figure 8) Toolkit utility subflows.



To run workflows from the Web Services interface, you must provide the ID of the workflow. The `GetWFAWorkflowID` subflow takes the name of the workflow as it appears in the WFA portal interface and returns the ID of that workflow. The example flows in section 4 demonstrate how this is used.

By default, WFA workflows are run asynchronously. This is because they can be scheduled for future execution, and it is also possible to insert approval points into the WFA workflows.

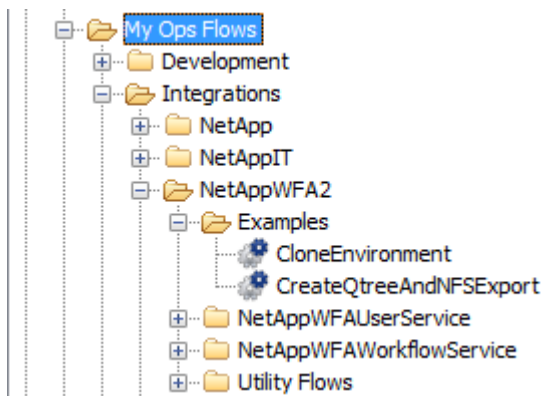
`WaitForWFAJobCompletion` allows you to create an end-to-end OO flow that runs the WFA workflow and then waits for it to complete without requiring looping constructs in the OO flow.

3.4 Example End-to-End OO Flows

The toolkit also contains two example OO flows that run two of the workflows provided with WFA as examples (shown in

Figure 9).

Figure 9) Toolkit example subflows.



The primary difference between these two OO flows is that `CreateQtreeAndNFSEExport` runs a WFA workflow that has a return parameter. Therefore that OO flow includes an extra step to obtain the return parameter and set it as an output of the flow so that it can be used by subsequent steps in the calling OO flow.

4 Creating an OO Flow to Run Your WFA Workflow

This section first describes the steps required to run a WFA workflow from the Web Services interface and then describes how to copy and customize the example OO flows provided in the toolkit to run your WFA workflows.

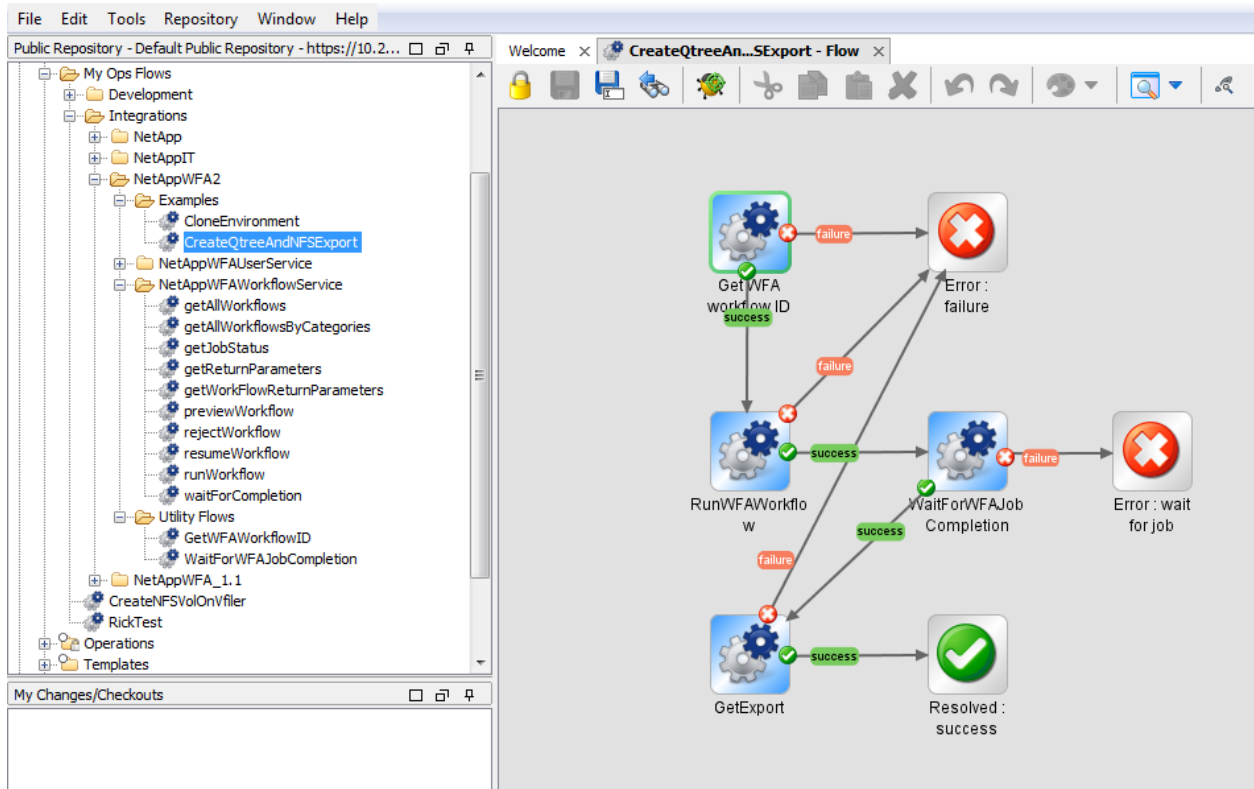
4.1 Anatomy of an OO Flow for WFA

The process required to run WFA workflows by using the Web Services interface involves four phases:

