



ワークフロー

OnCommand Workflow Automation

NetApp
October 09, 2025

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

目次

ワークフロー	1
OnCommand Workflow Automation の概要	1
WFA の機能	1
WFA ライセンス情報	2
Workflow Automation Designer の概要を参照してください	2
OnCommand Workflow Automation でのビルディングブロックの操作	2
プレイグラウンドデータベースとは	10
ワークフローの管理	11
定義済みのワークフローをカスタマイズする	12
ワークフローの作成	13
WFA ワークフローパックを作成します	39
WFA ワークフローパックを SCM リポジトリと統合する	45
ワークフローのビルディングブロックの作成	49
データソースタイプを作成します	49
コマンドを作成します	50
Finder を作成します	53
フィルタを作成します	54
ディクショナリエントリを作成します	55
関数を作成します	56
テンプレートを作成します	57
キャッシュクエリを作成します	57
定期的なスケジュールを作成	58
フィルタルールを定義します	59
承認ポイントを追加します	60
WFA のコーディングガイドライン	61
変数のガイドライン	61
インデントのガイドライン	65
コメントのガイドライン	66
ロギングのガイドライン	68
エラー処理のガイドライン	69
WFA での PowerShell と Perl の一般的な規則	72
カスタムの PowerShell モジュールと Perl モジュールを追加する場合の考慮事項	73
WFA のコマンドレットと機能	74
PowerShell および Perl WFA モジュール	74
PowerShell コマンドを Perl に変換する際の考慮事項	77
WFA のビルディングブロックに関するガイドライン	79
予約語	93
REST API の使用方法	94
学習資料への参照	95

Windows PowerShell の場合	95
Data ONTAP PowerShell ツールキット	96
Perl の場合	96
NetApp Manageability SDK の使用	97
Structured Query Language (SQL ; 構造化クエリ言語)	97
MVFLEX 表現言語 (MVEL)	97
正規表現	97
OnCommand Workflow Automation の関連ドキュメント	97
その他の参照	97
ツール参照	98

ワークフロー

OnCommand Workflow Automation の概要

OnCommand Workflow Automation (WFA) は、プロビジョニング、移行、運用停止、データ保護設定などのストレージ管理タスクの自動化に役立つソフトウェア解決策です。およびストレージのクローニングWFA を使用すると、プロセスで指定されたタスクを実行するためのワークフローを構築できます。

ワークフローは繰り返し実行される手順のタスクで、次の種類のタスクを含む一連の手順で構成されます。

- ・データベースまたはファイルシステム用のストレージのプロビジョニング、移行、または運用停止
- ・ストレージスイッチやデータストアなど、新しい仮想化環境をセットアップする
- ・エンドツーエンドのオーケストレーションプロセスの一環としてアプリケーション用のストレージをセットアップする

ストレージアーキテクトは、次のような、ベストプラクティスに従い、組織の要件を満たすワークフローを定義できます。

- ・必要な命名規則を使用します
- ・ストレージオブジェクトに一意のオプションを設定しています
- ・リソースを選択する
- ・内部構成管理データベース (CMDB) とチケット処理アプリケーションを統合する

WFA の機能

- ・ワークフローを構築するためのワークフロー設計ポータル

ワークフロー設計ポータルには、コマンド、テンプレート、ファインダ、フィルタ、ワークフローの作成に使用される関数です。設計者は、自動リソース選択、行の繰り返し (ループ) 、承認ポイントなどの高度な機能をワークフローに含めることができます。

ワークフローデザインポータルには、外部システムからデータをキャッシュするための、ディクショナリエントリ、キャッシュクエリ、データソースタイプなどのビルディングブロックも含まれています。

