



WFA 编码准则

OnCommand Workflow Automation

NetApp

October 09, 2025

This PDF was generated from <https://docs.netapp.com/zh-cn/workflow-automation/workflows/reference-guidelines-for-variables.html> on October 09, 2025. Always check docs.netapp.com for the latest.

目录

WFA 编码准则	1
变量准则	1
PowerShell 变量	1
Perl 变量	3
缩进准则	4
注释准则	5
PowerShell 注释	5
Perl 注释	6
日志记录准则	7
PowerShell 日志记录	7
Perl 日志记录	8
错误处理准则	9
PowerShell 错误处理	9
Perl 错误处理	11
WFA 的一般 PowerShell 和 Perl 约定	12
Windows 附带的 Perl 模块	13
添加自定义 PowerShell 和 Perl 模块的注意事项	13
WFA cmdlet 和功能	13
PowerShell 和 Perl WFA 模块	14
PowerShell 模块	14
Perl 模块	14
将 PowerShell 命令转换为 Perl 时的注意事项	17
命令输入类型	17
PowerShell 语句	17
Perl 语句	17
命令定义	19
WFA 组件准则	19
WFA 中的 SQL 准则	20
WFA 功能准则	22
WFA 词典条目的准则	22
命令准则	23
工作流准则	25
为远程系统类型创建验证脚本的准则	29
创建数据源类型的准则	30

WFA 编码准则

您应了解有关创建各种组件（例如筛选器，功能，命令和工作流）的一般 OnCommand Workflow Automation （WFA）编码准则，命名约定和建议。

变量准则

创建命令或数据源类型时，您必须了解 OnCommand Workflow Automation （WFA）中 PowerShell 和 Perl 变量的准则。

PowerShell 变量

准则	示例
对于脚本输入参数： <ul style="list-style-type: none">• 使用 Pascal 案例。• 请勿使用下划线。• 请勿使用缩写词。	<code>`\$VolumeName`</code> <code>`\$AutoDeleteOptions`</code> <code>`\$size`</code>
对于脚本内部变量： <ul style="list-style-type: none">• 使用 Camel 案例。• 请勿使用下划线。• 请勿使用缩写词。	<code>`\$newVolume`</code> <code>`\$qtreename`</code> <code>`\$time`</code>
对于功能： <ul style="list-style-type: none">• 使用 Pascal 案例。• 请勿使用下划线。• 请勿使用缩写词。	<code>GetVolumeSize</code>
变量名称不区分大小写。但是，为了提高可读性，不对同一名称使用不同的大写。	<code>`\$variable`</code> 与 <code>`\$Variable`</code> 相同。
变量名称应使用纯英文，并且应与脚本的功能相关。	请使用 <code>`\$name`</code> ，而不是 <code>`\$a`</code>
显式声明每个变量的数据类型。	字符串名称 大小
请勿使用特殊字符（! @ # & % ， .）和空格。	无

准则	示例
请勿使用 PowerShell 保留的关键字。	无
通过先放置必填参数，然后再放置可选参数来对输入参数进行分组。	<pre>param ([parameter(Mandatory=\$true)] [string]\$Type, [parameter(Mandatory=\$true)] [string]\$Ip, [parameter(Mandatory=\$false)] [string]\$VolumeName)</pre>
使用 `HelpMessage` 标注和有意义的帮助消息对所有输入变量进行注释。	<pre>[parameter(Mandatory=\$false, HelpMessage="LUN to map")] [string]\$LUNName</pre>
请勿使用 "Filer" 作为变量名称；请改用 "Array"。	无
如果参数获得枚举值，请使用 `ValidateSet` 标注。此操作会自动转换为参数的 Enum 数据类型。	<pre>[parameter(Mandatory=\$false, HelpMessage="Volume state")] [ValidateSet("online", "offline", "restricted")] [string]\$State</pre>
向以 "_capacity" 结尾的参数添加别名，以指示此参数的容量类型。	<p>"Create Volume" 命令使用别名，如下所示：</p> <pre>[parameter(Mandatory=\$false, HelpMessage="Volume increment size in MB")] [Alias("AutosizeIncrementSize_Capacity")] [int]\$AutosizeIncrementSize</pre>

准则	示例
向以 " 密码 " 结尾的参数添加别名，以指示此参数的密码类型。	<pre>param ([parameter(Mandatory=\$false, HelpMessage="In order to create an Active Directory machine account for the CIFS server or setup CIFS service for Storage Virtual Machine, you must supply the password of a Windows account with sufficient privileges")] [Alias("Pwd_Password")] [string]\$ADAdminPassword)</pre>

Perl 变量

准则	示例
对于脚本输入参数： <ul style="list-style-type: none"> • 使用 Pascal 案例。 • 请勿使用下划线。 • 请勿使用缩写词。 	<pre>`\$VolumeName` `\$AutoDeleteOptions` `\$size`</pre>
不要使用脚本内部变量的缩写。	<pre>`\$new_volume` `\$qtree_name` `\$time`</pre>
请勿使用缩写词来表示函数。	<pre>get_volume_size</pre>
变量名称区分大小写。为了提高可读性，不应在同一名称使用不同的大写。	<pre>`\$variable` 与 `\$Variable` 不同。`</pre>
变量名称应使用纯英文，并且应与脚本的功能相关。	请使用 <code>`\$name`</code> ，而不是 <code>`\$a`</code>
首先放置必需参数，然后再放置可选参数，对输入参数进行分组。	无