1. **Obtaining the ID of the desired workflow.** In this phase the existing WFA workflows are discovered from HP OO by using the WFA `getAllWorkflows` subflow. A list of all defined WFA workflows is returned, along with a short description and parameter definitions for each. The output is then reviewed to find the ID of the workflow that is to be run.
2. **Execution.** A selected WFA workflow is parameterized and executed by using the `runWorkflow` WFA Web Services method. The execution can also be scheduled for a later time. In either case, a job ID is returned, which should be used as a reference to retrieve further execution information.
3. **Tracking status.** The execution status of a WFA workflow can be retrieved by using the `getJobStatus` Web Services call along with the job ID. Wait until the job completes before moving to the next phase.
4. **Extracting return parameters.** If the WFA workflow has return parameters, those parameters are extracted and assigned to the OO flow results so that they can be used by other HP OO operations.

Figure 10 illustrates an example HP OO flow called `CreateQtreeAndNFSExport`.

Figure 10) CreateQtreeAndNFSExport OO flow.

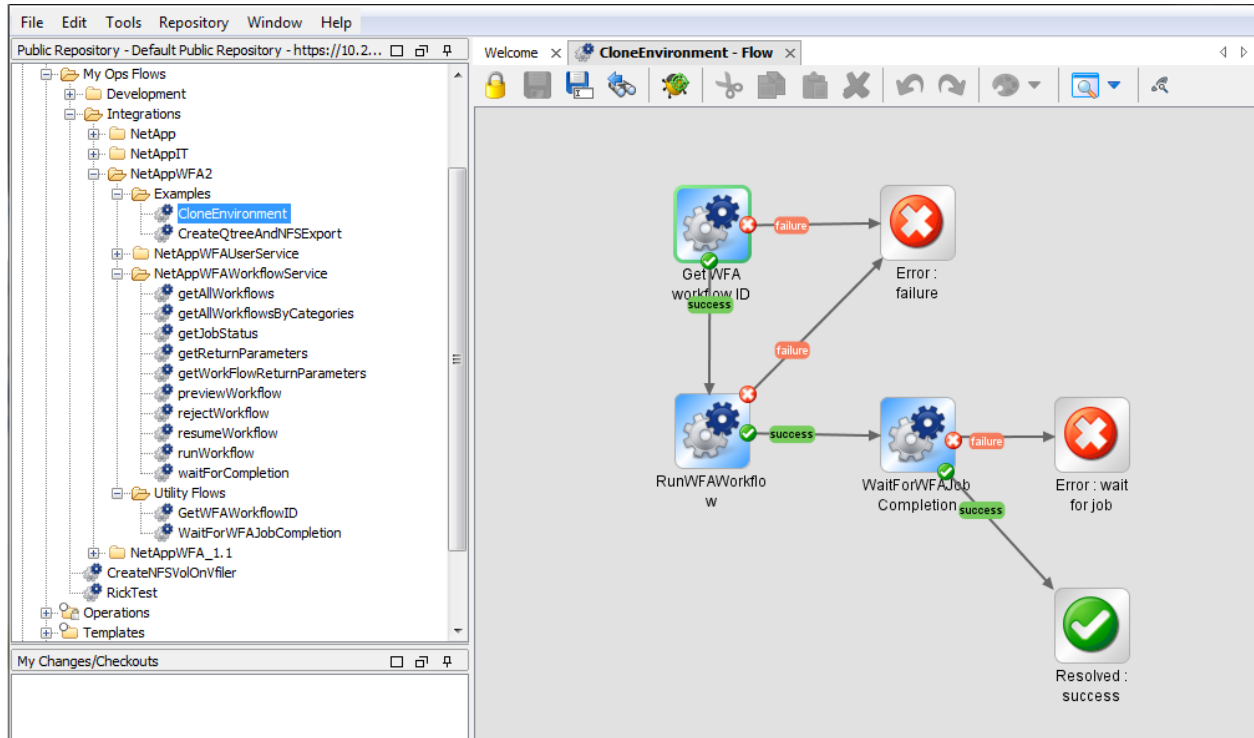


In this case, the utility flow `GetWFAWorkflowID` is used to find the ID of the workflow to execute. The workflow is run and the `WaitForWFAJobCompletion` subflow is used to wait for the job to complete.

