OnCommand Workflow Automation Designer features

OnCommand Workflow Automation 5.0

NetApp
May 03, 2022

Table of Contents

OnCommand Workflow Automation Designer features

OnCommand Workflow Automation includes various features to help you design storage workflows.

For more information about the features, see the next topics.

How repeat row works

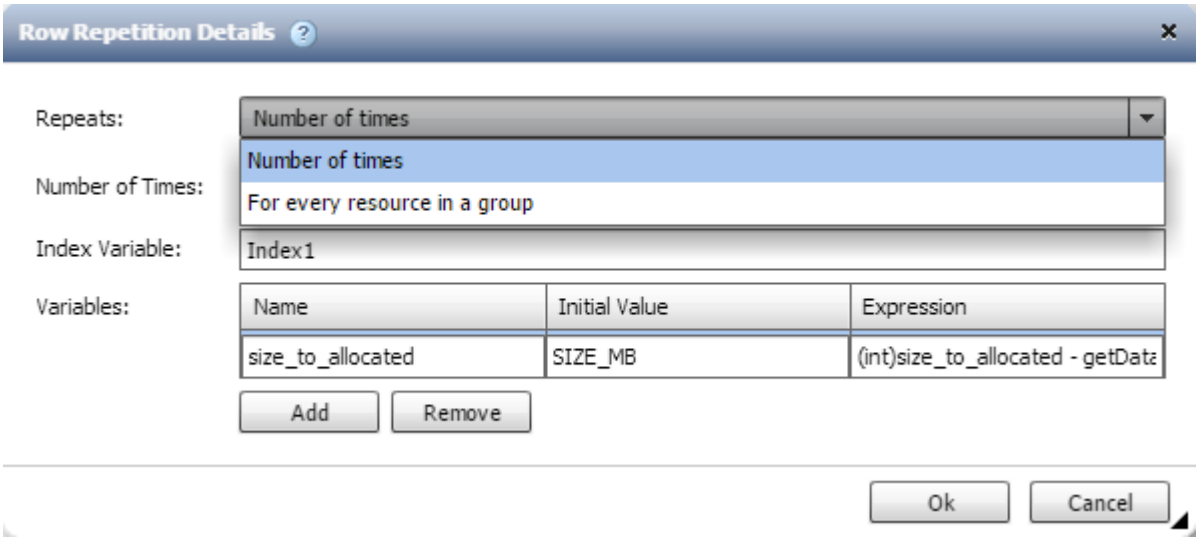
A workflow contains commands and command details arranged in rows. You can specify the commands in a row to be repeated for a fixed number of iterations or dynamic number of iterations based on the results of a search criteria.

The command details in a row can be specified to repeat a certain number of times or when the workflow is designed. The workflow can also be designed such that the number of times the row must repeat can be specified when the workflow is executed or scheduled for an execution. You can specify a search criteria for an object and the commands in a row can be set to repeat as many times as the objects are returned by the search criteria. Rows can also be set to repeat when certain conditions are met.

Row repetition variables

You can specify variables in the variable list that can be manipulated during the row iterations. For the variables, you can specify a name, a value with which the variables are initialized, and an MVFLEX Expression Language (MVEL) expression that is evaluated after every iteration of the row repetition.

The following illustration shows the repeat row options and an example of a row repetition variable:



Row repetition with approval points

When you have specified iterations of repeat rows for commands and included approval points, all the iterations of the commands before an approval point are executed. After you approve the approval point, the execution of all iterations of the successive commands continues until the next approval point.

The following illustration shows how the iterations of repeat rows are executed when an approval point is included in a workflow:

Repeat row examples in predefined workflows

You can open the following predefined workflows in the Designer to understand how repeat rows are used:

- Create a Clustered Data ONTAP NFS Volume
- Create VMware NFS Datastore on Clustered Data ONTAP Storage
- Establish Cluster Peering
- Remove a Clustered Data ONTAP Volume

What approval points are

Approval points are check points used in a workflow to pause the workflow execution and resume it based on a user approval.

The blue vertical bar shown in the following illustration is an approval point:

You can use approval points for incremental execution of a workflow, where sections of the workflow should be 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. Approval points do not handle any process between pausing and resuming of a workflow. Email and SNMP notifications are sent, as specified in the WFA configuration, and the storage operator can be asked to perform certain actions upon receiving the workflow pause notification. For example, the storage operator can send planning details to admin, approver, or operator for approval and resume the workflow when the approval is received.

