



WFA에 대한 코딩 지침

OnCommand Workflow Automation 5.0

NetApp
April 19, 2024

목차

WFA에 대한 코딩 지침	1
변수에 대한 지침	1
들여쓰기를 위한 지침	4
코멘트에 대한 지침	5
로깅 지침	7
오류 처리 지침	9
WFA에 대한 일반 PowerShell 및 Perl 규칙	12
사용자 지정 PowerShell 및 Perl 모듈을 추가할 때의 고려 사항	13
WFA cmdlet 및 기능	14
PowerShell 및 Perl WFA 모듈	14
PowerShell 명령을 Perl로 변환할 때의 고려 사항	17
WFA 구성 요소에 대한 지침입니다	19

WFA에 대한 코딩 지침

일반 WFA(OnCommand Workflow Automation) 코딩 지침, 명명 규칙, 필터, 기능, 명령, 워크플로우 등과 같은 다양한 구성 요소를 만드는 방법에 대한 권장 사항을 이해해야 합니다.

변수에 대한 지침

명령 또는 데이터 소스 유형을 생성할 때 WFA(OnCommand Workflow Automation)에서 PowerShell 및 Perl 변수에 대한 지침을 숙지해야 합니다.

PowerShell 변수

지침	예
스크립트 입력 매개 변수의 경우: <ul style="list-style-type: none">• 파스칼 케이스를 사용합니다.• 밑줄을 사용하지 마십시오.• 약어는 사용하지 마십시오.	'\$VolumeName' '\$AutoDeleteOptions' '\$Size'
스크립트 내부 변수: <ul style="list-style-type: none">• Camel 케이스를 사용합니다.• 밑줄을 사용하지 마십시오.• 약어는 사용하지 마십시오.	'\$newVolume' "\$qtreename" `\$time'
기능: <ul style="list-style-type: none">• 파스칼 케이스를 사용합니다.• 밑줄을 사용하지 마십시오.• 약어는 사용하지 마십시오.	'GetVolumeSize'
변수 이름은 대/소문자를 구분하지 않습니다. 그러나 가독성을 높이기 위해 같은 이름에 다른 대/소문자를 사용하지 않아야 합니다.	'\$variable'은(는) '\$Variable'과(와) 동일합니다.
변수 이름은 영문으로 되어 있어야 하며 스크립트의 기능과 관련이 있어야 합니다.	\$A가 아니라 \$NAME을 사용하십시오.
각 변수의 데이터 형식을 명시적으로 선언합니다.	[string] 이름입니다 [int] 크기

지침	예
특수 문자(! @#&%%,.) 및 공백.	없음
PowerShell 예약 키워드를 사용하지 마십시오.	없음
필수 매개변수를 먼저 배치한 다음 선택적 매개변수를 배치하여 입력 매개변수를 그룹화합니다.	<pre>param ([parameter (Mandatory=\$true)] [string]\$Type, [parameter (Mandatory=\$true)] [string]\$Ip, [parameter (Mandatory=\$false)] [string]\$VolumeName)</pre>
의미 있는 도움말 메시지와 함께 HelpMessage 주석을 사용하여 모든 입력 변수에 주석을 달 수 있습니다.	<pre>[parameter (Mandatory=\$false, HelpMessage="LUN to map")] [string]\$LUNName</pre>
파일러를 변수 이름으로 사용하지 말고 대신 배열(Array)을 사용하십시오.	없음
인수가 열거형 값을 가져오는 경우 ValidateSet 주석을 사용합니다. 그러면 해당 매개 변수의 Enum 데이터 형식이 자동으로 변환됩니다.	<pre>[parameter (Mandatory=\$false, HelpMessage="Volume state")] [ValidateSet ("online", "offline", "restricted")] [string]\$State</pre>
매개 변수가 용량 유형임을 나타내기 위해 ""_Capacity"로 끝나는 매개 변수에 별칭을 추가합니다.	<p>""볼륨 작성" 명령은 다음과 같이 별칭을 사용합니다.</p> <pre>[parameter (Mandatory=\$false, HelpMessage="Volume increment size in MB")] [Alias ("AutosizeIncrementSize_Capacity")] [int]\$AutosizeIncrementSize</pre>

지침	예
매개 변수가 암호 형식임을 나타내기 위해 ""_Password"로 끝나는 매개 변수에 별칭을 추가합니다.	<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 변수

지침	예
스크립트 입력 매개 변수의 경우: <ul style="list-style-type: none"> • 파스칼 케이스를 사용합니다. • 밑줄을 사용하지 마십시오. • 약어는 사용하지 마십시오. 	'\$VolumeName' '\$AutoDeleteOptions' '\$Size'
스크립트 내부 변수에는 약어를 사용하지 마십시오.	'\$new_volume' '\$qtree_NAME' `\$time'
기능에 약어를 사용하지 마십시오.	'get_volume_size'입니다
변수 이름은 대/소문자를 구분합니다. 가독성을 향상시키려면 같은 이름에 다른 대/소문자를 사용하지 않아야 합니다.	'\$variable'은(는) '\$Variable'과(와) 같지 않습니다.
변수 이름은 영문으로 되어 있어야 하며 스크립트의 기능과 관련이 있어야 합니다.	\$A가 아니라 \$NAME을 사용하십시오.
필수 매개변수를 먼저 배치한 다음 선택적 매개변수를 사용하여 입력 매개변수를 그룹화합니다.	없음

