NetApp

Understanding Workflow Automation designer

OnCommand Workflow Automation 5.1

NetApp
June 11, 2024

Table of Contents

Understanding Workflow Automation designer

You create workflows in the Workflow Automation (WFA) designer using the building blocks such as finders, filters, and commands. Understanding the building blocks and the workflow creation process is important before you start creating your workflows.

Working with the building blocks in OnCommand Workflow Automation

The Workflow Automation (WFA) workflows consist of several building blocks and WFA includes a library of the predefined building blocks. You can use the building blocks that WFA provides to create workflows that match the requirements of your organization.

WFA provides the structure for storage automation processes. WFA's flexibility is based on how the workflows are constructed by using the workflow building blocks.

The WFA building blocks are as follows:

- Dictionary entries
- Commands
- Filters
- Finders
- Functions
- Templates

You should understand how the building blocks are used in WFA to help you in creating the workflows.

What data sources are

A data source is a method for establishing a connection to other systems, files and databases in order to extract data. For example, a data source can be a connection to an Active IQ Unified Manager database of Active IQ Unified Manager 9.4 data source type.

You can add a custom data source to OnCommand Workflow Automation (WFA) for data acquisition after defining the required data source type by associating the caching scheme, the required port, and the acquisition method with the data source type.

WFA caches information through various data sources. WFA collects resource information from the data sources and formats it for the caching scheme. The cache tables, which are the tables inside the caching schemes, are formatted to match the dictionary entry objects. When you use a finder in workflows, it returns a dictionary object and the data from the dictionary object is populated from the cache tables. The process of acquiring data from the data sources is known as *data source acquisition*. You can use either a script-based method or a driver-based method for data source acquisition. The sources can be different from each other and data source acquisition might sample them at different time intervals. WFA then merges that information in to the database and superimposes the reservation data to maintain updated resource information in the database.

The WFA database includes several different caching schemes. A caching scheme is a set of tables and each

table includes information from a certain dictionary entry type; however, the tables might include combined information from multiple sources of a specific data source type. WFA uses the database information to understand the status of the resources, perform calculations, and execute commands on the resources.

What dictionary entries are

Dictionary entries are one of the OnCommand Workflow Automation (WFA) building blocks. You can use dictionary entries to represent object types and their relationships in your storage and storage-related environments. You can then use filters in workflows to return the value of the natural keys of the dictionary entries.

A dictionary entry is the definition of an object type that is supported by WFA. Each dictionary entry represents an object type and its relationship in the supported storage and storage-related environments. A dictionary object consists of a list of attributes, which might be type checked. A dictionary object with complete values describes an object instance of a type. In addition, the reference attributes describe the relationship of the object with the environment; for example, a volume dictionary object has many attributes, such as name, size_mb, and volume_guarantee. In addition, the volume dictionary object includes references to the aggregate and the array containing the volume in the form of array_id and aggregate_id.

The cache table of an object is a database containing a few or all of the dictionary entry's attributes that are marked for caching. For a dictionary entry to include a cache table, at least one of the dictionary entry's attributes must be marked for caching. Dictionary entries include natural keys, which are unique identifiers for the objects; for example, 7-Mode volumes are identified uniquely by their name and the IP address of the array containing them. Qtrees are identified by the qtree name, the volume name, and the array IP address. You must identify the dictionary attributes that are part of the dictionary entry's natural keys when creating dictionary entries.

How commands work

OnCommand Workflow Automation commands are the execution blocks for workflows. You can use a command for each step in your workflow.

WFA commands are written using PowerShell and Perl scripts. PowerShell commands use the Data ONTAP PowerShell toolkit and VMware PowerCLI, if the package is installed. Perl commands use the Perl distribution and Perl modules installed on the WFA server. If you include multiple scripting languages in a command, such as PowerShell and Perl, the appropriate script is chosen by WFA based on the operating system on which it is installed and the preferred order of language you have specified in the WFA configuration menu.

The scripts for the WFA commands include several parameters. These parameters might be mapped to dictionary entry attributes.

Note that each WFA command can include several Data ONTAP commands.