准则	示例
在 GetOptions 函数中，显式声明输入参数的每个变量的数据类型。	<pre>GetOptions ("Name=s"=>\\$Name, "Size=i"=>\\$Size)</pre>
请勿使用 "Filer" 作为变量名称；请改用 "Array"。	无
Perl 不包含枚举值的 <code>ValidateSet</code> 标注。如果参数获得枚举值，请使用显式 <code>if</code> 语句。	<pre>if (defined\$SpaceGuarantee&&!(\$SpaceG uaranteeeq'none '</pre>
	<code>\$SpaceGuaranteeeq'volume'</code>
	<pre>\$SpaceGuaranteeeq'file')) { die'Illegal SpaceGuarantee argument: \'\$.SpaceGuarantee.'\'; } ----</pre>
所有 Perl WFA 命令都必须使用 <code>strict pragma</code> 来阻止对变量，引用和子例程使用不安全的构造。	<pre>use strict; # the above is equivalent to use strictvars; use strictsubs; use strictrefs;</pre>
<p>所有 Perl WFA 命令都必须使用以下 Perl 模块：</p> <ul style="list-style-type: none"> • <code>getopt</code> 用于指定输入参数。 • <code>WFAUtil</code> 此选项用于提供用于命令日志记录，报告命令进度，连接到阵列控制器等的实用程序功能。 	<pre>use Getopt::Long; use NaServer; use WFAUtil;</pre>

缩进准则

在为 OnCommand Workflow Automation （WFA）编写 PowerShell 或 Perl 脚本时，您必须了解缩进准则。

准则	示例
一个选项卡等于四个空空格。	
使用制表符和花括号显示块的开始和结束。	<p>PowerShell 脚本</p> <pre> if (\$pair.length-ne 2) { throw "Got wrong input data" } </pre> <p>Perl 脚本</p> <pre> if (defined \$MaxDirectorySize) { # convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024; } </pre>
在操作集或代码块之间添加空白行。	<pre> \$options=\$option.trim(); \$pair=\$option.split(" "); Get-WFALogger -Info -messages \$("split options: "+ \$Pair) </pre>

注释准则

您必须了解适用于 OnCommand Workflow Automation （ WFA ） 的脚本中有关 PowerShell 和 Perl 注释的准则。

PowerShell 注释

准则	示例
使用 # 字符作为单行注释。	<pre># Single line comment \$options=\$option.trim();</pre>
使用 # 字符作为行尾注释。	<pre>\$options=\$option.trim(); # End of line comment</pre>
使用 <# 和 #> 字符作为块注释。	<pre><# This is a block comment #> \$options=\$option.trim();</pre>

Perl 注释

准则	示例
使用 # 字符表示单行注释。	<pre># convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024;</pre>
使用 # 字符作为行尾注释。	<pre>my \$MaxDirectorySizeBytes = \$MaxDirect orySize * 1024 * 1024; # convert to Bytes</pre>

准则	示例
在每行中使用 # 字符，并在开头和结尾使用空 # 为多行注释创建注释边框。	<pre># # This is a multi-line comment. Perl 5, unlike # Powershell, does not have direct support for # multi-line comments. Please use a '#' in every line # with an empty '#' at the beginning and end to create # a comment border #</pre>
请勿在 WFA 命令中包含注释代码和失效代码。但是，出于测试目的，您可以使用 Plain Old Documentation（POD）机制注释掉代码。	<pre>=begin comment # Set deduplication if (defined \$Deduplication && \$Deduplication eq "enabled") { \$wfaUtil- >sendLog("Enabling Deduplication"); } =end comment =cut</pre>

日志记录准则

在为 OnCommand Workflow Automation（WFA）编写 PowerShell 或 Perl 脚本时，您必须了解有关日志记录的准则。

PowerShell 日志记录

准则	示例
使用 Get-WFALogger cmdlet 进行日志记录。	<pre>Get-WFALogger -Info -message "Creating volume"</pre>

准则	示例
记录需要与 Data ONTAP ， VMware 和 PowerCLI 等内部软件包交互的每个操作。所有日志消息均可在工作流的执行状态历史记录的执行日志中找到。	无
记录传递到内部软件包的每个相关参数。	无
使用 Get-WFALogger cmdlet 时，请根据使用环境使用适当的日志级别。-Info ， -Error ， -warn 和 -Debug 是各种可用的日志级别。如果未指定日志级别，则默认日志级别为调试。	无

