Guidelines for WFA building blocks

OnCommand Workflow Automation

NetApp
October 09, 2025

This PDF was generated from https://docs.netapp.com/us-en/workflow-automation/workflows/reference-guidelines-for-sql-in-wfa.html on October 09, 2025. Always check docs.netapp.com for the latest.

Table of Contents

Guidelines for WFA building blocks

You must be aware of the guidelines for using Workflow Automation building blocks.

Guidelines for SQL in WFA

You must be aware of the guidelines for using SQL in OnCommand Workflow Automation (WFA) to write SQL queries for WFA.

SQL is used in the following places in WFA:

- SQL queries to populate user inputs for selection
- SQL queries for creating filters to filter objects of a specific dictionary entry type
- Static data in tables in the playground database
- A custom data source type of SQL type where the data has to be extracted from an external data source such as a custom configuration management database (CMDB).
- SQL queries for reservation and verification scripts

Guidelines	Example

SQL reserved keywords must be in uppercase characters.	`SELECT`
	`    vserver.name`
	`FROM`
	`    cm_storage.vserver vserver`

Table and column names must be in lowercase characters.	Table: aggregate Column: used_space_mb
Separate words with an underscore (_) character. Spaces are not allowed.	array_performance
Table name is defined in singular. A table is a collection of one or more entries.	“function”, not “functions”

Guidelines	Example

Use table aliases with meaningful names in SELECT queries.	```
	SELECT
	    vserver.name
	FROM
	    cm_storage.cluster cluster,
	    cm_storage.vserver vserver
	WHERE
	    vserver.cluster_id =
	cluster.id
	    AND cluster.name =
	'${ClusterName}'
	    AND vserver.type = 'cluster'
	ORDER BY
	    vserver.name ASC
	``

Guidelines	Example

If you have to refer to a filter input parameter or user input parameter in a filter query or user query, use the syntax as '${inputVariableName}.You can also use the syntax to refer to a command definition parameter in reservation scripts and verification scripts.	`SELECT`
	`    volume.name AS Name,`
	`    aggregate.name as Aggregate,`
	`    volume.size_mb AS 'Total Size (MB)',`
	`    voulme.used_size_mb AS 'Used Size (MB)',`
	`    volume.space_guarantee AS 'Space Guarantee'`
	`FROM`
	`    cm_storage.cluster,`
	`    cm_storage.aggregate,`
	`    cm_storage.vserver,`
	`    cm_storage.volume`
	`WHERE`
	`    cluster.id = vserver.cluster_id`
	`    AND aggregate.id = volume.aggregate_id`
	`    AND vserver.id = voulme.vserver_id`
	`    AND vserver.name = '${VserverName}'`
	`    AND cluster.name = '${ClusterName}'`
	`ORDER BY`
	`    volume.name ASC`

Use comments for complex queries. Some of the supported comment styles in queries are as follows:
• “--” until the end of the line
A space is mandatory after the second hyphen in this comment style.
• From a “#” character until the end of the line
• From a “/**” to the following "**`/`"sequence
	`/*`
	`multi-line`
	`comment`
	`*/`
	`--line comment`
	`SELECT`
	`    ip as ip, # comment till end of this line`
	`    NAME as name`
	`FROM --end of line comment`
	`    storage.array`

Guidelines for WFA functions

You can create functions to encapsulate commonly used and more complex logic in a named function, and then reuse the function as command parameter values or filter parameters values in OnCommand Workflow Automation (WFA).

Guidelines	Example
Use Camel case for a function name.	calculateVolumeSize
Variable names should be in plain English and related to the functionality of the function.	splitByDelimiter
Do not use abbreviations.	calculateVolumeSize, *not* calcVolSize
Functions are defined using MVFLEX Expression Language (MVEL).	None
The function definition should be specified according to the official Java Programming Language guidelines.	None

Guidelines for WFA dictionary entries

You must be aware of the guidelines for creating dictionary entries in OnCommand Workflow Automation (WFA).

