HP Operations Orchestration Software

Software Version: 7.60.01

HP Network Automation Integration

Document Release Date: April 2010

Software Release Date: April 2010

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

For information on open-source and third-party software acknowledgements, see in the documentation set for this release, Open-Source and Third-Party Software Acknowledgements (3rdPartyOpenNotices.pdf).

On the Web: Finding OO support and documentation

There are two Web sites where you can find support and documentation, including updates to OO Help systems, guides, and tutorials:

- The OO Support site
- BSA Essentials Network

Support

Documentation enhancements are a continual project at Hewlett-Packard Software. You can obtain or update the HP OO documentation set and tutorials at any time from the HP Software Product Manuals Web site. You will need an HP Passport to log in to the Web site.

To obtain HP OO documentation and tutorials

1. Go to the HP Software Product Manuals Web site (*http://support.openview.hp.com/selfsolve/manuals*).
2. Log in with your HP Passport user name and password.

   OR

   If you do not have an HP Passport, click **New users – please register** to create an HP Passport, then return to this page and log in.

   If you need help getting an HP Passport, see your HP OO contact.
3. In the **Product** list box, scroll down to and select **Operations Orchestration**.
4. In the **Product Version** list, click the version of the manuals that you're interested in.
5. In the **Operating System** list, click the relevant operating system.
6. Click the **Search** button.
7. In the **Results** list, click the link for the file that you want.

BSA Essentials Network

For support information, including patches, troubleshooting aids, support contract management, product manuals and more, visit the following site: *http://www.hp.com/go/bsaessentialsnetwork*

This is the **BSA Essentials Network** Web page. To sign in:

1. Click **Login Now**.
2. On the **HP Passport sign-in** page, enter your HP Passport user ID and password and then click **Sign-in**.
3. If you do not already have an HP Passport account, do the following:
   a. On the **HP Passport sign-in** page, click **New user registration**.
   b. On the **HP Passport new user registration** page, *enter the required information and then click **Continue***.
   c. On the confirmation page that opens, check your information and then click **Register**.
   d. On the **Terms of Service** page, read the Terms of use and legal restrictions, select the **Agree** button, and then click **Submit**.
4. On the **BSA Essentials Network** page, click **Operations Orchestration Community.**

The Operations Orchestration Community page contains links to announcements, discussions, downloads, documentation, help, and support.

Note: Contact your OO contact if you have any difficulties with this process.

In OO: How to find Help, PDFs, and tutorials

The HP Operations Orchestration software (HP OO) documentation set is made up of the following:

- Help for Central

  Central Help provides information to the following:

  - Finding and running flows
  - For HP OO administrators, configuring the functioning of HP OO
  - Generating and viewing the information available from the outcomes of flow runs

  The Central Help system is also available as a PDF document in the HP OO home directory, in the \Central\docs subdirectory.

- Help for Studio

  Studio Help instructs flow authors at varying levels of programming ability.

  The Studio Help system is also available as a PDF document in the HP OO home directory, in the \Studio\docs subdirectory.

- Animated tutorials for Central and Studio

  HP OO tutorials can each be completed in less than half an hour and provide basic instruction on the following:

  - In Central, finding, running, and viewing information from flows
  - In Studio, modifying flows

  The tutorials are available in the Central and Studio subdirectories of the HP OO home directory.

- Self-documentation for operations and flows in the Accelerator Packs and ITIL folders

  Self-documentation is available in the descriptions of the operations and steps that are included in the flows.

Table of Contents

Overview of HP Network Automation integration

With this integration, administrators can build HP Operations Orchestration (OO) flows that are integrated into HP Network Automation (NA).

To use this integration successfully, you should have administrator-level knowledge of the HP Network Automation software, with a working knowledge of the NA command line interface (CLI) in particular.

This document describes the configuration and usage of the HP Network Automation (NA), formerly Opsware Network Automation System (NAS), in integration with HP Operations Orchestration.

Use cases and scenarios

This section defines the major use cases for the HP Network Automation integration, and lists the operations and flows that you can use to implement them.

1. Apply an update to multiple devices and rollback changes for all devices if any update fails:
   - List Iterator
   - Run Command Script
   - Rollback Device to Last Snapshot
2. Check the status of a task:
   - List Tasks
3. Change the configuration on a Cisco router, but rollback changes if the router becomes unreachable:
   - Get Configurations By IP
   - Take Snapshot
   - Delayed Reload
   - Deploy Configuration
   - Cancel Delayed Reload
4. Perform diagnostics on a list of network devices:
   - List Iterator
   - Diagnose Device
   - List Diagnostic
   - Get Device Diagnostic
   - Generate Report From Lists
5. Rollback device configuration to last snapshot:
   - Get Configurations By IP
   - Deploy Configuration

Note: Each use case also has a corresponding flow in the **Samples** folder that implements it. For more details please see the *Operation and flow specifics* section of this document.

Installation and configuration Instructions

The integration for NA 6.x utilizes the SSH command line interface available for NA 6.x and requires that the NA SSH service be accessible from the RAS host. Version 7.x integration uses the NA Java API, which requires that NA has the SOAP Web service available and running (default).

In all versions of the NA integration, a Java Remote Action Service (RAS) must be installed that can access the NA core(s) that are targeted.

Versions

Operations Orchestration Version	HP Network Automation Version
7.60.01	6.x, 7.x*

*Tested through version 7.60 as of this writing.