Perl 日志记录

准则	示例
使用 WFAUtil sendLog 进行日志记录。	<pre>my wfa_util = WFAUtil->new(); eval { \$wfa_util->sendLog('INFO', "Connecting to the cluster: \$DestinationCluster"); }</pre>
记录需要与命令外部任何内容交互的每个操作，例如 Data ONTAP ， VMware 和 WFA 。使用 WFAUtil sendLog 例程创建的所有日志消息都存储在 WFA 数据库中。这些日志消息可用于已执行的工作流和命令。	无
记录传递给调用的例程的每个相关参数。	无
使用适当的日志级别。 -Info ， -Error ， -warn 和 -Debug 是各种可用的日志级别。	无

准则	示例
在 -Info 级别登录时，请准确而简洁。请勿在日志消息中指定实施详细信息，例如类名称和功能名称。请以简体中文描述确切步骤或确切错误。	<p>以下代码片段显示了一条好消息和一条坏消息的示例：</p> <pre>\$wfa_util->sendLog('WARN', "Removing volume: '.\$VolumeName); # Good Message</pre> <pre>\$wfa_util->sendLog('WARN', 'Invoking volume- destroy ZAPI: '.\$VolumeName); # Bad message</pre>

错误处理准则

您必须了解为 OnCommand Workflow Automation （WFA）编写 PowerShell 或 Perl 脚本时的错误处理准则。

PowerShell 错误处理

准则	示例
<p>PowerShell 运行时添加到 cmdlet 的常见参数包括错误处理参数，例如 ErrorAction 和 WarningAction：</p> <ul style="list-style-type: none"> • ErrorAction 参数用于确定 cmdlet 应如何对命令中的非终止错误做出响应。 • WarningAction 参数用于确定 cmdlet 应如何对命令发出的警告做出响应。 • "ErrorAction" 和 "WarningAction" 参数的有效值为 "stop"，"SilentlyContinue"，"Inquire" 和 "Continue"。 <p>有关详细信息，您可以在 PowerShell CLI 中使用 Get-Help About_CommonParameters 命令。</p>	<p>ErrorAction：以下示例显示了如何将非终止错误作为终止错误进行处理：</p> <pre>New-NcIgroup-Name \$IgroupName- Protocol \$Protocol-Type\$OSType- ErrorActionstop</pre> <p>WarningAction</p> <pre>New-VM-Name \$VMName-VM \$SourceVM- DataStore\$DataStoreName- VMHost\$VMHost- WarningActionSilentlyContinue</pre>

准则	示例
<p>如果传入异常的类型未知，请使用常规 <code>"try/catch"</code> 语句。</p>	<pre>try { "In Try/catch block" } catch { "Got exception" }</pre>
<p>如果已知传入异常的类型，请使用特定的 <code>"try/catch"</code> 语句。</p>	<pre>try { "In Try/catch block" } catch[System.Net.WebException], [System.IO. IOException] { "Got exception" }</pre>
<p>使用 <code>"finally"</code> 语句释放资源。</p>	<pre>try { "In Try/catch block" } catch { "Got exception" } finally { "Release resources" }</pre>

准则	示例
使用 PowerShell 自动变量访问有关异常的信息。	<pre>try { Get-WFALogger -Info -message \$("Creating Ipspace: " + \$Ipspace) New-NaNetIpspace-Name \$Ipspace } catch { Throw "Failed to create Ipspace. Message: " + \$_.Exception.Message; }</pre>

Perl 错误处理

准则	示例
Perl 不支持 try/catch 块的原生 语言。使用评估块检查和处理错误。请尽可能小地保留评估块。	<pre>eval { \$wfa_util->sendLog('INFO', "Quiescing the relationship : \$DestinationCluster://\$Destination Vserver /\$DestinationVolume"); \$server->snapmirror_quiesce('destination-vserver' => \$DestinationVserver, 'destination-volume' => \$DestinationVolume); \$wfa_util->sendLog('INFO', 'Quiesce operation started successfully.');</pre> <pre>}; \$wfa_util->checkEvalFailure("Failed to quiesce the SnapMirror relationship \$DestinationCluster://\$Destination Vserver /\$DestinationVolume", \$@);</pre>

WFA 的一般 PowerShell 和 Perl 约定

您必须了解 WFA 中用于创建与现有脚本一致的脚本的某些 PowerShell 和 Perl 约定。

- 使用有助于阐明您希望脚本执行的操作的变量。
- 编写可读代码，无需注释即可理解。
- 使脚本和命令尽可能简单。
- 对于 PowerShell 脚本：
 - 尽可能使用 cmdlet 。
 - 如果没有可用的 cmdlet ，请调用 .NET 代码。
- 对于 Perl 脚本：

- 以换行符结尾的 `ie` 语句。

如果没有换行符，则会打印脚本行号，这对于调试 WFA 执行的 Perl 命令不有用。

- 在 `"getopt"` 模块中，将命令的字符串参数设置为必填项。

Windows 附带的 Perl 模块

某些 Perl 模块与适用于 OnCommand Workflow Automation （WFA）的 Windows Active state Perl 分发版捆绑在一起。只有在这些 Perl 模块与 Windows 捆绑在一起的情况下，您才能在 Perl 代码中使用这些 Perl 模块来编写命令。

