



Fluxo de trabalho

OnCommand Workflow Automation

NetApp
October 09, 2025

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

Índice

Fluxo de trabalho	1
Visão geral do OnCommand Workflow Automation	1
Características DO WFA	1
Informações de licença WFA	2
Entendendo o designer do Workflow Automation	2
Trabalhando com os blocos de construção em OnCommand Workflow Automation	2
O que é um banco de dados playground	10
Gerenciamento de fluxos de trabalho	11
Personalizar fluxos de trabalho predefinidos	11
Criação de fluxos de trabalho	13
Crie pacotes de fluxo de trabalho WFA	40
Integração de pacotes de fluxo de trabalho WFA com o repositório SCM	45
Criando componentes básicos para fluxos de trabalho	49
Crie um tipo de fonte de dados	49
Crie um comando	50
Crie um localizador	53
Crie um filtro	54
Crie uma entrada de dicionário	55
Crie uma função	55
Crie um modelo	56
Crie uma consulta de cache	57
Crie agendas recorrentes	58
Definir regras de filtro	59
Adicione pontos de aprovação	60
Diretrizes de codificação para WFA	61
Diretrizes para variáveis	61
Diretrizes para indentação	65
Diretrizes para comentários	66
Diretrizes para o Registro de logs	67
Diretrizes para tratamento de erros	69
Convenções gerais do PowerShell e Perl para O WFA	72
Considerações para adicionar módulos personalizados PowerShell e Perl	73
Cmdlets e funções DO WFA	74
Módulos do PowerShell e Perl WFA	74
Considerações ao converter comandos do PowerShell para Perl	77
Diretrizes para blocos de construção WFA	79
Palavras reservadas	92
Como você usa APIs REST	93
Referências ao material de aprendizagem	94
Windows PowerShell	94
Kit de ferramentas do Data ONTAP PowerShell	95
Perl	95
SDK de gerenciamento do NetApp	96

Linguagem de consulta estruturada (SQL)	96
Linguagem de expressão MVFLEX (MVEL)	96
Expressões regulares	96
Documentação relacionada para o OnCommand Workflow Automation	96
Outras referências	96
Referências da ferramenta	96

Fluxo de trabalho

Visão geral do OnCommand Workflow Automation

O OnCommand Workflow Automation (WFA) é uma solução de software que ajuda a automatizar tarefas de gerenciamento de storage, como provisionamento, migração, desativação, configurações de proteção de dados e clonagem de storage. Você pode usar O WFA para criar fluxos de trabalho para concluir tarefas que são especificadas por seus processos.

Um fluxo de trabalho é uma tarefa repetitiva e processual que consiste em etapas sequenciais, incluindo os seguintes tipos de tarefas:

- Provisionamento, migração ou desativação de storage para bancos de dados ou sistemas de arquivos
- Configuração de um novo ambiente de virtualização, incluindo switches de storage e datastores
- Configuração do storage de uma aplicação como parte de um processo de orquestração completo

Os arquitetos de storage podem definir fluxos de trabalho para seguir as práticas recomendadas e atender a requisitos organizacionais, como os seguintes:

- Usando convenções de nomenclatura necessárias
- Definir opções exclusivas para objetos de armazenamento
- Selecionando recursos
- Integração de banco de dados de gerenciamento de configuração interna (CMDB) e aplicativos de emissão de tíquetes

Características DO WFA

- Portal de design de fluxo de trabalho para criar fluxos de trabalho

O portal de design de fluxo de trabalho inclui vários blocos de construção, como comandos, modelos, localizadores, filtros e funções, que são usados para criar fluxos de trabalho. O designer permite que você inclua recursos avançados para fluxos de trabalho, como seleção automatizada de recursos, repetição de linhas (looping) e pontos de aprovação.

O portal de design de fluxo de trabalho também inclui componentes básicos, como entradas de dicionário, consultas de cache e tipos de fonte de dados, para armazenar dados em cache de sistemas externos.

- Portal de execução para executar fluxos de trabalho, verificar o status da execução do fluxo de trabalho e acessar logs
- Opção Administration/Settings (Administração/Definições) para tarefas como configurar O WFA, ligar a fontes de dados e configurar credenciais do utilizador
- Interfaces de serviço da Web para invocar fluxos de trabalho de portais externos e software de orquestração de data center
- Armazenamento Automation Store para baixar pacotes WFA. O pacote ONTAP 9.7,0 é fornecido com WFA 5,1.

Informações de licença WFA

Nenhuma licença é necessária para usar o servidor OnCommand Workflow Automation.

Entendendo o designer do Workflow Automation

Você cria fluxos de trabalho no designer do Workflow Automation (WFA) usando os componentes básicos, como finders, filtros e comandos. Entender os componentes básicos e o processo de criação de fluxo de trabalho é importante antes de começar a criar seus fluxos de trabalho.

Trabalhando com os blocos de construção em OnCommand Workflow Automation

Os fluxos de trabalho de automação do fluxo de trabalho (WFA) consistem em vários blocos de construção e O WFA inclui uma biblioteca dos blocos de construção predefinidos. Você pode usar os componentes básicos que O WFA fornece para criar fluxos de trabalho que correspondam aos requisitos da sua organização.

A WFA fornece a estrutura para processos de automação de armazenamento. A flexibilidade DO WFA é baseada em como os fluxos de trabalho são construídos usando os blocos de construção do fluxo de trabalho.

Os blocos de construção DO WFA são os seguintes:

- Entradas do dicionário
- Comandos
- Filtros
- Localizadores
- Funções
- Modelos

Você deve entender como os blocos de construção são usados no WFA para ajudá-lo a criar os fluxos de trabalho.

Quais são as fontes de dados

Uma fonte de dados é um método para estabelecer uma conexão com outros sistemas, arquivos e bancos de dados, a fim de extrair dados. Por exemplo, uma fonte de dados pode ser uma conexão com um banco de dados Active IQ Unified Manager do tipo de fonte de dados Active IQ Unified Manager 9,4.

Você pode adicionar uma fonte de dados personalizada ao OnCommand Workflow Automation (WFA) para aquisição de dados depois de definir o tipo de fonte de dados necessário associando o esquema de armazenamento em cache, a porta necessária e o método de aquisição com o tipo de fonte de dados.

O WFA armazena informações em cache através de várias fontes de dados. O WFA coleta informações de recursos das fontes de dados e formata-as para o esquema de armazenamento em cache. As tabelas de cache, que são as tabelas dentro dos esquemas de cache, são formatadas para corresponder aos objetos de entrada do dicionário. Quando você usa um localizador em fluxos de trabalho, ele retorna um objeto de dicionário e os dados do objeto de dicionário são preenchidos a partir das tabelas de cache. O processo de

aquisição de dados das fontes de dados é conhecido como *aquisição de fonte de dados*. Você pode usar um método baseado em script ou um método baseado em driver para aquisição de fonte de dados. As fontes podem ser diferentes umas das outras e a aquisição da fonte de dados pode amostrá-las em intervalos de tempo diferentes. O WFA, em seguida, funde essas informações no banco de dados e sobrepõe os dados de reserva para manter as informações de recursos atualizadas no banco de dados.

O banco de dados WFA inclui vários esquemas de armazenamento em cache diferentes. Um esquema de cache é um conjunto de tabelas e cada tabela inclui informações de um determinado tipo de entrada de dicionário; no entanto, as tabelas podem incluir informações combinadas de várias fontes de um tipo de fonte de dados específico. O WFA usa as informações do banco de dados para entender o status dos recursos, executar cálculos e executar comandos nos recursos.

Quais são as entradas do dicionário

Entradas de dicionário são um dos blocos de construção do OnCommand Workflow Automation (WFA). Você pode usar entradas de dicionário para representar tipos de objetos e seus relacionamentos em ambientes relacionados ao armazenamento e armazenamento. Em seguida, você pode usar filtros em fluxos de trabalho para retornar o valor das chaves naturais das entradas do dicionário.

Uma entrada de dicionário é a definição de um tipo de objeto que é suportado pelo WFA. Cada entrada de dicionário representa um tipo de objeto e sua relação nos ambientes relacionados ao armazenamento e armazenamento suportados. Um objeto de dicionário consiste em uma lista de atributos, que podem ser verificados. Um objeto de dicionário com valores completos descreve uma instância de objeto de um tipo. Além disso, os atributos de referência descrevem a relação do objeto com o ambiente; por exemplo, um objeto de dicionário de volume tem muitos atributos, como nome, tamanho_mb e volume_guarantee. Além disso, o objeto de dicionário de volume inclui referências ao agregado e ao array que contém o volume na forma de array_id e aggregate_id.

A tabela de cache de um objeto é um banco de dados contendo alguns ou todos os atributos da entrada do dicionário que são marcados para cache. Para que uma entrada de dicionário inclua uma tabela de cache, pelo menos um dos atributos da entrada de dicionário deve ser marcado para cache. As entradas de dicionário incluem chaves naturais, que são identificadores exclusivos para os objetos; por exemplo, os volumes do modo 7 são identificados exclusivamente pelo seu nome e pelo endereço IP da matriz que os contém. As Qtrees são identificadas pelo nome de qtree, pelo nome do volume e pelo endereço IP da matriz. Você deve identificar os atributos do dicionário que fazem parte das chaves naturais da entrada do dicionário ao criar entradas do dicionário.

Como os comandos funcionam

Os comandos OnCommand Workflow Automation são os blocos de execução para fluxos de trabalho. Você pode usar um comando para cada etapa do seu fluxo de trabalho.

Os comandos WFA são escritos usando scripts PowerShell e Perl. Os comandos do PowerShell usam o kit de ferramentas do Data ONTAP PowerShell e o VMware PowerCLI, se o pacote estiver instalado. Os comandos Perl usam a distribuição Perl e os módulos Perl instalados no servidor WFA. Se você incluir várias linguagens de script em um comando, como PowerShell e Perl, o script apropriado é escolhido pelo WFA com base no sistema operacional no qual ele está instalado e na ordem de idioma preferida especificada no menu de configuração DO WFA.

Os scripts para os comandos WFA incluem vários parâmetros. Esses parâmetros podem ser mapeados para os atributos de entrada do dicionário.

Observe que cada comando WFA pode incluir vários comandos Data ONTAP.

Alguns dos comandos WFA são conhecidos como comandos Wait porque eles podem esperar por operações de longa duração e poll periodicamente - por exemplo, o comando **Wait for multiple volume moves**. O intervalo de espera no qual o comando de polling é executado pode ser configurado para verificar se a operação foi concluída.

Um comando WFA é iniciado pelo WFA enquanto o fluxo de trabalho está em sua fase de execução. O WFA executa os comandos em série, em ordem esquerda para direita e de cima para baixo. O Planejamento do fluxo de trabalho confirma a disponibilidade e validade dos parâmetros que são fornecidos ao comando. O servidor WFA fornece todos os parâmetros necessários para os comandos antes que os comandos sejam executados.

Os parâmetros para comandos são finalizados durante o Planejamento do fluxo de trabalho. Em seguida, o fluxo de trabalho passa esses parâmetros para os comandos durante o tempo de execução. Os comandos não podem repassar parâmetros ao fluxo de trabalho. No entanto, se você quiser trocar informações obtidas durante o tempo de execução entre comandos em um fluxo de trabalho, você pode usar os cmdlets ou funções Perl designados DO WFA PowerShell.

Os comandos DO WFA PowerShell não usam o sinalizador de parada -ErrorAction para os cmdlets do PowerShell; portanto, as execuções de fluxo de trabalho continuam mesmo quando os cmdlets falham devido a um erro. Se você quiser que o sinalizador de parada -ErrorAction seja incluído em um comando específico, você pode clonar o comando e modificar o script do PowerShell para adicionar o sinalizador.

A seguir estão os cmdlets do PowerShell e as funções Perl incluídas no WFA para permitir a troca de informações entre comandos:

Cmdlets do PowerShell	Funções Perl
Add-WfaWorkflowParameter	AddWfaWorkflowParameter
Get-WfaWorkflowParameter	GetWfaWorkflowParameter

Os parâmetros adicionados pelos cmdlets ou funções "add" a um comando podem ser recuperados por um comando que é executado posteriormente e usa os cmdlets ou funções "get". Por exemplo, em um comando DO PowerShell WFA, você pode usar o seguinte no código para adicionar um parâmetro chamado volumeId: Add-WfaWorkflowParameter -Name "VolumeUUID" -Value "12345" -AddAsReturnParameter \$true. Em seguida, você pode usar o seguinte em um comando subsequente para recuperar o valor de volumeId: \$volumeId = Get-WfaWorkflowParameter -Name volumeId.

Os comandos WFA podem consultar o banco de dados WFA e obter o resultado necessário. Isso permite que você construa um comando sem usar filtros e finders. Você pode usar as seguintes funções para consultar o banco de dados:

Cmdlet do PowerShell	Função Perl
Invoke-MySqlQuery por exemplo: Invoke- MySqlQuery -Query "SELECT cluster.name AS 'Cluster Name' FROM cm_storage.cluster"	InvokeMySqlQuery

Informações relacionadas

[Crie um filtro](#)

[Crie um localizador](#)

[Crie um comando](#)

Quais são os filtros

Você pode usar filtros WFA em seus fluxos de trabalho para selecionar os recursos necessários.

Um filtro WFA é uma consulta baseada em SQL que funciona no banco de dados WFA. Cada filtro retorna uma lista de elementos de um tipo específico de dicionário. Os elementos retornados são baseados nos critérios de seleção especificados na consulta SQL. Você deve estar ciente das sintaxes SQL para criar ou editar um filtro.

O que são os localizadores

Um localizador é uma combinação de um ou mais filtros que são usados em conjunto para identificar resultados comuns. Você pode usar um localizador em seus fluxos de trabalho para selecionar os recursos necessários para a execução do fluxo de trabalho.

Os localizadores podem aplicar uma ordem de classificação para diferenciar os resultados aplicáveis. Os localizadores retornam o melhor recurso com base nos critérios de seleção e classificação.

Os localizadores retornam um resultado ou nenhum resultado; portanto, eles podem ser usados para verificar a existência de certos elementos de armazenamento. No entanto, quando um localizador é usado como parte de uma definição de linha repetida, os conjuntos de resultados são usados para formar a lista de membros do grupo. Os filtros que são usados em finders retornam a chave natural do tipo de dicionário, no mínimo, mas podem retornar campos adicionais, cujo valor pode ser referenciado. Uma ordem de classificação pode ser aplicada a qualquer campo retornado da consulta SQL de um filtro.

Você pode testar os resultados de um localizador. Ao testar um localizador, você pode visualizar os resultados comuns de todos os filtros WFA, onde o resultado efetivo do localizador é destacado nos resultados. Ao usar um localizador em um fluxo de trabalho, você pode criar uma mensagem de erro personalizada para transmitir informações significativas ao operador de armazenamento.

Quais são as funções

Você pode usar uma função em seus fluxos de trabalho para uma tarefa complexa que deve ser concluída durante a fase de Planejamento do fluxo de trabalho.

Você pode escrever funções usando a MVFLEX Expression Language (MVEL). Você pode usar funções para reunir lógica comumente usada, bem como lógica mais complexa em uma função nomeada e reutilizá-la como valores para parâmetros de comando ou parâmetros de filtro. Você pode escrever uma função uma vez e usá-la em fluxos de trabalho. Você pode usar funções para lidar com tarefas repetitivas e tarefas que podem ser complexas, como definir uma convenção de nomenclatura complexa.

Funções podem usar outras funções durante a execução.

Quais são os esquemas

Um esquema representa o modelo de dados de um sistema. Um modelo de dados é

uma coleção de entradas de dicionário. Você pode definir um esquema e, em seguida, definir um tipo de fonte de dados. A fonte de dados define como os dados são adquiridos e o esquema é preenchido. Por exemplo, um esquema de vc adquire dados sobre seu ambiente virtual, como máquinas virtuais, hosts e datastores.

Os esquemas também podem ser preenchidos diretamente com dados por meio de fluxos de trabalho personalizados para resolver problemas específicos.

As entradas do dicionário são associadas a um esquema existente quando as entradas do dicionário são criadas. Entradas de dicionário também são associadas a consultas de cache, e as consultas de cache incluem consultas SQL.

Os esquemas podem adquirir dados usando o tipo de fonte de dados baseada em script ou o tipo de fonte de dados SQL. Os scripts são definidos ao criar o tipo de fonte de dados e as consultas SQL são definidas nas consultas de cache.

Os seguintes esquemas estão incluídos no WFA:

- **Modo 7 (armazenamento)**

Esquema para adquirir dados através do Active IQ Unified Manager a partir do Data ONTAP operando no modo 7D.

- **Clustered Data ONTAP (cm_storage)**

Esquema para adquirir dados por meio do Active IQ Unified Manager a partir do Clustered Data ONTAP.

- **Desempenho de 7 modos (desempenho)**

Esquema para adquirir dados de desempenho do Data ONTAP operando no modo 7 por meio do Performance Advisor.

- **Desempenho de cluster Data ONTAP (cm_performance)**

Esquema para adquirir dados de desempenho do Clustered Data ONTAP por meio do consultor de desempenho.

- **VMware vCenter (VC)**

Esquema para adquirir dados do VMware vCenter.

- **Playground (playground)**

Esquema que você pode preencher diretamente com dados.

Quais são os tipos de sistema remoto

O OnCommand Workflow Automation (WFA) comunica com os tipos de sistemas remotos. Um tipo de sistema remoto especifica o tipo de sistemas remotos com os quais O WFA pode se comunicar. Você pode configurar tipos de sistema remoto NO WFA. Por exemplo, o sistema Data ONTAP pode ser configurado como um tipo de sistema remoto.

Um tipo de sistema remoto tem os seguintes atributos:

- Nome
- Descrição
- Versão
- Protocolo
- Porta
- Tempo limite

Você pode ter um script Perl para cada tipo de sistema remoto para validar as credenciais do sistema remoto. Você pode armazenar as credenciais para os sistemas remotos configurados no WFA. Pode adicionar ou editar um novo tipo de sistema remoto personalizado. Você também pode clonar um tipo de sistema remoto existente. Você pode excluir um tipo de sistema remoto somente se nenhum sistema estiver associado a ele.

Como você usa modelos

Você pode usar modelos WFA em seus fluxos de trabalho como referência ou para aderir a políticas de uso.

Um modelo WFA atua como um blueprint de uma definição de objeto. Você pode definir um modelo incluindo as propriedades de um objeto e os valores para as propriedades do objeto. Em seguida, você pode usar o modelo para preencher as propriedades de uma definição de objeto em seus fluxos de trabalho.

Quando você usa um modelo, não é possível editar os campos que incluem os valores obtidos a partir do modelo. Portanto, você pode usar modelos para configurar políticas de uso e criação de objetos. Se você remover a associação de um modelo com o fluxo de trabalho depois de aplicar o modelo, os valores preenchidos do modelo permanecerão, mas você poderá editar os campos.

Como você usa categorias

Você pode categorizar seus fluxos de trabalho para organizar melhor os fluxos de trabalho e aplicar a funcionalidade de controle de acesso nos fluxos de trabalho.

Você pode categorizar fluxos de trabalho de forma que eles apareçam em grupos específicos no PORTAL DO WFA. Você também pode aplicar o recurso de controle de acesso em categorias de fluxo de trabalho. Por exemplo, você pode permitir que apenas certos operadores de armazenamento ou aprovadores visualizem determinadas categorias de fluxos de trabalho. Os operadores de armazenamento ou aprovadores podem executar apenas os fluxos de trabalho dentro da categoria para a qual foram concedidos direitos de acesso.

Os grupos do ative Directory também podem ser usados para controle de acesso a categorias.

Como funciona o versionamento de entidade

As entidades OnCommand Workflow Automation (WFA), como comandos e fluxos de trabalho, são versionadas. Você pode usar os números de versão para gerenciar facilmente alterações nas entidades WFA.

Cada entidade WFA inclui um número de versão no `major.minor.revision` formato - por exemplo, 1.1.20. Você pode incluir até três dígitos em cada parte do número da versão.

Antes de modificar o número de versão de uma entidade WFA, você deve estar ciente das seguintes regras:

- Os números de versão não podem ser alterados da versão atual para uma versão anterior.

- Cada parte da versão deve ser um número de 0 a 999.
- Novas entidades WFA são versionadas como 1,0,0, por padrão.
- O número de versão de uma entidade é retido durante a clonagem ou usando **Salvar como** para salvar uma cópia da entidade.
- Várias versões de uma entidade não podem existir em uma instalação DO WFA.

Quando você atualiza a versão de uma entidade WFA, a versão de sua entidade pai imediata é atualizada automaticamente. Por exemplo, a atualização da versão do comando **Create volume** atualiza o fluxo de trabalho **Create a NFS volume**, porque o fluxo de trabalho **Create a NFS volume** é uma entidade principal imediata do comando **Create volume**. A atualização automática das versões é aplicada da seguinte forma:

- Modificar a versão principal de uma entidade atualiza a versão menor de suas entidades pai imediatas.
- Modificar a versão menor de uma entidade atualiza a versão de revisão de suas entidades pai imediatas.
- Modificar a versão de revisão de uma entidade não atualiza nenhuma parte da versão de suas entidades pai imediatas.

A tabela a seguir lista as entidades WFA e suas entidades-mãe imediatas:

Entidade	Entidade principal imediata
Consulta de cache	<ul style="list-style-type: none"> • Tipo de fonte de dados
Modelo	<ul style="list-style-type: none"> • Fluxo de trabalho
Função	<ul style="list-style-type: none"> • Fluxo de trabalho • Modelo <p> Se uma função contiver caracteres especiais ou mistos, a versão de suas entidades pai imediatas pode não ser atualizada.</p>
Dicionário	<ul style="list-style-type: none"> • Modelo • Filtro • Consulta de cache • Comando • Tipos de fonte de dados que estão usando o método script
Comando	<ul style="list-style-type: none"> • Fluxo de trabalho
Filtro	<ul style="list-style-type: none"> • Localizador • Fluxo de trabalho
Localizador	<ul style="list-style-type: none"> • Fluxo de trabalho

Entidade	Entidade principal imediata
Tipo de fonte de dados	Nenhum
Fluxo de trabalho	Nenhum

Você pode procurar uma entidade no WFA usando as partes do número da versão ou o número da versão completa.

Se você excluir uma entidade pai, as entidades filho serão mantidas e sua versão não será atualizada para a exclusão.

Como o controle de versão funciona ao importar entidades

Se você importar entidades de versões anteriores ao Workflow Automation 2.2, as entidades serão versionadas como 1.0.0, por padrão. Se a entidade importada já estiver presente no servidor WFA, a entidade existente será substituída pela entidade importada.

A seguir estão as possíveis mudanças nas entidades WFA durante uma importação:

- Atualização de entidades

As entidades são substituídas por uma versão posterior.

- Reversão de entidades

As entidades são substituídas por uma versão anterior.



Quando você executa uma reversão de uma entidade, a versão de suas entidades pai imediatas é atualizada.

- Importação de novas entidades



Você não pode importar entidades seletivamente de um .dar arquivo.

Se uma versão posterior de uma entidade for importada, a versão de suas entidades pai imediatas será atualizada.

Se houver várias entidades filho para a entidade pai importada, somente o maior grau de alteração (maior, menor ou revisão) para as entidades filho será aplicado à entidade pai. Os exemplos a seguir explicam como esta regra funciona:

- Para uma entidade principal importada, se houver uma entidade filha com uma alteração menor e outra entidade filha com uma alteração de revisão, a alteração menor será aplicada à entidade pai.