Architecture

Depending on the integration version you are using, the RAS host will either communicate with the NA core using the NA SSH command line interface (for NA 6.x) or the NA Java API using SOAP over HTTP or HTTPS (for NA 7.x). The default port for SSH communication is **22**, and for SOAP it is **HTTPS (443)**, both of which can be changed if necessary.

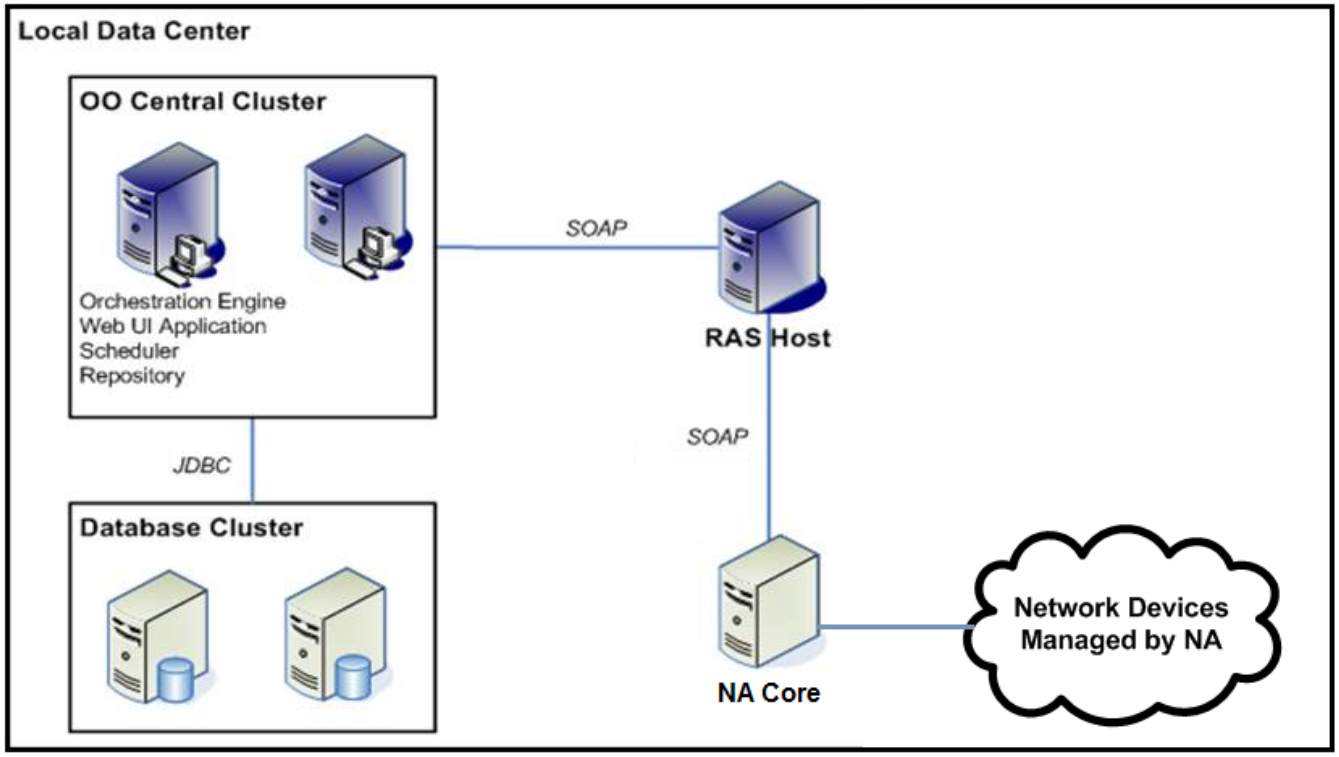

Figure 1 – HP Network Automation architecture

What you need to know about HP Network Automation before using the integration

You must have a working knowledge of the HP Network Automation Command Line Interface (CLI) to understand and use this integration effectively. All of the operations in this integration are based on CLI commands. The *Operation and flow specifics* section of this guide will help you correlate OO operations to CLI commands.

Important: In order to use operations and flows in the **Task** folder, you need to have workflow mode enabled on your Network Automation host. Please see the Network Automation documentation to learn how to accomplish this, and be sure that you understand the implications of enabling workflow mode.

Some of the flows in the Samples/ folder for this integration require custom data to be set up on the NA host. For more information please see the Network Automation User Guide chapter **Custom Data Setup**.

HP Network Automation integration operation and flow infrastructure

The HP Network Automation integration includes the following operations and flows in the OO Studio Library/Integrations/Hewlett-Packard/Network Automation/ folder.