After the job is completed, the `getReturnParameters` subflow is used to extract the export path, which is the return parameter from this particular WFA workflow.

The other example OO flow, `CloneEnvironment`, executes a similarly named example WFA workflow. The key difference in this flow is that the `CloneEnvironment` WFA flow does not have any return parameters, so the call to `getReturnParameters` is not required. This OO flow is shown in Figure 11.

Figure 11) `CloneEnvironment` example OO flow.



4.2 Customizing the Examples for Your WFA Workflow

This section shows how to customize the example flows to run any arbitrary WFA workflow by following these steps:

1. Select an example OO flow to start with.
2. Copy the chosen flow, open it, and rename it.
3. Update the input parameters to the OO flow to match the WFA workflow.
4. Update the workflow name in the call to `GetWFAWorkflowID`.
5. Update the input parameters to `RunWFAWorkflow` to match the WFA workflow inputs.
6. If there are return parameters, update them in the call to `getReturnParameters` (named `GetExport` in the example).

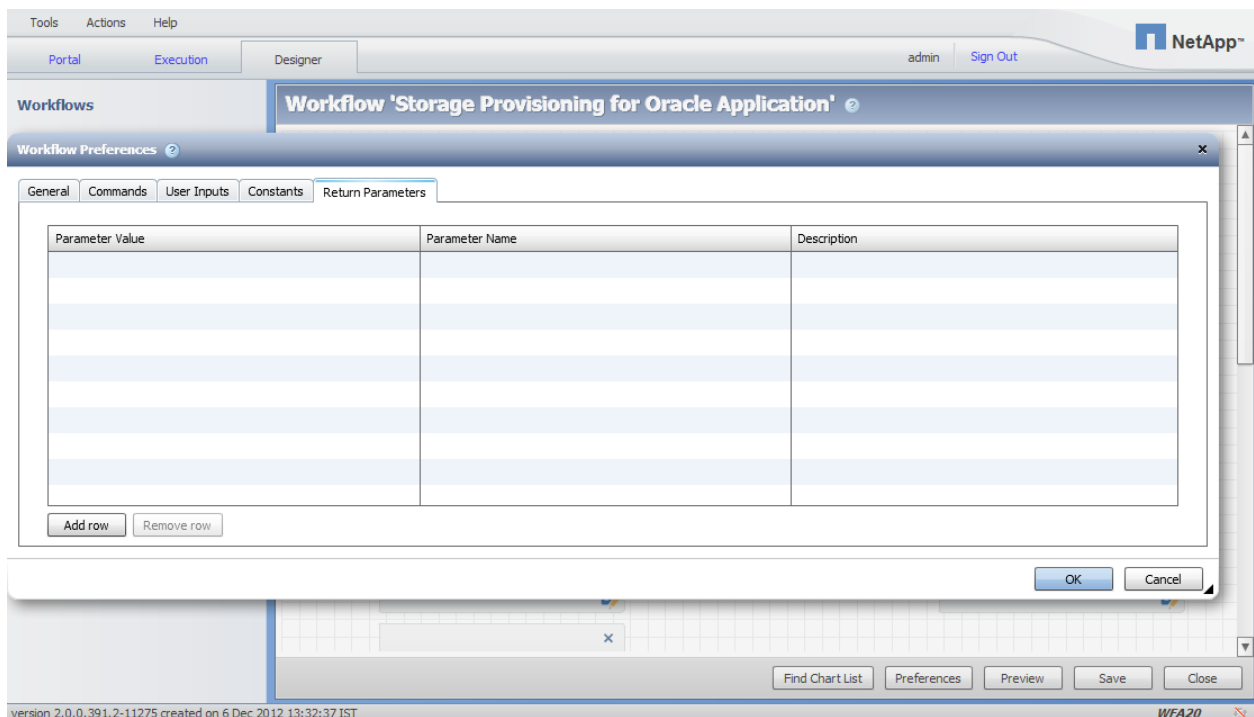
The rest of this section shows how to follow this process to create an OO flow to run another example WFA workflow.

Select an Example Flow

The first step is to determine which example flow to start with, based on whether the WFA workflow has output parameters. This section uses the example WFA workflow “Storage Provisioning for Oracle Application.”

1. Select this workflow in the WFA GUI, click the Preferences button, and click the Return Parameters tab (see Figure 12).

Figure 12) WFA workflow return parameters dialog box.



This workflow has no return parameters, so the `CloneEnvironment` example OO flow is used to start because it also has no return parameters.

2. In OO Studio, right-click the `CloneEnvironment` flow and select `Edit > Duplicate`.
3. Open the copy that was created and click the `Properties` tab.
4. Change the name to something appropriate; in this case, `ProvisionOracleStorage`.