A parte de revisão da versão do pai é incrementada.

- Para uma entidade pai importada, se houver uma entidade filha com uma alteração maior e outra entidade filha com uma alteração menor, a alteração principal será aplicada à entidade pai.

A parte menor da versão do pai é incrementada.

Exemplo de como as versões de entidades filho importadas afetam a versão pai

Considere o seguinte fluxo de trabalho no WFA: ""criar volume e exportar usando NFS - Custom"" 1,0,0.

Os comandos existentes incluídos no fluxo de trabalho são os seguintes:

- "Criar política de exportação - personalizada" 1.0.0
- "Criar volume - Personalizado" 1.0.0

Os comandos incluídos no .dar arquivo, que deve ser importado, são os seguintes:

- "Criar política de exportação - personalizada" 1.1.0
- "Criar volume - Personalizado" 2.0.0

Quando você importa .dar esse arquivo, a versão menor do fluxo de trabalho ""criar volume e exportar usando NFS - Personalizado"" é incrementada para 1,1,0.

O que é um banco de dados playground

O banco de dados playground é um banco de dados MySQL, que está incluído na instalação do servidor do Workflow Automation (WFA). Você pode adicionar tabelas ao banco de dados playground para incluir informações, que podem ser usadas por filtros e consultas SQL para entradas de usuário.

O banco de dados playground é um esquema que não pode ser acessado através do portal web DO WFA. Você pode usar um cliente MySQL, como SQLyog, Toad for MySQL e MySQL Workbench ou uma interface de linha de comando (CLI), para acessar o banco de dados.

Você deve usar as seguintes credenciais para acessar o banco de dados do playground:

- Nome de usuário: wfa
- Senha: Wfa123

As credenciais fornecem acesso completo ao banco de dados playground e acesso somente leitura a outros esquemas definidos no banco de dados MySQL WFA. Você pode criar as tabelas necessárias no banco de dados do playground.

Você pode adicionar as tags ou metadados que você está usando para objetos de armazenamento em seu ambiente a uma tabela no banco de dados do playground. As tags ou metadados podem então ser usados juntamente com as informações em outras tabelas de cache WFA por filtros WFA e consultas de entrada de usuário.

Por exemplo, você pode usar o banco de dados playground para os seguintes casos de uso:

- A marcação de agregados com o nome da unidade de negócio (BU) e a alocação de volumes para o barramento com base nessas tags
- Marcação de unidades do vFiler com nomes BU
- Adicionando detalhes de geografia ou localização a objetos de armazenamento
- Definindo o acesso de administradores de banco de dados para bancos de dados

Por exemplo, se você estiver usando o nome da BU como uma tag para os objetos de storage, como

agregados e unidades do vFiler, poderá criar uma tabela no banco de dados do playground que inclua o nome da BU. O nome BU pode então ser usado por filtros e consultas de entrada do usuário para seus fluxos de trabalho.

A seguir está um exemplo de tabela de banco de dados playground (playground.volume_bu):

array_ip	volume_name	BU
10.225.126.23	data_11	Marketing
10.225.126.28	arch_11	HR

A seguir está um exemplo de consulta SQL que você pode usar para filtrar volumes por BU:

```
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}'
```

Informações relacionadas

["SQLyog"](#)

["MySQL Workbench"](#)

["Toad para MySQL"](#)

Gerenciamento de fluxos de trabalho

Você pode personalizar fluxos de trabalho predefinidos ou criar novos fluxos de trabalho como parte do gerenciamento de fluxos de trabalho. Você também deve entender os conceitos relevantes antes de começar a gerenciar seus fluxos de trabalho.

Personalizar fluxos de trabalho predefinidos

Você pode personalizar um fluxo de trabalho predefinido do Workflow Automation (WFA) se não houver um fluxo de trabalho predefinido adequado para suas necessidades.

O que você vai precisar

Você deve ter identificado as modificações necessárias para o fluxo de trabalho predefinido.

Sobre esta tarefa

Perguntas e solicitação de suporte para o seguinte devem ser dirigidas à comunidade WFA:

- Qualquer conteúdo baixado da comunidade WFA
- Conteúdo WFA personalizado que você criou
- Conteúdo WFA que você modificou

Passos

1. Clique em **Workflow Design > fluxos de trabalho**.
2. Selecione o fluxo de trabalho predefinido que corresponde de perto ao seu requisito e, em seguida, clique  na barra de ferramentas.
3. No designer de fluxo de trabalho, faça as alterações necessárias nas guias apropriadas, como editar a descrição, adicionar ou excluir um comando, modificar os detalhes do comando e modificar a entrada do usuário.
4. Clique em **Preview**, insira as entradas de usuário necessárias para visualizar a execução do fluxo de trabalho e clique em **Preview** para visualizar os detalhes de Planejamento do fluxo de trabalho.
5. Clique em **OK** para fechar a janela de visualização.
6. Clique em **Salvar**.

Depois de terminar

Você pode testar o fluxo de trabalho modificado em seu ambiente de teste e marcar o fluxo de trabalho como pronto para produção.

Personalize o fluxo de trabalho criar um volume e um compartilhamento CIFS

Você pode personalizar seus fluxos de trabalho com base em suas necessidades. Por exemplo, você pode modificar o fluxo de trabalho predefinido *Create a volume e um CIFS Share* para incluir deduplicação e compactação.

Sobre esta tarefa

A personalização e ilustrações nesta tarefa são exemplos; você pode modificar os fluxos de trabalho DO WFA com base em suas necessidades.

Passos

1. Clique em **Workflow Design > fluxos de trabalho**.
2. Selecione o fluxo de trabalho **criar um volume e um compartilhamento CIFS** e clique  na barra de ferramentas.
3. Clique na guia **Detalhes** e edite a descrição do fluxo de trabalho no campo **Nome do fluxo de trabalho**.
4. Clique na guia **Workflow**, expanda o esquema **storage** e arraste e solte o comando **Configuração de desduplicação e compactação** entre os comandos **criar volume** e **criar compartilhamento CIFS**.
5. Coloque o cursor do Mouse abaixo do comando **Configuração de desduplicação e compactação** na primeira linha e clique  em .
6. Na guia **volume** da caixa de diálogo **parâmetros para 'Configuração de deduplicação e compactação'**, selecione a opção **usando um volume definido anteriormente** e, em seguida, selecione a opção

Share_volume no campo **Definir volume**, que é a variável volume objeto criada pelo comando **criar volume** no fluxo de trabalho.

7. Clique na guia **outros parâmetros** e execute as seguintes etapas:
 - a. Selecione **True** no campo **StartNow**.
 - b. Selecione **Inline** no campo **Compression**.
 - c. Digite '`sun-sat@1`' expressão no campo **Schedule**, que agenda deduplicação e compactação em todos os dias da semana às 1 da manhã
8. Clique em **OK**.
9. Clique em **Preview** para garantir que o Planejamento do fluxo de trabalho seja concluído com êxito e clique em **OK**.
10. Clique em **Salvar**.

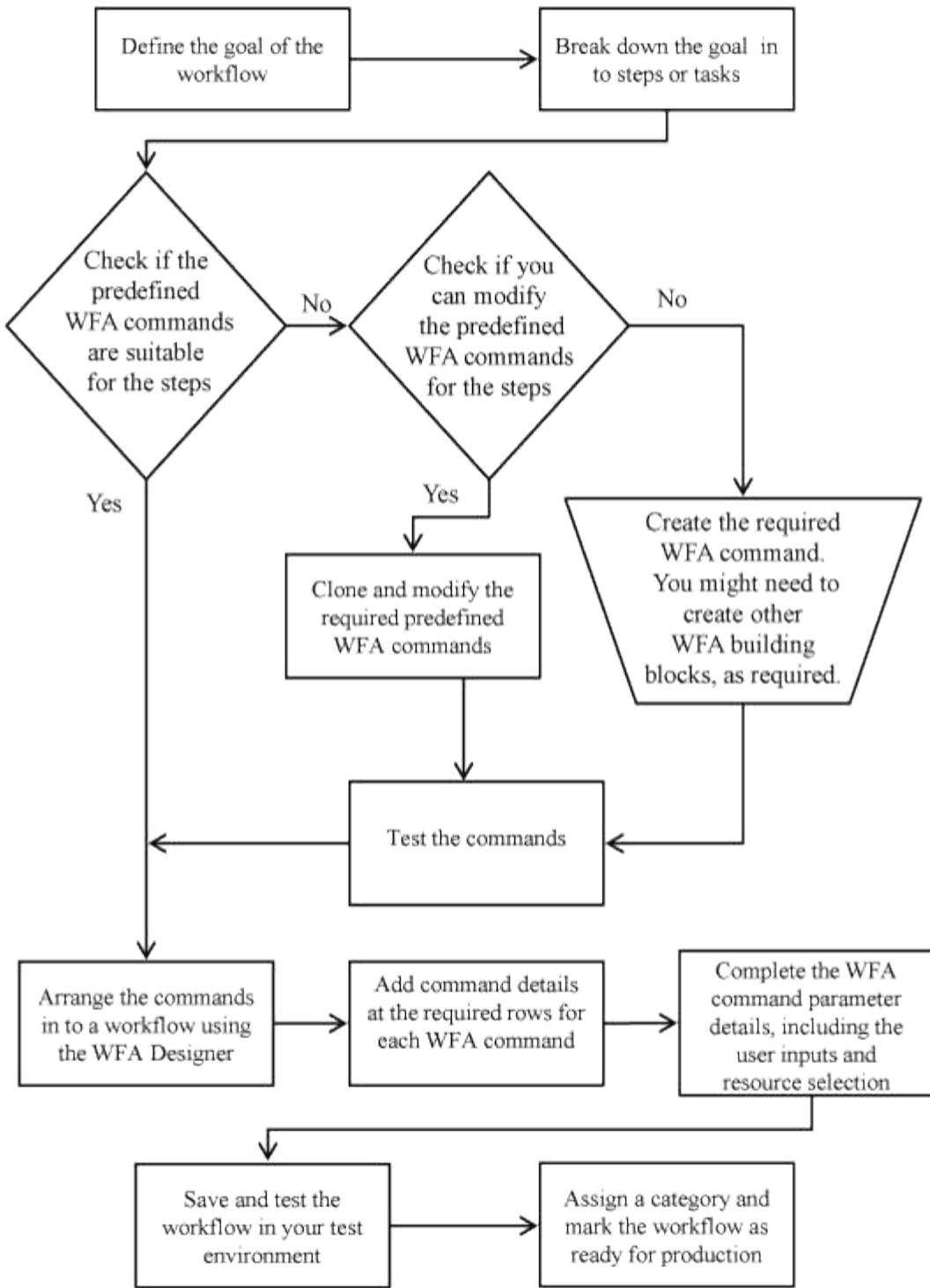
Criação de fluxos de trabalho

Se os fluxos de trabalho predefinidos não corresponderem aos seus requisitos, pode criar o fluxo de trabalho necessário. Antes de criar seus fluxos de trabalho, você deve entender os recursos disponíveis no designer DO WFA e criar uma lista de verificação de fluxo de trabalho.

Tarefas envolvidas na criação de fluxos de trabalho

A criação de fluxos de trabalho de automação de armazenamento no OnCommand Workflow Automation (WFA) inclui definir as etapas a serem executadas por um fluxo de trabalho e criar o fluxo de trabalho usando os componentes básicos DO WFA, como comandos, localizadores, filtros e entradas de dicionário.

O fluxograma a seguir ilustra o processo de criação de fluxo de trabalho:



Como você define fluxos de trabalho

Você deve dividir o objetivo de um fluxo de trabalho nas etapas que devem ser executadas pelo fluxo de trabalho. Em seguida, você pode organizar as etapas para concluir seu fluxo de trabalho.

Um fluxo de trabalho é um algoritmo que inclui uma série de etapas que são necessárias para concluir um processo de ponta a ponta. O escopo do processo pode variar, dependendo do objetivo do fluxo de trabalho. O objetivo de um fluxo de trabalho pode ser definido para lidar apenas com operações de armazenamento ou processos mais complexos, como lidar com redes, virtualização, sistemas DE TI e outros aplicativos como

parte de um único processo. Os fluxos de trabalho OnCommand Workflow Automation (WFA) são projetados por arquitetos de armazenamento e são executados por operadores de armazenamento.

Definir seu fluxo de trabalho inclui dividir o objetivo do fluxo de trabalho em uma série de etapas - por exemplo, criar um volume NFS inclui as seguintes etapas:

1. Criando um objeto de volume
2. Criar uma nova política de exportação e associar a política ao volume

Você pode usar um comando WFA ou um fluxo de trabalho para cada etapa do seu fluxo de trabalho. O WFA inclui comandos e fluxos de trabalho predefinidos, que são baseados em casos de uso comuns de armazenamento. Se você não encontrar um comando ou fluxo de trabalho predefinido que possa ser usado para uma etapa específica, você pode fazer um dos seguintes procedimentos:

- Escolha um comando ou fluxo de trabalho predefinido que corresponda de perto à etapa e, em seguida, clone e modifique o comando ou fluxo de trabalho predefinido de acordo com seus requisitos.
- Crie um novo comando ou fluxo de trabalho.

Em seguida, você pode organizar os comandos ou fluxos de trabalho em um novo fluxo de trabalho para criar o fluxo de trabalho que cumpre sua meta.

No início da execução do fluxo de trabalho, O WFA planeja a execução e verifica se o fluxo de trabalho pode ser executado usando a entrada para o fluxo de trabalho e os comandos. Ao Planejar o fluxo de trabalho, toda a seleção de recursos e entrada de usuário são resolvidos para criar um plano de execução. Após o Planejamento ser concluído, O WFA executa o plano de execução, que consiste em um conjunto de comandos WFA com parâmetros aplicáveis.

Como as entradas do usuário são definidas

As entradas de usuário do OnCommand Workflow Automation (WFA) são opções de entrada de dados disponíveis durante a execução de fluxos de trabalho. Você deve definir os parâmetros de entrada do usuário para seus fluxos de trabalho para melhorar a flexibilidade e a usabilidade de seus fluxos de trabalho.

As entradas do usuário são mostradas como campos de entrada, que podem ser preenchidos com dados relevantes ao visualizar ou executar fluxos de trabalho. Você pode criar um campo de entrada de usuário ao especificar os detalhes do comando em um fluxo de trabalho prefixando um rótulo ou variável com o sinal de dólar. Por exemplo, o VolumeName cria um campo de entrada de usuário Nome de volume. O WFA preenche automaticamente o separador User Inputs (entradas do utilizador) na janela Workflow <workflow name> (fluxo de trabalho) com as etiquetas de entrada do utilizador que criou. Você também pode definir o tipo de entrada do usuário e personalizar os campos de entrada modificando os atributos de entrada do usuário, como tipo, nome de exibição, valores padrão e valores de validação.

Opções do tipo de entrada do usuário

- * String*

Você pode usar uma expressão regular para valores válidos - por exemplo, a*.

Strings, como 0d e 0f, são avaliadas como números semelhantes aos 0d avaliados como 0 do tipo duplo.

- Número

Você pode definir um intervalo numérico que pode ser selecionado - por exemplo, 1 through 15.

- **Enum**

Você pode criar valores de enumeração que podem ser selecionados ao preencher o campo de entrada do usuário usando o tipo enum. Opcionalmente, você pode bloquear os valores de enum que você criou para garantir que apenas os valores que você criou sejam selecionados para a entrada do usuário.

- **Consulta**

Você pode selecionar o tipo de consulta quando quiser que a entrada do usuário seja selecionada a partir dos valores disponíveis no cache WFA. Por exemplo, você pode usar a seguinte consulta para preencher automaticamente os campos de entrada do usuário com os valores de endereço IP e nome do cache WFA: **SELECT ip, name FROM storage.array**. Você pode opcionalmente bloquear os valores recuperados por uma consulta para que apenas os resultados tentados pela consulta sejam selecionados.

- **Consulta (multi-select)**

O tipo de consulta (multi-select), que é semelhante ao tipo de consulta, permite a seleção de vários valores durante a execução do fluxo de trabalho. Por exemplo, os usuários podem selecionar vários volumes ou um volume juntamente com seus compartilhamentos e exportações. Você pode permitir que os usuários selecionem várias linhas ou restringir a seleção a uma única linha. Selecionar uma linha seleciona os valores de todas as colunas da linha selecionada.

Você pode usar as seguintes funções ao usar o tipo de consulta (seleção múltipla) de entrada de usuário:

- GetSize
- GetValueAt
- getValueAt2D
- getValueFrom2DByRowKey
 - **Booleano**

Você pode usar o tipo booleano para exibir uma caixa de seleção na caixa de diálogo de entrada do usuário. Você deve usar o tipo booleano para entradas de usuário que têm "true" e "false" como os valores possíveis.

- **Tabela**

Você pode usar o tipo de tabela de entrada de usuário para especificar os cabeçalhos de coluna de uma tabela que podem ser usados para inserir vários valores durante a execução do fluxo de trabalho. Por exemplo, uma tabela que pode ser usada para especificar uma lista de nomes de nós e nomes de portas. Você também pode especificar um dos seguintes tipos de entrada de usuário para os cabeçalhos de coluna para validar os valores que são inseridos durante o tempo de execução:

- Cadeia de caracteres
- Número
- Enum
- Booleano * `String`Query é o tipo de entrada padrão do usuário para os cabeçalhos de coluna. Você deve clicar duas vezes na coluna tipo para especificar um tipo de entrada de usuário diferente.

Você pode abrir o fluxo de trabalho Create SnapMirror policy and rules no Designer para ver como os tipos de entrada de usuário são usados na entrada de usuário "SnapMirrorPolicyRule".

Você pode usar as seguintes funções ao usar o tipo de tabela de entrada do usuário:

- GetSize
- GetValueAt
- getValueAt2D
- getValueFrom2DByRowKey você pode abrir o fluxo de trabalho **criar e configurar uma Máquina Virtual de armazenamento com Infinite volume** no Designer para ver como o tipo de tabela é usado.

- **Senha**

Você pode usar o tipo de senha para entradas de usuário que são destinadas a inserir senhas. A palavra-passe introduzida pelo utilizador é encriptada e apresentada como uma sequência de carateres de asterisco na aplicação WFA e nos ficheiros de registo. Você pode usar as seguintes funções para descriptografar a senha, que pode ser usada pelo comando:

- Para comandos Perl: WFAUtil::getWfaInputPassword (em inglês)
- Para comandos do PowerShell: Get-WfaInputPassword -EncryptedPassword

Aqui, a senha é a senha criptografada que é passada pelo WFA para o comando.

- **Dicionário**

Você pode adicionar os dados da tabela para a entrada do dicionário selecionado. O atributo de entrada do dicionário seleciona o atributo que deve ser retornado. Você pode selecionar um único valor ou vários valores durante a execução do fluxo de trabalho. Por exemplo, você pode selecionar um único volume ou vários volumes. Por padrão, valores únicos são selecionados. Você também pode selecionar regras para filtragem. Uma regra consiste em um atributo de entrada de dicionário, um operador e um valor. O atributo também pode incluir atributos de suas referências.

Por exemplo, você pode especificar uma regra para agregados listando todos os agregados com nome começando com a cadeia "aggr" e ter um tamanho disponível maior que 5 GB. A primeira regra no grupo é o atributo name, com o operador starts-with`e o valor aggr. A segunda regra para o mesmo grupo é o atributo `available_size_mb, com o operador > e o valor 5000.

A tabela a seguir lista as opções que você pode aplicar aos tipos de entrada de usuário:

Opção	Descrição
A validar	<p>Você pode validar o tipo de entradas do usuário para que apenas valores válidos sejam inseridos pelos usuários:</p> <ul style="list-style-type: none">• Os tipos de string e número de entrada do usuário podem ser validados com os valores inseridos durante o tempo de execução do fluxo de trabalho.• O tipo de string também pode ser validado com uma expressão regular.• O tipo de número é um campo numérico de ponto flutuante e pode ser validado usando um intervalo numérico especificado.

Opção	Descrição
Valores de bloqueio	Você pode bloquear os valores dos tipos de consulta e enum para impedir que o usuário substitua os valores suspensos e ativar a seleção apenas dos valores exibidos.
Marcação como obrigatória	Você pode marcar as entradas do usuário como obrigatórias para que os usuários tenham que inserir certas entradas de usuário para continuar com a execução do fluxo de trabalho.
Agrupamento	Você pode agrupar entradas de usuário relacionadas e fornecer um nome para o grupo de entrada de usuário. Os grupos podem ser expandidos e colapsados na caixa de diálogo de entrada do usuário. Você pode selecionar um grupo que deve ser expandido por padrão.
Condições de aplicação	Com a capacidade condicional de entrada de usuário, você pode definir o valor de uma entrada de usuário com base no valor inserido para outra entrada de usuário. Por exemplo, em um fluxo de trabalho que configura o protocolo nas, você pode especificar a entrada de usuário necessária para o protocolo como NFS para habilitar a entrada de usuário "Read/Write host lists".

Como você mapeia parâmetros de comando

Os parâmetros nos comandos do Workflow Automation (WFA) são mapeados para atributos específicos e referências de entrada de dicionário com base em certas regras. Você deve estar ciente das regras para mapear parâmetros de comando ao criar ou editar um comando WFA.

O mapeamento de parâmetros de comando define como os detalhes do comando são definidos nos fluxos de trabalho. Os parâmetros de comando mapeados de um comando são exibidos em abas quando você está especificando os detalhes do comando para comandos em fluxos de trabalho. As guias são nomeadas com base no nome do grupo especificado na coluna Nome do objeto da guia Mapeamento de parâmetros. Os parâmetros que não são mapeados são exibidos na guia outros parâmetros quando você está especificando os detalhes do comando em fluxos de trabalho.

As regras para mapeamento de parâmetros de comando são aplicáveis com base na categoria de comando e como os comandos são representados no editor de fluxo de trabalho.

A seguir estão as categorias de comando:

- Comandos que criam objetos
- Comandos que atualizam objetos
- Comandos que removem objetos

- Comandos que lidam com objetos pai e filho opcionais
- Comandos que atualizam associações entre objetos

As regras estão listadas abaixo para cada categoria:

Todas as categorias de comandos

Ao mapear um parâmetro de comando, você deve usar o caminho natural com base em como o comando é usado em fluxos de trabalho.

Os exemplos a seguir mostram como você pode definir um caminho natural:

- Para o parâmetro ArrayIP, dependendo do comando, você deve usar o atributo aggregate.array.ip da entrada do dicionário volume e não o atributo array.ip.

Isso é importante quando um fluxo de trabalho cria um volume e, em seguida, executa uma etapa adicional com o volume criado consultando-o. Os seguintes exemplos são semelhantes:

- Volume.Aggregate.array.ip da entrada do dicionário Qtree
 - Volume.Aggregate.array.ip da entrada do dicionário LUN
- Para Cluster usado em comandos, você deve usar um dos seguintes:
 - SVM.cluster.primary_address da entrada do dicionário volume
 - Volume.vserver.cluster.primary_address da entrada do dicionário Qtree

Comandos que criam objetos

Esta categoria de comandos é usada para um dos seguintes:

- Encontrando um objeto pai e definindo novos objetos
- Procurar um objeto e criar o objeto se o objeto não existir

Você deve usar as seguintes regras de mapeamento de parâmetros para esta categoria de comandos:

- Mapeie os parâmetros relevantes do objeto que é criado para a entrada do dicionário do objeto.
- Mapeie o objeto pai através das referências da entrada do dicionário criada.
- Certifique-se de que o atributo relevante esteja presente na entrada do dicionário ao adicionar um novo parâmetro.