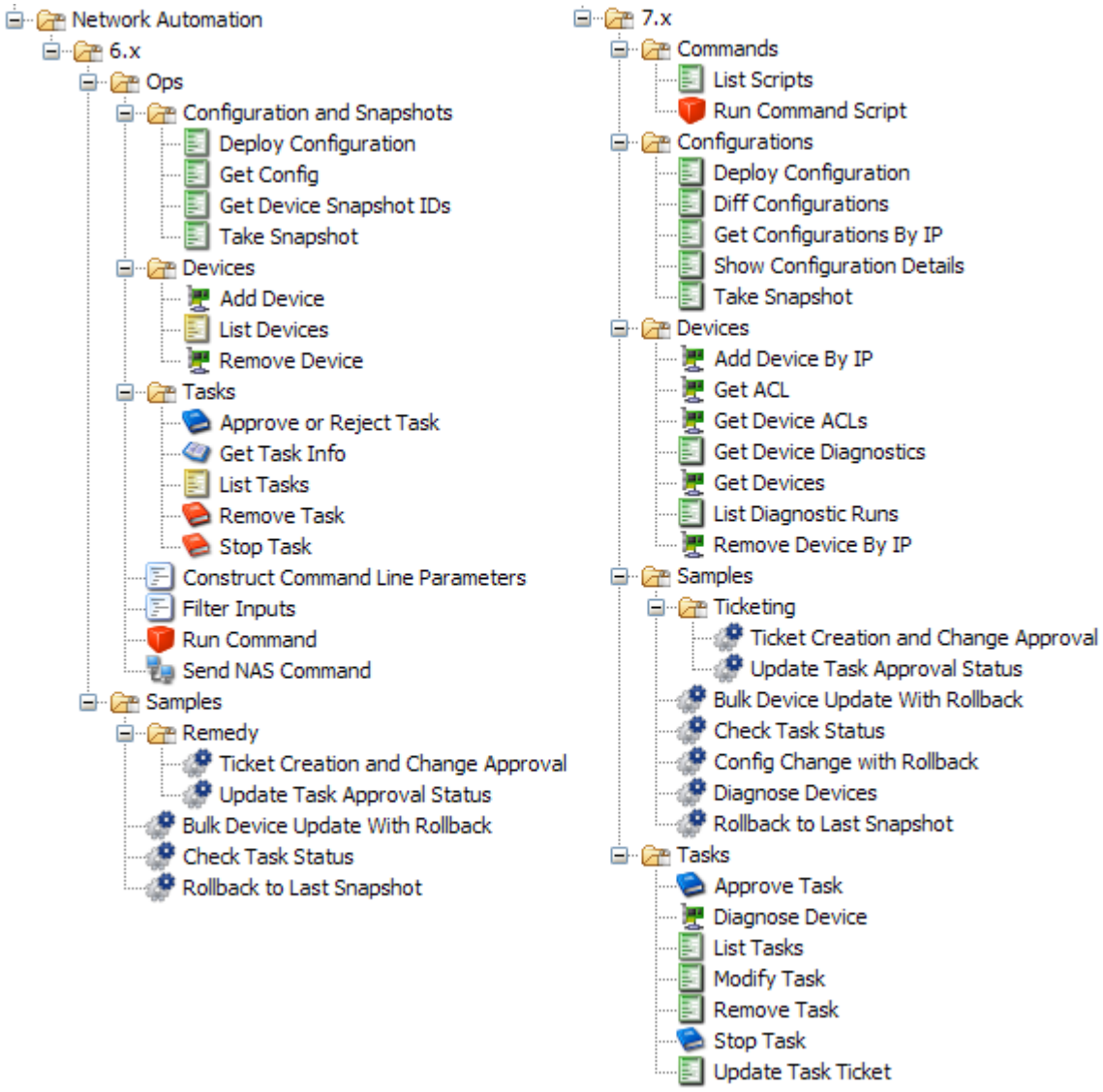

Figure 2 - HP Network Automation integration operation and flow infrastructure

Common inputs in the integration

OO flows and operations use inputs to specify how they obtain the data that they need and when the data is obtained. The following inputs are used consistently throughout the HP Network Automation integration's operations and flows.

hostname (6.x), **coreHost** (7.x)

The Network Automation server with which you want to communicate. You can specify this using the DNS name (for example, nahost.mydomain.com) or IP address (for example, 10.2.255.116). There is no default for this input.

username (6.x), **coreUsername** (7.x)

The user name to use to log on to the NA core server.

password (6.x), **corePassword** (7.x)

The password for the user to log on to the NA core host. This value is an encrypted input, and has no default value.

coreProtocol (7.x)

This is the protocol to use for communication with the NA core server. Valid values are **http** and **https**. The default value for **coreProtocol** is **https**.

corePort (7.x)

The port to use for communication with the NA core host. The default value for **corePort** is **443**.

Operation and flow specifics

This section describes the HP Network Automation integration's flows and operations, including any operation- or flow-specific inputs. The items are grouped by their basic function in the following groups:

- Commands
- Configurations
- Devices
- Samples
- Tasks

The operations and flows listed in this section all correspond to the Network Automation version 7.x integration, but the information applies to the NA version 6.x integration with only minor changes in input names. For details refer to the **Description** tabs for the associated 6.x operations and flows in Studio.

Commands

This section contains content that does the actual communication with the NA core server, and applies only to version 7.x of the integration since the command interface for version 6.x is a simple SSH session.

List Scripts

The **List Scripts** operation queries the NA core host for the list of scripts that are installed. You can filter this list by setting the inputs in the operation. This operation corresponds to the **List Scripts** CLI command.

The operation produces a **Success** response if the operation succeeds. It produces a **Failure** response if the operation fails.

Other results are:

commandStatus

The return status of the NA command (for example, **200**).

Text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

commandName

The name of the NA command, script, or diagnostic (if there is more than one script, use a comma-delimited list).

customScriptID

The custom script ID (if there is more than one script, use a comma-delimited list).

lastModifiedDate

The date when this object was last modified.

scriptData

The script data (if there is more than one script, use a comma-delimited list).

scriptType

The type of script (**1** for command, **2** for diagnostic, **3** for advanced). (If there is more than one script, use a comma-delimited list).

scriptMode

Cisco IOS (Internetwork Operating System) mode (if there is more than one script, use a comma-delimited list).

taskType

The type of the task (if there is more than one script, use a comma-delimited list).

