



Creating building blocks for workflows

OnCommand Workflow Automation

NetApp
October 09, 2025

Table of Contents

Creating building blocks for workflows	1
Create a data source type	1
Create a command	2
Test the reservation script for commands	4
Create a finder	5
Create a filter	6
Create a dictionary entry	6
Create a function	7
Create a template	8
Create a cache query	9
Create recurring schedules	10
After you finish	10
Define filter rules	10
Add approval points	11

Creating building blocks for workflows

Workflow Automation (WFA) includes several building blocks, which are used to construct workflows. You can create the WFA buildings blocks that are required for your workflows.

Create a data source type

You must create a data source type to enable data acquisition from a data source, which is not predefined in OnCommand Workflow Automation (WFA).

What you'll need

- You must have created the required dictionary entry and scheme if you are creating a custom data source type that is not predefined in WFA.
- You must be aware of PowerShell scripting to create a data source type that uses the script method.

Steps

1. Click **Data Source Design > Data Source Types**.
2. Click  on the toolbar.
3. In the **New Data Source Type** dialog box, enter or select the required details in the **Data source**, **Data source version**, and **Scheme** fields.
4. In the **Default port** field, enter the port number.

2638

The port number you have entered is populated when you add this data source type for data acquisition. By default, the port is used by WFA to communicate with the data source and the port should be open on the data source server.

5. From the **Method** list, select the method that WFA must use to acquire data:

If you have selected...	Then...
SQL	From the Driver type list, select one of the following drivers that is appropriate for the data source: <ul style="list-style-type: none">• MySQL Connector/J• MySQL Server JDBC Driver 3.0• Oracle JDBC Driver 11.2.0.3

If you have selected...	Then...
SCRIPT	<p>In the Script field, enter the PowerShell script that is used to connect and retrieve data from the data source.</p> <p> The data in the dictionary entry equivalent CSV files should include tabs as field separator. For example, see the PowerShell script for the VMware vCenter data source type.</p>

6. Click **Save**.

Create a command

You can create a WFA command to complete a specific task in your workflow if there is no predefined WFA command that is suitable for the task.

What you'll need

You must know how to write the required code for the command using either PowerShell or Perl.

Steps

1. Click **Workflow Design > Commands**.
2. Click  on the toolbar.
3. In the **Properties** tab of the **New Command Definition** dialog box, enter or select the required details in the **Name**, **Description**, and **Timeout** fields.
 - a. In the **String Representation** field, enter a string representation for the command using MVEL syntax.

VolumeName + ">" + SnapshotName

The string representation for a command is used to display the information you want to see in the workflow design during planning and execution. You must use only the parameters of the command in string representation for a command.

4. In the **Code** tab, perform the following steps:
 - a. Select the required scripting language for the command from the **Script Language** list.

You can click  and select an additional language for the command.

- b. Enter the appropriate code for the command in the selected language tab.

If you want to use password type for user inputs in the PowerShell script, you must create an alias for the parameter and include `_Password` in the attribute. For Perl script, you can specify the type as `Password` in the **Parameters Definition** tab.



The following special characters are not supported in the command editor: "<", "&", and XML tags "< >".

```
param (
    [parameter(Mandatory=$false, HelpMessage="Specify an AD
administrator password.")]
    [Alias("ADAdminPassword_Password")] [string]$ADAdminPassword
)
```

5. In the **Parameters Definition** tab, perform the following steps:

- Click **Discover Parameters** to populate the parameters definition table.

The parameters and their attributes are extracted from the code and displayed in the table. For example, the **Array** and **VolumeName** parameters are extracted from the following code:

```
param (
    [parameter(Mandatory=$true, HelpMessage="Array name or IP
address")]
    [string]$Array,

    [parameter(Mandatory=$true, HelpMessage="Volume name")]
    [string]$VolumeName,
)
```

- Click the description column of the parameters to edit the description.

You cannot edit any other field in this tab.

6. In the **Parameters Mapping** tab, perform the following steps for each parameter:

- From the **Type** column, select the appropriate dictionary object.
- In the **Attribute** column, enter or select the appropriate attributes for the dictionary object from the list.

After entering an attribute, you can enter a period (.) and include another attribute of that object.

Enter `cm_storage.volume` as type and `aggregate.name` as the attribute for the `AggregateName` parameter.

- In the **Object Name** column, enter an object name.

The object name is used for grouping the parameters under a tab in the Parameters for <command> dialog box when you are specifying the command details in a workflow.

