



워크플로우

OnCommand Workflow Automation

NetApp

October 09, 2025

This PDF was generated from <https://docs.netapp.com/ko-kr/workflow-automation/workflows/concept-overview-of-oncommand-workflow-automation.html> on October 09, 2025. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# 목차

워크플로우 .....	1
OnCommand Workflow Automation 개요 .....	1
WFA 기능을 제공합니다 .....	1
WFA 라이센스 정보입니다 .....	1
Workflow Automation 설계자 이해 .....	1
OnCommand Workflow Automation에서 빌딩 블록 작업 .....	2
운동장 데이터베이스입니다 .....	9
워크플로 관리 .....	10
미리 정의된 워크플로 사용자 지정 .....	10
워크플로우 생성 .....	12
WFA 워크플로우 팩을 생성합니다 .....	37
WFA 워크플로우 팩을 SCM 리포지토리에 통합합니다 .....	42
워크플로우를 위한 구성 요소 생성 .....	45
데이터 원본 형식을 만듭니다 .....	45
명령을 생성합니다 .....	46
파인더를 생성합니다 .....	50
필터를 만듭니다 .....	50
사전 항목을 만듭니다 .....	51
함수를 만듭니다 .....	52
템플릿을 만듭니다 .....	53
캐시 쿼리를 만듭니다 .....	53
반복 일정을 생성합니다 .....	54
필터 규칙을 정의합니다 .....	55
승인 지점을 추가합니다 .....	56
WFA에 대한 코딩 지침 .....	57
변수에 대한 지침 .....	57
들여쓰기를 위한 지침 .....	60
코멘트에 대한 지침 .....	61
로깅 지침 .....	63
오류 처리 지침 .....	64
WFA에 대한 일반 PowerShell 및 Perl 규칙 .....	67
사용자 지정 PowerShell 및 Perl 모듈을 추가할 때의 고려 사항 .....	68
WFA cmdlet 및 기능 .....	69
PowerShell 및 Perl WFA 모듈 .....	69
PowerShell 명령을 Perl으로 변환할 때의 고려 사항 .....	72
WFA 구성 요소에 대한 지침입니다 .....	74
예약된 단어 .....	87
REST API 사용 방법 .....	87
학습 자료에 대한 참고 자료 .....	88

Windows PowerShell	88
Data ONTAP PowerShell 툴킷	89
Perl	89
NetApp 관리 SDK	90
SQL(구조적 쿼리 언어)	90
MVFLEX Expression Language(MVEL)	90
정규식입니다	90
OnCommand Workflow Automation 관련 문서	90
기타 참조	90
도구 참조	91

# 워크플로우

## OnCommand Workflow Automation 개요

WFA(OnCommand Workflow Automation)는 프로비저닝, 마이그레이션, 사용 중단, 데이터 보호 구성 등과 같은 스토리지 관리 작업을 자동화하는 소프트웨어 클로닝할 수 있습니다. WFA를 사용하여 프로세스에서 지정한 작업을 완료하기 위해 워크플로우를 구축할 수 있습니다.

워크플로는 다음과 같은 유형의 작업을 포함하여 순차적 단계로 구성된 반복 및 절차 작업입니다.

- 데이터베이스 또는 파일 시스템용 스토리지 프로비저닝, 마이그레이션 또는 폐기
- 스토리지 스위치 및 데이터 저장소를 포함한 새로운 가상화 환경 설정
- 애플리케이션용 스토리지를 광범위한 오케스트레이션 프로세스의 일부로 설정

스토리지 설계자는 워크플로우를 정의하여 모범 사례를 따르고 다음과 같은 조직의 요구사항을 충족할 수 있습니다.

- 필요한 명명 규칙을 사용합니다
- 스토리지 객체에 대한 고유한 옵션 설정
- 자원 선택
- 내부 구성 관리 데이터베이스(CMDB)와 티켓팅 애플리케이션 통합

## WFA 기능을 제공합니다

- 워크플로우 설계 포털을 통해 워크플로우를 구축할 수 있습니다

워크플로 디자인 포털에는 명령, 템플릿, 찾기, 필터, 워크플로우 생성에 사용되는 기능을 설명합니다. 디자이너는 자동화된 리소스 선택, 행 반복(반복) 및 승인 지점과 같은 워크플로에 고급 기능을 포함할 수 있습니다.

또한 워크플로 디자인 포털에는 외부 시스템의 데이터를 캐시하기 위한 사전 항목, 캐시 쿼리 및 데이터 원본 형식과 같은 구성 요소도 포함되어 있습니다.

- 워크플로우 실행, 워크플로우 실행 상태 확인 및 액세스 로그를 위한 실행 포털입니다
- WFA 설정, 데이터 소스에 연결, 사용자 자격 증명 구성 등의 작업에 대한 관리/설정 옵션입니다
- 웹 서비스 인터페이스를 통해 외부 포털 및 데이터 센터 오케스트레이션 소프트웨어에서 워크플로우를 호출할 수 있습니다
- 스토리지 자동화 스토어에서 WFA 팩을 다운로드합니다. ONTAP 9.7.0 팩은 WFA 5.1과 함께 번들로 제공됩니다.

## WFA 라이센스 정보입니다

OnCommand Workflow Automation 서버를 사용하기 위해 라이센스가 필요하지 않습니다.

## Workflow Automation 설계자 이해

WFA(Workflow Automation) 디자이너에서 파인더, 필터, 명령 등의 구성 요소를 사용하여

워크플로우를 생성합니다. 워크플로우를 생성하기 전에 구성 요소와 워크플로우 생성 프로세스를 이해하는 것이 중요합니다.

## OnCommand Workflow Automation에서 빌딩 블록 작업

WFA(Workflow Automation) 워크플로우는 여러 개의 구성 요소로 구성되며 WFA에는 사전 정의된 구성 요소 라이브러리가 포함되어 있습니다. WFA에서 제공하는 구성 요소를 사용하여 조직의 요구사항에 맞는 워크플로우를 생성할 수 있습니다.

WFA는 스토리지 자동화 프로세스에 대한 구조를 제공합니다. WFA의 유연성은 워크플로우 구성 요소를 사용하여 워크플로우를 구성하는 방식을 기반으로 합니다.

WFA 구성 요소는 다음과 같습니다.

- 사전 항목
- 명령
- 필터
- 파인더
- 기능
- 템플릿

WFA에서 구성 요소를 사용하여 워크플로우를 생성하는 방법을 이해해야 합니다.

데이터 원본은 무엇입니까

데이터 원본은 데이터를 추출하기 위해 다른 시스템, 파일 및 데이터베이스에 대한 연결을 설정하는 방법입니다. 예를 들어, 데이터 소스는 Active IQ Unified Manager 9.4 데이터 소스 유형의 Active IQ Unified Manager 데이터베이스에 대한 연결이 될 수 있습니다.

필수 데이터 소스 유형을 정의한 후 데이터 수집을 위해 WFA(OnCommand Workflow Automation)에 사용자 지정 데이터 소스를 추가할 수 있습니다. 캐싱 체계, 필요한 포트 및 획득 방법을 데이터 소스 유형과 연결하면 됩니다.

WFA는 다양한 데이터 소스를 통해 정보를 캐싱합니다. WFA는 데이터 소스에서 리소스 정보를 수집하고 캐싱 체계에 맞게 형식을 지정합니다. 캐시 테이블(캐싱 구조 내의 테이블)은 사전 항목 개체와 일치하도록 서식이 지정됩니다. 워크플로에서 파인더를 사용하면 사전 개체가 반환되고 사전 개체의 데이터가 캐시 테이블에서 채워집니다. 데이터 소스에서 데이터를 획득하는 프로세스를 \_데이터 소스 획득\_ 이라고 합니다. 스크립트 기반 방법이나 드라이버 기반 방법을 사용하여 데이터 소스를 획득할 수 있습니다. 소스는 서로 다를 수 있으며 데이터 소스 획득은 서로 다른 시간 간격으로 이를 샘플링할 수 있습니다. 그런 다음 WFA가 해당 정보를 데이터베이스에 병합하고 예약 데이터를 중첩하여 데이터베이스의 업데이트된 리소스 정보를 유지합니다.

WFA 데이터베이스에는 여러 가지 다른 캐싱 스키마가 포함되어 있습니다. 캐싱 체계는 테이블 집합이며 각 테이블에는 특정 사전 항목 형식의 정보가 포함되어 있지만 특정 데이터 소스 형식의 여러 원본에서 가져온 결합된 정보가 포함될 수 있습니다. WFA는 데이터베이스 정보를 사용하여 리소스 상태를 이해하고, 계산을 수행하고, 리소스에 대한 명령을 실행합니다.

사전 항목이 무엇입니까

사전 항목은 WFA(OnCommand Workflow Automation) 구성 요소 중 하나입니다. 사전 항목을

사용하여 스토리지 및 스토리지 관련 환경에서 개체 유형과 해당 관계를 나타낼 수 있습니다. 그런 다음 워크플로의 필터를 사용하여 사전 항목의 기본 키 값을 반환할 수 있습니다.

사전 항목은 WFA에서 지원하는 개체 유형의 정의입니다. 각 사전 항목은 지원되는 스토리지 및 스토리지 관련 환경에서 개체 유형과 해당 관계를 나타냅니다. 사전 개체는 형식 검사를 받을 수 있는 특성 목록으로 구성됩니다. 완전한 값이 있는 사전 개체는 형식의 개체 인스턴스를 설명합니다. 또한 참조 속성은 개체와 환경의 관계를 설명합니다. 예를 들어 볼륨 사전 개체에는 이름, size\_MB 및 volume\_보증 같은 여러 특성이 있습니다. 또한 볼륨 사전 개체에는 array\_id 및 aggregate\_id 형식으로 볼륨을 포함하는 배열과 Aggregate에 대한 참조가 포함됩니다

개체의 캐시 테이블은 캐시용으로 표시된 사전 항목의 일부 또는 모든 특성이 포함된 데이터베이스입니다. 캐시 테이블을 포함하는 사전 항목의 경우 캐싱 대상으로 적어도 하나의 사전 항목의 특성을 표시해야 합니다. 사전 항목에는 개체의 고유 식별자인 고유 키가 포함됩니다. 예를 들어 7-Mode 볼륨은 고유한 이름 및 해당 항목이 포함된 어레이의 IP 주소로 식별됩니다. qtree는 qtree 이름, 볼륨 이름 및 어레이 IP 주소로 식별됩니다. 사전 항목을 만들 때 사전 항목의 기본 키에 속하는 사전 특성을 식별해야 합니다.

#### 명령이 작동하는 방식

OnCommand Workflow Automation 명령은 워크플로우의 실행 블록입니다. 워크플로의 각 단계에 대해 명령을 사용할 수 있습니다.

WFA 명령은 PowerShell 및 Perl 스크립트를 사용하여 작성됩니다. 패키지가 설치된 경우 PowerShell 명령은 Data ONTAP PowerShell 툴킷과 VMware PowerCLI를 사용합니다. Perl 명령은 WFA 서버에 설치된 Perl 배포 및 Perl 모듈을 사용합니다. 명령에 PowerShell 및 Perl과 같은 여러 스크립팅 언어를 포함할 경우, 해당 스크립트가 설치된 운영 체제 및 WFA 구성 메뉴에 지정된 기본 언어 순서에 따라 WFA에서 적절한 스크립트를 선택합니다.

WFA 명령에 대한 스크립트에는 여러 매개 변수가 포함됩니다. 이러한 매개 변수는 사전 항목 속성에 매핑될 수 있습니다.

각 WFA 명령에는 여러 Data ONTAP 명령이 포함될 수 있습니다.

일부 WFA 명령은 오래 실행되는 작업을 기다리고 주기적으로 폴링할 수 있기 때문에 대기 명령이라고 합니다. 예를 들어 \* 여러 볼륨 이동 대기 \* 명령. 폴링 명령이 실행되는 대기 간격은 작업이 완료되었는지 확인하도록 구성할 수 있습니다.

WFA 명령은 워크플로가 실행 단계에 있는 동안 WFA에서 시작됩니다. WFA는 왼쪽에서 오른쪽 및 위에서 아래로 순서대로 명령을 순차적으로 실행합니다. 워크플로를 계획하면 명령에 제공되는 매개 변수의 사용 가능성 및 유효성을 확인할 수 있습니다. WFA 서버는 명령을 실행하기 전에 명령에 필요한 모든 매개 변수를 제공합니다.

명령에 대한 매개 변수는 워크플로 계획 중에 완료됩니다. 그런 다음 워크플로는 실행 시간 동안 이러한 매개 변수를 명령에 전달합니다. 명령은 매개 변수를 워크플로로 다시 전달할 수 없습니다. 하지만 워크플로의 명령 간에 실행 시간 중에 얻은 정보를 교환하려면 지정된 WFA PowerShell cmdlet 또는 Perl 기능을 사용할 수 있습니다.

WFA PowerShell 명령은 PowerShell cmdlet에 대해 -ErrorAction 중지 플래그를 사용하지 않으므로 오류로 인해 cmdlet이 실패할 경우에도 워크플로우 작업이 계속 진행됩니다. 특정 명령에 -ErrorAction 중지 플래그를 포함하려는 경우 명령을 복제하고 PowerShell 스크립트를 수정하여 플래그를 추가할 수 있습니다.

다음은 명령 간 정보 교환을 위해 WFA에 포함된 PowerShell cmdlet 및 Perl 기능입니다.

PowerShell cmdlet	Perl 기능
Add-WfaWorkflowParameter 를 참조하십시오	addWfaWorkflowParameter 를 참조하십시오

PowerShell cmdlet	Perl 기능
Get-WfaWorkflowParameter 를 참조하십시오	getWfaWorkflowParameter를 참조하십시오

""add"" cmdlet 또는 명령에 추가된 파라미터는 이후에 실행되는 명령으로 검색하여 ""get"" cmdlet 또는 기능을 사용할 수 있습니다. 예를 들어 PowerShell WFA 명령에서 다음 코드를 사용하여 volumeId라는 매개 변수를 추가할 수 있습니다. "Add-WfaWorkflowParameter -Name "VolumeUUID" -Value "12345" -AddAsReturnParameter\$ true" 그런 다음 후속 명령에서 다음을 사용하여 volumeId 값('\$volumeId=get-WfaWorkflowParameter-Name volumeId')을 검색할 수 있습니다.

WFA 명령은 WFA 데이터베이스를 쿼리하고 필요한 결과를 가져올 수 있습니다. 이렇게 하면 필터와 파인더를 사용하지 않고 명령을 구성할 수 있습니다. 다음 함수를 사용하여 데이터베이스를 쿼리할 수 있습니다.

PowerShell cmdlet	Perl 함수
Invoke - MySqlQuery 예: "Invoke - MySqlQuery - Query" cm_storage.cluster ""에서 cluster.name 를 'Cluster Name'으로 선택합니다	invokeMySqlQuery 를 선택합니다

- 관련 정보 \*

[필터를 만듭니다](#)

[파인더를 생성합니다](#)

[명령을 생성합니다](#)

[필터란](#)

워크플로우에서 WFA 필터를 사용하여 필요한 리소스를 선택할 수 있습니다.

WFA 필터는 WFA 데이터베이스에서 작동하는 SQL 기반 쿼리입니다. 각 필터는 특정 사전 형식의 요소 목록을 반환합니다. 반환된 요소는 SQL 쿼리에 지정된 선택 조건을 기반으로 합니다. 필터를 만들거나 편집하려면 SQL 구문을 알고 있어야 합니다.

[어떤 장애물이 있는지 확인합니다](#)

파인더는 일반적인 결과를 식별하기 위해 함께 사용되는 하나 이상의 필터를 조합한 것입니다. 워크플로우의 파인더를 사용하여 워크플로우 실행에 필요한 리소스를 선택할 수 있습니다.

파인더는 정렬 순서를 적용하여 해당 결과를 구분할 수 있습니다. 파인더는 선택 기준 및 정렬을 기반으로 최적의 리소스를 반환합니다.

파인더는 결과를 하나 또는 전혀 반환하지 않으므로 특정 스토리지 요소의 존재 여부를 확인하는 데 사용할 수 있습니다. 그러나 finder 를 반복 행 정의의 일부로 사용하면 결과 집합을 사용하여 그룹의 멤버 목록을 구성할 수 있습니다. 파인더에 사용되는 필터는 최소한 사전 형식의 기본 키를 반환하지만 값을 참조할 수 있는 추가 필드를 반환할 수 있습니다. 정렬 순서는 필터 SQL 쿼리의 반환된 필드에 적용될 수 있습니다.

파인더의 결과를 테스트할 수 있습니다. 파인더를 테스트할 때 모든 WFA 필터의 일반적인 결과를 볼 수 있으며, 검색 결과에서 파인더의 유효 결과가 강조 표시됩니다. 워크플로우에서 파인더를 사용할 경우 스토리지 운영자에게 의미 있는

정보를 전달하기 위해 사용자 지정 오류 메시지를 생성할 수 있습니다.

기능은 무엇입니까

워크플로의 계획 단계에서 완료해야 하는 복잡한 작업에 대해 워크플로의 함수를 사용할 수 있습니다.

MVFLEX Expression Language(MVEL)를 사용하여 함수를 작성할 수 있습니다. 함수를 사용하면 일반적으로 사용되는 논리는 물론 명명된 함수에 더 복잡한 논리를 결합하여 명령 매개 변수나 필터 매개 변수의 값으로 다시 사용할 수 있습니다. 함수를 한 번 작성하여 워크플로 전체에서 사용할 수 있습니다. 함수를 사용하면 복잡한 명명 규칙을 정의하는 등 복잡한 반복 작업과 작업을 처리할 수 있습니다.

함수는 실행 중에 다른 함수를 사용할 수 있습니다.

구성표는 무엇입니까

스키마는 시스템의 데이터 모델을 나타냅니다. 데이터 모델은 사전 항목의 모음입니다. 스키마를 정의한 다음 데이터 소스 형식을 정의할 수 있습니다. 데이터 원본은 데이터 획득 방법과 스키마가 채워지는 방법을 정의합니다. 예를 들어 VC 스키마는 가상 머신, 호스트 및 데이터 저장소와 같은 가상 환경에 대한 데이터를 가져옵니다.

또한 특정 문제를 해결하도록 사용자 지정된 워크플로우를 통해 스키마에 데이터를 직접 채울 수 있습니다.

사전 항목은 사전 항목을 만들 때 기존 스키마와 연결됩니다. 사전 항목은 캐시 쿼리에도 연결되며 캐시 쿼리에는 SQL 쿼리가 포함됩니다.

스키마는 스크립트 기반 데이터 소스 유형 또는 SQL 데이터 소스 유형을 사용하여 데이터를 획득할 수 있습니다. 데이터 소스 형식을 만드는 동안 스크립트가 정의되고 SQL 쿼리가 캐시 쿼리에 정의됩니다.

WFA에는 다음 구성표가 포함되어 있습니다.

- \* 7-Mode (스토리지) \*

7-Mode로 운영되는 Data ONTAP에서 Active IQ Unified Manager를 통해 데이터를 획득하기 위한 구성표,

- \* clustered Data ONTAP(cm\_storage) \*

Clustered Data ONTAP에서 Active IQ Unified Manager를 통해 데이터를 취득하기 위한 계획

- \* 7-Mode 성능(성능) \*

Performance Advisor를 통해 7-Mode로 운영되는 Data ONTAP의 성능 데이터를 획득하기 위한 계획

- \* clustered Data ONTAP 성능(cm\_performance) \*

Performance Advisor를 통해 clustered Data ONTAP의 성능 데이터를 획득하기 위한 계획

- \* VMware vCenter(VC) \*

VMware vCenter에서 데이터를 획득하기 위한 체계입니다.

- \* 운동장(운동장) \*

## 데이터를 직접 채울 수 있는 구성표

### 어떤 원격 시스템 유형인지 확인하십시오

OnCommand Workflow Automation(WFA)는 원격 시스템 유형과 통신합니다. 원격 시스템 유형은 WFA가 통신할 수 있는 원격 시스템의 유형을 지정합니다. WFA에서 원격 시스템 유형을 구성할 수 있습니다. 예를 들어 Data ONTAP 시스템을 원격 시스템 유형으로 구성할 수 있습니다.

원격 시스템 유형에는 다음과 같은 특성이 있습니다.

- 이름
- 설명
- 버전
- 프로토콜
- 포트
- 시간 초과

각 원격 시스템 유형에 대해 Perl 스크립트를 사용하여 원격 시스템의 자격 증명을 검증할 수 있습니다. WFA에 구성된 원격 시스템에 대한 자격 증명을 저장할 수 있습니다. 새 사용자 지정 원격 시스템 유형을 추가하거나 편집할 수 있습니다. 기존 원격 시스템 유형을 복제할 수도 있습니다. 연결된 시스템이 없는 경우에만 원격 시스템 유형을 삭제할 수 있습니다.

### 서식 파일 사용 방법

WFA 템플릿을 레퍼런스 또는 사용 정책에 따라 워크플로우에 사용할 수 있습니다.

WFA 템플릿은 객체 정의의 청사진 역할을 합니다. 개체의 속성과 개체의 속성 값을 포함하여 템플릿을 정의할 수 있습니다. 그런 다음 템플릿을 사용하여 워크플로에서 개체 정의의 속성을 채울 수 있습니다.

서식 파일을 사용할 때는 서식 파일에서 가져온 값을 포함하는 필드를 편집할 수 없습니다. 따라서 템플릿을 사용하여 사용 정책을 설정하고 개체를 만들 수 있습니다. 템플릿을 적용한 후 템플릿과 워크플로의 연결을 제거하면 템플릿에 입력된 값은 그대로 유지되지만 필드를 편집할 수 있습니다.

### 범주를 사용하는 방법

워크플로를 분류하여 워크플로를 보다 효율적으로 구성하고 워크플로에 액세스 제어 기능을 적용할 수 있습니다.

WFA 포털에서 특정 그룹에 표시되도록 워크플로우를 분류할 수 있습니다. 워크플로 범주에 액세스 제어 기능을 적용할 수도 있습니다. 예를 들어, 특정 스토리지 운영자 또는 승인자만 특정 워크플로우 범주를 볼 수 있도록 허용할 수 있습니다. 스토리지 운영자 또는 승인자는 액세스 권한이 부여된 범주 내의 워크플로우만 실행할 수 있습니다.

Active Directory 그룹을 사용하여 범주에 대한 액세스 제어를 수행할 수도 있습니다.

## 엔터티 버전 관리 작동 방식

명령 및 워크플로우와 같은 WFA(OnCommand Workflow Automation) 엔터티의 버전은 분산됩니다. 버전 번호를 사용하여 WFA 엔터티의 변경 사항을 쉽게 관리할 수 있습니다.

각 WFA 엔터티에는 1.1.20과 같이 'jor.minor.revision' 형식의 버전 번호가 포함되어 있습니다. 버전 번호의 각 부분에 최대 3자리 숫자를 포함할 수 있습니다.

WFA 엔터티의 버전 번호를 수정하기 전에 다음 규칙을 알고 있어야 합니다.

- 버전 번호는 현재 버전에서 이전 버전으로 변경할 수 없습니다.
- 버전 각 부분은 0에서 999 사이의 숫자여야 합니다.
- 새로운 WFA 엔터티의 버전은 기본적으로 1.0.0으로 관리됩니다.
- 개체의 버전 번호는 복제 시 또는 \* 다른 이름으로 저장 \* 을 사용하여 개체의 복사본을 저장할 때 유지됩니다.
- WFA 설치 버전에는 여러 버전의 엔터티가 포함되어 있을 수 없습니다.