下表列出了与 Windows for WFA 捆绑在一起的 Perl 数据库模块。

数据库模块	Description
DBD : : mysql	Perl5 数据库接口驱动程序，用于连接到 MySQL 数据库。
尝试 : : 微型	利用评估块最大限度地减少常见错误。
XML : : libxml	libxml2 的接口，可通过 DOM，SAX 和 XIReader 接口提供 XML 和 HTML 解析程序。
DBD : : Cassandra	适用于 Cassandra 的 Perl5 数据库接口驱动程序，使用 CQL3 查询语言。

添加自定义 PowerShell 和 Perl 模块的注意事项

在将自定义 PowerShell 和 Perl 模块添加到 OnCommand Workflow Automation （WFA）之前，您必须了解某些注意事项。通过自定义 PowerShell 和 Perl 模块，您可以使用自定义命令创建工作流。

- 在执行 WFA 命令期间，所有自定义 PowerShell 模块都会添加到 WFA 安装目录 `/wally/modules` 中。
- 添加到 `WFA/perl` 目录的所有自定义 Perl 模块都包含在 `@INC` 库中。
- 在 WFA 备份操作中，不会备份自定义 PowerShell 和 Perl 模块。
- WFA 还原操作不会还原自定义 PowerShell 和 Perl 模块。

您必须手动备份自定义 PowerShell 和 Perl 模块，才能将其复制到新的 WFA 安装中。

模块目录中的文件夹名称必须与模块名称相同。

WFA cmdlet 和功能

OnCommand Workflow Automation （WFA）提供了多个 PowerShell cmdlet 以及可在

WFA 命令中使用的 PowerShell 和 Perl 功能。

您可以使用以下 PowerShell 命令查看 WFA 服务器提供的所有 PowerShell cmdlet 和功能：

- `Get-Command -Module WFAWrapper`
- `Get-Command -Module WFA`

您可以在 `WFAUtil.pm` 模块中查看 WFA 服务器提供的所有 Perl 功能。通过 WFA 帮助模块的帮助部分 WFA PowerShell cmdlet 帮助和 WFA Perl 方法帮助，可以访问所有 PowerShell cmdlet 和功能以及 Perl 功能。

PowerShell 和 Perl WFA 模块

要为工作流编写脚本，您必须了解适用于 OnCommand Workflow Automation （WFA）的 PowerShell 或 Perl 模块。


PowerShell 模块

准则	示例
只要工具包可用，就可以使用 Data ONTAP PS 工具包调用 API 。	Add VLAN 命令使用的工具包如下所示： <code>Add-NaNetVlan-Interface \$Interface-VLANs \$VlanID</code>
如果 Data ONTAP PS 工具包中没有 cmdlet ，请使用 <code>invoke-SSH</code> 命令在 Data ONTAP 上调用命令行界面。	<code>Invoke-NaSsh-Name \$ArrayName-Command "ifconfig -a" - Credential \$Credentials</code>

Perl 模块

NaServer 模块用于 WFA 命令。通过 NaServer 模块，可以调用 Data ONTAP API ， 这些 API 用于主动管理 Data ONTAP 系统。

准则	示例
<p>只要 NetApp 易管理性 SDK 可用，就可以使用 NaServer 模块调用 API。</p>	<p>以下示例显示了如何使用 NaServer 模块执行恢复 SnapMirror 操作：</p> <pre> eval { \$wfa_util->sendLog('INFO', "Connecting to the cluster: \$DestinationCluster"); my \$server = \$wfa_util- >connect(\$DestinationClusterIp, \$DestinationVserver); my \$sm_info = \$server- >snapmirror_get('destination-vserver' => \$DestinationVserver, 'destination-volume' => \$DestinationVolume); my \$sm_state = \$sm_info- >{'attributes'}->{'snapmirror- info'}->{'mirror-state'}; my \$sm_status = \$sm_info- >{'attributes'}->{'snapmirror- info'}->{'relationship-status'}; \$wfa_util->sendLog('INFO', "SnapMirror relationship is \$sm_state (\$sm_status)"); if (\$sm_status ne 'quiesced') { \$wfa_util->sendLog('INFO', 'The status needs to be quiesced to resume transfer.');</pre>
16	<pre> } else { my \$result = \$server- >snapmirror_resume('destination-vserver' => \$DestinationVserver, 'destination-volume' => \$DestinationVolume); \$wfa_util->sendLog('INFO', "Result of resume: \$result"); } }</pre>

准则	示例
<p>如果 Data ONTAP API 不可用，请使用 <code>executeSystemCli</code> 实用程序方法调用 Data ONTAP 命令行界面。</p> <div>  <div>不支持 <code>executeSystemCli</code>，目前仅适用于 7- 模式 Data ONTAP。</div> </div>	无

将 PowerShell 命令转换为 Perl 时的注意事项

将 PowerShell 命令转换为 Perl 时，您必须了解一些重要注意事项，因为 PowerShell 和 Perl 具有不同的功能。

命令输入类型

OnCommand Workflow Automation (WFA) 允许 workflow 设计人员在定义命令时使用数组和哈希作为命令的输入。使用 Perl 定义命令时，不能使用这些输入类型。如果希望 Perl 命令接受数组和哈希输入，可以在设计器中将输入定义为字符串。然后，命令定义可以解析输入，并根据需要传递该输入以创建数组或哈希。输入的问题描述 说明了输入应采用的格式。

```
my @input_as_array = split(',', $InputString); #Parse the input string of
format val1,val2 into an array

my %input_as_hash = split /[:=]/, $InputString; #Parse the input string of
format key1=val1;key2=val2 into a hash.
```