A seguir estão os cenários de exceção para esta regra:

- Alguns objetos que são criados não têm uma entrada de dicionário correspondente e apenas o objeto pai é mapeado para a entrada relevante do dicionário pai - por exemplo, o comando **Create VIF** - no qual apenas uma matriz pode ser mapeada para a entrada do dicionário de matriz.
- O mapeamento de parâmetros não é necessário

Por exemplo, o parâmetro ExecutionTimeout no comando **criar ou redimensionar agregado** é um parâmetro não mapeado.

Os seguintes comandos certificados são exemplos para esta categoria:

- Criar volume

- Criar LUN

Comandos que atualizam objetos

Esta categoria de comandos é usada para encontrar um objeto e atualizar os atributos.

Você deve usar as seguintes regras de mapeamento de parâmetros para esta categoria de comandos:

- Mapeie os objetos que são atualizados para a entrada do dicionário.
- Não mapeie os parâmetros que são atualizados para o objeto.

Por exemplo, no comando **Set volume State**, o parâmetro volume é mapeado, mas o novo Estado não é mapeado.

Comandos que removem objetos

Esta categoria de comandos é usada para encontrar um objeto e excluí-lo.

Você deve mapear o objeto que é excluído pelo comando para sua entrada de dicionário. Por exemplo, no comando **Remover volume**, o volume a ser excluído é mapeado para os atributos e referências relevantes da entrada do dicionário volume.

Comandos que lidam com objetos pai e filho opcionais

Você deve usar as seguintes regras de mapeamento de parâmetros para esta categoria de comandos:

- Não mapeie qualquer parâmetro obrigatório de um comando como referência a partir de um parâmetro opcional do comando.

Essa regra é mais relevante quando um comando lida com objetos filho opcionais de um objeto pai específico. Nesse caso, o objeto filho e pai devem ser mapeados explicitamente. Por exemplo, no comando **Stop Deduplication jobs**, o comando pára uma tarefa de deduplicação em execução em um volume específico quando especificado junto com Array ou em todos os volumes da matriz dada. Nesse caso, o parâmetro array deve ser mapeado diretamente para a entrada do dicionário array e não para volume.Array porque o volume é um parâmetro opcional neste comando.

- Se existir uma relação pai e filho entre entradas de dicionário no nível lógico, mas não entre as instâncias reais em um comando específico, esses objetos devem ser mapeados separadamente.

Por exemplo, no comando **mover volume**, volume é movido de seu agregado pai atual para um novo agregado de destino. Portanto, os parâmetros de volume são mapeados para uma entrada do dicionário de volume e os parâmetros agregados de destino são mapeados separadamente para a entrada do dicionário agregado, mas não como volume.aggregate.name.

Comandos que atualizam associações entre objetos

Para esta categoria de comandos, você deve mapear tanto a associação quanto os objetos para entradas relevantes do dicionário. Por exemplo, no comando Adicionar volume ao vFiler, os parâmetros volume e vFiler são mapeados para os atributos relevantes das entradas do dicionário volume e do vFiler.

Como você define constantes

Você pode criar e usar constantes para definir um valor, que pode ser usado em um

único fluxo de trabalho. Constantes são definidas em um nível de fluxo de trabalho.

As constantes usadas no fluxo de trabalho e seu valor são exibidas na janela de monitoramento do fluxo de trabalho durante o Planejamento e a execução. Você deve usar nomes exclusivos para constantes.

Você pode usar as seguintes convenções de nomenclatura para definir constantes:

- Maiúscula para a primeira letra de cada palavra, sem sublinhados ou espaços entre palavras

Todos os termos e abreviações devem usar maiúsculas - por exemplo, `ActualVolumeSizeInMB`.

- Maiúsculas para todas as letras

Você pode usar sublinhados para separar palavras - por exemplo,
`AGGREGATE_USED_SPACE_THRESHOLD`.

Você pode incluir os seguintes valores como para constantes de fluxo de trabalho:

- Números
- Cordas
- Expressões MVEL

As expressões são avaliadas durante as fases de Planejamento e execução dos fluxos de trabalho. Nas expressões, você não deve referenciar variáveis definidas em um loop.

- Entradas do utilizador
- Variáveis

Como a repetição de linha funciona

Um fluxo de trabalho contém comandos e detalhes de comando organizados em linhas. Você pode especificar os comandos em uma linha a serem repetidos para um número fixo de iterações ou número dinâmico de iterações com base nos resultados dos critérios de pesquisa.

Os detalhes do comando em uma linha podem ser especificados para repetir um determinado número de vezes ou quando o fluxo de trabalho é projetado. O fluxo de trabalho também pode ser projetado de modo que o número de vezes que a linha deve repetir pode ser especificado quando o fluxo de trabalho é executado ou programado para uma execução. Você pode especificar critérios de pesquisa para um objeto e os comandos em uma linha podem ser definidos para repetir quantas vezes os objetos forem retornados pelos critérios de pesquisa. As linhas também podem ser definidas para repetir quando determinadas condições forem atendidas.

Variáveis de repetição de linhas

Você pode especificar variáveis na lista de variáveis que podem ser manipuladas durante as iterações de linha. Para as variáveis, você pode especificar um nome, um valor com o qual as variáveis são inicializadas e uma expressão MVFLEX Expression Language (MVEL) que é avaliada após cada iteração da repetição de linha.

A ilustração a seguir mostra as opções de repetição de linha e um exemplo de uma variável de repetição de linha:

Row Repetition Details [?](#) X

Repeats*	Number of times	
Number of Times*	Number of times	
	For every resource in a group	
Index Variable*	Index1	
Variables	Name	Initial Value
	size_to_alloc	SIZE_MB
		(int)size_to_allocated - getData()

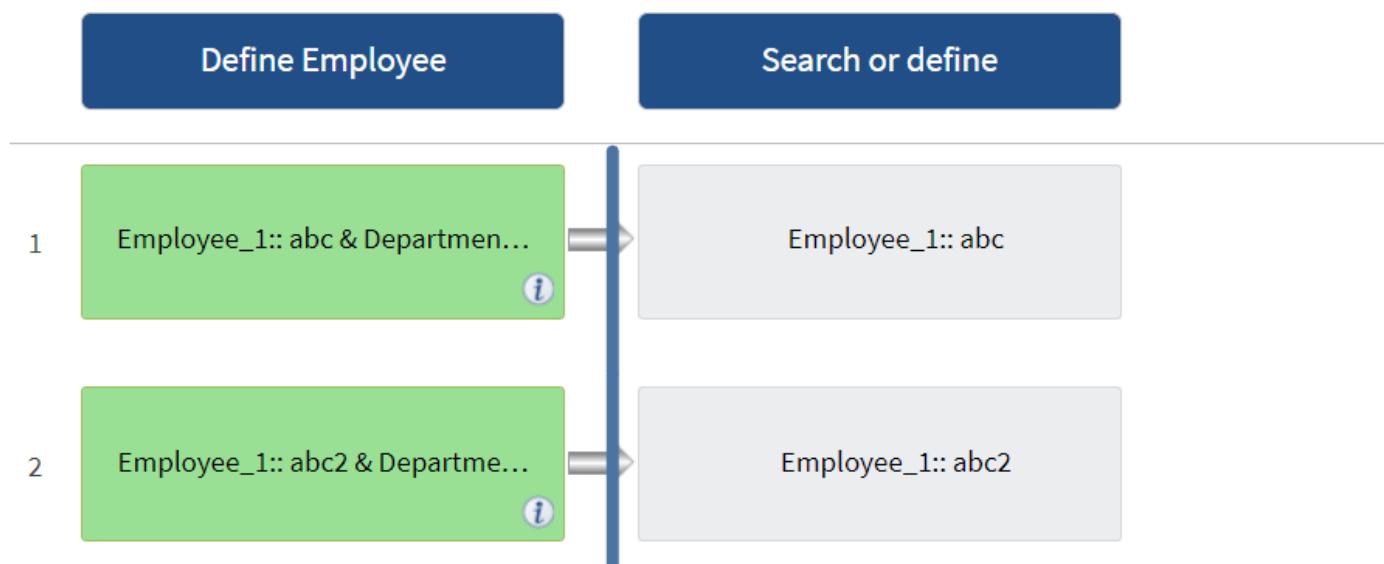
[Add](#) [Remove](#)

[Cancel](#) [OK](#)

Repetição de linhas com pontos de aprovação

Quando tiver especificado iterações de linhas repetidas para comandos e incluído pontos de aprovação, todas as iterações dos comandos antes de um ponto de aprovação são executadas. Depois de aprovar o ponto de aprovação, a execução de todas as iterações dos comandos sucessivos continua até o próximo ponto de aprovação.

A ilustração a seguir mostra como as iterações de linhas repetidas são executadas quando um ponto de aprovação é incluído em um fluxo de trabalho:



Repete exemplos de linhas em fluxos de trabalho predefinidos

Você pode abrir os seguintes fluxos de trabalho predefinidos no Designer para entender como as linhas de repetição são usadas:

- Criar um volume de NFS Data ONTAP em cluster
- Crie o armazenamento de dados do VMware NFS no storage Data ONTAP em cluster
- Estabelecer peering de cluster
- Remover um volume Data ONTAP agrupado

Como a seleção de recursos funciona

O OnCommand Workflow Automation (WFA) usa algoritmos de pesquisa para selecionar recursos de armazenamento para execução de fluxo de trabalho. Você deve entender como a seleção de recursos funciona para projetar fluxos de trabalho de forma eficiente.

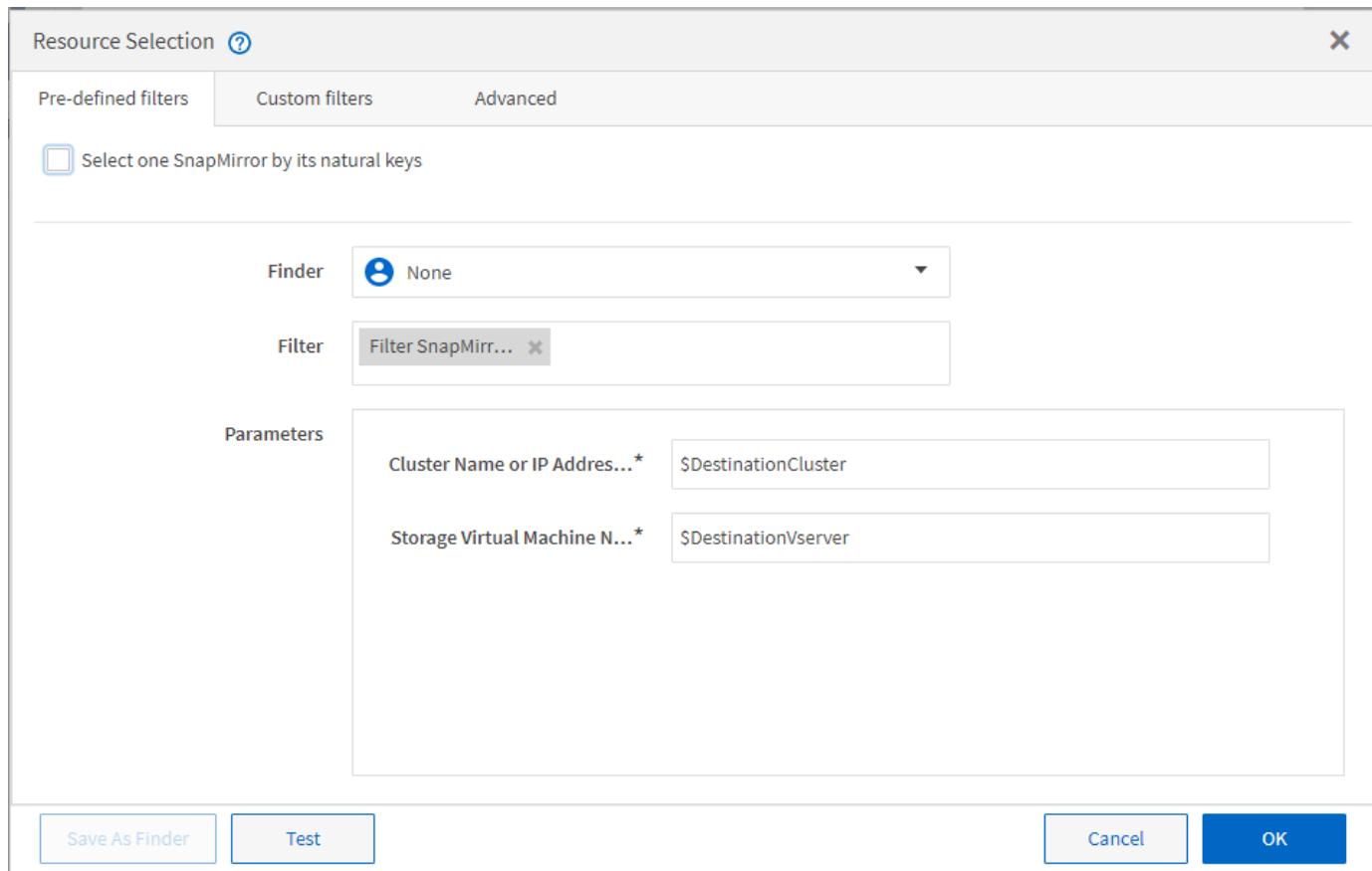
WFA seleciona recursos de entrada de dicionário - como unidades do vFiler, agregados e máquinas virtuais - usando algoritmos de pesquisa. Os recursos selecionados são então usados para executar o fluxo de trabalho. Os algoritmos de busca DO WFA fazem parte dos blocos de construção DO WFA, e incluem finders e filtros. Para localizar e selecionar os recursos necessários, os algoritmos de pesquisa pesquisam os dados armazenados em cache de diferentes repositórios, como Active IQ Unified Manager, VMware vCenter Server e um banco de dados. Por padrão, um filtro está disponível para cada entrada de dicionário para procurar um recurso com base em suas chaves naturais.

Você deve definir os critérios de seleção de recursos para cada comando em seu fluxo de trabalho. Além disso, você pode usar um localizador para definir os critérios de seleção de recursos em cada linha do fluxo de trabalho. Por exemplo, quando você está criando um volume que requer uma quantidade específica de espaço de armazenamento, você pode usar o localizador ""Localizar agregado por capacidade disponível"" no comando ""criar volume""", que seleciona um agregado com uma quantidade específica de espaço disponível e cria o volume nele.

Você pode definir um conjunto de regras de filtro para recursos de entrada de dicionário, como unidades do vFiler, agregados e máquinas virtuais. As regras de filtro podem conter um ou mais grupos de regras. Uma regra consiste em um atributo de entrada de dicionário, um operador e um valor. O atributo também pode incluir atributos de suas referências. Por exemplo, você pode especificar uma regra para agregados da seguinte forma: Listar todos os agregados que têm nomes começando com a cadeia "aggr" e ter mais de 5 GB de espaço disponível. A primeira regra no grupo é o atributo "nome", com o operador "starts-with", e o valor "agir". A segunda regra para o mesmo grupo é o atributo "available_size_mb", com o operador ">", e o valor "5000". Você pode definir um conjunto de regras de filtro juntamente com filtros públicos. A opção Definir regras de filtro será desativada se tiver selecionado um localizador. A opção Salvar como Finder será desativada se você tiver selecionado a caixa de seleção Definir regras de filtro.

Além dos filtros e localizadores, você pode usar um comando de pesquisa ou definição para procurar recursos disponíveis. O comando search ou define é a opção preferida sobre os comandos no-op. O comando pesquisar e definir pode ser usado para definir recursos do tipo de entrada do dicionário certificado e do tipo de entrada do dicionário personalizado. O comando pesquisar ou definir procura recursos, mas não executa nenhuma ação no recurso. No entanto, quando um localizador é usado para procurar recursos, ele é usado no contexto de um comando, e as ações definidas pelo comando são executadas nos recursos. Os recursos retornados por um comando de pesquisa ou definição são usados como variáveis para os outros comandos no fluxo de trabalho.

A ilustração a seguir mostra que um filtro é usado para a seleção de recursos:



Exemplos de seleção de recursos em fluxos de trabalho predefinidos

Você pode abrir os detalhes do comando dos seguintes fluxos de trabalho predefinidos no Designer para entender como as opções de seleção de recursos são usadas:

- Criar um volume de NFS Data ONTAP em cluster
- Estabelecer peering de cluster
- Remover um volume Data ONTAP agrupado

Como a reserva funciona

A funcionalidade de reserva de recursos do OnCommand Workflow Automation reserva os recursos necessários para garantir que os recursos estejam disponíveis para a execução bem-sucedida de fluxos de trabalho.

Os comandos WFA podem reservar os recursos necessários e remover a reserva depois que o recurso estiver disponível no banco de dados de cache WFA, normalmente após uma aquisição de cache. O recurso de reserva garante que os recursos reservados estejam disponíveis para o fluxo de trabalho até o período de expiração da reserva que você configurou nas configurações DO WFA.

Você pode usar o recurso de reserva para excluir recursos reservados por outros fluxos de trabalho durante a seleção de recursos. Por exemplo, se um fluxo de trabalho que reservou 100 GB de espaço em um agregado for programado para execução após uma semana, e você estiver executando outro fluxo de trabalho que usa o comando **Create volume**, o fluxo de trabalho que está sendo executado não consome o espaço reservado pelo fluxo de trabalho agendado para criar um novo volume. Além disso, o recurso de reserva permite que fluxos de trabalho sejam executados em paralelo.

Ao visualizar um fluxo de trabalho para execução, o planejador WFA considera todos os objetos reservados, incluindo os objetos existentes no banco de dados de cache. Se você ativou a reserva, os efeitos dos fluxos de trabalho programados e os fluxos de trabalho que estão sendo executados em paralelo e a existência de elementos de armazenamento serão considerados ao Planejar o fluxo de trabalho.

A seta na ilustração a seguir mostra que a reserva está ativada para o fluxo de trabalho:

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					

Exemplos de reservas em fluxos de trabalho predefinidos

Você pode abrir os seguintes fluxos de trabalho predefinidos no Designer para entender como a reserva é usada:

- Ambiente clone
- Criar um volume Data ONTAP agrupado
- Estabelecer peering de cluster
- Remover um volume Data ONTAP agrupado

Qual é a nomenclatura incremental

Nomenclatura incremental é um algoritmo que permite nomear os atributos em um fluxo de trabalho com base nos resultados de pesquisa de um parâmetro. Você pode nomear os atributos com base em um valor incremental ou uma expressão personalizada. A funcionalidade de nomenclatura incremental ajuda você a implementar uma convenção de nomenclatura com base em suas necessidades.

Você pode usar a funcionalidade de nomenclatura incremental ao projetar fluxos de trabalho para nomear dinamicamente os objetos criados pelo fluxo de trabalho. A funcionalidade permite especificar critérios de

pesquisa para um objeto usando o recurso de seleção de recursos e o valor retornado pelos critérios de pesquisa é usado para o atributo do objeto. Além disso, você pode especificar um valor para o atributo se nenhum objeto foi encontrado com os critérios de pesquisa especificados.

Você pode usar uma das seguintes opções para nomear os atributos:

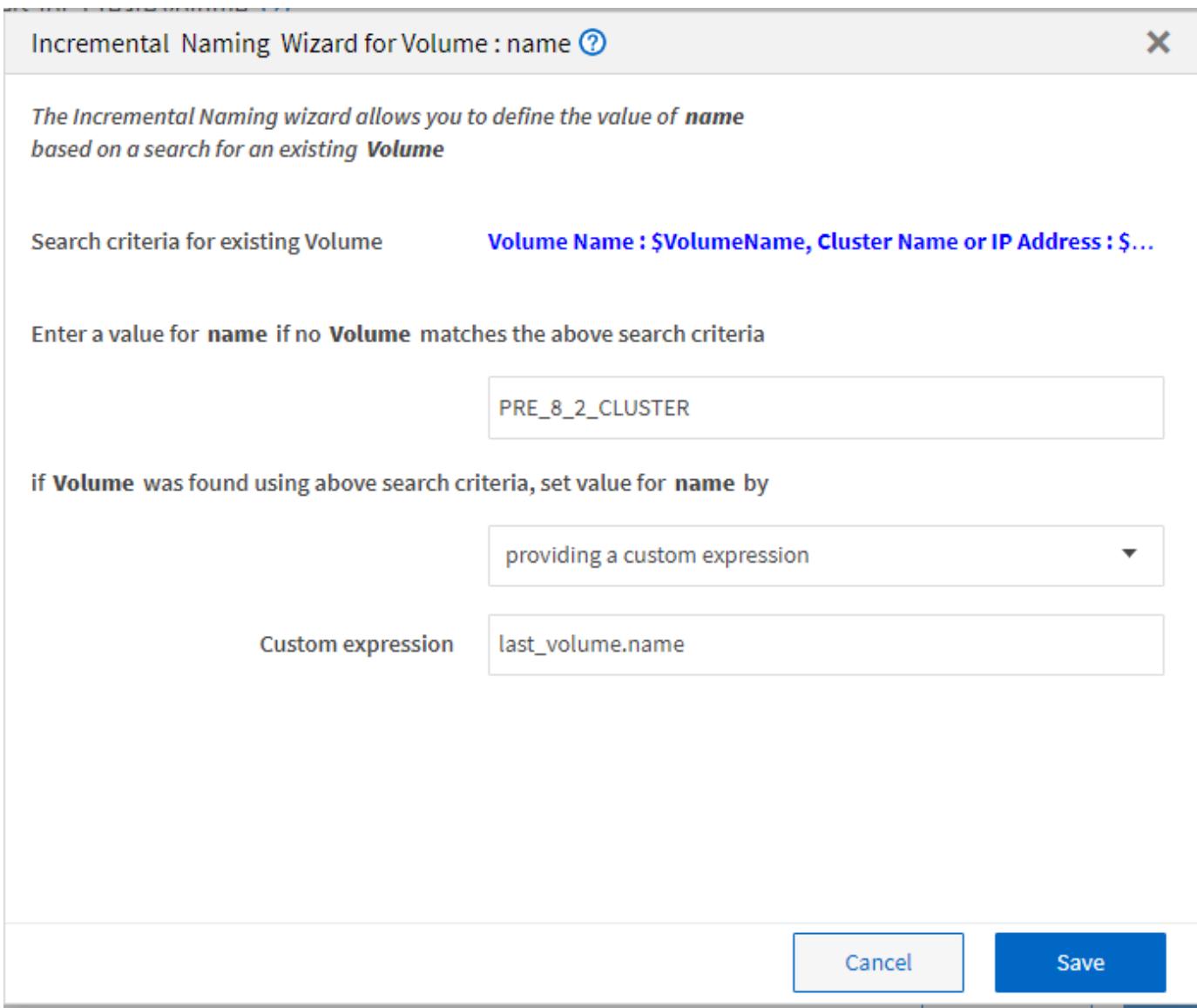
- Fornecendo um valor de incremento e sufixo

Você pode fornecer um valor que deve ser usado juntamente com o valor do objeto encontrado pelos critérios de pesquisa e incremento com o número especificado. Por exemplo, se você quiser criar volumes com a convenção de nomenclatura de `file_name_unique_number_environment`, você pode usar um localizador para encontrar o último volume pelo prefixo de nome e incrementar o número exclusivo por 1, bem como adicionar o nome do sufixo ao nome do volume. Se o último prefixo do nome do volume encontrado foi `vf_023_prod` e você está criando três volumes, os nomes dos volumes criados são `vf_024_prod`, `vf_025_prod` e `VF_026_prod`.