Approvals might not be required at all times. In some scenarios, the approval might be required only if a particular condition is met and the conditions can be configured when an approval point is added. For example,

consider a workflow that is designed to increase the size of a volume. You can add an approval point at the beginning of the workflow for the storage operator to obtain approval from the managers when the increase in the volume size results in an 85% usage of the space in the aggregate that contains the volume. During the workflow execution and on selecting a volume that results in this condition, the execution is stopped until it is approved.

The condition that is set up for the approval point can have one of the following options:

- Without any condition
- When the variable you have specified is found
- When the variable you have specified is not found
- When the expression you have specified evaluates to true

There is no limitation on the number of approval points in a workflow. You can insert approval points before commands in a workflow and set the commands after the approval point to wait for approval before execution. Approval points provide information, such as time of change, user, and comments, allowing you to see when and why the workflow execution was paused or resumed. The approval point comments can include MVEL expressions.

Approval point examples in predefined workflows

You can open the following predefined workflows in the Designer to understand how approval points are used:

- Remove a Clustered Data ONTAP Volume
- Controller and shelf upgrade of an HA pair
- Migrate Volumes

How continue on failure works

The continue on failure feature helps you to configure a step in a workflow so that the workflow execution can continue even if the step fails. You can address the failed steps and resolve the issue that caused the step to fail by accessing the `wfa.log` file or by clicking the icon.

A workflow that has one or more such failed steps is in the Partially Successful state after the execution is complete. You can configure a step so that the workflow execution continues even if the step fails by selecting the required option in the Advanced tab of the Parameters for <command_name> dialog box.

If a step is not configured to continue on failure, the workflow execution is aborted if the step fails.

If a step that is configured to continue on failure fails, you can set the workflow to be executed by using one of the following options:

- Abort workflow execution (default option)
- Continue execution from the next step
- Continue execution from the next row

How resource selection works

OnCommand Workflow Automation (WFA) uses search algorithms to select storage resources for workflow execution. You should understand how resource selection works in order to design workflows efficiently.

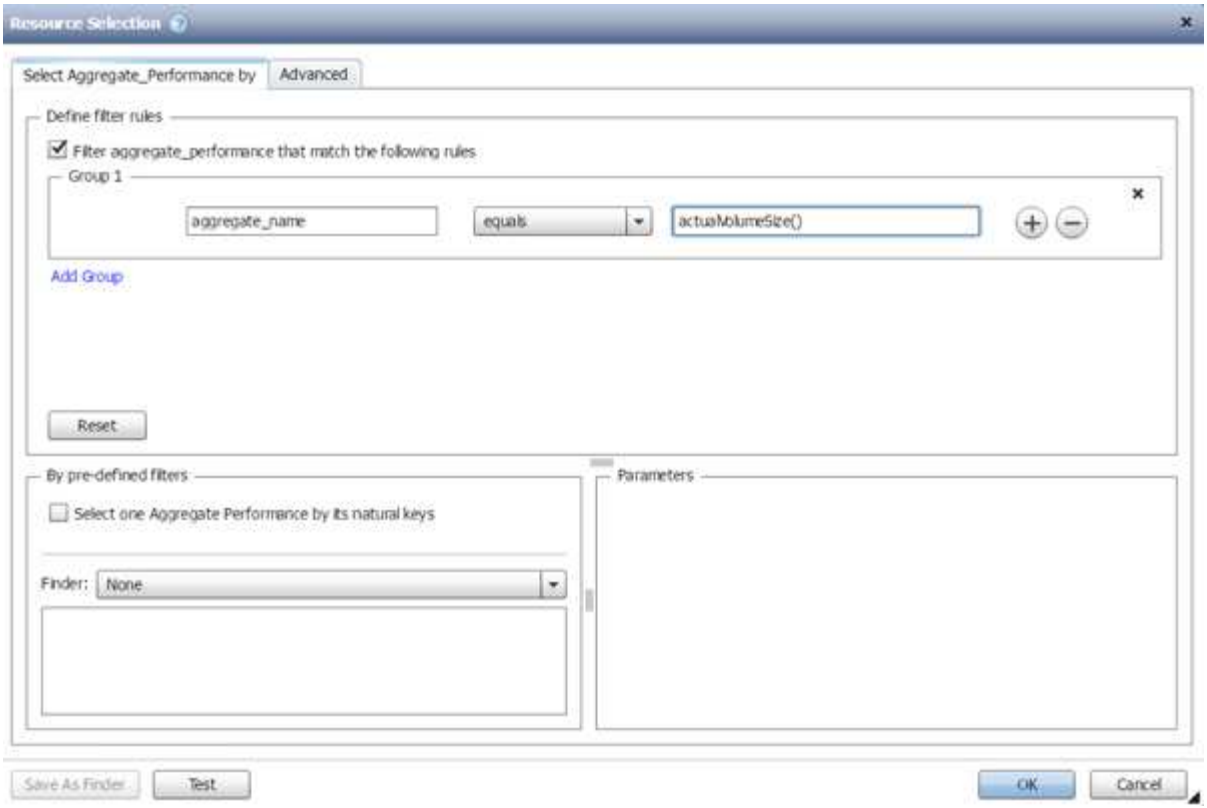
WFA selects dictionary entry resources—such as vFiler units, aggregates, and virtual machines—using search algorithms. The selected resources are then used for executing the workflow. The WFA search algorithms are part of the WFA building blocks, and include finders and filters. To locate and select the required resources, the search algorithms search through the data that is cached from different repositories, such as OnCommand Unified Manager, VMware vCenter Server, and a database. By default, a filter is available for every dictionary entry for searching a resource based on its natural keys.