- ・実行ポータル：ワークフローの実行、ワークフローの実行ステータスの確認、ログへのアクセスを行います
- ・WFA の設定、データソースへの接続、ユーザクレデンシャルの設定などのタスクの管理 / 設定オプション
- ・Web サービスインターフェイスを使用して、外部ポータルやデータセンターオーケストレーションソフトウェアからワークフローを起動できます
- ・Storage Automation Store で WFA パックをダウンロードしてください。ONTAP 9.7.0 パックは WFA 5.1 にバンドルされています。

WFA ライセンス情報

OnCommand Workflow Automation サーバを使用するために必要なライセンスはありません。

Workflow Automation Designer の概要を参照してください

Workflow Automation (WFA) 設計者が、ファインダ、フィルタ、コマンドなどのビルディングブロックを使用してワークフローを作成します。ワークフローの作成を開始する前に、ビルディングブロックとワークフロー作成プロセスについて理解することが重要です。

OnCommand Workflow Automation でのビルディングブロックの操作

Workflow Automation (WFA) のワークフローは複数のビルディングブロックで構成されており、WFA には定義済みのビルディングブロックのライブラリが含まれています。WFA のビルディングブロックを使用して、組織の要件に合ったワークフローを作成できます。

WFA は、ストレージ自動化プロセスの構造を提供します。WFA の柔軟性は、ワークフロービルディングブロックを使用してワークフローをどのように構築するかによって決まります。

WFA のビルディングブロックは次のとおりです。

- ・辞書エントリ
- ・コマンド
- ・フィルタ
- ・ファインダ
- ・機能
- ・テンプレート

WFA でビルディングブロックがどのように使用されるかを理解しておくと、ワークフローの作成に役立ちます。

データソースとは

データソースは、データを抽出するために、他のシステム、ファイル、およびデータベースとの接続を確立する手段の 1 つです。たとえば、データソースは、Active IQ Unified Manager 9.4 データソースタイプの Active IQ Unified Manager データベースに接続できます。

キャッシング方式、必要なポート、およびデータソースタイプとデータソースタイプを関連付けることで、データ収集に必要なデータソースタイプを定義したあとで、OnCommand Workflow Automation (WFA) にカスタムデータソースを追加できます。

WFA は、さまざまなデータソースを介して情報をキャッシングします。WFA は、データソースからリソース情報を収集し、キャッシング方式に対応するようにフォーマットします。キャッシングスキーム内のテーブルである

キャッシュテーブルは、ディクショナリエントリオブジェクトと一致するようにフォーマットされます。ワークフローで Finder を使用すると、ディクショナリオブジェクトが返され、ディクショナリオブジェクトのデータがキャッシュテーブルから取り込まれます。データソースからデータを取得するプロセスを、_data source acquisition_ と呼びます。データソースの取得には、スクリプトベースまたはドライバベースの方法を使用できます。ソース間で違いが生じることがあり、データソースの取得時に異なる間隔でサンプルが収集されることがあります。WFA はその情報をデータベースにマージし、予約データをスーパーインポーズしてデータベース内の更新されたリソース情報を維持します。

WFA データベースには、いくつかの異なるキャッシュ方式があります。キャッシュスキームはテーブルのセットで、各テーブルには特定のディクショナリエントリタイプの情報が含まれます。ただし、テーブルには、特定のデータソースタイプの複数のソースからの情報が結合されている場合があります。WFA では、データベース情報を使用して、リソースのステータスを把握し、計算を実行し、リソースに対してコマンドを実行します。

ディクショナリエントリとは

ディクショナリエントリは、OnCommand Workflow Automation (WFA) のビルディングブロックの 1 つです。ディクショナリエントリを使用して、ストレージ環境とストレージ関連環境におけるオブジェクトタイプとその関係を表すことができます。その後、ワークフローでフィルタを使用して、ディクショナリエントリの自然キーの値を返すことができます。