Parameters

The parameters available for the script (if if there is more than one script, use a comma-delimited list).

variableData

The script or command variables (if there is more than one script, use a comma-delimited list).

The inputs for the operation are:

scriptType

The integer type of the desired script or diagnostic. This may be command, diagnostic, or advanced (**1** for command, **2** for diagnostic, **3** for advanced).

scriptName

The script or command name.

userType

The user-defined script type (category). For example, "Core Provisioning Scripts". This is only applicable for command scripts and advanced scripts.

scriptMode

The script mode. For command scripts and diagnostics, the script's level of device access (such as **Cisco IOS enable**); for advanced scripts, the device family (such as **Cisco IOS**).

ids

The object IDs to use or filter by.

Run Command Script

The **Run Command Script** operation runs an existing script, specified by name, against a device or group of devices. The proper variant of the script will be applied to each device. If no variant of the script supports a given device, that device is skipped. The script is run as a system task.

The operation produces a **Success** response if the command script is successfully scheduled. It produces a **Failure** response if the scheduling of the command script fails.

Note: A **Success** response does not necessarily indicate that the script has successfully completed, only that the inputs have sufficiently passed validation for it to be placed on the NA queue.

Other results are:

commandStatus

The status of the NA command.

Text

The response text of the NA command.

stackTrace

The stack trace (If any) of the NA command.

The inputs for the operation are:

scriptName

The script or command name.

isAdvanced

Specifies whether the command should be an advanced script. The valid values are **True** and **False**.

ip

The IP address of the network device.

variables

A list of variables that are supplied to the script. The variables are provided as a list of name=value pairs, separated by commas. Values can be surrounded in single-quotes ('). Within a quoted value, a single-quote can be embedded with two single-quote characters. For example, `"variable1=value1,varable2='this is ''value 2'''"`

returnImmediately

A boolean value which controls how the command script is run. A value of **true**, **0,** or **yes**, will cause the operation to return once the command script is scheduled. A value of **false, 1,** or **no,** will cause the operation to wait until the script has been run before it returns.

parameters

The command line parameters for the script to run. These are ignored for regular command scripts.

lineByLine

A value of **1** specifies a line-by-line deployment of the script; a value of **0** specifies a file-based deployment. This is required by regular command scripts, but is ignored for advanced scripts.

startDate

The date of the start of the task. Use one of the following formats:

- YYYY-MM-DD HH:MM:SS, for example 2002-09-06 12:30:00
- YYYY-MM-DD HH:MM, for example 2002-09-06 12:30
- YYYY-MM-DD, for example 2002-09-06
- YYYY/MM/DD, for example 2002/09/06
- YYYY:MM:DD:HH:MM, for example 2002:09:06:12:30
- now
- today
- yesterday
- tomorrow
- "*number time descriptor* ", for example "3 days ago" where *number* is a positive integer; *time* is `seconds`, `minutes`, `hours`, `days`, `weeks`, `months`, or `years`; *descriptor* is `ago`, `before`, `later`, or `after`.

groupName

The group name of a list of networked devices.

repeat

Specifies the time interval for running the command repeatedly. Use one of the following formats:

- #min
- #:#
- #days
- #weeks
- #months

where # is a positive integer. *#:#* indicates *hours:minutes where* the two integers do not have to be the same.

comment

The comment to tie to this action.

Configurations

Deploy Configuration

The **Deploy Configuration** operation deploys a configuration on a given device using the deploy configuration task in NA.

The operation produces a **Success** response if the configuration is successfully deployed. It produces a **Failure** response if anything goes wrong.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

id

The ID of the configuration.

ip

The IP address of the network device.

returnImmediately

Specifies whether the operation returns results once the task is scheduled. The valid values are **true** and **false.** A value of **true** causes the operation to wait for the task to complete before returning; a value of **false** causes the operation to return immediately after the script is scheduled.

Diff Configurations

The **Diff Configurations** operation is used to compare two configurations, and the differences, if there are any, are returned in the result field **returnResult**.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

id1

Configuration ID 1.

id2

Configuration ID 2.

Get Configurations By IP

The **Get Configurations By IP** operation returns a comma-separated list of the configuration IDs that are available to a specified device.

The operation produces a **Success** response if the command completes. It produces a **Failure** response if something goes wrong.

Other results are:

lastModifiedDate

The date when this object was last modified. Use the format yyyy-MM-dd HH:mm:ss.SSS (for example, 2010-02-22 23:45:39.0).

The inputs for the operation are:

ip

The IP address of the device to search for configuration IDs.

Show Configuration Details

The **Show Configuration Details** operation retrieves detailed information about a specific device configuration.

The operation produces a **Success** response if the details are retrieved correctly. Otherwise, it produces a **Failure** response.

The inputs for the operation are:

id

The ID of the configuration object to process.

Take Snapshot

The **Take Snapshot** operation saves the configuration (snapshot) of a device, which you can use as part of a rollback or backup operation.

The operation produces a **Success** response if the snapshot is saved successfully. Otherwise, it produces a **Failure** response.

The inputs for the operation are:

ip

The IP address of the network device. The **ip** and **groupName** inputs cannot both be empty.

groupName

The group name of a list of networked devices. The **ip** and **groupName** inputs cannot both be empty. If you supply a value for the **ip** input, the **groupName** value is ignored.

comment

The comment to tie to this action.