Guidelines	Example
Dictionary entry names must contain only alphanumeric characters and underscores.	Cluster_License Switch_23
Dictionary entry names must start with an uppercase character. Begin every word in the name with an uppercase character and separate each word with an underscore (_).	Volume Array_License
Dictionary entry attribute names should not include the name of the dictionary entry.	None
Attributes and references in a dictionary entry must be in lowercase characters.	aggregate, size_mb
Separate words with an underscore. Spaces are not allowed.	resource_pool

Guidelines	Example
Dictionary entries cannot include references that are from a different scheme. When a dictionary entry requires cross-reference to an object in a different scheme, ensure that all the natural keys of the object being referred to are present in the dictionary entry.	Array_Performance dictionary entry requires all the natural keys of the Array dictionary entry as direct attributes in it.
Use appropriate data types for attributes.	None
Use Long data type for size or space-related attributes.	size_mb and available_size_mb in storage.Volume dictionary entry
Use Enum when an attribute has a fixed set of values.	raid_type in storage.Volume dictionary entry
Set “To be Cached” as true for an attribute or reference when a data source provides value for that attribute or reference.For Active IQ Unified Manager data source, add cacheable attributes if the data source can provide the value to it.	None
Set “Can be Null” as true if the data source providing the value for this attribute or reference can return NULL.	None
Provide a meaningful description to each attribute and reference.The description is displayed in command details when designing a workflow.	None
Do not use “id” as the name of an attribute in dictionary entries.It is reserved for internal WFA usage.	None

Related information

References to learning material

Guidelines for commands

You must be aware of the guidelines for creating commands in OnCommand Workflow Automation (WFA).

Guidelines	Example
Use an easily identifiable name for commands.	`Create Qtree`
Use spaces to delimit words and each word must start with an uppercase character.	`Create Volume`

Guidelines	Example
Provide a description to explain the functionality of the command, including the expected outcome of the optional parameters.	None
By default, the timeout for standard commands is 600 seconds. The default timeout is set while creating the command. Change the default value only if the command might take a longer time to complete.	`Create Volume` command
In case of long-running operations, create two commands—one to invoke the long-running operation and another to report the progress of the operation periodically. The first command should be a `Standard Execution` command type and the second should be `Wait for Condition` command type.	`Create VSM` and `Wait for VSM` commands
Prefix the `Wait for condition` command names with “Wait” for easy identification.	`Wait for CM Volume Move`
Use an appropriate waiting interval for the “Wait for condition” commands. The specified value governs the interval at which the polling command gets executed to check if the long-running operation is complete.	60s sampling interval for the `Wait for VSM` command
For the `Wait for condition` commands, use an appropriate timeout based on the expected time for the long-running operation to complete. The expected time might be considerably longer if the operation involves data transfer over a network.	A VSM baseline transfer can take many days to complete. Therefore, the specified timeout is 6 days.

String representation

The string representation for a command displays the details of a command in a workflow design during planning and execution. Only the command parameters can be used in the string representation for a command.

Guidelines	Example
Avoid using attributes that do not have any value. An attribute without a value is displayed as NA.	VolName 10.68.66.212[NA]aggr1/testVol7
Separate different entries in the string representation using the following delimiters: [ ] , / :	*`ArrayName[ArrayIp]`*

Guidelines	Example
Provide meaningful labels to every value in string representation.	*Volume name=VolumeName*

Command definition language

Commands can be written using the following supported scripting languages:

- PowerShell
- Perl

Command parameter definition

The command parameters are described by Name, Description, Type, a default value for the parameter, and whether the parameter is mandatory. The parameter type can be String, Boolean, Integer, Long, Double, Enum, DateTime, Capacity, Array, Hashtable, Password, or an XmlDocument. While the values for most of the types are intuitive, the values for Array and Hashtable should be in a particular format as described in the following table:

Guidelines	Example
Ensure that the value for an Array input type is a list of values, separated by comma.	`[parameter(Mandatory=$false, HelpMessage="Months in which the schedule executes.")] [array]$CronMonths` Input is passed as following: 0,3,6,9
Ensure that the value for a Hashtable input type is a list of key=value pairs, separated by semicolon.	`[parameter(Mandatory=$false, HelpMessage="Volume names and size (in MB)")] [hashtable]$VolumeNamesAndSize` Input is passed as following: Volume1=100;Volume2=250;Volume3=50

Guidelines for workflows

You must be aware of the guidelines for creating or modifying a predefined workflow for OnCommand Workflow Automation (WFA).