WFA Entity의 버전을 업데이트하면 해당 상위 엔터티의 버전이 자동으로 업데이트됩니다. 예를 들어, \* 볼륨 생성 \* 명령의 버전을 업데이트하면 \* NFS 볼륨 생성 \* 워크플로우가 \* 볼륨 생성 \* 명령의 직접적인 상위 엔터티이므로 \* NFS 볼륨 생성 \* 워크플로우가 업데이트됩니다. 버전에 대한 자동 업데이트는 다음과 같이 적용됩니다.

- 엔티티의 주 버전을 수정하면 직접 실행 모체 엔티티의 부 버전이 업데이트됩니다.
- 요소의 부 버전을 수정하면 직접 실행 모체 요소의 수정본 버전이 업데이트됩니다.
- 요소의 수정본 버전을 수정해도 그 직계 모체 요소의 버전 일부가 업데이트되지 않습니다.

다음 표에는 WFA 엔터티와 해당 상위 엔터티가 나와 있습니다.

요소	직접 모체 요소
캐시 쿼리입니다	<ul style="list-style-type: none"><li>데이터 소스 유형입니다</li></ul>
템플릿	<ul style="list-style-type: none"><li>워크플로우</li></ul>
기능	<ul style="list-style-type: none"><li>워크플로우</li><li>템플릿</li></ul> <p> 함수에 특수 또는 혼합 대/소문자 문자가 포함된 경우 해당 상위 요소의 직접 버전은 업데이트되지 않을 수 있습니다.</p>
사전	<ul style="list-style-type: none"><li>템플릿</li><li>필터</li><li>캐시 쿼리입니다</li><li>명령</li><li>스크립트 메서드를 사용하는 데이터 소스 유형입니다</li></ul>

요소	직접 모체 요소
명령	• 워크플로우
필터	• 파인더 • 워크플로우
파인더	• 워크플로우
데이터 소스 유형입니다	없음
워크플로우	없음

WFA에서 버전 번호의 일부 또는 전체 버전 번호를 사용하여 엔터티를 검색할 수 있습니다.

모체 요소를 삭제하면, 종속 요소는 보존되고 해당 버전은 삭제 대상으로 업데이트되지 않습니다.

엔터티를 가져올 때 버전 관리 작동 방식

Workflow Automation 2.2 이전 버전에서 요소를 가져오면 기본적으로 엔터티의 버전이 1.0.0으로 지정됩니다. 가져온 엔터티가 WFA 서버에 이미 있는 경우 가져온 엔터티로 기존 엔터티를 덮어씁니다.

가져오는 동안 WFA 엔터티가 변경될 수 있습니다.

- 요소 업그레이드

요소가 최신 버전으로 대치됩니다.

- 요소 롤백

요소가 이전 버전으로 대치됩니다.



요소의 롤백을 수행하면 해당 인접 상위 요소의 버전이 업데이트됩니다.

- 새 요소 불러오기



'.dar' 파일에서 요소를 선택적으로 불러올 수 없습니다.

요소의 차기 버전을 불러오면, 즉시 모체 요소의 버전이 업데이트됩니다.

불러온 모체 요소에 여러 개의 종속 요소가 있을 경우, 종속 요소에 대한 가장 높은 변경 수준(주, 부 또는 수정본)만 모체 요소에 적용됩니다. 다음 예에서는 이 규칙의 작동 방식을 설명합니다.

- 불러온 모체 요소의 경우, 사소한 변경 사항이 있는 종속 요소와 수정본이 변경된 다른 종속 요소가 있으면, 모체 요소에 사소한 변경 사항이 적용됩니다.

모체 버전의 수정본 부분이 증가합니다.

- 불러온 모체 요소의 경우, 주요 변경 사항이 있는 종속 요소와 사소한 변경 사항이 있는 다른 종속 요소가 있으면, 주요 변경 사항이 모체 요소에 적용됩니다.

상위 버전의 부 부분이 증가합니다.

불러온 하위 요소의 버전이 모체 버전에 영향을 주는 방법의 예

WFA에서 다음 워크플로우를 고려하십시오. "Create Volume and export using NFS-Custom' 1.0.0.

워크플로에 포함된 기존 명령은 다음과 같습니다.

- 수출정책 만들기-사용자 정의 1.0.0
- Create Volume - Custom" 1.0.0

가져올 .dar 파일에 포함된 명령어는 다음과 같다.

- 수출정책 만들기-사용자지정 1.1.0
- Create Volume - Custom" 2.0.0

이 ".dar" 파일을 가져오면 "Create Volume and export using NFS-Custom'" 워크플로우의 부 버전이 1.1.0으로 증가합니다.

## 운동장 데이터베이스입니다

운동장 데이터베이스는 MySQL 데이터베이스로, WFA(Workflow Automation) 서버 설치에 포함됩니다. 운동장 데이터베이스에 테이블을 추가하여 정보를 포함할 수 있습니다. 이러한 정보는 필터 및 SQL 쿼리에서 사용자 입력에 사용할 수 있습니다.

운동장 데이터베이스는 WFA 웹 포털을 통해 액세스할 수 없는 스키마입니다. SQLyog, MySQL용 Toad, MySQL Workbench 또는 CLI(명령줄 인터페이스)와 같은 MySQL 클라이언트를 사용하여 데이터베이스에 액세스할 수 있습니다.

운동장 데이터베이스에 액세스하려면 다음 자격 증명을 사용해야 합니다.

- 사용자 이름: WFA
- 암호: Wfa123

자격 증명을 통해 운동장 데이터베이스에 완벽하게 액세스하고 WFA MySQL 데이터베이스에 정의된 다른 스키마에 읽기 전용 액세스를 제공할 수 있습니다. 운동장 데이터베이스에 필요한 테이블을 만들 수 있습니다.

환경의 스토리지 개체에 사용 중인 태그나 메타데이터를 놀이터 데이터베이스의 테이블에 추가할 수 있습니다. 그런 다음 WFA 필터 및 사용자 입력 쿼리를 통해 태그나 메타데이터를 다른 WFA 캐시 표의 정보와 함께 사용할 수 있습니다.

예를 들어 다음과 같은 사용 사례에서 운동장 데이터베이스를 사용할 수 있습니다.

- BU(Business Unit) 이름으로 애그리게이트를 태깅하고 이러한 태그를 기반으로 볼륨을 버스에 할당합니다
- vFiler 유닛을 BU 이름으로 태그 지정
- 스토리지 객체에 지역 또는 위치 세부 정보 추가

- 데이터베이스 관리자의 데이터베이스 액세스 정의

예를 들어 BU의 이름을 애그리게이트 및 vFiler 유닛과 같은 스토리지 객체의 태그로 사용하는 경우, BU 이름이 포함된 테이블을 놀이터 데이터베이스에 생성할 수 있습니다. 그런 다음 BU 이름을 워크플로에 대한 필터 및 사용자 입력 쿼리에서 사용할 수 있습니다.

다음은 Playground 데이터베이스 테이블(Playground.volume\_bu)의 예입니다.

Array_IP입니다	볼륨_이름	사업부
10.225.126.23	데이터_11	마케팅
10.225.126.28	아치_11	시간

다음은 BU로 볼륨을 필터링하는 데 사용할 수 있는 SQL 쿼리의 예입니다.

```

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

```

- 관련 정보 \*

"SQLyog"

"MySQL 워크벤치"

"MySQL용 TOAD"

## 워크플로 관리

미리 정의된 워크플로를 사용자 지정하거나 워크플로 관리의 일부로 새 워크플로를 만들 수 있습니다. 또한 워크플로 관리를 시작하기 전에 관련 개념을 이해해야 합니다.

### 미리 정의된 워크플로 사용자 지정

요구사항에 적합한 사전 정의된 워크플로우가 없는 경우 사전 정의된 워크플로우 자동화(WFA) 워크플로우를 커스터マイ징할 수 있습니다.

## 필요한 것

사전 정의된 워크플로에 필요한 수정 사항을 식별해야 합니다.

## 이 작업에 대해

다음 사항에 대한 질문 및 지원 요청은 WFA 커뮤니티로 리디렉션해야 합니다.

- WFA 커뮤니티에서 다운로드한 모든 콘텐츠
- 생성한 맞춤형 WFA 콘텐츠입니다
- 수정한 WFA 콘텐츠입니다

## 단계

1. 워크플로우 설계 \* > \* 워크플로우 \* 를 클릭합니다.
2. 요구 사항과 가장 일치하는 미리 정의된 워크플로를 선택한 다음 을 클릭합니다  를 클릭합니다.
3. 워크플로 디자이너에서 설명 편집, 명령 추가 또는 삭제, 명령 세부 정보 수정, 사용자 입력 수정 등 적절한 탭에서 필요한 변경 작업을 수행합니다.
4. Preview \*(미리 보기 \*)를 클릭하고 필요한 사용자 입력을 입력하여 워크플로 실행을 미리 본 다음 \* Preview \*(미리 보기 \*)를 클릭하여 워크플로의 계획 세부 정보를 봅니다.
5. 확인 \* 을 클릭하여 미리보기 창을 닫습니다.
6. 저장 \* 을 클릭합니다.

## 작업을 마친 후

테스트 환경에서 수정한 워크플로를 테스트한 다음 워크플로를 생산 준비 상태로 표시할 수 있습니다.

볼륨 생성 및 **CIFS** 공유 워크플로우를 사용자 지정합니다

요구사항에 따라 워크플로우를 커스터마이징할 수 있습니다. 예를 들어 사전 정의된 \_Create a Volume과 CIFS Share\_workflow를 수정하여 중복 제거 및 압축을 포함할 수 있습니다.

## 이 작업에 대해

이 작업의 사용자 지정 및 그림은 예이며, 요구사항에 따라 WFA 워크플로우를 수정할 수 있습니다.

## 단계

1. 워크플로우 설계 \* > \* 워크플로우 \* 를 클릭합니다.
2. 볼륨 생성 및 CIFS 공유 \* 워크플로우를 선택하고 를 클릭합니다  를 클릭합니다.
3. Details \* 탭을 클릭하고 \* Workflow name \* 필드에서 워크플로의 설명을 편집합니다.
4. Workflow \* 탭을 클릭하고 \* storage \* 스키마를 확장한 다음 \* 볼륨 생성 \* 과 \* CIFS 공유 생성 \* 명령 사이에 \* 데이터 중복 제거 및 압축 \* 설정 명령을 끌어서 놓습니다.
5. 첫 번째 행의 \* SETUP Deduplication and compression \* 명령 아래에 마우스 커서를 놓고 클릭합니다 .
6. '설정 중복 제거 및 압축 매개 변수' \* 대화 상자의 \* 볼륨 \* 탭에서 이전에 정의된 볼륨 \* 옵션을 사용하여 \* 를 선택한 다음 \* 볼륨 정의 \* 필드에서 \* share\_volume \* 옵션을 선택합니다. \* 옵션은 워크플로에서 \* 볼륨 생성 \* 명령으로 생성된 볼륨 개체 변수입니다.

7. Other Parameters \* 탭을 클릭하고 다음 단계를 수행하십시오.
  - a. StartNow \* 필드에서 \* true \* 를 선택합니다.
  - b. Compression \* 필드에서 \* Inline \* 을 선택합니다.
  - c. 매일 오전 1시에 중복 제거 및 압축을 예약하는 \* 스케줄 \* 필드에 ``sun@1" 식을 입력합니다
8. 확인 \* 을 클릭합니다.
9. 미리 보기 \* 를 클릭하여 워크플로 계획이 성공적으로 완료되었는지 확인한 다음 \* 확인 \* 을 클릭합니다.
10. 저장 \* 을 클릭합니다.

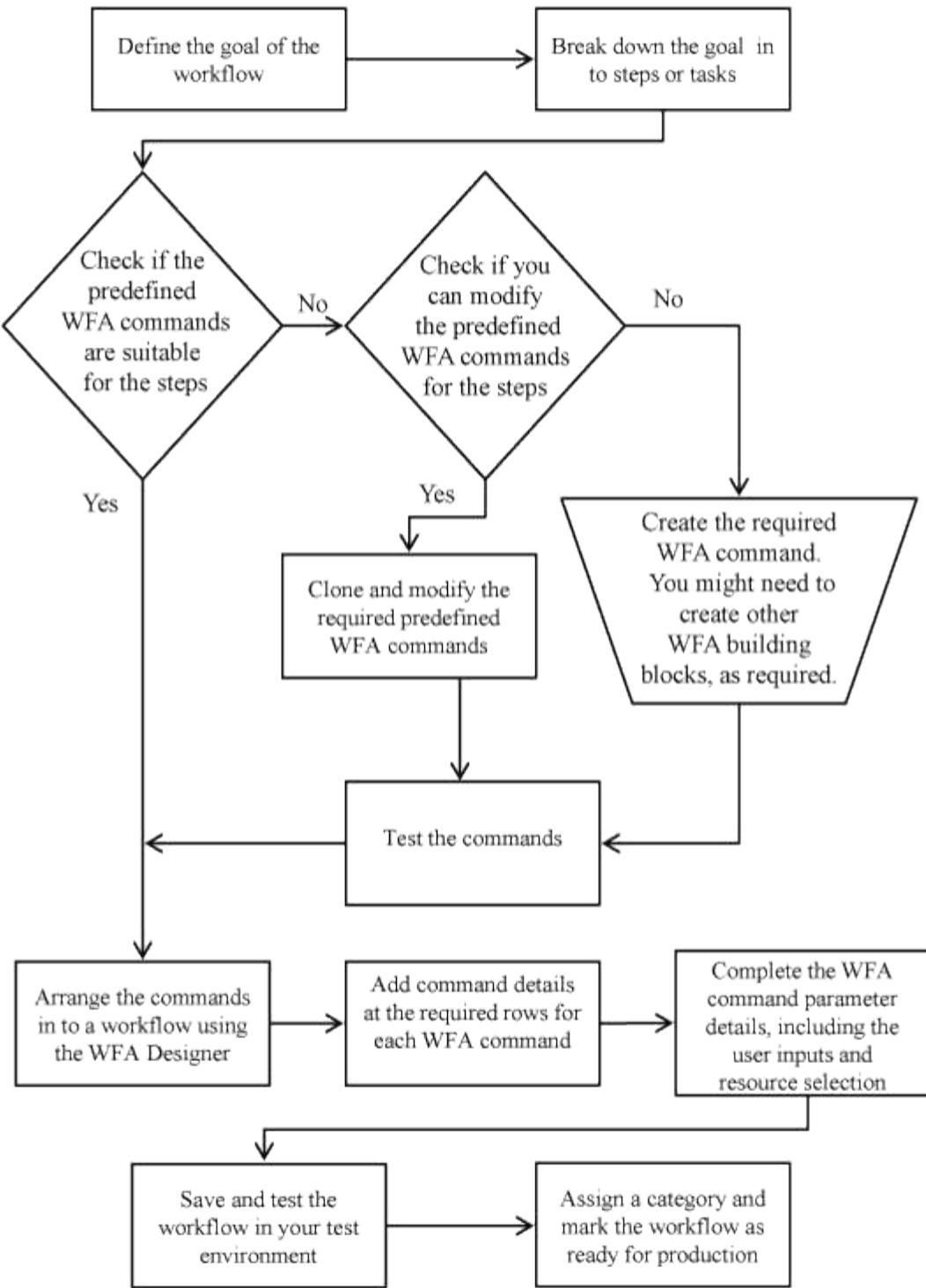
## 워크플로우 생성

사전 정의된 워크플로가 요구사항에 맞지 않으면 필요한 워크플로를 만들 수 있습니다. 워크플로우를 생성하기 전에 WFA 디자이너에서 사용할 수 있는 기능을 이해하고 워크플로우 체크리스트를 만들어야 합니다.

### 워크플로 생성과 관련된 작업

WFA(OnCommand Workflow Automation)에서 스토리지 자동화 워크플로우를 생성하려면 워크플로우에서 수행할 단계를 정의하고 명령, 찾기, 필터, 사전 항목 등과 같은 WFA 구성 요소를 사용하여 워크플로우를 생성해야 합니다.

다음 순서도는 워크플로 만들기 프로세스를 보여 줍니다.



## 워크플로우를 정의하는 방법

워크플로의 목표를 워크플로에서 실행해야 하는 단계로 나누어야 합니다. 그런 다음 워크플로를 완료하기 위한 단계를 정렬할 수 있습니다.

워크플로는 종단 간 프로세스를 완료하는 데 필요한 일련의 단계를 포함하는 알고리즘입니다. 프로세스의 범위는 워크플로의 목표에 따라 달라질 수 있습니다. 워크플로우 목표는 스토리지 운영만 처리하거나 네트워킹, 가상화, IT 시스템 및 기타 애플리케이션을 단일 프로세스로 처리하는 등 보다 복잡한 프로세스를 처리하도록 정의할 수 있습니다. OnCommand Workflow Automation(WFA) 워크플로우는 스토리지 설계자가 설계하고 스토리지 운영자가 실행합니다.

워크플로우 정의에는 워크플로우의 목표를 일련의 단계로 세분하는 작업이 포함됩니다. 예를 들어, NFS 볼륨을 생성하는 단계에는 다음 단계가 포함됩니다.

1. 볼륨 개체 생성
2. 새 엑스포트 정책을 생성하고 정책을 볼륨과 연결합니다

워크플로우의 각 단계에 WFA 명령 또는 워크플로우를 사용할 수 있습니다. WFA에는 일반 스토리지 사용 사례에 기반한 사전 정의된 명령 및 워크플로우가 포함되어 있습니다. 특정 단계에 사용할 수 있는 미리 정의된 명령 또는 워크플로를 찾을 수 없는 경우 다음 중 하나를 수행할 수 있습니다.

- 단계와 가장 일치하는 미리 정의된 명령 또는 워크플로우를 선택한 다음, 요구사항에 따라 사전 정의된 명령 또는 워크플로우를 클론 복제 및 수정합니다.
- 새 명령 또는 워크플로를 만듭니다.

그런 다음 새 워크플로에서 명령 또는 워크플로를 정렬하여 목표를 달성하는 워크플로를 만들 수 있습니다.

워크플로우 실행을 시작할 때 WFA는 실행을 계획하고 워크플로우 및 명령에 대한 입력을 사용하여 워크플로우를 실행할 수 있는지 확인합니다. 워크플로를 계획할 때 모든 리소스 선택 및 사용자 입력을 해결하여 실행 계획을 만듭니다. 계획 수립이 완료된 후 WFA는 실행 계획을 실행하며, 실행 계획은 해당 매개 변수를 포함하는 WFA 명령 세트로 구성됩니다.

#### 사용자 입력 정의 방법

WFA(OnCommand Workflow Automation) 사용자 입력은 워크플로우를 실행하는 동안 사용할 수 있는 데이터 입력 옵션입니다. 워크플로우의 유연성과 사용 편의성을 개선하려면 사용자 입력 매개 변수를 정의해야 합니다.

사용자 입력은 입력 필드로 표시되며 워크플로를 미리 보거나 실행할 때 관련 데이터로 채울 수 있습니다. 달러 기호 (\$)로 레이블 또는 변수를 미리 고정하여 워크플로에서 명령 세부 정보를 지정할 때 사용자 입력 필드를 만들 수 있습니다. 예를 들어, \$VolumeName은 Volume Name 사용자 입력 필드를 생성합니다. WFA는 Workflow <Workflow name> 창의 User Inputs 탭에 사용자가 만든 사용자 입력 레이블을 자동으로 채웁니다. 또한 유형, 표시 이름, 기본값 및 유효성 검사 값과 같은 사용자 입력 특성을 수정하여 사용자 입력 유형을 정의하고 입력 필드를 사용자 지정할 수도 있습니다.

#### 사용자 입력 유형 옵션

- \* 문자열 \*

유효한 값(예: \*)에 정규식을 사용할 수 있습니다.

0d 0f 등의 문자열은 double형 0으로 평가되는 0d와 비슷한 숫자로 평가된다.

- \* 번호 \*

선택할 수 있는 숫자 범위(예: 1 ~ 15)를 정의할 수 있습니다.

- \* Enum \*

enum 형식을 사용하여 사용자 입력 필드를 채울 때 선택할 수 있는 열거형 값을 만들 수 있습니다. 사용자가 만든 enum 값을 선택적으로 참가 사용자 입력에 대해 만든 값만 선택하도록 할 수 있습니다.

- \* 쿼리 \*

WFA 캐시에 있는 값에서 사용자 입력을 선택하려면 쿼리 유형을 선택할 수 있습니다. 예를 들어, 다음 쿼리를 사용하여 WFA 캐시의 IP 주소 및 이름 값으로 사용자 입력 필드를 자동으로 채울 수 있습니다. \* 'Select IP, name from storage.array.' \* 쿼리에 의해 검색된 값만 선택하도록 쿼리에 의해 검색된 값을 선택적으로 잠글 수 있습니다.

- \* 쿼리(다중 선택) \*

쿼리 유형과 유사한 쿼리(다중 선택) 형식을 사용하면 워크플로를 실행하는 동안 여러 값을 선택할 수 있습니다. 예를 들어, 사용자는 공유 및 내보내기와 함께 여러 볼륨 또는 볼륨을 선택할 수 있습니다. 사용자가 여러 행을 선택할 수 있도록 하거나 선택 영역을 단일 행으로 제한할 수 있습니다. 행을 선택하면 선택한 행의 모든 열에서 값이 선택됩니다.

사용자 입력의 쿼리(다중 선택) 유형을 사용할 때 다음 함수를 사용할 수 있습니다.

- getSize
- getValueAt
- getValueAt2D
- getValueFrom2DByRowKey

- \* 부울 \*

부울 형식을 사용하여 사용자 입력 대화 상자에 확인란을 표시할 수 있습니다. 가능한 값으로 ""true" 및 ""false""를 가진 사용자 입력에 부울 형식을 사용해야 합니다.

- \* 표 \*

사용자 입력의 테이블 형식을 사용하여 워크플로를 실행하는 동안 여러 값을 입력하는 데 사용할 수 있는 테이블의 열 머리글을 지정할 수 있습니다. 예를 들어, 노드 이름 및 포트 이름의 목록을 지정하는 데 사용할 수 있는 테이블입니다. 또한 다음 사용자 입력 형식 중 하나를 열 머리글에 지정하여 런타임 중에 입력한 값의 유효성을 검사할 수도 있습니다.

- 문자열
- 번호
- 번호
- 부울
- 쿼리 '검색'은 열 머리글의 기본 사용자 입력 유형입니다. 다른 사용자 입력 유형을 지정하려면 유형 열을 두 번 클릭해야 합니다.

디자이너에서 SnapMirror 정책 및 규칙 만들기 워크플로를 열어 사용자 입력 형식이 "PolicyRule" 사용자 입력에 어떻게 사용되는지 확인할 수 있습니다.

사용자 입력의 테이블 유형을 사용할 때 다음 기능을 사용할 수 있습니다.

- getSize
- getValueAt
- getValueAt2D
- GetValueFrom2DByRowKey Designer에서 \* 무한 확장 볼륨 \* 을 사용하여 스토리지 가상 머신 생성 및 구성 워크플로우를 열어 테이블 유형 사용 방법을 확인할 수 있습니다.

◦ \* 암호 \*

암호 입력을 위한 사용자 입력에 암호 유형을 사용할 수 있습니다. 사용자가 입력한 암호는 암호화되어 WFA 애플리케이션 및 로그 파일에 별표 문자 시퀀스로 표시됩니다. 다음 기능을 사용하여 암호를 해독한 다음 명령을 통해 사용할 수 있습니다.

- Perl 명령의 경우: `WFAUtil::getWfaInputPassword($password)`
- PowerShell 명령의 경우 `Get-WfaInputPassword - EncryptedPassword$ password` 를 사용합니다

여기서 `$password`는 WFA를 통해 명령으로 전달되는 암호화된 암호입니다.