Devices

Add Device By IP

The **Add Device By IP** operation adds a device to the NA list of managed devices.

The operation produces a **Success** response if the device is added successfully. Otherwise, it produces a **Failure response**.

Note: A **Success** response from this operation does not indicate that NA has communicated with the specified device. It indicates that the device passed a rudimentary set of input validations, and that NA will now try to run discovery against this device. Any errors are returned in the **returnResult** result.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

ip

The IP address of the network device.

Get ACL

The **Get ACL** operation retrieves an Access Control List (ACL) from the NA host.

The operation produces a **Success** response if the ACL is found. It produces a **Failure** response if there is an error.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

aclScript

The script data for the ACL. If there is more than one ACL, this is a comma-delimited list.

deviceAclID

The device's ACL ID number. If there is more than one ACL, this is a comma-delimited list.

aclID

The ACL ID number. If there is more than one ACL, this is a comma-delimited list.

aclHandle

The ACL handle. If there is more than one ACL, this is a comma-delimited list.

aclType

The type of the ACL. If there is more than one ACL, this is a comma-delimited list.

appState

The application state. If there is more than one ACL, this is a comma-delimited list.

lastModifiedAccessLogID

The access log ID for the last modification. If there is more than one ACL, this is a comma-delimited list.

mostRecent

Specifies whether this is the most recent ACL. The valid values are **1** for yes and **0** for no. If there is more than one ACL, this is a comma-delimited list.

lastModifiedDate

The date when this object was last modified. Use the format yyyy-MM-dd HH:mm:ss.SSS (for example, 2010-02-22 23:45:39.0).

The inputs for the operation are:

id

The ID of the ACL of a device.

Get Device ACLs

The **Get Device ACLs** operation retrieves a comma-separated list of ACL IDs associated with a device.

The operation produces a **Success** response if the command completes. Otherwise, it produces a **Failure** response.

Note: This operation does not check the validity of the device ID sent to it. Therefore, an invalid device ID will produce the same results as a valid device ID with no ACLs associated with it—an empty ACL list and a response of **Success**.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

deviceAclID

The device's ACL ID number. there is more than one ACL, this is a comma-delimited list.

aclID

The ACL ID number. there is more than one ACL, this is a comma-delimited list.

aclHandle

The ACL handle. there is more than one ACL, this is a comma-delimited list..

aclType

The type of the ACL. there is more than one ACL, this is a comma-delimited list.

appState

The application state. there is more than one ACL, this is a comma-delimited list.

lastModifiedAccessLogID

The access log ID for last modification. there is more than one ACL, this is a comma-delimited list.

mostRecent

Specifies whether this is the most recent ACL. The valid values are **1** for yes and **0** for no. If there is more than one ACL, this is a comma-delimited list.

lastModifiedDate

The date when this object was last modified. Use the format yyyy-MM-dd HH:mm:ss.SSS.

The inputs for the operation are:

deviceID

A valid device ID.

recent

Specifies whether to only list ACLs that are most recent. The valid values are **true** and **false**.

includeScript

Set to **1** to include Script in the results.

includeApp

Set to **1** to include Application in the results.

Get Device Diagnostics

The **Get Device Diagnostics** operation retrieves the data from a specific diagnostic task run, and returns the diagnostic data in the primary result **returnResult**.

The operation produces a **Success** response if the command completes. Otherwise, it produces a **Failure** response.

Other results are:

returnResult

The result of NA diagnostic run.

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command

The inputs for the operation are:

id

The Diagnostic ID for the diagnostic task run you wish to retrieve.

Get Devices

The **Get Devices** operation retrieves a list of devices or detailed information about a specific device. The scope of the list returned can be modified using the inputs **ip**, **groupName**, and **family**.

The operation produces a **Success** response if the command completes. Otherwise, it produces a **Failure** response.

Other results are:

returnResult

Comma delimited list of devices or information about a single device

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

deviceID

The ID of the device. If there is more than one device, this is a comma-delimited list.

deviceHostname

The hostname of the device. If there is more than one device, this is a comma-delimited list.

deviceIP

The IP address of the device. If there is more than one device, this is a comma-delimited list.

deviceSiteName

The site of the device. If there is more than one device, this is a comma-delimited list.

deviceStatus

The status of the device. If there is more than one device, this is a comma-delimited list.

deviceType

The type of the device. If there is more than one device, this is a comma-delimited list.

The inputs for the operation are:

ip

The IP address of the network device.

groupName

The name of the group that the device is in.

family

The name of the family the device is in. For example, "Cisco IOS".

List Diagnostic Runs

The **List Diagnostic Runs** operation retrieves a list of historic IDs for a particular diagnostic run against a particular device which you specify by IP address.

The operation produces a response of **Success** if the command completes. Otherwise, it produces a **Failure** response.

Other results are:

returnResult

A comma-delimited list of NA diagnostic run IDs.

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

ip

The IP address of the network device.

diagnosticName

The name of the diagnostic to search for. Examples are **NA Routing Table**, **NA Interfaces**, and **NA OSPF Neighbors**. In Studio, see the **NA Diagnostics** selection list located in the Library/Configuration/Selection Lists/ folder for a complete list of diagnostics.

Note: If you are integrating with a Network Automation version earlier than 7.50, change the default selection list to **NAS Diagnostics** selection list.

Remove Device By IP

The **Remove Device By IP** operation removes a device you specify by IP address from the NA inventory.

The operation produces a **Success** response if the command completes. Otherwise, it produces a **Failure** response.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

ip

The IP address of the device to remove.