지침	예
GetOptions 함수에서 입력 매개 변수에 대한 각 변수의 데이터 형식을 명시적으로 선언합니다.	<pre>GetOptions ("Name=s"=>\\$Name, "Size=i"=>\\$Size)</pre>
파일러를 변수 이름으로 사용하지 말고 대신 배열 (Array)을 사용하십시오.	없음
Perl에는 열거 값에 대한 "ValidateSet" 주석이 포함되지 않습니다. 인수가 열거 값을 갖는 경우에는 명시적 "if" 문을 사용합니다.	<pre>if (defined\$SpaceGuarantee&&!(\$SpaceG uaranteeeq'none'</pre>
	\$SpaceGuaranteeeq'volume'
	<pre>\$SpaceGuaranteeeq'file')) { die'Illegal SpaceGuarantee argument: \'.'\$SpaceGuarantee.'\'; } ----</pre>
모든 Perl WFA 명령은 변수, 참조 및 서브루틴에 안전하지 않은 구조를 사용하지 않도록 "strict" pragma를 사용해야 합니다.	<pre>use strict; # the above is equivalent to use strictvars; use strictsubs; use strictrefs;</pre>
<p>모든 Perl WFA 명령은 다음 Perl 모듈을 사용해야 합니다.</p> <ul style="list-style-type: none"> • getopt 입력 매개 변수를 지정하는 데 사용됩니다. • WFAUtil 이 기능은 명령 로깅, 명령 진행률 보고, 어레이 컨트롤러에 연결 등에 제공되는 유틸리티 기능에 사용됩니다. 	<pre>use Getopt::Long; use NaServer; use WFAUtil;</pre>

들여쓰기를 위한 지침

WFA(OnCommand Workflow Automation)용 PowerShell 또는 Perl 스크립트를 작성할 때 들여쓰기에 대한 지침을 숙지해야 합니다.

지침	예
탭은 네 개의 빈 공간과 같습니다.	
탭과 중괄호를 사용하여 블록의 시작과 끝을 표시합니다.	<p>PowerShell 스크립트</p> <pre> if (\$pair.length-ne 2) { throw "Got wrong input data" } </pre> <p>Perl 스크립트</p> <pre> if (defined \$MaxDirectorySize) { # convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024; } </pre>
작업 집합 또는 코드 청크 사이에 빈 줄을 추가합니다.	<pre> \$options=\$option.trim(); \$pair=\$option.split(" "); Get-WFALogger -Info -messages \$("split options: "+ \$Pair) </pre>

코멘트에 대한 지침

WFA(OnCommand Workflow Automation)용 스크립트에서 PowerShell 및 Perl 설명에 대한 지침을 반드시 숙지하십시오.

PowerShell 설명

지침	예
한 줄 코멘트에 # 문자를 사용합니다.	<pre># Single line comment \$options=\$option.trim();</pre>
줄 끝 코멘트의 경우 # 문자를 사용합니다.	<pre>\$options=\$option.trim(); # End of line comment</pre>
블록 코멘트에 <# 및 #> 문자를 사용합니다.	<pre><# This is a block comment #> \$options=\$option.trim();</pre>

Perl 설명

지침	예
한 줄 코멘트에 # 문자를 사용합니다.	<pre># convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024;</pre>
줄 설명 끝에 # 문자를 사용합니다.	<pre>my \$MaxDirectorySizeBytes = \$MaxDirect orySize * 1024 * 1024; # convert to Bytes</pre>

지침	예
시작 부분과 끝 부분에 빈 #이 있는 모든 줄의 #문자를 사용하여 여러 줄 메모에 대한 설명 테두리를 만듭니다.	<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>
WFA 명령에 주석 및 비활성 코드를 포함하지 마십시오. 그러나 테스트를 위해 POD(Plain Old Documentation) 메커니즘을 사용하여 코드를 주석 처리할 수 있습니다.	<pre>=begin comment # Set deduplication if (defined \$Deduplication && \$Deduplication eq "enabled") { \$wfaUtil- >sendLog("Enabling Deduplication"); } =end comment =cut</pre>

로깅 지침

WFA(OnCommand Workflow Automation)용 PowerShell 또는 Perl 스크립트를 작성할 때 로깅 지침에 대해 알고 있어야 합니다.

PowerShell 로깅

지침	예
로깅에 Get-WFALogger cmdlet을 사용합니다.	<pre>Get-WFALogger -Info -message "Creating volume"</pre>

지침	예
Data ONTAP, VMware, PowerCLI 등의 내부 패키지와 상호 작용해야 하는 모든 작업을 기록합니다. 모든 로그 메시지는 워크플로의 실행 상태 기록에 있는 실행 로그에서 사용할 수 있습니다.	없음
내부 패키지에 전달되는 모든 관련 인수를 기록합니다.	없음
사용 컨텍스트에 따라 Get-WFALogger cmdlet을 사용할 때 적절한 로그 수준을 사용합니다. -Info, -Error, -Warn 및 -Debug 사용 가능한 다양한 로그 수준입니다. 로그 수준을 지정하지 않으면 기본 로그 수준은 디버그입니다.	없음

Perl 로깅

지침	예
로깅에 WFAUtil sendLog를 사용합니다.	<pre>my wfa_util = WFAUtil->new(); eval { \$wfa_util->sendLog('INFO', "Connecting to the cluster: \$DestinationCluster"); }</pre>
Data ONTAP, VMware, WFA와 같은 외부 명령어와 상호 작용해야 하는 모든 작업을 기록합니다. WFAUtil sendLog 루틴을 사용하여 생성한 모든 로그 메시지가 WFA 데이터베이스에 저장됩니다. 이러한 로그 메시지는 실행된 워크플로 및 명령에 사용할 수 있습니다.	없음
호출된 루틴에 전달된 모든 관련 인수를 기록합니다.	없음
적절한 로그 수준을 사용합니다. - 정보, - 오류, - 경고 및 - 디버그가 사용 가능한 다양한 로그 수준입니다.	없음

지침	예
Info 수준에서 기록할 때는 정확하고 간결해야 합니다. 로그 메시지에 클래스 이름 및 함수 이름과 같은 구현 세부 정보를 지정하지 마십시오. 정확한 단계 또는 정확한 오류를 일반 영어로 설명합니다.	<p>다음 코드 조각에서는 양호한 메시지와 잘못된 메시지의 예를 보여 줍니다.</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>

오류 처리 지침

WFA(OnCommand Workflow Automation)용 PowerShell 또는 Perl 스크립트를 작성할 때 오류 처리에 대한 지침을 숙지해야 합니다.