◦ \* 사전 \*

선택한 사전 항목에 대한 테이블 데이터를 추가할 수 있습니다. 사전 항목 속성은 반환할 속성을 선택합니다. 워크플로를 실행하는 동안 단일 값 또는 여러 값을 선택할 수 있습니다. 예를 들어, 단일 볼륨 또는 여러 볼륨을 선택할 수 있습니다. 기본적으로 단일 값이 선택됩니다. 필터링할 규칙을 선택할 수도 있습니다. 규칙은 사전 항목 속성, 연산자 및 값으로 구성됩니다. 이 특성에는 해당 참조의 속성도 포함될 수 있습니다.

예를 들어, 이름이 ""aggr""이고 사용 가능한 크기가 5GB보다 큰 모든 애그리게이트를 나열하여 집계에 대한 규칙을 지정할 수 있습니다. 그룹의 첫 번째 규칙은 이름, 연산자 'starts-with', 값 aggr입니다. 같은 그룹의 두 번째 규칙은 연산자 > 와 값 5000 을 가진 속성 Available\_size\_MB 입니다

다음 표에는 사용자 입력 유형에 적용할 수 있는 옵션이 나와 있습니다.

옵션을 선택합니다	설명
검증 중	<p>사용자가 유효한 값만 입력하도록 사용자 입력 유형을 검증할 수 있습니다.</p> <ul style="list-style-type: none"><li>사용자 입력의 문자열 및 숫자 유형은 워크플로의 실행 시간 동안 입력한 값으로 유효성을 검사할 수 있습니다.</li><li>문자열 형식은 정규식을 사용하여 유효성을 검사할 수도 있습니다.</li><li>숫자 형식은 숫자 부동 소수점 필드이며 지정된 숫자 범위를 사용하여 유효성을 검사할 수 있습니다.</li></ul>
잠금 값	<p>사용자가 드롭다운 값을 덮어쓰지 않고 표시된 값만 선택할 수 있도록 쿼리 및 enum 형식의 값을 잠글 수 있습니다.</p>
필수로 표시	<p>사용자가 특정 사용자 입력을 입력해야 워크플로를 계속 실행할 수 있도록 사용자 입력을 필수로 표시할 수 있습니다.</p>
그룹화	<p>관련 사용자 입력을 그룹화하고 사용자 입력 그룹의 이름을 제공할 수 있습니다. 사용자 입력 대화 상자에서 그룹을 확장하고 축소할 수 있습니다. 기본적으로 확장해야 하는 그룹을 선택할 수 있습니다.</p>

옵션을 선택합니다	설명
조건 적용	조건부 사용자 입력 기능을 사용하면 다른 사용자 입력에 대해 입력된 값을 기반으로 사용자 입력 값을 설정할 수 있습니다. 예를 들어 NAS 프로토콜을 구성하는 워크플로우에서 프로토콜에 필요한 사용자 입력을 NFS로 지정하여 ""읽기/쓰기 호스트 목록"" 사용자 입력을 활성화할 수 있습니다.

명령 매개 변수를 매팅하는 방법입니다

WFA(Workflow Automation) 명령의 매개 변수는 특정 규칙에 따라 특정 특성 및 사전 항목 참조에 매팅됩니다. WFA 명령을 생성하거나 편집할 때 명령 매개 변수를 매팅하는 규칙을 알고 있어야 합니다.

명령 매개 변수 매팅은 워크플로에서 명령 세부 정보를 정의하는 방법을 정의합니다. 명령의 매팅된 명령 매개 변수는 워크플로우 명령에 대한 명령 세부 정보를 지정할 때 탭에 표시됩니다. 탭은 매개 변수 매팅 탭의 개체 이름 열에 지정된 그룹 이름을 기반으로 이름이 자정됩니다. 워크플로에서 명령 세부 정보를 지정할 때 매팅되지 않은 매개 변수는 기타 매개 변수 탭에 표시됩니다.

명령 매개 변수 매팅에 대한 규칙은 명령 범주 및 워크플로 편집기에서 명령이 표시되는 방법을 기반으로 적용됩니다.

다음은 명령 범주입니다.

- 개체를 만드는 명령입니다
- 개체를 업데이트하는 명령입니다
- 개체를 제거하는 명령입니다
- 선택적 부모 및 자식 개체를 처리하는 명령입니다
- 개체 간의 연결을 업데이트하는 명령입니다

규칙은 각 범주에 대해 아래에 나열되어 있습니다.

모든 명령 범주

명령 매개 변수를 매팅할 때는 워크플로에서 명령이 사용되는 방식을 기준으로 고유 경로를 사용해야 합니다.

다음 예제에서는 자연 경로를 정의하는 방법을 보여 줍니다.

- ArrayIP 매개 변수의 경우 명령에 따라 array.ip 속성이 아니라 Volume 사전 항목의 aggregate.array.ip 특성을 사용해야 합니다.

이는 워크플로가 볼륨을 생성한 다음 생성된 볼륨을 참조하여 추가 단계를 수행할 때 중요합니다. 다음은 이와 유사한 예입니다.

- Qtree 사전 항목의 volume.aggregate.array.IP입니다
- volume.aggregate.array.ip LUN 사전 항목의 IP입니다
- 명령에 사용되는 클러스터의 경우 다음 중 하나를 사용해야 합니다.

- 볼륨 사전 항목의 vserver.cluster.primary\_address입니다
- Qtree 사전 항목의 volume.vserver.cluster.primary\_address입니다

개체를 만드는 명령입니다

이 명령 범주는 다음 중 하나에 사용됩니다.

- 부모 개체 찾기 및 새 개체 정의
- 개체가 없는 경우 개체를 검색하고 개체를 만듭니다

이 명령 범주에 대해 다음 매개 변수 매팅 규칙을 사용해야 합니다.

- 생성된 개체의 관련 매개 변수를 개체의 사전 항목에 매팅합니다.
- 생성된 사전 항목의 참조를 통해 부모 개체를 매팅합니다.
- 새 매개 변수를 추가할 때 사전 항목에 관련 특성이 있는지 확인합니다.

다음은 이 규칙의 예외 시나리오입니다.

- 만들어진 일부 개체에는 해당 사전 항목이 없으며 부모 개체만 관련 부모 사전 항목에 매팅됩니다. 예를 들어, 배열 사전 항목에 배열만 매팅할 수 있는 \* VIF\* 만들기 명령-- 이 항목은 관련 부모 사전 항목에 매팅됩니다.
- 매개 변수 매팅은 필요하지 않습니다

예를 들어, \* Create 또는 resize aggregate \* 명령의 ExecutionTimeout 매개 변수는 매팅되지 않은 매개 변수입니다.

이 범주에는 다음과 같은 인증된 명령이 있습니다.

- 볼륨 생성
- LUN을 생성합니다

개체를 업데이트하는 명령입니다

이 명령 범주는 객체를 찾고 속성을 업데이트하는 데 사용됩니다.

이 명령 범주에 대해 다음 매개 변수 매팅 규칙을 사용해야 합니다.

- 사전 항목에 업데이트된 개체를 매팅합니다.
- 객체에 대해 업데이트된 매개 변수를 매팅하지 마십시오.

예를 들어 \* 볼륨 상태 설정 \* 명령에서 볼륨 매개 변수가 매팅되지만 새 상태는 매팅 해제됩니다.

개체를 제거하는 명령입니다

이 명령 범주는 개체를 찾아 삭제하는 데 사용됩니다.

명령에 의해 삭제된 개체를 해당 사전 항목에 매팅해야 합니다. 예를 들어 \* 볼륨 제거 \* 명령에서 삭제할 볼륨은 볼륨 사전 항목의 관련 속성 및 참조에 매팅됩니다.

선택적 부모 및 자식 개체를 처리하는 명령입니다

이 명령 범주에 대해 다음 매개 변수 매팅 규칙을 사용해야 합니다.

- 명령의 필수 매개 변수를 명령의 선택적 매개 변수의 참조로 매팅하지 마십시오.

이 규칙은 명령이 특정 부모 개체의 선택적 자식 개체를 처리하는 경우에 더 관련이 있습니다. 이 경우 자식 및 부모 개체를 명시적으로 매팅해야 합니다. 예를 들어, \* 중복 제거 작업 중지 \* 명령에서 명령은 특정 볼륨에 대해 스토리지 또는 지정된 스토리지의 모든 볼륨에 대해 실행 중인 중복 제거 작업을 중지합니다. 이 경우 이 명령에서 Volume 은 선택적 매개 변수이므로 array 매개 변수는 Volume.Array 가 아닌 배열 사전 항목에 직접 매팅되어야 합니다.

- 논리 수준에서 사전 항목 사이에 부모 및 자식 관계가 있지만 특정 명령의 실제 인스턴스 간에는 없는 경우 이러한 개체는 별도로 매팅되어야 합니다.

예를 들어, \* 볼륨 이동 \* 명령에서 볼륨은 현재 상위 애큰리게이트에 있는 새 대상 애큰리게이트로 이동됩니다. 따라서 볼륨 매개 변수는 볼륨 사전 항목에 매팅되고 대상 애큰리게이트 매개 변수는 volume.aggregate.name 가 아니라 애큰리게이트 사전 항목에 개별적으로 매팅됩니다.

개체 간의 연결을 업데이트하는 명령입니다

이 명령 범주의 경우 연관 및 객체를 관련 사전 항목에 모두 매팅해야 합니다. 예를 들어, vFiler에 볼륨 추가 명령에서 볼륨 및 vFiler 매개 변수는 볼륨 및 vFiler 사전 항목의 관련 특성에 매팅됩니다.

상수를 정의하는 방법

상수를 만들고 사용하여 값을 정의할 수 있으며, 이 값은 단일 워크플로에서 사용할 수 있습니다. 상수는 워크플로 수준에서 정의됩니다.

워크플로와 해당 값에 사용되는 상수는 계획 및 실행 중에 워크플로의 모니터링 창에 표시됩니다. 상수에는 고유한 이름을 사용해야 합니다.

다음 명명 규칙을 사용하여 상수를 정의할 수 있습니다.

- 각 단어의 첫 글자의 대문자이며, 단어 사이에 밑줄이나 공백이 없어야 합니다

모든 용어와 약어는 대문자(예: ActualVolumeSizeInMB)를 사용해야 합니다.

- 모두 대문자로 표시합니다

밑줄을 사용하여 단어를 구분할 수 있습니다. 예를 들어, 'aggregate\_used\_space\_threshold'를 사용할 수 있습니다.

워크플로 상수에 대한 값으로 다음을 포함할 수 있습니다.

- 번호
- 문자열
- MVEL 식

식은 워크플로의 계획 및 실행 단계에서 평가됩니다. 식에서 루프에 정의된 변수를 참조해서는 안 됩니다.

- 사용자 입력

- 변수

#### 행 반복 작동 방식

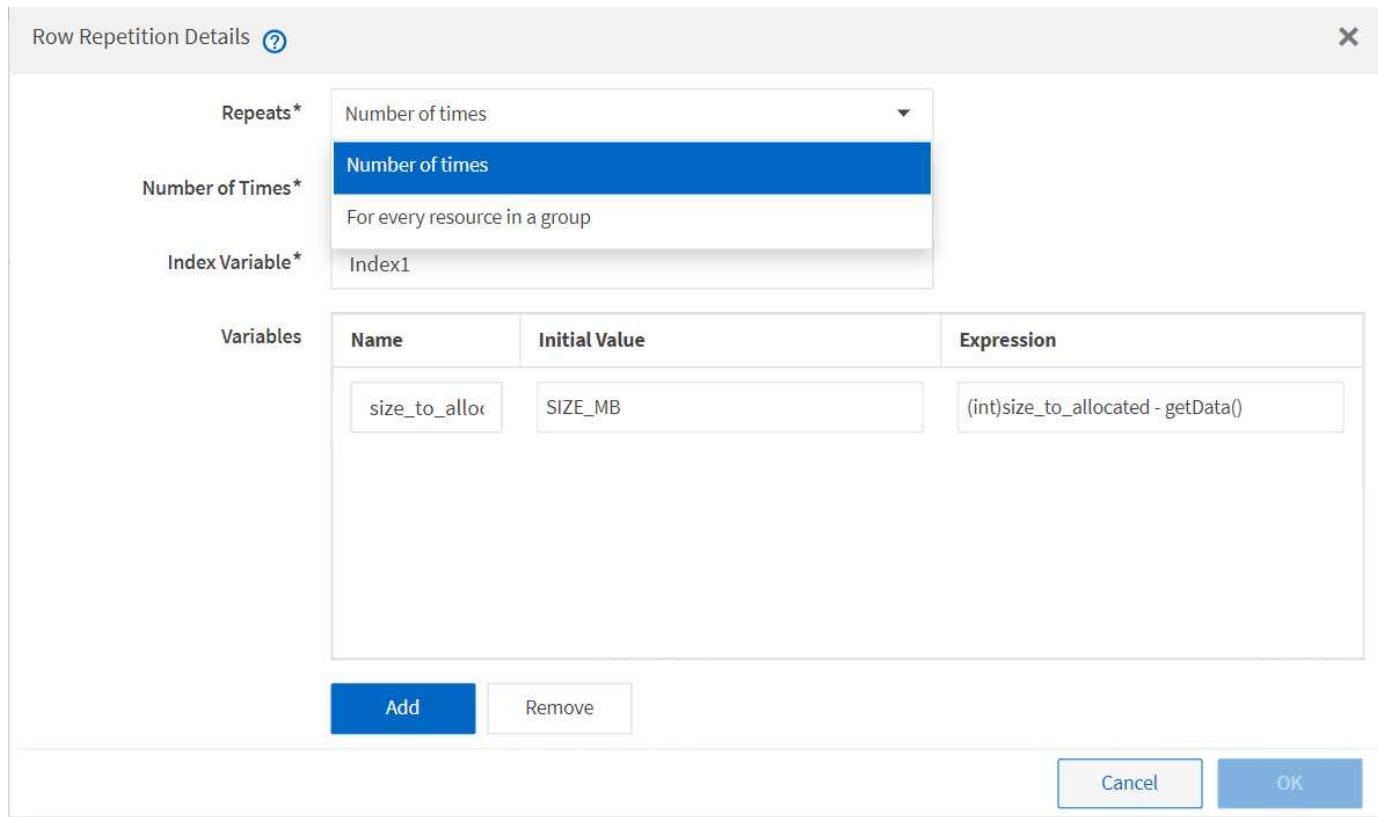
워크플로에는 명령 및 명령 세부 정보가 행으로 정렬되어 있습니다. 검색 조건 결과에 따라 고정된 반복 수나 동적 반복 수에 대해 반복할 행의 명령을 지정할 수 있습니다.

특정 횟수만큼 반복하거나 워크플로가 설계된 경우 행의 명령 세부 정보를 지정할 수 있습니다. 또한 워크플로를 실행하거나 실행을 예약할 때 행을 반복해야 하는 횟수를 지정할 수 있도록 워크플로를 설계할 수 있습니다. 개체에 대한 검색 조건을 지정할 수 있으며, 행의 명령은 검색 조건에 의해 반환된 개체만큼 반복되도록 설정할 수 있습니다. 특정 조건이 충족되면 행을 반복하도록 설정할 수도 있습니다.

#### 행 반복 변수

변수 목록에서 행 반복 중에 조작할 수 있는 변수를 지정할 수 있습니다. 변수의 경우 행 반복이 반복될 때마다 계산되는 이름, 변수가 초기화된 값 및 MVFLEX Expression Language(MVEL) 식을 지정할 수 있습니다.

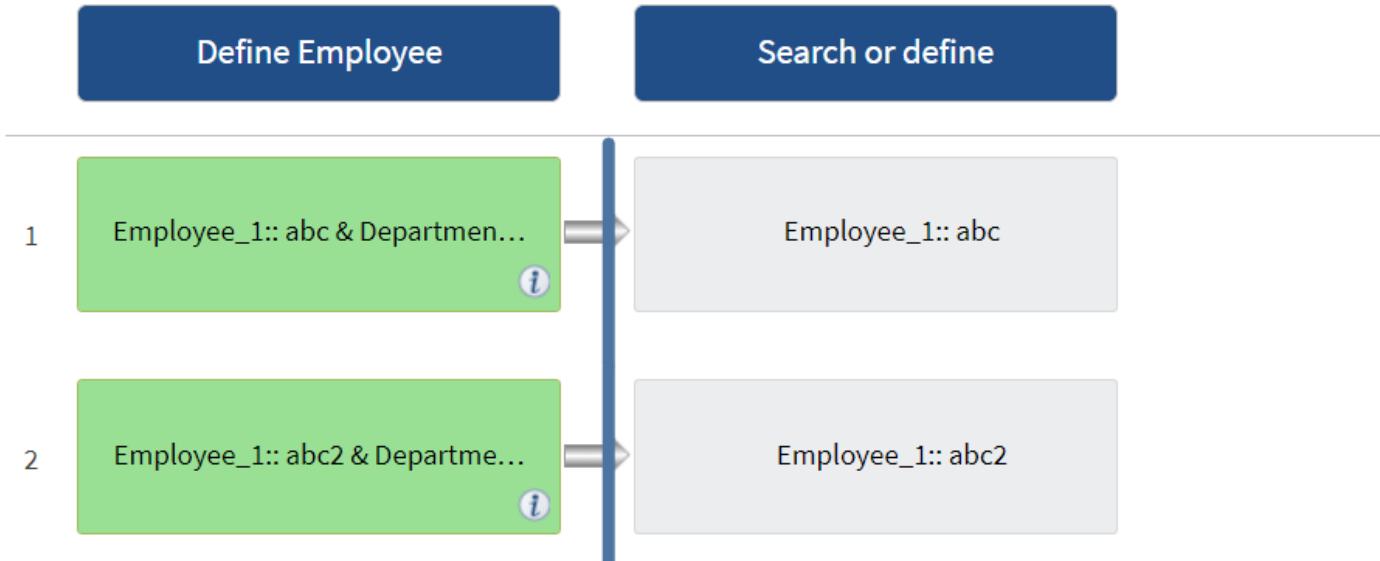
다음 그림에서는 반복 행 옵션과 행 반복 변수의 예를 보여 줍니다.



#### 승인 지점이 있는 행 반복

명령에 대해 반복 행을 지정하고 승인 지점을 포함시킨 경우, 승인 지점이 실행되기 전에 모든 명령이 반복됩니다. 승인 지점을 승인한 후 다음 승인 시점까지 연속 명령의 모든 반복 실행이 계속됩니다.

다음 그림에서는 승인 지점이 워크플로에 포함될 때 반복 행의 반복 실행 방법을 보여 줍니다.



미리 정의된 워크플로에서 행 예제를 반복합니다

디자이너에서 미리 정의된 다음 워크플로를 열어 반복 행이 사용되는 방식을 이해할 수 있습니다.

- Clustered Data ONTAP NFS 볼륨을 생성합니다
- Clustered Data ONTAP 스토리지에서 VMware NFS 데이터 저장소를 생성합니다
- 클러스터 피어링을 설정합니다
- Clustered Data ONTAP 볼륨을 제거합니다

#### 리소스 선택 작동 방식

WFA(OnCommand Workflow Automation)는 검색 알고리즘을 사용하여 워크플로우 실행을 위한 스토리지 리소스를 선택합니다. 워크플로를 효율적으로 설계하려면 리소스 선택이 어떻게 작동하는지 이해해야 합니다.

WFA는 검색 알고리즘을 사용하여 vFiler 유닛, 애그리게이트, 가상 머신 등의 사전 항목 리소스를 선택합니다. 그러면 선택한 리소스가 워크플로를 실행하는 데 사용됩니다. WFA 검색 알고리즘은 WFA 구성 요소에 포함되어 있으며 파인더와 필터를 포함합니다. 필요한 리소스를 찾아 선택하려면 검색 알고리즘이 Active IQ Unified Manager, VMware vCenter Server 및 데이터베이스와 같은 다른 리포지토리에서 캐시된 데이터를 검색합니다. 기본적으로 필터는 기본 키를 기준으로 리소스를 검색할 수 있는 모든 사전 항목에 대해 사용할 수 있습니다.

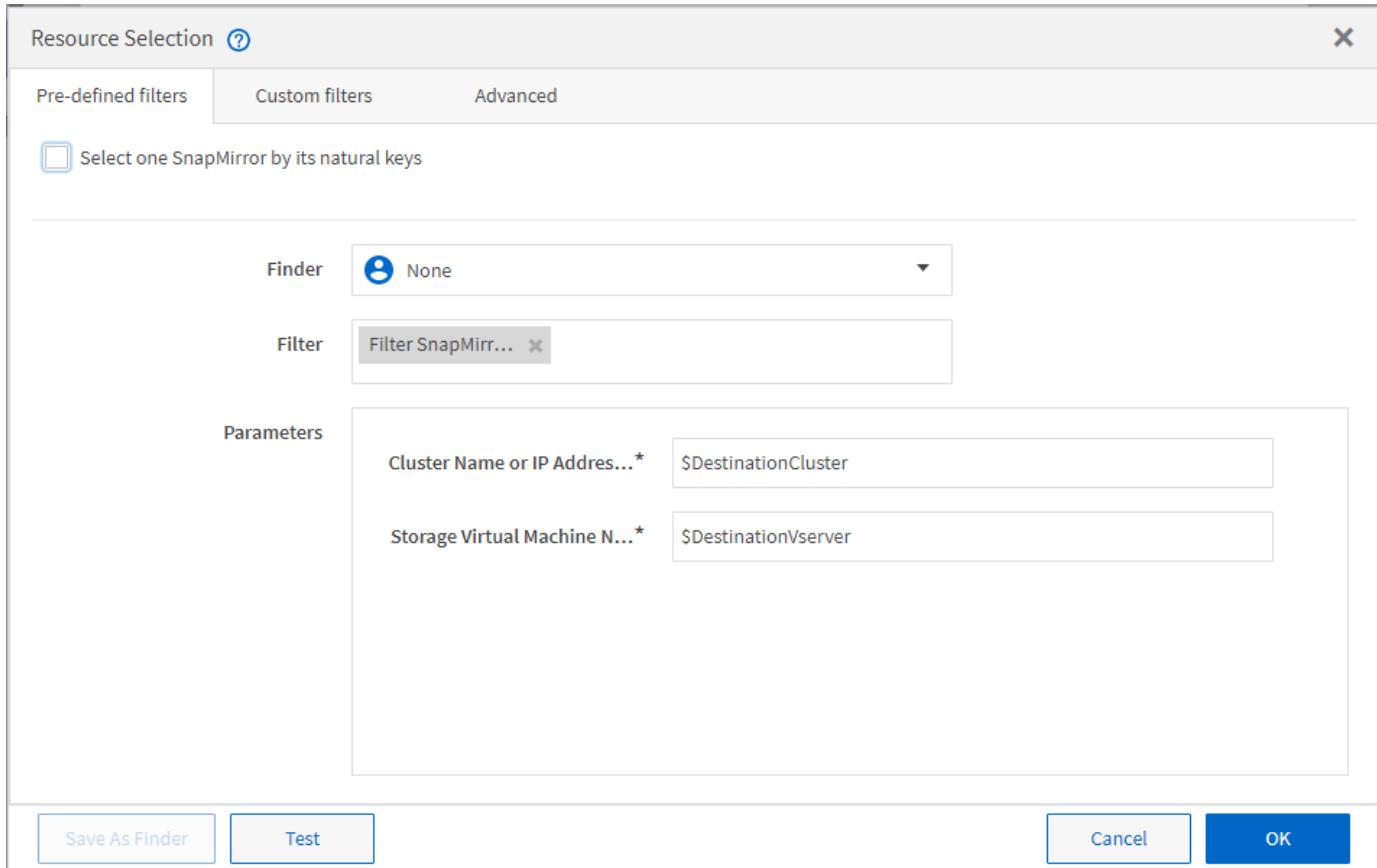
워크플로의 각 명령에 대한 자원 선택 기준을 정의해야 합니다. 또한 파인더를 사용하여 워크플로의 각 행에서 자원 선택 기준을 정의할 수 있습니다. 예를 들어, 특정 양의 저장 공간이 필요한 볼륨을 생성하는 경우 ""Create Volume"" 명령에서 ""Find aggregate by available capacity"" finder를 사용할 수 있습니다. 이 명령을 사용하면 특정 용량의 가용 공간이 있는 애그리게이트를 선택하고 해당 볼륨에 볼륨을 생성할 수 있습니다.

vFiler 유닛, 애그리게이트, 가상 머신 등의 사전 항목 리소스에 대한 필터 규칙 집합을 정의할 수 있습니다. 필터 규칙에는 하나 이상의 규칙 그룹이 포함될 수 있습니다. 규칙은 사전 항목 속성, 연산자 및 값으로 구성됩니다. 이 특성에는 해당 참조의 속성도 포함될 수 있습니다. 예를 들어, 다음과 같이 Aggregate에 대한 규칙을 지정할 수 있습니다. ""aggr"" 문자열로 시작하는 이름이 있고 사용 가능한 공간이 5GB를 초과하는 모든 집계를 나열합니다. 그룹의 첫 번째 규칙은 'name' 속성이고, 연산자 'starts-with', 'aggr' 값입니다. 같은 그룹에 대한 두 번째 규칙은 ""Available\_size\_MB""", 연산자 "">"", 값 ""5000""입니다. 공개 필터와 함께 필터 규칙 집합을 정의할 수 있습니다. 파인더를 선택한 경우 필터 규칙 정의 옵션이 비활성화됩니다. 필터 규칙 정의 확인란을 선택한 경우 Finder로 저장

옵션이 비활성화됩니다.