General guidelines

Guidelines	Example
Name the workflow such that it reflects the operation that is executed by the storage operator.	`Create a CIFS Share`
For workflow names, capitalize the initial letter of the first word and every word that is an object. Capitalize letters for abbreviations and acronyms.	Volume Qtree Create a Clustered Data ONTAP Qtree CIFS Share
For workflow descriptions, include all of the important steps of the workflow, including any prerequisites, result of the workflow, or conditional aspects of execution.	See the description of the sample workflow `Create VMware NFS Datastore on Clustered Data ONTAP Storage`, which includes the prerequisites.
Set “Ready For Production” to `true` only when the workflow is ready for production and can be displayed in the portal page.	None
By default, set “Consider reserved elements” to true. When previewing a workflow for execution, the WFA planner considers all of the objects that are reserved along with the existing objects in the cache database. Effects of other scheduled workflows or workflows executing in parallel are considered when planning a specific workflow if this option is set to `true`.	• Scenario 1 Workflow 1 creates a volume, and is scheduled to execute one week later. Workflow 2 creates qtrees or LUNs in volumes that are searched for, and if workflow 2 is executed within a day or so, you should turn off “Consider reserved elements” for workflow 2 to prevent it from considering the volume that is to be created in a week. • Scenario 2 Workflow 1 uses the `Create Volume` command. If there is a scheduled workflow 2 that consumes 100 GB from an aggregate, then workflow 1 must consider the requirements for workflow 2 during planning.

Guidelines	Example
By default, “Enable element existence validation” is set to `true`.	• Scenario 1 If you create a workflow that first removes a volume by name using the command `Remove Volume` only if the volume exists, and the re-creates it using another command such as `Create Volume` or `Clone Volume`, then the workflow should not use this flag. The effect of removing the volume will not be available to the `Create volume` command, thereby causing the workflow to fail. • Scenario 2 The `Create Volume` command is used in a workflow with a specific name as “vol198”. If this option is set to true, WFA planner checks during planning to see if a volume by that name exists in the given array. If the volume exists, the workflow fails during planning.
When the same command is selected more than once in a workflow, provide appropriate display names for the command instances.	The “Create, map, and protect LUNs with SnapVault” sample workflow uses the `Create Volume` command twice. However, it uses the display names as `Create Primary Volume` and `Create Secondary Volume` appropriately for the primary volume and the mirrored destination volume.

User inputs

Guidelines	Example

Names:
 	
• Start the name with the “$” character.
• Use an uppercase letter at the beginning of each word.
• Use uppercase letters for all terms and abbreviations.
• Do not use underscores.
	`$Array`
	
	`$VolumeName`

Guidelines	Example

Display names:
• Use an uppercase letter at the beginning of each word.
• Separate words with spaces.
• If inputs have specific units, specify the unit in brackets in the display name directly.
	`Volume Name`
	
	`Volume Size (MB)`

Descriptions: • Provide a meaningful description for each user input. • Provide examples when required. You should do this especially when the user input is expected to be in a specific format. The user input descriptions are displayed as tooltips for the user inputs during workflow execution.	Initiators to be added to an “iGroup”. For example, IQN or WWPN of the initiator.
Type: Select Enum as the type if you want to restrict the input to a specific set of values.	Protocol: “iscsi”, “fcp”, “mixed”

Type: Select Query as the type when the user can select from values available in the WFA cache.	$Array: QUERY type with query as follows:
	`SELECT`
	`  ip, name`
	`FROM`
	`  storage.array`

Type: Mark the user input as locked when the user input should be restricted to the values that are obtained from a query or should be restricted to only the supported Enum types.	$Array: Locked Query type: Only arrays in the cache can be selected.$Protocol: Locked Enum type with valid values as iscsi, fcp, mixed. No other value than the valid value is supported.
Type: Query TypeAdd additional columns as return values in the query when it helps the storage operator to make the right choice of user input.	$Aggregate: Provide name, total size, available size so that the operator knows the attributes before selecting the aggregate.

Guidelines	Example

Type: Query TypeSQL query for user inputs can refer to any other user inputs preceding it. This can be used to limit the results from a query based on other user inputs such as vFiler units of an array, volumes of an aggregate, LUNs in a storage virtual machine (SVM).	In the sample workflow `Create a Clustered Data ONTAP Volume`, the query for VserverName is as follows:
	
	`SELECT`
	`  vserver.name`
	`FROM`
	`  cm_storage.cluster cluster,`
	`  cm_storage.vserver vserver`
	`WHERE`
	`  vserver.cluster_id = cluster.id`
	`  AND cluster.name = '${ClusterName}'`
	`  AND vserver.type = 'cluster'`
	`ORDER BY`
	`  vserver.name ASC`
	
	The query refers to ${ClusterName}, where $ClusterName is the name of the user input preceding the $VserverName user input.