Update OO Flow Input Parameters

While the `Properties` tab is open, you also need to modify the input parameters of this flow to match the input parameters of the WFA workflow that you are going to call.

1. Back in the WFA, click the `User Inputs` tab in the `Workflow Preferences` dialog box, as shown in Figure 13.

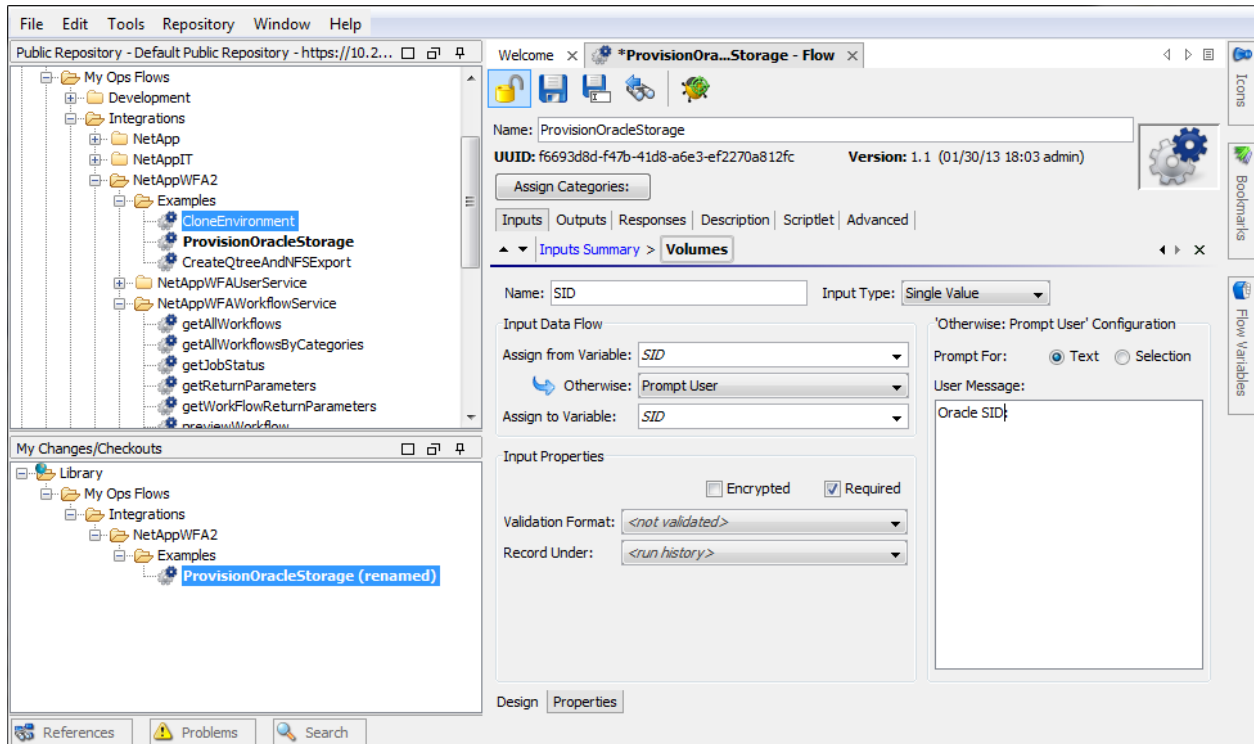
Figure 13) WFA workflow user inputs dialog box.

Name	Display Name	Description	Type	Values	Lock Values	Default Value	Group	Expanded by def	Mandatory
SID	Oracle SID	System ID of t...	String		<input type="checkbox"/>	OraDB123	Database Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ENV	Environment	Type of the en...	Enum	dev,tst,stg,prd	<input type="checkbox"/>	dev	Database Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Hostname	Physical DB Ho...	Hostname for ...	String		<input type="checkbox"/>	oradbhost1	Database Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TotalDataSize	Total Data Size...	Total usable si...	Number	0.5-1000	<input type="checkbox"/>	0.5	Size Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TotalRedoSize	Total Redo Siz...	Total usable si...	Number	0.5-1000	<input type="checkbox"/>	0.5	Size Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TotalArchSize	Total Arch Size...	Total usable si...	Number	0.5-1000	<input type="checkbox"/>	0.5	Size Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TotalExportSize	Total Export Si...	Total usable si...	Number	0.5-1000	<input type="checkbox"/>	0.5	Size Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TotalControlSize	Total Control Si...	Total control si...	Number	0.5-1000	<input type="checkbox"/>	0.5	Size Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
RootHosts	Root Hosts	Comma separa...	String		<input type="checkbox"/>	'host1-lab,host...	Export Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
RWHosts	Read-Write Hosts	Comma separa...	String		<input type="checkbox"/>	'host1-lab,host...	Export Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

