Hewlett Packard
Enterprise

HPE Database and Middleware Automation

Software Version: 10.10 or later
For the Linux operating system

Integration Guide for HP Operations Orchestration 10.01

Document Release Date: December 2015
Software Release Date: July 2013

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise 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. HPE 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 HPE 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 Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: **https://softwaresupport.hp.com**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: **https://hpp12.passport.hp.com/hppcf/createuser.do**

Or click the **the Register** link at the top of the HP Software Support page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at: **https://softwaresupport.hp.com**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

https://hpp12.passport.hp.com/hppcf/createuser.do

To find more information about access levels, go to:

https://softwaresupport.hp.com/web/softwaresupport/access-levels

HP Software Solutions Now accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

Contents

About the HPE DMA Integration with HP OO

This chapter contains the following general information about this integration:

Overview

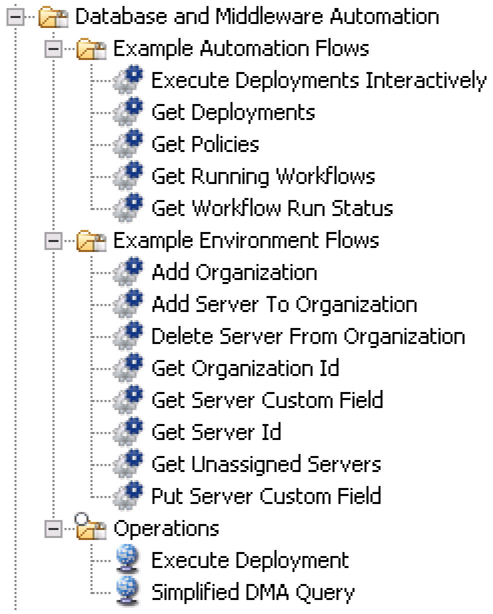
This document provides information about the HPE Database and Middleware Automation (DMA) Content Pack for HP Operations Orchestration (HP OO), a collection of example flows and operations that you can use to manage and invoke DMA automation content from HP OO. This document also provides information about the DMA Wizard for HP OO.

This overview contains general information that pertains to all of the DMA REST APIs. It includes the following information:

"What's Included in the HPE DMA Content Pack"
"Overview"
"Overview"
"Permissions and Capabilities"
"Overview"

What's Included in the HPE DMA Content Pack

The HPE DMA Content Pack contains the following example flows and operations:



- Database and Middleware Automation
  - Example Automation Flows
    - Execute Deployments Interactively
    - Get Deployments
    - Get Policies
    - Get Running Workflows
    - Get Workflow Run Status
  - Example Environment Flows
    - Add Organization
    - Add Server To Organization
    - Delete Server From Organization
    - Get Organization Id
    - Get Server Custom Field
    - Get Server Id
    - Get Unassigned Servers
    - Put Server Custom Field
  - Operations
    - Execute Deployment
    - Simplified DMA Query

Permissions and Capabilities

The DMA capabilities and permissions associated with the specified dmaUser's role (or roles) determine which resources that user can read (GET) and write (PUT, POST, DELETE).

Some flows require the dmaUser to have Administrator capability.

Supported Products and Platforms

This integration requires DMA version 10.10 (or later).

Operating Systems

The DMA version 10.10 or later platform runs on Red Hat Enterprise Linux.

For specific operating system versions supported by the DMA platform and each DMA workflow, see the *HPE Database and Middleware Automation Solution Pack Support Matrix* available here:

http://h20230.www2.hp.com/selfsolve/manuals

For DMA hardware requirements, see the *HPE Database and Middleware Automation Installation Guide*.

Software Requirements

This integration works with HP Operations Orchestration (HP OO) version 10.01 and requires the HP OO Base content pack version 1.0.121 or greater.

Note: You will need to use HP OO Studio to work with this integration. Installation instructions are, therefore, only provided for Windows installations. See "Testing and Deploying Flows" in the *HP OO Studio Authoring Guide* for information about publishing flows that you create to your staging or production HP OO environment.

Install the HPE DMA Content Pack for HP OO

To view or use the example flows and operations included in the DMA content pack for HP OO, you must install this content pack in the pertinent HP OO repository.

Note: Before you install the HPE DMA Content Pack, make sure that the HP OO base content pack is installed in HP OO Studio and the HP OO 10.01 patch is installed.

Note: On Windows 2008 and Windows 2008 R2 servers, you must have administrative privileges to install this content pack. To get administrative privileges, follow these steps:

1. On the Start menu select **All Programs**, and then **Accessories**.
2. Right-click **Command Prompt** and then select the **Run as administrator** option.

Download the HPE DMA Content Pack

1. Go to the HP Live Network Database and Middleware Automation community:

   http://hpln.hp.com/group/database-and-middleware-automation
2. Download and extract the XXXXX file.

Import the HPE DMA Content Pack into HP OO Studio

1. Open HP OO Studio.
2. Select **Import Content Pack**.
3. Navigate to where you extracted the XXXXX file and select the `OO-DMA-01-ContentInstaller.jar` file.

Import the DMA Content Pack into HP OO Central

Tip: You may import the DMA Content Pack into HP OO Central at a later time.

Note: Before you install the HPE DMA Content Pack, make sure that the HP OO base content pack has been imported into HP OO Central.

1. Log in to the HP OO Central, for example:

   `https://localhost:8443/oo`
2. Navigate to the Content Workspace.
3. Select **Deploy New Content**.
4. Select the plus sign (+) to open up a file browser.
5. Navigate to the HPE DMA content file, click **Select**, and then click **Deploy**.

Save the HPE DMA Configuration Wizard for HP OO

Download the latest version of the HP DMA Configuration Wizard for Operations Orchestration from the HP Live Network site for DMA:

https://hpln.hp.com/group/database-and-middleware-automation

Save the executable in the location of your choice.

Note: No installation is required for the HPE DMA wizard. It is provided as a program executable and can be run from any location. The wizard is now ready to use.

Related Topics:

"About the Wizard" on page 71

"Use the Wizard to Create OO Flows" on page 72

HPE DMA Flow Examples

The DMA content pack for HP OO provides two groups of example flows that you can use to automate various database and middleware administration tasks in your managed server environment:

- "Example Automation Flows" on the next page
- "Example Environment Flows" on page 37

This document contains a brief description of each example flow. Additional detailed information is available on the Description tab in the for each flow in the Properties viewer in HP OO Studio.

See Also:

"HPE DMA Operations" on page 64

"HPE DMA Wizard for HP OO" on page 71

Example Automation Flows

The DMA content pack includes the following example automation flows:

Flow Name	Description
"Example Flow: Execute Deployments Interactively"	Enables you to execute a deployment to run an DMA workflow from HP OO.
"Example Flow: Get Deployments"	Returns a comma-separated list of all the deployments that have been created on the specified DMA server.
"Example Flow: Get Policies"	Returns a list of all the policies that have been created on the specified DMA server.
"Example Flow: Get Running Workflows"	Returns a comma-separated list of all the workflows that are currently running (or have completed in the last 60 seconds) on the specified DMA server.
"Example Flow: Get Workflow Run Status"	Returns information about a particular workflow execution on the specified DMA server.

See Also:

"Example Environment Flows" on page 37

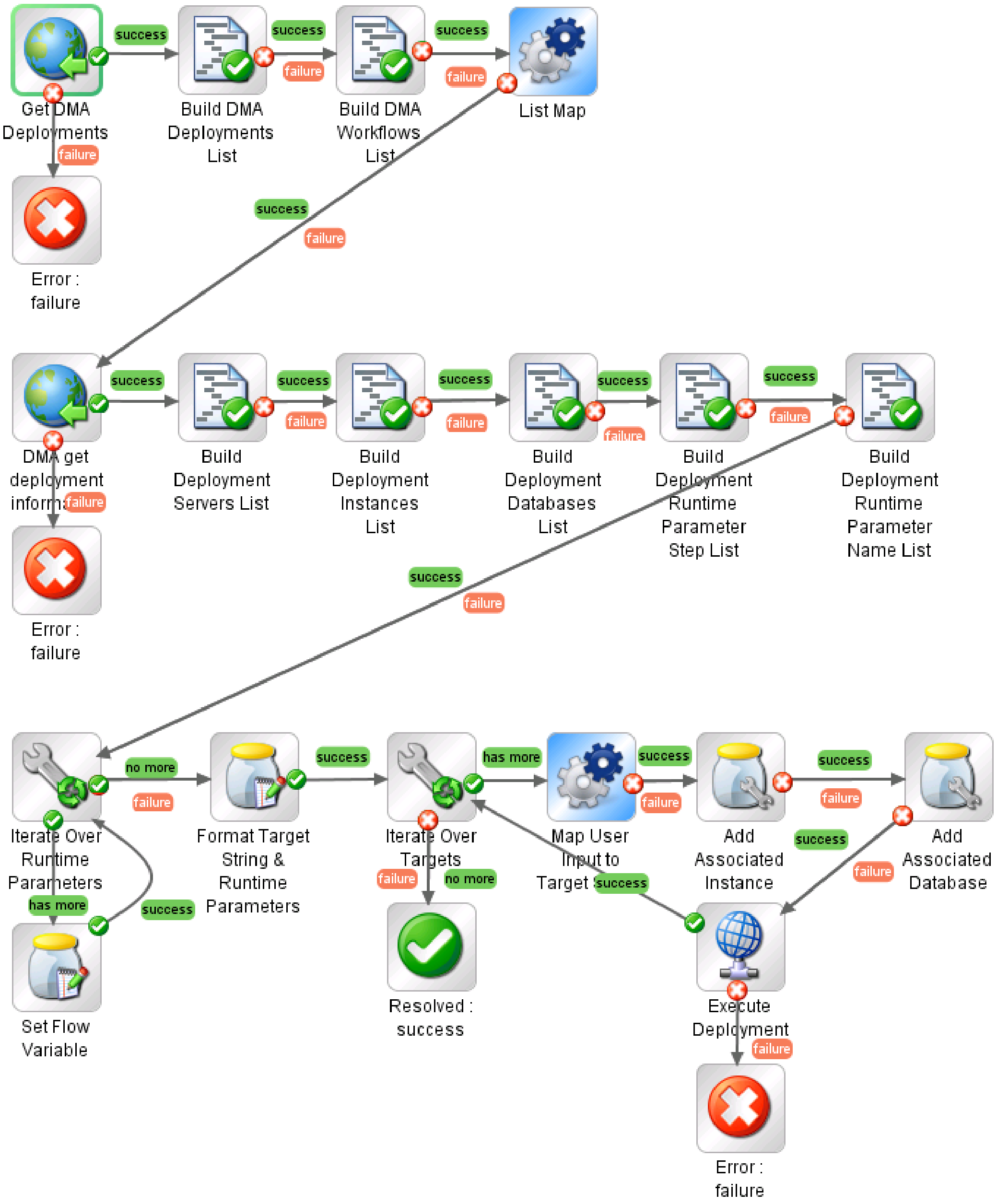
"HPE DMA Operations" on page 64

"HPE DMA Wizard for HP OO" on page 71

Example Flow: Execute Deployments Interactively

The Execute Deployments Interactively example flow enables you to run an DMA workflow from HP OO. The flow does the following things:

- Creates a list of all deployments defined on the specified DMA server.
- Asks you to select a deployment from a drop-down list.
- Collects all the information that it needs to execute the deployment.
- Asks you to specify the targets and any parameter values that must be specified at run time.
- Executes the deployment to run the workflow against the specified targets.



The flow makes numerous calls to the DMA REST API to gather information and, ultimately, execute the deployment.

If it successfully completes all of these processes, it produces a Success response. If it fails to collect the information it needs or fails to run the workflow, it produces a Failure response.

Note: To determine the result of the executed deployment, view the History page on your DMA server.

How to Use this Flow

Use this flow when you want to interactively select an DMA deployment and use that deployment to run a workflow on the DMA server.

Note: Deployment targets that include DMA Smart Groups are not currently supported by this flow.

This flow uses the "Operation: Execute Deployment" operation.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that returns a list of all workflows or all deployments.
- Invoke the DMA API to perform a GET operation that returns information about a specific workflow or deployment.
- Invoke the DMA API to perform a POST operation to run a workflow.
- Use XPath queries to retrieve information from the response bodies of the GET and POST operations.
- Create and present a list of choices to the user.
- Branch, iterate, and set flow variables based on the user's choices.

Flow Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to carry out the requested operation.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.

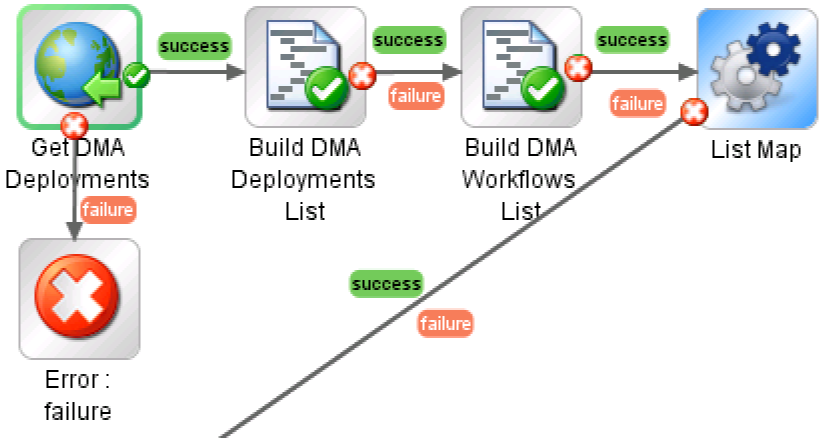
The flow prompts you for additional informations during Phase 1 and Phase 3.