- Fornecendo uma expressão personalizada

Você pode fornecer um valor que deve ser usado juntamente com o valor do objeto encontrado pelos critérios de pesquisa e adicionar valores adicionais com base na expressão inserida. Por exemplo, se você quiser criar um volume com a convenção de nomenclatura `last_volume_name_environment_name_padded_with_1`, você pode digitar a expressão `last_volume.name + ' ' + nextName("lab1")`. Se o último nome do volume encontrado for `VF_023`, o nome do volume criado será `vf_023_lab2`.

A ilustração a seguir mostra como uma expressão personalizada pode ser fornecida para especificar uma convenção de nomenclatura:



Qual é a execução condicional

A execução condicional ajuda você a projetar fluxos de trabalho que podem executar comandos quando condições específicas são atendidas.

A execução de comandos em um fluxo de trabalho pode ser dinâmica. Você pode especificar uma condição para a execução de cada comando ou uma linha de comandos em seu fluxo de trabalho. Por exemplo, você pode querer que o comando ""Adicionar volume ao conjunto de dados"" seja executado somente quando um conjunto de dados específico for encontrado e você não quiser que o fluxo de trabalho falhe se o conjunto de dados não for encontrado. Nesse caso, você pode habilitar o comando ""Adicionar volume ao conjunto de dados"" para pesquisar um conjunto de dados específico e, se ele não for encontrado, você pode desativar o comando no fluxo de trabalho.

Opções para execução condicional de comandos estão disponíveis na Dictionary object guia e na guia Avançado da caixa de diálogo parâmetros para commands.

Você pode cancelar um fluxo de trabalho ou desativar um comando específico no fluxo de trabalho. Além disso, você pode definir um comando para ser executado usando uma das seguintes opções:

- Sem qualquer condição
- Quando as variáveis que você especificou são encontradas
- Quando as variáveis que você especificou não são encontradas
- Quando a expressão que você especificou é verdadeira

Você também pode definir um comando para esperar por um intervalo de tempo específico.

Exemplos de execução condicional em fluxos de trabalho predefinidos

Você pode abrir os detalhes do comando dos seguintes fluxos de trabalho predefinidos no Designer para entender como a execução condicional de comandos é usada:

- Crie um volume Data ONTAP básico em cluster
- Criar um volume de NFS Data ONTAP em cluster

Como os parâmetros de retorno funcionam

Parâmetros de retorno são parâmetros que estão disponíveis após a fase de Planejamento de um fluxo de trabalho. Os valores retornados por esses parâmetros são úteis na depuração de um fluxo de trabalho. Você deve entender como os parâmetros de retorno funcionam e quais parâmetros podem ser usados como parâmetros de retorno para depurar fluxos de trabalho.

Você pode designar um conjunto de parâmetros, como atributos de variáveis, expressões e valores de entrada de usuário, em um fluxo de trabalho como parâmetros de retorno. Durante a execução do fluxo de trabalho, os valores dos parâmetros designados são preenchidos na fase de Planejamento e a execução do fluxo de trabalho é iniciada. Os valores desses parâmetros são então retornados da maneira como foram calculados nessa execução específica do fluxo de trabalho. Se você quiser depurar o fluxo de trabalho, você pode se referir aos valores que foram retornados pelos parâmetros.

Você pode especificar os parâmetros de retorno necessários em um fluxo de trabalho quando quiser ver quais são os valores calculados ou selecionados para esses parâmetros. Por exemplo, ao usar a lógica de seleção de recursos para selecionar um agregado em um fluxo de trabalho, você pode especificar agregado como parâmetro de retorno para que você possa ver qual agregado foi selecionado durante o Planejamento do fluxo de trabalho.

Antes de consultar os valores dos parâmetros de retorno para depurar seu fluxo de trabalho, você deve confirmar se a execução do fluxo de trabalho está concluída. Os valores dos parâmetros de retorno são definidos para cada execução do fluxo de trabalho. Se você tiver adicionado um parâmetro de retorno após várias execuções de um fluxo de trabalho, o valor desse parâmetro estará disponível apenas para execuções após a adição do parâmetro.

Parâmetros que podem ser usados como parâmetros de retorno

Parâmetros de retorno	Exemplo
Atributos variáveis que são escalares	<code>volume1.name</code> , que é um atributo da variável "nome do volume"
Constantes	<code>MAX_VOLUME_SIZE</code>

Parâmetros de retorno	Exemplo
Entradas do utilizador	Nome exclusivo
Expressões MVEL que envolvem atributos variáveis, constantes e entradas de usuário	volume1.name+'-'+\$clusterName
O parâmetro return que um comando adiciona durante a execução	O \$volumeUUID parâmetro é adicionado como um parâmetro de retorno quando você usa a seguinte linha em um comando PowerShell: Add-WfaWorkflowParameter -Name "VolumeUUID" -Value "12345" -AddAsReturnParameter \$true.

Exemplos de parâmetros de retorno em fluxos de trabalho predefinidos

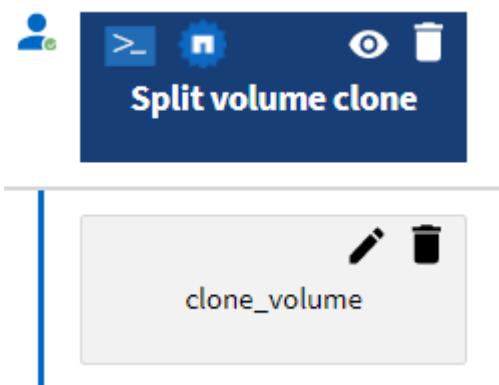
Se você quiser entender como os parâmetros de retorno são especificados, você pode abrir os seguintes fluxos de trabalho predefinidos no Designer e revisar os parâmetros de retorno especificados:

- Crie um volume NFS em um vFiler
- Crie um compartilhamento Qtree CIFS em um vFiler
- Crie um compartilhamento CIFS de volume do Data ONTAP em cluster

Quais são os pontos de aprovação

Os pontos de aprovação são pontos de verificação usados em um fluxo de trabalho para pausar a execução do fluxo de trabalho e retomá-lo com base na aprovação de um usuário.

A barra vertical azul mostrada na ilustração a seguir é um ponto de aprovação:



Você pode usar pontos de aprovação para execução incremental de um fluxo de trabalho, onde seções do fluxo de trabalho devem ser executadas somente após uma determinada condição ser atendida. Por exemplo, quando a próxima seção tem que ser aprovada ou quando a execução bem-sucedida da primeira seção é validada. Os pontos de aprovação não lidam com nenhum processo entre pausar e retomar um fluxo de trabalho. As notificações de e-mail e SNMP são enviadas, conforme especificado na configuração DO WFA, e o operador de armazenamento pode ser solicitado a executar determinadas ações ao receber a notificação de pausa do fluxo de trabalho. Por exemplo, o operador de storage pode enviar detalhes de Planejamento para o

administrador, aprovador ou operador para aprovação e retomar o fluxo de trabalho quando a aprovação for recebida.

As aprovações podem não ser necessárias em todos os momentos. Em alguns cenários, a aprovação pode ser necessária apenas se uma condição específica for atendida e as condições puderem ser configuradas quando um ponto de aprovação for adicionado. Por exemplo, considere um fluxo de trabalho projetado para aumentar o tamanho de um volume. Você pode adicionar um ponto de aprovação no início do fluxo de trabalho para que o operador de armazenamento obtenha aprovação dos gerentes quando o aumento no tamanho do volume resultar em um uso de 85% do espaço no agregado que contém o volume. Durante a execução do fluxo de trabalho e ao selecionar um volume que resulte nessa condição, a execução é interrompida até que seja aprovada.

A condição configurada para o ponto de aprovação pode ter uma das seguintes opções:

- Sem qualquer condição
- Quando a variável que você especificou é encontrada
- Quando a variável especificada não for encontrada
- Quando a expressão especificada for avaliada como verdadeira

Não há limitação no número de pontos de aprovação em um fluxo de trabalho. Você pode inserir pontos de aprovação antes de comandos em um fluxo de trabalho e definir os comandos após o ponto de aprovação para aguardar aprovação antes da execução. Os pontos de aprovação fornecem informações, como o tempo de mudança, o usuário e os comentários, permitindo que você veja quando e por que a execução do fluxo de trabalho foi pausada ou retomada. Os comentários do ponto de aprovação podem incluir expressões MVEL.

Exemplos de pontos de aprovação em fluxos de trabalho predefinidos

Você pode abrir os seguintes fluxos de trabalho predefinidos no Designer para entender como os pontos de aprovação são usados:

- Remover um volume Data ONTAP agrupado
- Atualização de controladora e compartimento de um par de HA
- Migrar volumes

Como você executa pontos finais DE DESCANSO personalizados

O OnCommand Workflow Automation (WFA) fornece um mecanismo para configurar os pontos finais REST personalizados para executar os fluxos de trabalho. Os pontos finais REST personalizados ajudam um arquiteto a configurar identificadores de recursos (URIs) fáceis de entender, intuitivos e uniformes para executar fluxos de trabalho, que seguem as CONVENÇÕES REST de POST, PUT ou DELETE com base na semântica do fluxo de trabalho. Esses URIs facilitam o desenvolvimento de código de cliente para desenvolvedores de clientes.

O WFA permite configurar um caminho de URI personalizado para execução de fluxo de trabalho através das chamadas de API. Cada segmento no caminho URI pode ser uma cadeia de caracteres ou um nome válido da entrada do usuário do fluxo de trabalho entre parênteses, por exemplo, /devops/{ProjectName}/clone. o fluxo de trabalho pode ser invocado como uma chamada para `https://WFAServer: HTTPS_PORT/rest/devops/Project1/clone/jobs`.

A validação para o caminho URI é a seguinte:

- O caminho DE REPOUSO deve começar com ""/"".
- Os caracteres permitidos são alfabetos, dígitos e sublinhado.
- O nome de entrada do utilizador deve estar rodeado por "".



Você deve verificar se o valor cercado por "" é um nome de entrada de usuário válido.

- Não deve haver segmentos de caminho vazios, por exemplo, //, //, e assim por diante.
- A configuração do método HTTP e a configuração do caminho URI personalizado devem ser configuradas ou não configuradas.

Como continuar na falha funciona

O recurso continuar com falha ajuda a configurar uma etapa em um fluxo de trabalho para que a execução do fluxo de trabalho possa continuar mesmo que a etapa falhe. Você pode resolver as etapas com falha e resolver o problema que causou a falha da etapa acessando o `wfa.log` arquivo ou clicando no ícone.

Um fluxo de trabalho que tenha uma ou mais dessas etapas falhadas está no estado parcialmente bem-sucedido após a conclusão da execução. Você pode configurar uma etapa para que a execução do fluxo de trabalho continue mesmo que a etapa falhe, selecionando a opção necessária na guia Avançado da caixa de diálogo parâmetros para <command_name>.

Se uma etapa não estiver configurada para continuar com a falha, a execução do fluxo de trabalho será abortada se a etapa falhar.

Se uma etapa configurada para continuar em falha falhar, você pode definir o fluxo de trabalho a ser executado usando uma das seguintes opções:

- Abortar a execução do fluxo de trabalho (opção padrão)
- Continue a execução a partir da próxima etapa
- Continue a execução a partir da próxima linha

Exemplo de lista de verificação de requisitos de fluxo de trabalho

Uma lista de verificação de requisitos de fluxo de trabalho inclui requisitos detalhados - como comandos, entrada de usuário e recursos - para um fluxo de trabalho planejado. Use a lista de verificação para Planejar seus fluxos de trabalho e identificar as lacunas nos requisitos.

Exemplo de lista de verificação de requisitos

A lista de verificação de requisitos de fluxo de trabalho de exemplo a seguir lista os requisitos para o fluxo de trabalho ""criar um volume Data ONTAP em cluster"". Você pode usar esta lista de verificação de exemplo como um modelo para listar seus requisitos de fluxo de trabalho.

Requisito	Descrição
Nome do fluxo de trabalho	Criar um volume Data ONTAP agrupado

Requisito	Descrição
Categoria	Provisionamento de storage
Descrição	O fluxo de trabalho cria um novo volume em uma SVM específica. Este fluxo de trabalho destina-se a um cenário em que um volume é provisionado e delegado para uso posterior.
Descrição de alto nível de como o fluxo de trabalho funciona	<ul style="list-style-type: none"> • O SVM que contém o volume é especificado pelo usuário (cluster, nomes SVM). • Um volume é criado com base no tamanho especificado. • A configuração do volume é descrita em um modelo.

Requisito	Descrição
Detalhes	<ul style="list-style-type: none"> • Use o comando Create cm volume • Detalhes do comando para Create cm volume: <ul style="list-style-type: none"> ◦ A execução é definida como sempre ◦ Os detalhes do volume são especificados preenchendo atributos para o volume ◦ Use o modelo Space Guaranteed Settings para configurar o volume ◦ O nome e o tamanho do volume são fornecidos pelo usuário. <p>O volume será montado no namespace SVM como /volname (sob o namespace raiz).</p> <ul style="list-style-type: none"> ◦ Use a função actualVolumeSize porque a reserva de snap será de 5%. ◦ A referência SVM é definida com a seguinte lógica de seleção de recursos: <ul style="list-style-type: none"> ▪ CM SVM by key — procura SVM por nome e cluster, que é fornecido pelo usuário ▪ CM SVM por tipo — somente dados SVMs (tipo: Cluster) ▪ SVM por estado — (estado em execução) ◦ A referência agregada é definida com a lógica de seleção de recursos como um localizador predefinido (CM agregado por limites de espaço e tipo RAID): <ul style="list-style-type: none"> ▪ CM agregado por capacidade disponível (capacidade: Tamanho do volume a ser provisionado, cluster dado pelo usuário) ▪ Agregado de CM por delegação à SVM ▪ Agregado CM por tipo RAID (RAID-DP) ▪ Agregado CM não aggr0 ▪ CM agregado por % de tamanho utilizado (limiar de 90, espaço aToBeProvisioned - tamanho fornecido, uma vez que a garantia é volume) ▪ CM agregado por excesso de compromisso (limite de 300, spaceToBeAllocated: Tamanho do volume que está sendo provisionado) ▪ Selecione o agregado com espaço livre máximo

Nome	Tipo	Descrição (valores de dados, validação e assim por diante)
Cluster	Consulta bloqueada (tabular)	<ul style="list-style-type: none"> • Cluster que hospeda o SVM • A consulta pode ser exibida em tabular com o endereço principal e o nome do cluster • Ordenar alfabeticamente por nome
SVM	Consulta bloqueada	<ul style="list-style-type: none"> • SVM no qual o volume é provisionado • A consulta deve exibir somente nomes SVM pertencentes ao cluster escolhido na entrada anterior <p>Mostrar somente SVMs do tipo cluster, não admin ou node (coluna do tipo cm_storage.vserver)</p> <ul style="list-style-type: none"> • Ordenar alfabeticamente
Volume	Cadeia de caracteres	<ul style="list-style-type: none"> • Nome do volume a ser criado
Tamanho em GB	Número inteiro	<ul style="list-style-type: none"> • Tamanho do volume a ser provisionado • Tamanho dos dados (a reserva instantânea deve ser considerada)

Comandos

Nome	Descrição	Estado
Criar CM volume	Cria um volume na SVM	Existente

Parâmetros de retorno

Nome	Valor
Nome do volume	Nome do volume provisionado
Nome agregado	Nome do agregado selecionado
Nome do nó	Nome do nó

Nome	Valor
Nome do cluster	Nome do cluster

Lacunas e problemas

1.	
2.	
3.	
4.	
5.	

Crie um fluxo de trabalho

Você pode usar o Workflow Automation (WFA) para criar fluxos de trabalho para tarefas como provisionamento, migração e desativação de storage para bancos de dados ou sistemas de arquivos. Você deve criar fluxos de trabalho quando os fluxos de trabalho predefinidos DO WFA não corresponderem aos seus requisitos.

O que você vai precisar

- Você deve ter entendido os conceitos para blocos de construção WFA.
- Você deve ter entendido recursos como repetição de linha, pontos de aprovação e seleção de recursos necessários para o fluxo de trabalho.
- Você precisa ter concluído o Planejamento necessário para o fluxo de trabalho, incluindo a lista de verificação de requisitos de fluxo de trabalho.
- Você deve ter criado o conteúdo de ajuda, que fornece informações sobre o fluxo de trabalho para os operadores de armazenamento.

Sobre esta tarefa

A construção de cada fluxo de trabalho pode variar com base na meta e exigência do fluxo de trabalho. Esta tarefa não fornece instruções para um fluxo de trabalho específico, mas fornece instruções gerais para criar um fluxo de trabalho.

Passos

1. Clique em **Workflow Design > fluxos de trabalho**.
2. Clique  na barra de ferramentas.
3. Na guia **Workflow**, execute as seguintes etapas:
 - a. Expanda o esquema necessário e clique duas vezes no necessário  (comando) ou  (fluxo de trabalho) na lista **passos disponíveis**.

Pode repetir este passo conforme necessário. Você pode arrastar e soltar etapas para reorganizar as etapas no editor de fluxo de trabalho.

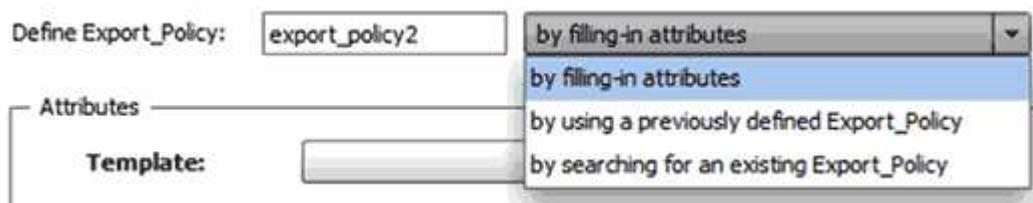
- b. **Opcional:** clique  para adicionar o número necessário de linhas, que são usadas para especificar detalhes para execução de etapas.

Cada etapa é executada com base nos detalhes da etapa especificada na linha e coluna especificadas. As etapas são executadas da esquerda para a direita e na ordem superior para a inferior.

- c. Posicione o cursor abaixo da etapa que você adicionou e clique  para adicionar detalhes da etapa para a execução da etapa, na linha desejada.

Para esta etapa...	Faça isso...
Fluxo de trabalho	Introduza as entradas de utilizador necessárias no separador fluxo de trabalho e a condição necessária no separador Avançado .
Comando	Na guia parâmetros para <command>, clique em cada guia objeto, selecione a opção necessária para definir os atributos do objeto e, em seguida, insira os detalhes necessários na guia Avançado e na guia outros parâmetros.
PESQUISE ou defina	Selecione o objeto de entrada do dicionário que deve ser pesquisado ou definido.

A ilustração a seguir mostra as opções disponíveis para definir os atributos do objeto:



Escolha a ação apropriada:

Para...	Faça isso...
preenchendo os atributos	<p>Insira o valor dos atributos usando as seguintes opções:</p> <ul style="list-style-type: none"> • Expressões • Variáveis • Entradas do utilizador • Seleção de recursos • Nomenclatura incremental <p>Você deve posicionar o cursor sobre os campos de atributo e clicar  para usar os recursos de seleção de recursos ou nomeação incremental.</p>

Para...	Faça isso...
usando um definido anteriormente <i>object</i>	Selecione o anteriormente <i>object</i> definido na caixa antes da lista de opções.
procurando por um existente <i>object</i>	<p>i. Clique em Digite os critérios de pesquisa para pesquisar o objeto usando o recurso de seleção de recursos.</p> <p>ii. Selecione uma das opções necessárias para execução se o objeto necessário não for encontrado:</p> <ul style="list-style-type: none"> ◦ Abortar fluxo de trabalho Esta opção aborta a execução do fluxo de trabalho se o objeto específico não for encontrado. ◦ Desativar este comando <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>This option disables only the current step and executes the workflow. **** Preencha os atributos `object` e execute o comando</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. Se pretender inserir um ponto de aprovação, clique  e introduza o comentário necessário para o ponto de aprovação.

Os comentários do ponto de aprovação podem incluir expressões MVEL.

5. Clique  no que está ao lado dos números de linha para executar o seguinte:

- Insira uma linha.
- Copie a linha.
- Repita a linha.

Você pode usar uma das seguintes opções para especificar a repetição dos parâmetros do comando:

- Número de vezes

Você pode usar essa opção para repetir a execução do comando para o número de repetições que você especificar. Por exemplo, você pode especificar que o comando "create qtree" deve ser repetido três vezes para criar três qtrees.

Você também pode usar essa opção para um número dinâmico de execuções de comandos. Por exemplo, você pode criar uma variável de entrada de usuário para o número de LUNs a serem criados e usar o número especificado pelo operador de armazenamento quando o fluxo de trabalho é executado ou programado.

- Para cada recurso em um grupo

Você pode usar essa opção e, em seguida, especificar um critério de pesquisa para um objeto. O comando é repetido tantas vezes quanto o objeto é retornado pelos critérios de pesquisa. Por exemplo, você pode procurar os nós em um cluster e repetir o comando ""Create iSCSI Logical Interface"" para cada nó.

- Adicione uma condição para execução da linha.
- Retire a linha.

6. Na guia **Detalhes**, execute as seguintes etapas:

- a. Especifique as informações necessárias nos campos **Nome do fluxo de trabalho** e **Descrição do fluxo de trabalho**.

O nome e a descrição do fluxo de trabalho devem ser exclusivos para cada fluxo de trabalho.

- b. * Opcional: * Especifique a versão da entidade.
- c. **Opcional:** desmarque a caixa de seleção **considerar elementos reservados** se você não quiser usar a capacidade de reserva.
- d. **Opcional:** desmarque a caixa de seleção **Ativar validação de existência de elementos** se não quiser habilitar a validação para elementos que existem com o mesmo nome.

7. Se pretender editar as entradas do utilizador, execute as seguintes etapas:

- a. Clique no separador **User Inputs** (entradas do utilizador).
- b. Clique duas vezes na entrada do usuário que você deseja editar.
- c. Na caixa de diálogo **Editar variável: <user input>**, edite a entrada do usuário.

8. Se você quiser adicionar constantes, execute as etapas a seguir

- a. Clique na guia **constantes** e, em seguida, adicione as constantes necessárias para o seu fluxo de trabalho usando o botão **Adicionar**.

Você pode definir constantes quando estiver usando um valor comum para definir os parâmetros para vários comandos. Por exemplo, veja a constante AGGREGATE_OVERFLOW_THRESHOLD usada no fluxo de trabalho "criar, mapear e proteger LUNs com SnapVault".

- b. Introduza o nome, a descrição e o valor de cada constante.

9. Clique na guia **Return Parameters** e, em seguida, adicione os parâmetros necessários para o seu fluxo de trabalho usando o botão **Add**.

Você pode usar parâmetros de retorno quando o Planejamento e a execução do fluxo de trabalho devem retornar alguns valores calculados ou selecionados durante o Planejamento. Você pode exibir os valores calculados ou selecionados na guia parâmetros de retorno da janela de monitoramento na visualização do fluxo de trabalho ou após a conclusão da execução do fluxo de trabalho.

Agregado: Você pode especificar o agregado como um parâmetro de retorno para ver qual agregado foi selecionado usando a lógica de seleção de recursos.

Se você incluiu um fluxo de trabalho filho no fluxo de trabalho e se os nomes de parâmetros de retorno do fluxo de trabalho filho contiverem um espaço, sinal de dólar ou uma função, especifique o nome do parâmetro de retorno entre colchetes no fluxo de trabalho pai para exibir o valor do parâmetro de retorno do fluxo de trabalho filho no fluxo de trabalho pai.