필터 및 찾기 외에도 검색 또는 정의 명령을 사용하여 사용 가능한 리소스를 검색할 수 있습니다. 검색 또는 정의 명령은 No-op 명령보다 선호되는 옵션입니다. 검색 및 정의 명령을 사용하여 인증된 사전 항목 유형과 사용자 지정 사전 항목 유형 모두의 리소스를 정의할 수 있습니다. 검색 또는 정의 명령은 리소스를 검색하지만 리소스에 대한 작업은 수행하지 않습니다. 그러나 finder를 사용하여 리소스를 검색할 경우 명령 컨텍스트에서 사용되고 명령에 의해 정의된 작업이 리소스에서 실행됩니다. 검색 또는 정의 명령으로 반환된 리소스는 워크플로의 다른 명령에 대한 변수로 사용됩니다.

다음 그림에서는 리소스 선택에 필터가 사용되었음을 보여 줍니다.



미리 정의된 워크플로의 리소스 선택 예

디자이너에서 미리 정의된 다음과 같은 워크플로의 명령 세부 정보를 열어 리소스 선택 옵션이 사용되는 방식을 이해할 수 있습니다.

- Clustered Data ONTAP NFS 볼륨을 생성합니다
- 클러스터 피어링을 설정합니다
- Clustered Data ONTAP 볼륨을 제거합니다

예약 방식

OnCommand Workflow Automation 리소스 예약 기능은 워크플로우를 성공적으로 실행하기 위해 리소스를 사용할 수 있도록 필요한 리소스를 예약합니다.

WFA 명령은 필요한 리소스를 예약하고 대개 캐시 획득 후 리소스가 WFA 캐시 데이터베이스에서 제공된 후에 예약을 제거할 수 있습니다. 예약 기능을 사용하면 WFA 구성 설정에서 구성한 예약 만료 기간 전까지 워크플로우에 대해 예약

리소스를 사용할 수 있습니다.

예약 기능을 사용하면 리소스를 선택하는 동안 다른 워크플로에서 예약한 리소스를 제외할 수 있습니다. 예를 들어 aggregate에서 100GB의 공간을 예약된 워크플로가 1주일 후에 실행되도록 예약된 경우, 또한 \* 볼륨 생성 \* 명령을 사용하는 다른 워크플로우를 실행 중입니다. 실행 중인 워크플로우에서 예약된 공간을 사용하여 새 볼륨을 생성하지 않습니다. 또한 예약 기능을 통해 워크플로우를 병렬로 실행할 수 있습니다.

실행 워크플로우를 미리 볼 때 WFA 플래너는 캐시 데이터베이스의 기존 개체를 비롯하여 예약된 모든 개체를 고려합니다. 예약을 사용하도록 설정한 경우, 계획된 워크플로우와 동시에 실행되는 워크플로우의 효과 및 스토리지 요소를 계획에 고려해야 합니다.

다음 그림의 화살표는 워크플로에 대해 예약이 활성화되어 있음을 나타냅니다.

## Workflow 'Abort SnapMirror relationship'

Details	Define Workflow	User Inputs	Constants	Return Parameters	Help Content	Advanced
Workflow Name*	Abort SnapMirror relationship					
Entity Version*	1.0.0					
Categories	Data Protection					
Workflow Description	The 'Abort SnapMirror' workflow stops ongoing transfers for a					
Ready For Production	<input checked="" type="checkbox"/>					
Consider Reserved Elements	<input checked="" type="checkbox"/> 					
Enable Element Existence Validation	<input checked="" type="checkbox"/>					
Minimum Software Versions	Clustered Data ONTAP 8.2.0					

사전 정의된 워크플로의 예약 예

Designer에서 다음과 같은 미리 정의된 워크플로를 열어 예약 사용 방법을 파악할 수 있습니다.

- 클론 환경
- Clustered Data ONTAP 볼륨을 생성합니다
- 클러스터 피어링을 설정합니다
- Clustered Data ONTAP 볼륨을 제거합니다

증분 이름은 입니다

증분 이름 지정은 매개 변수의 검색 결과에 따라 워크플로의 속성 이름을 지정할 수 있는 알고리즘입니다. 증분 값이나 사용자 정의 식을 기반으로 특성의 이름을 지정할 수 있습니다. 증분 명명 기능을 사용하면 요구 사항에 따라 명명 규칙을 구현할 수 있습니다.

워크플로우를 설계할 때 충분 명명 기능을 사용하여 워크플로우에서 생성한 객체의 이름을 동적으로 지정할 수 있습니다. 이 기능을 사용하면 리소스 선택 기능을 사용하여 개체에 대한 검색 기준을 지정할 수 있으며 검색 조건에 의해 반환되는 값은 개체의 특성에 사용됩니다. 또한 지정된 검색 조건으로 개체를 찾을 수 없는 경우 속성 값을 지정할 수 있습니다.

다음 옵션 중 하나를 사용하여 속성의 이름을 지정할 수 있습니다.

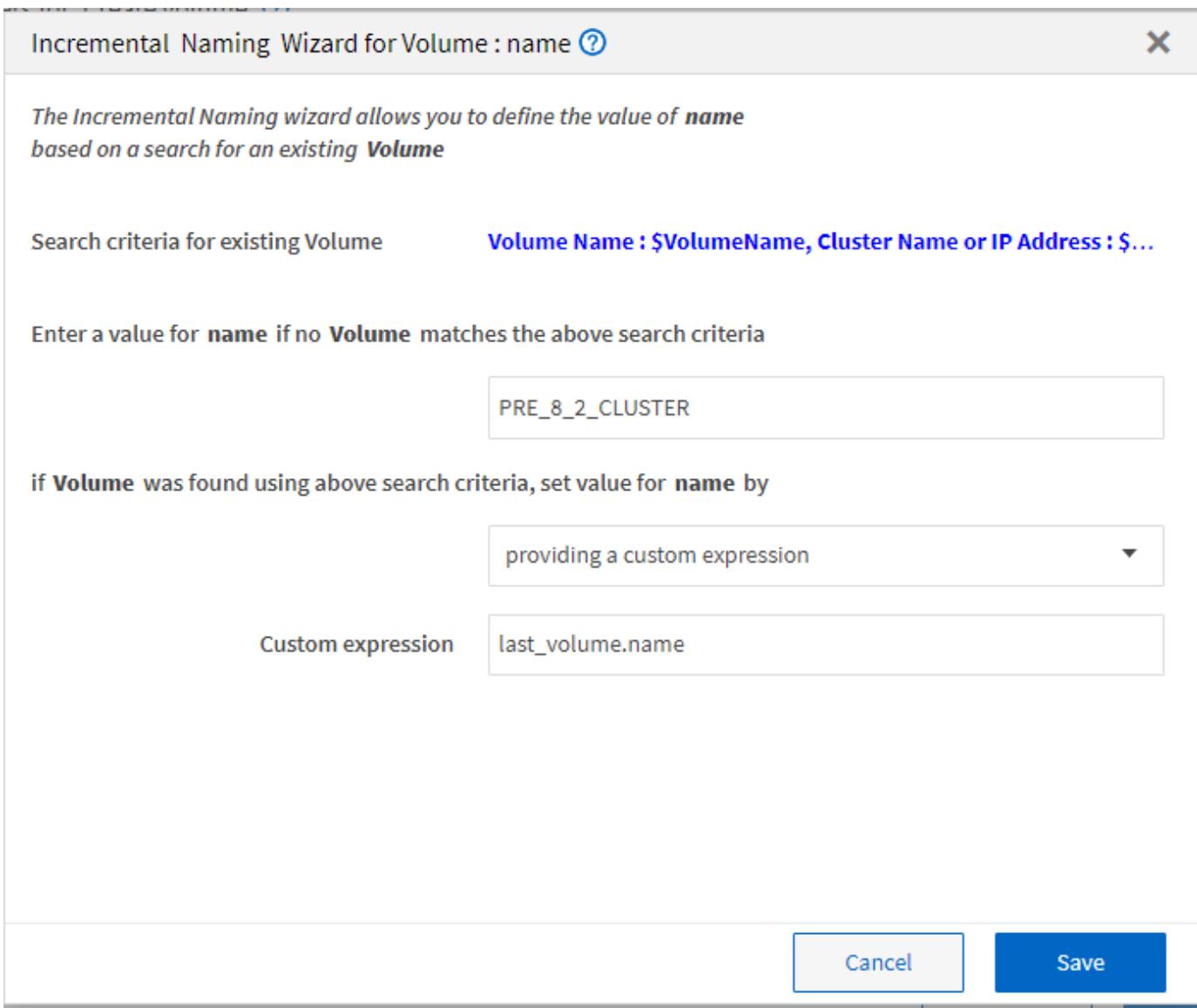
- 충분 값과 접미사를 제공합니다

검색 조건에 의해 찾은 개체 값과 함께 사용해야 하는 값을 제공하고 지정한 수로 충분할 수 있습니다. 예를 들어, "filer name\_unique number\_environment"의 명명 규칙을 사용하여 볼륨을 생성하려면 finder를 사용하여 이를 접두사로 마지막 볼륨을 찾고 고유 번호를 1씩 증가시키고 볼륨 이름에 접미사 이름을 추가할 수 있습니다. 검색된 마지막 볼륨 이름 접두사가 \_VF\_023prod이고 세 개의 볼륨을 생성하는 경우 생성된 볼륨의 이름은 \_VF\_024\_prod, VF\_025\_prod 및 VF\_026prod입니다.

- 사용자 정의 식 제공

검색 조건에 의해 찾은 개체 값과 함께 사용해야 하는 값을 제공하고 입력한 식에 따라 값을 추가할 수 있습니다. 예를 들어 "Last volume name\_environment name padded with 1"이라는 명명 규칙을 사용하여 볼륨을 생성하려면 "last\_volume.name+'\_'+nextName("lab1")" 식을 입력할 수 있습니다. 발견된 마지막 볼륨 이름이 VF\_023\_인 경우 생성된 볼륨의 이름은 \_VF\_023\_lab2입니다.

다음 그림에서는 명명 규칙을 지정하기 위해 사용자 정의 식을 제공하는 방법을 보여 줍니다.



## 조건부 실행이란

조건부 실행을 사용하면 지정된 조건이 충족될 때 명령을 실행할 수 있는 워크플로를 디자인할 수 있습니다.

워크플로우에서 명령을 실행하는 것은 동적인 작업이 될 수 있습니다. 워크플로의 각 명령 또는 명령 행의 실행 조건을 지정할 수 있습니다. 예를 들어, 특정 데이터세트가 있을 때만 ""데이터세트에 볼륨 추가"" 명령을 실행할 수 있으며 데이터 세트를 찾을 수 없을 경우 워크플로가 실패하지 않도록 할 수 있습니다. 이 경우 ""데이터세트에 볼륨 추가"" 명령을 사용하여 특정 데이터세트를 검색할 수 있으며, 이 명령이 없으면 워크플로우에서 해당 명령을 비활성화할 수 있습니다.

명령어의 조건부 실행을 위한 옵션은 '명령' 대화 상자의 '명령' 탭과 '고급' 탭에서 사용할 수 있습니다.

워크플로를 중단하거나 워크플로에서 특정 명령을 사용하지 않도록 설정할 수 있습니다. 또한 다음 옵션 중 하나를 사용하여 실행할 명령을 설정할 수 있습니다.

- 있습니다
- 지정한 변수를 찾을 수 있습니다

- 지정한 변수를 찾을 수 없는 경우
- 지정한 표현식이 TRUE이면

특정 시간 간격을 기다리도록 명령을 설정할 수도 있습니다.

미리 정의된 워크플로의 조건부 실행 예

Designer에서 다음과 같은 미리 정의된 워크플로의 명령 세부 정보를 열어 명령어의 조건부 실행 방법을 이해할 수 있습니다.

- 기본 clustered Data ONTAP 볼륨을 생성합니다
- Clustered Data ONTAP NFS 볼륨을 생성합니다

#### 반환 매개 변수의 작동 방식

반환 매개 변수는 워크플로의 계획 단계 이후에 사용할 수 있는 매개 변수입니다. 이러한 매개 변수에서 반환하는 값은 워크플로를 디버깅하는데 유용합니다. 반환 매개 변수의 작동 방식과 워크플로를 디버깅하기 위한 반환 매개 변수로 사용할 수 있는 매개 변수를 이해해야 합니다.

워크플로에서 변수 특성, 식 및 사용자 입력 값과 같은 매개 변수 집합을 반환 매개 변수로 지정할 수 있습니다. 워크플로 실행 중에 지정된 매개 변수의 값이 계획 단계에 채워지고 워크플로 실행이 시작됩니다. 그런 다음 이러한 매개 변수의 값이 워크플로의 특정 실행에서 계산된 방식으로 반환됩니다. 워크플로를 디버깅하려면 매개 변수에서 반환한 값을 참조할 수 있습니다.

이러한 매개 변수에 대해 계산되거나 선택된 값을 확인하려는 경우 워크플로에서 필수 반환 매개 변수를 지정할 수 있습니다. 예를 들어 자원 선택 논리를 사용하여 워크플로에서 집계를 선택할 때 집계를 반환 매개 변수로 지정하여 워크플로를 계획하는 동안 어떤 집계를 선택하였는지 볼 수 있습니다.

워크플로를 디버깅하기 위한 반환 매개 변수 값을 참조하기 전에 워크플로 실행이 완료되었는지 확인해야 합니다. 반환 매개 변수 값은 각 워크플로 실행에 대해 설정됩니다. 워크플로의 여러 실행 후에 반환 매개 변수를 추가한 경우에는 매개 변수를 추가한 후에만 해당 매개 변수의 값을 실행할 수 있습니다.

반환 매개 변수로 사용할 수 있는 매개 변수입니다

매개 변수를 반환합니다	예
스칼라 변수 특성입니다	"볼륨 이름" 변수의 속성인 volume1.name,'
상수	최대_볼륨_크기
사용자 입력	\$클러스터 이름
변수 특성, 상수 및 사용자 입력을 포함하는 MVEL 식입니다	volume1.name+'-'+\$clusterName
실행 중에 명령이 추가하는 반환 매개 변수입니다	PowerShell 명령에서 'Add-WfaWorkflowParameter-Name 'VolumeUUID'-Value"12345'-AddAsReturnParameter\$ true' 줄을 사용하면 '\$volumeUUID' 매개 변수가 반환 매개 변수로 추가됩니다

사전 정의된 워크플로의 반환 매개 변수 예

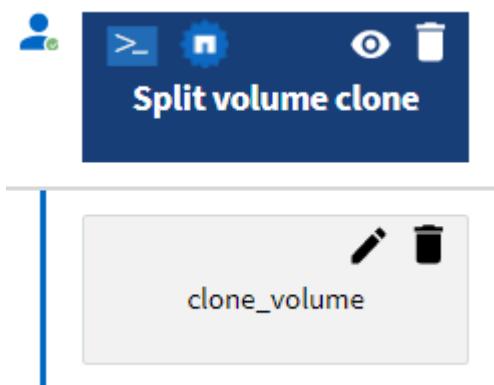
반환 매개 변수의 지정 방법을 이해하려면 Designer에서 다음 미리 정의된 워크플로를 열고 지정된 반환 매개 변수를 검토할 수 있습니다.

- vFiler에서 NFS 볼륨을 생성합니다
- vFiler에서 Qtree CIFS 공유를 생성합니다
- Clustered Data ONTAP 볼륨 CIFS 공유를 생성합니다

승인 지점은 무엇입니까

승인 지점은 워크플로에서 워크플로 실행을 일시 중지하고 사용자 승인에 따라 다시 시작하는 데 사용되는 확인 지점입니다.

다음 그림에 표시된 파란색 수직 막대는 승인점입니다.



워크플로를 점진적으로 실행하는 데 승인 지점을 사용할 수 있습니다. 이 경우 워크플로의 섹션은 특정 조건이 충족된 후에만 실행해야 합니다. 예를 들어, 다음 섹션을 승인해야 하거나 첫 번째 섹션을 성공적으로 실행해야 하는 경우를 들 수 있습니다. 승인 지점은 워크플로를 일시 중지하고 다시 시작하는 프로세스를 처리하지 않습니다. WFA 구성에 지정된 대로 이메일 및 SNMP 알림이 전송되고, 스토리지 운영자는 워크플로우 일시 중지 알림을 받을 때 특정 작업을 수행하도록 요청할 수 있습니다. 예를 들어, 스토리지 운영자는 승인을 위해 관리자, 승인자 또는 운영자에게 계획 세부 정보를 보내고 승인을 받으면 워크플로우를 재개할 수 있습니다.

승인이 항상 필요하지 않을 수 있습니다. 일부 시나리오에서는 특정 조건이 충족되고 승인 지점이 추가될 때 조건을 구성할 수 있는 경우에만 승인이 필요할 수 있습니다. 예를 들어, 볼륨의 크기를 늘리도록 설계된 워크플로우를 고려해 보십시오. 볼륨 크기가 증가하여 볼륨이 포함된 애그리게이트의 공간이 85% 사용되는 경우 스토리지 운영자가 관리자의 승인을 받을 수 있는 워크플로우 시작 시점에 승인 지점을 추가할 수 있습니다. 워크플로우 실행 중에 이 조건을 초래한 볼륨을 선택할 때, 실행이 승인될 때까지 실행이 중지됩니다.

승인 지점에 대해 설정된 조건은 다음 옵션 중 하나일 수 있습니다.

- 있습니다
- 지정한 변수가 발견되면
- 지정한 변수를 찾을 수 없는 경우
- 지정한 식이 true로 계산되면

워크플로의 승인 지점 수에 제한은 없습니다. 워크플로에서 명령 앞에 승인 지점을 삽입하고 승인 지점 뒤에 명령을 설정하여 실행 전에 승인을 기다릴 수 있습니다. 승인 지점은 변경 시간, 사용자, 메모 등의 정보를 제공하므로 워크플로

실행이 일시 중지 또는 재개된 시기와 이유를 확인할 수 있습니다. 승인 지점 메모에는 MVEL 식이 포함될 수 있습니다.

사전 정의된 워크플로의 승인점 예

디자이너에서 미리 정의된 다음 워크플로를 열어 승인 지점이 사용되는 방식을 이해할 수 있습니다.

- Clustered Data ONTAP 볼륨을 제거합니다
- 컨트롤러 및 HA 쌍의 쉘프 업그레이드
- 볼륨 마이그레이션

사용자 지정 REST 끝점을 실행하는 방법

OnCommand Workflow Automation(WFA)는 워크플로우를 실행하도록 사용자 지정 REST 끝점을 구성하는 메커니즘을 제공합니다. 사용자 지정 REST 끝점은 설계자가 이해하기 쉽고 직관적이며 통일된 URI(리소스 식별자)를 구성하여 워크플로 의미론을 기반으로 POST, PUT 또는 삭제의 나머지 규칙을 따르는 워크플로를 실행할 수 있도록 도와줍니다. 이러한 URI는 클라이언트 개발자를 위한 클라이언트 코드 개발을 용이하게 합니다.

WFA를 사용하면 API 호출을 통해 워크플로우 실행을 위한 사용자 지정 URI 경로를 구성할 수 있습니다. URI 경로의 각 세그먼트는 "/DevOps/{ProjectName}/clone"과 같이 대괄호로 묶인 워크플로의 사용자 입력의 문자열 또는 유효한 이름일 수 있습니다. 이 워크플로는 '[https://WFAServer:HTTPS\\_PORT/REST/DevOps/Project1/clone/jobs](https://WFAServer:HTTPS_PORT/REST/DevOps/Project1/clone/jobs)'에 대한 호출로 호출할 수 있습니다

URI 경로에 대한 유효성 검사는 다음과 같습니다.

- 나머지 경로는 """/"로 시작해야 합니다.
- 허용되는 문자는 영문자, 숫자 및 밑줄입니다.
- 사용자 입력 이름은 ""{}"(으)로 둘러싸야 합니다.



""{}"에 둘러싸인 값이 올바른 사용자 입력 이름인지 확인해야 합니다.

- //, /{}/ 등의 빈 경로 세그먼트가 없어야 합니다.
- HTTP 메서드 구성과 사용자 지정 URI 경로 구성은 둘 다 구성되거나 구성되지 않아야 합니다.

장애 시 계속 진행 방법

실패 시 계속 기능을 사용하면 단계가 실패하더라도 워크플로 실행이 계속될 수 있도록 워크플로의 단계를 구성할 수 있습니다. 실패한 단계를 처리하고 'WFA.log' 파일에 액세스하거나 를 클릭하여 단계에서 실패하는 문제를 해결할 수 있습니다 아이콘을 클릭합니다.

실패한 단계가 하나 이상 있는 워크플로는 실행이 완료된 후 부분적으로 성공한 상태입니다. 명령\_이름> 매개 변수 대화 상자의 고급 탭에서 필요한 옵션을 선택하여 단계가 실패한 경우에도 워크플로우 실행이 계속되도록 단계를 구성할 수 있습니다.

실패 시 단계를 계속하도록 구성하지 않으면 단계가 실패할 경우 워크플로 실행이 중단됩니다.

실패 시 계속하도록 구성된 단계가 실패하는 경우 다음 옵션 중 하나를 사용하여 워크플로가 실행되도록 설정할 수 있습니다.

- 워크플로 실행 중단(기본 옵션)
- 다음 단계에서 실행을 계속합니다
- 다음 행에서 실행을 계속합니다

#### 워크플로 요구 사항 체크리스트 예

워크플로 요구 사항 체크리스트에는 계획된 워크플로의 명령, 사용자 입력 및 리소스 등의 자세한 요구 사항이 포함되어 있습니다. 체크리스트를 사용하여 워크플로우를 계획하고 요구사항의 차이를 파악할 수 있습니다.

#### 요구 사항 체크리스트 예

다음 워크플로우 요구 사항 체크리스트에는 "clustered Data ONTAP 볼륨 생성" 워크플로우 요구 사항이 나와 있습니다. 이 샘플 체크리스트를 템플릿으로 사용하여 워크플로 요구 사항을 나열할 수 있습니다.