ディクショナリエントリは、WFA でサポートされるオブジェクトタイプの定義です。各ディクショナリエントリは、サポートされているストレージおよびストレージ関連の環境で、オブジェクトタイプとその関係を表します。ディクショナリオブジェクトは、タイプチェックの可能性がある属性のリストで構成されます。完全な値を持つディクショナリオブジェクトは、型のオブジェクトインスタンスを記述します。また、参照属性は、オブジェクトと環境との関係を記述します。たとえば、ボリューム辞書オブジェクトには name、size_MB、volume_guarantee などの多数の属性があります。また、ボリュームディクショナリオブジェクトには、アグリゲートへの参照、およびボリュームを含むアレイへの参照が array_id および aggregate_id の形式で含まれます

オブジェクトのキャッシュテーブルは、キャッシュ用にマークされたディクショナリエントリの属性の一部またはすべてを含むデータベースです。ディクショナリエントリにキャッシュテーブルを含めるには、少なくとも 1 つのディクショナリエントリの属性をキャッシュ用にマークする必要があります。ディクショナリエントリには、オブジェクトの一意の識別子であるナチュラルキーが含まれます。たとえば、7-Mode ボリュームは、それらのボリュームを含むアレイの名前と IP アドレスによって一意に識別されます。qtree は、qtree 名、ボリューム名、およびアレイ IP アドレスによって識別されます。ディクショナリエントリを作成するときに、ディクショナリエントリのナチュラルキーの一部であるディクショナリ属性を識別する必要があります。

コマンドの仕組み

OnCommand Workflow Automation コマンドは、ワークフローの実行ブロックです。ワークフローの各ステップにコマンドを使用できます。

WFA コマンドは、PowerShell スクリプトと Perl スクリプトを使用して記述します。PowerShell コマンドでは、パッケージがインストールされている場合、Data ONTAP PowerShell Toolkit と VMware PowerCLI を使用します。Perl コマンドでは、WFA サーバにインストールされている Perl モジュールと Perl モジュールを使用します。PowerShell や Perl などのコマンドに複数のスクリプト言語を含める場合は、WFA のインストール先のオペレーティングシステムと、WFA の設定メニューで指定した言語の優先順位に基づいて、適切なスクリプトが WFA によって選択されます。

WFA コマンドのスクリプトには、いくつかのパラメータが含まれています。これらのパラメータは、ディクショナリエントリ属性にマッピングされる場合があります。

WFA コマンドには、それぞれ複数の Data ONTAP コマンドを含めることができます。

WFA コマンドの中には、実行時間の長い処理を待機し、定期的にポーリングすることができるものもあるため、wait コマンドと呼ばれるものもあります。たとえば、「* wait for multiple volume Moves *」コマンドなどです。ポーリングコマンドが実行される待機間隔は、処理が完了したかどうかを確認するように設定できます。

WFA コマンドは、ワークフローの実行フェーズ中に WFA によって開始されます。WFA は、コマンドを左から右、上から下の順に実行します。ワークフローを計画することで、コマンドに指定するパラメータを使用できるかどうかと、その有効性を確認できます。コマンドの実行前に、WFA サーバからコマンドに必要なすべてのパラメータが提供されます。

コマンドのパラメータは、ワークフローの計画中に確定します。ワークフローは、実行時にこれらのパラメータをコマンドに渡します。コマンドからワークフローにパラメータを戻すことはできません。ただし、ワークフローのコマンド間での実行時に取得した情報を交換する場合は、指定の WFA PowerShell コマンドレットまたは Perl の機能を使用できます。

WFA PowerShell コマンドでは、PowerShell コマンドレットの -ErrorAction stop フラグは使用されません。そのため、エラーが原因でコマンドレットが失敗した場合でもワークフローは続行されます。ErrorAction stop フラグを特定のコマンドに指定する場合は、コマンドをクローニングして PowerShell スクリプトを変更し、フラグを追加します。

WFA に含まれている PowerShell コマンドと Perl 機能を使用して、各コマンド間で情報を交換することができます。

PowerShell コマンドレット	Perl 関数
Add-WfaWorkflowParameter	addWfaWorkflowParameter
Get-WfaWorkflowParameter	getWfaWorkflowParameter