Type: Use Boolean type with values as “true, false” for user inputs that are Boolean in nature. This helps in writing internal expressions in the workflow design using the user input directly. For example, $UserInputName rather than $UserInputName == ''`Yes'.	`$CreateCIFSShare`: Boolean type with valid values as “true” or “false”
Type:For string and number type, use regular expressions in the values column when you want to validate the value with specific formats. Use regular expressions for IP address and network mask inputs.	Location-specific user input can be expressed as “[A-Z][A-Z]\-0[1-9]”. This user input accepts values such as “US-01”, “NB-02”, but not “nb-00”.
Type: For number type, a range-based validation can be specified in the values column.	For Number of LUNs to be created, the entry in the Values column is 1-20.
Group: Group related user inputs into appropriate buckets and name the group.	“Storage Details” for all storage-related user inputs. “Datastore Details” for all VMware-related user inputs.

Guidelines	Example
Mandatory: If the value of any user input is necessary for the workflow to execute, mark the user input as mandatory. This ensures that the user input screen mandatorily accepts that input from the user.	“$VolumeName” in the “Create NFS Volume” workflow.
Default value: If a user input has a default value that can work for most of the workflow executions, provide the values. This helps in allowing the user to provide fewer inputs during execution, if the default serves the purpose.	None

Constants, variables, and returns parameters

Guidelines	Example
Constants: Define constants when using a common value for defining parameters to multiple commands.	*AGGREGATE_OVERCOMMITMENT_THRESHOLD* in the `Create, map, and protect LUNs with SnapVault sample workflow`.
Constants:Names • Use an uppercase letter at the beginning of each word. • Use uppercase letters for all terms and abbreviations. • Do not use underscores. • Use uppercase letters for all letters of constant names.	*AGGREGATE_USED_SPACE_THRESHOLD* *ActualVolumeSizeInMB*
Variables: Provide a name to an object defined in one of the command parameter boxes. Variables are automatically generated names and can be changed.	None
Variables: Names Use lowercase characters for variable names.	volume1 cifs_share
Return parameters: Use return parameters when the workflow planning and execution should return some calculated or selected values during planning. The values are made available in the preview mode when the workflow is executed from a web service as well.	Aggregate: If the aggregate is selected using the resource selection logic, then the actual selected aggregate can be defined as a return parameter.

Guidelines for creating validation scripts for remote system types

You must be aware of the guidelines for creating validation scripts that are used to test the remote system types that you define in OnCommand Workflow Automation (WFA).

- The Perl script that you create must be similar to the sample script provided in the Validation Script window.
- The output of your validation script must be similar to that of the sample script.

Sample validation script

```
# Check connectivity.
# Return 1 on success.
# Return 0 on failure and set $message
sub checkCredentials {
my ($host, $user, $passwd, $protocol, $port, $timeout) = @_;
#
# Please add the code to check connectivity to $host using $protocol here.
#
return 1;
}
```

Guidelines for creating data source types

You must be aware of the guidelines for creating data source types that are used to define custom data sources for OnCommand Workflow Automation (WFA).

You can define a data source type by using one of the following methods:

- SQL: You can use the WFA SQL guidelines to define select queries from data sources based on an external database.
- SCRIPT: You can write a PowerShell script that provides the data for a specific scheme of dictionary entries.

The guidelines for creating data source types are as follows:

- You should use PowerShell language must be used to create script.
- The PowerShell script should provide the output for each dictionary entry in its current working directory.
- The data files should be named `dictionary_entry.csv`, where the name of the dictionary entry should be in lower-case characters.

  The predefined data source type that collects information from Performance Advisor uses a SCRIPT-based data source type. The output files are named `array_performance.csv` and `aggregate_performance.csv`.

- The `.csv` file should include the content in the exact order as that of the dictionary entry attributes.

A dictionary entry includes attributes in the following order: array_ip, date, day, hour, cpu_busy, total_ops_per_sec, disk_throughput_per_sec.

The PowerShell script adds data to the `.csv` file in the same order.

```
$values = get-Array-CounterValueString ([REF]$data)
Add-Content $arrayFile ([byte[]][char[]] "\N
t$arrayIP't$date't$day't$hour't$values'n")
```

- You should use Encoding to ensure that the data output from the script is loaded into the WFA cache accurately.
- You should use \N while entering a Null value in the `.csv` file.

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.