PowerShell 오류 처리

지침	예
<p>PowerShell 런타임에 의해 cmdlet에 추가된 일반적인 매개 변수에는 ErrorAction 및 WarningAction 같은 오류 처리 매개 변수가 포함됩니다.</p> <ul style="list-style-type: none"> ErrorAction 매개 변수는 cmdlet이 명령에서 발생하는 비종료 오류에 대응하는 방법을 결정합니다. WarningAction 매개 변수는 cmdlet이 명령의 경고에 반응하는 방식을 결정합니다. Stop, SilentlyContinue, Inquire 및 Continue 는 ErrorAction 및 WarningAction 매개 변수에 유효한 값입니다. <p>자세한 내용은 PowerShell CLI에서 'Get-Help About_CommonParameters' 명령을 사용할 수 있습니다.</p>	<p>ErrorAction: 다음 예제에서는 비종료 오류를 종료 오류로 처리하는 방법을 보여 줍니다.</p> <pre>'New-Nclgroup-Name\$IgroupName- Protocol\$Protocol-Type\$OSType- ErrorActionstop'</pre> <p>경고 조치</p> <pre>'New-VM-Name\$VMName-VM\$SourceVM- datastore\$DataStoreName-VMHost\$VMHost- WarningActionSilentlyContinue'를 참조하십시오</pre>

지침	예
<p>들어오는 예외의 유형을 알 수 없는 경우 일반 <code>try/catch</code> 문을 사용합니다.</p>	<pre>try { "In Try/catch block" } catch { "Got exception" }</pre>
<p>들어오는 예외의 유형이 알려진 경우 특정 <code>try/catch</code> 문을 사용합니다.</p>	<pre>try { "In Try/catch block" } catch[System.Net.WebExceptional], [System.IO. IOException] { "Got exception" }</pre>
<p>리소스를 해제하려면 <code>finally</code> 문을 사용하십시오.</p>	<pre>try { "In Try/catch block" } catch { "Got exception" } finally { "Release resources" }</pre>

지침	예
<p>PowerShell 자동 변수를 사용하여 예외에 대한 정보에 액세스합니다.</p>	<pre> try { Get-WFALogger -Info -message \$("Creating Ipspace: " + \$Ipspace) New-NaNetIpspace-Name \$Ipspace } catch { Throw "Failed to create Ipspace. Message: " + \$_.Exception.Message; } </pre>

Perl 오류 처리

지침	예
<p>Perl에는 try/catch 블록에 대한 기본 언어 지원이 포함되어 있지 않습니다. eval 블록을 사용하여 오류를 검사하고 처리합니다. 평가 블록을 가능한 한 작게 유지합니다.</p>	<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>

WFA에 대한 일반 PowerShell 및 Perl 규칙

WFA에서 기존 스크립트와 일치하는 스크립트를 생성하기 위해 사용되는 특정 PowerShell 및 Perl 규칙을 이해해야 합니다.

- 스크립트를 어떻게 실행할지 명확히 하는 데 도움이 되는 변수를 사용합니다.
- 주석 없이 이해할 수 있는 읽을 수 있는 코드를 작성합니다.
- 스크립트와 명령을 최대한 간단하게 유지하십시오.
- PowerShell 스크립트의 경우:
 - 가능하면 cmdlet을 사용하십시오.
 - 사용할 수 있는 cmdlet이 없으면 .NET 코드를 호출합니다.
- Perl 스크립트의 경우:

- "die" 문은 항상 줄 바꿈 문자로 끝나십시오.

줄 바꿈 문자가 없으면 스크립트 줄 번호가 인쇄되며, 이 번호는 WFA에서 실행되는 Perl 명령을 디버깅하는 데 도움이 되지 않습니다.

- ""getopt"" 모듈에서 문자열 인수를 필수 명령으로 만듭니다.

Perl 모듈은 Windows와 함께 번들로 제공됩니다

일부 Perl 모듈은 Windows WFA(Active State Perl Distribution for OnCommand Workflow Automation)와 함께 번들로 제공됩니다. 이러한 Perl 모듈은 Windows와 함께 번들로 제공되는 경우에만 명령 작성에 Perl 코드에서 사용할 수 있습니다.

다음 표에는 Windows for WFA에 번들로 제공되는 Perl 데이터베이스 모듈이 나열되어 있습니다.

데이터베이스 모듈	설명
DBD::MySQL	MySQL 데이터베이스에 연결할 수 있는 Perl5 데이터베이스 인터페이스 드라이버.
시도 아주 작다	평가 블록으로 일반적인 실수를 최소화합니다.
XML::LibXML	DOM, SAX 및 XmlReader 인터페이스를 사용하여 XML 및 HTML 파서 를 제공하는 libxml2 인터페이스입니다.
DBD::Cassandra	CQL3 쿼리 언어를 사용하는 Cassandra용 Perl5 데이터베이스 인터페이스 드라이버.

사용자 지정 PowerShell 및 Perl 모듈을 추가할 때의 고려 사항

WFA(OnCommand Workflow Automation)에 사용자 지정 PowerShell 및 Perl 모듈을 추가하기 전에 고려해야 할 사항이 있습니다. 사용자 지정 PowerShell 및 Perl 모듈을 사용하면 사용자 지정 명령을 사용하여 워크플로우를 생성할 수 있습니다.

- WFA 명령을 실행하는 동안 모든 맞춤형 PowerShell 모듈을 WFA 설치 디렉토리 `_/posh/modules_`에 자동으로 가져옵니다.
- `_WFA/perl_directory`에 추가된 모든 사용자 지정 Perl 모듈은 `_@INC_library`에 포함되어 있습니다.
- 사용자 지정 PowerShell 및 Perl 모듈은 WFA 백업 작업의 일부로 백업되지 않습니다.
- 사용자 지정 PowerShell 및 Perl 모듈은 WFA 복원 작업의 일부로 복원되지 않습니다.