요구 사항	설명
워크플로 이름	Clustered Data ONTAP 볼륨을 생성합니다
범주	스토리지 프로비저닝
설명	워크플로우에서 특정 SVM에 새 볼륨을 생성합니다. 이 워크플로는 나중에 사용할 수 있도록 볼륨이 프로비저닝되고 위임된 경우에 사용됩니다.
워크플로 작동 방식에 대한 고급 설명	<ul style="list-style-type: none"> <li>• 볼륨이 포함된 SVM은 사용자(클러스터, SVM 이름)에서 지정합니다.</li> <li>• 볼륨은 지정된 크기를 기반으로 생성됩니다.</li> <li>• 볼륨 구성은 템플릿에 설명되어 있습니다.</li> </ul>

요구 사항	설명
세부 정보	<ul style="list-style-type: none"> <li>• Create CM Volume * 명령을 사용합니다</li> <li>• CM 볼륨 생성 * 에 대한 명령 세부 정보: <ul style="list-style-type: none"> <li>◦ 실행은 항상 으로 설정됩니다</li> <li>◦ 볼륨 세부 사항은 볼륨에 대한 속성을 채워 지정합니다</li> <li>◦ 공간 보장 설정 * 템플릿을 사용하여 볼륨을 구성합니다</li> <li>◦ 볼륨 이름 및 크기는 사용자가 제공합니다.</li> </ul> </li> <li>볼륨은 SVM 네임스페이스에서 '/volname'(루트 네임스페이스 아래)으로 마운트됩니다.</li> <li>◦ 스냅 예약 공간이 5%가 되기 때문에 * actualVolumeSize * 기능을 사용합니다.</li> <li>◦ SVM 참조는 다음과 같은 리소스 선택 로직으로 정의됩니다. <ul style="list-style-type: none"> <li>▪ CM SVM by key — 사용자가 제공하는 이름 및 클러스터를 기준으로 SVM을 검색합니다</li> <li>▪ 유형 전용 데이터 SVM별 CM SVM(유형 =클러스터)</li> <li>▪ 상태별 SVM --(상태 = 실행 중)</li> </ul> </li> <li>◦ 애그리게이트 참조는 리소스 선택 로직으로 사전 정의된 finder(공간 임계값 및 RAID 유형에 따른 CM Aggregate)로 정의됩니다. <ul style="list-style-type: none"> <li>▪ 가용 용량별 CM 애그리게이트(용량 = 프로비저닝할 볼륨의 크기, 사용자가 제공한 클러스터)</li> <li>▪ SVM으로 위임하여 CM 애그리게이트</li> <li>▪ RAID 유형별 CM Aggregate(RAID-DP)</li> <li>▪ CM 집계 값이 aggr0이 아닙니다</li> <li>▪ 사용된 크기별 CM 집계(% 임계값 = 90, spacewToBeProvisioned = 제공된 크기, 볼륨 보장이므로)</li> <li>▪ 초과 약속에 의한 CM 집계(임계값 = 300, spacebToBeAllocated = 프로비저닝되는 볼륨의 크기)</li> <li>▪ 사용 가능한 최대 공간이 있는 애그리게이트를 선택합니다</li> </ul> </li> </ul>

이름	유형	설명(데이터 값, 유효성 검사 등)
클러스터	잠긴 쿼리(테이블 형식)	<ul style="list-style-type: none"> <li>SVM을 호스팅하는 클러스터</li> <li>쿼리는 클러스터의 기본 주소와 이름을 사용하여 표 형식으로 표시할 수 있습니다</li> <li>이름을 기준으로 사전순으로 정렬합니다</li> </ul>
SVM	잠긴 쿼리	<ul style="list-style-type: none"> <li>볼륨이 프로비저닝되는 SVM</li> <li>쿼리는 이전 입력에서 선택한 클러스터에 속한 SVM 이름만 표시해야 합니다</li> </ul> <p>admin 또는 node가 아닌 클러스터 유형 SVM만 표시합니다(cm_storage.vserver의 열 유형).</p> <ul style="list-style-type: none"> <li>알파벳순으로 정렬합니다</li> </ul>
볼륨	문자열	<ul style="list-style-type: none"> <li>생성할 볼륨의 이름입니다</li> </ul>
크기(GB)	정수	<ul style="list-style-type: none"> <li>프로비저닝할 볼륨의 크기입니다</li> <li>데이터 크기(스냅 예약 공간을 고려해야 함)</li> </ul>

- 명령 \*

이름	설명	상태
CM 볼륨을 생성합니다	SVM에서 볼륨 생성	기존

- 매개 변수 반환 \*

이름	값
볼륨 이름입니다	프로비저닝된 볼륨의 이름입니다
애그리게이트 이름입니다	선택한 애그리게이트의 이름입니다
노드 이름	노드의 이름입니다
클러스터 이름입니다	클러스터의 이름입니다

- 격차 및 문제 \*

1.	
2.	
3.	
4.	
5.	

워크플로를 만듭니다

WFA(Workflow Automation)를 사용하여 데이터베이스 또는 파일 시스템의 스토리지 프로비저닝, 마이그레이션, 서비스 해제 등의 작업을 위한 워크플로우를 생성할 수 있습니다. 사전 정의된 WFA 워크플로우가 귀사의 요구사항과 일치하지 않을 경우 워크플로우를 생성해야 합니다.

#### 필요한 것

- WFA 구성 요소에 대한 개념을 이해해야 합니다.
- 워크플로에 필요한 반복 행, 승인 지점 및 자원 선택과 같은 기능을 이해해야 합니다.
- 워크플로 요구 사항 체크리스트를 포함하여 워크플로에 필요한 계획을 완료해야 합니다.
- 스토리지 운영자에게 워크플로우에 대한 정보를 제공하는 도움말 콘텐츠를 만들어야 합니다.

#### 이 작업에 대해

각 워크플로의 구조는 워크플로의 목표와 요구 사항에 따라 달라질 수 있습니다. 이 작업에서는 특정 워크플로에 대한 지침을 제공하지 않지만 워크플로를 만들기 위한 일반적인 지침을 제공합니다.

#### 단계

1. 워크플로우 설계 \* > \* 워크플로우 \* 를 클릭합니다.
2. 을 클릭합니다  를 클릭합니다.
3. Workflow \* 탭에서 다음 단계를 수행하십시오.

- a. 필요한 스키마를 확장한 다음 필요한 스키마를 두 번 클릭합니다  (명령) 또는  (워크플로) \* 사용 가능한 단계 \* 목록에서 선택합니다.

필요에 따라 이 단계를 반복할 수 있습니다. 단계를 끌어서 놓아 워크플로 편집기의 단계를 다시 정렬할 수 있습니다.

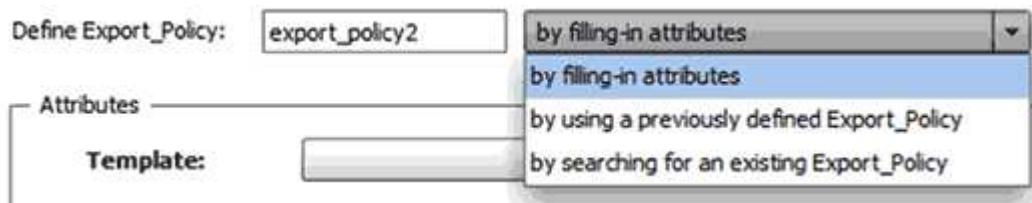
- b. \* 선택 사항: \* 를 클릭합니다  필요한 행 수를 추가하는 데 사용되며, 이 수는 단계 실행을 위한 세부 정보를 지정하는 데 사용됩니다.

각 단계는 지정된 행 및 열의 지정된 단계 세부 정보를 기반으로 실행됩니다. 단계는 왼쪽에서 오른쪽으로, 위에서 아래로 실행됩니다.

- c. 추가한 단계 아래에 커서를 놓고 클릭합니다  필요한 행에서 단계 실행에 대한 단계 세부 정보를 추가합니다.

이 단계의 경우...	수행할 작업...
워크플로우	Workflow *(작업 흐름 *) 탭에 필요한 사용자 입력을 입력하고 * Advanced *(고급 *) 탭에 필요한 조건을 입력합니다.
명령	명령> 탭의 매개 변수 탭에서 각 개체 탭을 클릭하고 필요한 옵션을 선택하여 개체 특성을 정의한 다음 고급 탭과 기타 매개 변수 탭에 필요한 세부 정보를 입력합니다.
검색 또는 정의	검색하거나 정의해야 하는 사전 항목 개체를 선택합니다.

다음 그림에서는 개체 특성을 정의하는 데 사용할 수 있는 옵션을 보여 줍니다.



적절한 작업을 선택합니다.

대상...	수행할 작업...
속성을 채웁니다	<p>다음 옵션을 사용하여 속성 값을 입력합니다.</p> <ul style="list-style-type: none"> <li>• 표현식</li> <li>• 변수</li> <li>• 사용자 입력</li> <li>• 리소스 선택</li> <li>• 증분 이름 지정</li> </ul> <p>속성 필드 위에 커서를 놓고 를 클릭해야 합니다 [...] 리소스 선택 또는 증분 이름 지정 기능을 사용합니다.</p>
이전에 정의된 'object'를 사용합니다	옵션 목록 앞의 상자에서 이전에 정의한 '개체'를 선택합니다.

대상...	수행할 작업...
기존의 'object'을(를) 검색합니다	<ul style="list-style-type: none"> <li>i. 리소스 선택 기능을 사용하여 객체를 검색하려면 * 검색 기준 입력 * 을 클릭합니다.</li> <li>ii. 필요한 개체를 찾을 수 없는 경우 실행에 필요한 옵션 중 하나를 선택합니다.           <ul style="list-style-type: none"> <li>◦ 워크플로를 중단합니다</li> </ul> <p>이 옵션은 특정 개체를 찾을 수 없는 경우 워크플로 실행을 중단합니다.</p> <ul style="list-style-type: none"> <li>◦ 이 명령을 비활성화합니다</li> </ul> </li> </ul> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>This option disables only the current step and executes the workflow. **** 'object'에 대한 속성을 채우고 명령어를 수행한다</p> </div> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>This option enables you to enter the required attributes and execute the workflow.</p> </div>

4. 승인 지점을 삽입하려면  승인 지점에 필요한 설명을 입력합니다.

승인 지점 메모에는 MVEL 식이 포함될 수 있습니다.

5. 을 클릭합니다  행 번호 옆에 있는 다음 작업을 수행합니다.

- 행을 삽입합니다.
- 행을 복사합니다.
- 행을 반복합니다.

다음 옵션 중 하나를 사용하여 명령 매개 변수의 반복을 지정할 수 있습니다.

▪ 횟수

이 옵션을 사용하여 지정한 반복 횟수만큼 명령 실행을 반복할 수 있습니다. 예를 들어, ""qtree 생성"" 명령을 세 번 반복하여 세 개의 qtree를 만들도록 지정할 수 있습니다.

이 옵션을 사용하여 동적 명령 실행 수를 지정할 수도 있습니다. 예를 들어, 생성할 LUN 수에 대한 사용자 입력 변수를 생성하고 워크플로우를 실행하거나 스케줄을 지정할 때 스토리지 운영자가 지정한 번호를 사용할 수 있습니다.

- 그룹의 모든 리소스에 대해

이 옵션을 사용한 다음 개체의 검색 조건을 지정할 수 있습니다. 이 명령은 검색 조건에 의해 개체가 반환되는 횟수만큼 반복됩니다. 예를 들어, 클러스터에서 노드를 검색하고 각 노드에 대해 ""iSCSI 논리 인터페이스 생성"" 명령을 반복할 수 있습니다.

- 행 실행을 위한 조건을 추가합니다.
- 행을 제거합니다.

6. Details \* 탭에서 다음 단계를 수행하십시오.

- 워크플로 이름 \* 및 \* 워크플로 설명 \* 필드에 필요한 정보를 지정합니다.

워크플로 이름 및 설명은 각 워크플로에 대해 고유해야 합니다.

- \* 선택 사항: \* 엔터티 버전을 지정합니다.
- \* 선택 사항: \* 예약 기능을 사용하지 않으려면 \* 예약된 요소 고려 \* 확인란을 지웁니다.
- \* 선택 사항: \* 같은 이름의 요소에 대한 유효성 검사를 사용하지 않으려면 \* 요소 존재 유효성 검사 사용 \* 확인란의 선택을 취소합니다.

7. 사용자 입력을 편집하려면 다음 단계를 수행하십시오.

- 사용자 입력 \* 탭을 클릭합니다.
- 편집할 사용자 입력을 두 번 클릭합니다.
- 변수 편집:<사용자 입력> \* 대화 상자에서 사용자 입력을 편집합니다.

8. 상수를 추가하려면 다음 단계를 수행하십시오

- 상수 \* 탭을 클릭한 다음 \* 추가 \* 단추를 사용하여 워크플로에 필요한 상수를 추가합니다.

여러 명령에 대한 매개 변수를 정의하기 위해 공통 값을 사용할 때 상수를 정의할 수 있습니다. 예를 들어 ""SnapVault을 사용하여 LUN 생성, 매픽 및 보호' 워크플로우에 사용되는 aggregate\_over오버커밋\_threshold 상수를 참조하십시오.

- 각 상수의 이름, 설명 및 값을 입력합니다.

9. 반환 매개 변수 \* 탭을 클릭한 다음 \* 추가 \* 버튼을 사용하여 워크플로에 필요한 매개 변수를 추가합니다.

워크플로 계획 및 실행이 계획 중에 계산되거나 선택된 일부 값을 반환해야 하는 경우 반환 매개 변수를 사용할 수 있습니다. 계산 또는 선택한 값을 Workflow 미리 보기의 모니터링 창에 있는 Return Parameters(매개 변수 반환) 탭 또는 워크플로 실행이 완료된 후에 볼 수 있습니다.

Aggregate: Aggregate를 반환 매개 변수로 지정하여 리소스 선택 로직을 사용하여 선택된 Aggregate를 확인할 수 있습니다.

워크플로에 하위 워크플로를 포함했고 하위 워크플로 반환 매개 변수 이름에 공백, 달러 기호(\$)가 포함된 경우 또는 상위 워크플로의 하위 워크플로 반환 매개 변수 값을 보려면 부모 워크플로의 대괄호 안에 반환 매개 변수 이름을 지정해야 합니다.

매개 변수 이름이...	다음으로 지정...
'ChildWorkflow1.abc\$value'	'ChildWorkflow1["abc\$"+"value"]'

매개 변수 이름이...	다음으로 지정...
'ChildWorkflow1.\$value'	'ChildWorkflow1["\$" + "value"]'
'ChildWorkflow1.value\$'	'ChildWorkflow1.value\$'
ChildWorkflow1.P N	'ChildWorkflow1["P N"]'
"ChildWorkflow1.return_string("hw")"	"ChildWorkflow1["return_string(\"hw\")"]"

10. \* 선택 사항: \* 도움말 콘텐츠 \* 탭을 클릭하여 워크플로에 대해 만든 도움말 콘텐츠 파일을 추가합니다.
11. 미리 보기 \* 를 클릭하고 워크플로 계획이 성공적으로 완료되었는지 확인합니다.
12. 확인 \* 을 클릭하여 미리보기 창을 닫습니다.
13. 저장 \* 을 클릭합니다.

작업을 마친 후

테스트 환경에서 워크플로를 테스트한 다음 \*WorkflowName\* > \*Details\*에서 워크플로를 생산 준비 상태로 표시합니다.

워크플로 도움말 콘텐츠를 만듭니다

WFA(OnCommand Workflow Automation) 관리자와 설계자로, 워크플로우를 설계할 때는 워크플로우에 대한 도움말 콘텐츠를 생성하고 워크플로우에 포함할 수 있습니다.

필요한 것

HTML을 사용하여 웹 페이지를 만드는 방법을 알고 있어야 합니다.

이 작업에 대해

이 도움말은 워크플로우를 실행하는 스토리지 운영자에게 워크플로우에 대한 워크플로우 및 사용자 입력에 대한 정보를 제공해야 합니다.

단계

1. workflow-help라는 이름의 폴더를 만듭니다.
2. HTML 편집기나 텍스트 편집기를 사용하여 도움말 콘텐츠를 만든 다음 워크플로 도움말 폴더에 index.htm 파일로 저장합니다.

JavaScript 파일은 도움말 콘텐츠의 일부로 포함해서는 안 됩니다. 지원되는 파일 확장자는 다음과 같습니다.

- jpg
- jpeg
- gif
- png입니다
- XML
- thmx

- htm
- .html을 참조하십시오
- CSS

Windows에서 만든 thumb.db 파일도 포함할 수 있습니다.

3. 워크플로 도움말 폴더에서 도움말 콘텐츠와 관련된 색인 .htm 파일과 기타 파일을 사용할 수 있는지 확인합니다.
4. 폴더의 '.zip' 파일을 만들고 '.zip' 파일의 크기가 2MB를 넘지 않도록 합니다.

NFS volume-help.zip을 만듭니다

5. 도움말 콘텐츠를 만든 워크플로를 편집한 다음 \* 설정 \* > \* 도움말 콘텐츠 \* > \* 찾아보기 \* 를 클릭하여 '.zip' 파일을 업로드합니다.

## **WFA 워크플로우 팩을 생성합니다**

스토리지 자동화 및 통합 요구사항에 맞는 WFA(OnCommand Workflow Automation)에서 워크플로우 팩을 생성할 수 있습니다.

단계

1. 웹 브라우저를 통해 WFA 창에 로그인합니다.
2. 콘텐츠 관리 \* > \* 팩 \* 을 클릭합니다.
3. 새 팩 \* 아이콘을 클릭합니다.
4. 새 팩 \* 대화 상자에서 \* 이름 \*, \* 작성자 \*, \* 버전 \* 및 \* 설명 \* 필드에 값을 입력합니다.
5. 저장 \* 을 클릭합니다.
6. 새 팩이 \* Packs \* 창에 생성되었는지 확인합니다.

## **WFA 워크플로우 팩에 엔터티를 추가합니다**

WFA(OnCommand Workflow Automation)에서 스토리지 자동화 및 통합 요구사항에 대해 하나 이상의 엔터티를 워크플로우 팩에 추가할 수 있습니다.

이 작업에 대해

다음 요소에서 팩을 제거할 수 있습니다.

- 워크플로우
- 파인더
- 필터
- 명령
- 기능
- 템플릿
- 구성표
- 사전

- 데이터 소스 형식
- 원격 시스템 유형
- 캐시 쿼리
- 범주

#### 단계

1. 웹 브라우저를 통해 \* WFA \* 창에 로그인합니다.
2. Workflow Design 포털에서 \* 추가할 엔터티를 찾아 \* <Entity> \* 를 클릭합니다.
3. Entity \* 창에서 팩에 추가할 요소를 선택합니다.
4. Add to Pack \* 아이콘을 클릭합니다.

"Add to Pack"은 인증이 \* None으로 설정된 엔터티에 대해서만 활성화됩니다

5. Add to Pack <Entity> \* (팩에 추가) 대화 상자의 \* Available Packs \* (사용 가능한 팩 \*) 드롭다운 목록에서 엔터티를 추가할 팩을 선택합니다.
6. 확인 \* 을 클릭합니다.

#### OnCommand Workflow Automation 팩을 삭제합니다

더 이상 필요하지 않은 경우 WFA(OnCommand Workflow Automation)에서 팩을 삭제할 수 있습니다. 팩을 삭제하면 팩과 연관된 모든 요소가 삭제됩니다.

#### 이 작업에 대해

- 팩의 일부인 엔터티에 종속성이 있는 경우 팩을 삭제할 수 없습니다.

예를 들어 사용자 지정 워크플로의 일부인 명령이 포함된 팩을 삭제하려고 하면 사용자 지정 워크플로가 팩에 종속되므로 삭제 작업이 실패합니다. 사용자 지정 워크플로를 삭제한 후에만 팩을 삭제할 수 있습니다.

- 팩의 일부인 엔터티는 개별적으로 삭제할 수 없습니다.

팩의 일부인 엔터티를 삭제하려면 해당 엔터티가 포함된 팩을 삭제해야 합니다. 엔터티가 여러 팩의 일부인 경우 해당 엔터티가 해당 엔터티가 포함된 모든 팩이 삭제된 경우에만 WFA 서버에서 삭제됩니다.

#### 단계

1. 웹 브라우저를 통해 admin으로 WFA에 로그인합니다.
2. 콘텐츠 관리 \* > \* 팩 \* 을 클릭합니다.
3. 삭제할 팩을 선택하고 를 클릭합니다 .
4. 팩 \* 삭제 확인 대화 상자에서 \* 확인 \* 을 클릭합니다.

#### OnCommand Workflow Automation 콘텐츠를 내보냅니다

사용자가 생성한 WFA(OnCommand Workflow Automation) 콘텐츠를 '.dar' 파일로 저장하고 다른 사용자와 콘텐츠를 공유할 수 있습니다. WFA 콘텐츠에는 사용자가 생성한 전체 콘텐츠 또는 워크플로우, 찾기, 명령, 사전 용어 등의 특정 항목이 포함될 수 있습니다.

## 필요한 것

- 내보낼 WFA 콘텐츠에 액세스할 수 있어야 합니다.
- 내보낼 콘텐츠에 인증된 콘텐츠에 대한 참조가 포함되어 있는 경우 콘텐츠를 가져올 때 해당 인증된 콘텐츠 팩을 시스템에서 사용할 수 있어야 합니다.

이러한 팩은 Storage Automation Store에서 다운로드할 수 있습니다.

## 이 작업에 대해

- 다음 유형의 인증된 콘텐츠는 내보낼 수 없습니다.
  - 인증받은 NetApp 컨텐츠
  - ?PS(Professional Services)에서 개발한 컨텐츠로, PS에서 설치한 사용자 정의 설치에서만 사용할 수 있습니다
  - 사용자가 개발한 팩입니다
- 내보낸 개체에 종속된 모든 개체도 내보내집니다.

예를 들어 워크플로를 내보내면 워크플로에 대한 종속 명령, 필터 및 파イン더도 내보내집니다.

- 잠긴 개체를 내보낼 수 있습니다.

다른 사용자가 개체를 가져올 때 해당 개체는 잠긴 상태로 유지됩니다.

## 단계

- 웹 브라우저를 통해 WFA에 로그인합니다.
- 필요한 콘텐츠 내보내기:

원하는 작업	수행할 작업...
사용자가 만든 모든 콘텐츠를 단일 '.dar' 파일로 내보냅니다	<ol style="list-style-type: none"><li>설정 * 을 클릭하고 * 유지 관리 * 에서 * 모든 워크플로 내보내기 * 를 클릭합니다.</li><li>'.dar' 파일의 파일 이름을 지정한 다음 * 내보내기 * 를 클릭합니다.</li></ol>
특정 콘텐츠 내보내기	<ol style="list-style-type: none"><li>콘텐츠를 내보낼 창으로 이동합니다.</li><li>창에서 하나 이상의 항목을 선택한 다음 을 클릭합니다 .</li><li>내보내기 형식 대화 상자에서 '.dar' 파일의 파일 이름을 지정한 다음 * 내보내기 * 를 클릭합니다.</li></ol>

- 다른 이름으로 저장 \* 대화 상자에서 '.dar' 파일을 저장할 위치를 지정한 다음 \* 저장 \* 을 클릭합니다.

## OnCommand Workflow Automation 콘텐츠를 가져옵니다

WFA(사용자 생성 OnCommand Workflow Automation) 콘텐츠(예: 워크플로우, 파イン더 및 명령)를 가져올 수 있습니다. 또한 다른 WFA 설치에서 내보낸 콘텐츠, 스토리지 자동화 스토어

또는 WFA 커뮤니티에서 다운로드한 콘텐츠, Data ONTAP PowerShell 툴킷, Perl NMSDK 툴킷을 포함한 팩 등을 가져올 수 있습니다.