PowerShell 语句

以下示例显示了如何将阵列输入传递到 PowerShell 和 Perl 中。这些示例介绍了输入 `CronMonth`，其中指定了 cron 作业计划运行的月份。有效值为整数 -1 到 11。值 -1 表示计划每月执行一次。任何其他值表示特定月份，0 表示 1 月，11 表示 12 月。

```
[parameter(Mandatory=$false, HelpMessage="Months in which the schedule
executes. This is a comma separated list of values from 0 through 11.
Value -1 means all months.")]
[ValidateRange(-1, 11)]
[array]$CronMonths,
```

Perl 语句

```

GetOptions(
    "Cluster=s"           => \$Cluster,
    "ScheduleName=s"      => \$ScheduleName,
    "Type=s"              => \$Type,
    "CronMonths=s"        => \$CronMonths,
) or die 'Illegal command parameters\n';

sub get_cron_months {
    return get_cron_input_hash('CronMonths', \$CronMonths, 'cron-month',
-1,
        11);
}

sub get_cron_input_hash {
    my $input_name = shift;
    my $input_value = shift;
    my $zapi_element = shift;
    my $low = shift;
    my $high = shift;
    my $exclude = shift;

    if (!defined $input_value) {
        return undef;
    }

    my @values = split(',', $input_value);

    foreach my $val (@values) {
        if ($val !~ /^[+-]?[0-9]+$/) {
            die
                "Invalid value '$input_value' for $input_name: $val must
be an integer.\n";
        }
        if ($val < $low || $val > $high) {
            die
                "Invalid value '$input_value' for $input_name: $val must
be from $low to $high.\n";
        }
        if (defined $exclude && $val == $exclude) {
            die
                "Invalid value '$input_value' for $input_name: $val is not
valid.\n";
        }
    }
    # do something
}

```

命令定义

要实现相同的功能，可能需要在 Perl 中将 PowerShell 中使用管道运算符的单行表达式扩展为多个语句块。下表显示了其中一个 wait 命令的示例。

PowerShell 语句	Perl 语句
<pre># Get the latest job which moves the specified volume to the specified aggregate. \$job = Get-NcJob -Query \$query</pre>	<pre>where {\$_.JobDescription -eq "Split" + \$VolumeCloneName}</pre>
Select-Object -First 1 ----	<pre>my \$result = \$server- >job_get_iter('query' => {'job-type' => 'VOL_CLONE_SPLIT'}, 'desired-attributes' => { 'job-type' => '', 'job-description' => '', 'job-progress' => '', 'job-state' => '' }); my @jobarray; for my \$job (@{ \$result- >{'attributes-list'}}) { my \$description = \$job->{'job- description'}; if(\$description =~ /\$VolumeCloneName/) { push(@jobarray, \$job) } }</pre>

WFA 组件准则

您必须了解使用 Workflow Automation 组件的准则。

WFA 中的 SQL 准则

您必须了解使用 OnCommand Workflow Automation 中的 SQL （WFA）为 WFA 写入 SQL 查询的准则。

SQL 在 WFA 中的以下位置使用：

- 用于填充用户输入以供选择的 SQL 查询
- 用于创建筛选器以筛选特定词典条目类型的对象的 SQL 查询
- 实践数据库中的表中的静态数据
- SQL 类型的自定义数据源类型，其中必须从自定义配置管理数据库（CMDB）等外部数据源提取数据。
- SQL 查询预留和验证脚本

准则	示例
SQL 保留关键字必须为大写字符。	<pre>SELECT vsERVER.name FROM cm_storage.vserver vserver</pre>
表和列名称必须为小写字符。	表：聚合 列：used_space_MB
使用下划线（_）字符分隔词语。不允许使用空格。	array_performance
表名称以单个形式定义。表是一个或多个条目的集合。	"函数`"，而非"函数`"
在选择查询中使用具有有意义名称的表别名。	<pre>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</pre>

准则	示例
<p>如果您必须在筛选器查询或用户查询中引用筛选器输入参数或用户输入参数，请使用以下语法： "\${inputVariableName}"。您也可以使用此语法引用预留脚本和验证脚本中的命令定义参数。</p>	<pre>SELECT volume.name AS Name, aggregate.name as Aggregate, volume.size_mb AS 'Total Size (MB)', volume.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 = volume.vserver_id AND vserver.name = '\${VserverName}' AND cluster.name = '\${ClusterName}' ORDER BY volume.name ASC</pre>
<p>对复杂查询使用注释。查询中支持的一些注释模式如下：</p> <ul style="list-style-type: none">• " - - "，直到行尾为止 <p>在此注释模式下，第二个连字符后面必须有空格。</p> <ul style="list-style-type: none">• 从 " # " 字符到行尾• 从 " /* " 到以下 " */ " 序列	<pre>/* 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</pre>