Se o nome do parâmetro for...	Especificar como...
ChildWorkflow1.abc\$value	ChildWorkflow1["abc\$"+ "value"]
ChildWorkflow1.\$value	ChildWorkflow1["\$"+ "value"]
ChildWorkflow1.value\$	ChildWorkflow1.value\$
ChildWorkflow1.P_N	ChildWorkflow1["P_N"]
ChildWorkflow1.return_string("HW")	ChildWorkflow1["return_string(\"HW\")"]

10. **Opcional:** clique na guia **conteúdo da Ajuda** para adicionar o arquivo de conteúdo de ajuda que você criou para o fluxo de trabalho.
11. Clique em **Preview** e certifique-se de que o Planejamento do fluxo de trabalho seja concluído com êxito.
12. Clique em **OK** para fechar a janela de visualização.
13. Clique em **Salvar**.

Depois de terminar

Teste o fluxo de trabalho em seu ambiente de teste e marque o fluxo de trabalho como pronto para produção em **WorkflowName > Details**.

Criar conteúdo de ajuda de fluxo de trabalho

Administradores e arquitetos do OnCommand Workflow Automation que projetam fluxos de trabalho podem criar conteúdo de ajuda para os fluxos de trabalho e incluí-lo no fluxo de trabalho.

O que você vai precisar

Você deve estar ciente de como criar páginas da Web usando HTML.

Sobre esta tarefa

A ajuda deve fornecer informações sobre o fluxo de trabalho e as entradas do usuário para o fluxo de trabalho para o operador de armazenamento que executa o fluxo de trabalho.

Passos

1. Crie uma pasta com o seguinte nome: Workflow-help.
2. Crie o conteúdo de ajuda usando um editor HTML ou um editor de texto e salve-o como um index.htm

arquivo na workflow-help pasta.

Você não deve incluir arquivos JavaScript como parte do conteúdo da ajuda. A seguir estão as extensões de arquivo suportadas:

- .jpg
- .jpeg
- .gif
- .png
- .xml
- .thmx
- .htm
- .html
- .css

Você também pode incluir o Thumbs.db arquivo, que é criado pelo Windows.

3. Verifique se o index.htm arquivo e outros arquivos associados ao conteúdo da ajuda, como imagens, estão disponíveis na workflow-help pasta.
4. Crie um .zip ficheiro da pasta e certifique-se de que o tamanho do .zip ficheiro não seja superior a 2 MB.

Create an NFS volume-help.zip

5. Edite o fluxo de trabalho para o qual você criou o conteúdo de ajuda e clique em **Setup > Help Content > Browse** para carregar o .zip arquivo.

Crie pacotes de fluxo de trabalho WFA

Você pode criar pacotes de fluxo de trabalho no OnCommand Workflow Automation (WFA) para seus requisitos de integração e automação de storage.

Passos

1. Faça login na janela DO WFA através de um navegador da Web.
2. Clique em **Gerenciamento de conteúdo > Pacotes**.
3. Clique no ícone **New Pack**.
4. Na caixa de diálogo **New Pack**, insira valores para os campos **Name**, **Author**, **Version** e **Description**.
5. Clique em **Salvar**.
6. Verifique se o novo pacote foi criado na janela **Packs**.

Adicione entidades aos pacotes de fluxo de trabalho DO WFA

Você pode adicionar uma ou mais entidades a um workflow pack no OnCommand Workflow Automation (WFA) para seus requisitos de integração e automação de storage.

Sobre esta tarefa

Você pode remover um pacote das seguintes entidades:

- Fluxo de trabalho
- Localizadores
- Filtros
- Comandos
- Funções
- Modelos
- Esquemas
- Dicionário
- Tipos de fontes de dados
- Tipos de sistema remoto
- Consultas de cache
- Categorias

Passos

1. Faça login na janela **WFA** através de um navegador da Web.
2. No portal **Workflow design**, navegue até a entidade que deseja adicionar e clique em **<Entity>**.
3. Na janela **Entity**, selecione a entidade que deseja adicionar ao pacote.
4. Clique no ícone **Adicionar ao pacote**.

"Add to Pack" é ativado apenas para entidades para as quais a certificação está definida como **None**.

5. Na caixa de diálogo **Add to Pack <Entity>**, na lista suspensa **Available Packs**, selecione o pacote ao qual deseja adicionar a entidade.
6. Clique em **OK**.

Eliminar pacotes OnCommand Workflow Automation

Você pode excluir um pacote do OnCommand Workflow Automation (WFA) se você não precisar mais dele. Se você excluir um pacote, todas as entidades associadas ao pacote serão excluídas.

Sobre esta tarefa

- Você não pode excluir um pacote se houver dependências nas entidades que fazem parte do pacote.

Por exemplo, se você tentar excluir um pacote que inclui um comando que faz parte de um fluxo de trabalho personalizado, a operação de exclusão falhará porque o fluxo de trabalho personalizado depende do pacote. Só pode eliminar o pacote depois de eliminar o fluxo de trabalho personalizado.

- As entidades que fazem parte de um pacote não podem ser eliminadas individualmente.

Para excluir uma entidade que faz parte de um pacote, você deve excluir o pacote que contém essa entidade. Se uma entidade for parte de vários pacotes, a entidade será excluída do servidor WFA somente quando todos os pacotes que contêm essa entidade forem excluídos.

Passos

1. Faça login NO WFA através de um navegador da Web como administrador.
2. Clique em **Gerenciamento de conteúdo > Pacotes**.
3. Selecione o pacote que pretende eliminar e clique  em .
4. Na caixa de diálogo de confirmação **Delete Pack**, clique em **OK**.

Exportar conteúdo OnCommand Workflow Automation

Você pode salvar o conteúdo do OnCommand Workflow Automation (WFA) criado pelo usuário como um **.dar** arquivo e compartilhar o conteúdo com outros usuários. O conteúdo DO WFA pode incluir todo o conteúdo criado pelo usuário ou itens específicos, como fluxos de trabalho, localizadores, comandos e termos de dicionário.

O que você vai precisar

- Você deve ter acesso ao conteúdo DO WFA que deseja exportar.
- Se o conteúdo a exportar contiver referências a conteúdo certificado, os pacotes de conteúdo certificados correspondentes devem estar disponíveis no sistema quando o conteúdo for importado.

Esses pacotes podem ser baixados do repositório de automação do storage.

Sobre esta tarefa

- Não é possível exportar os seguintes tipos de conteúdo certificado:
 -  - Conteúdo certificado pela NetApp
 -  - Conteúdo desenvolvido por Serviços profissionais (PS), que está disponível apenas em instalações personalizadas feitas pela PS
 -  - pacotes desenvolvidos pelos usuários
- Todos os objetos que dependem do objeto exportado também são exportados.

Por exemplo, exportar um fluxo de trabalho também exporta os comandos, filtros e localizadores dependentes para o fluxo de trabalho.

- Você pode exportar objetos bloqueados.

Os objetos permanecem no estado bloqueado quando são importados por outros utilizadores.

Passos

1. Faça login NO WFA através de um navegador da Web.
2. Exportar o conteúdo necessário:

Se você quiser...	Faça isso...
Exporte todo o conteúdo criado pelo usuário como um .dar único arquivo	<ol style="list-style-type: none"> Clique em Configurações e, em Manutenção, clique em Exportar todos os fluxos de trabalho. Especifique um nome de arquivo para o .dar arquivo e clique em Exportar.

Se você quiser...	Faça isso...
Exportar conteúdo específico	<p>a. Navegue até a janela a partir da qual você deseja exportar conteúdo.</p> <p>b. Selecione um ou mais itens na janela e clique  em .</p> <p>c. Na caixa de diálogo Exportar como, especifique um nome de arquivo para o .dar arquivo e clique em Exportar.</p>

3. Na caixa de diálogo **Salvar como**, especifique o local onde deseja salvar o .dar arquivo e clique em **Salvar**.

Importar conteúdo OnCommand Workflow Automation

Você pode importar conteúdo do OnCommand Workflow Automation (WFA) criado pelo usuário, como fluxos de trabalho, localizadores e comandos. Você também pode importar conteúdo que é exportado de outra instalação DO WFA, conteúdo que é baixado do repositório de automação de storage ou da COMUNIDADE WFA, bem como pacotes, incluindo kits de ferramentas do Data ONTAP PowerShell e kits de ferramentas Perl NMSDK.

O que você vai precisar

- Você deve ter acesso ao conteúdo DO WFA que deseja importar.
- O conteúdo que você deseja importar deve ter sido criado em um sistema que esteja executando a mesma versão ou uma versão anterior do WFA.

Por exemplo, se você estiver executando O WFA 2,2, não será possível importar conteúdo criado usando O WFA 3,0.

- Você pode importar conteúdo desenvolvido em versões N-2 do WFA apenas para O WFA 5,1.
- Se o .dar arquivo fizer referência a conteúdo certificado pela NetApp, os pacotes de conteúdo certificados pela NetApp devem ser importados.

Os pacotes de conteúdo certificados pela NetApp podem ser baixados do repositório de automação do storage. Você deve consultar a documentação do pacote para verificar se todos os requisitos foram atendidos.

Passos

1. Faça login NO WFA através de um navegador da Web.
2. Clique em **Configurações** e, em **Manutenção**, clique em **Importar fluxos de trabalho**.
3. Clique em **escolha Arquivo** para selecionar o .dar arquivo que deseja importar e clique em **Importar**.
4. Na caixa de diálogo **Importar sucesso**, clique em **OK**.

Importar pacotes de fluxo de trabalho WFA

Você pode importar pacotes de fluxo de trabalho do servidor para o OnCommand

Workflow Automation (WFA) para seus requisitos de integração e automação de storage.

O que você vai precisar

Você deve ter acesso ao conteúdo DO WFA no servidor que deseja importar.

Passos

1. Faça login na janela **WFA** através de um navegador da Web.
2. Clique em **Gerenciamento de conteúdo > Pacotes**.
3. Clique no ícone **Importar do servidor**.
4. Na caixa de diálogo **Importar da pasta do servidor**, no campo **localização da pasta no sistema do servidor**, insira a localização do pacote no servidor em um formato de cadeia de caracteres, por exemplo, `C:\work\packs\test`.
5. Clique em **OK**.
6. Verifique se o pacote é importado na janela **Packs**.

Considerações ao importar conteúdo do OnCommand Workflow Automation

Você deve estar ciente de certas considerações ao importar conteúdo criado pelo usuário, conteúdo que é exportado de outra instalação do OnCommand Workflow Automation (WFA) ou conteúdo baixado do repositório de automação de storage ou da COMUNIDADE WFA.

- O conteúdo DO WFA é salvo como um `.dar` arquivo e pode incluir todo o conteúdo criado pelo usuário de outro sistema ou itens específicos, como fluxos de trabalho, localizadores, comandos e termos de dicionário.
- Quando uma categoria existente é importada de um `.dar` arquivo, o conteúdo importado é mesclado com o conteúdo existente na categoria.

Por exemplo, considere que existem dois fluxos de trabalho WF1 e WF2 na Categoria A no servidor WFA. Se os fluxos de trabalho WF3 e WF4 na Categoria A forem importados para o servidor WFA, a categoria A conterá fluxos de trabalho WF1, WF2, WF3 e WF4 após a importação.

- Se o `.dar` arquivo contiver entradas de dicionário, as tabelas de cache correspondentes às entradas do dicionário serão atualizadas automaticamente.

Se as tabelas de cache não forem atualizadas automaticamente, uma mensagem de erro será registrada no `wfa.log` arquivo.

- Ao importar um `.dar` arquivo que tem uma dependência em um pacote que não está presente no servidor WFA, O WFA tenta identificar se todas as dependências nas entidades são atendidas.
 - Se uma ou mais entidades estiverem faltando ou se uma versão inferior de uma entidade for encontrada, a importação falhará e uma mensagem de erro será exibida.

A mensagem de erro fornece detalhes dos pacotes que devem ser instalados para atender às dependências.

- Se for encontrada uma versão superior de uma entidade ou se a certificação tiver sido alterada, é apresentada uma caixa de diálogo genérica sobre a incompatibilidade de versão e a importação é concluída.

Os detalhes de incompatibilidade de versão são registrados em um `wfa.log` arquivo.

- Perguntas e solicitações de suporte para o seguinte devem ser direcionadas para a COMUNIDADE WFA:
 - Qualquer conteúdo baixado da comunidade WFA
 - Conteúdo WFA personalizado que você criou
 - Conteúdo WFA que você modificou

Identificação do pacote durante a atualização

Durante o processo de atualização, o OnCommand Workflow Automation (WFA) identifica e classifica as entidades em um pacote. Se você tiver excluído qualquer entidade de um pacote antes da atualização, o pacote não será identificado durante a atualização.

Durante o processo de atualização, O WFA compara os pacotes no banco de dados com a lista de pacotes que foram lançados no repositório de automação do storage para identificar os pacotes que foram instalados antes da atualização. A identificação do pacote classifica, assim, os pacotes existentes no banco de dados.

O WFA executa os seguintes processos para identificar e classificar pacotes:

- Mantém uma lista de pacotes lançados no repositório de automação do storage para comparar e identificar os pacotes que foram instalados antes da atualização.
- Classifica as entidades em um pacote como parte da sincronização do repositório de automação do storage, se o repositório de automação do storage estiver habilitado.
- Classifica as entidades em pacotes usando a lista atualizada.

A identificação de pacotes é aplicável apenas a pacotes com certificação NetApp que foram baixados do repositório de automação de storage.

Se um pacote não for identificado durante a atualização, você pode reimportar o pacote para identificá-lo no WFA. Os arquivos `wfa.log` fornecem detalhes sobre as entidades que não foram identificadas como um pacote durante a atualização.

Integração de pacotes de fluxo de trabalho WFA com o repositório SCM

Você pode integrar o pacote OnCommand Workflow Automation (WFA) com o repositório Gerenciamento de Controle de fonte (SCM).

Você precisa ter credenciais de administrador ou arquiteto.

Ferramentas SCM como GitHub, Perforce e SVN exigem que você mapeie um diretório local para verificar o código do servidor de repositório SCM. Esse mapeamento de diretório local é chamado de localização de cliente SCM. você deve configurar o cliente SCM com um local de sistema de arquivos como uma área de cliente.

Você pode configurar o cliente SCM em um sistema de servidor WFA. Você precisa ter acesso ao sistema do servidor WFA para operações SCM

Faça check-in em um novo pacote de fluxo de trabalho para o SCM

Você pode criar um novo pacote de fluxo de trabalho com o OnCommand Workflow

Automation (WFA) e verificá-lo no Gerenciamento de Controle de origem (SCM).

O que você vai precisar

O SCM precisa ser configurado e você precisa ter credenciais de administrador ou arquiteto.

Passos

1. Faça login na janela **WFA** através de um navegador da Web.
2. Crie um novo pacote de fluxo de trabalho.

[Crie um pacote de automação de fluxo de trabalho](#)

3. Adicione entidades ao pacote que você criou.

[Adicione entidades a um pacote OnCommand Workflow Automation](#)

4. Clique no ícone **Exportar para servidor**.
5. Na caixa de diálogo Exportar para pasta de servidor, no campo **localização da pasta no sistema de servidor**, insira a localização do sistema de arquivos onde o pacote deve ser salvo no servidor que contém o cliente SCM.

Para editar ou reexportar o pacote ou o conteúdo, clique no ícone **Unlock**.

6. No local do cliente SCM, verifique o conteúdo do pacote para o servidor SCM.

Verifique uma nova versão de um pacote de fluxo de trabalho WFA

Você pode atualizar a versão de um pacote no OnCommand Workflow Automation (WFA) e, em seguida, verificar o pacote atualizado para um novo local no servidor Gerenciamento de Controle de fonte (SCM).

O que você vai precisar

O SCM precisa ser configurado e você precisa ter credenciais de administrador ou arquiteto.

Passos

1. Faça login na janela **WFA** através de um navegador da Web.
2. Clique em **Gerenciamento de conteúdo > Pacotes**.
3. Clique no ícone **Edit Pack**.
4. Na caixa de diálogo **Pacote <pack name>**, no campo **versão**, atualize a versão do pacote.
5. Clique em **Salvar**.
6. Clique no ícone **Exportar para servidor** no nível do pacote.
7. Na caixa de diálogo **Exportar para pasta de servidor**, no campo **localização da pasta no sistema de servidor**, insira uma nova localização do sistema de arquivos.

Se o pacote tiver sido guardado anteriormente na C:\p4\cdot\1.0.0 localização do sistema de ficheiros, guarde-o agora no C:\p4\cdot\2.0.0 local.

8. No local do cliente SCM, verifique o conteúdo do pacote para um novo local no servidor SCM.

Se o pacote foi salvo anteriormente //depot/wfa/packs/cdot/1.0.0 no caminho no servidor SCM,

você pode salvá-lo em outro local, como `//depot/wfa/packs/cdot/2.0.0`.

Atualizar pacotes de fluxo de trabalho WFA a partir do servidor SCM

Você pode atualizar um pacote no servidor Gerenciamento de Controle de fonte (SCM) e importar o pacote atualizado para o OnCommand Workflow Automation (WFA).

O que você vai precisar

O SCM precisa ser configurado e você precisa ter credenciais de administrador ou arquiteto

Sobre esta tarefa

Se você fizer quaisquer alterações ou atualizações em um pacote no servidor SCM, o administrador ou arquiteto precisa resolver os conflitos, se houver, usando as ferramentas diff fornecidas pelo SCM. O WFA adapta o ficheiro diff XML para mostrar apenas as alterações relevantes.

Antes de importar o pacote, você é notificado sobre as alterações que vão para o conteúdo do pacote WFA.

Passos

1. Faça login na janela DO WFA através de um navegador da Web.
2. Importe o pacote atualizado para O WFA.

"Importar pacotes de fluxo de trabalho WFA"



Se o banco de dados WFA já contiver o mesmo pacote, o conteúdo do pacote será substituído.

Verifique os pacotes de fluxo de trabalho WFA existentes para o servidor SCM

Você pode fazer o check-in de pacotes já existentes para o servidor de gerenciamento de controle de fonte (SCM) do OnCommand Workflow Automation (WFA).

O que você vai precisar

O SCM precisa ser configurado e você precisa ter credenciais de administrador ou arquiteto.

Passos

1. Faça login na janela **WFA** através de um navegador da Web.
2. Clique em **Gerenciamento de conteúdo > Pacotes**.
3. Clique no ícone **Exportar para servidor**.
4. Na caixa de diálogo **Exportar para pasta do servidor**, no campo **localização da pasta no sistema do servidor**, insira o local da pasta do servidor onde o pacote é salvo no servidor.

Isso exporta o pacote em uma forma explodida no sistema de arquivos onde o cliente SCM é criado.

5. No local do cliente SCM, verifique o conteúdo do pacote para o servidor SCM.
6. Use as ferramentas de comparação fornecidas pelo SCM para verificar as alterações em relação à versão SCM do pacote.

Remover pacotes de fluxo de trabalho WFA de entidades

Você pode remover um pacote das entidades no OnCommand Workflow Automation (WFA) e verificar o pacote atualizado para o servidor Gerenciamento de Controle de fonte (SCM).

O que você vai precisar

O SCM precisa ser configurado e você precisa ter credenciais de administrador ou arquiteto.

Sobre esta tarefa

Você pode remover um pacote das seguintes entidades:

- Fluxo de trabalho
- Localizadores
- Filtros
- Comandos
- Funções
- Modelos
- Esquemas
- Dicionário
- Tipos de fontes de dados
- Tipos de sistema remoto
- Consultas de cache
- Categorias

Passos

1. Faça login na janela **WFA** através de um navegador da Web.
2. No portal **Workflow design**, navegue até a entidade que deseja remover e clique em **<Entity>**.
3. Clique no ícone **Remover do pacote**.
4. Na caixa de diálogo **Remover do pacote <Entity>**, selecione o pacote que deseja excluir dessa entidade.
5. Clique em **OK**.
6. Clique na guia **Packs**.
7. Clique no ícone **Exportar para servidor**.
8. Na caixa de diálogo **Exportar para pasta do servidor**, no campo **localização da pasta no sistema do servidor**, insira o local da pasta do servidor onde o pacote é salvo no servidor.

Isso exporta o pacote em forma explodida no sistema de arquivos onde o cliente SCM é criado.

9. No local do cliente SCM, verifique o conteúdo do pacote para o servidor SCM.
10. Use as ferramentas de comparação fornecidas pelo SCM para verificar as alterações em relação à versão SCM do pacote.

Reverter um pacote de fluxo de trabalho WFA para sua versão anterior no SCM

Você pode reverter um pacote para a versão anterior no Gerenciamento de Controle de fonte (SCM) e importá-lo para o OnCommand Workflow Automation (WFA).

O que você vai precisar

O SCM precisa ser configurado e você precisa ter credenciais de administrador ou arquiteto.

Passos

1. No local do cliente SCM, reverta o pacote para uma versão anterior no local do sistema de arquivos usando ferramentas SCM.

O cliente SCM é sincronizado com o número exato de mudança em que você está interessado.

2. Faça login na janela DO WFA através de um navegador da Web.
3. Importe o pacote atualizado para O WFA.

["Importar pacotes de fluxo de trabalho WFA"](#)



Isso reverte o banco de dados DO WFA para a versão anterior.

Criando componentes básicos para fluxos de trabalho

O Workflow Automation (WFA) inclui vários componentes básicos, que são usados para construir fluxos de trabalho. Você pode criar os blocos WFA Buildings que são necessários para seus fluxos de trabalho.

Crie um tipo de fonte de dados

Você deve criar um tipo de fonte de dados para habilitar a aquisição de dados de uma fonte de dados, que não está predefinida no OnCommand Workflow Automation (WFA).

O que você vai precisar

- Você deve ter criado a entrada e o esquema de dicionário necessários se você estiver criando um tipo de fonte de dados personalizado que não está predefinido no WFA.
- Você deve estar ciente do script do PowerShell para criar um tipo de fonte de dados que use o método de script.

Passos

1. Clique em **Data Source Design > Data Source Types**.
2. Clique na barra de ferramentas.
3. Na caixa de diálogo **novo tipo de fonte de dados**, insira ou selecione os detalhes necessários nos campos **fonte de dados**, **versão da fonte de dados** e **Esquema**.
4. No campo **default port** (porta padrão), insira o número da porta.

2638

O número da porta que introduziu é preenchido quando adiciona este tipo de fonte de dados para

aquisição de dados. Por padrão, a porta é usada pelo WFA para se comunicar com a fonte de dados e a porta deve estar aberta no servidor de origem de dados.

5. Na lista **método**, selecione o método que O WFA deve usar para adquirir dados:

Se você selecionou...	Então...
SQL	<p>Na lista tipo de driver, selecione um dos seguintes drivers apropriados para a fonte de dados:</p> <ul style="list-style-type: none">• Conector MySQL/J• Driver JDBC do servidor MySQL 3,0• * Oracle JDBC Driver 11,2,0,3*
SCRIPT	<p>No campo Script, insira o script do PowerShell que é usado para conectar e recuperar dados da fonte de dados.</p> <p> Os dados na entrada do dicionário arquivos CSV equivalentes devem incluir guias como separador de campo. Por exemplo, consulte o script do PowerShell para o tipo de fonte de dados do VMware vCenter.</p>

6. Clique em **Salvar**.

Crie um comando

Você pode criar um comando WFA para concluir uma tarefa específica em seu fluxo de trabalho se não houver um comando WFA predefinido adequado para a tarefa.