2. Modify the input parameters in OO to match this list.

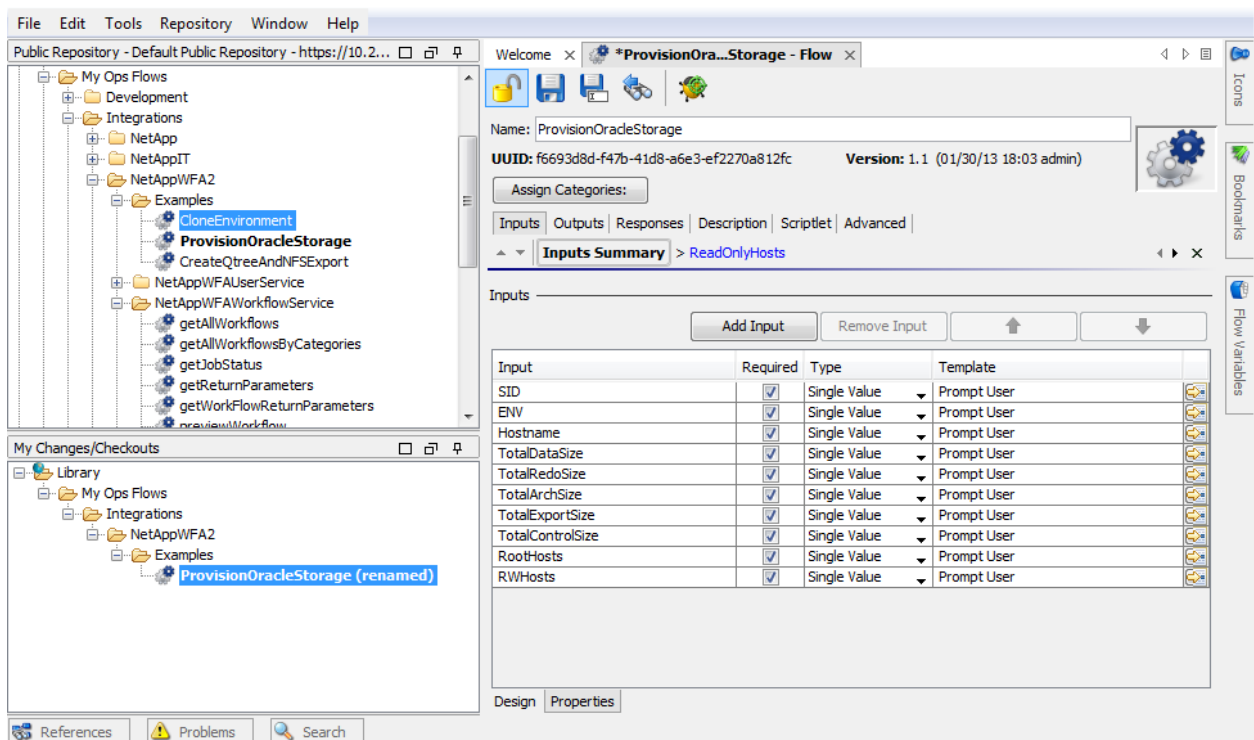
Back in OO Studio, we changed all the input variables to match the names in WFA. Although we plan to call this flow from a self-service portal, we set the Otherwise to Prompt User and the User Message to the WFA “Display Name” to make it easier to test the OO flow in the debugger. Figure 14 shows the settings for the first input parameter. Repeat this process for the remaining inputs. Note that all inputs are required.

Figure 14) OO flow input parameters.



Upon completion, the inputs screen appears as shown in Figure 15.

Figure 15) ProvisionOracleStorage OO flow inputs.