How it Works

The flow has four major phases. Each is briefly described here. For additional detail, view the properties of each step in the flow by using the HP OO Studio Inspector.

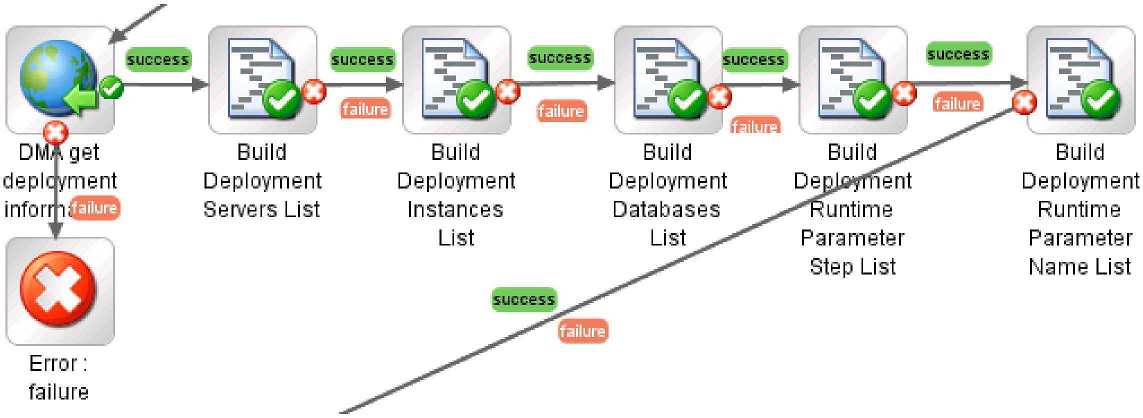
Phase 1: Select a deployment

Phase 1 prompts you for the flow inputs and then uses the DMA REST API to get a list of all the deployments and another list of all the workflows that exist on the DMA server. It then presents a drop-down list of deployments to choose from. You must select a deployment to proceed.



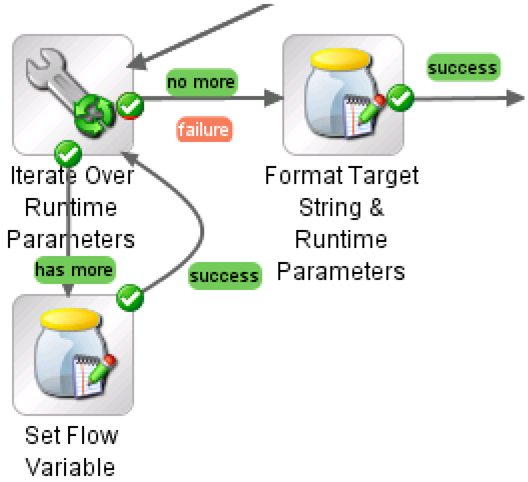
Phase 2: Get information required to execute the selected deployment

Phase 2 queries the DMA server to get all the information about the selected deployment. This includes the available targets (servers, instances, or databases) and any parameters that must be specified at run time. It then presents a drop-down list of targets to choose from. You must select one or more targets to proceed. To select multiple targets, use CTRL+click or SHIFT+click.


success
success
success
success
success
failure
failure
failure
failure
DMA get deployment information
failure
Build Deployment Servers List
Build Deployment Instances List
Build Deployment Databases List
Build Deployment Runtime Parameter Step List
Build Deployment Runtime Parameter Name List
success
failure
Error : failure

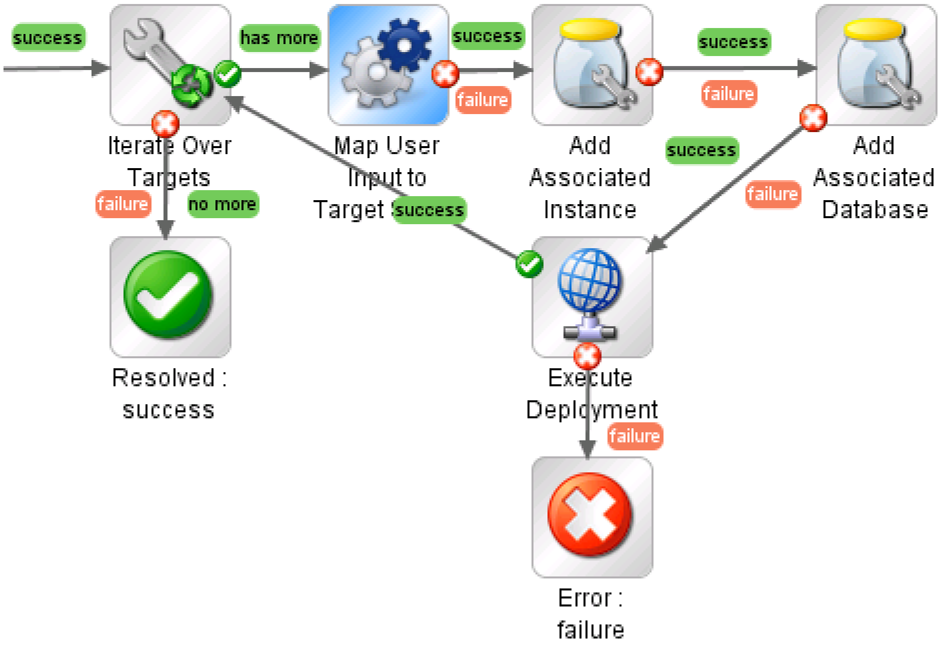
Phase 3: Execute the deployment to run the workflow

Phase 3 asks you to specify values for any Runtime parameters this deployment requires.


no more
failure
success
has more
success
Iterate Over Runtime Parameters
Format Target String & Runtime Parameters
Set Flow Variable

Phase 4: Execute the deployment

Phase 4 executes the deployment to run the workflow against all selected targets.



Permissions and Constraints

To successfully execute a deployment and run a workflow using this flow, the specified dmaUser must have the following permissions:

- READ permission on the workflow
- READ and EXECUTE permission on the deployment
- READ permission on the organizations where the targets reside

Users who have a role with Administrator capability can execute any deployment against any targets.

This flow does not support Smart Groups. Deployments that use Smart Groups to specify targets cannot be executed using this flow.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

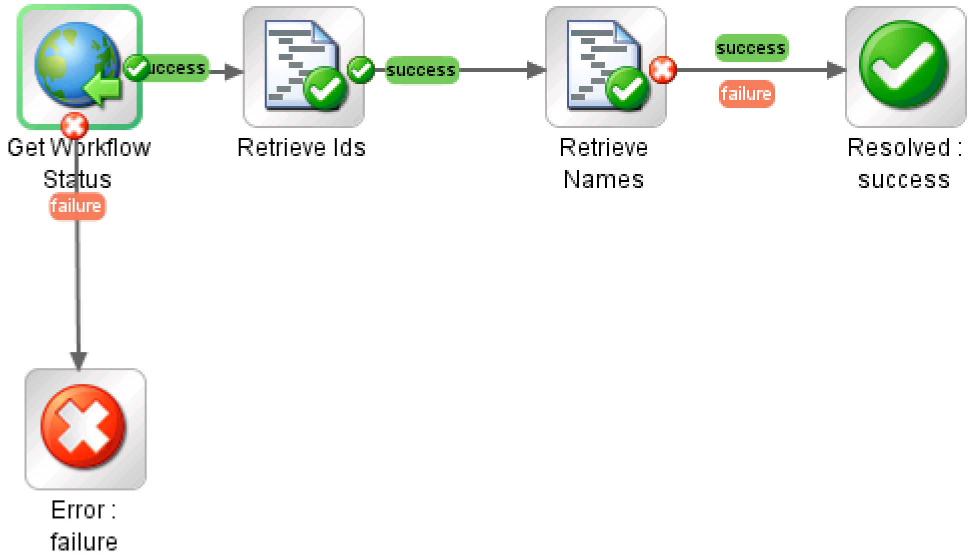
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Get Deployments

The Get Deployments example flow returns a comma-separated list of all the deployments that have been created on the specified DMA server. The flow returns the list of deployments in three different formats: deploymentNames, deploymentUUIDs, and deploymentUrls.



The flow produces a Success response as long as it successfully queries the DMA server using the DMA API. It produces a Failure response if it cannot query the DMA server for any reason.

How to Use this Flow

Use this flow when you want to obtain a list of all deployments defined on the DMA server.

You can use this flow as a step (subflow) in a larger flow that uses the "Operation: Execute Deployment" operation to run an DMA workflow using a specific deployment.

Note that this flow is used in the "Example Flow: Execute Deployments Interactively" example flow.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that returns the list of deployments currently defined on the specified DMA server.
- Use XPath queries to retrieve the URLs and names of the deployments from the response body of the GET operation.
- Use a scriptlet filter to extract the deployment UUIDs from the URLs.

Permissions and Constraints

Only those deployments for which the specified dmaUser has a role with READ permission will be included in the list. Users who have a role with Administrator capability can list all deployments.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to list the deployments.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
deploymentNames	Provision Oracle Software Linux 6.x 001, Provision WebSphere 8 and Depl Mgr 014, Run MS SQL Compliance Audit 005	List of deployments by name.
deploymentUUIDs	90cefae83eeccde1013eed02d3b1006d, 90cefae83ef26cbb013ef28074460002, 90cefae83ef1920b013ef1faf0f406c0	List of deployments by UUID.
deploymentUrls	https://dma1.mycompany.com:8443/dma/api/auto/deployment/90cefae83eeccde1013eed02d3b1006d, https://dma1.mycompany.com:8443/dma/api/auto/deployment/90cefae83ef26cbb013ef28074460002, https://dma1.mycompany.com:8443/dma/api/auto/deployment/90cefae83ef1920b013ef1faf0f406c0	List of deployments by URL.

How it Works

1. The Get Workflow Status step uses the DMA API to perform the following GET request:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/auto/deployment

   The GET request returns an XML feed that looks like this:

   ```
   <feed>
   ```

```
xmlns="http://www.w3.org/2005/Atom"
xmlns:sop="http://www.hp.com/dma/api/sop"
xmlns:env="http://www.hp.com/datapal/api/env">
<id>https://dma1.mycompany.com:8443/dma/api/auto/deployment</id>
<author>
    <name>HP DMA</name>
</author>
<updated>2013-05-30T10:12:47Z</updated>
<title>Deployments</title>
<link rel='self' type='application/atom+xml'
    href='https://dma1.mycompany.com:8443/dma/api/auto/deployment'/>
<entry>
    <id>https://dma1.mycompany.com:8443/dma/api/auto/deployment/
        90cefae83eeccde1013eed02d3b1006d</id>
    <published>2013-05-30T10:12:47Z</published>
    <updated>2013-05-28T21:19:45Z</updated>
    <title>Provision Oracle Software Linux 6.x 001</title>
    <link rel='self' type='application/atom+xml'
       href='https://dma1.mycompany.com:8443/dma/api/auto/deployment/
       90cefae83eeccde1013eed02d3b1006d'/>
    <link rel='edit' type='application/atom+xml'
       href='https://dma1.mycompany.com:8443/dma/api/auto/deployment/
       90cefae83eeccde1013eed02d3b1006d'/>
    <link rel='alternate' type='text/html'
       href='https://dma1.mycompany.com:8443/dma/sop/deploy/view/
       90cefae83eeccde1013eed02d3b1006d'/>
    <link rel='parent' type='application/atom+xml'
       href='https://dma1.mycompany.com:8443/dma/api/auto/workflow/
       90cefae83eeccde1013eecfe5b610051'
       title='Provision Oracle Software'/>
    <sop:deployment name='Provision Oracle Software Linux 6.x 001'
         workflow='Provision Oracle Software'/>
</entry>
<entry>
    <id>https://dma1.mycompany.com:8443/dma/api/auto/deployment/
        90cefae83ef26cbb013ef28074460002</id>
    <published>2013-05-30T10:12:47Z</published>
    <updated>2013-05-29T22:55:04Z</updated>
    <title>Provision WebSphere 8 and Depl Mgr 014</title>
    <link rel='self' type='application/atom+xml'
       href='https://dma1.mycompany.com:8443/dma/api/auto/deployment/
       90cefae83ef26cbb013ef28074460002'/>
    <link rel='edit' type='application/atom+xml'
       href='https://dma1.mycompany.com:8443/dma/api/auto/deployment/
       90cefae83ef26cbb013ef28074460002'/>
    <link rel='alternate' type='text/html'
          href='https://dma1.mycompany.com:8443/dma/sop/deploy/view/
          90cefae83ef26cbb013ef28074460002'/>
    <link rel='parent' type='application/atom+xml'
       href='https://dma1.mycompany.com:8443/dma/api/auto/workflow/
       90cefae83ef1920b013ef1f7130006b9'
       title='Provision WebSphere 8 and Deployment Manager'/>
    <sop:deployment name='Provision WebSphere 8 and Depl Mgr 014'
       workflow='Provision WebSphere 8 and Deployment Manager' />
</entry>
<entry>
    <id>https://dma1.mycompany.com:8443/dma/api/auto/deployment/
        90cefae83ef1920b013ef1faf0f406c0</id>
        <published>2013-05-30T10:12:47Z</published>
        <updated>2013-05-29T20:29:14Z</updated>
        <title>Run MS SQL Compliance Audit 005</title>
        <link rel='self' type='application/atom+xml'
           href='https://dma1.mycompany.com:8443/dma/api/auto/deployment/
           90cefae83ef1920b013ef1faf0f406c0'/>
        <link rel='edit' type='application/atom+xml'
           href='https://dma1.mycompany.com:8443/dma/api/auto/deployment/
           90cefae83ef1920b013ef1faf0f406c0'/>
        <link rel='alternate' type='text/html'
           href='https://dma1.mycompany.com:8443/dma/sop/deploy/view/
           90cefae83ef1920b013ef1faf0f406c0'/>
        <link rel='parent' type='application/atom+xml'
           href='https://dma1.mycompany.com:8443/dma/api/auto/workflow/
           90cefae83ef1920b013ef1f7130006b9'
           title='Run MS SQL Compliance Audit'/>
        <sop:deployment name='Run MS SQL Compliance Audit 005'
            workflow='Run MS SQL Compliance Audit'/>
    </entry>
</feed>
```

If the GET request does not return valid XML, the flow terminates with a Failure response.