The unmapped parameters are displayed in the **Other parameters** tab of the Parameters for <command> dialog box when you are specifying the command details in a workflow.

7. In the **Reservation** tab, enter a reservation script using SQL queries to reserve the resources that are required by the command during a scheduled workflow execution:

a. **Optional:** In the **Reservation Representation** field, enter a string representation for the reservation using MVEL syntax.

```
``Add rule for SnapMirror label ``  
" + SnapMirrorLabel + "[ .code] to the SnapMirror policy`  
" + PolicyName + "``
```

The string representation is used to display the details of the resources reserved in the Reservations window.



The reservation script must not perform any operation on databases except the cm_storage, cm_performance, storage, performance, vc, and custom schemes.

8. **Optional:** In the **Verification** tab, enter an SQL query to verify whether the command has affected the data sources and the WFA cache as expected so that the reservation can be removed.

The SQL query that you enter can only consist of SQL SELECT statements.

- a. Click **Test Verification** to test the verification script.
- b. In the **Verification** dialog box, enter the required test parameters.
- c. If you do not want to use the reservation data to test the verification script, clear the **Use *reservation data in test** field.
- d. Click **Test**.
- e. After reviewing the test result, close the dialog box.

9. Click **Test** to test the command.

10. In the **Testing Command <command name>** dialog box, click **Test**.

The result of the test is displayed in the Log messages section of the dialog box.

11. Click **Save**.

Test the reservation script for commands

You can test the reservation scripts you have written for OnCommand Workflow Automation (WFA) commands on the playground database to ensure that the scripts are working fine and not affecting the WFA database tables.

About this task

The default WFA installation path is used in this procedure. If you changed the default location during installation, you must use the changed WFA installation path.

Steps

1. Open a command prompt on the WFA server and change directories to the following location:
c:\Program Files\NetApp\WFA\mysql\bin
2. Create a dump of the WFA database using the following command: mysqldump -u wfa -pWfa123 --single-transaction --skip-add-drop-table database_tables> dump_location

```
Command to create a dump of the cm_storage database tables: mysqldump -u wfa -pWfa123  
--single-transaction --skip-add-drop-table cm_storage> c:\tmp\cmSt2.sql
```

3. Restore the dump you have created on to the WFA playground database using the following command:
mysql -u wfa -pWfa123 playground < dump_location

```
mysql -u wfa -pWfa123 playground < c:\tmp\cmSt2.sql
```

4. Create or edit a WFA command and write the reservation script in the **Reservation** tab.

You must ensure that the reservation and verification scripts use only the playground database.

5. Create or edit a workflow, include the command in the workflow, and then execute the workflow.
6. Verify that the reservation and verification scripts are working as expected.

The WFA data source acquisition process does not update the playground database. You must remove the reservations created by the command manually.

Create a finder

You can create a WFA finder that can search for resources if there is no predefined WFA finder that is suitable for searching the required resources.

What you'll need

You must have created the required filters that are used in the finder.

Steps

1. Click **Workflow Design > Finders**.
2. Click  on the toolbar.
3. In the **Properties** tab of the **New Finder** dialog box, enter or select the required details in the **Name**, **Type**, and **Description** fields.
4. In the **Filters** tab, select the required filters from the **Available Filters** list and click .

You can add or remove filters based on your requirement.

5. In the Returned Attributes tab, select the required attributes for the filter from the **Available** list and click .

6. **Optional:** Click **Test** to test the finder.
 - a. In the **Test Finder <FinderName>** dialog box, enter the required test parameters.
 - b. Clear the **Use reservation data in test** check box if you do not want to use the reservation data for testing the finder.
 - c. Click **Test**.

The result of the test is displayed.

- d. Close the dialog box.

7. Click **Save**.

Create a filter

You can create a WFA filter that can search for resources if there is no predefined WFA filter that is suitable for the task.

What you'll need

You must know the appropriate SQL syntaxes to create the filter.

Steps

1. Click **Workflow Design > Filters**.
2. Click  on the toolbar.
3. In the **Properties** tab of the **New Filter** dialog box, enter or select the required details in the **Name**, **Dictionary type**, and **Description** fields.
4. In the **Query** tab, enter the appropriate SQL query for the filter.