Some of the WFA commands are known as wait commands because they can wait for long-running operations and poll periodically—for example, the **Wait for multiple volume moves** command. The waiting interval at which the polling command is executed can be configured to check if the operation has been completed.

A WFA command is initiated by WFA while the workflow is in its execution phase. WFA executes the commands serially, in left-to-right and top-to-bottom order. The planning of the workflow confirms the availability and validity of the parameters that are supplied to the command. The WFA server supplies all the parameters required for the commands before the commands are executed.

Parameters to commands are finalized during workflow planning. The workflow then passes these parameters

to the commands during execution time. The commands cannot pass parameters back to the workflow. However, if you want to exchange information that is obtained during execution time between commands in a workflow, you can use the designated WFA PowerShell cmdlets or Perl functions.

WFA PowerShell commands do not use the -ErrorAction stop flag for the PowerShell cmdlets; therefore, workflow executions continue even when the cmdlets fail because of an error. If you want the -ErrorAction stop flag to be included in a specific command, you can clone the command and modify the PowerShell script to add the flag.

The following are the PowerShell cmdlets and Perl functions that are included in WFA to enable exchange of information between commands:

PowerShell cmdlets	Perl functions
Add-WfaWorkflowParameter	addWfaWorkflowParameter
Get-WfaWorkflowParameter	getWfaWorkflowParameter

Parameters added by the "add" cmdlets or functions to a command can be retrieved by a command that is executed subsequently and uses the "get" cmdlets or functions. For example, in a PowerShell WFA command, you can use the following in the code to add a parameter named volumeId: `Add-WfaWorkflowParameter -Name "VolumeUUID" -Value "12345" -AddAsReturnParameter $true`. Then, you can use the following in a subsequent command to retrieve the value of volumeId: `$volumeId = Get-WfaWorkflowParameter -Name volumeId`.

WFA commands can query the WFA database and obtain the required result. This enables you to construct a command without using filters and finders. You can use the following functions to query the database:

PowerShell cmdlet	Perl function
Invoke-MySqlQuery For example: `Invoke-MySqlQuery -Query "SELECT cluster.name AS 'Cluster Name' FROM cm_storage.cluster"`	invokeMySqlQuery

Related information

[Create a filter](#)

[Create a finder](#)

[Create a command](#)

What filters are

You can use WFA filters in your workflows to select the required resources.

A WFA filter is an SQL-based query that works on the WFA database. Each filter returns a list of elements of a specific dictionary type. The returned elements are based on the selection criteria specified in the SQL query. You must be aware of SQL syntaxes to create or edit a filter.

What finders are

A finder is a combination of one or more filters that are used together to identify common results. You can use a finder in your workflows to select the required resources for workflow execution.

Finders might apply a sorting order to differentiate the applicable results. Finders return the best resource based on the selection criteria and sorting.

Finders return either one result or no result; therefore, they can be used to verify the existence of certain storage elements. However, when a finder is used as part of a repeat row definition, the result sets are used to form the list of members in the group. Filters that are used in finders return the natural key of the dictionary type, at a minimum, but might return additional fields, whose value can be referenced. A sorting order might be applied to any returned field of a filter's SQL query.

You can test the results of a finder. When testing a finder, you can view the common results of all the WFA filters, where the effective result of the finder is highlighted in the results. When using a finder in a workflow, you can create a customized error message to convey meaningful information to the storage operator.

What functions are

You can use a function in your workflows for a complex task that has to be completed during the planning phase of the workflow.

You can write functions by using the MVFLEX Expression Language (MVEL). You can use functions to put together commonly used logic as well as more complex logic in a named function and reuse it as values for command parameters or filter parameters. You can write a function once and use it across workflows. You can use functions to handle repetitive tasks and tasks that might be complex, such as defining a complex naming convention.

Functions might use other functions during their execution.

What schemes are

A scheme represents the data model for a system. A data model is a collection of dictionary entries. You can define a scheme and then define a data source type. The data source defines how the data is acquired and the scheme is populated. For example, a vc scheme acquires data about your virtual environment, such as virtual machines, hosts, and datastores.