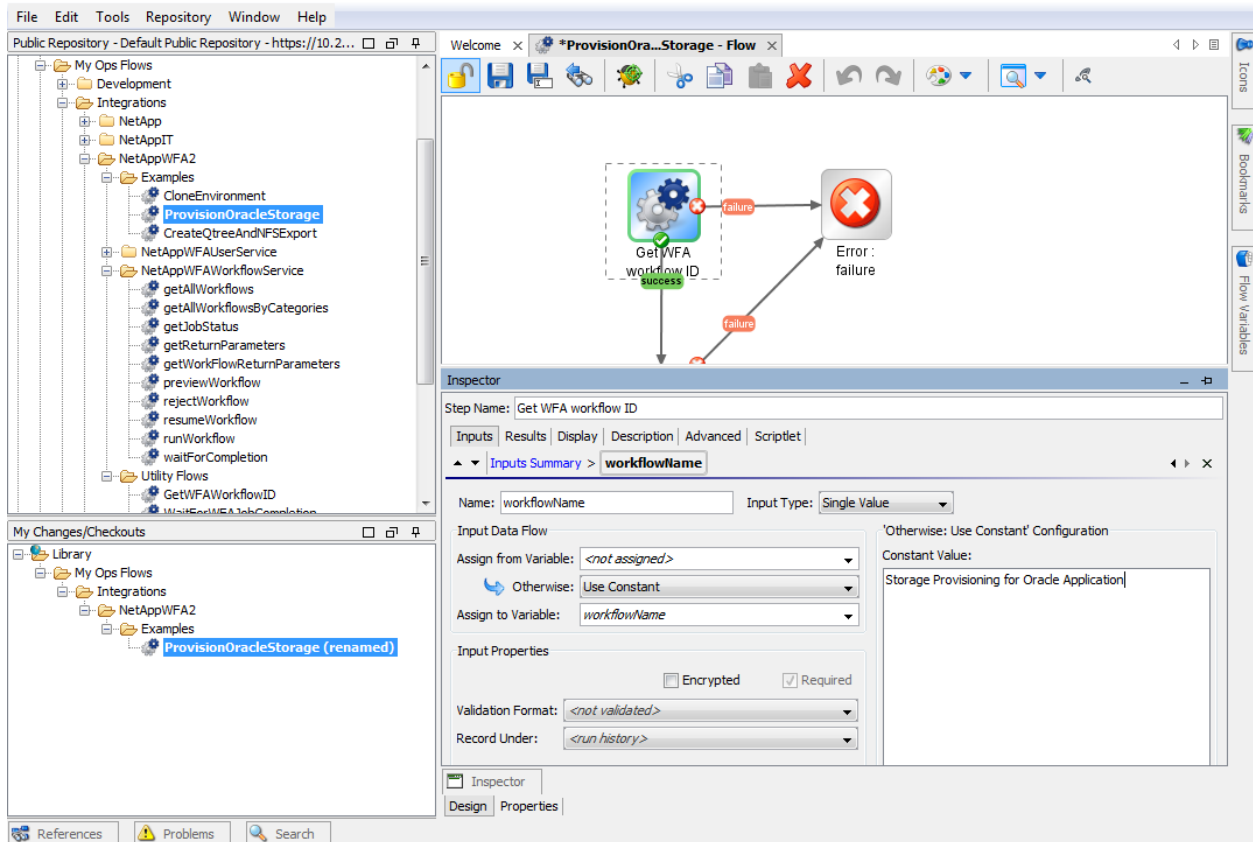
This WFA workflow does not have any return parameters, but if it did, you should click the Outputs tab and configure the OO flow outputs to match the WFA return parameters.

Update OO Flow Steps

The final task is to update the steps in the OO flow:

1. Click the Design tab at the bottom of the OO Studio to view the flow steps.
2. Double-click the `GetWFAworkflowID` step.
There is only one input to this step, which is the name of the WFA workflow.
3. Change the Constant Value entry from Clone Environment to Storage Provisioning for Oracle Application to match the name of the WFA workflow (see Figure 16).

Figure 16) Update the workflowName input to the GetWFAWorkflowID step.