You should define the resource selection criteria for each command in your workflow. In addition, you can use a finder to define the resource selection criteria in each row of your workflow. For example, when you are creating a volume that requires a specific amount of storage space, you can use the “Find aggregate by available capacity” finder in the “Create Volume” command, which selects an aggregate with a specific amount of available space and creates the volume on it.

You can define a set of filter rules for dictionary entry resources, such as vFiler units, aggregates, and virtual machines. Filter rules can contain one or more groups of rules. A rule consists of a dictionary entry attribute, an operator, and a value. The attribute can also include attributes of its references. For example, you can specify a rule for aggregates as follows: List all aggregates that have names starting with the string “aggr” and have more than 5 GB of available space. The first rule in the group is the attribute “name”, with the operator “starts-with”, and the value “aggr”. The second rule for the same group is the attribute “available_size_mb”, with the operator “>”, and the value “5000”. You can define a set of filter rules along with public filters. The Define filter rules option is disabled if you have selected a finder. The Save as Finder option is disabled if you have selected the Define filter rules check box.

In addition to the filters and finders, you can use a search or define command to search for available resources. The search or define command is the preferred option over the No-op commands. The search and define command can be used to define resources of both the certified dictionary entry type and the custom dictionary entry type. The search or define command searches for resources but does not perform any action on the resource. However, when a finder is used to search for resources, it is used in the context of a command, and the actions defined by the command are executed on the resources. The resources returned by a search or define command are used as variables for the other commands in the workflow.

The following illustration shows that a filter is used for resource selection:



Resource selection examples in predefined workflows

You can open the command details of the following predefined workflows in the Designer to understand how resource selection options are used:

- Create a Clustered Data ONTAP NFS Volume
- Establish Cluster Peering
- Remove a Clustered Data ONTAP Volume

How reservation works

OnCommand Workflow Automation resource reservation capability reserves the required resources to ensure that the resources are available for successful execution of workflows.

WFA commands can reserve the required resources and remove the reservation after the resource is available in the WFA cache database, typically after a cache acquisition. The reservation capability ensures that the reserved resources are available for the workflow until the reservation expiration period that you have configured in the WFA configuration settings.

You can use the reservation capability to exclude resources reserved by other workflows during resource selection. For example, if a workflow that has reserved 100 GB of space on an aggregate is scheduled for execution after a week, and you are executing another workflow that uses the **Create Volume** command, the workflow that is executing does not consume the space reserved by the scheduled workflow to create a new volume. In addition, the reservation capability enables workflows to be executed in parallel.

When previewing a workflow for execution, the WFA planner considers all the reserved objects, including the existing objects in the cache database. If you have enabled reservation, the effects of the scheduled workflows

and the workflows that are executing in parallel, and the existence of storage elements are considered when planning the workflow.