WFA 功能准则

您可以创建一些函数来将常用且更复杂的逻辑封装在命名函数中，然后在 OnCommand Workflow Automation （WFA）中将该函数重复用作命令参数值或筛选参数值。

准则	示例
使用 Camel 大小写作为函数名称。	计算卷大小
变量名称应使用纯英文，并与函数的功能相关。	splitByDelimiter
请勿使用缩写词。	calculateVolumeSize ， <i>not</i> calcVolSize
函数使用 MVFLEX 表达式语言（MVEL）定义。	无
应根据官方 Java 编程语言准则指定函数定义。	无

WFA 词典条目的准则

您必须了解在 OnCommand Workflow Automation （WFA）中创建词典条目的准则。

准则	示例
词典条目名称只能包含字母数字字符和下划线。	集群许可证 switch_23
词典条目名称必须以大写字母开头。名称中的每个词以大写字母开头，每个词以下划线（_）分隔。	Volume Array_License
词典条目属性名称不应包含词典条目的名称。	无
词典条目中的属性和引用必须使用小写字母。	聚合， size_mb
使用下划线分隔词语。不允许使用空格。	resource_pool
词典条目不能包含来自不同方案的引用。如果某个词典条目要求交叉引用不同方案中的某个对象，请确保所引用对象的所有自然键都出现在该词典条目中。	array_Performance 词典条目要求使用数组词典条目的所有自然键作为直接属性。
为属性使用适当的数据类型。	无
对于大小或空间相关属性，请使用长数据类型。	storage.Volume 词典条目中的 size_MB 和 available_size_MB

准则	示例
如果属性具有一组固定值，请使用 Enum 。	storage.Volume 词典条目中的 raid_type
如果数据源为某个属性或引用提供了值，请将该属性或引用的 "设置为缓存" 设置为 true 。对于 Active IQ Unified Manager 数据源，如果数据源可以为它提供值，请添加可缓存属性。	无
如果为此属性或引用提供值的数据源可以返回 NULL ，则将 "can be Null" 设置为 true 。	无
为每个属性和参考提供有意义的问题描述。设计工作流时，问题描述 将显示在命令详细信息中。	无
请勿在词典条目中使用 "id" 作为属性名称。它是为内部 WFA 使用预留的。	无

- 相关信息 *

参考学习材料

命令准则

您必须了解在 OnCommand Workflow Automation （ WFA ） 中创建命令的准则。

准则	示例
使用可轻松识别的命令名称。	创建 qtrees
请使用空格分隔单词，并且每个词都必须以大写字母开头。	创建卷
提供问题描述 以说明命令的功能，包括可选参数的预期结果。	无
默认情况下，标准命令的超时时间为 600 秒。创建命令时会设置默认超时时间。只有在命令可能需要较长时间才能完成时，才更改默认值。	Create Volume 命令
对于长时间运行的操作，请创建两个命令：一个用于调用长时间运行的操作，另一个用于定期报告操作进度。第一个命令应为 Standard Execution 命令类型，第二个命令应为 wait for condition command 类型。	Create VSM and Wait for VSM commands.

准则	示例
为 wait for condition 命令名称添加前缀 "wait`"，以便于识别。	等待 CM 卷移动
对 "wait for condition`" 命令使用适当的等待间隔。指定值用于控制执行轮询命令以检查长时间运行的操作是否已完成的间隔。	wait for VSM 命令的 60 秒采样间隔
对于 wait for condition 命令，请根据长时间运行操作的预期完成时间使用适当的超时时间。如果操作涉及通过网络传输数据，则预期时间可能会显著延长。	完成 VSM 基线传输可能需要数天时间。因此，指定的超时为 6 天。

字符串表示

命令的字符串表示形式可在规划和执行期间在工作流设计中显示命令的详细信息。命令的字符串表示只能使用命令参数。

准则	示例
避免使用不具有任何值的属性。不带值的属性显示为 NA。	volname 10.68.66.212[NA]aggr1/testVol7.
使用以下分隔符分隔字符串表示形式中的不同条目： []， /：	` ArrayName[ArrayIp]_`
以字符串表示形式为每个值提供有意义的标签。	` 卷名称 = 卷名称 _`

命令定义语言

可以使用以下受支持的脚本语言编写命令：

- PowerShell
- Perl

命令参数定义

命令参数通过名称，问题描述，类型，参数的默认值以及参数是否为必填项进行说明。参数类型可以是字符串，布尔值，整数，长，双，枚举，日期时间，容量，阵列，可哈希，密码或 XIDocument。虽然大多数类型的值都是直观的，但 Array 和 Hastable 的值应采用如下表所述的特定格式：