Samples

Ticket Creation and Change Approval

The **Ticket Creation and Change Approval** flow shows the process for creating a ticket in Remedy, and attaching the ticket number to the NA task. This flow was designed to be kicked off via Remedy using the RSFlowInvoke utility (see *Tools* for more information).

Note: This flow will not work properly unless you reference the appropriate Remedy operations. Please substitute with the appropriate operations for your Remedy and substitute variables as necessary. Furthermore, the flow references the custom field **ChangeID** for tasks in NA.

The flow produces a **Success** response if linked tickets between Remedy and NA are created. Otherwise, it produces a **Failure** response.

The inputs for the flow are:

TaskID

The NA task ID.

remedyMidTier

The Remedy mid-tier server.

remedyARServer

The Remedy AR server.

remedyUser

The Remedy username.

remedyPassword

The Remedy user's password.

Update Task Approval Status

The **Update Task Approval Status** flow shows the framework for a task approval integration between NA and a (hypothetical) Remedy system. The Remedy system kicks off this flow with the **remedy_status** parameter to reject or approve the NA task. Operations Orchestration then updates the NA ticket with the information from Remedy.

Note: This flow assumes that events have already been set up for NA to interact with Remedy. Remedy operations in the flow are placeholders, and must be replaced with flows customized for your Remedy installation.

The flow produces a **Success** response if the NA task was approved (or rejected) in both Remedy and NA, and a **Failure** response if it fails to synchronize approval or rejection of the NA task with Remedy.

Other results are:

cmdResult

The output of the last command.

The inputs for the operation are:

remedy_ticketnumber

The Remedy ticket number (not used currently).

remedy_status

The Remedy ticket's ticket. The valid values are **Approved** and **Rejected.** Assigning any other value will cause a failure.

TaskID

The NA task to approve or reject.

Bulk Device Update With Rollback

The **Bulk Device Update With Rollback** flow runs a command script (simple) across a sequence of devices. If a failure occurs for any of the devices, the flow rolls back all of the changes made in reverse.

The flow produces a **Success** response if the script is run successfully against the entire list of devices. If an error occurs and all of the changes are rolled back successfully, the flow generates an **Error in Deployment, Rollback Complete** response. If an error occurs and all of the changes are not rolled back successfully, the flow produces a **Failure** response.

Other results are:

commandOutput

The output from the last NA command.

The inputs for the flow are:

deviceList

A comma-delimited list of NA device IP addresses in order of execution to run the script against.

scriptName

The name of the command script to run against the devices, enclosed in quotes (“).

Check Task Status

The **Check Task Status** flow demonstrates how to use the **List Tasks** operation to retrieve the status of a given task.

The flow produces a **Success** response if the state of the task is determined, and a **Failure** response if the task is in a non-stable state or the command execution failed.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the flow are:

TaskID

The ID of the task check status on.

Config Change with Rollback

The **Config Change with Rollback** flow demonstrates how to modify the configuration on a Cisco device, with a failsafe timeout that will undo changes if communication with the Cisco device is lost.

The flow produces a **Success** response if the configuration is deployed and committed to the device. Otherwise, it produces a **Failure** response.

The inputs for the flow are:

host

The IP address of your Cisco router.

username

The username for the host.

password

The password for the host.

enablePassword

The enabled mode password for the host.

groupName

The name of the NA group containing the device.

Diagnose Devices

The **Diagnose Devices** flow schedules the run of a diagnostic for each device (IP) in a list of network devices.

The flow produces a **Success** response if the diagnostic run is successful for each device. Otherwise, it produces a **Failure** response.

Other results are:

returnResult

The diagnostics report on the devices.

The inputs for the flow are:

deviceIPs

A comma-delimited list of network device IPs on which to run a diagnostic.

diagnosticName

The name of the diagnostic to run on the IPs. This is any valid diagnostic name in NA.

waitTime

The time to wait for the diagnostic tasks to complete (in seconds).

Rollback to Last Snapshot

The **Rollback to Last Snapshot** flow demonstrates how to find the last known good snapshot for a device, and roll back the device's configuration to that point.

The flow produces a **Success** response if the last snapshot is found and the configuration is deployed to the device. Otherwise, it produces a **Failure** response.

The inputs for the flow are:

ip

The IP address of the device for which to find the last snapshot and roll back the configuration.

Tasks

Approve Task

The **Approve Task** operation allows you to approve or reject a task which you specify by ID.

The flow produces a **Success** response if the task is successfully approved (or rejected), and a **Failure** response if something goes wrong.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

id

The ID of the object to process.

approve

If **True**, approve the task; if **False**, reject it.

Diagnose Device

The **Diagnose Device** operation runs a diagnostic task against a device or device group.

The operation produces a **Success** response if the diagnostic run is scheduled successfully. Otherwise, it produces a **Failure** response.

Other results are:

returnResult

The text result of NA diagnostic

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

diagnosticName

The name of the diagnostic for which to search. Examples are **NA Routing Table**, **NA Interfaces**, and **NA OSPF Neighbors.** See the **NA Diagnostics** selection list for a complete list of diagnostics.