O que você vai precisar

Você deve saber como escrever o código necessário para o comando usando PowerShell ou Perl.

Passos

1. Clique em **Workflow Design > comandos**.
2. Clique  na barra de ferramentas.
3. Na guia **Propriedades** da caixa de diálogo **Nova Definição de comando**, insira ou selecione os detalhes necessários nos campos **Nome**, **Descrição** e **tempo limite**.
 - a. No campo **String representation**, insira uma representação de string para o comando usando a sintaxe MVEL.

VolumeName + ">" + SnapshotName

A representação de string de um comando é usada para exibir as informações que você deseja ver no design do fluxo de trabalho durante o Planejamento e a execução. Você deve usar apenas os parâmetros do comando na representação de string para um comando.

- a. **Opcional:** se você estiver criando um comando Wait, selecione **Wait for Condition** na seção **Command type** e defina o valor necessário no campo **Waiting Interval(s)**.

4. Na guia **Código**, execute as seguintes etapas:

- a. Selecione a linguagem de script necessária para o comando na lista **linguagem de script**.

Você pode clicar e selecionar um idioma adicional para o comando.

- b. Introduza o código apropriado para o comando no separador de idioma selecionado.

Se você quiser usar o tipo de senha para entradas de usuário no script do PowerShell, crie um alias para o parâmetro e inclua `_Password` no atributo. Para o script Perl, você pode especificar o tipo como **Password** na guia **Parameters Definition**.



Os caracteres especiais a seguir não são suportados no editor de comandos: ">", "&" e etiquetas XML ">".

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

5. Na guia **Definição de parâmetros**, execute as seguintes etapas:

- a. Clique em **Discover Parameters** para preencher a tabela de definição de parâmetros.

Os parâmetros e seus atributos são extraídos do código e exibidos na tabela. Por exemplo, os parâmetros `Array` e `VolumeName` são extraídos do seguinte código:

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

- a. Clique na coluna de descrição dos parâmetros para editar a descrição.

Não é possível editar nenhum outro campo nesta guia.

6. Na guia **Mapeamento de parâmetros**, execute as seguintes etapas para cada parâmetro:

- a. Na coluna **Type**, selecione o objeto de dicionário apropriado.

- b. Na coluna **Atributo**, insira ou selecione os atributos apropriados para o objeto do dicionário na lista.

Depois de inserir um atributo, você pode inserir um ponto (.) e incluir outro atributo desse objeto.

Digite `cm_storage.volume` como tipo e `aggregate.name` como o atributo para o parâmetro `AggregateName`.

- c. Na coluna **Nome do objeto**, insira um nome de objeto.

O nome do objeto é usado para agrupar os parâmetros em uma guia na caixa de diálogo parâmetros para <command> quando você está especificando os detalhes do comando em um fluxo de trabalho.

Os parâmetros não mapeados são exibidos na guia **outros parâmetros** da caixa de diálogo parâmetros para <command> quando você estiver especificando os detalhes do comando em um fluxo de trabalho.

7. Na guia **Reservation**, insira um script de reserva usando consultas SQL para reservar os recursos necessários pelo comando durante a execução de um fluxo de trabalho programado:

- a. **Opcional:** no campo **representação de reserva**, insira uma representação de cadeia de caracteres para a reserva usando a sintaxe MVEL.

"Adicionar regra para SnapMirror label"

```
" + SnapMirrorLabel + "[.code] Para a política SnapMirror "
```

```
" + PolicyName + "``"
```

A representação de cadeia de caracteres é usada para exibir os detalhes dos recursos reservados na janela Reservas.



O script de reserva não deve executar nenhuma operação em bancos de dados, exceto os esquemas `cm_storage`, `cm_performance`, armazenamento, desempenho, vc e personalizados.

8. **Opcional:** na guia **Verificação**, insira uma consulta SQL para verificar se o comando afetou as fontes de dados e o cache WFA conforme esperado para que a reserva possa ser removida.

A consulta SQL que você insere só pode consistir em instruções SQL SELECT.

- a. Clique em **Verificação de teste** para testar o script de verificação.
- b. Na caixa de diálogo **Verificação**, insira os parâmetros de teste necessários.
- c. Se você não quiser usar os dados de reserva para testar o script de verificação, desmarque o campo **usar *dados de reserva no teste**.
- d. Clique em **Teste**.
- e. Depois de rever o resultado do teste, feche a caixa de diálogo.

9. Clique em **Test** para testar o comando.

10. Na caixa de diálogo **Testing Command <command name>**, clique em **Test**.

O resultado do teste é exibido na seção mensagens de Registro da caixa de diálogo.

11. Clique em **Salvar**.

Teste o script de reserva para comandos

Você pode testar os scripts de reserva que você escreveu para comandos OnCommand

Workflow Automation (WFA) no banco de dados playground para garantir que os scripts estejam funcionando bem e não afetando as tabelas de banco de dados DO WFA.

Sobre esta tarefa

O caminho de instalação padrão DO WFA é usado neste procedimento. Se você alterou o local padrão durante a instalação, você deve usar o caminho de instalação alterado DO WFA.

Passos

1. Abra um prompt de comando no servidor WFA e mude os diretórios para o seguinte local: c:\Program Files\NetApp\WFA\mysql\bin
2. Crie um dump do banco de dados WFA usando o seguinte comando: mysql -u wfa -pWfa123 --single-transaction --skip-add-drop-table database_tables> dump_location

Comando para criar um despejo das tabelas da base de dados cm_storage: mysql -u wfa -pWfa123 --single-transaction --skip-add-drop-table cm_storage> c:\tmp\cmSt2.sql

3. Restaure o despejo que você criou no banco de dados DO playground DO WFA usando o seguinte comando: mysql -u wfa -pWfa123 playground < dump_location

```
mysql -u wfa -pWfa123 playground < c:\tmp\cmSt2.sql
```

4. Crie ou edite um comando WFA e escreva o script de reserva na guia **Reservation**.

Você deve garantir que os scripts de reserva e verificação usam apenas o banco de dados playground.

5. Crie ou edite um fluxo de trabalho, inclua o comando no fluxo de trabalho e execute o fluxo de trabalho.
6. Verifique se os scripts de reserva e verificação estão funcionando conforme esperado.

O processo de aquisição da fonte de dados WFA não atualiza a base de dados do playground. Você deve remover as reservas criadas pelo comando manualmente.

Crie um localizador

Você pode criar um localizador WFA que pode procurar recursos se não houver um localizador WFA predefinido adequado para pesquisar os recursos necessários.

O que você vai precisar

Você deve ter criado os filtros necessários que são usados no localizador.

Passos

1. Clique em **Workflow Design > Finders**.
2. Clique  na barra de ferramentas.
3. Na guia **Propriedades** da caixa de diálogo **novo Finder**, insira ou selecione os detalhes necessários nos campos **Nome**, **tipo** e **Descrição**.
4. Na guia **filtros**, selecione os filtros necessários na lista **filtros disponíveis** e clique  em .

Você pode adicionar ou remover filtros com base em suas necessidades.

5. Na guia atributos retornados, selecione os atributos necessários para o filtro na lista **disponível** e clique  em .
 6. **Opcional:** clique em **Teste** para testar o localizador.
 - a. Na caixa de diálogo **Test Finder <FinderName>**, insira os parâmetros de teste necessários.
 - b. Desmarque a caixa de seleção **usar dados de reserva no teste** se não quiser usar os dados de reserva para testar o localizador.
 - c. Clique em **Teste**.
 - O resultado do teste é exibido.
 - d. Feche a caixa de diálogo.
7. Clique em **Salvar**.

Crie um filtro

Você pode criar um FILTRO WFA que pode procurar recursos se não houver um filtro WFA predefinido adequado para a tarefa.

O que você vai precisar

Você deve saber as sintaxes SQL apropriadas para criar o filtro.

Passos

1. Clique em **Workflow Design > Filters**.
2. Clique  na barra de ferramentas.
3. Na guia **Propriedades** da caixa de diálogo **novo filtro**, insira ou selecione os detalhes necessários nos campos **Nome**, **tipo de Dicionário** e **Descrição**.
4. Na guia **consulta**, insira a consulta SQL apropriada para o filtro.

Você deve inserir uma única consulta SQL e, opcionalmente, usar parâmetros de entrada. Você deve usar a seguinte sintaxe para usar um parâmetro de entrada: \${ParameterName} .

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

5. Clique em **Atualizar** para preencher a tabela **parâmetros de entrada** e a lista **atributos retornados**.

Essas informações são obtidas a partir da consulta SQL que você inseriu. Por exemplo, se você usar o exemplo de consulta SQL da etapa anterior, ip é exibido em atributos retornados e ArrayName é exibido em parâmetros de entrada. Você pode editar as entradas nas colunas **Label** e **Description**.

6. **Opcional:** clique em **Teste** para testar o filtro.
 - a. Na caixa de diálogo **Test Filter <FilterName>** (Teste de filtro), insira os parâmetros de teste

necessários.

- b. Desmarque a caixa de seleção **usar dados de reserva no teste** se não quiser usar os dados de reserva para testar o filtro.
- c. Clique em **Teste**.

O resultado do teste é exibido.

- d. Feche as caixas de diálogo.

7. Clique em **Salvar**.

Crie uma entrada de dicionário

Você pode criar uma entrada de dicionário WFA quando quiser definir um novo tipo de objeto e sua relação em seu ambiente de armazenamento.

Passos

1. Clique em **Design da fonte de dados > Dicionário**.
2. Clique  na barra de ferramentas.
3. Na caixa de diálogo **Nova entrada do Dicionário**, insira os detalhes necessários nos campos **Nome do tipo de objeto** e **Descrição**.
4. Para o campo **Scheme**, execute uma das seguintes ações:
 - Selecione um dos esquemas disponíveis na lista.
 - Clique em **Adicionar novo esquema**, insira o **Nome do esquema** obrigatório na caixa de diálogo novo esquema e clique em **Adicionar**.
5. Clique em **Adicionar linha** e execute as seguintes etapas para descrever o atributo:
 - a. Clique na coluna **Nome** e insira o nome do atributo.
 - b. Na coluna **Type**, selecione o tipo necessário.

A coluna **comprimento da cadeia** é preenchida e editável se você selecionou a cadeia de caracteres como o tipo. Além disso, a coluna **values** é editável se você selecionou **enum** como o tipo.

- c. Marque as caixas de seleção apropriadas para o atributo nos **Natural Key**, **To be Cached**, e **Can be Null columns**.

Se tiver selecionado a caixa de verificação **Natural Key**, não poderá selecionar a caixa de verificação **pode ser nulo**.

- d. Adicione os atributos necessários para o objeto dicionário.
- e. **Opcional:** Selecione a caixa de seleção **valores nas colunas de chave natural diferenciam maiúsculas de minúsculas** se você quiser que as teclas naturais sejam sensíveis a maiúsculas e minúsculas.

6. Clique em **Salvar**.

Crie uma função

Você pode criar uma FUNÇÃO WFA que pode ser usada como um utilitário, se não houver uma função WFA predefinida que seja adequada para sua tarefa.

O que você vai precisar

Você deve conhecer as sintaxes MVFLEX Expression Language (MVEL) para criar uma função.

Sobre esta tarefa

Você deve incluir o seguinte para as definições de função:

- Nome: Nome da função

Você não deve usar uma palavra reservada na sintaxe MVEL. Cada função deve ter um nome exclusivo.

- Definição MVEL: Uma cadeia de caracteres especificando a sintaxe MVEL da definição da função

Passos

1. Clique em **Workflow Design > funções**.
2. Clique  na barra de ferramentas.
3. Na caixa de diálogo **Nova função**, insira ou selecione os detalhes necessários nos campos **Descrição da função** e **Definição da função**.

```
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);
}
```

O campo **Function name** é preenchido a partir dos dados usados na sintaxe MVEL.

4. * Opcional: * Clique * Teste * para testar a função:

- a. Na seção **expressão** da caixa de diálogo **Teste**, insira a expressão necessária da função.

actualVolumeSize(600, 1)

- b. Clique em **Teste**.

O resultado do teste é exibido.

- c. Feche a caixa de diálogo.

5. Clique em **Salvar**.

Crie um modelo

Você pode criar um modelo que pode ser usado como um blueprint para preencher atributos em detalhes de comando.

Passos

1. Clique em **Workflow Design > Templates**.
2. Clique  na barra de ferramentas.
3. Na caixa de diálogo **novo modelo**, insira ou selecione os detalhes necessários nos campos **Nome**, **tipo** e **Descrição**.

A tabela atributos é preenchida com base no objeto do dicionário selecionado no campo **tipo**.

4. Clique na coluna valor de cada atributo e execute um dos seguintes procedimentos:
 - Introduza ou selecione o valor pretendido na lista.
 - Insira uma entrada de usuário - por exemplo, \$size para entrada de usuário de tamanho.
5. Clique em **Salvar**.

Crie uma consulta de cache

Você pode definir uma consulta de cache quando quiser armazenar em cache informações sobre um objeto de dicionário no banco de dados WFA a partir de um tipo de fonte de dados. Você pode criar uma consulta de cache e associá-la a uma entrada de dicionário e a um ou mais tipos de fonte de dados, como o Active IQ Unified Manager 6,1.

O que você vai precisar

Você deve saber as sintaxes SQL apropriadas para criar uma consulta de cache.

Passos

1. Clique em **Data Source Design > Cache queries**.
2. Clique  na barra de ferramentas.
3. Na caixa de diálogo **Add Cache Query** (Adicionar consulta de cache), selecione a entrada do dicionário e o tipo de fonte de dados necessários.
4. Na seção ""SQL SELECT query"", insira a consulta SQL apropriada.