准则	示例
确保 Array 输入类型的值是一个值列表，以逗号分隔。	<pre>[parameter(Mandatory=\$false, HelpMessage="Months in which the schedule executes.")] [array]\$CronMonths</pre> <p>输入的传递方式如下： 0 ， 3 ， 6 ， 9</p>
确保可哈希表输入类型的值为 key=value 对的列表，以分号分隔。	<pre>[parameter(Mandatory=\$false, HelpMessage="Volume names and size (in MB)")] [hashtable]\$VolumeNamesAndSize</pre> <p>输入的传递方式如下： volume1=100 ； Volume2=250 ； Volume3=50</p>

workflow 准则

您必须了解为 OnCommand Workflow Automation （ WFA ） 创建或修改预定义 workflow 的准则。

一般准则

准则	示例
为 workflow 命名，使其反映存储操作员执行的操作。	创建 CIFS 共享
对于 workflow 名称，请将第一个词的首字母以及对象的每个词大写。首字母大写表示缩写词和首字母缩略语。	<p>Volume</p> <p>qtree</p> <p>创建集群模式 Data ONTAP qtree CIFS 共享</p>
有关 workflow 说明，请包括 workflow 的所有重要步骤，包括任何前提条件， workflow 结果或执行的条件方面。	请参见示例 workflow 的问题描述 在集群模式 Data ONTAP 存储上创建 VMware NFS 数据存储库，其中包括前提条件。
仅当 workflow 已准备好投入生产并可显示在门户页面中时，才将 "Ready for 生产" 设置为 true 。	无

准则	示例
默认情况下，将 "consider reserved elements" 设置为 true。预览要执行的工作流时，WFA 规划器会考虑预留的所有对象以及缓存数据库中的现有对象。如果此选项设置为 true，则在规划特定工作流时会考虑其他计划工作流或并行执行的工作流的影响。	<ul style="list-style-type: none"> 场景 1 <p>工作流 1 将创建一个卷，并计划在一周后执行。工作流 2 会在搜索的卷中创建 qtree 或 LUN，如果工作流 2 在一天后执行，则应关闭工作流 2 的 "Consider reserved elements"，以防止其考虑要在一周内创建的卷。</p> <ul style="list-style-type: none"> 场景 2 <p>工作流 1 使用 Create Volume 命令。如果计划的工作流 2 占用聚合的 100 GB 空间，则工作流 1 必须在规划期间考虑工作流 2 的要求。</p>
默认情况下，"启用元素存在验证" 设置为 true。	<ul style="list-style-type: none"> 场景 1 <p>如果您创建的工作流首先使用命令 Remove Volume Only if the volume exists 按名称删除卷，然后使用另一个命令（例如 Create Volume 或 Clone Volume，）重新创建卷，则此工作流不应使用此标志。Create volume 命令无法使用删除卷的效果，从而导致工作流失败。</p> <ul style="list-style-type: none"> 场景 2 <p>在特定名称为 "vol198" 的工作流中使用 Create Volume 命令。</p> <p>如果此选项设置为 true，则 WFA 规划器会在规划期间检查给定阵列中是否存在使用该名称的卷。如果卷存在，则在规划期间工作流将失败。</p>
如果在工作流中多次选择同一命令，请为命令实例提供适当的显示名称。	"使用 SnapVault 创建，映射和保护 LUN" 示例工作流使用 创建卷 命令两次。但是，对于主卷和镜像目标卷，它会相应地使用显示名称 Create Primary Volume 和 Create Secondary Volume。

用户输入

准则	示例
<p>名称：</p> <ul style="list-style-type: none"> 名称以 "` \$ " 字符开头。 在每个词的开头使用大写字母。 所有术语和缩写词均使用大写字母。 请勿使用下划线。 	<p>` \$Array`</p> <p>` \$VolumeName`</p>
<p>显示名称：</p> <ul style="list-style-type: none"> 在每个词的开头使用大写字母。 使用空格分隔词语。 如果输入包含特定单位，请直接在显示名称的方括号中指定单位。 	<p>卷名称</p> <p>卷大小 (MB)</p>
<p>说明：</p> <ul style="list-style-type: none"> 为每个用户输入提供有意义的问题描述。 根据需要提供示例。 <p>您尤其应在用户输入应采用特定格式时执行此操作。</p> <p>用户输入说明将在工作流执行期间显示为用户输入的工具提示。</p>	<p>要添加到 "`iGroup` " 的启动程序。例如，启动程序的 IQN 或 WWPN。</p>
<p>Type：如果要限制输入为一组特定值，请选择 Enum 作为类型。</p>	<p>协议： "`iSCSI` "， "`FCP` "， " mi "</p>
<p>Type：选择 Query 作为用户可以从 WFA 缓存中的可用值中选择的类型。</p>	<p>\$Array：查询类型，如下所示：</p> <pre>SELECT ip, name FROM storage.array</pre>
<p>type：如果用户输入应限制为从查询获取的值或仅限于受支持的枚举类型，请将用户输入标记为 locked。</p>	<p>\$Array：Locked Query type：只能选择缓存中的阵列。\$Protocol：Locked Enum 类型，有效值为 iSCSI，FCP，混合。不支持有效值以外的其他值。</p>
<p>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.</p>	<p>\$aggregate：提供名称，总大小和可用大小，以便操作员在选择聚合之前了解属性。</p>