Note: If integrating with a Network Automation version earlier than 7.50, change the default selection list to **NAS Diagnostics** selection list.

ip

The IP address of the network device.

groupName

The group name of a list of networked devices.

comment

The comment to tie to this action.

repeat

Specifies the time interval for running the diagnostic repeatedly. Use one of the following formats:

- #min
- #:#
- #days
- #weeks
- #months

where # is a positive integer. *#:#* indicates *hours:minutes where* the two integers do not have to be the same.

startDate

- The date of the start of the task. This can be in one of the following formats: YYYY:MM:DD:HH:MM, for example 2002:09:06:12:30
- now
- tomorrow

returnImmediately

The valid values are value **1** (true) or **0** (false). If the value is **1**, the operation returns once the task is scheduled. If the value is **0**, the operation waits for the task to complete before returning.

Get Task Info

The **Get Task Info** operation returns a list of all of the fields of an NA task, including custom fields.

The operation produces a **Success** response if the task information is retrieved, and it is not in a blocking state. It produces a **Failure** response if the task information is not retrieved, and a **Blocked** response if the task information is retrieved, but its state is **pending**, **requested**, **running**, **waiting**, or **starting**.

Other results are:

TaskResult

The output from the task run.

TaskStatus

The status of the task. Possible task states are **pending**, **succeeded**, **failed**, **running**, **paused**, **starting**, **waiting**, **synchronous**, **skipped**, and **warning**.

The inputs for the operation are:

TaskID

The ID of the task for which to retrieve information.

List Tasks

The **List Tasks** operation returns a comma-delimited list of NA task IDs. The list can be filtered using various inputs. If none of these inputs are supplied, then all tasks are listed.

The operation produces a **Success** response if the task list is successfully retrieved. Otherwise, it produces a **Failure** response.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

TaskID

The task ID. If there is more than one task, this is a comma-delimited list.

taskName

The name of the task. If there is more than one task, this is a comma-delimited list.

taskStatus

The task status. The value must be an integer which cannot be larger than 255. If there is more than one task, this is a comma-delimited list.

taskType

The type of task. If there is more than one task, this is a comma-delimited list.

taskCreateDate

The task creation date in the format YYYY:MM:DD:HH:mm.

taskScheduleDate

The date the task is scheduled for next in the format YYYY:MM:DD:HH:mm.

taskData

The task data (in XML). This can be blank (null). If there is more than one task, this is a comma-delimited list.

The inputs for the operation are:

id

The ID of the task.

taskName

The name of the task.

taskStatus

Display only those tasks with the specified status: **pending**, **succeeded**, **failed**, **running**, **paused**, **starting**, **waiting**, **synchronous**, **skipped**, or **warning**.

fqdn

The Fully Qualified Domain Name or IP address of a network device.

startDateOn

Display only those tasks whose schedule date falls on or after the given date in the format yyyy:MM:dd:HH:mm.

endDateOn

Display only those tasks whose schedule date falls on or before the given date in the format yyyy:MM:dd:HH:mm.

parentId

A task ID. Display only those tasks whose parent is the task specified by the given task ID.

Modify Task

The **Modify Task** operation is used to modify various details of a given task. For more information on what can be modified, see the **Inputs** section below.

The operation produces a **Success** response if the task modification is successful. Otherwise, it produces a **Failure** response.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

id

The ID of the object to process.

retryInterval

The number of seconds between retries.

expensive

Is the task expensive? If set to **1** the task is marked as expensive. Any other value causes the command to fail.

days

This input differs depending on the task:

- For weekly tasks, **days** is a comma-delimited list of weekdays. Each item in the list is a day of the week upon which the task should be run. Valid weekdays are: **sun**, **mon**, **tue**, **wed**, **thur**, **fri**, and **sat**.

- For monthly tasks, **days** is a single integer between 1 and 31, corresponding to the day of the month upon which the task should be run.

retryCount

The number of times to retry.

repeatType

The metric by which a task repeats. The valid values are **1** (once), **2** (periodically), **3** (daily), **4** (weekly), and **5** (monthly). If you modify this value, then modify the repeat interval or days accordingly.

duration

The estimated duration that the task will run (in minutes).

startDate

The date of the start of the task. This can be in one of the following formats:

- YYYY-MM-DD HH:MM:SS for example, 2002-09-06 12:30:00
- YYYY-MM-DD HH:MM for example, 2002-09-06 12:30
- YYYY-MM-DD for example, 2002-09-06
- YYYY/MM/DD for example, 2002/09/06
- YYYY:MM:DD:HH:MM for example, 2002:09:06:12:30
- now
- today
- yesterday
- tomorrow
- "*number time descriptor* ", for example "3 days ago" where *number* is a positive integer; *time* is `seconds`, `minutes`, `hours`, `days`, `weeks`, `months`, or `years`; *descriptor* is `ago`, `before`, `later`, or `after`.

repeatInterval

This input differs depending on the task:

- For periodic tasks, this is the period in minutes.
- For monthly tasks, each bit of the integer (except the last) represents a day. We recommend using the **days** input to modify the days on which a monthly task runs. This option is invalid with all other tasks.

approve

Specifies whether to approve the task. The valid values are **true** (approve the task) and **false** (rejects the task).

overrideApproval

Override the approval requirement. The value of this input is entered in the comment field of the task, as is the reason why the approval was overridden.

Note: approve and **overrideApproval** are mutually exclusive options. You cannot set values for both, or the command will fail.

customFieldName

The custom field name.

customFieldValue

The custom field value.

comment

The comment to tie to this action.

Remove Task

The **Remove Task** operation removes a task from NA, whether it has run or not.

The operation produces a **Success** response if the task is removed from the NA host. Otherwise, it produces a **Failure** response.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command

The inputs for the operation are:

id

The ID of the task to remove.