A consulta SQL a seguir armazena em cache informações sobre o objeto do dicionário de disco do tipo de fonte de dados do Active IQ Unified Manager 6,1:

```

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. Se você quiser testar a consulta SQL, clique em **Test**.

Se você tiver selecionado mais de um tipo de fonte de dados, a caixa de diálogo testar consulta de cache será aberta e permitirá que você selecione o tipo de origem de dados necessário.

O resultado do teste é exibido.

6. Feche a caixa de diálogo.

7. Clique em **Salvar**.

Crie agendas recorrentes

O OnCommand Workflow Automation (WFA) fornece duas opções de agendamento para fluxos de trabalho. Você pode agendar um fluxo de trabalho para ser executado uma vez em um momento específico ou criar agendas recorrentes e associar as programações a fluxos de trabalho para que os fluxos de trabalho sejam executados rotineiramente.

Sobre esta tarefa

Uma programação que você criou pode ser reutilizada e associada a vários fluxos de trabalho.

Passos

1. Clique em **execução > horários**.
2. Clique  na barra de ferramentas.
3. Na caixa de diálogo **Nova programação**, insira ou selecione um nome, descrição e frequência para a programação.

Para a frequência, tem de introduzir a hora no formato de 24 horas. A hora do servidor WFA é aplicada às programações.

4. Clique em **OK**.

Depois de terminar

- Você pode associar a programação a um fluxo de trabalho ao executar o fluxo de trabalho usando a opção **Executar recursivamente**.
- Você pode visualizar os detalhes de um fluxo de trabalho e sua associação com um agendamento clicando em **execução > horários recorrentes**.

O Planejamento de recursos e execução dos fluxos de trabalho que estão programados para serem executados uma vez são feitos imediatamente quando os fluxos de trabalho são cronogramas. No entanto, o Planejamento de recursos e execução para os fluxos de trabalho com agendamento recorrente ocorre na hora agendada e não quando a programação está associada a um fluxo de trabalho.

Definir regras de filtro

Você pode definir um conjunto de regras para filtrar recursos de entrada de dicionário, como unidades do vFiler, agregados e máquinas virtuais. Você pode personalizar as regras para fluxos de trabalho existentes e novos fluxos de trabalho enquanto as cria.

Passos

1. Faça login NO WFA através de um navegador da Web como administrador.
2. Clique em **Workflow Design > fluxos de trabalho**.
3. Na janela **fluxos de trabalho**, clique duas vezes no fluxo de trabalho que deseja modificar.

É apresentada a janela Workflow <workflow name> (fluxo de trabalho).

4. Defina um conjunto de regras escolhendo uma das seguintes opções:

Se você quiser...	Então faça isso...
Procure recursos quando os comandos em uma linha forem repetidos	<ol style="list-style-type: none">Clique em um número de linha e selecione Repetir linha.Na caixa de diálogo repetições de linha, selecione a opção para cada recurso no grupo na lista suspensa repetições.Selecione um tipo de recurso.Clique no link Digite os critérios de pesquisa.

Se você quiser...	Então faça isso...
Procure recursos necessários em entradas de comando	<p>a. Clique  em .</p> <p>b. Na caixa de diálogo parâmetros para <command_name>, selecione a opção procurando por uma opção <dictionary object> existente na lista suspensa Definir <dictionary object>.</p> <p>c. Clique no link Digite os critérios de pesquisa.</p>
PESQUISE recursos referenciados por variáveis em entradas de comando	<p>a. Clique  em .</p> <p>b. Na caixa de diálogo parâmetros para <command_name>, selecione a opção preenchendo atributos na lista suspensa Definir <dictionary object>.</p> <p>c. Clique  em para obter um campo marcado com .</p>
Entradas de comando Name do tipo String	<p>a. Clique  em .</p> <p>b. Na caixa de diálogo parâmetros para <command_name>, selecione a opção preenchendo atributos na lista suspensa Definir <dictionary object>.</p> <p>c. Clique  para obter um campo de cadeia de caracteres.</p>

5. Na caixa de diálogo **seleção de recursos**, marque a caixa de seleção **Definir regras de filtro**.

Se você tiver selecionado uma das opções na caixa de diálogo Finder drop-down of Resource Selection (seleção de recursos), a caixa de seleção Definir regras de filtro será desativada. O valor para o localizador deve ser definido como ""nenhum"" para que as regras de filtro definam sejam ativadas.

6. Insira o atributo, operador e valor da regra.

O valor deve ser fornecido dentro de aspas simples. As regras de filtro podem conter um ou mais grupos.

7. Clique em **OK**.

Adicione pontos de aprovação

Você pode adicionar um ponto de aprovação como um ponto de verificação em um fluxo de trabalho para pausar a execução do fluxo de trabalho e retomá-lo com base em sua aprovação. Você pode usar pontos de aprovação para execução incremental de um fluxo de trabalho, em que seções do fluxo de trabalho são executadas somente após uma determinada condição ser atendida - por exemplo, quando a próxima seção tiver que ser aprovada ou quando a execução bem-sucedida da primeira seção for validada.

Passos

1. Faça login NO WFA através de um navegador da Web como arquiteto ou administrador.
2. Clique em **Workflow Design > fluxos de trabalho**.
3. Na janela **fluxos de trabalho**, clique duas vezes no fluxo de trabalho que deseja modificar.
4. Na janela **Workflow <workflow name>**, clique no  ícone à esquerda da etapa para a qual deseja adicionar o ponto de aprovação.

Você pode adicionar pontos de aprovação para uma ou mais etapas.

5. Na caixa de diálogo **novo ponto de aprovação**, forneça os detalhes do comentário e da condição.
6. Clique em **OK**.

Diretrizes de codificação para WFA

Você deve entender as diretrizes gerais de codificação do OnCommand Workflow Automation (WFA), convenções de nomenclatura e recomendações sobre a criação de vários blocos de construção, como filtros, funções, comandos e fluxos de trabalho.

Diretrizes para variáveis

Você deve estar ciente das diretrizes para variáveis PowerShell e Perl no OnCommand Workflow Automation (WFA) ao criar um comando ou um tipo de fonte de dados.

Variáveis do PowerShell

Diretrizes	Exemplo
Para parâmetros de entrada de script:	<p>\$VolumeName</p> <ul style="list-style-type: none"> • Use Pascal Case. • Não utilize sublinhados. • Não utilize abreviaturas.
Para variáveis internas de script:	<p>\$AutoDeleteOptions</p> <p>\$Size</p> <p>\$newVolume</p> <p>\$qtreeName</p> <p>\$time</p>
Para funções:	GetVolumeSize

Diretrizes	Exemplo
Nomes de variáveis não são sensíveis a maiúsculas e minúsculas. No entanto, para melhorar a legibilidade, você não deve usar letras maiúsculas diferentes para o mesmo nome.	\$variable é o mesmo que \$Variable.
Os nomes das variáveis devem estar em inglês simples e devem estar relacionados à funcionalidade do script.	Use \$name e não \$a.
Declare o tipo de dados para cada variável, explicitamente.	<pre>[string]\$name [int]\$tamanho</pre>
Não use caracteres especiais (! a e % , .) e espaços.	Nenhum
Não use palavras-chave reservadas do PowerShell.	Nenhum
Agrupe os parâmetros de entrada colocando os parâmetros obrigatórios primeiro seguidos pelos parâmetros opcionais.	<pre>param([parameter(Mandatory=\$true)] [string]\$Type, [parameter(Mandatory=\$true)] [string]\$Ip, [parameter(Mandatory=\$false)] [string]\$VolumeName)</pre>
Comente todas as variáveis de entrada usando <i>HelpMessage</i> anotação com uma mensagem de ajuda significativa.	<pre>[parameter(Mandatory=\$false, HelpMessage="LUN to map")] [string]\$LUNName</pre>
Não use "Filer" como um nome de variável; use "Array" em vez disso.	Nenhum
`_ValidateSet_` Use a anotação nos casos em que o argumento obtém valores enumerados. Isso se traduz automaticamente para o tipo de dados Enum para o parâmetro.	<pre>[parameter(Mandatory=\$false, HelpMessage="Volume state")] [ValidateSet("online", "offline", "restricted")] [string]\$State</pre>

Diretrizes	Exemplo
<p>Adicione um alias a um parâmetro que termine com "<u>_capacity</u>" para indicar que o parâmetro é do tipo capacidade.</p>	<p>O comando "Create volume" usa aliases da seguinte forma:</p> <pre>[parameter(Mandatory=\$false, HelpMessage="Volume increment size in MB")] [Alias("AutosizeIncrementSize_Capacity")] [int]\$AutosizeIncrementSize</pre>
<p>Adicione um alias a um parâmetro que termine com "<u>_Password</u>" para indicar que o parâmetro é do tipo de senha.</p>	<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>

Variáveis Perl

Diretrizes	Exemplo
<p>Para parâmetros de entrada de script:</p> <ul style="list-style-type: none"> • Use Pascal Case. • Não utilize sublinhados. • Não utilize abreviaturas. 	<pre>\$VolumeName \$AutoDeleteOptions \$Size</pre>
<p>Não use abreviações para variáveis internas de script.</p>	<pre>\$new_volume \$qtree_name \$time</pre>
<p>Não utilize abreviaturas para funções.</p>	<pre>get_volume_size</pre>

Diretrizes	Exemplo
Nomes de variáveis são sensíveis a maiúsculas e minúsculas. Para melhorar a legibilidade, você não deve usar letras maiúsculas diferentes para o mesmo nome.	\$variable não é o mesmo que \$Variable.
Os nomes das variáveis devem estar em inglês simples e devem estar relacionados à funcionalidade do script.	Use \$name e não \$a.
Agrupe os parâmetros de entrada colocando os parâmetros obrigatórios primeiro, seguidos pelos parâmetros opcionais.	Nenhum
Na função GetOptions, declare explicitamente o tipo de dados de cada variável para parâmetros de entrada.	<pre>GetOptions ("Name=s"=>\\$Name, "Size=i"=>\\$Size)</pre>
Não use "Filer" como um nome de variável; use "Array" em vez disso.	Nenhum
Perl não inclui a <i>ValidateSet</i> anotação para valores enumerados. Use declarações explícitas "if" para casos em que argumento obtém valores enumerados.	<pre>if (defined\$SpaceGuarantee&&!(\$SpaceGuaranteeeq'none')</pre>
	\$SpaceGuaranteeeq'volume'
	\$SpaceGuaranteeeq'file')) { die'Illegal SpaceGuarantee argument: \\".\$SpaceGuarantee.\\"; }

Todos os comandos Perl WFA devem usar o pragma "strict" para desencorajar o uso de construções inseguras para variáveis, referências e sub-rotinas.	<pre>use strict; # the above is equivalent to use strictvars; use strictsubs; use strictrefs;</pre>

Diretrizes	Exemplo
<p>Todos os comandos Perl WFA devem usar os seguintes módulos Perl:</p> <ul style="list-style-type: none"> • Getopt <p>Isso é usado para especificar parâmetros de entrada.</p> • WFAUtil <p>Isso é usado para funções de utilitário que são fornecidas para Registro de comandos, relatório do progresso do comando, conexão a controladores de array e assim por diante.</p> 	<pre data-bbox="850 196 1155 304">use Getopt::Long; use NaServer; use WFAUtil;</pre>

Diretrizes para indentação

Você deve estar ciente das diretrizes para indentação ao escrever um script PowerShell ou Perl para OnCommand Workflow Automation (WFA).

Diretrizes	Exemplo
Um separador é igual a quatro espaços vazios.	
Use abas e chaves para mostrar o início e o fim de um bloco.	<p>Script do PowerShell</p> <pre data-bbox="850 1157 1351 1347">if (\$pair.length-ne 2) { throw "Got wrong input data" }</pre> <p>Script Perl</p> <pre data-bbox="850 1522 1383 1833">if defined \$MaxDirectorySize { # convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024; }</pre>

Diretrizes	Exemplo
Adicione linhas em branco entre conjuntos de operações ou blocos de código.	<pre>\$options=\$option.trim(); \$pair=\$option.split(" "); Get-WFALogger -Info -messages \$(`"split options: `"+\$Pair)</pre>

Diretrizes para comentários

Você deve estar ciente das diretrizes para comentários do PowerShell e Perl em seus scripts para OnCommand Workflow Automation (WFA).

Comentários do PowerShell

Diretrizes	Exemplo
Use o caractere nº para um comentário de linha única.	<pre># Single line comment \$options=\$option.trim();</pre>
Use o caractere nº para um comentário de fim de linha.	<pre>\$options=\$option.trim(); # End of line comment</pre>
Use os caracteres no e no> para um comentário em bloco.	<pre><# This is a block comment #> \$options=\$option.trim();</pre>

Perl comentários

Diretrizes	Exemplo
Use o caractere nº para um comentário de linha única.	<pre># convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024;</pre>
Use o caractere nº para o comentário de fim de linha.	<pre>my \$MaxDirectorySizeBytes = \$MaxDirect orySize * 1024 * 1024; # convert to Bytes</pre>
Use o caractere nº em cada linha com um número vazio no início e no fim para criar uma borda de comentário para comentários multilinhas.	<pre># # This is a multi-line comment. Perl 5, unlike # Powershell, does not have direct support for # multi-line comments. Please use a '#' in every line # with an empty '#' at the beginning and end to create # a comment border #</pre>
Não inclua código comentado e morto nos comandos DO WFA. No entanto, para fins de teste, você pode usar o mecanismo DE Documentação Velha simples (POD) para comentar o código.	<pre>=begin comment # Set deduplication if(defined \$Deduplication && \$Deduplication eq "enabled") { \$wfaUtil- >sendLog ("Enabling Deduplication"); } =end comment =cut</pre>

Diretrizes para o Registro de logs

Você deve estar ciente das diretrizes para o Registro ao escrever um script PowerShell

ou Perl para OnCommand Workflow Automation (WFA).

Log do PowerShell

Diretrizes	Exemplo
Use o cmdlet Get-WFALogger para Registro.	<pre>Get-WFALogger -Info -message "Creating volume"</pre>
Registre todas as ações que requerem interação com pacotes internos, como Data ONTAP, VMware e PowerCLI. Todas as mensagens de log estão disponíveis em Logs de execução no histórico de status de execução dos fluxos de trabalho.	Nenhum
Registre cada argumento relevante que é passado para pacotes internos.	Nenhum
Use níveis de log apropriados ao usar o cmdlet Get-WFALogger, dependendo do contexto de uso. -Info, -Error, -WARN e -Debug são os vários níveis de log disponíveis. Se um nível de log não for especificado, então o nível de log padrão é Debug.	Nenhum

Registro Perl

Diretrizes	Exemplo
Use o sendLog do WFAUtil para Registrar.	<pre>my wfa_util = WFAUtil->new(); eval { \$wfa_util->sendLog('INFO', "Connecting to the cluster: \$DestinationCluster"); }</pre>
Registre todas as ações que exigem interação com qualquer coisa externa ao comando, como Data ONTAP, VMware e WFA. Todas as mensagens de log criadas usando a rotina sendLog do WFAUtil são armazenadas no banco de dados DO WFA. Essas mensagens de log estão disponíveis para o fluxo de trabalho e comando executados.	Nenhum
Registre cada argumento relevante passado para a rotina que foi chamada.	Nenhum

Diretrizes	Exemplo
Use níveis de log apropriados.-Info, -Error, -WARN e -Debug são os vários níveis de log disponíveis.	Nenhum
Ao Registrar no nível -Info, seja preciso e conciso. Não especifique detalhes de implementação, como nome da classe e nome da função em mensagens de log. Descreva a etapa exata ou o erro exato em inglês simples.	O snippet de código a seguir mostra um exemplo de uma mensagem boa e uma mensagem ruim: <pre>\$wfa_util->sendLog('WARN', "Removing volume: '.'.\$VolumeName); # Good Message</pre> <pre>\$wfa_util->sendLog('WARN', 'Invoking volume- destroy ZAPI: '.\$VolumeName); # Bad message</pre>

Diretrizes para tratamento de erros

Você deve estar ciente das diretrizes para manipulação de erros ao escrever um script PowerShell ou Perl para OnCommand Workflow Automation (WFA).

Manipulação de erros do PowerShell

Diretrizes	Exemplo
Parâmetros comuns adicionados aos cmdlets pelo PowerShell runtime incluem parâmetros de manipulação de erros, como ErrorAction e WarningAction: <ul style="list-style-type: none"> O parâmetro ErrorAction determina como um cmdlet deve reagir a um erro que não encerra o comando. O parâmetro WarningAction determina como um cmdlet deve reagir a um aviso do comando. Stop, SilentlyContinuar, inquirir e continuar são os valores válidos para os parâmetros ErrorAction e WarningAction. Para obter mais informações, você pode usar o Get-Help about_CommonParameters comando na CLI do PowerShell.	ErrorAction: O exemplo a seguir mostra como lidar com um erro não-terminação como um erro de terminação: <pre>New-NcIgroup-Name \$IgroupName- Protocol \$Protocol-Type\$OSType- ErrorActionstop</pre> WarningAction <pre>New-VM-Name \$VMName-VM \$SourceVM- DataStore\$DataStoreName- VMHost\$VMHost- WarningActionSilentlyContinue</pre>

Diretrizes	Exemplo
Use a instrução geral "try/catch" se o tipo da exceção recebida for desconhecido.	<pre data-bbox="842 192 1204 508"> try { "In Try/catch block" } catch { "Got exception" }</pre>
Use a instrução específica "try/catch" se o tipo da exceção recebida for conhecido.	<pre data-bbox="842 625 1432 1026"> try { "In Try/catch block" } catch[System.Net.WebException], [System.IO. IOException] { "Got exception" }</pre>
Use a declaração "finalmente" para liberar recursos.	<pre data-bbox="842 1142 1204 1617"> try { "In Try/catch block" } catch { "Got exception" } finally { "Release resources" }</pre>

Diretrizes	Exemplo
<p>Use variáveis automáticas do PowerShell para acessar informações sobre exceções.</p>	<pre data-bbox="845 192 1416 713"> try { Get-WFALogger -Info -message \$("Creating Ipspace: " + \$Ipspace) New-NaNetIpspace-Name \$Ipspace } catch { Throw "Failed to create Ipspace. Message: " + \$_.Exception.Message; }</pre>

Manipulação de erros Perl

Diretrizes	Exemplo
<p>Perl não inclui suporte de linguagem nativa para blocos try/catch. Use blocos eval para verificar e lidar com erros. Mantenha os blocos eval o mais pequenos possível.</p>	<pre data-bbox="840 200 1460 1277"> eval { \$wfa_util->sendLog('INFO', "Quiescing the relationship : \$DestinationCluster://\$Destination Vserver /\$DestinationVolume"); \$server->snapmirror_quiesce('destination-vserver' => \$DestinationVserver, 'destination-volume' => \$DestinationVolume); \$wfa_util->sendLog('INFO', 'Quiesce operation started successfully.'); }; \$wfa_util->checkEvalFailure("Failed to quiesce the SnapMirror relationship \$DestinationCluster://\$Destination Vserver /\$DestinationVolume", \$@); </pre>

Convenções gerais do PowerShell e Perl para O WFA

Você deve entender certas convenções PowerShell e Perl que são usadas no WFA para criar scripts que são consistentes com scripts existentes.

- Use variáveis que ajudam a esclarecer o que você quer que o script faça.
- Escreva código legível que pode ser entendido sem comentários.
- Mantenha os scripts e comandos o mais simples possível.
- Para scripts do PowerShell:
 - Use cmdlets sempre que possível.
 - Invoque o código .NET quando não houver cmdlet disponível.
- Para scripts Perl:
 - Sempre termine as declarações "die" com carateres de nova linha.

Na ausência de um caractere de nova linha, o número da linha do script é impresso, o que não é útil para depurar comandos Perl executados pelo WFA.

- No módulo "getopt", torne obrigatório os argumentos de string para um comando.

Módulos Perl empacotados com Windows

Alguns módulos Perl são empacotados com a distribuição Perl de estado ativo do Windows para OnCommand Workflow Automation (WFA). Você pode usar esses módulos Perl em seu código Perl para escrever comandos, apenas se eles estiverem empacotados com o Windows.

A tabela a seguir lista os módulos de banco de dados Perl que são empacotados com o Windows para WFA.

Módulo da base de dados	Descrição
DBD::mysql	Driver de interface de banco de dados Perl5 que permite que você se conecte ao banco de dados MySQL.
Tente::Tiny	Minimiza erros comuns com blocos de avaliação.
XML::libxml	Interface para libxml2 que fornece analisadores XML e HTML com interfaces Dom, SAX e XMLReader.
DBD::Cassandra	Driver de interface de banco de dados Perl5 para Cassandra que usa a linguagem de consulta CQL3.

Considerações para adicionar módulos personalizados PowerShell e Perl

Você deve estar ciente de certas considerações antes de adicionar módulos personalizados PowerShell e Perl ao OnCommand Workflow Automation (WFA). Os módulos personalizados PowerShell e Perl permitem que você use comandos personalizados para criar fluxos de trabalho.

- Durante a execução dos comandos WFA, todos os módulos personalizados do PowerShell são adicionados ao diretório de instalação DO WFA /Posh/modules são importados automaticamente.
- Todos os módulos Perl personalizados adicionados ao WFA/perl diretório estão incluídos na biblioteca __Inc__.
- Os módulos personalizados PowerShell e Perl não são copiados como parte da operação de backup DO WFA.
- Os módulos personalizados PowerShell e Perl não são restaurados como parte da operação de restauração DO WFA.

Você deve fazer o backup manual de módulos personalizados do PowerShell e Perl para copiá-los para uma nova instalação DO WFA.

O nome da pasta no diretório dos módulos deve ser o mesmo do nome do módulo.

Cmdlets e funções DO WFA

O OnCommand Workflow Automation (WFA) fornece vários cmdlets do PowerShell, bem como funções do PowerShell e Perl que você pode usar em seus comandos DO WFA.

Você pode exibir todos os cmdlets e funções do PowerShell fornecidos pelo SERVIDOR WFA usando os seguintes comandos do PowerShell:

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

Você pode visualizar todas as funções Perl fornecidas pelo servidor WFA `WFAUtil.pm` no módulo. As seções de ajuda, os cmdlets DO WFA PowerShell e a ajuda dos métodos Perl do WFA, dos links de suporte DO módulo Ajuda DO WFA permitem o acesso a todos os cmdlets e funções do PowerShell e às funções Perl.

Módulos do PowerShell e Perl WFA

Você deve estar ciente dos módulos PowerShell ou Perl para OnCommand Workflow Automation (WFA) para escrever scripts para seus fluxos de trabalho.

Módulos do PowerShell

Diretrizes	Exemplo
Use o Kit de Ferramentas PS do Data ONTAP para invocar APIs sempre que o kit de ferramentas estiver disponível.	O <code>Add_VLAN</code> comando usa o kit de ferramentas da seguinte forma: <code>Add-NaNetVlan-Interface \$Interface-Vlans\$VlanID</code>
Se não houver cmdlets disponíveis no Kit de Ferramentas PS do Data ONTAP, use o <code>Invoke-SSH</code> comando para chamar a CLI no Data ONTAP.	<code>Invoke-NaSsh-Name \$ArrayName-Command "ifconfig -a"-Credential \$Credentials</code>

Módulos Perl

O módulo `NaServer` é usado nos comandos WFA. O módulo `NaServer` permite a invocação de APIs Data ONTAP, que são usadas no gerenciamento ativo de sistemas Data ONTAP.

Diretrizes	Exemplo
<p>Use o módulo NaServer para invocar APIs sempre que o SDK de gerenciamento do NetApp estiver disponível.</p>	<p>O exemplo a seguir mostra como o módulo NaServer é usado para uma operação RESUME SnapMirror:</p> <pre> eval { \$wfa_util->sendLog('INFO', "Connecting to the cluster: \$DestinationCluster"); my \$server = \$wfa_util- >connect (\$DestinationClusterIp, \$DestinationVserver); my \$sm_info = \$server- >snapmirror_get('destination-vserver' => \$DestinationVserver, 'destination-volume' => \$DestinationVolume); my \$sm_state = \$sm_info- >{ 'attributes' }->{ 'snapmirror- info' }->{ 'mirror-state' }; my \$sm_status = \$sm_info- >{ 'attributes' }->{ 'snapmirror- info' }->{ 'relationship-status' }; \$wfa_util->sendLog('INFO', "SnapMirror relationship is \$sm_state (\$sm_status)"); if (\$sm_status ne 'quiesced') { \$wfa_util->sendLog('INFO', 'The status needs to be quiesced to resume transfer.'); } else { my \$result = \$server- >snapmirror_resume('destination-vserver' => \$DestinationVserver, 'destination-volume' => \$DestinationVolume); \$wfa_util->sendLog('INFO', "Result of resume: \$result"); } } </pre>

Diretrizes	Exemplo
<p>Se uma API do Data ONTAP não estiver disponível, chame a CLI do Data ONTAP usando o método do utilitário executeSystemCli.</p> <p> O executeSystemCli não é suportado e está atualmente disponível apenas para o Data ONTAP que funciona no modo 7D.</p>	<p>Nenhum</p>

Considerações ao converter comandos do PowerShell para Perl

Você deve estar ciente de certas considerações importantes ao converter comandos do PowerShell para Perl porque o PowerShell e o Perl têm recursos diferentes.

Tipos de entrada de comando

O OnCommand Workflow Automation (WFA) permite que os designers de fluxo de trabalho usem arrays e hash como entrada para o comando ao definir um comando. Esses tipos de entrada não podem ser usados quando o comando é definido usando Perl. Se você quiser que um comando Perl aceite entradas de array e hash, você pode definir a entrada como uma string no designer. A definição do comando pode então analisar a entrada, que é passada para criar um array ou hash, conforme necessário. A descrição para a entrada descreve o formato em que a entrada é esperada.

```
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.
```

Declaração do PowerShell

Os exemplos a seguir mostram como uma entrada de array pode ser passada para o PowerShell e Perl. Os exemplos descrevem a entrada CronMonth, que especifica o mês em que a tarefa cron está programada para ser executada. Os valores válidos são inteiros -1 a 11. Um valor de -1 indica que o cronograma é executado a cada mês. Qualquer outro valor denota um mês específico, com 0 sendo janeiro e 11 sendo dezembro.