새로운 WFA 설치에 복사하려면 사용자 지정 PowerShell 및 Perl 모듈을 수동으로 백업해야 합니다.

모듈 디렉토리의 폴더 이름은 모듈 이름의 폴더 이름과 같아야 합니다.

WFA cmdlet 및 기능

OnCommand Workflow Automation(WFA)에서는 몇 가지 PowerShell cmdlet과 WFA 명령에서 사용할 수 있는 PowerShell 및 Perl 기능을 제공합니다.

다음 PowerShell 명령을 사용하여 WFA 서버에서 제공하는 모든 PowerShell cmdlet 및 기능을 볼 수 있습니다.

- 'get-Command-Module WFAWrapper'
- get-Command-Module WFA를 선택합니다

WFA 서버에서 제공하는 모든 Perl 기능은 'WFAUtil.pm' 모듈에서 확인할 수 있습니다. WFA PowerShell cmdlet 도움말 및 WFA Perl 메서드의 도움말, WFA 도움말 모듈 지원 링크의 도움말 섹션에서는 모든 PowerShell cmdlet 및 기능, Perl 기능에 액세스할 수 있습니다.

PowerShell 및 Perl WFA 모듈

워크플로우 스크립트를 작성하려면 OnCommand Workflow Automation(WFA)용 PowerShell 또는 Perl 모듈을 숙지해야 합니다.


PowerShell 모듈

지침	예
Data ONTAP PS 툴킷을 사용하면 툴킷이 사용 가능할 때마다 API를 호출할 수 있습니다.	Add VLAN 명령은 다음과 같이 툴킷을 사용한다. 'Add-NaNetVlan-Interface\$Interface-VLAN\$VlanID'입니다
Data ONTAP PS 툴킷에서 사용할 수 있는 cmdlet이 없으면 Invoke-SSH 명령을 사용하여 Data ONTAP에서 CLI를 호출합니다.	'Invoke-NaSsh-Name\$ArrayName-Command'ifconfig-a"-Credential\$Credentials'를 선택합니다

Perl 모듈

NaServer 모듈은 WFA 명령에 사용됩니다. NaServer 모듈을 사용하면 Data ONTAP 시스템의 활성 관리에 사용되는 Data ONTAP API를 호출할 수 있습니다.

지침	예
<p>NaServer 모듈을 사용하면 NetApp Manageability SDK를 사용할 수 있을 때마다 API를 호출할 수 있습니다.</p>	<p>다음 예에서는 NServer 모듈을 사용하여 SnapMirror 작업을 다시 시작하는 방법을 보여 줍니다.</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> <pre> } else { my \$result = \$server- >snapmirror_resume('destination-vserver' => \$DestinationVserver, 'destination-volume' => \$DestinationVolume); \$wfa_util->sendLog('INFO', "Result of resume: \$result"); } } </pre>

지침	예
<p>Data ONTAP API를 사용할 수 없는 경우 executeSystemCli 유틸리티 메시지를 사용하여 Data ONTAP CLI를 호출합니다.</p> <div>  <p>executeSystemCli는 지원되지 않으며 현재 7-Mode에서 작동하는 Data ONTAP에서만 사용할 수 있습니다.</p> </div>	없음

PowerShell 명령을 Perl로 변환할 때의 고려 사항

PowerShell 및 Perl의 기능은 다르므로 PowerShell 명령을 Perl로 변환할 때 고려해야 할 중요한 사항에 대해 알고 있어야 합니다.

명령 입력 유형

WFA(OnCommand Workflow Automation)를 사용하면 워크플로우 설계자가 명령을 정의할 때 명령에 대한 입력으로 어레이 및 해시를 사용할 수 있습니다. 이 입력 유형은 명령이 Perl을 사용하여 정의된 경우 사용할 수 없습니다. Perl 명령이 배열 및 해시 입력을 허용하도록 하려면 디자이너에서 입력을 문자열로 정의할 수 있습니다. 그런 다음 명령 정의를 통해 입력을 구문 분석할 수 있습니다. 이 입력은 필요에 따라 배열 또는 해시를 만들기 위해 전달됩니다. 입력에 대한 설명은 입력이 필요한 형식을 설명합니다.

```
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 명령문

다음 예에서는 스토리지 입력을 PowerShell 및 Perl로 전달하는 방법을 보여 줍니다. 이 예에서는 Cron 작업이 실행되도록 예약된 월을 지정하는 CronMonth 입력을 설명합니다. 유효한 값은 -1에서 11 사이의 정수입니다. 값이 -1이면 스케줄이 매달 실행됨을 나타냅니다. 다른 값은 특정 월을 나타내며 0은 1월, 11은 12월입니다.

```
[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 진술

```

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
}

```

명령 정의

PowerShell에서 파이프 연산자를 사용하는 한 줄 식은 동일한 기능을 구현하기 위해 Perl의 여러 문 블록으로 확장해야 할 수도 있습니다. 다음 표에는 대기 명령 중 하나의 예가 나와 있습니다.

PowerShell 명령문	Perl 진술
<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 구성 요소에 대한 지침입니다

Workflow Automation 구성 요소를 사용하기 위한 지침에 대해 알고 있어야 합니다.

WFA에서의 SQL에 대한 지침입니다

WFA(SQL in OnCommand Workflow Automation)를 사용하여 WFA에 대한 SQL 쿼리를 작성하는 방법에 대한 지침을 알고 있어야 합니다.

SQL은 WFA의 다음 위치에서 사용됩니다.