Stop Task

The **Stop Task** operation attempts to stop a running task you specify by ID. If you do not specify an ID, then all tasks are stopped.

The operation produces a **Success** response if the operation completes. Otherwise, it produces a **Failure** response.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

id

The ID of the running task to stop. If this is left null or empty, then all running tasks are stopped.

Update Task Ticket

The **Update Task Ticket** operation updates the ticket number associated with an NA task.

Before you can assign a ticket number to a task, you must enable **PASTicketNumber** custom data in NA. In order to do that:

1. Open Network Automation client in a browser. The typical URL is *https://<NA server IP or FQDN>/index.jsp*.
2. Go to **Admin**->**Custom Data Setup**.
3. Choose **Tasks** from the **Custom Data Setup** combo box.
4. Check **API Name** and enter **PASTicketNumber** for the **API Name** and **Display Name**.
5. Click **Save**.

The PAS Ticket Number is available in the **Task Information** page for each task.

The operation produces a **Success** response if the ticket number is updated for the given task ID. Otherwise, it produces a **Failure** response.

Other results are:

commandStatus

The status of the NA command.

text

The response text of the NA command.

stackTrace

The stack trace (if any) of the NA command.

The inputs for the operation are:

id

The task ID to update.

ticketNumber

The ticket number to update with.

Hidden inputs

closeSession

This input is applicable only for NA 7.x. It closes the internal session used by OO to maintain the NA session. In previous versions of the NA integration, OO logged in and logged out of the NA system for every step, potentially slowing down flows. When set to **False** (the default) or not defined, the session is maintained through the lifetime of the flow's execution. When set to **True**, the internal session as well as the underlying NA connection is terminated at the end of the step. This input may be exposed in future versions of the NA integration.

Launching flows

NA can be configured to launch OO flows via the OO REST service. Using the various REST-based services, you can use the following URL syntaxes to interact with HP OO Central.

Note: In the following, synchronous flow execution means that Central does not return a result until the flow run has completed. In asynchronous flow execution, the flow result is returned immediately after the flow is launched.

- To retrieve a list of flows from Central:
  `https://<ooserver>:<port>/PAS/services/http/list`

  where

  `<ooserver>` is the machine on which Central is installed.

  `<port>` is the port that was specified for the HTTPS (HTTP over Secure Sockets Layer (SSL)) protocol when Central was installed.

- To synchronously execute a flow identified by name and location in the OO Studio Library or Central repository:
  `https://<ooserver>:<port>/PAS/services/http/execute/<library_path>`

  where

  `<ooserver>` is the machine on which Central is installed.

  `<port>` is the port that was specified for the HTTPS (HTTP over SSL) protocol when Central was installed.

  `<library_path>` is the location of the flow within the Central repository, including the name of the flow.

- To synchronously execute a flow by UUID:
  `https://<ooserver>:<port>/PAS/services/http/execute/<flow_UUID>`

  where

  `<ooserver>` is the machine on which Central is installed.

  `<port>` is the port that was specified for the HTTPS (HTTP over SSL) protocol when Central was installed.

  `<flow_UUID>` is the universally unique ID of the flow within the Central repository.

- To asynchronously execute a flow by name:
  `https://<ooserver>:<port>/PAS/services/http/execute_async/<library_path>`

  where

  `<ooserver>` is the machine on which Central is installed.

  `<port>` is the port that was specified for the HTTPS (HTTP over SSL) protocol when Central was installed.

  `<library_path>` is the location of the flow within the Central repository, including the name of the flow.

- To execute a flow by UUID (returns immediately after the flow is launched):
  `https://<ooserver>:<port>/PAS/services/http/execute_async/<flow_UUID>`

  where

  `<ooserver>` is the machine on which Central is installed.

  `<port>` is the port that was specified for the HTTPS (HTTP over SSL) protocol when Central was installed.

  `<flow_UUID>` is the universally unique ID of the flow within the Central repository.

Customizing the integration

Adding custom data fields to NA

Some of the **Sample** flows in this integration require Custom Data Fields to be set up on the NA host. The changes needed should be described in detail above. For more information on what custom data fields are, please see the *Network Automation User Guide* chapter **Custom Data Setup.**

(NA 6.x) Using a different remote shell (such as Telnet)

Non-SSH access to NA 6.x is no longer supported, but it is possible to change the protocol used. To do this, you must create a local copy of the NA integration, then change the local copy of the OO **Send NAS Command** operation to use the Telnet protocol with, for example, the **RAS Telnet Shell** operation.

Security

OO uses the standard SSH protocol to integrate with NA 6.x and SOAP to integrate with NA 7.x. Since NA 7.x does not automatically forward to secured ports in its SOAP service and allows HTTP communication in its Web service, users should confirm that the protocol used is HTTPS before sending sensitive information. As of OO 7.50, an internal session is maintained within the flow run context to avoid repeated logon authentication. This session information aside from the ID of the session, is not readily visible or logged within the OO system.

Tools

Following are OO tools that you can use with the HP NA integration:

- **RSFlowInvoke.exe** and **JRSFlowInvoke.jar**

  RSFlowInvoke (RSFlowInvoke.exe or the Java version, JRSFlowInvoke.jar) is a command-line utility that allows you to start a flow without using Central (although the Central service must be running). RSFlowInvoke is useful when you want to start a flow from an external system, such as a monitoring application that can use a command line to start a flow.

  These tools are available in the %OO_home%/Studio/tools/ folder.