准则	示例
<p>type：用于用户输入的查询 TypeSQL 查询可以引用之前的任何其他用户输入。这可用于根据其他用户输入限制查询结果，例如，阵列的 vFiler 单元，聚合的卷，Storage Virtual Machine（SVM）中的 LUN。</p>	<p>在示例工作流 创建集群模式 Data ONTAP 卷，VserverName 的查询如下：</p> <pre>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</pre> <p>查询引用的是 <code>\${clustername}</code>，其中 <code>\$clustername</code> 是 <code>\$VserverName</code> 用户输入之前的用户输入名称。</p>
<p>Type：对于本质上为布尔值的用户输入，请使用布尔类型，值为 <code>"true" , "false"</code>。这有助于在工作流设计中直接使用用户输入编写内部表达式。例如，<code>\$UserInputName</code> 而不是 <code>\$UserInputName = "Yes"</code>。</p>	<p><code>`\$CreateCIFSShare`</code>：有效值为 <code>"true"</code> 或 <code>"false"</code> 的布尔类型</p>
<p>type：对于字符串和数字类型，如果要使用特定格式验证值，请在值列中使用正则表达式。</p> <p>使用正则表达式输入 IP 地址和网络掩码。</p>	<p>位置特定的用户输入可以表示为 <code>"[A-Z][A-Z]\-[01-9]"</code>。此用户输入可接受 <code>"US-01"</code>，<code>"NNB-02"</code> 等值，但不能接受 <code>"NB - 00"</code> 等值。</p>
<p>type：对于数字类型，可以在值列中指定基于范围的验证。</p>	<p>要创建的 LUN 数量，"值"列中的条目为 1-20。</p>
<p>Group：将相关用户输入分组到相应的存储分段中并为组命名。</p>	<p>所有与存储相关的用户输入均为 <code>"s存储详细信息"</code>。所有与 VMware 相关的用户输入均为 <code>datastore Details</code>。</p>
<p>必填：如果要执行工作流，需要任何用户输入的值，请将用户输入标记为必填。这样可以确保用户输入屏幕能够强制接受用户输入的内容。</p>	<p><code>"创建 NFS 卷"</code> 工作流中的 <code>"\$VolumeName"</code>。</p>

准则	示例
默认值：如果用户输入的默认值可用于大多数工作流执行，请提供这些值。这有助于用户在执行期间提供较少的输入，前提是默认值可以满足此目的。	无

常量，变量和返回参数

准则	示例
常量：使用通用值为多个命令定义参数时定义常量。	使用 <code>SnapVault</code> 示例工作流创建，映射和保护 LUN 中的 <code>aggregate_overcommit_threshold</code> 。
常量：名称 <ul style="list-style-type: none"> • 在每个词的开头使用大写字母。 • 所有术语和缩写词均使用大写字母。 • 请勿使用下划线。 • 所有常量名称的字母均使用大写字母。 	<code>aggregate_used_space_threshold</code> <code>ActualVolumeSizeInMB</code>
变量：为在其中一个命令参数框中定义的对象提供一个名称。变量是自动生成的名称，可以进行更改。	无
Variables：名称对变量名称使用小写字符。	卷 1 <code>cifs_share</code>
返回参数：如果工作流规划和执行在规划期间应返回一些已计算或选定的值，请使用返回参数。如果也从 Web 服务执行工作流，则这些值将在预览模式下可用。	聚合：如果使用资源选择逻辑选择聚合，则可以将实际选定聚合定义为返回参数。

为远程系统类型创建验证脚本的准则

您必须了解创建验证脚本的准则，这些脚本用于测试您在 OnCommand Workflow Automation（WFA）中定义的远程系统类型。

- 您创建的 Perl 脚本必须与验证脚本窗口中提供的示例脚本类似。
- 验证脚本的输出必须与示例脚本的输出类似。

验证脚本示例

```
# 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;
}
```

创建数据源类型的准则

您必须了解用于定义 OnCommand Workflow Automation （ WFA ） 自定义数据源的数据源类型的创建准则。

您可以使用以下方法之一定义数据源类型：

- SQL：您可以使用 WFA SQL 准则基于外部数据库定义来自数据源的选择查询。
- 脚本：您可以编写一个 PowerShell 脚本，为词典条目的特定方案提供数据。

创建数据源类型的准则如下：

- 您应使用 PowerShell 语言创建脚本。
- PowerShell 脚本应为其当前工作目录中的每个词典条目提供输出。
- 数据文件应命名为 `dictional_entry.csv`，其中词典条目的名称应使用小写字母。

从 Performance Advisor 收集信息的预定义数据源类型使用基于脚本的数据源类型。输出文件名为 `array_performance.csv` 和 `aggregate_performance.csv`。

- `.csv` 文件中的内容应与词典条目属性中的内容完全相同。

词典条目按以下顺序包括属性：`array_IP`，日期，日期，小时，`CPU_busy`，总操作数每秒，磁盘吞吐量每秒

PowerShell 脚本会按相同顺序将数据添加到 `.csv` 文件。

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

- 您应使用编码来确保脚本输出的数据准确加载到 WFA 缓存中。
- 在 `.csv` 文件中输入空值时，应使用 `\N`。

版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。