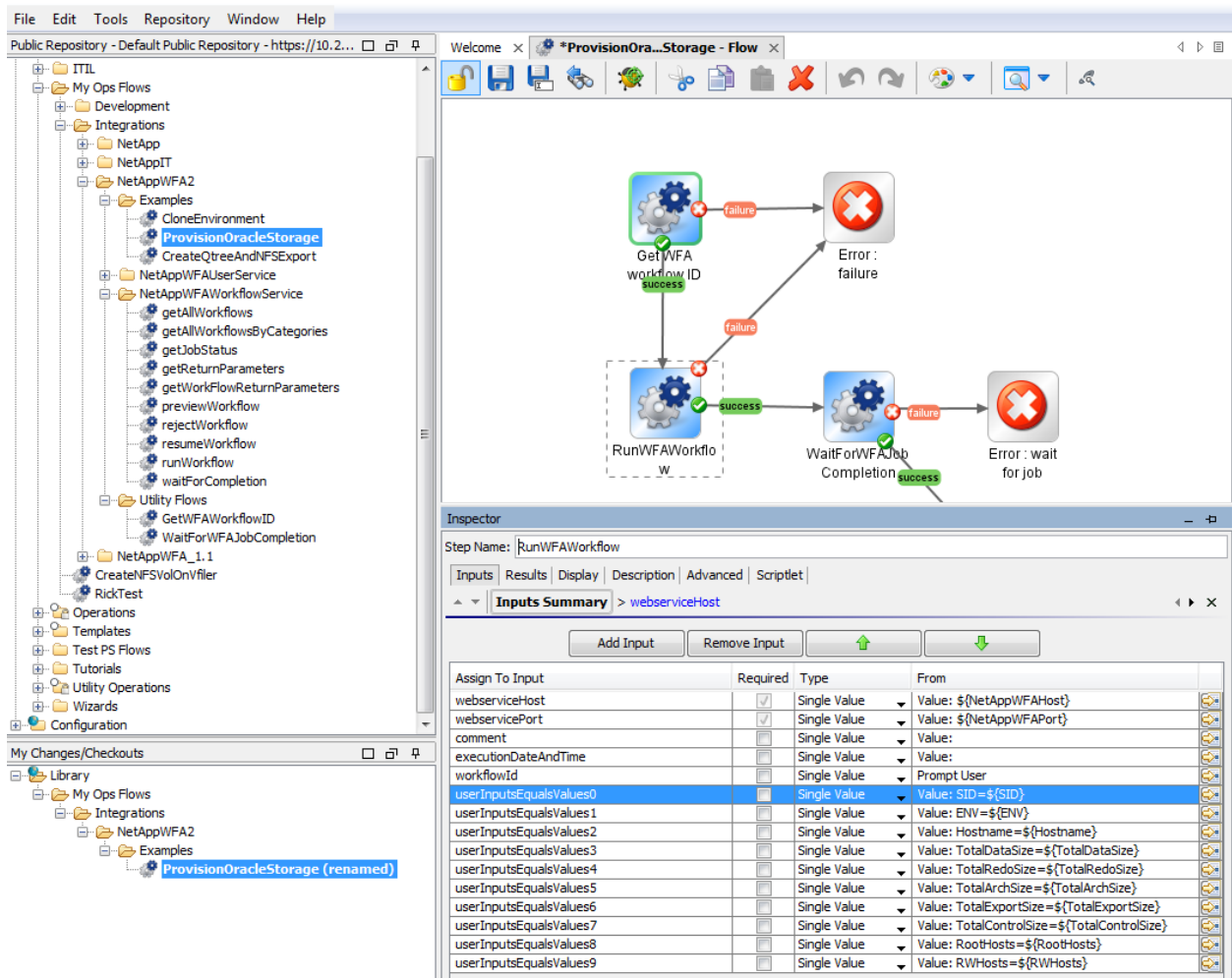


4. Double-click the `RunWFAWorkflow` step. The WFA `runWorkflow` Web Services call takes the workflow inputs as an array of name=value parameters. The HP OO operation used here converts all the `userInputsEqualsValues*` parameters into elements of the XML array.

Note: The `runWorkflow` OO subflow provided in the toolkit comes with 10 predefined parameters. If your workflow takes fewer than 10 parameters, you can leave them blank and the OO operation ignores them. If you need more than 10, you can add more by calling them `userInputsEqualsValues10` and up.

5. Update the inputs by setting them to the names of the `WFAInputs=${name of the OO flow input}`. Since the OO flow inputs were named the same as the WFA inputs, they all appear as `SID=${SID}`. The result of changing all the step inputs is shown in Figure 17.

Figure 17) Modify the inputs to RunWFAWorkflow.



4.3 Capturing Output Parameters

The example used here does not include the return parameters. This section describes how to handle return parameters for WFA workflows that include them.

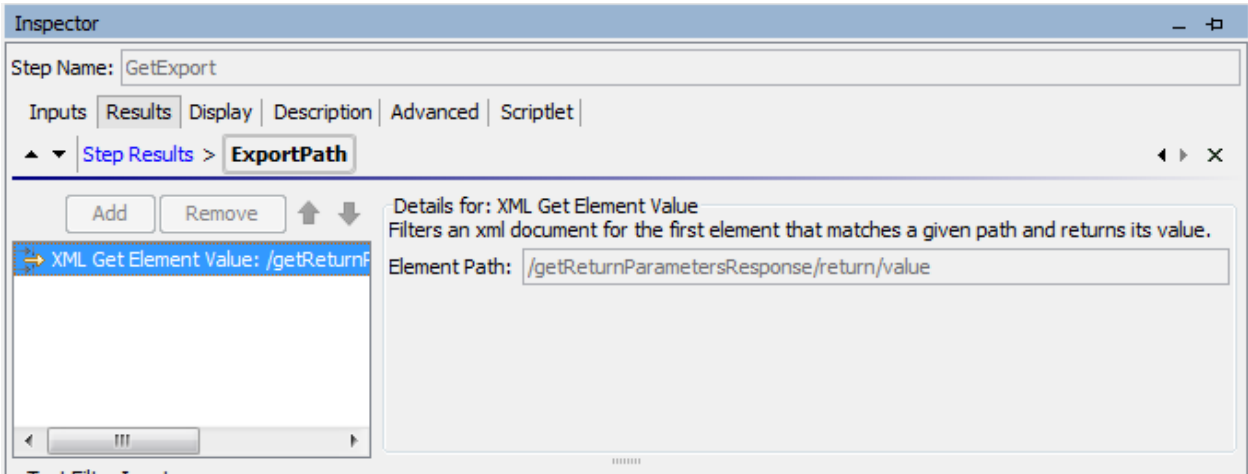
The WFA Web Services API includes a `getReturnParameters` method to extract the return parameters from a WFA workflow. This section uses the `CreateQtreeAndNFSEExport` example workflow to illustrate this. The return parameter from that workflow is the export that is created by the workflow.