2. The Retrieve IDs step uses the following XPath query to extract the deploymentUrls (highlighted above) from the XML feed:

   `/*[local-name()='feed']/*[local-name()='entry']/*[local-name()='id']`

   It then uses the following scriptlet filter to extract the deploymentUUIDs from the deploymentUrls:

   `scriptletResult = scriptletInput.match(/[a-z0-9]{32}/g,'')`

3. The Retrieve Names step uses the following XPath query to extract the deploymentNames from the XML feed:

   `/feed/entry/title/text()`

4. If the flow has reached this point, it terminates with a Success response.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

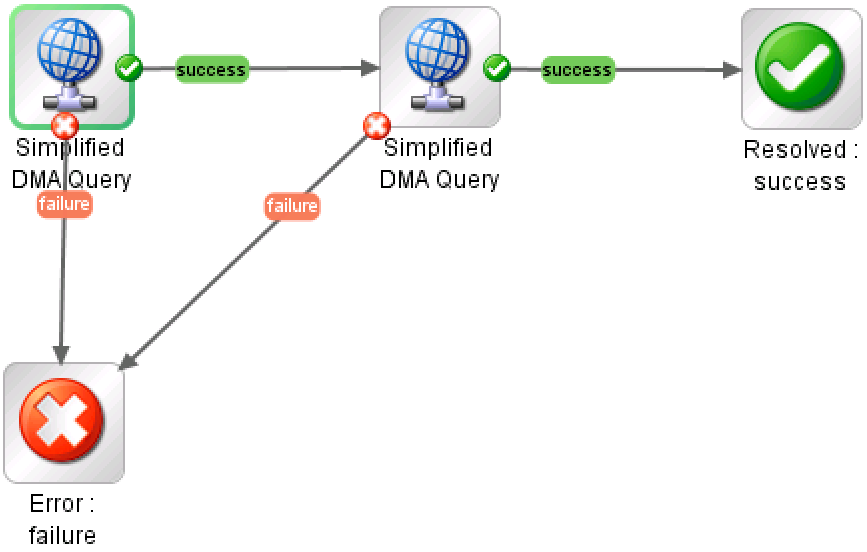
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Get Policies

The Get Policies example flow returns a list of all the policies that have been created on the specified DMA server. The flow returns the list of policies in three different formats: policyNames, policyUUIDs, and policyUrls.



The flow produces a Success response if it successfully retrieves and parses the list of policies. It produces a Failure response if it cannot retrieve or parse the list for any reason.

How to Use this Flow

Use this flow when you want to obtain a list of all policies defined on the DMA server.

You can use this flow as a step (subflow) in a larger flow that uses the "Operation: Execute Deployment" operation to run an DMA workflow.

Note that this flow contains two instances of the "Operation: Simplified DMA Query" operation.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that returns the list of policies currently defined on the specified DMA server.
- Use XPath queries to retrieve the URLs and names of the policies from the response body of the GET operation.
- Use a scriptlet filter to extract the policy UUIDs from the URLs.

Permissions and Constraints

Only those policies for which the specified dmaUser has a role with READ permission will be included in the list. Users who have a role with Administrator capability can list all policies.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to list the policies.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
policyNames	Discovery|MS SQL: Authentication|MS SQL: Patching	List of policies by name.
policyUUIDs	90cefae83eeccde1013eed09ede20375| 90cefae83eeccde1013eed0d1c31154e| 90cefae83eeccde1013eed0d1c091544	List of policies by UUID.
policyUrls	https://dma1.mycompany.com:8443/dma/api/ auto/policy/90cefae83eeccde1013eed09ede20375| https://dma1.mycompany.com:8443/dma/api/ auto/policy/90cefae83eeccde1013eed0d1c31154e| https://dma1.mycompany.com:8443/dma/api/ auto/policy/90cefae83eeccde1013eed0d1c091544	List of policies by URL.

How it Works

1. The first "Operation: Simplified DMA Query" step uses the DMA API to perform the following GET request:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/auto/policy

   The GET request returns an XML feed that looks like this:

   ```
   <feed xmlns="http://www.w3.org/2005/Atom"
         xmlns:sop="http://www.hp.com/dma/api/sop">
   ```

```
<id>https://dma1.mycompany.com:8443/dma/api/auto/policy</id>
<author>
   <name>HP DMA</name>
</author>
<updated>2013-05-30T11:10:12Z</updated>
<title>Policies</title>
<link rel='self' type='application/atom+xml'
   href='https://dma1.mycompany.com:8443/dma/api/auto/policy'/>
<entry>
   <id>https://dma1.mycompany.com:8443/dma/api/auto/policy/
       90cefae83eeccde1013eed09ede20375</id>
   <published>2013-05-30T11:10:12Z</published>
   <updated>2013-05-28T21:27:31Z</updated>
   <title>Discovery</title>
   <link rel='self' type='application/atom+xml'
      href='https://dma1.mycompany.com:8443/dma/api/auto/policy/
      90cefae83eeccde1013eed09ede20375'/>
   <link rel='edit' type='application/atom+xml'
      href='https://dma1.mycompany.com:8443/dma/api/auto/policy/
      90cefae83eeccde1013eed09ede20375'/>
   <link rel='alternate' type='text/html'
      href='https://dma1.mycompany.com:8443/dma/sop/policy/view/
      90cefae83eeccde1013eed09ede20375'/>
   <sop:policy name='Discovery' locked='true'/>
</entry>
<entry>
   <id>https://dma1.mycompany.com:8443/dma/api/auto/policy/
       90cefae83eeccde1013eed0d1c31154e</id>
   <published>2013-05-30T11:10:12Z</published>
   <updated>2013-05-28T21:30:59Z</updated>
   <title>MS SQL: Authentication</title>
   <link rel='self' type='application/atom+xml'
      href='https://dma1.mycompany.com:8443/dma/api/auto/policy/
      90cefae83eeccde1013eed0d1c31154e'/>
   <link rel='edit' type='application/atom+xml'
      href='https://dma1.mycompany.com:8443/dma/api/auto/policy/
      90cefae83eeccde1013eed0d1c31154e'/>
   <link rel='alternate' type='text/html'
      href='https://dma1.mycompany.com:8443/dma/sop/policy/view/
      90cefae83eeccde1013eed0d1c31154e'/>
   <sop:policy name='MS SQL: Authentication' locked='true'/>
</entry>
<entry>
   <id>https://dma1.mycompany.com:8443/dma/api/auto/policy/
       90cefae83eeccde1013eed0d1c091544</id>
   <published>2013-05-30T11:10:12Z</published>
   <updated>2013-05-28T21:30:59Z</updated>
   <title>MS SQL: Patching</title>
   <link rel='self' type='application/atom+xml'
      href='https://dma1.mycompany.com:8443/dma/api/auto/policy/
      90cefae83eeccde1013eed0d1c091544'/>
   <link rel='edit' type='application/atom+xml'
      href='https://dma1.mycompany.com:8443/dma/api/auto/policy/
      90cefae83eeccde1013eed0d1c091544'/>
   <link rel='alternate' type='text/html'
      href='https://dma1.mycompany.com:8443/dma/sop/policy/view/
      90cefae83eeccde1013eed0d1c091544'/>
   <sop:policy name='MS SQL: Patching' locked='true'/>
   </entry>
</feed>
```

The step uses the following XPath query to extract the policyUrls (highlighted above) from the XML feed:

`/feed/entry/id/text()`

It then uses the following scriptlet filter to extract the policyIds from the policyUrls:

`scriptletResult = scriptletInput.match(/[a-z0-9]{32}/g,'')`

If the GET request does not return valid XML, the flow terminates with a Failure response.

2. The second "Operation: Simplified DMA Query" step uses the DMA API to perform the same GET request, and then it uses this XPath query to extract the policyNames from the XML:

   `//sop:policy/@name`

   If this GET request does not return valid XML, the flow terminates with a Failure response.

3. If the two "Operation: Simplified DMA Query" steps successfully retrieve and extracts the list of policies in the three different formats, the flow terminates with a Success response.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

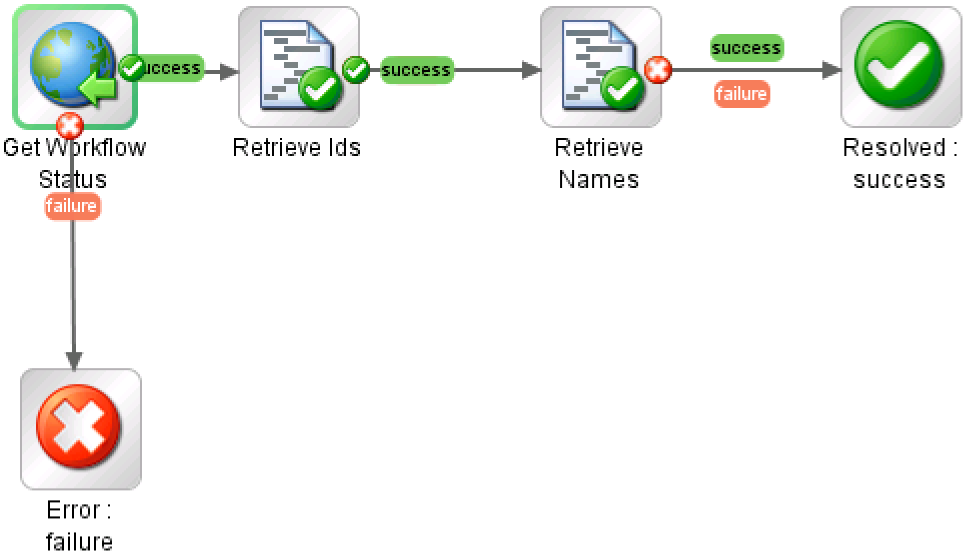
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Get Running Workflows

The Get Running Workflows example flow returns a comma-separated list of all the workflows that are currently running (or have completed in the last 60 seconds) on the specified DMA server. The flow returns the list of workflow executions in three different formats: workflowNames, workflowUUIDs, and workflowUrls.



The flow produces a Success response as long as it successfully queries the DMA server using the DMA API. It produces a Failure response if it cannot query the DMA server for any reason.

How to Use this Flow

Use this flow when you want to obtain a list of all workflows currently running on an DMA server.

You can use this flow as a step (subflow) in a larger flow that uses the "Example Flow: Get Workflow Run Status" flow to retrieve detailed information about a particular workflow execution..

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that returns the list of workflows currently running on the specified DMA server.
- Use XPath queries to retrieve the URLs and names of the workflow executions from the response body of the GET operation.
- Use a scriptlet filter to extract the workflow execution UUIDs from the URLs.

Permissions and Constraints

To view (GET) information about a workflow execution, the specified dmaUser must have a role with READ permission on the organization were the targets reside or EXECUTE permission on the pertinent deployment.

Users who have a role with Administrator capability can list all workflow executions.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to list the workflows.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
workflowNames	Copy of Discovery,Copy of Provision Oracle Software,Copy of Create Sybase Database	List of running workflows by name.
workflowUUIDs	90cefae83ef26cbb013ef55a902e001f, 90cefae83ef26cbb013ef55a94d90048, 90cefae83ef26cbb013ef55a989b0070	List of running workflows by UUID.
workflowUrls	https://dma1.mycompany.com:8443/dma/api/auto/running/workflow/90cefae83ef26cbb013ef55a902e001f|https://dma1.mycompany.com:8443/dma/api/auto/running/workflow/90cefae83ef26cbb013ef55a94d90048|https://dma1.mycompany.com:8443/dma/api/auto/running/workflow/90cefae83ef26cbb013ef55a989b0070	List of running workflows by URL.