- 선택할 사용자 입력을 채우기 위한 SQL 쿼리입니다
- 특정 사전 항목 유형의 개체를 필터링하기 위한 필터를 만들기 위한 SQL 쿼리입니다
- 운동장 데이터베이스의 테이블에 있는 정적 데이터
- CMDB(사용자 정의 구성 관리 데이터베이스)와 같은 외부 데이터 소스에서 데이터를 추출해야 하는 SQL 유형의 사용자 정의 데이터 소스 유형입니다.
- SQL은 예약 및 검증 스크립트를 쿼리합니다

지침	예
SQL 예약 키워드는 대문자여야 합니다.	<div>SELECT vserver.name FROM cm_storage.vserver vserver</div>
테이블 및 열 이름은 소문자여야 합니다.	표: 애그리게이트 열: used_space_MB
밑줄(_) 문자로 단어를 구분하십시오. 공백은 허용되지 않습니다.	array_performance.(배열 성능
테이블 이름은 단수명으로 정의됩니다. 테이블은 하나 이상의 항목의 모음입니다.	"기능"이 아니라 "기능"

지침	예
<p>선택 쿼리에 의미 있는 이름이 있는 테이블 별칭을 사용합니다.</p>	<pre> SELECT vsrvr.name FROM cm_storage.cluster cluster, cm_storage.vsrvr vsrvr WHERE vsrvr.cluster_id = cluster.id AND cluster.name = '\${ClusterName}' AND vsrvr.type = 'cluster' ORDER BY vsrvr.name ASC </pre>

지침	예
<p>필터 쿼리 또는 사용자 쿼리에서 필터 입력 매개 변수 또는 사용자 입력 매개 변수를 참조해야 하는 경우 구문을 '\${inputVariableName}'으로 사용합니다. 또한 이 구문을 사용하여 예약 스크립트 및 검증 스크립트에서 명령 정의 매개 변수를 참조할 수도 있습니다.</p>	<pre> 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 </pre>
<p>복잡한 쿼리에 주석을 사용합니다. 쿼리에서 지원되는 주석 스타일 중 일부는 다음과 같습니다.</p> <ul style="list-style-type: none"> • "-"를 끝까지 누릅니다 <p>이 주석 스타일의 두 번째 하이픈 뒤에 공백이 있어야 합니다.</p> <ul style="list-style-type: none"> • ""#" 문자부터 줄의 끝까지 • "\ * "에서 다음 " * /" 시퀀스로 	<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 기능에 대한 지침입니다

함수를 만들어 일반적으로 사용되는 논리를 명명된 함수에 캡슐화한 다음 함수를 명령 매개 변수 값이나 WFA(OnCommand Workflow Automation)의 필터 매개 변수 값으로 다시 사용할 수 있습니다.

지침	예
기능 이름에 Camel 케이스를 사용합니다.	CalculateVolumeSize(계산 볼륨 크기)
변수 이름은 영문으로 되어 있어야 하며 함수의 기능과 관련이 있어야 합니다.	splitByDelimiter 를 참조하십시오
약어는 사용하지 마십시오.	calculateVolumeSize, _not_calcVolSize
함수는 MVFLEX Expression Language(MVEL)를 사용하여 정의됩니다.	없음
함수 정의는 공식 Java 프로그래밍 언어 지침에 따라 지정해야 합니다.	없음

WFA 사전 항목에 대한 지침입니다

WFA(OnCommand Workflow Automation)에서 사전 항목을 생성하기 위한 지침에 대해 알고 있어야 합니다.

지침	예
사전 항목 이름에는 영숫자와 밑줄만 사용할 수 있습니다.	Cluster_License 스위치_23
사전 항목 이름은 대문자로 시작해야 합니다. 이름의 모든 단어를 대문자로 시작하고 각 단어를 밑줄(_)로 구분합니다.	볼륨 Array_License를 선택합니다
사전 항목 속성 이름은 사전 항목의 이름을 포함할 수 없습니다.	없음
사전 항목의 속성 및 참조는 소문자여야 합니다.	애그리게이트, 크기_MB
밑줄로 단어를 구분합니다. 공백은 허용되지 않습니다.	resource_pool을 선택합니다

지침	예
사전 항목에는 다른 구성표의 참조가 포함될 수 없습니다. 사전 항목에 다른 스키마의 개체에 대한 상호 참조가 필요한 경우 참조되는 개체의 모든 기본 키가 사전 항목에 있는지 확인합니다.	Array_Performance 사전 항목에는 Array 사전 항목의 모든 고유 키가 직접 특성으로 필요합니다.
속성에 적절한 데이터 형식을 사용합니다.	없음
크기 또는 공간 관련 속성에 긴 데이터 형식을 사용합니다.	size_MB and available_size_MB in storage.Volume 사전 항목
특성에 고정 값 집합이 있는 경우 Enum 을 사용합니다.	storage.Volume 사전 항목의 RAID_TYPE
데이터 소스에서 해당 특성 또는 참조에 대한 값을 제공하는 경우 특성 또는 참조에 대해 ""캐시되도록""를 * true * 로 설정합니다. OnCommand Unified Manager 데이터 소스의 경우 데이터 소스에서 해당 값을 제공할 수 있으면 캐시 가능 특성을 추가합니다.	없음
이 특성 또는 참조에 대한 값을 제공하는 데이터 소스가 NULL을 반환할 수 있는 경우 ""Null일 수 있음""을 * TRUE * 로 설정합니다.	없음
각 특성 및 참조에 의미 있는 설명을 제공합니다. 이 설명은 워크플로를 디자인할 때 명령 세부 정보에 표시됩니다.	없음
사전 항목의 속성 이름으로 ""id""를 사용하지 마십시오. 내부 WFA 사용을 위해 예약되어 있습니다.	없음

명령 지침

WFA(OnCommand Workflow Automation)에서 명령을 생성하기 위한 지침에 대해 알고 있어야 합니다.