#### 필요한 것

- 가져올 WFA 콘텐츠에 액세스할 수 있어야 합니다.
- 가져오려는 콘텐츠가 동일한 버전 또는 이전 버전의 WFA를 실행 중인 시스템에 생성되어 있어야 합니다.

예를 들어, WFA 2.2를 실행 중인 경우 WFA 3.0을 사용하여 생성한 콘텐츠를 가져올 수 없습니다.

- WFA의 N-2 버전에서 개발한 콘텐츠는 WFA 5.1로만 가져올 수 있습니다.
- '.dar' 파일에서 NetApp 인증 콘텐츠를 참조하는 경우 NetApp 인증 콘텐츠 팩을 가져와야 합니다.

NetApp 인증 콘텐츠 팩은 스토리지 자동화 스토어에서 다운로드할 수 있습니다. 팩 설명서를 참조하여 모든 요구 사항이 충족되었는지 확인해야 합니다.

#### 단계

1. 웹 브라우저를 통해 WFA에 로그인합니다.
2. 설정 \* 을 클릭하고 \* 유지 관리 \*에서 \* 워크플로 가져오기 \*를 클릭합니다.
3. 가져올 '.dar' 파일을 선택하려면 \* 파일 선택 \* 을 클릭하고 \* 가져오기 \*를 클릭합니다.
4. 가져오기 성공 \* 대화 상자에서 \* 확인 \*을 클릭합니다.

#### WFA 워크플로우 팩을 가져옵니다

스토리지 자동화 및 통합 요구사항에 맞게 서버에서 WFA(OnCommand Workflow Automation)로 워크플로우 팩을 가져올 수 있습니다.

#### 필요한 것

가져올 서버의 WFA 콘텐츠에 액세스할 수 있어야 합니다.

#### 단계

1. 웹 브라우저를 통해 \* WFA \* 창에 로그인합니다.
2. 콘텐츠 관리 \* > \* 팩 \* 을 클릭합니다.
3. 서버에서 가져오기 \* 아이콘을 클릭합니다.
4. 서버 폴더에서 가져오기 \* 대화 상자의 \* 서버 시스템 \*의 폴더 위치 필드에 서버의 팩 위치를 문자열 형식(예: 'C:\work\pack\test')으로 입력합니다.
5. 확인 \* 을 클릭합니다.
6. 팩을 \* Packs \* (팩 \*) 창에서 가져왔는지 확인합니다.

#### OnCommand Workflow Automation 콘텐츠를 가져오는 동안 고려해야 할 사항입니다

사용자가 생성한 콘텐츠, WFA(OnCommand Workflow Automation) 설치에서 내보낸 콘텐츠 또는 스토리지 자동화 스토어 또는 WFA 커뮤니티에서 다운로드한 콘텐츠를 가져올 때는 주의해야 할 몇 가지 사항이 있습니다.

- WFA 콘텐츠는 '.dar' 파일로 저장되며 다른 시스템의 전체 사용자 생성 콘텐츠나 워크플로우, 찾기, 명령, 사전 용어 등의 특정 항목을 포함할 수 있습니다.
- 기존 범주를 '.dar' 파일에서 가져오면 가져온 콘텐츠가 범주의 기존 컨텐츠와 병합됩니다.

예를 들어, WFA 서버의 범주 A에 WF1 및 WF2의 두 가지 워크플로우가 있다고 가정해 보겠습니다. 범주 A의 워크플로 WF3 및 WF4 를 WFA 서버로 가져오면 범주 A에 가져오기 후 워크플로 WF1, WF2, WF3 및 WF4 가 포함됩니다.

- '.dar' 파일에 사전 항목이 있으면 해당 사전 항목에 해당하는 캐시 테이블이 자동으로 업데이트됩니다.

캐시 테이블이 자동으로 업데이트되지 않으면 'WFA.log' 파일에 오류 메시지가 기록됩니다.

- WFA 서버에 없는 팩에 종속된 '.DAR' 파일을 가져올 때 WFA는 엔터티의 모든 종속성이 충족되는지 여부를 확인하려고 시도합니다.
  - 하나 이상의 엔터티가 없거나 더 낮은 버전의 엔터티가 발견되면 불러오기에 실패하고 오류 메시지가 표시됩니다.

오류 메시지는 종속성을 충족시키기 위해 설치해야 하는 팩의 세부 정보를 제공합니다.

◦ 더 높은 버전의 엔터티가 발견되거나 인증이 변경된 경우 버전 불일치에 대한 일반 대화 상자가 표시되고 가져오기가 완료됩니다.

버전 불일치 세부 정보는 'WFA.log' 파일에 기록됩니다.

- 다음 사항에 대한 질문 및 지원 요청은 WFA 커뮤니티로 리디렉션해야 합니다.
  - WFA 커뮤니티에서 다운로드한 모든 콘텐츠
  - 생성한 맞춤형 WFA 콘텐츠입니다
  - 수정한 WFA 콘텐츠입니다

#### 업그레이드 중 팩 식별

업그레이드 프로세스 중에 OnCommand Workflow Automation(WFA)는 엔터티를 하나의 팩으로 식별 및 분류합니다. 업그레이드 전에 팩 엔터티를 삭제한 경우 업그레이드 중에 팩이 식별되지 않습니다.

업그레이드 프로세스 중에 WFA는 데이터베이스의 팩을 스토리지 자동화 스토어에 릴리스된 팩 목록과 비교하여 업그레이드 전에 설치된 팩을 확인합니다. 따라서 팩 식별은 데이터베이스의 기존 팩을 분류합니다.

WFA는 팩을 식별하고 분류하기 위해 다음 프로세스를 수행합니다.

- Storage Automation Store에 릴리즈된 팩 목록을 유지 관리하여 업그레이드 전에 설치된 팩을 비교하고 확인합니다.
- Storage Automation Store가 활성화된 경우 팩의 엔터티를 Storage Automation Store 동기화의 일부로 분류합니다.
- 업데이트된 목록을 사용하여 요소를 팩으로 분류합니다.

팩 ID는 스토리지 자동화 스토어에서 다운로드한 NetApp 인증 팩에만 적용됩니다.

업그레이드 중에 팩이 식별되지 않으면 팩을 다시 가져와 WFA에서 식별할 수 있습니다. wfa.log 파일은 업그레이드 중에 팩으로 식별되지 않은 엔터티에 대한 세부 정보를 제공합니다.

## **WFA 워크플로우 팩을 SCM 리포지토리에 통합합니다**

WFA(OnCommand Workflow Automation) 팩을 SCM(소스 제어 관리) 저장소와 통합할 수 있습니다.

관리자 또는 설계자 자격 증명이 있어야 합니다.

GitHub, Perforce 및 SVN과 같은 SCM 툴을 사용하면 로컬 디렉토리를 매핑하여 SCM 리포지토리 서버에서 코드를 체크아웃할 수 있습니다. 이 로컬 디렉토리 매핑을 \_SCM 클라이언트 위치라고 합니다. \_ 파일 시스템 위치를 클라이언트 영역으로 사용하여 SCM 클라이언트를 설정해야 합니다.

WFA 서버 시스템에서 SCM 클라이언트를 설정할 수 있으며 SCM 작업을 위해서는 WFA 서버 시스템에 액세스할 수 있어야 합니다.

### **SCM에 새 워크플로우 팩을 체크 인합니다**

WFA(OnCommand Workflow Automation)를 사용하여 새 워크플로우 팩을 생성한 다음 SCM(소스 제어 관리)에서 확인할 수 있습니다.

필요한 것

SCM을 설정해야 하며 관리자 또는 설계자 자격 증명이 있어야 합니다.

단계

1. 웹 브라우저를 통해 \* WFA \* 창에 로그인합니다.
2. 새 워크플로우 팩을 만듭니다.

### **워크플로우 자동화 팩을 만듭니다**

3. 작성한 팩에 요소를 추가합니다.

### **OnCommand Workflow Automation 팩에 요소를 추가합니다**

4. 서버로 내보내기 \* 아이콘을 클릭합니다.
5. 서버 폴더로 내보내기 대화 상자의 서버 시스템\*의 폴더 위치 필드에 SCM 클라이언트가 포함된 서버에 팩을 저장할 파일 시스템 위치를 입력합니다.

팩이나 내용을 편집하거나 다시 내보내려면 \* Unlock \* (잠금 해제 \*) 아이콘을 클릭합니다.

6. SCM 클라이언트 위치에서 SCM 서버로 pack 내용을 체크 인한다.

### **WFA 워크플로우 팩의 새 버전을 확인합니다**

WFA(OnCommand Workflow Automation)에서 팩 버전을 업데이트한 다음 업데이트된 팩을 SCM(소스 제어 관리) 서버의 새 위치로 확인할 수 있습니다.

필요한 것

SCM을 설정해야 하며 관리자 또는 설계자 자격 증명이 있어야 합니다.

#### 단계

1. 웹 브라우저를 통해 \* WFA \* 창에 로그인합니다.
2. 콘텐츠 관리 \* > \* 팩 \* 을 클릭합니다.
3. Edit Pack \* 아이콘을 클릭합니다.
4. Pack <pack name>\* 대화 상자의 \* 버전 \* 필드에서 팩 버전을 업데이트합니다.
5. 저장 \* 을 클릭합니다.
6. 팩 수준에서 \* 서버로 내보내기 \* 아이콘을 클릭합니다.
7. Export to Server Folder \* (서버 폴더로 내보내기 \*) 대화 상자의 \* Folder location at server system \* (서버 시스템의 \* 폴더 위치) 필드에 새 파일 시스템 위치를 입력합니다.

이전에 'C:\p4\cDOT\1.0.0' 파일 시스템 위치에 저장했다면 이제 'C:\p4\cDOT\2.0.0' 위치에 저장하십시오.

8. SCM 클라이언트 위치에서 SCM 서버의 새 위치에 팩 콘텐츠를 체크 인합니다.

SCM 서버의 '//depot/WFA/pack/cDOT/1.0.0' 경로에 팩이 이전에 저장된 경우 //depot/WFA/pack/cDOT/2.0.0' 등의 다른 위치에 팩을 저장할 수 있습니다

#### SCM 서버에서 WFA 워크플로우 팩을 업데이트합니다

SCM(소스 제어 관리) 서버에서 팩을 업데이트한 다음 업데이트된 팩을 WFA(OnCommand Workflow Automation)로 가져올 수 있습니다.

#### 필요한 것

SCM을 설정해야 하며 관리자 또는 설계자 자격 증명이 있어야 합니다

#### 이 작업에 대해

SCM 서버의 팩을 변경하거나 업데이트하는 경우 관리자 또는 설계자는 SCM에서 제공하는 diff 도구를 사용하여 충돌을 해결해야 합니다. WFA는 관련 변경 사항만 표시하도록 XML diff 파일을 조정합니다.

팩을 가져오기 전에 WFA 팩 콘텐츠의 변경 사항에 대한 알림을 받게 됩니다.

#### 단계

1. 웹 브라우저를 통해 WFA 창에 로그인합니다.
2. 업데이트된 팩을 WFA로 가져옵니다.

#### "WFA 워크플로우 팩을 가져옵니다"



WFA 데이터베이스에 이미 동일한 팩이 포함되어 있는 경우 팩 콘텐츠를 덮어씁니다.

#### 기존 WFA 워크플로우 팩을 SCM 서버에 체크 인합니다

WFA(OnCommand Workflow Automation)에서 기존 팩을 SCM(소스 제어 관리) 서버에 체크 인할 수 있습니다.

## 필요한 것

SCM을 설정해야 하며 관리자 또는 설계자 자격 증명이 있어야 합니다.

## 단계

1. 웹 브라우저를 통해 \* WFA \* 창에 로그인합니다.
2. 콘텐츠 관리 \* > \* 팩 \* 을 클릭합니다.
3. 서버로 내보내기 \* 아이콘을 클릭합니다.
4. 서버 폴더로 내보내기 \* 대화 상자의 \* 서버 시스템 \* 의 폴더 위치 필드에 서버 팩이 저장된 서버 폴더 위치를 입력합니다.

그리면 SCM 클라이언트가 생성되는 파일 시스템에서 팩이 분해된 형태로 내보내집니다.

5. SCM 클라이언트 위치에서 SCM 서버로 pack 내용을 체크 인합니다.
6. SCM에서 제공하는 diff 도구를 사용하여 팩의 SCM 버전과 비교하여 변경 사항을 확인합니다.

## 엔터티로부터 WFA 워크플로우 팩을 제거합니다

WFA(OnCommand Workflow Automation)에 있는 엔터티로부터 팩을 제거하고 업데이트된 팩을 SCM(소스 제어 관리) 서버로 체크 인할 수 있습니다.

## 필요한 것

SCM을 설정해야 하며 관리자 또는 설계자 자격 증명이 있어야 합니다.

## 이 작업에 대해

다음 요소에서 팩을 제거할 수 있습니다.

- 워크플로우
- 파인더
- 필터
- 명령
- 기능
- 템플릿
- 구성표
- 사전
- 데이터 소스 형식
- 원격 시스템 유형
- 캐시 쿼리
- 범주

## 단계

1. 웹 브라우저를 통해 \* WFA \* 창에 로그인합니다.
2. Workflow 디자인 포털에서 \* 제거할 엔터티를 찾아 \* <Entity> \* 를 클릭합니다.

3. Pack \*에서 제거 아이콘을 클릭합니다.
  4. [팩에서 제거] <엔터티> \* 대화 상자에서 해당 엔터티로부터 삭제할 팩을 선택합니다.
  5. 확인 \*을 클릭합니다.
  6. Packs \* 탭을 클릭합니다.
  7. 서버로 내보내기 \* 아이콘을 클릭합니다.
  8. 서버 폴더로 내보내기 \* 대화 상자의 \* 서버 시스템 \*의 폴더 위치 필드에 서버 팩이 저장된 서버 폴더 위치를 입력합니다.
- 그리면 SCM 클라이언트가 생성되는 파일 시스템에서 팩이 분해된 형태로 내보내집니다.
9. SCM 클라이언트 위치에서 SCM 서버로 pack 내용을 체크 인합니다.
  10. SCM에서 제공하는 diff 도구를 사용하여 팩의 SCM 버전과 비교하여 변경 사항을 확인합니다.

#### SCM에서 WFA 워크플로우 팩을 이전 버전으로 롤백합니다

SCM(소스 제어 관리)에서 팩을 이전 버전으로 롤백하고 WFA(OnCommand Workflow Automation)로 가져올 수 있습니다.

##### 필요한 것

SCM을 설정해야 하며 관리자 또는 설계자 자격 증명이 있어야 합니다.

##### 단계

1. SCM 클라이언트 위치에서 SCM 툴을 사용하여 파일 시스템 위치의 이전 버전으로 팩을 롤백합니다.  
SCM 클라이언트는 관심 있는 정확한 변경 번호와 동기화됩니다.
2. 웹 브라우저를 통해 WFA 창에 로그인합니다.
3. 업데이트된 팩을 WFA로 가져옵니다.

#### "WFA 워크플로우 팩을 가져옵니다"



WFA 데이터베이스가 이전 버전으로 롤백됩니다.

## 워크플로우를 위한 구성 요소 생성

WFA(Workflow Automation)에는 다양한 구성 요소가 포함되어 있으며, 워크플로우 구성에 사용됩니다. 워크플로우에 필요한 WFA 구성 요소를 생성할 수 있습니다.

##### 데이터 원본 형식을 만듭니다

WFA(OnCommand Workflow Automation)에서 미리 정의되어 있지 않은 데이터 소스에서 데이터를 수집할 수 있도록 데이터 소스 유형을 생성해야 합니다.

##### 필요한 것

- WFA에 사전 정의되어 있지 않은 사용자 지정 데이터 소스 유형을 생성하는 경우 필수 사전 항목 및 구성표를 만들어야 합니다.
- 스크립트 방법을 사용하는 데이터 소스 유형을 생성하려면 PowerShell 스크립팅에 대해 알고 있어야 합니다.

#### 단계

1. 데이터 소스 디자인 \* > \* 데이터 소스 유형 \* 을 클릭합니다.
2. 을 클릭합니다 를 클릭합니다.
3. 새 데이터 원본 유형 \* 대화 상자에서 \* 데이터 원본 \*, \* 데이터 원본 버전 \* 및 \* 구성표 \* 필드에 필요한 세부 정보를 입력하거나 선택합니다.
4. 기본 포트 \* 필드에 포트 번호를 입력합니다.

2638

입력한 포트 번호는 데이터 획득을 위해 이 데이터 소스 유형을 추가할 때 채워집니다. 기본적으로 이 포트는 WFA에서 데이터 소스와 통신하기 위해 사용되며 포트가 데이터 소스 서버에서 열려 있어야 합니다.

5. Method \* 목록에서 WFA가 데이터를 획득하는 데 사용해야 하는 방법을 선택합니다.

다음을 선택한 경우...	그러면...
SQL	<p>드라이버 유형* 목록에서 데이터 원본에 적합한 다음 드라이버 중 하나를 선택합니다.</p> <ul style="list-style-type: none"> <li>• * MySQL 커넥터/J *</li> <li>• * MySQL Server JDBC 드라이버 3.0*</li> <li>• * Oracle JDBC 드라이버 11.2.0.3 *</li> </ul>
스크립트	<p>Script * 필드에 데이터 소스에서 데이터를 연결하고 검색하는 데 사용되는 PowerShell 스크립트를 입력합니다.</p> <p> 사전 항목 등가 CSV 파일의 데이터는 탭을 필드 구분 기호로 포함해야 합니다. 예를 들어 VMware vCenter 데이터 소스 유형에 대한 PowerShell 스크립트를 참조하십시오.</p>

6. 저장 \* 을 클릭합니다.

#### 명령을 생성합니다

작업에 적합한 사전 정의된 WFA 명령이 없는 경우 WFA 명령을 생성하여 워크플로우에서 특정 작업을 완료할 수 있습니다.

#### 필요한 것

PowerShell 또는 Perl을 사용하여 명령에 필요한 코드를 작성하는 방법을 알아야 합니다.

## 단계

1. 워크플로 디자인 \* > \* 명령 \* 을 클릭합니다.
2. 을 클릭합니다  를 클릭합니다.
3. 새 명령 정의 \* 대화 상자의 \* 속성 \* 탭에서 \* 이름 \*, \* 설명 \* 및 \* 시간 초과 \* 필드에 필요한 세부 정보를 입력하거나 선택합니다.
  - a. 문자열 표현 \* 필드에 MVEL 구문을 사용하여 명령의 문자열 표현을 입력합니다.

"VolumeName"⇒"SnapshotName"

명령의 문자열 표현은 계획 및 실행 중에 워크플로 디자인에 표시할 정보를 표시하는 데 사용됩니다. 명령의 문자열 표현에는 명령의 매개 변수만 사용해야 합니다.

- a. \* 선택 사항: \* 대기 명령을 작성하는 경우 \* 명령 유형 \* 섹션에서 \* 조건 대기 \* 를 선택하고 \* 대기 간격 \* 필드에서 필수 값을 설정합니다.
4. Code \* (코드 \*) 탭에서 다음 단계를 수행합니다.

- a. 스크립트 언어 \* 목록에서 명령에 필요한 스크립트 언어를 선택합니다.

를 클릭하고 명령에 사용할 언어를 추가로 선택할 수 있습니다.

- b. 선택한 언어 탭에 명령에 대한 적절한 코드를 입력합니다.

PowerShell 스크립트에서 사용자 입력에 암호 유형을 사용하려면 매개 변수의 별칭을 만들고 속성에 '\_Password'를 포함해야 합니다. Perl 스크립트의 경우 \* Parameters Definition \* 탭에서 유형을 \* Password \* 로 지정할 수 있습니다.



명령 편집기에서는 "<", "&" 및 XML 태그 "<>"와 같은 특수 문자가 지원되지 않습니다.

```
param (
    [parameter(Mandatory=$false, HelpMessage="Specify an AD administrator password.")]
    [Alias ("ADAdminPassword_Password")]    [string] $ADAdminPassword
)
```

5. 매개 변수 정의 \* 탭에서 다음 단계를 수행하십시오.

- a. 매개변수 정의 테이블을 채우려면 \* 매개변수 검색 \* 을 클릭합니다.

매개 변수 및 해당 특성은 코드에서 추출되어 테이블에 표시됩니다. 예를 들어, Array 및 VolumeName 매개 변수는 다음 코드에서 추출됩니다.

```

param (
    [parameter(Mandatory=$true, HelpMessage="Array name or IP address")]
    [string]$Array,
    [parameter(Mandatory=$true, HelpMessage="Volume name")]
    [string]$VolumeName,
)

```

a. 매개 변수의 설명 열을 클릭하여 설명을 편집합니다.

이 탭의 다른 필드는 편집할 수 없습니다.

6. 매개 변수 맵핑 \* 탭에서 각 매개 변수에 대해 다음 단계를 수행합니다.

a. Type \* 열에서 적절한 사전 개체를 선택합니다.

b. Attribute \* 열의 목록에서 사전 개체에 적절한 속성을 입력하거나 선택합니다.

특성을 입력한 후 마침표(.)를 입력하고 해당 개체의 다른 특성을 포함할 수 있습니다.

type 으로 cm\_storage.volume 을 입력하고, gregateName 매개 변수의 attribute 로 aggregate.name 를 입력합니다.

c. 개체 이름 \* 열에 개체 이름을 입력합니다.

개체 이름은 워크플로에서 명령 세부 정보를 지정할 때 <command> 매개 변수 대화 상자의 탭에서 매개 변수를 그룹화하는 데 사용됩니다.

워크플로우에서 명령 세부 정보를 지정할 때 매핑되지 않은 매개 변수는 <command> 매개 변수 대화 상자의 \* 기타 매개 변수 \* 탭에 표시됩니다.

7. 예약 \* 탭에서 SQL 쿼리를 사용하여 예약 스크립트를 입력하여 예약된 워크플로우 실행 중에 명령에 필요한 리소스를 예약합니다.

a. \* 선택 사항: \* 예약 표시 \* 필드에 MVEL 구문을 사용하여 예약에 대한 문자열 표현을 입력합니다.

"""SnapMirror 레이블에 규칙 추가"

"+ SnapMirrorLabel+"[.code]"를 SnapMirror 정책에 적용합니다

"+ PolicyName"""

문자열 표현은 예약 창에 예약된 리소스의 세부 정보를 표시하는 데 사용됩니다.



예약 스크립트는 cm\_storage, cm\_performance, storage, performance, vc, 사용자 정의 스키마를 선택합니다.

8. \* 선택 사항: \* \* 검증 \* 탭에서 SQL 쿼리를 입력하여 명령이 데이터 소스와 WFA 캐시에 예상대로 영향을 미치는지 여부를 확인하고, 예약 사항을 제거할 수 있도록 합니다.

입력하는 SQL 쿼리는 SQL SELECT 문으로만 구성할 수 있습니다.

- a. 검증 스크립트를 테스트하려면 \* 테스트 검증 \* 을 클릭하십시오.
  - b. Verification \* (검증 \*) 대화 상자에서 필요한 테스트 매개변수를 입력합니다.
  - c. 예약 데이터를 사용하여 검증 스크립트를 테스트하지 않으려면 \* test \* 필드에 \* 예약 데이터 사용 필드의 선택을 취소합니다.
  - d. Test \* 를 클릭합니다.
  - e. 테스트 결과를 검토한 후 대화 상자를 닫습니다.
9. Test \* 를 클릭하여 명령을 테스트합니다.
10. 테스트 명령 <명령 이름> \* 대화 상자에서 \* 테스트 \* 를 클릭합니다.