How it Works

1. The Get Workflow Status step uses the DMA API to perform the following GET request:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/auto/running

   The GET request returns an XML feed that looks like this:

   ```
   <feed>
   ```

```
    xmlns='http://www.w3.org/2005/Atom'
    xmlns:sop='http://www.hp.com/dma/api/sop'>
    <id>https://dma1.mycompany.com:8443/dma/api/auto/running</id>
    <author>
        <name>HP DMA</name>
    </author>
    <updated>2013-05-30T12:14:26Z</updated>
    <title>Active Workflows</title>
    <link rel='self' type='application/atom+xml'
       href='https://dma1.mycompany.com:8443/dma/api/auto/running' />
    <entry>
       <id>https://dma1.mycompany.com:8443/dma/api/auto/running/workflow/
           90cefae83ef26cbb013ef55a902e001f</id>
       <published>2013-05-30T12:12:33Z</published>
       <updated>2013-05-30T12:12:33Z</updated>
       <title>Copy of Discovery</title>
       <link rel='alternate' type='application/atom+xml'
          href='https://dma1.mycompany.com:8443/dma/api/auto/running/
          workflow/90cefae83ef26cbb013ef55a902e001f' />
       <link rel='alternate' type='text/html'
          href='https://dma1.mycompany.com:8443/dma/sop/workflow/view/
          90cefae83eeccde1013eed0a873304c2' />
       <sop:target server='target1' />
       <sop:status state='Running'/>
       <sop:deployment name='Full Discovery Deployment'/>
    </entry>
    <entry>
       <id>https://dma1.mycompany.com:8443/dma/api/auto/running/workflow/
           90cefae83ef26cbb013ef55a94d90048</id>
       <published>2013-05-30T12:12:34Z</published>
       <updated>2013-05-30T12:12:34Z</updated>
       <title>Copy of Provision Oracle Software</title>
       <link rel='alternate' type='application/atom+xml'
          href='https://dma1.mycompany.com:8443/dma/api/auto/running/
          workflow/90cefae83ef26cbb013ef55a94d90048' />
       <link rel='alternate' type='text/html'
          href='https://dma1.mycompany.com:8443/dma/sop/workflow/view/
          90cefae83eeccde1013eed0a873304c2' />
       <sop:target server='target2' />
       <sop:status state='Running'/>
       <sop:deployment name='Prov Oracle Software Linux 004'/>
    </entry>
    <entry>
       <id>https://dma1.mycompany.com:8443/dma/api/auto/running/workflow/
           90cefae83ef26cbb013ef55a989b0070</id>
       <published>2013-05-30T12:12:35Z</published>
       <updated>2013-05-30T12:12:35Z</updated>
       <title>Copy of Create Sybase Database</title>
       <link rel='alternate' type='application/atom+xml'
          href='https://dma1.mycompany.com:8443/dma/api/auto/running/
          workflow/90cefae83ef26cbb013ef55a989b0070' />
       <link rel='alternate' type='text/html'
          href='https://dma1.mycompany.com:8443/dma/sop/workflow/view/
          90cefae83eeccde1013eed0a873304c2' />
       <sop:target server='target3' />
       <sop:status state='Failed'/>
       <sop:deployment name='Create Sybase DB 014'/>
    </entry>
</feed>
```

If the GET request does not return valid XML, the flow terminates with a Failure response.

2. The Retrieve IDs step uses the following XPath query to extract the workflowUrls (highlighted above) from the XML feed:

   ```
   /*[local-name()='feed']/*[local-name()='entry']/*[local-name()='id']
   ```

   It then uses the following scriptlet filter to extract the workflowUUIDs from the workflowUrls:

   ```
   scriptletResult = scriptletInput.match(/[a-z0-9]{32}/g,'')
   ```

3. The Retrieve Names step uses the following XPath query to extract the workflowNames from the XML feed:

   ```
   /*[local-name()='feed']/*[local-name()='entry']/*[local-name()='title']
   ```

4. If the flow has reached this point, it terminates with a Success response.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

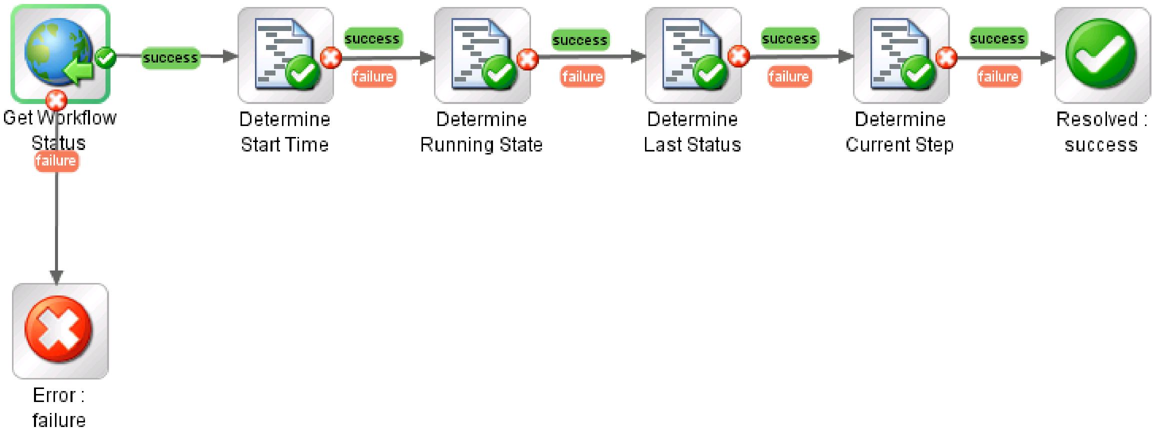
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Get Workflow Run Status

The Get Workflow Run Status example flow returns information about a particular workflow execution on the specified DMA server.



The flow produces a Success response as long as the DMA API returns information about the specified workflow execution. It produces a Failure response if the API does not return valid information.

The possible DMA workflow execution states are:

State	Description
RUNNING	The workflow is currently running on the specified target.
SUCCESS	The workflow has finished running, and the final step executed was the built-in Success step.
FAILED	The workflow has finished running, and the final step executed was the built-in Failure step. This usually indicates that an earlier step in the workflow failed to accomplish its objective.
FINISHED	The workflow ran to completion, but the final step executed was neither the built-in Success step nor the built-in Failure step.
SKIPPED	Pertains only to scheduled deployments. If a workflow is already running at the scheduled execution time, DMA will not attempt to start the workflow again. The scheduled deployment will be marked SKIPPED on the History page.
CANCELLED	An DMA user cancelled the workflow execution.
SKIPPED	This status pertains only to scheduled deployments. It indicates that the workflow was scheduled to run, but the execution was skipped. This would happen, for example, or if a previous execution of this workflow were still running on that target.
ABORTED	DMA was unable to run the workflow. This usually means that the target server was not configured to run workflows, or a connectivity problem arose between either the DMA server or the SA server and one or more of the target servers.

How to Use this Flow

Use this flow when you want to know detailed information about a specific workflow execution and you have its UUID.

You can use the "Example Flow: Get Running Workflows" example flow to get the UUID for any current workflow execution that the specified dmaUser has permission to view.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that searches for a specific workflow execution.
- Use XPath queries to retrieve information about the workflow execution from the response body of the GET operation.
- Use a scriptlet filter to compute the value of a result.

Permissions and Constraints

To view (GET) information about a workflow execution, the specified dmaUser must have a role with READ permission on the organization were the targets reside or EXECUTE permission on the pertinent deployment. Users who have a role with Administrator capability can list all workflow executions.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to carry out the requested operation.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
executionUUID	The universally unique identifier (UUID) of the workflow execution.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
lastStatusUpdate	2013-05-30T14:45:35Z	Time when the last step to execute was started.
currentRuntime	00:01:43	Elapsed time that the workflow has been running (in HH:MM:SS format). More precisely, the elapsed time between the startTime and the lastStatusUpdate.
currentStepName	Discovery Parameters	The name of the step that was initiated or running when the flow performed the GET request.
executionStatus	Running	The current status of the workflow execution: Initiated, Running, Finished, Success, Failure, Cancelled, Aborted, or Skipped See the *DMA User Guide* for definitions of these workflow execution states.
startTime	2013-05-30T14:44:31Z	Date and time that the workflow execution was initiated.

How it Works

1. The Get Workflow Status step uses the DMA API to perform a GET request using the specified organizationName and serverName:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/auto/running/<*executionUUID*>

   The GET request returns an XML feed that looks like the following example. Each `<entry>` element represents a step in the workflow that has been executed or is currently running. The `rc` attribute contains the return code for those steps that have finished. The `rc` attribute for the step that is currently running is empty.

```
<feed>
    xmlns='http://www.w3.org/2005/Atom'
    xmlns:sop='http://www.hp.com/dma/api/sop'>
    <id>https://dma1.mycompany.com:8443/dma/api/auto/running/workflow/
        90cefae83ef26cbb013ef5e5b19e0de0</id>
    <author>
        <name>HP DMA</name>
    </author>
    <updated>2013-05-30T14:44:31Z</updated>
    <title>Copy of Discovery</title>
    <link rel='self' type='application/atom+xml'
       href='https://dma1.mycompany.com:8443/dma/api/auto/running/
       workflow/90cefae83ef26cbb013ef5e5b19e0de0' />
    <link rel='alternate' type='text/html'
       href='https://dma1.mycompany.com:8443/dma/sop/workflow/view/
       90cefae83eeccde1013eed0a873304c2' />
    <entry>
        <id>https://dma1.mycompany.com:8443/dma/api/auto/running/
            workflow/90cefae83ef26cbb013ef5e5b19e0de0/step/
            90cefae83ef26cbb013ef5e5b1c80de6</id>
        <published>2013-05-30T14:44:31Z</published>
        <updated>2013-05-30T14:44:56Z</updated>
        <title>Discovery Parameters</title>
        <link rel='alternate' type='application/atom+xml'
           href='https://dma1.mycompany.com:8443/dma/api/auto/running/
           workflow/90cefae83ef26cbb013ef5e5b19e0de0/step/
           90cefae83ef26cbb013ef5e5b1c80de6' />
        <link rel='alternate' type='text/html'
           href='https://dma1.mycompany.com:8443/dma/sop/step/view/
           90cefae83eeccde1013eed09eb430313' />
        <sop:status state='Finished' rc='0'/>
    </entry>
    <entry>
        <id>https://dma1.mycompany.com:8443/dma/api/auto/running/
            workflow/90cefae83ef26cbb013ef5e5b19e0de0/step/
            90cefae83ef26cbb013ef5e6133a0e0d</id>
        <published>2013-05-30T14:44:56Z</published>
        <updated>2013-05-30T14:45:15Z</updated>
        <title>Discover WebSphere</title>
        <link rel='alternate' type='application/atom+xml'
           href='https://dma1.mycompany.com:8443/dma/api/auto/running/
           workflow/90cefae83ef26cbb013ef5e5b19e0de0/step/
           90cefae83ef26cbb013ef5e6133a0e0d' />
        <link rel='alternate' type='text/html'
           href='https://dma1.mycompany.com:8443/dma/sop/step/view/
           90cefae83eeccde1013eed09ec91033f' />
        <sop:status state='Finished' rc='1'/>
    </entry>
    <entry>
        <id>https://dma1.mycompany.com:8443/dma/api/auto/running/
            workflow/90cefae83ef26cbb013ef5e5b19e0de0/step/
            90cefae83ef26cbb013ef5e6604f0e78</id>
        <published>2013-05-30T14:45:15Z</published>
        <updated>2013-05-30T14:45:34Z</updated>
        <title>Discover Oracle Databases</title>
        <link rel='alternate' type='application/atom+xml'
           href='https://dma1.mycompany.com:8443/dma/api/auto/running/
           workflow/90cefae83ef26cbb013ef5e5b19e0de0/step/
           90cefae83ef26cbb013ef5e6604f0e78' />
        <link rel='alternate' type='text/html'
           href='https://dma1.mycompany.com:8443/dma/sop/step/view/
           90cefae83eeccde1013eed09e0a50206' />
        <sop:status state='Finished' rc='1'/>
    </entry>
    <entry>
        <id>https://dma1.mycompany.com:8443/dma/api/auto/running/
            workflow/90cefae83ef26cbb013ef5e5b19e0de0/step/
            90cefae83ef26cbb013ef5e6a9880ee4</id>
        <published>2013-05-30T14:45:34Z</published>
        <updated>2013-05-30T14:45:35Z</updated>
        <title>Discover SQL Databases</title>
        <link rel='alternate' type='application/atom+xml'
           href='https://dma1.mycompany.com:8443/dma/api/auto/running/
           workflow/90cefae83ef26cbb013ef5e5b19e0de0/step/
           90cefae83ef26cbb013ef5e6a9880ee4' />
        <link rel='alternate' type='text/html'
           href='https://dma1.mycompany.com:8443/dma/sop/step/view/
           90cefae83eeccde1013eed09e8e402d5' />
        <sop:status state='Running' rc=''/>
    </entry>
    <sop:target server='target5' />
    <sop:script-execution url='https://dma1.mycompany.com:8443/dma/api/
         auto/running/workflow/90cefae83ef26cbb013ef5e5b19e0de0/
         scriptExecution/90cefae83ef26cbb013ef5e5b3b90e08'/>
    <sop:status state='Running'/>
    <sop:deployment name='Full Discovery Deployment'/>
</feed>
```

If the GET request does not return valid XML, the flow terminates with a Failure response.