You must enter a single SQL query and optionally use input parameters. You should use the following syntax to use an input parameter: \${ParameterName}.

```
SELECT
    array.ip
FROM
    storage.array
WHERE
    array.name = '${ArrayName}'
```

5. Click **Refresh** to populate the **Input Parameters** table and the **Returned Attributes** list.

This information is obtained from the SQL query that you have entered. For example, if you use the SQL query example from the previous step, ip is displayed in Returned Attributes and ArrayName is displayed in Input Parameters. You can edit the entries in the **Label** and **Description** columns.

6. **Optional:** Click **Test** to test the filter.

- a. In the **Test Filter <FilterName>** dialog box, enter the required test parameters.
- b. Clear the **Use reservation data in test** check box if you do not want to use the reservation data for testing the filter.
- c. Click **Test**.

The test result is displayed.

- d. Close the dialog boxes.

7. Click **Save**.

Create a dictionary entry

You can create a WFA dictionary entry when you want to define a new object type and its relationship in your storage environment.

Steps

1. Click **Data Source Design > Dictionary**.
2. Click  on the toolbar.
3. In the **New Dictionary Entry** dialog box, enter the required details in the **Name of object type** and the **Description** fields.
4. For the **Scheme** field, perform one of the following actions:
 - Select one of the available scheme from the list.
 - Click **Add New Scheme**, enter the required **Scheme Name** in the New Scheme dialog box, and then click **Add**.
5. Click **Add row**, and perform the following steps to describe the attribute:
 - a. Click the **Name** column and enter the name of the attribute.
 - b. From the **Type** column, select the required type.

The **String Length** column is populated and is editable if you selected string as the type. Also, the **Values** column is editable if you selected **enum** as the type.

 - c. Select the appropriate check boxes for the attribute from the **Natural Key**, **To be Cached**, and **Can be Null** columns.

If you have selected the **Natural Key** check box, you cannot select the **Can be Null** check box.

 - d. Add the required attributes for the dictionary object.
 - e. **Optional:** Select the **Values in natural key columns are case sensitive** check box if you want the natural keys to be case-sensitive.
6. Click **Save**.

Create a function

You can create a WFA function that can be used as a utility, if there is no predefined WFA function that is suitable for your task.

What you'll need

You must know MVFLEX Expression Language (MVEL) syntaxes to create a function.

About this task

You must include the following for the function definitions:

- Name: name of the function

You must not use a reserved word in MVEL syntax. Each function must have a unique name.

- MVEL definition: a string specifying the MVEL syntax of the function definition

Steps

1. Click **Workflow Design > Functions**.
2. Click  on the toolbar.
3. In the **New Function** dialog box, enter or select the required details in the **Function description** and

Function definition fields.

```
def actualVolumeSize(data_size, snap_pct)
{
    if (snap_pct < 0 ) {
        snap_pct = 0;
    } else if (snap_pct > 99) {
        snap_pct = 99;
    }

    div = 1 - (snap_pct/100);
    return (int)(data_size/div);
}
```

The **Function name** field is populated from the data that is used in the MVEL syntax.

4. Optional: Click **Test** to test the function:

- In the **Expression** section of the **Test** dialog box, enter the required expression of function.

```
actualVolumeSize(600, 1)
```

- Click **Test**.

The test result is displayed.

- Close the dialog box.

5. Click **Save**.

Create a template

You can create a template that can be used as a blueprint for filling up attributes in command details.

Steps

- Click **Workflow Design > Templates**.
- Click  on the toolbar.
- In the **New Template** dialog box, enter or select the required details in the **Name**, **Type**, and **Description** fields.

The Attributes table is populated based on the dictionary object you have selected in the **Type** field.

- Click the value column of each attribute and perform one of the following:
 - Enter or select the required value from the list.
 - Enter a user input entry—for example, \$size for size user input.
- Click **Save**.

Create a cache query

You can define a cache query when you want to cache information about a dictionary object in the WFA database from a data source type. You can create a cache query and associate it with a dictionary entry and one or more data source types, such as Active IQ Unified Manager 6.1.

What you'll need

You must know the appropriate SQL syntaxes to create a cache query.

Steps

1. Click **Data Source Design > Cache Queries**.
2. Click  on the toolbar.
3. In the **Add Cache Query** dialog box, select the required dictionary entry and data source type.
4. In the “SQL select query” section, enter the appropriate SQL query.