```
[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,
```

Declaração Perl

```

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
}

```

Definição do comando

Uma expressão de uma linha no PowerShell usando um operador pipe pode ter que ser expandida em vários blocos de instruções em Perl para alcançar a mesma funcionalidade. Um exemplo de um dos comandos Wait é mostrado na tabela a seguir.

Declaração do PowerShell	Declaração Perl
<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- >job_get_iter('query' => { 'job-type' => 'VOL_CLONE_SPLIT' }, 'desired-attributes' => { 'job-type' => '', 'job-description' => '', 'job-progress' => '', 'job-state' => '' }); my @jobarray; for my \$job (@{ \$result- >{ 'attributes-list' } }) { my \$description = \$job->{ 'job- description' }; if(\$description =~ /\$VolumeCloneName/) { push(@jobarray, \$job) } }</pre>

Diretrizes para blocos de construção WFA

Você deve estar ciente das diretrizes para o uso de componentes básicos do Workflow Automation.

Diretrizes para SQL no WFA

Você deve estar ciente das diretrizes para usar SQL no OnCommand Workflow Automation (WFA) para escrever consultas SQL para WFA.

SQL é usado nos seguintes locais no WFA:

- Consultas SQL para preencher entradas do usuário para seleção
- Consultas SQL para criar filtros para filtrar objetos de um tipo de entrada de dicionário específico
- Dados estáticos em tabelas no banco de dados playground
- Um tipo de fonte de dados personalizado do tipo SQL onde os dados devem ser extraídos de uma fonte de dados externa, como um banco de dados de gerenciamento de configuração personalizado (CMDB).
- Consultas SQL para scripts de reserva e verificação

Diretrizes	Exemplo
Palavras-chave reservadas SQL devem estar em caracteres maiúsculos.	<pre>SELECT vserver.name FROM cm_storage.vserver vserver</pre>
Os nomes de tabela e coluna devem estar em caracteres minúsculos.	<p>Tabela: Agregado Coluna: Used_space_mb</p>
Separe palavras com um caractere sublinhado (_). Não são permitidos espaços.	<p>array_performance</p>
O nome da tabela é definido no singular. Uma tabela é uma coleção de uma ou mais entradas.	<p>"função", não "funções"</p>
Use aliases de tabela com nomes significativos em consultas selecionadas.	<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>

Diretrizes	Exemplo
<p>Se você tiver que se referir a um parâmetro de entrada de filtro ou parâmetro de entrada de usuário em uma consulta de filtro ou consulta de usuário, use a sintaxe como "inputVariableName". Você também pode usar a sintaxe para se referir a um parâmetro de definição de comando em scripts de reserva e scripts de verificação.</p>	<pre> SELECT volume.name AS Name, aggregate.name AS Aggregate, volume.size_mb AS 'Total Size (MB)', voulme.used_size_mb AS 'Used Size (MB)', volume.space_guarantee AS 'Space Guarantee' FROM cm_storage.cluster, cm_storage.aggregate, cm_storage.vserver, cm_storage.volume WHERE cluster.id = vserver.cluster_id AND aggregate.id = volume.aggregate_id AND vserver.id = voulme.vserver_id AND vserver.name = '\${VserverName}' AND cluster.name = '\${ClusterName}' ORDER BY volume.name ASC </pre>
<p>Use comentários para consultas complexas. Alguns dos estilos de comentário suportados nas consultas são os seguintes:</p> <ul style="list-style-type: none"> • "--" até ao final da linha <p>Um espaço é obrigatório após o segundo hífen neste estilo de comentário.</p> <ul style="list-style-type: none"> • A partir de um caractere "" até o final da linha • A partir de uma sequência "/*" to the following " */" 	<pre> /* multi-line comment */ --line comment SELECT ip as ip, # comment till end of this line NAME as name FROM --end of line comment storage.array </pre>

Diretrizes para funções WFA

Você pode criar funções para encapsular lógica comumente usada e mais complexa em uma função nomeada e, em seguida, reutilizar a função como valores de parâmetro de comando ou valores de parâmetros de filtro no OnCommand Workflow Automation (WFA).

Diretrizes	Exemplo
Use Camel Case para um nome de função.	CalculateVolumeSize
Nomes de variáveis devem estar em inglês simples e relacionados à funcionalidade da função.	SplitByDelimiter
Não utilize abreviaturas.	CalculateVolumeSize, <i>not</i> calcVolSize
As funções são definidas usando MVFLEX Expression Language (MVEL).	Nenhum
A definição da função deve ser especificada de acordo com as diretrizes oficiais da linguagem de Programação Java.	Nenhum

Diretrizes para entradas do dicionário WFA

Você deve estar ciente das diretrizes para criar entradas de dicionário no OnCommand Workflow Automation (WFA).

Diretrizes	Exemplo
Os nomes de entrada do dicionário devem conter apenas caracteres alfanuméricos e sublinhados.	Cluster_License Switch_23
Os nomes de entrada do dicionário devem começar com um caractere maiúsculo. Comece cada palavra no nome com um caractere maiúsculo e separe cada palavra com um sublinhado (_).	Volume Array_License
Os nomes dos atributos de entrada do dicionário não devem incluir o nome da entrada do dicionário.	Nenhum
Atributos e referências em uma entrada de dicionário devem ser em caracteres minúsculos.	agregado, size_mb
Separe palavras com um sublinhado. Não são permitidos espaços.	resource_pool

Diretrizes	Exemplo
Entradas de dicionário não podem incluir referências que são de um esquema diferente. Quando uma entrada de dicionário requer referência cruzada para um objeto em um esquema diferente, certifique-se de que todas as chaves naturais do objeto a ser referido estão presentes na entrada do dicionário.	A entrada do dicionário Array_Performance requer todas as chaves naturais da entrada do dicionário Array como atributos diretos nela.
Use tipos de dados apropriados para atributos.	Nenhum
Use o tipo de dados longo para atributos relacionados ao tamanho ou ao espaço.	Size_mb e available_size_mb no armazenamento. Entrada do dicionário de volume
Use Enum quando um atributo tiver um conjunto fixo de valores.	raid_type no storage. Entrada do dicionário de volume
Defina "to be cached" como true para um atributo ou referência quando uma fonte de dados fornece valor para esse atributo ou referência. Para a fonte de dados Active IQ Unified Manager, adicione atributos armazenável em cache se a fonte de dados puder fornecer o valor a ele.	Nenhum
Defina "can be NULL" como true se a fonte de dados que fornece o valor para este atributo ou referência pode retornar NULL.	Nenhum
Forneça uma descrição significativa para cada atributo e referência. A descrição é exibida em detalhes de comando ao projetar um fluxo de trabalho.	Nenhum
Não use "id" como o nome de um atributo em entradas de dicionário. Ele é reservado para uso interno DO WFA.	Nenhum

Informações relacionadas

[Referências ao material de aprendizagem](#)

Diretrizes para comandos

Você deve estar ciente das diretrizes para criar comandos no OnCommand Workflow Automation (WFA).

Diretrizes	Exemplo
Use um nome facilmente identificável para comandos.	Create Qtree
Use espaços para delimitar palavras e cada palavra deve começar com um caractere maiúsculo.	Create Volume
Forneça uma descrição para explicar a funcionalidade do comando, incluindo o resultado esperado dos parâmetros opcionais.	Nenhum
Por padrão, o tempo limite para comandos padrão é de 600 segundos. O tempo limite padrão é definido durante a criação do comando. Altere o valor padrão somente se o comando puder levar mais tempo para ser concluído.	Create Volume comando
No caso de operações de longa duração, crie dois comandos - um para invocar a operação de longa duração e outro para relatar o progresso da operação periodicamente. O primeiro comando deve ser um Standard Execution tipo de comando e o segundo deve ser Wait for Condition tipo de comando.	Create VSM e Wait for VSM comandos
Prefix os Wait for condition nomes de comando com "wait" para fácil identificação.	Wait for CM Volume Move
Use um intervalo de espera apropriado para os comandos "wait for condition". O valor especificado controla o intervalo no qual o comando de polling é executado para verificar se a operação de longa duração está concluída.	60s intervalo de amostragem para o Wait for VSM comando
Para os Wait for condition comandos, use um tempo limite apropriado com base no tempo esperado para a operação de longa duração ser concluída. O tempo esperado pode ser consideravelmente mais longo se a operação envolver transferência de dados em uma rede.	Uma transferência de linha de base VSM pode levar muitos dias para ser concluída. Portanto, o tempo limite especificado é de 6 dias.

Representação de cadeia de carateres

A representação de string de um comando exibe os detalhes de um comando em um design de fluxo de trabalho durante o Planejamento e a execução. Somente os parâmetros do comando podem ser usados na representação de string para um comando.

Diretrizes	Exemplo
Evite usar atributos que não tenham nenhum valor. Um atributo sem um valor é exibido como na.	VolName 10.68.66.212[na]aggr1/testVol7
Separe diferentes entradas na representação de string usando os seguintes delimitadores: [] , / :	ArrayName [ArrayIp]
Forneça rótulos significativos para cada valor na representação de cadeia de caracteres.	Volume name=VolumeName

Linguagem de definição de comando

Os comandos podem ser escritos usando as seguintes linguagens de script suportadas:

- PowerShell
- Perl

Definição do parâmetro do comando

Os parâmetros do comando são descritos por Nome, Descrição, tipo, um valor padrão para o parâmetro e se o parâmetro é obrigatório. O tipo de parâmetro pode ser String, Boolean, Integer, Long, Double, Enum, DateTime, Capacity, Array, Hashtable, Password ou XmlDocument. Embora os valores para a maioria dos tipos sejam intuitivos, os valores para Array e Hashtable devem estar em um formato específico, conforme descrito na tabela a seguir:

Diretrizes	Exemplo
Certifique-se de que o valor de um tipo de entrada Array seja uma lista de valores, separados por vírgula.	<pre>[parameter (Mandatory=\$false, HelpMessage="Months in which the schedule executes.")] [array]\$CronMonths</pre> <p>A entrada é passada da seguinte forma: 0,3,6,9</p>
Certifique-se de que o valor de um tipo de entrada Hashtable seja uma lista de pares chave-valor, separados por ponto-e-vírgula.	<pre>[parameter (Mandatory=\$false, HelpMessage="Volume names and size (in MB)")] [hashtable]\$VolumeNamesAndSize</pre> <p>A entrada é passada da seguinte forma: volume1-100; Volume2-250; Volume3-50</p>

Diretrizes para fluxos de trabalho

Você deve estar ciente das diretrizes para criar ou modificar um fluxo de trabalho

predefinido para o OnCommand Workflow Automation (WFA).

Orientações gerais

Diretrizes	Exemplo
Nomeie o fluxo de trabalho de modo que ele reflita a operação executada pelo operador de armazenamento.	Create a CIFS Share
Para nomes de fluxo de trabalho, capitalize a letra inicial da primeira palavra e cada palavra que é um objeto. Letras maiúsculas para abreviaturas e acrônimos.	Volume Qtree Crie um compartilhamento CIFS de Qtree Data ONTAP em cluster
Para descrições de fluxo de trabalho, inclua todas as etapas importantes do fluxo de trabalho, incluindo quaisquer pré-requisitos, resultado do fluxo de trabalho ou aspetos condicionais de execução.	Veja a descrição do fluxo de trabalho de amostra Create VMware NFS Datastore on Clustered Data ONTAP Storage, que inclui os pré-requisitos.
Defina "Pronto para produção" como true somente quando o fluxo de trabalho estiver pronto para produção e puder ser exibido na página do portal.	Nenhum
Por padrão, defina "considerar elementos reservados" como verdadeiro. Ao visualizar um fluxo de trabalho para execução, o planejador WFA considera todos os objetos que são reservados junto com os objetos existentes no banco de dados de cache. Efeitos de outros fluxos de trabalho programados ou fluxos de trabalho executados em paralelo são considerados ao Planejar um fluxo de trabalho específico, se essa opção estiver definida como true.	<ul style="list-style-type: none">• Cenário 1 O fluxo de trabalho 1 cria um volume e está programado para ser executado uma semana depois. O fluxo de trabalho 2 cria qtrees ou LUNs em volumes que são pesquisados e, se o fluxo de trabalho 2 for executado dentro de um dia ou mais, você deve desativar "considerar elementos reservados" para o fluxo de trabalho 2 para evitar que ele considere o volume que deve ser criado em uma semana.• Cenário 2 O fluxo de trabalho 1 usa o Create Volume comando. Se houver um fluxo de trabalho programado 2 que consome 100 GB de um agregado, o fluxo de trabalho 1 deve considerar os requisitos do fluxo de trabalho 2 durante o Planejamento.

Diretrizes	Exemplo
<p>Por padrão, ""enable element existence validation"" é definido como true.</p>	<ul style="list-style-type: none"> • Cenário 1 <p>Se você criar um fluxo de trabalho que primeiro remove um volume por nome usando o comando Remove Volume somente se o volume existir, e o recriá-lo usando outro comando Create Volume como ou Clone Volume, então o fluxo de trabalho não deve usar esse sinalizador. O efeito de remover o volume não estará disponível para o Create volume comando, fazendo com que o fluxo de trabalho falhe.</p> <ul style="list-style-type: none"> • Cenário 2 <p>O Create Volume comando é usado em um fluxo de trabalho com um nome específico como "vol198".</p> <p>Se essa opção estiver definida como true, o planejador DO WFA verifica durante o Planejamento para ver se existe um volume com esse nome na matriz específica. Se o volume existir, o fluxo de trabalho falhará durante o Planejamento.</p>
<p>Quando o mesmo comando for selecionado mais de uma vez em um fluxo de trabalho, forneça nomes de exibição apropriados para as instâncias de comando.</p>	<p>O fluxo de trabalho de exemplo "criar, mapear e proteger LUNs com SnapVault" usa o Create Volume comando duas vezes. No entanto, ele usa os nomes de exibição como Create Primary Volume e Create Secondary Volume adequadamente para o volume primário e o volume de destino espelhado.</p>

Entradas do utilizador

Diretrizes	Exemplo
<p>Nomes:</p> <ul style="list-style-type: none"> • Comece o nome com o caractere "". • Use uma letra maiúscula no início de cada palavra. • Use letras maiúsculas para todos os termos e abreviaturas. • Não utilize sublinhados. 	<p>\$Array \$VolumeName</p>

Diretrizes	Exemplo
<p>Nomes de exibição:</p> <ul style="list-style-type: none"> • Use uma letra maiúscula no início de cada palavra. • Separe palavras com espaços. • Se as entradas tiverem unidades específicas, especifique a unidade entre parênteses diretamente no nome do visor. 	Volume Name Volume Size (MB)
<p>Descrições:</p> <ul style="list-style-type: none"> • Forneça uma descrição significativa para cada entrada de usuário. • Forneça exemplos quando necessário. <p>Você deve fazer isso especialmente quando a entrada do usuário é esperada para estar em um formato específico.</p> <p>As descrições de entrada do usuário são exibidas como dicas de ferramentas para as entradas do usuário durante a execução do fluxo de trabalho.</p>	Iniciadores a serem adicionados a um "iGroup". Por exemplo, IQN ou WWPN do iniciador.
<p>Tipo: Selecione Enum como o tipo se você quiser restringir a entrada a um conjunto específico de valores.</p>	Protocolo: "iscsi", "fcp", "indexado"
<p>Tipo: Selecione consulta como o tipo quando o usuário pode selecionar entre os valores disponíveis no cache WFA.</p>	<p>Tipo DE CONSULTA com consulta da seguinte forma:</p> <pre data-bbox="842 1269 1494 1469"> SELECT ip, name FROM storage.array </pre>
<p>Tipo: Marque a entrada do usuário como bloqueada quando a entrada do usuário deve ser restrita aos valores obtidos de uma consulta ou deve ser restrita apenas aos tipos Enum suportados.</p>	<p>Tipo de consulta: Somente os storages no cache podem ser selecionados. Protocolo: Tipo de Enum bloqueado com valores válidos como iscsi, fcp, misto. Nenhum outro valor além do valor válido é suportado.</p>
<p>Type: Query TypeAdicione colunas adicionais como valores de retorno na consulta quando ajuda o operador de armazenamento a fazer a escolha certa da entrada do usuário.</p>	Forneça nome, tamanho total, tamanho disponível para que o operador conheça os atributos antes de selecionar o agregado.

Diretrizes	Exemplo
<p>Tipo: Consulta TypeSQL consulta para entradas de usuário pode se referir a quaisquer outras entradas de usuário que a precedem. Isso pode ser usado para limitar os resultados de uma consulta com base em outras entradas do usuário, como unidades do vFiler de um array, volumes de um agregado, LUNs em uma máquina virtual de storage (SVM).</p>	<p>No fluxo de trabalho de exemplo Create a Clustered Data ONTAP Volume,, a consulta para VserverName é a seguinte:</p> <pre data-bbox="848 316 1403 844"> SELECT vserver.name FROM cm_storage.cluster cluster, cm_storage.vserver vserver WHERE vserver.cluster_id = cluster.id AND cluster.name = '\${ClusterName}' AND vserver.type = 'cluster' ORDER BY vserver.name ASC </pre> <p>A consulta refere-se a "ClusterName", em que "ClusterName" é o nome da entrada do usuário anterior à entrada do usuário VserverName.</p>
<p>Tipo: Use o tipo booleano com valores como "true, false" para entradas de usuário que são booleanas por natureza. Isso ajuda a escrever expressões internas no design do fluxo de trabalho usando a entrada do usuário diretamente. Por exemplo, UserInputName em vez de UserInputName.</p>	<p>\$CreateCIFSShare: Tipo booleano com valores válidos como "verdadeiro" ou "falso"</p>
<p>Tipo: para tipo de cadeia de caracteres e número, use expressões regulares na coluna valores quando quiser validar o valor com formatos específicos. Use expressões regulares para entradas de endereço IP e máscara de rede.</p>	<p>A entrada de usuário específica de localização pode ser expressa como "[A-Z][A-Z] 0[1-9]". Esta entrada de usuário aceita valores como "US-01", "NB-02", mas não "nb-00".</p>
<p>Tipo: Para o tipo de número, uma validação baseada em intervalo pode ser especificada na coluna valores.</p>	<p>Para o número de LUNs a serem criados, a entrada na coluna valores é 1-20.</p>
<p>Grupo: Agrupar entradas de usuários relacionadas em intervalos apropriados e nomear o grupo.</p>	<p>"Detalhes de armazenamento" para todas as entradas de usuário relacionadas ao armazenamento. ""Detalhes do datastore"" para todas as entradas de usuário relacionadas ao VMware.</p>

Diretrizes	Exemplo
Obrigatório: Se o valor de qualquer entrada de usuário for necessário para que o fluxo de trabalho seja executado, marque a entrada de usuário como obrigatória. Isso garante que a tela de entrada do usuário aceita obrigatoriamente essa entrada do usuário.	No fluxo de trabalho "Create NFS volume".
Valor padrão: Se uma entrada de usuário tiver um valor padrão que possa funcionar para a maioria das execuções de fluxo de trabalho, forneça os valores. Isso ajuda a permitir que o usuário forneça menos entradas durante a execução, se o padrão serve a finalidade.	Nenhum

Constantes, variáveis e parâmetros de retorno

Diretrizes	Exemplo
Constantes: Defina constantes ao usar um valor comum para definir parâmetros para vários comandos.	<code>AGREGAÇÃO_COMPROMETIMENTO_THRESHOLD</code> no Create, map, and protect LUNs with SnapVault sample workflow.
Constantes: nomes	<code>AGREEMENT_USED_SPACE_THRESHOLD</code> <code>ActualVolumeSizeInMB</code>
• Use uma letra maiúscula no início de cada palavra. • Use letras maiúsculas para todos os termos e abreviaturas. • Não utilize sublinhados. • Use letras maiúsculas para todas as letras de nomes constantes.	
Variáveis: Forneça um nome para um objeto definido em uma das caixas de parâmetro de comando. Variáveis são nomes gerados automaticamente e podem ser alteradas.	Nenhum
Variáveis: Os nomes usam caracteres minúsculos para nomes de variáveis.	<code>volume1</code> <code>partilha_cifs</code>

<p>Parâmetros de retorno: Use os parâmetros de retorno quando o Planejamento e a execução do fluxo de trabalho devem retornar alguns valores calculados ou selecionados durante o Planejamento. Os valores são disponibilizados no modo de visualização quando o fluxo de trabalho é executado a partir de um serviço da Web também.</p>	<p>Agregado: Se o agregado for selecionado usando a lógica de seleção de recursos, o agregado selecionado real pode ser definido como um parâmetro de retorno.</p>
---	---

Diretrizes para criar scripts de validação para tipos de sistema remotos

Você deve estar ciente das diretrizes para criar scripts de validação que são usados para testar os tipos de sistema remoto que você define no OnCommand Workflow Automation (WFA).

- O script Perl que você criar deve ser semelhante ao script de exemplo fornecido na janela Script de Validação.
- A saída do script de validação deve ser semelhante à do script de exemplo.

Script de validação de amostra

```
# 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;
}
```

Diretrizes para criar tipos de fonte de dados

Você deve estar ciente das diretrizes para criar tipos de fonte de dados que são usados para definir fontes de dados personalizadas para o OnCommand Workflow Automation (WFA).

Você pode definir um tipo de fonte de dados usando um dos seguintes métodos:

- SQL: Você pode usar as diretrizes WFA SQL para definir consultas selecionadas de fontes de dados com base em um banco de dados externo.
- Script: Você pode escrever um script do PowerShell que fornece os dados para um esquema específico de entradas de dicionário.

As diretrizes para a criação de tipos de fonte de dados são as seguintes:

- Você deve usar a linguagem do PowerShell deve ser usada para criar script.

- O script do PowerShell deve fornecer a saída para cada entrada do dicionário em seu diretório de trabalho atual.
- Os arquivos de dados devem ser nomeados `dictionary_entry.csv`, onde o nome da entrada do dicionário deve ser em caracteres minúsculos.

O tipo de fonte de dados predefinido que coleta informações do Performance Advisor usa um tipo de fonte de dados baseado EM SCRIPT. Os ficheiros de saída têm o nome `array_performance.csv` e `aggregate_performance.csv`.

- O `.csv` arquivo deve incluir o conteúdo na ordem exata como os atributos de entrada do dicionário.

Uma entrada de dicionário inclui atributos na seguinte ordem: `Array_IP`, `data`, `dia`, `hora`, `CPU_Busy`, `Total_OPS_per_sec`, `Disk_throughput_per_sec`

O script do PowerShell adiciona dados ao `.csv` arquivo na mesma ordem.

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

- Você deve usar Encoding para garantir que a saída de dados do script seja carregada no cache DO WFA com precisão.
- Você deve usar N ao inserir um valor nulo no `.csv` arquivo.

Palavras reservadas

OnCommand Workflow Automation (WFA) inclui algumas palavras reservadas. Você não deve usar as palavras reservadas em fluxos de trabalho para qualquer atributo ou parâmetros, como nomes de variáveis, entrada de usuário, constantes e parâmetros de retorno.

A seguir está uma lista das palavras reservadas no WFA:

• e	• Flutuação	• proto
• Array	• flutuação	• retorno
• afirmar	• para	• Tempo de execução
• Booleano	• foreach	• SecurityManager
• booleano	• função	• Curto
• Byte	• se	• curto
• byte	• importar	• soundslike
• char	• import_static	• StrictMath
• Caráter	• em	• Cadeia de carateres
• CharSequence	• instanceof	• StringBuffer
• Classe	• int	• StringBuilder
• Carregador de classes	• Número inteiro	• strsim
• Compilador	• is	• interruptor
• contém	• isdef	• Sistema
• convertable_to	• Longo	• Rosca
• def	• longo	• ThreadLocal
• do	• Matemática	• verdadeiro
• Duplo	• novo	• até
• duplo	• nulo	• var
• do contrário	• Número	• Vazio
• vazio	• Objeto	• enquanto
• falso	• ou	• com

Como você usa APIs REST

Você pode usar as APIS REST fornecidas pelo Workflow Automation (WFA) para invocar fluxos de trabalho de portais externos e o software de orquestração do data center. O WFA suporta tipos de conteúdo XML e JSON para todas as APIs REST.

O WFA permite que serviços externos acessem várias coleções de recursos, como fluxos de trabalho, usuários, filtros e localizadores, por meio de caminhos URI. Os serviços externos podem usar métodos HTTP, como GET, PUT, POST e DELETE, nesses URIs para executar operações CRUD nos recursos.

Você pode executar várias ações por meio das APIs REST DO WFA, incluindo as seguintes:

- Acesse definições de fluxo de trabalho e metadados.
- Execute fluxos de trabalho e monitore sua execução.
- Exibir usuários e funções e alterar senhas.
- Execute e teste filtros de seleção de recursos.
- Executar e testar os localizadores de recursos.

- Gerenciar credenciais de storage ou outros objetos de data center.
- Exibir fontes de dados e tipos de fontes de dados.

REST documentation tem mais informações sobre APIs REST:

https://wfa_server_ip:port/rest/docs

wfa_server_ip É o endereço IP do servidor WFA e a porta é o número da porta TCP que você usou para o servidor WFA durante a instalação.



O WFA verifica os tokens CSRF (Cross-Site Request Forgery) para solicitações provenientes da IU da Web. No entanto, O WFA não verifica os tokens CSRF para SOLICITAÇÕES REST recebidas provenientes de clientes REST ou software de orquestração.

Referências ao material de aprendizagem

Você deve estar ciente de certas práticas de script e programação para criar fluxos de trabalho avançados do Workflow Automation (WFA). Você pode usar material de referência para aprender sobre as opções necessárias antes de criar blocos de construção ou fluxos de trabalho DO WFA.

Windows PowerShell

O WFA usa scripts do PowerShell para operações de fluxo de trabalho. A tabela a seguir inclui referências ao material de aprendizado para o PowerShell:

Introdução ao Windows PowerShell	http://msdn.microsoft.com/en-us/library/windows/desktop/aa973757(v=vs.85).aspx
Desenvolvimento do PowerShell — ambiente de Scripting integrado (ISE)	https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/ise/introducing-the-windows-powershell-ise?view=powershell-7.2
.NET Diretrizes de nomes de Framework	http://msdn.microsoft.com/en-us/library/xzf533w0%28v=vs.71%29.aspx
Estilo de código do PowerShell	http://get-powershell.com/post/2011/04/13/Extra-Points-for-Style-when-writing-PowerShell-Code.aspx
Tente/Catch do PowerShell finalmente	http://technet.microsoft.com/en-us/library/dd315350.aspx
Variáveis Automáticas do PowerShell	http://technet.microsoft.com/en-us/library/dd347675.aspx
Relatório de erros do PowerShell	https://docs.microsoft.com/en-us/powershell/scripting/developer/cmdlet/error-reporting-concepts?view=powershell-7.2

Parâmetros comuns do PowerShell	https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_commonparameters?view=powershell-7.2
---------------------------------	---

Kit de ferramentas do Data ONTAP PowerShell

O kit de ferramentas do Data ONTAP PowerShell é fornecido junto com O WFA. Você pode usar os cmdlets do kit de ferramentas do PowerShell para invocar comandos do Data ONTAP a partir de um script do PowerShell. Para obter mais informações, consulte a Ajuda do Kit de Ferramentas do Data ONTAP PowerShell, que você pode acessar a partir do seguinte local:

`WFA_install_location\WFA\PosH\Modules\DataONTAP\webhelp\index.html`.

`WFA_install_location` É o diretório de instalação DO WFA `C:\Program Files\NetApp` e é o diretório de instalação padrão.

A tabela a seguir inclui referências a informações sobre o kit de ferramentas do Data ONTAP PowerShell:

Artigos do kit de ferramentas do ONTAP PowerShell	https://community.netapp.com/t5/Tech-OnTap-Articles/The-Data-ONTAP-PowerShell-Toolkit/ta-p/85933
Comunidade NetApp do Toolkit do ONTAP PowerShell	https://community.netapp.com/t5/forums/filteredbylabelpage/board-id/microsoft-cloud-and-virtualization-discussions/label-name/powershell%20toolkit

Perl

O WFA suporta comandos Perl para operações de fluxo de trabalho. Quando você instala O WFA, os módulos Perl e Perl necessários são instalados no servidor WFA.

["Guia do Usuário ActivePerl"](#)

Você também pode acessar o *ActivePerl User Guide* a partir do seguinte local:

`WFA_install_location\WFA\Perl64\html\index.html`.

`WFA_install_location` É o diretório de instalação DO WFA, e `C:\Program Files\NetApp` is the default installation directory. "WFA usa scripts Perl para operações de fluxo de trabalho. A tabela a seguir inclui referências ao material de aprendizagem para Perl:

Perl moderno: 2014	http://modernperlbooks.com/books/modern_perl_2014/index.html
Documentação de programação Perl	http://perldoc.perl.org/
Linguagem de programação Perl	http://www.perl.org/

SDK de gerenciamento do NetApp

Os módulos Perl necessários do SDK de gerenciamento do NetApp são empacotados junto com O WFA. Esses módulos Perl são necessários para usar os comandos Perl no WFA. Para obter mais informações, consulte a documentação do SDK de gerenciamento do NetApp, que você pode acessar a partir do seguinte local: `WFA_install_location\WFA\perl\NMSDK\html`.

`WFA_install_location` É o diretório de instalação DO WFA `C:\Program Files\NetApp` e é o diretório de instalação padrão.

Linguagem de consulta estruturada (SQL)

A sintaxe DE SELEÇÃO SQL é usada em filtros e para preencher as entradas do usuário.

["MYSQL SELECIONE sintaxe"](#)

Linguagem de expressão MVFLEX (MVEL)

Você pode usar sintaxe de expressão MVEL em fluxos de trabalho WFA - por exemplo, em funções e variáveis.

Para obter mais informações, consulte o *MVEL Language Guide*.

Expressões regulares

Você pode usar expressão regular (regex) no WFA.

["ActionScript 3,0 usando expressões regulares"](#)

Documentação relacionada para o OnCommand Workflow Automation

Existem documentos e ferramentas adicionais para ajudá-lo a aprender a executar uma configuração mais avançada do seu servidor OnCommand Workflow Automation (WFA).

Outras referências

O espaço de automação do fluxo de trabalho na comunidade NetApp fornece recursos de aprendizado adicionais, incluindo os seguintes:

- **Comunidade NetApp**

["Comunidade NetApp: Automação de fluxo de trabalho \(WFA\)"](#)

Referências da ferramenta

- **Matriz de interoperabilidade**

Lista combinações suportadas de componentes de hardware e versões de software.

["Matriz de interoperabilidade"](#)

Informações sobre direitos autorais

Copyright © 2025 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTE DOCUMENTO. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSAENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTE SOFTWARE, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.