Schemes can also be populated directly with data through workflows that are customized to solve specific problems.

Dictionary entries are associated with an existing scheme when the dictionary entries are created. Dictionary entries are also associated with cache queries, and cache queries include SQL queries.

Schemes can acquire data using either script based data source type or SQL data source type. The scripts are defined while creating the data source type and SQL queries are defined in the cache queries.

The following schemes are included in WFA:

- **7-Mode (storage)**

Scheme to acquire data through Active IQ Unified Manager from Data ONTAP operating in 7-Mode.

- **Clustered Data ONTAP (cm_storage)**

  Scheme to acquire data through Active IQ Unified Manager from clustered Data ONTAP.

- **7-Mode Performance (performance)**

  Scheme to acquire performance data of Data ONTAP operating in 7-Mode through Performance Advisor.

- **Clustered Data ONTAP Performance (cm_performance)**

  Scheme to acquire performance data of clustered Data ONTAP through Performance Advisor.

- **VMware vCenter (vc)**

  Scheme to acquire data from VMware vCenter.

- **Playground (playground)**

  Scheme that can you can directly populate with data.

What remote system types are

OnCommand Workflow Automation (WFA) communicates with remote system types. A remote system type specifies the type of remote systems with which WFA can communicate. You can configure remote system types in WFA. For example, Data ONTAP system can be configured as a remote system type.

A remote system type has the following attributes:

- Name
- Description
- Version
- Protocol
- Port
- Timeout

You can have a Perl script for each remote system type to validate the credentials of the remote system. You can store the credentials for the remote systems configured on WFA. You can add or edit a new custom remote system type. You can also clone an existing remote system type. You can delete a remote system type only if no systems are associated with it.

How you use templates

You can use WFA templates in your workflows as a reference or for adhering to usage policies.

A WFA template acts as a blueprint of an object definition. You can define a template by including the properties of an object and the values for the object's properties. Then, you can use the template for populating the properties of an object definition in your workflows.

When you use a template, you cannot edit the fields that include the values that are obtained from the template. Therefore, you can use templates for setting up usage policies and creation of objects. If you remove the association of a template with the workflow after you have applied the template, the values populated from the template remain, but you can edit the fields.

How you use categories

You can categorize your workflows to better organize the workflows and to apply access control capability on the workflows.

You can categorize workflows such that they appear in specific groups on the WFA portal. You can also apply access control capability on workflow categories. For example, you can allow only certain storage operators or approvers to view certain categories of workflows. Storage operators or approvers can execute only the workflows within the category for which they have been granted access rights.

Active Directory groups also can be used for access control to categories.

How entity versioning works

The OnCommand Workflow Automation (WFA) entities, such as commands and workflows, are versioned. You can use the version numbers to easily manage changes to the WFA entities.

Each WFA entity includes a version number in the `major.minor.revision` format—for example, 1.1.20. You can include up to three digits in each part of the version number.

Before modifying the version number of a WFA entity, you must be aware of the following rules:

- Version numbers cannot be changed from the current version to an earlier version.
- Each part of the version must be a number from 0 through 999.
- New WFA entities are versioned as 1.0.0, by default.
- An entity's version number is retained when cloning or using **Save As** to save a copy of the entity.
- Multiple versions of an entity cannot exist in a WFA installation.

When you update the version of a WFA entity, the version of its immediate parent entity is updated automatically. For example, updating the version of the **Create Volume** command updates the **Create an NFS Volume** workflow, because the **Create an NFS Volume** workflow is an immediate parent entity of the **Create Volume** command. The automatic update to versions is applied as follows:

- Modifying the major version of an entity updates the minor version of its immediate parent entities.
- Modifying the minor version of an entity updates the revision version of its immediate parent entities.
- Modifying the revision version of an entity does not update any part of the version of its immediate parent entities.