2. If the GET request successfully returns information about the specified workflow execution, the subsequent steps use the following XPath queries (or filters) to extract information from the XML response body returned by the GET:

Step Name	Result	XPath Query and Filter
Determine Start Time State	startTime	/*[local-name()='feed']/*[local-name()='updated']
Determine Running State	executionStatus	/*[local-name()='feed']/*[local-name()='status']/@state
Determine Last Status Update Time	lastStatusUpdate	/*[local-name()='feed']/*[local-name()='entry' and last()]/*[local-name()='updated'] RegEx filter: '[^\,]*$'

Determine Current Step	currentStepName
	currentRuntime
		/*[local-name()='feed']/*[local-name()='entry'][*[local-name()='status'][@*[local-name()='state'] = 'Running' or @*[local-name()='state'] = 'Initiated']]/*[local-name()='title']
		Scriptlet used to compute currentRuntime:
		`var jsStartTime = ('' + startTime).replace(/-/g,"/").replace(/[TZ]/g," ");`
		`var jsLastUpdateTime = ('' + lastStatusUpdate).replace(/-/g,"/").replace(/[TZ]/g," ");`
		
		`var startDate = new Date(jsStartTime);`
		`var lastDate = new Date(jsLastUpdateTime);`
		
		`var milliseconds = lastDate.getTime() - startDate.getTime();`
		`var totalSeconds = milliseconds / 1000;`
		`var hours = parseInt( totalSeconds / 3600 ) % 24;`
		`var minutes = parseInt( totalSeconds / 60 ) % 60;`
		`var seconds = totalSeconds % 60;`
		
		`currentRuntime = (hours < 10 ? "0" + hours : hours) + ":" + (minutes < 10 ? "0" + minutes : minutes) + ":" + (seconds < 10 ? "0" + seconds : seconds);`
		
		`scriptletContext.putGlobal("currentRuntime",`

Step Name	Result	XPath Query and Filter

		`currentRuntime);`
		`scriptletResult = currentRuntime;`

3. If the flow reaches this point, it terminates with a Success response.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

"Example Environment Flows" on the next page

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Environment Flows

The DMA content pack includes the following example environment flows:

Flow Name	Description
"Example Flow: Add Organization"	Creates a new organization in the DMA environment.
"Example Flow: Add Server to Organization"	Adds a server to a new or existing organization in the DMA environment.
"Example Flow: Delete Server From Organization"	Deletes the specified server from the specified organization in the DMA environment.
"Example Flow: Get Organization Id"	Returns the UUID of the specified organization in the DMA environment.
"Example Flow: Get Server Custom Field"	Returns the value of the specified server Custom Field.
"Example Flow: Get Server Id"	Returns the UUID of the specified server.
"Example Flow: Get Unassigned Servers"	Returns a list of servers that are available to add to DMA organizations.
"Example Flow: Put Server Custom Field"	Modifies the value of the specified server Custom Field.

See Also:

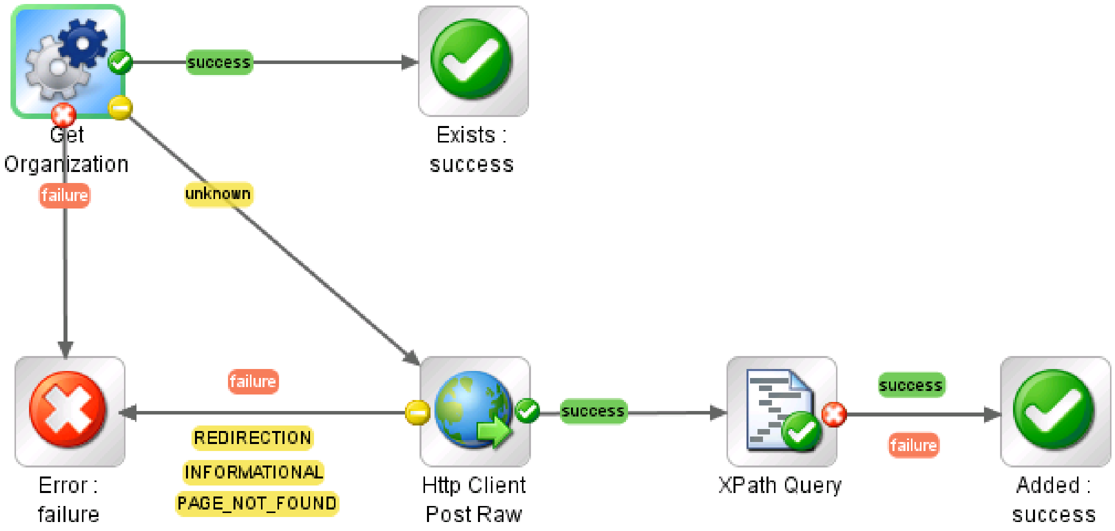
"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

"HPE DMA Wizard for HP OO" on page 71

Example Flow: Add Organization

The Add Organization example flow creates a new organization in the DMA environment. It first searches the DMA environment to determine whether the organization already exists. If the organization does not exist, it adds a new organization to the environment. The flow returns the UUID of the new or existing organization and the URL that you can use to access the organization using the DMA API.



The flow produces a Success response if it successfully adds the new organization or the organization already exists. The flow produces a Failure response if it is unable to either find or add the organization.

The flow also produces a Failure response if it is unable to contact the specified DMA server for any reason (for example: incorrect DMA server host name or IP address, incorrect user name or password, or a network connectivity problem).

How to Use this Flow

Use this flow when you want to create a new organization in your DMA environment.

You can use this flow in combination with the "Example Flow: Add Server to Organization" flow to populate your DMA environment.

After you add one or more servers to an organization, you can use the "Example Automation Flows" to run and monitor DMA workflows in your environment.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that searches for an existing organization in the DMA environment by name and returns information about that organization, including its UUID.
- Invoke the DMA API to perform a POST operation that creates a new organization in the DMA environment.

- Use XPath queries to retrieve the UUID and URL of the organization from the response body of the GET or POST operation.

Permissions and Constraints

The following constraints determine what you can accomplish with this flow:

- The specified dmaUser must have a role with Administrator capability.
- Organization names must be unique.

When you create (POST) an organization using this flow, no roles have permission to READ, WRITE, or DEPLOY to the new organization. You can set the permissions by performing a PUT operation on the organization (or by using the DMA user interface).

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has privileges required to add a new organization or get information about an existing organization.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
organizationName	Name of the organization that will be retrieved from or created in the DMA environment.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
organizationUUID	8f82f39d2e084d0b012e0a28bcfa0000	The universally unique identifier (UUID) of the organization. This is a 128-bit number (32 hexadecimal characters) that is unique within the DMA environment.
organizationUrl	https://dma1.mycompany.com:8443/dma/api/env/organization/8f82f39d2e084d0b012e0a28bcfa0000	URL of the organization. This URL can directly be used directly in subsequent DMA REST API calls (for example: a GET).

How it Works

1. The Get Organization Id step uses the DMA API to perform a GET request using the specified organizationName:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/organization?orgName=<*organizationName*>

   If the GET request fails for any reason, the flow terminates with a Failure reponse.
2. If an organization with that name exists in the DMA environment, the XPath Query step extracts the UUID of that organization from the XML response body returned by the GET:

   /*[local-name()='entry']/*[local-name()='id']

   The flow then terminates with a Success response.
3. If an organization with that name does not exist, the flow attempts to create it by performing a POST request:

   POST https://<*dmaServer*>:<*dmaPort*>/dma/api/env/organization

   Example of the request body (payload) for the POST:

   ```
   <entry xmlns="http://www.w3.org/2005/Atom"
   xmlns:env="http://www.hp.com/datapal/api/env">
      <title>NewOrg</title>
      <env:organization name='NewOrg'>
         <env:permissions/>
      </env:organization>
   </entry>
   ```

   If the POST succeeds, the XPath Query step extracts the UUID of the new organization from the response body returned by the POST. The flow then terminates with a Success response.

   If the POST fails, the flow terminates with a Failure response.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

"Example Environment Flows" on page 37

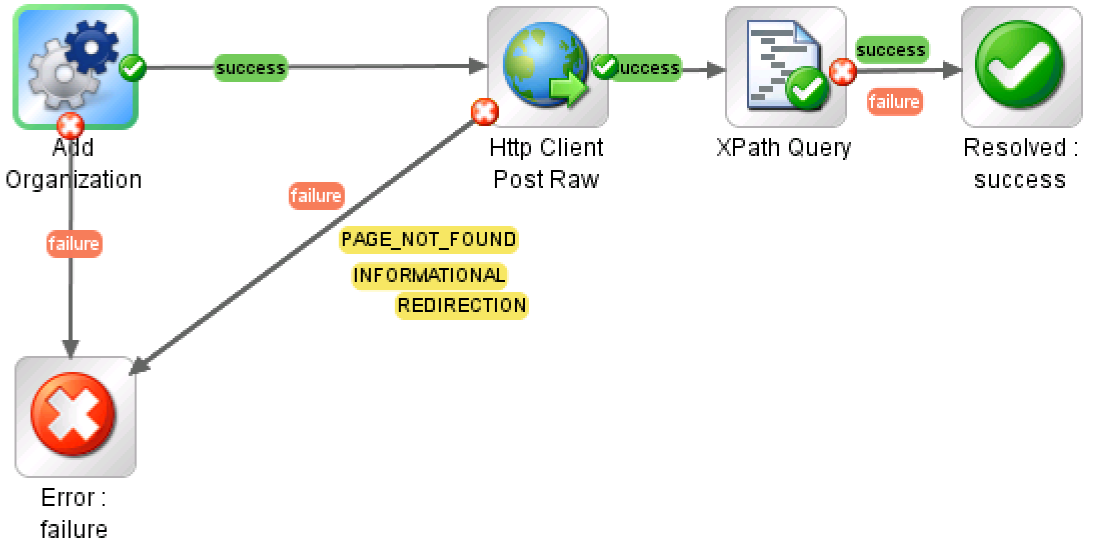
"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Add Server to Organization

The Add Server to Organization example flow adds a server to an organization in the DMA environment. It first searches the DMA environment to determine whether the specified organization already exists. If the organization does not exist, it adds a new organization to the environment. It then attempts to add the new server to the organization.

If it successfully adds the server, the flow returns the UUID of the new server and the URL that you can use to access the server using the DMA API.



The flow produces a Success response if it successfully adds the new server. The flow produces a Failure response under the following conditions:

- It is unable to either find or add the specified organization.
- It is unable to either add the specified server.
- The specified server is already included in an existing organization.
- The specified server is not included in the unassigned servers pool.
- The specified dmaUser does not have WRITE permission for the specified organization.

The flow also produces a Failure response if it is unable to contact the specified DMA server for any reason (for example: incorrect DMA server host name or IP address, incorrect user name or password, or a network connectivity problem).

How to Use this Flow

Use this flow when you want to add a server to an organization in your DMA environment.

To get a list of servers that can be added to organizations, use the "Example Flow: Get Unassigned Servers" flow.

To delete a server in an existing organization, use the "Example Flow: Delete Server From Organization" flow.

After you add one or more servers to an organization, you can use the "Example Automation Flows" to run and monitor DMA workflows in your environment.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that searches for an existing organization in the DMA environment by name and returns information about that organization, including its UUID.
- Invoke the DMA API to perform a POST operation that creates a new organization in the DMA environment.
- Use XPath queries to retrieve the UUID and URL of the organization or server from the response body of the GET or POST operation.
- Invoke the DMA API to perform a POST operation that adds a server to an existing organization.

Permissions and Constraints

The following constraints determine what you can accomplish with this flow:

- The specified dmaUser must have a role with Administrator capability.
- Organization names must be unique.
- Server names must be unique.
- A server cannot be added to an organization if it is already assigned to another organization. Only unassigned servers can be added (see "Example Flow: Get Unassigned Servers" on page 57).
- A server must already exist in the server management platform before it can be added to an organization. With HP Server Automation (SA), for example, a server must be managed by SA and have the DMA Client Files policy (attached and remediated). Servers that meet these criteria appear in the list of unassigned servers.

When you add (POST) a server to an organization using this flow, no roles have permission to READ, WRITE, or DEPLOY to the new server. You can set the permissions by performing a PUT operation on the organization (or by using the DMA user interface).

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has privileges required to add a new organization or get information about an existing organization.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of

Name	Description
	whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
organizationName	Name of the organization that will be retrieved from or created in the DMA environment.
serverName	Name of the server that will be added to the organization.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
serverUUID	ff8080813eecf0a2013ef02fb1030005	The universally unique identifier (UUID) of the server. This is a 128-bit number (32 hexadecimal characters) that is unique within the DMA environment.
serverUrl	https://dma1.mycompany.com:8443/dma/api/env/server/ff8080813eecf0a2013ef02fb1030005	URL of the server. This URL can directly be used directly in subsequent DMA REST API calls (for example: a GET).