지침	예
명령에 쉽게 식별할 수 있는 이름을 사용합니다.	'Qtree 생성'
공백을 사용하여 단어를 구분하고 각 단어는 대문자로 시작해야 합니다.	볼륨 만들기

지침	예
선택적 매개 변수의 예상 결과를 포함하여 명령의 기능을 설명하는 설명을 제공합니다.	없음
기본적으로 표준 명령의 시간 제한은 600초입니다. 명령을 생성하는 동안 기본 시간 초과가 설정됩니다. 명령이 완료되는 데 시간이 더 오래 걸릴 수 있는 경우에만 기본값을 변경하십시오.	볼륨 만들기 명령
장기 실행 작업의 경우 두 개의 명령을 만듭니다. 하나는 장기 실행 작업을 호출하는 명령이고 다른 하나는 작업 진행률을 주기적으로 보고하는 명령입니다. 첫 번째 명령은 '표준 실행' 명령 유형이고 두 번째 명령은 '조건 대기' 명령 유형이어야 합니다.	VSM 생성 및 VSM 대기 명령
쉽게 식별할 수 있도록 condition 명령 이름 앞에 "wait"(대기)를 붙입니다.	CM 볼륨 이동 대기
""조건 대기"" 명령에 적절한 대기 간격을 사용하십시오. 지정된 값은 장기 실행 작업이 완료되었는지 확인하기 위해 폴링 명령을 실행하는 간격을 제어합니다.	"VSM 대기" 명령의 60s 샘플링 간격
'조건 대기' 명령의 경우 장시간 실행 작업이 완료될 때까지 예상되는 시간에 따라 적절한 시간 제한을 사용하십시오. 네트워크 상의 데이터 전송과 관련된 작업의 경우 예상 시간이 상당히 길어질 수 있습니다.	VSM 기본 전송은 완료하는 데 며칠이 걸릴 수 있습니다. 따라서 지정된 시간 초과는 6일입니다.

문자열 표현

명령의 문자열 표현에는 계획 및 실행 중에 워크플로 디자인에 명령의 세부 정보가 표시됩니다. 명령의 문자열 표현에는 명령 매개변수만 사용할 수 있습니다.

지침	예
값이 없는 특성은 사용하지 마십시오. 값이 없는 특성은 NA로 표시됩니다.	volname 10.68.66.212 [NA]aggr1/testVol7
다음 구분 기호를 사용하여 문자열 표현의 다른 항목을 구분합니다. [], /:	'ArrayName[ArrayIp]'입니다
문자열 표현의 모든 값에 의미 있는 레이블을 제공합니다.	'볼륨 이름 = VoumeName'

명령 정의 언어입니다

지원되는 다음 스크립트 언어를 사용하여 명령을 작성할 수 있습니다.

- PowerShell을 사용합니다

- Perl

명령 매개 변수 정의

명령 매개 변수는 이름, 설명, 유형, 매개 변수의 기본값 및 매개 변수가 필수인지 여부에 따라 설명됩니다. 매개 변수 형식은 String, Boolean, Integer, Long, Double 일 수 있습니다. Enum, DateTime, Capacity, Array, Hashtable, 암호 또는 XmlDocument 입니다. 대부분의 형식에 대한 값은 직관적이지만 Array 및 Hashtable 의 값은 다음 표에 설명된 특정 형식이어야 합니다.

지침	예
Array 입력 형식의 값이 쉼표로 구분된 값 목록인지 확인합니다.	<pre>[parameter (Mandatory=\$false, HelpMessage="Months in which the schedule executes.")] [array] \$CronMonths</pre> <p>입력은 0,3,6,9 입니다</p>
Hashtable 입력 형식의 값이 세미콜론으로 구분된 key=value 쌍의 목록인지 확인합니다.	<pre>[parameter (Mandatory=\$false, HelpMessage="Volume names and size (in MB) ")] [hashtable] \$VolumeNamesAndSize</pre> <p>입력은 다음과 같이 전달됩니다. Volume1 = 100; Volume2 = 250; volume3 = 50</p>

워크플로 지침

WFA(OnCommand Workflow Automation)에 대해 미리 정의된 워크플로우를 생성하거나 수정하기 위한 지침에 대해 알고 있어야 합니다.

일반 지침

지침	예
스토리지 운영자가 실행하는 작업을 반영하도록 워크플로의 이름을 지정합니다.	CIFS 공유를 생성합니다
워크플로 이름의 경우 첫 번째 단어와 개체인 모든 단어의 첫 글자를 대문자로 표시합니다. 약어 및 머리글자의 대문자로 표시합니다.	<p>볼륨</p> <p>qtree입니다</p> <p>Clustered Data ONTAP 트리 CIFS 공유를 생성합니다</p>

지침	예
워크플로 설명의 경우 필수 조건, 워크플로의 결과 또는 실행 조건적 측면을 포함하여 워크플로의 중요한 모든 단계를 포함합니다.	<p>샘플 워크플로에 대한 설명을 참조하십시오</p> <p>필수 구성 요소를 포함하는 'Create VMware NFS Datastore on Clustered Data ONTAP Storage'를 선택합니다.</p>
워크플로가 생산 준비가 되어 있고 포털 페이지에 표시될 수 있는 경우에만 ""생산 준비""를 *참*으로 설정합니다.	없음
<p>기본적으로 ""예약된 요소 고려"를 *TRUE*로 설정합니다.</p> <p>실행 워크플로우를 미리 볼 때 WFA 플래너는 캐시 데이터베이스의 기존 개체와 함께 예약된 모든 개체를 고려합니다. 이 옵션이 *TRUE*로 설정된 경우 특정 워크플로를 계획할 때 다른 예약된 워크플로 또는 병렬로 실행되는 워크플로의 효과를 고려합니다.</p>	<ul style="list-style-type: none"> • 시나리오 1 <p>워크플로 1은 볼륨을 생성하며 1주일 후에 실행되도록 예약되어 있습니다. Workflow 2는 검색된 볼륨에 qtree 또는 LUN을 생성하며, 워크플로우 2가 하루 이내에 실행되는 경우 워크플로우 2에 대해 ""예약된 요소 고려"를 해제해야만 일주일 내에 생성되는 볼륨을 고려할 수 있습니다.</p> <ul style="list-style-type: none"> • 시나리오 2 <p>워크플로 1은 '볼륨 만들기' 명령을 사용합니다. Aggregate에서 100GB을 사용하는 예약된 워크플로 2가 있는 경우 계획 중에 워크플로 1은 워크플로 2에 대한 요구 사항을 고려해야 합니다.</p>
기본적으로 ""요소 존재 유효성 검사 사용""은 *TRUE*로 설정됩니다.	<ul style="list-style-type: none"> • 시나리오 1 <p>볼륨이 존재하는 경우에만 '볼륨 제거' 명령을 사용하여 먼저 이름별로 볼륨을 제거하는 워크플로우를 생성한 후 '볼륨 생성' 또는 '볼륨 클론'과 같은 다른 명령을 사용하여 볼륨을 다시 생성하는 경우 워크플로우에서 이 플래그를 사용하지 않아야 합니다. 볼륨 제거 효과는 볼륨 만들기 명령에 사용할 수 없게 되어 워크플로가 실패합니다.</p> <ul style="list-style-type: none"> • 시나리오 2 <p>볼륨 만들기 명령은 특정 이름이 "vol198"인 워크플로에서 사용됩니다.</p> <p>이 옵션을 *true*로 설정하면 WFA 플래너가 계획 중에 해당 이름의 볼륨이 지정된 어레이에 있는지 확인합니다. 볼륨이 있는 경우 계획 중에 워크플로가 실패합니다.</p>