The arrow in the following illustration shows that reservation is enabled for the workflow:

Reservation examples in predefined workflows

You can open the following predefined workflows in the Designer to understand how reservation is used:

- Clone Environment
- Create a Clustered Data ONTAP Volume
- Establish Cluster Peering
- Remove a Clustered Data ONTAP Volume

What incremental naming is

Incremental naming is an algorithm that enables you to name the attributes in a workflow based on the search results for a parameter. You can name the attributes based on an incremental value or a custom expression. The incremental naming functionality helps you implement a naming convention based on your requirement.

You can use the incremental naming functionality when designing workflows to dynamically name the objects created by the workflow. The functionality enables you to specify a search criteria for an object using the resource selection feature and the value returned by the search criteria is used for the object's attribute. In addition, you can specify a value for the attribute if no object was found with the specified search criteria.

You can use one of the following options for naming the attributes:

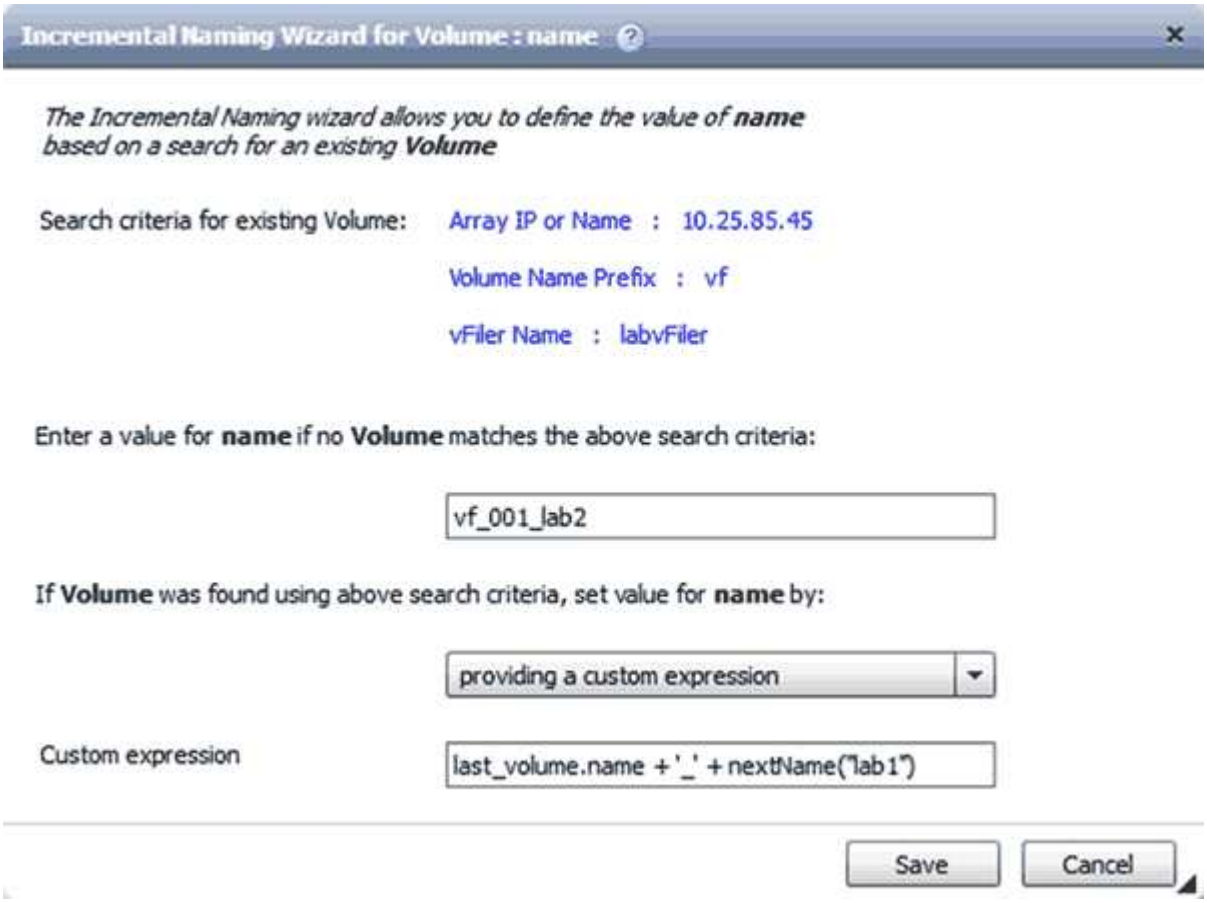
- Providing an increment value and suffix

  You can provide a value that should be used along with the value of the object found by the search criteria and increment with the number you specify. For example, if you want to create volumes with the naming convention of *filer name_unique number_environment*, you can use a finder to find the last volume by its name prefix and increment the unique number by 1, as well as add the suffix name to the volume name. If the last volume name prefix found was *vf_023_prod* and you are creating three volumes, the names for the volumes created are *vf_024_prod*, *vf_025_prod*, and *vf_026_prod*.