How it Works

1. The Get Organization Id step uses the DMA API to perform a GET request using the specified organizationName:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/organization?orgName=<*organizationName*>

   If the GET request cannot be completed, the flow terminates with a Failure reponse.
2. If an organization with that name exists in the DMA environment, the XPath Query step extracts the UUID of that organization from the XML response body returned by the GET:

   /*[local-name()='entry']/*[local-name()='id']
3. If an organization with that name does not exist, the flow attempts to create it by performing a POST request:

   POST https://<*dmaServer*>:<*dmaPort*>/dma/api/env/organization

   Example of the request body (payload) for the POST:

   ```
   <entry xmlns="http://www.w3.org/2005/Atom"
   xmlns:env="http://www.hp.com/datapal/api/env">
      <title>NewOrg</title>
      <env:organization name='NewOrg'>
         <env:permissions/>
      </env:organization>
   </entry>
   ```

   If the POST succeeds, the XPath Query step extracts the UUID of the new organization from the response body returned by the POST.

If the POST fails, the flow terminates with a Failure response.

4. The flow then attempts to add the specified server to the organization by performing a POST request:

   POST https://<*dmaServer*>:<*dmaPort*>/dma/api/env/server

   Example of the request body (payload) for the POST:

   ```
   <entry xmlns='https://www.w3.org/2005/Atom'
   xmlns:env='https://www.hp.com/datapal/api/env'>
       <link rel='parent' type='application/atom+xml'
   href='https://DMA1.mycompany.com:8443/dma/api/env/organization/8f82f39d2e084d0b01
   2e0a28bcfa0000' />
       <env:server name='target3.mycompany.com'>
       </env:server>
   </entry>
   ```

   If the POST succeeds, the XPath Query step extracts the UUID of the server from the response body returned by the POST.

   If the POST fails, the flow terminates with a Failure response.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

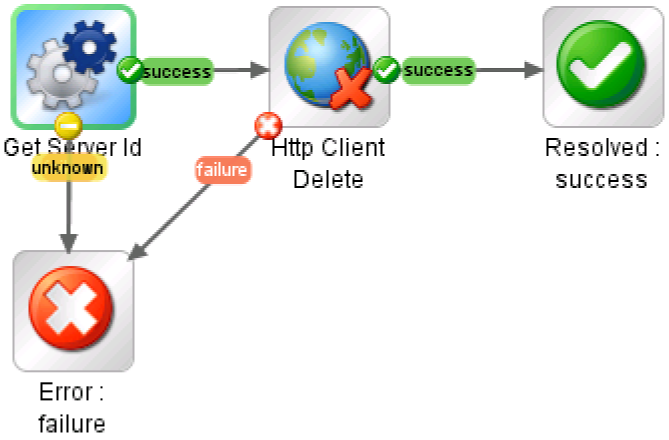
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Delete Server From Organization

The Delete Server From Organization example flow deletes the specified server from the specified organization in the DMA environment. This server then becomes part of the unassigned servers pool and can be added to any organization.



The flow produces a Success response if it successfully deletes the server from the organization. It produces a Failure response under the following conditions:

- It is unable to either find the specified server.
- The specified dmaUser does not have a role with both READ and WRITE permission for the specified organization.
- The specified server contains one or more instances.
- The specified server is currently a target for one or more deployments.

The flow also produces a Failure response if it is unable to contact the specified DMA server for any reason (for example: incorrect DMA server host name or IP address, incorrect user name or password, or a network connectivity problem).

How to Use this Flow

Use this flow when you want to delete a server from an organization and return it to the unassigned servers pool.

Note that the "Example Flow: Get Server Id" example flow is used as a step (subflow) in this flow.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that searches for a specific server in the DMA environment by name and returns the server's UUID.
- Use XPath queries to retrieve the UUID of the server from the response body of the GET.
- Invoke the DMA API to perform a DELETE operation that deletes the server from the specified organization.

Permissions and Constraints

The following constraints determine what you can accomplish with this flow:

- The specified dmaUser must have a role with both READ and WRITE permission for the specified organization.
- You cannot delete a server that contains one or more instances.
- You cannot delete a server that is specified as a target for any deployment.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has privileges required to get information about and make changes to the specified organization.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
organizationName	Name of the organization in which the specified server resides.
serverName	Name of the server that will be deleted from the specified organization.

Results

This flow does not produce results other than the Success or Failure response.

How it Works

1. The Get Server ID step (subflow) retrieves the UUID of the specified server:
   a. The Http Client Detailed Return step uses the DMA API to perform a GET request using the specified organizationName and serverName:

      GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/server?orgName=<*organizationName*>&serverName=<*serverName*>

      If the GET request cannot be completed, the flow terminates with a Failure reponse.

   b. If the GET request successfully returns information about the specified server, the Find Server ID step extracts the UUID of that server from the XML response body returned by the GET:

      /*[local-name()='feed']/*[local-name()='id']

2. The Http Client Delete step uses the DMA API to perform a DELETE request using the the UUID of the specified server.

   If the DELETE request completes successfully (return code is 200), the flow terminates with a Success reponse.

   If the DELETE request cannot be completed, the flow terminates with a Failure reponse.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

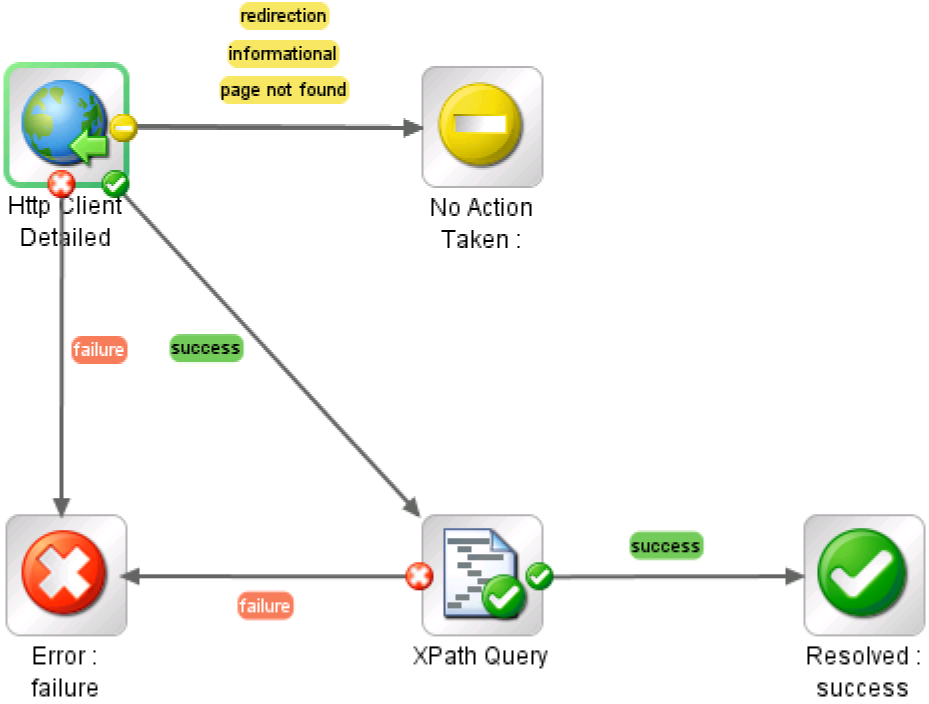
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Get Organization Id

The Get Organization Id example flow returns the UUID of the specified organization in the DMA environment. This is useful if you want to specify this organization's UUID in subsequent steps that access the DMA API.



The flow produces a Success response if it successfully retrieves the organization's UUID.

The flow produces a No Action Taken response under the following conditions:

- It cannot find the specified organization.
- The specified dmaUser does not have READ permission for the specified organization.

The flow produces a Failure response if it cannot retrieve the UUID for any other reason.

How to Use this Flow

Use this flow when you know the name of an organization but you do not know its UUID.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that searches for a specific organization in the DMA environment by name and returns the organization's UUID.
- Use XPath queries to retrieve the UUID of the organization from the response body of the GET operation.

Permissions and Constraints

The specified dmaUser must have a role with READ permission for the specified organization.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has privileges required to get information about an existing organization.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
organizationName	Name of the organization whose UUID will be returned.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
organizationUUID	ff8080813eecf0a2013ef02fb1030005	The universally unique identifier (UUID) of the organization. This is a 128-bit number (32 hexadecimal characters) that is unique within the DMA environment.

How it Works

1. The Http Client Detailed Return step uses the DMA API to perform a GET request using the specified organizationName:

   GET https://<*dmaorganization*>:<*dmaPort*>/dma/api/env/organization?orgName=<*organizationName*>&organizationName=<*organizationName*>

   If the GET request does not return valid XML, the flow terminates with a Failure response.

   If the GET request returns valid XML but does not successfully retrieve information about the organization, the flow terminates with a No Action Taken response.

2. If the GET request successfully returns information about the specified organization, the XPath Query

step extracts the UUID of that organization from the XML response body returned by the GET:

/*[local-name()='feed']/*[local-name()='id']

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

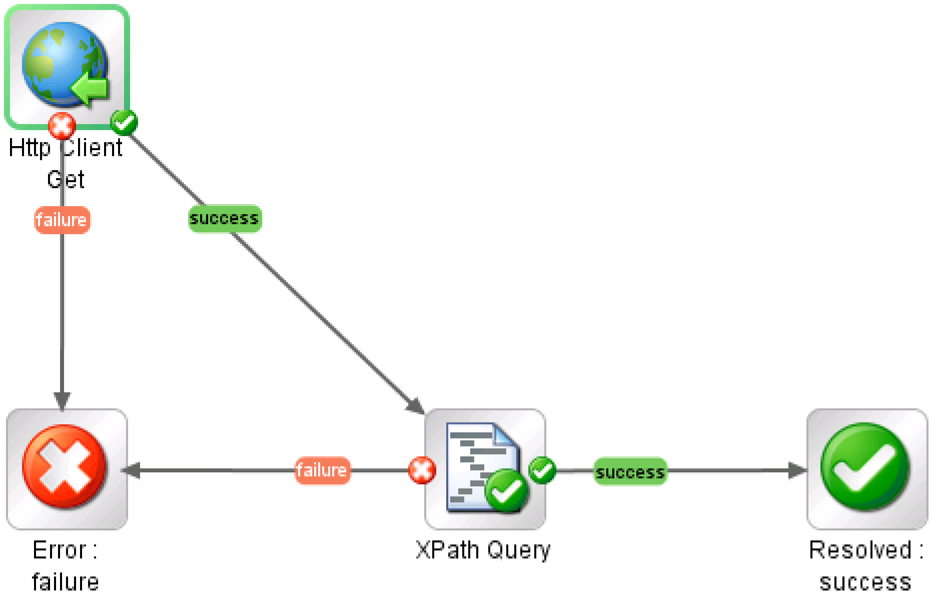
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Get Server Custom Field

The Get Server Custom Field example flow returns the value of the specified server Custom Field.



The flow produces a Success response if it successfully retrieves the value of the Custom Field. It produces a Failure response if it cannot retrieve the Custom Field value for any reason.

How to Use this Flow

Use this flow when you want to know the value of a specific Custom Field for a specific server. You must know the name of the server, the name of the organization in which it resides, and the name of the Custom Field.

You can use the "Example Flow: Put Server Custom Field" flow to change the value of the Custom Field.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that searches for a specific target server in the DMA environment by specifying the name of the server and the name of the organization in which it resides.
- Use an XPath query to retrieve the Custom Field value from the response body of the GET operation.

Permissions and Constraints

The specified dmaUser must have a role with READ permission for the specified organization.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to get information about the specified organization.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
organizationName	Name of the organization in which the specified server resides.
serverName	Name of the server whose Custom Field value will be returned.
customFieldName	Name of the Custom Field whose value will be returned.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
fieldValue	Chicago	The value of the specified server Custom Field.

How it Works

1. The Http Client Get step uses the DMA API to perform a GET request using the specified organizationName and serverName:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/server?orgName=<*organizationName*>&serverName=<*serverName*>

   If the GET request does not find the server, the flow terminates with a Failure response.
2. If the GET request successfully returns information about the specified server, the XPath Query step extracts the Custom Field value from the XML response body returned by the GET:

   //*[local-name()="custom-field" and @name="${customFieldName}"]/@value
3. If the flow finds the server and successfully extracts the value of the Custom Field, the flow terminates with a Success response.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

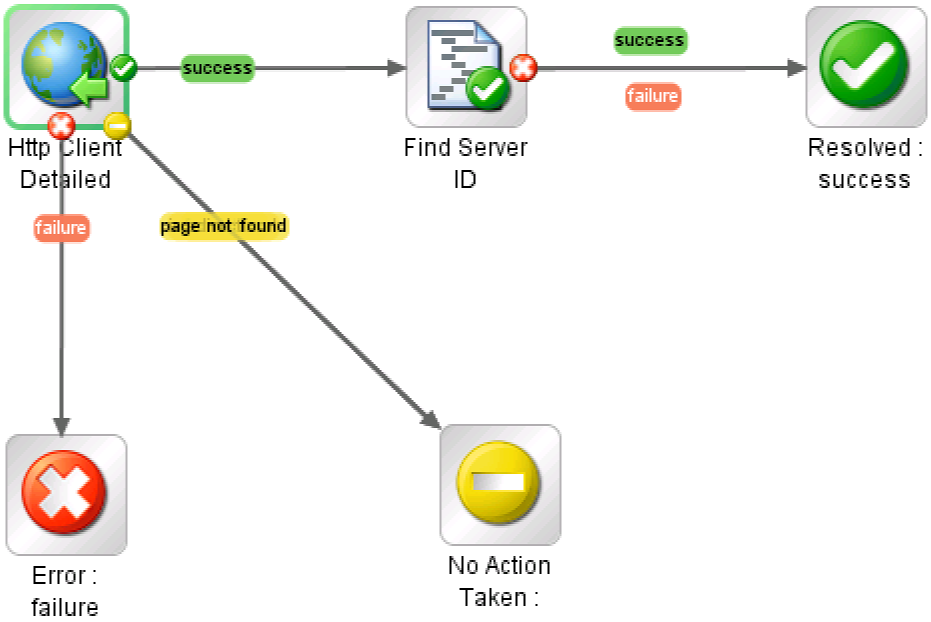
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Get Server Id

The Get Server Id example flow returns the UUID of the specified server. This is useful if you want to specify this server's UUID in subsequent steps that access the DMA API.