지침	예
워크플로에서 같은 명령을 두 번 이상 선택한 경우 명령 인스턴스에 적절한 표시 이름을 제공합니다.	SnapVault로 LUN 생성, 매핑 및 보호 샘플 워크플로에서는 볼륨 생성 명령을 두 번 사용합니다. 그러나 이 표시 이름은 기본 볼륨 및 미러링된 대상 볼륨에 대해 "기본 볼륨 생성" 및 "보조 볼륨 생성"으로 적절하게 사용됩니다.

사용자 입력

지침	예
<p>이름:</p> <ul style="list-style-type: none"> • 이름을 "\$" 문자로 시작합니다. • 각 단어의 시작 부분에 대문자를 사용합니다. • 모든 용어와 약어에 대문자를 사용합니다. • 밑줄을 사용하지 마십시오. 	<p>'\$Array'</p> <p>'\$VolumeName'</p>
<p>표시 이름:</p> <ul style="list-style-type: none"> • 각 단어의 시작 부분에 대문자를 사용합니다. • 단어를 공백으로 구분합니다. • 입력에 특정 단위가 있는 경우 표시 이름에 대괄호로 단위를 직접 지정합니다. 	<p>볼륨 이름</p> <p>볼륨 크기(MB)</p>
<p>설명:</p> <ul style="list-style-type: none"> • 각 사용자 입력에 대해 의미 있는 설명을 제공합니다. • 필요한 경우 예를 제공합니다. <p>특히 사용자 입력이 특정 형식이어야 할 경우 이 작업을 수행해야 합니다.</p> <p>사용자 입력 설명은 워크플로 실행 중 사용자 입력에 대한 도구 설명으로 표시됩니다.</p>	<p>"iGroup"에 추가할 이니시에이터입니다. 예를 들어 이니시에이터의 IQN 또는 WWPN입니다.</p>
<p>Type: 특정 값 집합으로 입력을 제한하려면 Enum 을 형식으로 선택합니다.</p>	<p>프로토콜: iSCSI, FCP, 혼합</p>

지침	예
<p>유형: 사용자가 WFA 캐시에 있는 값 중에서 선택할 수 있는 경우 유형으로 Query를 선택합니다.</p>	<p>\$Array: 다음과 같은 쿼리가 있는 쿼리 유형:</p> <pre>SELECT ip, name FROM storage.array</pre>
<p>Type: 사용자 입력이 쿼리에서 가져온 값으로 제한되거나 지원되는 Enum 형식으로만 제한되어야 하는 경우 사용자 입력을 locked로 표시합니다.</p>	<p>\$Array: 잠긴 쿼리 유형: 캐시에 있는 배열만 선택할 수 있습니다.\$Protocol: iSCSI, FCP, 혼합으로 유효한 값을 가진 잠긴 Enum 형식입니다. 유효한 값 이외의 값은 지원되지 않습니다.</p>
<p>유형: 쿼리 유형 추가 열을 쿼리에 반환 값으로 추가하면 저장소 운영자가 사용자 입력을 올바르게 선택할 수 있습니다.</p>	<p>\$Aggregate: 이름, 총 크기 및 사용 가능한 크기를 제공하여 운영자가 애그리게이트를 선택하기 전에 속성을 알 수 있도록 합니다.</p>
<p>유형: 사용자 입력에 대한 쿼리 TypeSQL 쿼리는 그 앞의 다른 사용자 입력을 참조할 수 있습니다. 어레이의 vFiler 유닛, 애그리게이트의 볼륨, SVM(스토리지 가상 머신)의 LUN 등 다른 사용자 입력을 기반으로 쿼리 결과를 제한하는 데 사용할 수 있습니다.</p>	<p>샘플 워크플로우 'Create a clustered Data ONTAP Volume'에서 VserverName에 대한 쿼리는 다음과 같습니다.</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>이 쿼리는 \${clusterName}을(를) 참조합니다. 여기서 \$clusterName은 \$VserverName 사용자 입력 앞의 사용자 입력 이름입니다.</p>