- Providing a custom expression

You can provide a value that should be used along with the value of the object found by the search criteria and add additional values based on the expression you enter. For example, if you want to create a volume with the naming convention of *last volume name_environment name padded with 1*, you can enter the expression `last_volume.name + '' + nextName("lab1").` If the last volume name found was `_vf_023`, the name for the volume created is *vf_023_lab2*.

The following illustration shows how a custom expression can be provided to specify a naming convention:



What conditional execution is

Conditional execution helps you to design workflows that can execute commands when specified conditions are met.

Execution of commands in a workflow can be dynamic. You can specify a condition for the execution of each command or a row of commands in your workflow. For example, you might want the “Add volume to dataset” command to be executed only when a specific dataset is found and you do not want the workflow to fail if the dataset is not found. In this case, you can enable the “Add volume to dataset” command to search for a specific dataset and if it is not found, you can disable the command in the workflow.

Options for conditional execution of commands are available in the *Dictionary object* tab and the Advanced tab of the Parameters for *commands* dialog box.

You can abort a workflow or disable a specific command in the workflow. In addition, you can set a command to be executed using one of the following options:

- Without any condition

- When the variables you have specified are found
- When the variables you have specified are not found
- When the expression you have specified is true

You can also set a command to wait for a specific time interval.

Conditional execution examples in predefined workflows

You can open the command details of the following predefined workflows in the Designer to understand how conditional execution of commands are used:

- Create a basic Clustered Data ONTAP Volume
- Create a Clustered Data ONTAP NFS Volume

How return parameters work

Return parameters are parameters that are available after the planning phase of a workflow. The values returned by these parameters are useful in debugging a workflow. You should understand how return parameters work and what parameters can be used as return parameters to debug workflows.

You can designate a set of parameters, such as variable attributes, expressions, and user input values, in a workflow as return parameters. During workflow execution, the values of the designated parameters are populated in the planning phase and execution of the workflow starts. The values of these parameters are then returned the way they were calculated in that specific execution of the workflow. If you want to debug the workflow, you can refer to the values that were returned by the parameters.

You can specify the required return parameters in a workflow when you want to see what are the calculated or selected values for those parameters. For example, when using resource selection logic to select an aggregate in a workflow, you can specify `aggregate` as the return parameter so that you can see which aggregate was selected during the planning of the workflow.

Before referring to the values of the return parameters for debugging your workflow, you should confirm that the execution of the workflow is complete. The return parameter values are set for each workflow execution. If you have added a return parameter after several executions of a workflow, the value of that parameter is available only for executions after the addition of the parameter.

Parameters that can be used as return parameters

Return parameters	Example
Variable attributes that are scalar	`volume1.name`, which is an attribute of the “volume name” variable
Constants	MAX_VOLUME_SIZE
User inputs	$clusterName

Return parameters	Example
MVEL expressions that involve variable attributes, constants, and user inputs	volume1.name+'-'+$clusterName
The return parameter that a command adds during execution	The `$volumeUUID` parameter is added as a return parameter when you use the following line in a PowerShell command: `Add-WfaWorkflowParameter -Name "VolumeUUID" -Value “12345” -AddAsReturnParameter $true`.

Examples of return parameters in predefined workflows

If you want to understand how return parameters are specified, you can open the following predefined workflows in the Designer and review the specified return parameters:

- Create an NFS Volume in a vFiler
- Create a Qtree CIFS Share in a vFiler
- Create a Clustered Data ONTAP Volume CIFS Share

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 OnCommand Unified Manager from Data ONTAP operating in 7-Mode.

- **Clustered Data ONTAP (cm_storage)**

  Scheme to acquire data through OnCommand 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 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 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
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

Entity	Immediate parent entity
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.

  When you perform a rollback of an entity, the version of its immediate parent entities are updated.

- Import of new entities

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.