테스트 결과는 대화 상자의 로그 메시지 섹션에 표시됩니다.

11. 저장 \* 을 클릭합니다.

명령에 대한 예약 스크립트를 테스트합니다

WFA(OnCommand Workflow Automation) 명령에 대해 작성한 예약 스크립트를 운동장 데이터베이스에서 테스트하여 스크립트가 제대로 작동하고 WFA 데이터베이스 테이블에 영향을 미치지 않는지 확인할 수 있습니다.

이 작업에 대해

기본 WFA 설치 경로가 이 절차에 사용됩니다. 설치 중에 기본 위치를 변경한 경우 변경된 WFA 설치 경로를 사용해야 합니다.

단계

1. WFA 서버에서 명령 프롬프트를 열고 디렉터리를 'c:\Program Files\NetApp\WFA\MySQL\bin'으로 변경합니다
2. mysqldump -u WFA -pWfa123—single-transaction—skip-add-drop-table database\_table > dump\_location 명령을 사용하여 WFA 데이터베이스의 덤프를 생성합니다

CM\_STORAGE 데이터베이스 테이블의 덤프를 생성하는 명령: mysqldump -u WFA -pWfa123—single-transaction—skip-add-drop -table cm\_storage > c:\tmp\cmSt2.sql"

3. mysql-u dWFA-pwfa123 fground <dump\_location' 명령을 사용하여 생성한 덤프를 WFA 운동장 데이터베이스에 복원합니다

mysql -u WFA -pWfa123 놀이터 <c:\tmp\cmSt2.sql"

4. WFA 명령을 생성 또는 편집하고 \* Reservation \* 탭에 예약 스크립트를 작성합니다.

예약 및 검증 스크립트에서 놀이터 데이터베이스만 사용하는지 확인해야 합니다.

5. 워크플로를 만들거나 편집하고, 워크플로에 명령을 추가한 다음 워크플로를 실행합니다.
6. 예약 및 검증 스크립트가 예상대로 작동하는지 확인합니다.

WFA 데이터 소스 수집 프로세스에서는 놀이터 데이터베이스가 업데이트되지 않습니다. 수동으로 명령에 의해 생성된 예약을 제거해야 합니다.

## 파인더를 생성합니다

필요한 리소스를 검색하는 데 적합한 사전 정의된 WFA 파인더가 없는 경우 리소스를 검색할 수 있는 WFA 파인더를 생성할 수 있습니다.

### 필요한 것

파인더에 사용되는 필수 필터를 만들어야 합니다.

### 단계

1. Workflow Design \* > \* Finders \* 를 클릭합니다.
2. 을 클릭합니다  를 클릭합니다.
3. 새 파인더 \* 대화 상자의 \* 속성 \* 탭에서 \* 이름 \*, \* 유형 \* 및 \* 설명 \* 필드에 필요한 정보를 입력하거나 선택합니다.
4. Filters \* (필터 \*) 탭의 \* Available Filters \* (사용 가능한 필터 \*) 목록에서 필요한 필터를 선택하고 를 클릭합니다  
.

요구 사항에 따라 필터를 추가하거나 제거할 수 있습니다.

5. 반환된 특성 탭의 \* 사용 가능 \* 목록에서 필터에 필요한 특성을 선택하고 을 클릭합니다 .
6. \* 선택 사항: \* 검색 기능을 테스트하려면 \* 테스트 \* 를 클릭하십시오.
  - a. Test Finder <FinderName>\* 대화 상자에서 필요한 테스트 매개 변수를 입력합니다.
  - b. 파인더를 테스트하는 데 예약 데이터를 사용하지 않으려면 \* Use reservation data in test \* 확인란의 선택을 취소합니다.
  - c. Test \* 를 클릭합니다.
7. 저장 \* 을 클릭합니다.

## 필터를 만듭니다

작업에 적합한 사전 정의된 WFA 필터가 없는 경우, 리소스를 검색할 수 있는 WFA 필터를 생성할 수 있습니다.

### 필요한 것

필터를 만들려면 적절한 SQL 구문을 알아야 합니다.

### 단계

1. Workflow Design \* > \* Filters \* 를 클릭합니다.
2. 을 클릭합니다  를 클릭합니다.
3. 새 필터 \* 대화 상자의 \* 속성 \* 탭에서 \* 이름 \*, \* 사전 유형 \* 및 \* 설명 \* 필드에 필요한 정보를 입력하거나 선택합니다.
4. 쿼리 \* 탭에서 필터에 적절한 SQL 쿼리를 입력합니다.

단일 SQL 쿼리를 입력하고 선택적으로 입력 매개 변수를 사용해야 합니다. 입력 매개 변수를 사용하려면 '\${ParameterName}' 구문을 사용해야 합니다.

```
SELECT
    array.ip
FROM
    storage.array
WHERE
    array.name = '${ArrayName}'
```

5. Refresh \* 를 클릭하여 \* Input Parameters \* 테이블과 \* returned Attributes \* 목록을 채웁니다.

이 정보는 입력한 SQL 쿼리에서 가져옵니다. 예를 들어, 이전 단계에서 SQL 쿼리 예제를 사용하면 반환된 속성에 IP가 표시되고 ArrayName 이 입력 매개 변수에 표시됩니다. Label \* 및 \* Description \* 열에서 항목을 편집할 수 있습니다.

6. \* 선택 사항: \* 테스트 \* 를 클릭하여 필터를 테스트합니다.

- a. Test Filter <FilterName>\* 대화 상자에서 필요한 테스트 매개 변수를 입력합니다.
- b. 필터를 테스트하기 위해 예약 데이터를 사용하지 않으려면 \* Use reservation data in test \* 확인란의 선택을 취소합니다.
- c. Test \* 를 클릭합니다.

테스트 결과가 표시됩니다.

- d. 대화 상자를 닫습니다.

7. 저장 \* 을 클릭합니다.

## 사전 항목을 만듭니다

스토리지 환경에서 새 개체 유형과 해당 관계를 정의하려는 경우 WFA 사전 항목을 생성할 수 있습니다.

### 단계

1. 데이터 소스 디자인 \* > \* 사전 \* 을 클릭합니다.
2. 을 클릭합니다  를 클릭합니다.
3. 새 사전 항목 \* 대화 상자의 \* 개체 유형 \* 이름 및 \* 설명 \* 필드에 필수 세부 정보를 입력합니다.
4. Scheme \* 필드에 대해 다음 작업 중 하나를 수행합니다.
  - 목록에서 사용 가능한 구성표 중 하나를 선택합니다.
  - 새 구성표 추가 \* 를 클릭하고 새 구성표 대화 상자에 필수 \* 구성표 이름 \* 을 입력한 다음 \* 추가 \* 를 클릭합니다.
5. 행 추가 \* 를 클릭하고 다음 단계를 수행하여 속성을 설명합니다.
  - a. 이름 \* 열을 클릭하고 속성 이름을 입력합니다.
  - b. Type \* 열에서 필요한 유형을 선택합니다.

String Length \* 열은 형식으로 문자열을 선택한 경우 채워지고 편집할 수 있습니다. 또한 \* enum \* 을 유형으로 선택한 경우 \* Values \* 열을 편집할 수 있습니다.

c. Natural Key, to be Cached, and Can be Null columns에서 해당 특성의 확인란을 선택합니다.

Natural Key \* 확인란을 선택한 경우에는 \* Null일 수 있음 \* 확인란을 선택할 수 없습니다.

d. 사전 개체에 필요한 특성을 추가합니다.

e. \* 선택 사항: \* 자연 키 열에 있는 \* 값은 대/소문자를 구분하도록 하려면 \* 확인란을 선택합니다.

6. 저장 \* 을 클릭합니다.

## 함수를 만듭니다

작업에 적합한 사전 정의된 WFA 기능이 없는 경우, WFA 기능을 유ти리티로 사용할 수 있습니다.

### 필요한 것

함수를 만들려면 MVFLEX Expression Language(MVEL) 구문을 알아야 합니다.

### 이 작업에 대해

함수 정의에 대해 다음을 포함해야 합니다.

- 이름: 함수의 이름입니다

MVEL 구문에서 예약된 단어를 사용해서는 안 됩니다. 각 함수에는 고유한 이름이 있어야 합니다.

- MVEL 정의: 함수 정의의 MVEL 구문을 지정하는 문자열입니다

### 단계

1. 워크플로 디자인 \* > \* 함수 \* 를 클릭합니다.
2. 을 클릭합니다 를 클릭합니다.
3. 새 기능 \* 대화 상자에서 \* 함수 설명 \* 및 \* 함수 정의 \* 필드에 필요한 세부 정보를 입력하거나 선택합니다.

```
def actualVolumeSize(data_size, snap_pct)
{
    if (snap_pct < 0 ) {
        snap_pct = 0;
    } else if (snap_pct > 99) {
        snap_pct = 99;
    }

    div = 1 - (snap_pct/100);
    return (int)(data_size/div);
}
```

함수 이름 \* 필드는 MVEL 구문에 사용되는 데이터로 채워집니다.

4. \* 선택 사항: \* 기능을 테스트하려면 \* Test \* 를 클릭합니다.
  - a. Test \* 대화 상자의 \* Expression \* 섹션에서 필요한 함수 식을 입력합니다.

실제 볼륨 크기(600, 1)
  - b. Test \* 를 클릭합니다.

테스트 결과가 표시됩니다.

  - c. 대화 상자를 닫습니다.
5. 저장 \* 을 클릭합니다.

## 템플릿을 만듭니다

명령 세부 정보에 속성을 채우기 위한 청사진으로 사용할 수 있는 템플릿을 만들 수 있습니다.

### 단계

1. Workflow Design \* > \* Templates \* 를 클릭합니다.
2. 을 클릭합니다  를 클릭합니다.
3. 새 템플릿 \* 대화 상자에서 \* 이름 \*, \* 유형 \* 및 \* 설명 \* 필드에 필요한 세부 정보를 입력하거나 선택합니다.

특성 테이블은 \* Type \* 필드에서 선택한 사전 개체를 기반으로 채워집니다.
4. 각 속성의 값 열을 클릭하고 다음 중 하나를 수행합니다.
  - 목록에서 필요한 값을 입력하거나 선택합니다.
  - 사용자 입력 항목(예: 사용자 입력 크기에 대한 '\$size')을 입력합니다.
5. 저장 \* 을 클릭합니다.

## 캐시 쿼리를 만듭니다

WFA 데이터베이스에서 데이터 소스 유형의 사전 개체에 대한 정보를 캐싱하려는 경우 캐시 쿼리를 정의할 수 있습니다. 캐시 쿼리를 만들어 사전 항목 및 Active IQ Unified Manager 6.1과 같은 하나 이상의 데이터 원본 형식과 연결할 수 있습니다.

### 필요한 것

캐시 쿼리를 만들려면 적절한 SQL 구문을 알아야 합니다.

### 단계

1. 데이터 원본 디자인 \* > \* 캐시 쿼리 \* 를 클릭합니다.
2. 을 클릭합니다  를 클릭합니다.
3. Add Cache Query \* (캐시 쿼리 추가 \*) 대화 상자에서 필요한 사전 항목과 데이터 원본 유형을 선택합니다.
4. "SQL SELECT QUERY" 섹션에 적절한 SQL 쿼리를 입력합니다.

다음 SQL 쿼리는 Active IQ Unified Manager 6.1 데이터 소스 유형의 디스크 사전 개체에 대한 정보를 캐시합니다.

```

SELECT
    disk.objId AS id,
    disk.name AS NAME,
    disk.uid AS uid,
    disk.effectiveInterfaceType AS TYPE,
    disk.rpm AS rpm,
    disk.homeNodeId AS home_node_id,
    disk.ownerNodeId AS owner_node_id,
    disk.model AS model,
    disk.serialNumber AS serial_number,
    disk.totalBytes/1024/1024 AS size_mb,
    disk.shelf AS shelf,
    disk.shelfBay AS shelf_bay,
    disk.pool AS pool,
    disk.vendor AS vendor,
    LOWER(disk.raidPosition) AS raid_position,
    disk.containerTypeRaw AS container_type,
    disk.clusterId AS cluster_id
FROM
    netapp_model_view.disk disk

```

## 5. SQL 쿼리를 테스트하려면 \* 테스트 \* 를 클릭합니다.

둘 이상의 데이터 원본 유형을 선택한 경우 캐시 쿼리 테스트 대화 상자가 열리고 필요한 데이터 원본 유형을 선택할 수 있습니다.

테스트 결과가 표시됩니다.

## 6. 대화 상자를 닫습니다.

## 7. 저장 \* 을 클릭합니다.

반복 일정을 생성합니다

OnCommand Workflow Automation(WFA)에서는 워크플로우에 대한 두 가지 예약 옵션을 제공합니다. 워크플로우를 특정 시간에 한 번 실행하도록 예약하거나 반복되는 일정을 만들고 일정을 워크플로에 연결하여 정기적으로 워크플로우를 실행할 수 있습니다.

이 작업에 대해

사용자가 만든 일정을 다시 사용할 수 있으며 여러 워크플로와 연결할 수 있습니다.

단계

1. Execution \* > \* Schedules \* 를 클릭합니다.
2. 을 클릭합니다  를 클릭합니다.
3. 새 일정 \* 대화 상자에서 일정의 이름, 설명 및 빈도를 입력하거나 선택합니다.

주파수의 경우 24시간 형식으로 시간을 입력해야 합니다. WFA 서버 시간이 일정에 적용됩니다.

#### 4. 확인 \* 을 클릭합니다.

작업을 마친 후

- 반복 실행 \* 옵션을 사용하여 워크플로를 실행할 때 일정을 워크플로에 연결할 수 있습니다.
- \* Execution \* > \* Recurring Schedules \* 를 클릭하여 워크플로와 스케줄과의 연관성에 대한 세부 정보를 볼 수 있습니다.

워크플로가 예약되면 즉시 실행되도록 예약된 워크플로의 리소스 및 실행 계획이 완료됩니다. 그러나 되풀이 일정이 있는 워크플로의 리소스 및 실행 계획은 일정이 워크플로와 연결되어 있지 않고 예약된 시간에 수행됩니다.

#### 필터 규칙을 정의합니다

vFiler 유닛, 애그리게이트, 가상 머신 등의 사전 항목 리소스를 필터링하기 위한 규칙 집합을 정의할 수 있습니다. 기존 워크플로 및 새 워크플로를 만드는 동안 해당 워크플로에 대한 규칙을 사용자 지정할 수 있습니다.

단계

1. 웹 브라우저를 통해 admin으로 WFA에 로그인합니다.
2. 워크플로우 설계 \* > \* 워크플로우 \* 를 클릭합니다.
3. 워크플로 \* 창에서 수정할 워크플로를 두 번 클릭합니다.

Workflow <Workflow name>(작업 흐름 이름) 창이 표시됩니다.

4. 다음 옵션 중 하나를 선택하여 규칙 집합을 정의합니다.

원하는 작업	다음을 수행하십시오.
행의 명령이 반복되면 리소스를 검색합니다	<ol style="list-style-type: none"><li>a. 행 번호를 클릭하고 * 행 반복 * 을 선택합니다.</li><li>b. 행 반복 대화 상자의 * 반복 * 드롭다운 목록에서 * 그룹의 모든 리소스에 대해 * 옵션을 선택합니다.</li><li>c. 리소스 유형을 선택합니다.</li><li>d. 검색 조건 입력 * 링크를 클릭합니다.</li></ol>
명령 입력에 필요한 리소스를 검색합니다	<ol style="list-style-type: none"><li>a. 을 클릭합니다 .</li><li>b. command_name 매개 변수 대화 상자의 * 정의 &lt;사전 개체&gt; * 드롭다운 목록에서 기존 &lt;사전 개체&gt; * 옵션을 검색하여 * 를 선택합니다.</li><li>c. 검색 조건 입력 * 링크를 클릭합니다.</li></ol>

원하는 작업	다음을 수행하십시오.
명령 입력에서 변수가 참조하는 리소스를 검색합니다	<p>a. 을 클릭합니다 .</p> <p>b. command_name&gt; 매개 변수 대화 상자의 * 정의 &lt;사전 개체&gt; * 드롭다운 목록에서 * attributes * 를 입력하여 * 옵션을 선택합니다.</p> <p>c. 을 클릭합니다  으로 표시된 필드의 경우 .</p>
String 형식의 이름 명령입니다	<p>a. 을 클릭합니다 .</p> <p>b. command_name&gt; 매개 변수 대화 상자의 * 정의 &lt;사전 개체&gt; * 드롭다운 목록에서 * attributes * 를 입력하여 * 옵션을 선택합니다.</p> <p>c. 을 클릭합니다  문자열 필드의 경우.</p>

## 5. 리소스 선택 \* 대화 상자에서 \* 필터 규칙 정의 \* 확인란을 선택합니다.

자원 선택 대화 상자의 찾기 드롭다운에서 옵션 중 하나를 선택한 경우 필터 규칙 정의 확인란이 비활성화됩니다. 필터 정의 규칙을 사용하려면 finder 값을 ""없음""으로 설정해야 합니다.

## 6. 규칙의 속성, 연산자 및 값을 입력합니다.

값은 작은따옴표 안에 입력해야 합니다. 필터 규칙에는 하나 이상의 그룹이 포함될 수 있습니다.

## 7. 확인 \* 을 클릭합니다.

## 승인 지점을 추가합니다

워크플로에서 승인 지점을 체크포인트로 추가하여 워크플로 실행을 일시 중지하고 승인 결과에 따라 재개할 수 있습니다. 워크플로의 증분 실행을 위해 승인 지점을 사용할 수 있습니다. 이 경우 워크플로의 섹션은 특정 조건이 충족된 후에만 실행됩니다. 예를 들어 다음 섹션을 승인해야 하거나 첫 번째 섹션을 성공적으로 실행할 때 실행됩니다.

### 단계

1. 웹 브라우저를 통해 설계자 또는 관리자로 WFA에 로그인합니다.
2. 워크플로우 설계 \* > \* 워크플로우 \* 를 클릭합니다.
3. 워크플로 \* 창에서 수정할 워크플로를 두 번 클릭합니다.
4. Workflow <Workflow name>\* 창에서 를 클릭합니다  아이콘을 클릭하여 승인 지점을 추가할 단계 왼쪽에 있습니다.

하나 이상의 단계에 대한 승인 지점을 추가할 수 있습니다.

5. [새 승인 지점] \* 대화 상자에서 설명 및 조건 세부 정보를 제공합니다.
6. 확인 \* 을 클릭합니다.

# 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"><li>파스칼 케이스를 사용합니다.</li><li>밑줄을 사용하지 마십시오.</li><li>약어는 사용하지 마십시오.</li></ul>	'\$VolumeName' '\$AutoDeleteOptions' '\$Size'
스크립트 내부 변수: <ul style="list-style-type: none"><li>Camel 케이스를 사용합니다.</li><li>밑줄을 사용하지 마십시오.</li><li>약어는 사용하지 마십시오.</li></ul>	'\$newVolume' '"\$qtreeName" '\$time'
기능: <ul style="list-style-type: none"><li>파스칼 케이스를 사용합니다.</li><li>밑줄을 사용하지 마십시오.</li><li>약어는 사용하지 마십시오.</li></ul>	'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) 같습니다
변수 이름은 영문으로 되어 있어야 하며 스크립트의 기능과 관련이 있어야 합니다.	'\$a'가 아니라 '\$name'을(를) 사용합니다
각 변수의 데이터 형식을 명시적으로 선언합니다.	[string] 이름입니다 [int] 크기
특수 문자(! @#&%..) 및 공백.	없음

지침	예
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 ' <i>HelpMessage</i> ' 주석을 사용하여 모든 입력 변수에 주석을 달 수 있습니다.	<pre>[parameter(Mandatory=\$false,HelpMe ssage="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,HelpMe ssage="Volume state")] [ValidateSet("online","offline","r estricted")] [string]\$State</pre>
매개 변수가 용량 유형임을 나타내기 위해 ""_Capacity"로 끝나는 매개 변수에 별칭을 추가합니다.	""볼륨 작성" 명령은 다음과 같이 별칭을 사용합니다.
	<pre>[parameter(Mandatory=\$false,HelpMe ssage="Volume increment size in MB")] [Alias("AutosizeIncrementSize_Capa city")] [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:	'\$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.	'\$a'가 아니라 '\$name'을(를) 사용합니다
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"=&gt;\\$Name,     "Size=i"=&gt;\\$Size )</pre>
파일러를 변수 이름으로 사용하지 말고 대신 배열(Array)을 사용하십시오.	없음
Perl에는 열거 값에 대한 ' <i>ValidateSet</i> ' 주석이 포함되지 않습니다. 인수가 열거 값을 갖는 경우에는 명시적 "'if'" 문을 사용합니다.	<pre>if (defined\$SpaceGuarantee&amp;&amp;! (\$SpaceG uaranteeeq 'none'</pre>
	\$SpaceGuaranteeq'volume'
	\$SpaceGuaranteeq'file')) { die'Illegal SpaceGuarantee argument: \\".\$SpaceGuarantee.\\"; }
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>
모든 Perl WFA 명령은 다음 Perl 모듈을 사용해야 합니다.	<pre>use Getopt::Long; use NaServer; use WFAUtil;</pre>
<ul style="list-style-type: none"> <li>• getopt</li> </ul> <p>입력 매개 변수를 지정하는 데 사용됩니다.</p> <ul style="list-style-type: none"> <li>• WFAUtil</li> </ul> <p>이 기능은 명령 로깅, 명령 진행률 보고, 어레이 컨트롤러에 연결 등에 제공되는 유ти리티 기능에 사용됩니다.</p>	

## 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 블럭의 시작과 끝을 표시합니다.	PowerShell 스크립트 <pre>if (\$pair.length-ne 2) { throw "Got wrong input data" }</pre>
	Perl 스크립트 <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>

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

## Perl 설명

지침	예
한 줄 코멘트에 # 문자를 사용합니다.	# convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024;
줄 설명 끝에 # 문자를 사용합니다.	my \$MaxDirectorySizeBytes = \$MaxDirect orySize * 1024 * 1024; # convert to Bytes
시작 부분과 끝 부분에 빈 #이 있는 모든 줄의 #문자를 사용하여 여러 줄 메모에 대한 설명 테두리를 만듭니다.	# # 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 #

지침	예
WFA 명령에 주석 및 비활성 코드를 포함하지 마십시오. 그러나 테스트를 위해 POD(Plain Old Documentation) 메커니즘을 사용하여 코드를 주석 처리할 수 있습니다.	=begin comment # Set deduplication if(defined \$Deduplication && \$Deduplication eq "enabled") { \$wfaUtil->sendLog ("Enabling Deduplication"); } =end comment =cut

## 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.	Get-WFALogger -Info -message "Creating volume"
Data ONTAP, VMware 및 PowerCLI와 같은 내부 패키지와의 상호 작용이 필요한 모든 작업을 기록합니다. 모든 로그 메시지는 워크플로 실행 상태 기록의 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-&gt;new(); eval {     \$wfa_util-&gt;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
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-&gt;sendLog('WARN', "Removing volume: '.'.\$VolumeName); # Good Message</pre> <pre>\$wfa_util-&gt;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