The flow produces a Success response if it successfully retrieves the server's UUID.

The flow produces a No Action Taken response under the following conditions:

- It cannot find the specified organization or server.
- The specified dmaUser does not have READ permission for the specified organization.

The flow produces a Failure response if it cannot retrieve the UUID for any other reason.

How to Use this Flow

Use this flow when you know the name of a target server and the organization in which it resides, but you do not know the server's UUID.

The "Example Flow: Put Server Custom Field" and "Example Flow: Delete Server From Organization" flows show you how to use this flow as a step in another flow.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that searches for a specific server in the DMA environment by name and returns the server's UUID.
- Use XPath queries to retrieve the UUID of the server from the response body of the GET operation.

Permissions and Constraints

The specified dmaUser must have a role with READ permission for the specified organization.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has privileges required to get information about an existing organization.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
organizationName	Name of the organization in which the specified server resides.
serverName	Name of the server whose UUID will be returned.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
serverUUID	ff8080813eecf0a2013ef02fb1030005	The universally unique identifier (UUID) of the server. This is a 128-bit number (32 hexadecimal characters) that is unique within the DMA environment.

How it Works

1. The Http Client Detailed Return step uses the DMA API to perform a GET request using the specified organizationName and serverName:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/server?orgName=<*organizationName*>&serverName=<*serverName*>

   If the GET request does not return valid XML, the flow terminates with a Failure response.

   If the GET request returns valid XML but does not successfully retrieve information about the server, the flow terminates with a No Action Taken response.

2. If the GET request successfully returns information about the specified server, the Find Server ID step extracts the UUID of that server from the XML response body returned by the GET:

   /*[local-name()='feed']/*[local-name()='id']

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

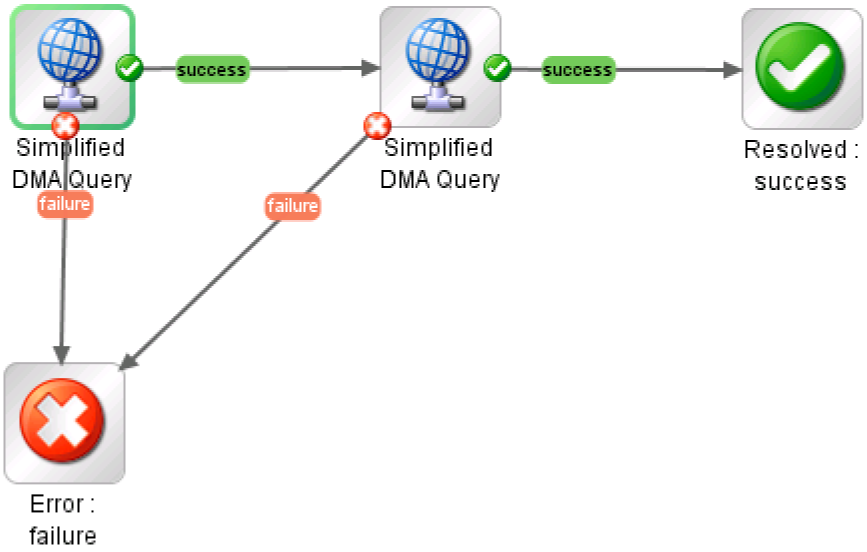
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Get Unassigned Servers

The Get Unassigned Servers example flow returns a list of servers that are available to add to DMA organizations. The flow returns the list of servers in three different formats: serverNames, serverIds, and serverUrls. Only the serverNames are relevant in the DMA context, however.



The flow produces a Success response if it successfully retrieves and parses the list of servers. It produces a Failure response if it cannot retrieve or parse the list for any reason.

How to Use this Flow

Use this flow when you want to find out which managed servers are available to add to DMA organizations.

You can use the "Example Flow: Add Server to Organization" flow to add a single server to an organization. You can also write a flow that iteratively adds available servers to one or more organizations.

Note that this flow contains two instances of the "Operation: Simplified DMA Query" operation.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform a GET operation that returns the list of managed servers that can be added to DMA organizations.
- Use XPath queries to retrieve the URLs and names of the servers from the response body of the GET operation.
- Use a scriptlet filter to extract the server IDs from the URLs.

Permissions and Constraints

The specified dmaUser must have a role with Administrator capability.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has a role with Administrator capability.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.

Results

This flow produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
serverNames	target1|target2|target3	List of unassigned servers by name.
serverIds	70001|10001|30001	List of unassigned servers by SA server ID. Note that these are not UUIDs. They are internal IDs used by SA.
serverUrls	https://dma1.mycompany.com:8443/dma/api/env/unassignedserver/70001|https://dma1.mycompany.com:8443/dma/api/env/unassignedserver/10001|https://dma1.mycompany.com:8443/dma/api/env/unassignedserver/30001	List of unassigned servers by URL.

How it Works

1. The first "Operation: Simplified DMA Query" step uses the DMA API to perform the following GET request:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/unassignedserver

   The GET request returns an XML feed that looks like this:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <feed xmlns="http://www.w3.org/2005/Atom">
   ```

```
<id>https://DMA1.mycompany.com:8443/dma/api/env/unassignedserver</id> <author> <name>HP
DMA</name> </author>
<updated>2013-02-26T12:49:35Z</updated>
<title>Unassigned Servers</title>
<link rel='self' type='application/atom+xml'
href='https://DMA1.mycompany.com:8443/dma/api/env/unassignedserver'/>
<entry>
   <id>
 https://DMA1.mycompany.com:8443/dma/api/env/unassignedserver/70001
   </id>
   <published>2013-02-26T12:49:35Z</published>
   <updated>2013-02-26T12:49:35Z</updated>
   <title>target1</title>
   <link link='self' type='application/atom+xml'
     href='https://DMA1.mycompany.com:8443/dma/api/env/unassignedserver/70001'/>
</entry>
<entry>
   <id>
 https://DMA1.mycompany.com:8443/dma/api/env/unassignedserver/10001
   </id>
   <published>2013-02-26T12:49:35Z</published>
   <updated>2013-02-26T12:49:35Z</updated>
   <title>target2</title>
   <link link='self' type='application/atom+xml'
     href='https://DMA1.mycompany.com:8443/dma/api/env/unassignedserver/10001'/>
</entry>
<entry>
   <id>
 https://DMA1.mycompany.com:8443/dma/api/env/unassignedserver/30001
   </id>
   <published>2013-02-26T12:49:35Z</published>
   <updated>2013-02-26T12:49:35Z</updated>
   <title>target3</title>
   <link link='self' type='application/atom+xml'
     href='https://DMA1.mycompany.com:8443/dma/api/env/unassignedserver/30001'/>
</entry>
</feed>
```

The step uses the following XPath query to extract the serverUrls (highlighted above) from the XML feed:

```
/feed/entry/id/text()
```

It then uses the following scriptlet filter to extract the serverIds from the serverUrls:

```
var matchArray = scriptletInput.match(/\/unassignedserver\/[0-9]{5,}/g);
var partialString = matchArray.toString(); var returnString = scriptletResult =
partialString.replace(/\/unassignedserver\//g,'');
scriptletResult = returnString.replace(/,/g,delimiter);
```

If the GET request does not return valid XML, the flow terminates with a Failure response.

2. The second "Operation: Simplified DMA Query" step uses the DMA API to perform the same GET request, and then it uses this XPath query to extract the serverNames from the XML:

   ```
   /feed/entry/title/text()
   ```

   If this GET request does not return valid XML, the flow terminates with a Failure response.
3. If the two "Operation: Simplified DMA Query" steps successfully retrieve and extracts the list of unassigned servers in the three different formats, the flow terminates with a Success response.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

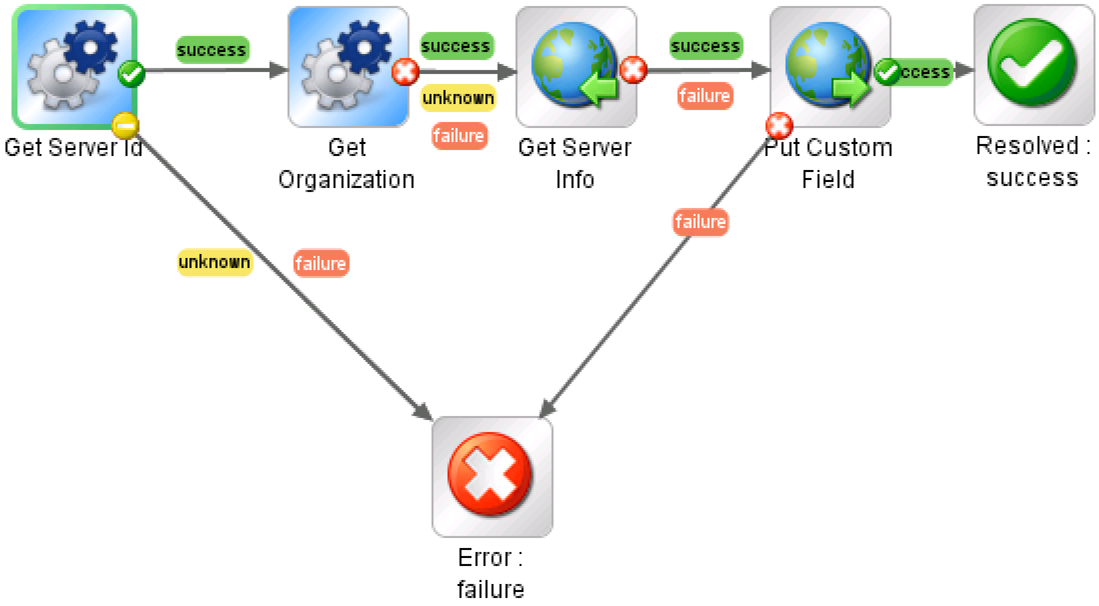
"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Example Flow: Put Server Custom Field

The Put Server Custom Field example flow modifies the value of the specified server Custom Field.



The flow produces a Success response if it successfully modifies the value of the Custom Field. It produces a Failure response if it cannot modify the Custom Field value for any reason.

How to Use this Flow

Use this flow when you want to change the value of a specific Custom Field for a specific server. You must know the name of the server, the name of the organization in which it resides, and the name of the Custom Field.

This flow uses the "Example Flow: Get Server Id" and "Example Flow: Get Organization Id" flows as steps (subflows).

You can use the "Example Flow: Get Server Custom Field" flow to learn the current value of a Custom Field.

What You Can Learn

The implementation of this example flow shows you how to do the following things:

- Invoke the DMA API to perform GET operations that search for the following things in the DMA environment:

  – UUID of the specified server (must also specify the organization name)

  – UUID of the organization in which the server resides

  – Detailed information about the server, including the names of its Custom Fields
- Use an XML Get Element filter to extract the definitions of the Custom Fields from the server details returned by the GET request.
- Invoke the DMA API to perform a PUT operation on the specified server Custom Field.

Permissions and Constraints

The specified dmaUser must have a role with both READ and WRITE permissions for the specified organization.

Inputs

This flow requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to get information about the specified organization and modify the Custom Field.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
organizationName	Name of the organization in which the specified server resides.
serverName	Name of the server whose Custom Field value will be modified.
customFieldName	Name of the Custom Field whose value will be modified.
customFieldValue	Value to which the Custom Field will be set.

Results

This flow does not produces results other than the Success and Failure responses.

How it Works

1. The "Example Flow: Get Server Id" step uses the DMA API to perform a GET request using the specified organizationName and serverName:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/server?orgName=<*organizationName*>&serverName=<*serverName*>

   It then uses an XPath query to extract the server's UUID from the response body of the GET.

   If the GET request does not find the server and extract the UUID, the flow terminates with a Failure response.

2. The "Example Flow: Get Organization Id" step performs a GET request using the specified organizationName:

GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/organization?orgName=<*organizationName*>

It then uses an XPath query to extract the organization's UUID from the response body of the GET.

3. The Get Server Info step performs a GET using the previously retrieved server UUID to get detailed information about the server, including a list of its Custom Fields:

   GET https://<*dmaServer*>:<*dmaPort*>/dma/api/env/server<*serverUUID*>

   It then uses the following XML Get Element filter to extract the list of Custom Fields from the response body of the GET:

   //*[local-name()='custom-field']

   It stores this list in the queryServerResult step result. This flow (as shipped) does not use this result as an input to a subsequent step, however. This information is provided strictly for your understanding.
4. The Put Custom Field step modifies the value of the the pertinent `<env:custom-field>` element in the response body from the GET in step 3 to create the request body (payload) for the following PUT request:

   PUT https://<*dmaServer*>:<*dmaPort*>/dma/api/env/server<*serverUUID*>
5. If the flow successfully modifies the value of the Custom Field, the flow terminates with a Success response. If it fails for any reason, it terminates with a Failure reponse.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

HPE DMA Operations

The DMA content pack for HP OO provides two operations:

Operation	Description
"Operation: Execute Deployment" on the next page	Executes a specific deployment to run a specific workflow.
"Operation: Simplified DMA Query" on page 68	Obtains information from the specified DMA server by executing a GET request via the DMA REST API.

These operations are provided primarily as learning tools that you can use as you create your own operations and flows.

This document contains a brief description of each operation. Additional detailed information is available on the Description tab in the for each operation in the Properties viewer in HP OO Studio.

See Also:

"HPE DMA Flow Examples" on page 12

"HPE DMA Wizard for HP OO" on page 71

Operation: Execute Deployment

The Execute Deployment operation executes a specific deployment to run a specific workflow. This operation is used by the "Example Flow: Execute Deployments Interactively" flow.

The operation produces a Success response if it successfully runs the workflow.It produces a Failure response if it fails to do so for any reason.

> **Tip:** To quickly and convenient create a flow that runs a workflow using a specific deployment, use the "HPE DMA Wizard for HP OO".

Permissions and Constraints

To run a workflow using the DMA API, the specified dmaUser must have a role with READ permission for the organization where the target resides and EXECUTE permission for the deployment. Users who have a role with Administrator capability can execute any deployment.

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Inputs

This operation accepts the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to execute a deployment to run a workflow against a target.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.
workflowName	Name of the workflow to run.
deploymentName	Name of the deployment that will be used to run the workflow.
deploymentParameters	Values for all Runtime deployment parameters required by the specified

Name	Description
	deployment. Parameters must be separated by \n. <step name>.<parameter name>=<value>\n<step name>.<parameter name>=<value>\n <step name>.<parameter name>=<value>\n For example: Ping Server.Hostname=target1.mycompany.com\nPingServer.Ping Count=4
targetServer	Name of the target server where the workflow will run.
targetInstance	Name of the target instance where the workflow will run.
targetDatabase	Name of the target database where the workflow will run.
nonBlocking	If TRUE, the operation will resolve immediate after it initiates the workflow. If FALSE, the operation will wait for the workflow execution to finish before resolving.

Additional optional parameters that enable you to specify proxy server information or a timeout for the operation are available. See the Inputs tab for this operation in HP OO Studio.

Results

This operation produces the following results. You can assign these results to outputs for use in subsequent steps.

Name	Example	Description
exception	com.hp.dma.oo.ras.ActionException: Invalid User and/or Password	Any exception thrown by the operation.
returnCode	0	Return code for the operation (0 = Success, -1 = Failure).
returnResult	Step "Ping Computer" completed with return code 0	Return code of the each step executed.
sessionId	iconclude5164169279501586959	HP OO session ID.
workflowRunStatus	Success	The current status of the workflow execution: Initiated, Running, Finished, Success, Failure, Cancelled, Aborted, or Skipped See the *DMA User Guide* for definitions of these workflow execution states.
workflowRunUrl	https://dma1.mycompany.com:8443/dma/api/auto/running/workflow/90cefae83ef26cbb013ef7ab57411491	The URL for the workflow execution created by this operation.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

"HPE DMA Operations" on page 64

Operation: Simplified DMA Query

The Simplified DMA Query operation obtains information from the specified DMA server by executing a GET request via the DMA REST API. It returns a list of results based on an XPath query.

You control the type of information returned by the operation by specifying the xpathQuery and apiPath variables.

The operation produces a Success response if it successfully performs the GET and extracts the correct information from the XML response body. It produces a Failure response if it cannot complete the GET or extract the information for any reason.

Implementation Details

The DMA API returns information as Atom feeds that utilize a default namespace. Default namespaces are not supported by HP OO 9.x XPath utility operations. As a result, working with the DMA API through HP OO XPath utility functions frequently requires utilizing the XPath function `local-name()` in XPath queries. The Simplified DMA Query Operation enables you to use simply path queries instead of utilizing the `local-name()` function. This is intended to make retrieving information from DMA more intuitive.

Permissions and Constraints

Each DMA API call requires different permissions and imposes its own constraints.

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Inputs

This operation requires the following inputs:

Name	Description
dmaServer	Host name or IP address of the DMA server.
dmaPort	The port used to communicate with the DMA server (default is 8443).
dmaProtocol	The protocol used to perform the DMA API calls (default is https).
dmaUser	The DMA user who has the privileges required to make the specified DMA API call.
dmaPassword	The password for the specified dmaUser.
trustAllCerts	Enables weak security over SSL. Any SSL certificate will be trusted regardless of whether it was issued by a trusted certification authority (CA). Do not set this to TRUE in a production environment.

Name	Description
delimiter	The character that will be used to separate items when a result contains a list of of items (default is |).
xpathQuery	The XPath query that will be used to extract information from the XML response body from the GET operation (for example: /feed/entry/id/text() or /dma/api/auto/policy).
apiPath	The path that specifies which DMA API call to use (for example:/dma/api/auto/policy or /dma/api/env/unassignedserver).

Additional optional parameters that enable you to specify proxy server information or a timeout for the operation are available. See the Inputs tab for this operation in HP OO Studio.

Results

This operation produces the following results. You can assign these results to outputs for use in subsequent steps. The examples shown here were generated using the following inputs:

xpathQuery = /feed/entry/id/text()

apiPath = /dma/api/auto/policy

Name	Example	Description
exception	com.hp.dma.oo.ras.ActionException: Invalid User and/or Password	Any exception thrown by the operation.
queryStatus	Success	Terminal state of the operation.
returnCode	0	Return code for the operation (0 = Success, -1 = Failure).
returnResult	https://dma1.mycompany.com:8443/dma/api/auto/policy/90cefae83eeccde1013eed09ede20375|https://dma1.mycompany.com:8443/dma/api/auto/policy/90cefae83eeccde1013eed0d1c31154e|https://dma1.mycompany.com:8443/dma/api/auto/policy/90cefae83eeccde1013eed0d1c091544|https://dma1.mycompany.com:8443/dma/api/auto/policy/90cefae83eeccde1013eed0d1bd9152b|https://dma1.mycompany.com:8443/dma/api/auto/policy/90cefae83eeccde1013eed0d1bb41515	List of items extracted from the GET response body XML. The contents of this list depend on the value of the xpathQuery and apiPath inputs and any filters applied to the output.
sessionId	iconclude5164169279501586931	HP OO session ID.

API Information

For information about the DMA API, see the interactive *API Reference Guide* available on your DMA server at this URL:

https://<*DMA Server host name*>:8443/dma/api

This guide provides complete documentation for all supported API calls and responses.

Related Topics

"Example Environment Flows" on page 37

"Example Automation Flows" on page 13

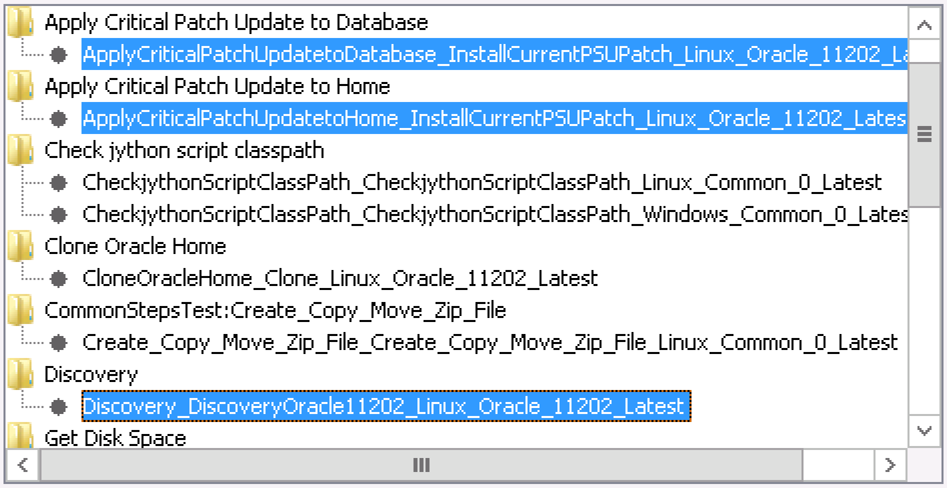
"HPE DMA Operations" on page 64

HPE DMA Wizard for HP OO

The HP DMA Configuration Wizard for Operations Orchestration creates HP OO flows that execute DMA deployments.

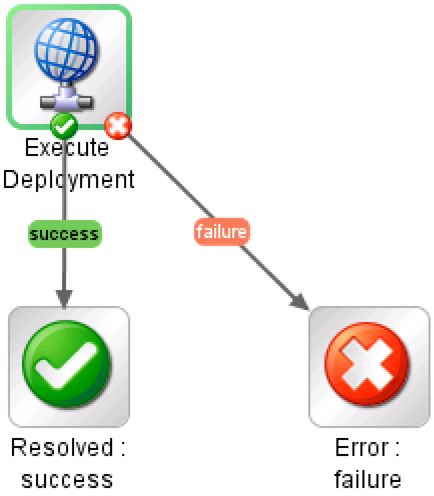
About the Wizard

When you run the wizard, you can specify any number of deployments. The wizard uses the DMA REST API to create a list of deployments available on the specified DMA server:



- Apply Critical Patch Update to Database
  - ApplyCriticalPatchUpdatetoDatabase_InstallCurrentPSUPatch_Linux_Oracle_11202_La
- Apply Critical Patch Update to Home
  - ApplyCriticalPatchUpdatetoHome_InstallCurrentPSUPatch_Linux_Oracle_11202_Lates
- Check jython script classpath
  - CheckjythonScriptClassPath_CheckjythonScriptClassPath_Linux_Common_0_Latest
  - CheckjythonScriptClassPath_CheckjythonScriptClassPath_Windows_Common_0_Lates
- Clone Oracle Home
  - CloneOracleHome_Clone_Linux_Oracle_11202_Latest
- CommonStepsTest:Create_Copy_Move_Zip_File
  - Create_Copy_Move_Zip_File_Create_Copy_Move_Zip_File_Linux_Common_0_Latest
- Discovery
  - Discovery_DiscoveryOracle11202_Linux_Oracle_11202_Latest
- Get Disk Space

The wizard then creates an HP OO flow like this for each deployment that you select from the list:



Runtime parameters in the HPE DMA deployment become inputs to the HP OO flow. By default, the user who runs the flow is prompted for the values of these inputs.

You can use these flows stand-alone, or you can include them in other flows.

The flow uses the "Operation: Execute Deployment" operation included in the DMA Content Pack for HP OO.

Use the Wizard to Create OO Flows

The HP DMA Configuration Wizard for Operations Orchestration creates an HP OO flow that executes an DMA deployment. It enables you to select one or more deployments from a list and creates one HP OO flow for each deployment that you select.

> **Note:** Make sure that you have the wizard available. See "Save the HPE DMA Configuration Wizard for HP OO" on page 11 for more information.
>
> Before running the wizard, make sure that the repository where you will create the DMA flows is a local repository and is not open in HP OO Studio.

To run the HP DMA Configuration Wizard for Operations Orchestration:

1. Run the wizard executable from the location where you stored it in "Save the HPE DMA Configuration Wizard for HP OO" on page 11.
2. On the wizard Welcome page, click **Next**.
3. On the wizard Specify Repository page, specify the location where the HP OO project will be created. This directory must already exist. For example:

   ```
   C:\DMAWizardProject
   ```

   Click **Next**.
4. On the wizard Connect to DMA page, specify the following information for your DMA server:

Setting	Description
HPE DMA Host	Host name or IP address of the DMA server.
HPE DMA Port	Port used to communicate with DMA (default is 8443).
Trust all SSL certificates	Select this box if you do not want HP OO to require the DMA server to present a trusted certificate (from a certificate authority). **Caution:** Use this setting ONLY for development or testing purposes. Do NOT use this setting this in a production environment.
User Name	The DMA user who will execute the deployment.
Password	The password for the specified DMA user.

   Click **Next**.

5. On the Select Deployments page, select one or more deployments from the list. Use CTRL+click to select more than one.

   Note: Currently there is an HP OO flow name limit of 128 characters. If the combination of your HPE DMA deployment name and workflow name is greater than 128 characters, HP OO Studio will fail when you import the HP OO project that was created by HPE DMA. Either select a different deployment or rename your workflow and/or deployment to be less than a combined 128 characters.

   Click **Next**.
6. On the Summary page, review the deployments that will be included in your HP OO flow.

   Click **Finish**.

Load the Project Created by the DMA Wizard into HP OO

1. Open HP OO Studio.
2. In the upper-left pane, select **Import Project**.
3. Navigate to the folder that you specified in the HPE DMA Wizard for the project, for example:

   `C:\DMAWizardProject`
4. Select **OK**.

Related Topic:

"Save the HPE DMA Configuration Wizard for HP OO" on page 11

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Integration Guide for HP Operations Orchestration 10.01 (Database and Middleware Automation 10.10 or later)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to hpe_dma_docs@hpe.com.

We appreciate your feedback!