The following SQL query caches information about the disk dictionary object from the Active IQ Unified Manager 6.1 data source type:

```
SELECT
    disk.objId AS id,
    disk.name AS NAME,
    disk.uid AS uid,
    disk.effectiveInterfaceType AS TYPE,
    disk.rpm AS rpm,
    disk.homeNodeId AS home_node_id,
    disk.ownerNodeId AS owner_node_id,
    disk.model AS model,
    disk.serialNumber AS serial_number,
    disk.totalBytes/1024/1024 AS size_mb,
    disk.shelf AS shelf,
    disk.shelfBay AS shelf_bay,
    disk.pool AS pool,
    disk.vendor AS vendor,
    LOWER(disk.raidPosition) AS raid_position,
    disk.containerTypeRaw AS container_type,
    disk.clusterId AS cluster_id
FROM
    netapp_model_view.disk disk
```

5. If you want to test the SQL query, click **Test**.

If you have selected more than one data source type, the Test Cache Query dialog box opens and enables you to select the required data source type.

The test result is displayed.

6. Close the dialog box.

7. Click **Save**.

Create recurring schedules

OnCommand Workflow Automation (WFA) provides two scheduling options for workflows. You can either schedule a workflow to execute once at a specific time or you can create recurring schedules and associate the schedules to workflows so that the workflows are executed routinely.

About this task

A schedule that you have created can be reused and associated with several workflows.

Steps

1. Click **Execution > Schedules**.
2. Click  on the toolbar.
3. In the **New Schedule** dialog box, enter or select a name, description, and frequency for the schedule.

For frequency, you must enter time in 24-hour format. The WFA server time is applied to the schedules.

4. Click **OK**.

After you finish

- You can associate the schedule to a workflow when you execute the workflow by using the **Execute recurrently** option.
- You can view the details of a workflow and its association with a schedule by clicking **Execution > Recurring Schedules**.

The resource and execution planning for the workflows that are scheduled to execute once are done immediately when the workflows are scheduled. However, the resource and execution planning for the workflows with recurring schedule occur at the scheduled time and not when the schedule is associated with a workflow.

Define filter rules

You can define a set of rules for filtering dictionary entry resources such as vFiler units, aggregates, and virtual machines. You can customize the rules for existing workflows and for new workflows while you are creating them.

Steps

1. Log in to WFA through a web browser as an admin.
2. Click **Workflow Design > Workflows**.
3. In the **Workflows** window, double-click the workflow that you want to modify.

The Workflow <workflow name> window is displayed.

4. Define a set of rules by choosing one of the following options:

If you want to...	Then do this...
Search for resources when the commands in a row are repeated	<ol style="list-style-type: none">Click a row number and select Repeat row.In the Row Repetitions dialog box, select the For every resource in the group option from the Repeats drop-down list.Select a resource type.Click the Enter search criteria link.
Search for resources required in command inputs	<ol style="list-style-type: none">Click .In the Parameters for <command_name> dialog box, select the by searching for an existing <dictionary object> option from the Define <dictionary object> drop-down list.Click the Enter search criteria link.
Search for resources referenced by variables in command inputs	<ol style="list-style-type: none">Click .In the Parameters for <command_name> dialog box, select the by filling in attributes option from the Define <dictionary object> drop-down list.Click  for a field marked with .
Name command inputs of String type	<ol style="list-style-type: none">Click .In the Parameters for <command_name> dialog box, select the by filling in attributes option from the Define <dictionary object> drop-down list.Click  for a string field.

5. In the **Resource Selection** dialog box, select the **Define filter rules** check box.

If you have selected one of the options from the Finder drop-down of Resource Selection dialog box, the Define filter rules check box is disabled. The value for the finder must be set to "None" for the Define filter rules to be enabled.

6. Enter the attribute, operator, and value for the rule.

The value must be provided within single quotation marks. The filter rules can contain one or more groups.

7. Click **OK**.

Add approval points

You can add an approval point as a checkpoint in a workflow to pause the workflow

execution and resume it based on your approval. You can use approval points for incremental execution of a workflow, where sections of the workflow are executed only after a certain condition is met—for example, when the next section has to be approved or when successful execution of the first section is validated.

Steps

1. Log in to WFA through a web browser as an architect or an admin.
2. Click **Workflow Design > Workflows**.
3. In the **Workflows** window, double-click the workflow that you want to modify.
4. In the **Workflow <workflow name>** window, click the  icon to the left of the step for which you want to add the approval point.

You can add approval points for one or more steps.

5. In the **New Approval Point** dialog box, provide the comment and condition details.
6. Click **OK**.

Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.