The output of the `getReturnParameters` Web Services method appears as follows:

```
<getReturnParametersResponse>
  <return>
    <name>ExportPath</name>
    <value>/vol/dantest/anotherq</value>
  </return>
</getReturnParametersResponse>
```

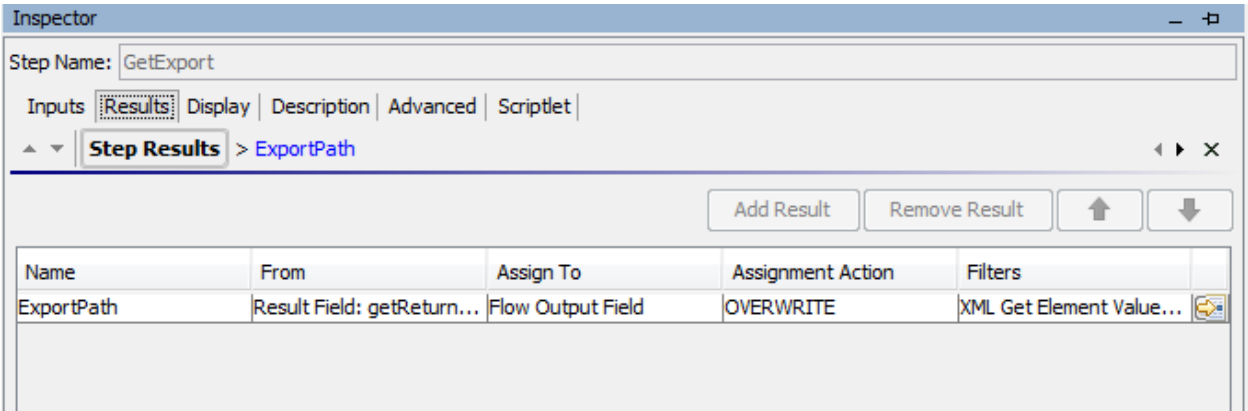
We want the value of the export, so we use a filter on the return to extract the value. As shown in Figure 18, we use the XML Get Element Value filter with the element path of `/getReturnParametersResponse/return/value`.

Figure 18) GetExport XPath statement.



This result is assigned to the Flow Output Field (as shown in Figure 19) to make sure that it gets passed out of the OO flow so that this flow can be used by higher-level OO flows.

Figure 19) GetExport subflow results.



Note: The `getReturnParameters` subflow provided in the toolkit is capable of extracting multiple parameters in a single invocation. If you have a workflow with multiple return parameters, the example process used in `CreateQtreesAndNFSEExport` (shown previously) can be used multiple times in your subflow using different values for the `parameters0` input to extract the additional return parameters.

5 Conclusion

This document describes how to use the OnCommand WFA Toolkit for HP OO to execute the OnCommand WFA workflows from HP OO. It provides the foundation for end-to-end automation scenarios for IT as a service providers and enterprises that are leveraging the best of these two automation frameworks. Using best practices, a walkthrough of the individual steps to set up the integration and run a selected WFA workflow triggered by HP OO is described. The document also shows how to modify the example OO flows that come in the toolkit to run any other WFA workflow.

For an experienced HP OO consultant with basic knowledge of OnCommand WFA, performing the steps described in this document should take only a few minutes, assuming that both HP OO and OnCommand WFA are installed with existing WFA storage workflows.

References

The following references were used in this technical report:

- OnCommand WFA Web Services Primer:
<https://communities.netapp.com/docs/DOC-15285>
- OnCommand WFA Community:
https://communities.netapp.com/community/products_and_solutions/storage_management_software/workflow-automation

Acknowledgements

The authors would like to acknowledge the following people for their assistance with the development of this toolkit and technical report:

- Rabe Toubali, Peter Lees, Hendrik Land, Adolf Hohl, and Daniel Chan, all of whom were contributing authors of the previous version of this technical report, for their review of this update.
- Ginny Smith, Lucian Revnic, Shay Wissotzky Cohen, Muneer Mubashir, and Ravi Srinivasan of HP Software, for their support during the development of the toolkit.
- Kevin Hill and Yaron Haimsohn, for their tireless support of WFA users.
- Kristina Brand, Esther Smitha, and the entire TR editing team for their help finding the mistakes we made.

Refer to the [Interoperability Matrix Tool](#) (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

[Go further, faster®](#)