The following table lists the WFA entities and their immediate parent entities:

Entity	Immediate parent entity
Cache query	• Data source type

Entity	Immediate parent entity
Template	• Workflow
Function	• Workflow • Template ⓘ If a function contains special or mixed case characters, the version of its immediate parent entities might not be updated.
Dictionary	• Template • Filter • Cache query • Command • Data source types which are using script method
Command	• Workflow
Filter	• Finder • Workflow
Finder	• Workflow
Data source type	None
Workflow	None

You can search for an entity in WFA either using the parts of the version number or the complete version number.

If you delete a parent entity, the child entities are retained and their version is not updated for the deletion.

How versioning works when importing entities

If you import entities from versions earlier than Workflow Automation 2.2, the entities are versioned as 1.0.0, by default. If the imported entity is already present in the WFA server, the existing entity is overwritten with the imported entity.

The following are the potential changes to WFA entities during an import:

• Upgrade of entities

  The entities are replaced with a later version.

• Rollback of entities

The entities are replaced with an earlier version.

> **(i)** When you perform a rollback of an entity, the version of its immediate parent entities are updated.

• Import of new entities

> **(i)** You cannot selectively import entities from a `.dar` file.

If a later version of an entity is imported, the version of its immediate parent entities is updated.

If there are multiple child entities to the imported parent entity, only the highest degree of change (major, minor, or revision) to the child entities is applied to the parent entity. The following examples explain how this rule works:

• For an imported parent entity, if there is one child entity with a minor change and another child entity with a revision change, the minor change is applied to the parent entity.

  The revision part of the parent's version is incremented.

• For an imported parent entity, if there is one child entity with a major change and another child entity with a minor change, the major change is applied to the parent entity.

  The minor part of the parent's version is incremented.

Example of how the versions of imported child entities affect the parent's version

Consider the following workflow in WFA: "Create Volume and export using NFS - Custom" 1.0.0.

The existing commands included in the workflow are as follows:

• "Create Export Policy - Custom" 1.0.0
• "Create Volume - Custom" 1.0.0

The commands included in the `.dar` file, which is to be imported, are as follows:

• "Create Export Policy - Custom" 1.1.0
• "Create Volume - Custom" 2.0.0

When you import this `.dar` file, the minor version of the "Create Volume and export using NFS - Custom" workflow is incremented to 1.1.0.

What a playground database is

The playground database is a MySQL database, which is included in the Workflow Automation (WFA) server installation. You can add tables to the playground database to include information, which can be used by filters and SQL queries for user inputs.

The playground database is a schema that cannot be accessed through the WFA web portal. You can use a MySQL client, such as SQLyog, Toad for MySQL, and MySQL Workbench or a command-line interface (CLI), to access the database.

You must use the following credentials to access the playground database:

- User name: wfa
- Password: Wfa123

The credentials provide complete access to the playground database and read-only access to other schemas defined in the WFA MySQL database. You can create the required tables in the playground database.

You can add the tags or metadata that you are using for storage objects in your environment to a table in the playground database. The tags or metadata can then be used along with the information in other WFA cache tables by WFA filters and user input queries.

For example, you can use the playground database for the following use cases:

- Tagging aggregates with business unit (BU) name and allocating volumes to the BUs based on these tags
- Tagging vFiler units with BU names
- Adding geography or location details to storage objects
- Defining access of database admins to databases

For example, if you are using the name of the BU as a tag for the storage objects, such as aggregates and vFiler units, you can create a table in the playground database that includes the name of the BU. The BU name can then be used by filters and user input queries for your workflows.

The following is an example playground database table (playground.volume_bu):

array_ip	volume_name	BU
10.225.126.23	data_11	Marketing
10.225.126.28	arch_11	HR

The following is an example SQL query that you can use to filter volumes by BU:

```
SELECT
    vol.name,
    array.ip AS 'array.ip'
FROM
    storage.volume AS vol,
    storage.array AS array,
    playground.volume_bu AS vol_bu
WHERE
    vol.array_id = array.id
    AND array.ip = vol_bu.array_ip
    AND vol.name = vol_bu.volume_name
    AND vol_bu.bu = '{$bu}'
```

Related information

SQLyog

MySQL Workbench

Toad for MySQL