<p><b>지침</b></p> <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"> <li>• ErrorAction 매개 변수는 cmdlet이 명령에서 발생하는 비종료 오류에 대응하는 방법을 결정합니다.</li> <li>• WarningAction 매개 변수는 cmdlet이 명령의 경고에 반응하는 방식을 결정합니다.</li> <li>• Stop, SilentlyContinue, Inquire 및 Continue 는 ErrorAction 및 WarningAction 매개 변수에 유효한 값입니다.</li> </ul> <p>자세한 내용은 PowerShell CLI에서 'Get-Help About_CommonParameters' 명령을 사용할 수 있습니다.</p>	<p><b>예</b></p> <p>ErrorAction: 다음 예제에서는 비종료 오류를 종료 오류로 처리하는 방법을 보여 줍니다.</p> <pre>New-NcIgroup-Name \$IgroupName-Protocol \$Protocol-Type\$OSType-ErrorActionstop</pre>
<p>들어오는 예외의 유형을 알 수 없는 경우 일반 ""try/catch"" 문을 사용합니다.</p>	<p><b>경고 조치</b></p> <pre>New-VM-Name \$VMName-VM \$SourceVM-DataStore\$DataStoreName-VMHost\$VMHost-WarningActionSilentlyContinue</pre>
<p>들어오는 예외의 유형이 알려진 경우 특정 ""try/catch"" 문을 사용합니다.</p>	<pre>try {     "In Try/catch block" } catch {     "Got exception" }</pre>
	<pre>try {     "In Try/catch block" } catch[System.Net.WebException], [System.IO.IOException] {     "Got exception" }</pre>

<p><b>지침</b></p> <p>리소스를 해제하려면 ""finally"" 문을 사용하십시오.</p>	<p><b>예</b></p> <pre>try {     "In Try/catch block" } catch {     "Got exception" } finally {     "Release resources" }</pre>
<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
Perl에는 try/catch 블록에 대한 기본 언어 지원이 포함되어 있지 않습니다. eval 블록을 사용하여 오류를 검사하고 처리합니다. 평가 블록을 가능한 한 작게 유지합니다.	<pre> eval {      \$wfa_util-&gt;sendLog('INFO',         "Quiescing the relationship :         \$DestinationCluster://\$Destination         Vserver         /\$DestinationVolume"     );     \$server-&gt;snapmirror_quiesce(         'destination-vserver' =&gt;         \$DestinationVserver,         'destination-volume'  =&gt;         \$DestinationVolume     );     \$wfa_util-&gt;sendLog('INFO',         'Quiesce operation         started successfully.'); };  \$wfa_util-&gt;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 파서를 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' 디렉토리에 추가된 모든 사용자 지정 Perl 모듈은 \_@INC\_Library에 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 툴킷에 사용할 수 있는 cmdlet이 없는 경우 Data ONTAP에서 'Invoke-SSH' 명령을 사용하여 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-&gt;sendLog('INFO',         "Connecting to the cluster: \$DestinationCluster"     );     my \$server         = \$wfa_util- &gt;connect(\$DestinationClusterIp, \$DestinationVserver);      my \$sm_info = \$server- &gt;snapmirror_get(         'destination-vserver' =&gt; \$DestinationVserver,         'destination-volume' =&gt; \$DestinationVolume     );      my \$sm_state = \$sm_info- &gt;{ 'attributes' }-&gt;{ 'snapmirror- info' }-&gt;{ 'mirror-state' };     my \$sm_status = \$sm_info- &gt;{ 'attributes' }-&gt;{ 'snapmirror- info' }-&gt;{ 'relationship-status' };      \$wfa_util-&gt;sendLog('INFO',         "SnapMirror relationship is \$sm_state (\$sm_status)");      if (\$sm_status ne 'quiesced')     {         \$wfa_util-&gt;sendLog('INFO',             'The status needs to be quiesced to resume transfer.');     } else {         my \$result = \$server- &gt;snapmirror_resume(             'destination-vserver' =&gt; \$DestinationVserver,             'destination-volume' =&gt; \$DestinationVolume         );         \$wfa_util-&gt;sendLog('INFO', 71 "Result of resume: \$result");     } } </pre>

지침	예
Data ONTAP API를 사용할 수 없는 경우 executeSystemCli 유ти리티 메서드를 사용하여 Data ONTAP CLI를 호출합니다.	없음

 executeSystemCli는 지원되지 않으며  
현재 7-Mode에서 작동하는 Data ONTAP에서만 사용할 수 있습니다.

## PowerShell 명령을 Perl으로 변환할 때의 고려 사항

PowerShell 및 Perl의 기능은 다르므로 PowerShell 명령을 Perl으로 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 -10이면 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 !~ /^[+-]?\d+$/) {
            die
                "Invalid value '$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>
<pre>Select-Object -First 1 ----</pre>	<pre>my \$result = \$server- &gt;job_get_iter(     'query' =&gt; { 'job-type' =&gt; 'VOL_CLONE_SPLIT' },     'desired-attributes' =&gt; {         'job-type' =&gt; '',         'job-description' =&gt; '',         'job-progress' =&gt; '',         'job-state' =&gt; ''     } ); my @jobarray; for my \$job (@{ \$result- &gt;{ 'attributes-list' } }) {     my \$description = \$job-&gt;{ '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.	<pre>SELECT     vserver.name FROM     cm_storage.vserver vserver</pre>
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

선택 쿼리에 의미 있는 이름이 있는 테이블 별칭을 사용합니다.

예

```
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
```

## 지침

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.

## 예

```
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
```

복잡한 쿼리에 주석을 사용합니다. 쿼리에서 지원되는 주석 스타일 중 일부는 다음과 같습니다.

- " - "를 끝까지 누릅니다

이 주석 스타일의 두 번째 하이픈 뒤에 공백이 있어야 합니다.

- """#" 문자부터 줄의 끝까지
- """/\* """에서 다음 """ \* /""" 시퀀스로

```
/*
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
```

## **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 ""캐시됨""을 true로 설정합니다. Active IQ 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 true로 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 사용용으로 예약되어 있습니다.	없음

- 관련 정보 \*

## 학습 자료에 대한 참고 자료

### 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]
문자열 표현의 모든 값에 의미 있는 레이블을 제공합니다.	'Volume name=VolumeName'

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 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.	사전 요구 사항이 포함된 샘플 워크플로우 'Clustered Data ONTAP 스토리지에 VMware NFS 데이터 저장소 생성'에 대한 설명을 참조하십시오.
워크플로가 생산 준비가 되어 있고 포털 페이지에 표시할 수 있는 경우에만 "생산 준비"를 "참"으로 설정합니다.	없음
기본적으로 "예약된 요소 고려"를 true로 설정합니다. 실행 워크플로우를 미리 볼 때 WFA 플래너는 캐시 데이터베이스의 기존 개체와 함께 예약된 모든 개체를 고려합니다. 이 옵션이 "참"으로 설정된 경우 특정 워크플로를 계획할 때 다른 예약된 워크플로 또는 병렬로 실행되는 워크플로의 효과를 고려합니다	<ul style="list-style-type: none"> <li>시나리오 1</li> <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> <li>시나리오 2</li> <p>워크플로 1은 '볼륨 만들기' 명령을 사용합니다. Aggregate에서 100GB 을 사용하는 예약된 워크플로 2가 있는 경우 계획 중에 워크플로 1은 워크플로 2에 대한 요구 사항을 고려해야 합니다.</p> </ul>
기본적으로 "요소 존재 유효성 검사 사용"은 "참"으로 설정됩니다	<ul style="list-style-type: none"> <li>시나리오 1</li> <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 사용해서는 안 됩니다. 볼륨 제거 효과는 볼륨 만들기 명령에 사용할 수 없게 되어 워크플로가 실패합니다.</p> <li>시나리오 2</li> <p>볼륨 만들기 명령은 특정 이름이 "vol198"인 워크플로에서 사용됩니다.</p> <p>이 옵션이 true로 설정된 경우 WFA 플래너는 계획 중에 해당 이름의 볼륨이 지정된 어레이에 있는지 확인합니다. 볼륨이 있는 경우 계획 중에 워크플로가 실패합니다.</p> </ul>
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
이름: <ul style="list-style-type: none"><li>• 이름을 ""\$" 문자로 시작합니다.</li><li>• 각 단어의 시작 부분에 대문자를 사용합니다.</li><li>• 모든 용어와 약어에 대문자를 사용합니다.</li><li>• 밑줄을 사용하지 마십시오.</li></ul>	'\$Array' '\$VolumeName'
표시 이름: <ul style="list-style-type: none"><li>• 각 단어의 시작 부분에 대문자를 사용합니다.</li><li>• 단어를 공백으로 구분합니다.</li><li>• 입력에 특정 단위가 있는 경우 표시 이름에 대괄호로 단위를 직접 지정합니다.</li></ul>	볼륨 이름 볼륨 크기(MB)
설명: <ul style="list-style-type: none"><li>• 각 사용자 입력에 대해 의미 있는 설명을 제공합니다.</li><li>• 필요한 경우 예를 제공합니다.</li></ul> 특히 사용자 입력이 특정 형식이어야 할 경우 이 작업을 수행해야 합니다.	"iGroup"에 추가할 이니시에이터입니다. 예를 들어 이니시에이터의 IQN 또는 WWPN입니다.
사용자 입력 설명은 워크플로 실행 중 사용자 입력에 대한 도구 설명으로 표시됩니다.	
Type: 특정 값 집합으로 입력을 제한하려면 Enum 을 형식으로 선택합니다.	프로토콜: iSCSI, FCP, 혼합
유형: 사용자가 WFA 캐시에 있는 값 중에서 선택할 수 있는 경우 유형으로 Query를 선택합니다.	\$Array: 다음과 같은 쿼리가 있는 쿼리 유형: <pre>SELECT     ip, name FROM     storage.array</pre>
Type: 사용자 입력이 쿼리에서 가져온 값으로 제한되거나 지원되는 Enum 형식으로만 제한되어야 하는 경우 사용자 입력을 locked로 표시합니다.	\$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.

지침	예
유형: 쿼리 유형 추가 열을 쿼리에 반환 값으로 추가하면 저장소 운영자가 사용자 입력을 올바르게 선택할 수 있습니다.	\$Aggregate: 이름, 총 크기 및 사용 가능한 크기를 제공하여 운영자가 애그리게이트를 선택하기 전에 속성을 알 수 있도록 합니다.
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.	샘플 워크플로우 'Create a clustered Data ONTAP Volume'에서 VserverName에 대한 쿼리는 다음과 같습니다.
	<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>
	이 쿼리는 \${clusterName}을(를) 참조합니다. 여기서 \$clusterName은 \$VserverName 사용자 입력 앞의 사용자 입력 이름입니다.
형식: Boolean 형식의 사용자 입력에 대해 값이 ""true, false""인 부울 형식을 사용합니다. 이렇게 하면 사용자 입력을 직접 사용하여 워크플로 디자인에 내부 식을 작성하는 데 도움이 됩니다. 예를 들어, \$UserInputName=="Yes' 대신 \$UserInputName을 사용할 수 있습니다.	'\$CreateCIFSShare': 유효한 값이 ""true"" 또는 ""false""인 부울 형식
type: 문자열 및 숫자 형식의 경우 특정 형식으로 값의 유효성을 검사하려는 경우 값 열에 정규식을 사용합니다.	위치별 사용자 입력은 ""[A-Z][A-Z]\-0[1-9]""로 나타낼 수 있습니다. 이 사용자 입력에는 ""US-01""""NB-02""가 허용되지만 nb-00'은 사용할 수 없습니다.
IP 주소 및 네트워크 마스크 입력에 정규식을 사용합니다.	
유형: 숫자 형식의 경우 값 열에 범위 기반 유효성 검사를 지정할 수 있습니다.	생성할 LUN 수의 경우 값 열의 항목은 1~20입니다.
그룹: 관련 사용자 입력을 적절한 버킷으로 그룹화하고 그룹 이름을 지정합니다.	모든 스토리지 관련 사용자 입력에 대한 "스토리지 세부 정보". 모든 VMware 관련 사용자 입력에 대한 "Datastore Details"

지침	예
필수: 워크플로우를 실행하기 위해 사용자 입력 값이 필요한 경우 사용자 입력을 필수로 표시합니다. 이렇게 하면 사용자 입력 화면이 사용자의 입력을 받아 입력하게 됩니다.	"Create NFS Volume" 워크플로우에서 ""\$VolumeName".
기본값: 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.	없음

#### 상수, 변수 및 반환 매개 변수

지침	예
상수: 여러 명령에 대한 매개 변수를 정의하기 위해 공통 값을 사용할 때 상수를 정의합니다.	_aggregate_over_averge_threshold_의 경우 SnapVault 샘플 워크플로우를 사용하여 LUN을 생성, 매플 및 보호합니다.
상수: 이름 <ul style="list-style-type: none"> <li>각 단어의 시작 부분에 대문자를 사용합니다.</li> <li>모든 용어와 약어에 대문자를 사용합니다.</li> <li>밑줄을 사용하지 마십시오.</li> <li>상수 이름의 모든 문자는 대문자를 사용합니다.</li> </ul>	_aggregate_used_space_threshold_입니다 _ActualVolumeSizeInMB_
변수: 명령 매개 변수 상자 중 하나에 정의된 개체에 이름을 제공합니다. 변수는 자동으로 생성된 이름이며 변경할 수 있습니다.	없음
변수: 이름은 변수 이름에 소문자를 사용합니다.	볼륨1 cifs_share 를 참조하십시오
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.	집계: 리소스 선택 논리를 사용하여 집계를 선택한 경우 실제 선택된 집계를 반환 매개 변수로 정의할 수 있습니다.

#### 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 컨텐츠가 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.

## 예약된 단어

OnCommand Workflow Automation(WFA)에는 예약된 단어가 몇 개 포함되어 있습니다. 변수 이름, 사용자 입력, 상수 및 반환 매개 변수와 같은 특성이나 매개 변수에 대해서는 워크플로에서 예약된 단어를 사용하지 않아야 합니다.

다음은 WFA에 예약된 단어 목록입니다.

• 및	• 부동	• 프로를 참조하십시오
• 스토리지	• 부동	• 돌아가기
• 어설션	• 용	• 런타임
• 부울	• 용서하십시오	• 보안 관리자
• 부울	• 기능	• 짧게
• 바이트	• If(경우	• 짧게
• 바이트	• 가져오기	• 사운드처럼
• 문자	• IMPORT_STATIC	• 스트릭tath
• 문자	• 인치	• 문자열
• 차트시퀀스	• instanceof(인스턴스)	• StringBuffer 를 클릭합니다
• 클래스	• 내부	• StringBuilder 를 선택합니다
• 클래스 로더	• 정수	• Strsim
• 컴파일러	• 있습니다	• 스위치
• 포함	• isdef	• 시스템
• 변환 가능 _	• 길게	• 나사산
• DEF	• 길게	• ThreadLocal 을 선택합니다
• 해야 합니다	• 수학	• 참
• 더블	• 신규	• 그때까지
• 더블	• null입니다	• VAR
• 기타	• 번호	• 무효입니다
• 비어 있습니다	• 오브젝트	• 있습니다
• 거짓	• 또는	• 와 함께

## REST API 사용 방법

WFA(Workflow Automation)에서 제공하는 REST API를 사용하여 외부 포털 및 데이터 센터

오케스트레이션 소프트웨어에서 워크플로우를 호출할 수 있습니다. WFA는 모든 REST API에 대해 XML 및 JSON 콘텐츠 유형을 지원합니다.

WFA를 사용하면 외부 서비스에서 워크플로우, 사용자, 필터, 파인더와 같은 다양한 리소스 컬렉션을 액세스할 수 있으며 URI 경로를 통해 외부 서비스는 GET, PUT, POST, DELETE 등의 HTTP 메소드를 사용할 수 있습니다. 이러한 URI로 리소스에 대해 CRUD 작업을 수행합니다.

WFA REST API를 통해 다음을 비롯한 몇 가지 작업을 수행할 수 있습니다.

- 워크플로우 정의 및 메타데이터에 액세스
- 워크플로우 실행 및 실행 상태 모니터링
- 사용자 및 역할을 보고 암호를 변경합니다.
- 리소스 선택 필터를 실행하고 테스트합니다.
- 리소스 파인더 실행 및 테스트
- 스토리지 또는 기타 데이터 센터 오브젝트의 자격 증명 관리
- 데이터 원본 및 데이터 원본 형식을 봅니다.

\_Rest 설명서\_REST API에 대한 자세한 내용:

[https://WFA\\_server\\_IP: port/rest/docs](https://WFA_server_IP: port/rest/docs)

WFA\_SERVER\_IP는 WFA 서버의 IP 주소이며, port는 설치 중에 WFA 서버에 사용된 TCP 포트 번호입니다.



WFA는 웹 UI에서 수신되는 요청에 대해 CSRF(사이트 간 요청 위조) 토큰을 확인합니다. 하지만 WFA는 REST 클라이언트나 오케스트레이션 소프트웨어에서 수신되는 REST 요청에 대해 CSRF 토큰을 확인하지 않습니다.

## 학습 자료에 대한 참고 자료

고급 WFA(Workflow Automation) 워크플로우를 생성하기 위한 특정 스크립팅 및 프로그래밍 방식에 대해 알고 있어야 합니다. 참조 자료를 사용하여 WFA 구성 요소 또는 워크플로우를 생성하기 전에 필요한 옵션에 대해 알아볼 수 있습니다.

### Windows PowerShell

WFA에서는 워크플로우 작업에 PowerShell 스크립트를 사용합니다. 다음 표에는 PowerShell에 대한 학습 자료에 대한 참조가 포함되어 있습니다.

Windows PowerShell 시작하기	<a href="http://msdn.microsoft.com/en-us/library/windows/desktop/aa973757(v=vs.85).aspx">http://msdn.microsoft.com/en-us/library/windows/desktop/aa973757(v=vs.85).aspx</a>
PowerShell 개발 — ISE(Integrated Scripting Environment)	<a href="https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/ise/introducing-the-windows-powershell-ise?view=powershell-7.2">https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/ise/introducing-the-windows-powershell-ise?view=powershell-7.2</a>

_.NET 프레임워크 이름 지침	<a href="http://msdn.microsoft.com/en-us/library/xzf533w0%28v=vs.71%29.aspx">http://msdn.microsoft.com/en-us/library/xzf533w0%28v=vs.71%29.aspx</a>
PowerShell 코드 스타일	<a href="http://get-powershell.com/post/2011/04/13/Extra-Points-for-Style-when-writing-PowerShell-Code.aspx">http://get-powershell.com/post/2011/04/13/Extra-Points-for-Style-when-writing-PowerShell-Code.aspx</a>
PowerShell 시험 사용/캐치	<a href="http://technet.microsoft.com/en-us/library/dd315350.aspx">http://technet.microsoft.com/en-us/library/dd315350.aspx</a>
PowerShell 자동 변수	<a href="http://technet.microsoft.com/en-us/library/dd347675.aspx">http://technet.microsoft.com/en-us/library/dd347675.aspx</a>
PowerShell 오류 보고	<a href="https://docs.microsoft.com/en-us/powershell/scripting/developer/cmdlet/error-reporting-concepts?view=powershell-7.2">https://docs.microsoft.com/en-us/powershell/scripting/developer/cmdlet/error-reporting-concepts?view=powershell-7.2</a>
PowerShell 일반 매개 변수	<a href="https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_commonparameters?view=powershell-7.2">https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_commonparameters?view=powershell-7.2</a>

## Data ONTAP PowerShell 툴킷

Data ONTAP PowerShell 툴킷은 WFA와 함께 번들로 제공됩니다. PowerShell 툴킷 cmdlet을 사용하여 PowerShell 스크립트에서 Data ONTAP 명령을 호출할 수 있습니다. 자세한 내용은 다음 위치에서 액세스할 수 있는 \_Data ONTAP PowerShell Toolkit 도움말\_을 참조하십시오.

"WFA\_INSTALL\_LOCATION\WFA\posh\Modules\DataONTAP\WebHelp\index.html"

'WFA\_INSTALL\_LOCATION'은 WFA 설치 디렉토리이며, 기본 설치 디렉토리는 'C:\Program Files\NetApp'입니다.

다음 표에는 Data ONTAP PowerShell 툴킷에 대한 정보에 대한 참조가 포함되어 있습니다.

ONTAP PowerShell 툴킷 문서입니다	<a href="https://community.netapp.com/t5/Tech-OnTap-Articles/The-Data-ONTAP-PowerShell-Toolkit/ta-p/85933">https://community.netapp.com/t5/Tech-OnTap-Articles/The-Data-ONTAP-PowerShell-Toolkit/ta-p/85933</a>
ONTAP PowerShell Toolkit NetApp 커뮤니티	<a href="https://community.netapp.com/t5/forums/filteredbylabelpage/board-id/microsoft-cloud-and-virtualization-discussions/label-name/powershell%20toolkit">https://community.netapp.com/t5/forums/filteredbylabelpage/board-id/microsoft-cloud-and-virtualization-discussions/label-name/powershell%20toolkit</a>

## Perl

WFA는 워크플로우 작업을 위해 Perl 명령을 지원합니다. WFA를 설치할 때 필요한 Perl 및 Perl 모듈이 WFA 서버에 설치됩니다.

["ActivePerl 사용 설명서"](#)

또한 다음 위치에서 \_ActivePerl 사용자 가이드 \_에 액세스할 수 있습니다:

"WFA\_install\_location\WFA\Perl64\HTML\index.html".

'WFA\_INSTALL\_LOCATION'은 WFA 설치 디렉토리이고, 'C:\Program Files\NetApp은 기본 설치 디렉토리입니다. WFA는 워크플로우 작업을 위해 Perl 스크립트를 사용합니다. 다음 표에는 Perl에 대한 학습 자료에 대한 참조가 포함되어 있습니다.

최신 Perl: 2014	<a href="http://modernperlbooks.com/books/modern_perl_2014/index.html">http://modernperlbooks.com/books/modern_perl_2014/index.html</a>
Perl 프로그래밍 설명서	<a href="http://perldoc.perl.org/">http://perldoc.perl.org/</a>
Perl 프로그래밍 언어	<a href="http://www.perl.org/">http://www.perl.org/</a>

## NetApp 관리 SDK

NetApp Manageability SDK의 필수 Perl 모듈은 WFA와 함께 번들로 제공됩니다. 이러한 Perl 모듈은 WFA에서 Perl 명령을 사용하는 데 필요합니다. 자세한 내용은 다음 위치에서 액세스할 수 있는 NetApp Manageability SDK 문서를 참조하십시오. 'WFA\_install\_location\WFA\perl\NMSDK\HTML'.

WFA\_INSTALL\_LOCATION은 WFA 설치 디렉토리이고, 기본 설치 디렉토리는 C:\Program Files\NetApp입니다.

## SQL(구조적 쿼리 언어)

SQL SELECT 구문은 필터에 사용되고 사용자 입력을 채우는 데 사용됩니다.

["MySQL 구문 선택"](#)

## MVFLEX Expression Language(MVEL)

함수 및 변수와 같은 WFA 워크플로우에서 MVEL expression syntax는 사용할 수 있습니다.

자세한 내용은 \_MVEL 언어 가이드\_를 참조하십시오.

## 정규식입니다

WFA에서는 정규식(regex)을 사용할 수 있습니다.

["정규식을 사용하는 ActionScript 3.0"](#)

## OnCommand Workflow Automation 관련 문서

WFA(OnCommand Workflow Automation) 서버에 대한 고급 구성의 수행하는 방법에 대해 자세히 알아볼 수 있는 추가 문서 및 툴이 있습니다.

## 기타 참조

NetApp 커뮤니티 내의 Workflow Automation 스페이스에서는 다음을 포함한 추가 학습 리소스를 제공합니다.

- NetApp 커뮤니티 \*

## 도구 참조

- \* 상호 운용성 매트릭스 \*

에는 지원되는 하드웨어 구성 요소 및 소프트웨어 버전 조합이 나와 있습니다.

["상호 운용성 매트릭스"](#)

## 저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그레픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 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 라이센스도 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 있으며 재사용이 불가능하며 취소 불가능한 라이센스를 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 라이센스는 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.