지침	예
<p>유형:</p> <p>부울 형식의 사용자 입력에는 값이 <code>""true, false""</code>인 부울 형식을 사용합니다. 이렇게 하면 사용자 입력을 직접 사용하여 워크플로 디자인에 내부 식을 작성하는 데 도움이 됩니다. 예를 들어, <code>\$UserInputName=="Yes"</code> 대신 <code>\$UserInputName</code>을 사용할 수 있습니다.</p>	<p><code>'\$CreateCIFSShare: 유효한 값을 ""true"" 또는 ""false""로 갖는 부울 형식</code></p>
<p>유형:</p> <p>문자열 및 숫자 형식의 경우 특정 형식으로 값의 유효성을 검사하려는 경우 값 열에 정규식을 사용합니다.</p> <p>IP 주소 및 네트워크 마스크 입력에 정규식을 사용합니다.</p>	<p>위치별 사용자 입력은 <code>""[A-Z][A-Z]\-0[1-9]""</code>로 나타낼 수 있습니다. 이 사용자 입력에는 <code>""US-01""NB-02""</code>가 허용되지만 <code>nb-00</code>은 사용할 수 없습니다.</p>
<p>유형:</p> <p>숫자 형식의 경우 값 열에 범위 기반 유효성 검사를 지정할 수 있습니다.</p>	<p>생성할 LUN 수의 경우 값 열의 항목은 1~20입니다.</p>
<p>그룹:</p> <p>관련 사용자 입력을 적절한 버킷으로 그룹화하고 그룹 이름을 지정합니다.</p>	<p>모든 스토리지 관련 사용자 입력에 대한 "스토리지 세부 정보"입니다. 모든 VMware 관련 사용자 입력에 대한 "저장 세부 정보"입니다.</p>
<p>필수:</p> <p>사용자 입력 값이 워크플로를 실행하는 데 필요한 경우 사용자 입력을 필수로 표시합니다. 이렇게 하면 사용자 입력 화면이 사용자의 입력을 받아 입력하게 됩니다.</p>	<p>"Create NFS Volume" 워크플로우에서 <code>""\$VolumeName"</code>.</p>
<p>기본값:</p> <p>사용자 입력에 대부분의 워크플로 실행에 사용할 수 있는 기본값이 있는 경우 값을 제공합니다. 이렇게 하면 기본값이 목적을 수행하는 경우 사용자가 실행 중에 더 적은 입력을 제공할 수 있습니다.</p>	<p>없음</p>

상수, 변수 및 반환 매개 변수

지침	예
<p>상수: 여러 명령에 대한 매개 변수를 정의하기 위해 공통 값을 사용할 때 상수를 정의합니다.</p>	<p><code>_aggregate_over</code>오버커밋_threshold_의 경우 SnapVault를 사용하여 LUN 생성, 매핑 및 보호 샘플 워크플로우</p>

지침	예
<p>상수:이름</p> <ul style="list-style-type: none"> • 각 단어의 시작 부분에 대문자를 사용합니다. • 모든 용어와 약어에 대문자를 사용합니다. • 밑줄을 사용하지 마십시오. • 상수 이름의 모든 문자는 대문자를 사용합니다. 	<p>_aggregate_used_space_threshold_입니다</p> <p>_ActualVolumeSizeInMB_</p>
<p>변수: 명령 매개 변수 상자 중 하나에 정의된 개체에 이름을 제공합니다. 변수는 자동으로 생성된 이름이며 변경할 수 있습니다.</p>	없음
<p>변수: 이름은 변수 이름에 소문자를 사용합니다.</p>	<p>볼륨1</p> <p>cifs_share 를 참조하십시오</p>
<p>반환 매개 변수: 워크플로 계획 및 실행이 계획 중에 계산되거나 선택된 일부 값을 반환해야 할 때 반환 매개 변수를 사용합니다. 이 값은 웹 서비스에서 워크플로가 실행될 때 미리 보기 모드에서 사용할 수 있습니다.</p>	<p>집계: 리소스 선택 논리를 사용하여 집계를 선택한 경우 실제 선택된 집계를 반환 매개 변수로 정의할 수 있습니다.</p>

원격 시스템 유형에 대한 검증 스크립트 생성 지침

WFA(OnCommand Workflow Automation)에서 정의하는 원격 시스템 유형을 테스트하는 데 사용되는 검증 스크립트를 생성하기 위한 지침에 대해 알고 있어야 합니다.

- 작성하는 Perl 스크립트는 Validation 스크립트 창에 제공된 샘플 스크립트와 비슷해야 합니다.
- 유효성 검사 스크립트의 출력은 샘플 스크립트의 출력과 비슷해야 합니다.

샘플 검증 스크립트

```
# 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;
}
```

데이터 소스 형식 만들기 지침

WFA(OnCommand Workflow Automation)에 대한 사용자 지정 데이터 소스를 정의하는 데 사용되는 데이터 소스 유형을 생성하기 위한 지침에 대해 알고 있어야 합니다.

다음 방법 중 하나를 사용하여 데이터 소스 형식을 정의할 수 있습니다.

- SQL: WFA SQL 지침을 사용하여 외부 데이터베이스를 기반으로 데이터 소스에서 선택한 쿼리를 정의할 수 있습니다.
- 스크립트: 사전 항목의 특정 스키마에 대한 데이터를 제공하는 PowerShell 스크립트를 작성할 수 있습니다.

데이터 소스 형식 만들기에 대한 지침은 다음과 같습니다.

- 스크립트를 생성하려면 PowerShell 언어를 사용해야 합니다.
- PowerShell 스크립트는 현재 작업 디렉토리의 각 사전 항목에 대한 출력을 제공해야 합니다.
- 데이터 파일의 이름은 dictionary_entry.csv여야 합니다. 여기서 사전 항목의 이름은 소문자여야 합니다.

Performance Advisor로부터 정보를 수집하는 사전 정의된 데이터 소스 유형은 스크립트 기반 데이터 소스 유형을 사용합니다. 출력 파일의 이름은 array_performance.csv, aggregate_performance.csv입니다.

- CSV 파일에는 사전 항목 속성의 순서와 정확히 일치하는 콘텐츠가 포함되어야 합니다.

사전 항목에는 array_ip, date, day, hour, cpu_busy, Total_ops_per_sec, disk_throughput_per_sec

PowerShell 스크립트는 같은 순서로 .csv 파일에 데이터를 추가합니다.

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

- Encoding 을 사용하여 스크립트의 데이터 출력이 WFA 캐시에 정확하게 로드되도록 해야 합니다.
- '.csv' 파일에 Null 값을 입력할 때 \N을 사용해야 합니다.

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.