コマンドに「add」コマンドレットまたは関数によって追加されたパラメータは、後で実行され、「get」コマンドレットまたは関数を使用するコマンドによって取得できます。たとえば、PowerShell WFA コマンドのコードで次のように指定すると、volumeID というパラメータを追加できます。Add-WfaWorkflowParameter -Name "VolumeUUID" -value "12345" -AddAsReturnParameter \$true"次に、このコマンドのあとこのコマンドで次のように指定すると、ボリューム ID の値が取得されます。「\$volumeID = Get-WfaWorkflowParameter -Name volumeID」。

WFA コマンドは WFA データベースを照会して、必要な結果を取得できます。これにより、フィルタやファインダを使用せずにコマンドを作成できます。次の関数を使用して、データベースを照会できます。

PowerShell コマンドレット	Perl 関数
例：「Invoke-NaMySqlQuery -Query」 cm_storage.cluster から「Cluster Name」に cluster.name を選択します	invokeMySqlQuery

- 関連情報 *

[フィルタを作成します](#)

[Finder を作成します](#)

[コマンドを作成します](#)

フィルタとは

ワークフローの WFA フィルタを使用して、必要なリソースを選択できます。

WFA フィルタは、WFA データベースに対して機能する SQL ベースのクエリです。各フィルタは、特定のディクショナリタイプの要素のリストを返します。返されるエレメントは、SQL クエリで指定された選択基準に基づいています。フィルタを作成または編集するには、SQL 構文に注意する必要があります。

ファインダとは

Finder は、共通の結果を識別するために一緒に使用される 1 つ以上のフィルタの組み合わせです。ワークフローの Finder を使用して、ワークフローの実行に必要なリソースを選択できます。

ファインダでは、ソート順序を適用して該当する結果を区別できます。ファインダは、選択条件とソートに基づいて最適なリソースを返します。

ファインダは 1 つの結果を返すため、またはまったく結果を返しません。そのため、特定のストレージ要素の有無を確認するために使用できます。ただし、繰り返し行定義の一部として Finder を使用する場合は、結果セットを使用してグループ内のメンバーのリストが形成されます。ファインダで使用されるフィルタは、ディクショナリタイプの自然キーを最小値で返しますが、値を参照できる追加のフィールドを返すことがあります。ソート順序は、フィルタの SQL クエリの戻りフィールドに適用される場合があります。

Finder の結果をテストできます。Finder をテストすると、すべての WFA フィルタの一般的な結果を確認できます。この場合、Finder の有効な結果が結果で強調表示されます。ワークフローで Finder を使用する場合は、ストレージオペレータに意味のある情報を伝えるカスタマイズされたエラーメッセージを作成できます。

機能とは

ワークフローの計画フェーズで完了する必要のある複雑なタスクに対して、ワークフロー内の関数を使用できます。

MVFLEX 式言語（MVEL）を使用して関数を記述できます。関数を使用すると、一般的に使用されるロジックと、名前付き関数内により複雑なロジックをまとめて、それをコマンドパラメーターまたはフィルタパラメーターの値として再利用できます。関数を一度作成し、ワークフロー全体で使用できます。関数を使用すると、複雑な命名規則の定義など、複雑な繰り返し作業や作業を処理することができます。

関数は実行中に他の関数を使用する場合があります。

方式は何ですか

スキームはシステムのデータモデルを表します。データモデルは、ディクショナリエンティのコレクションです。スキームを定義してから、データソースタイプを定義できます。データソースは、データの取得方法とスキームの設定方法を定義します。たとえば、VC スキームは、仮想マシン、ホスト、データストアなどの仮想環境に関するデー

タを取得します。

スキームには、特定の問題を解決するようにカスタマイズされたワークフローを通じて、データを直接入力することもできます。

ディクショナリエントリは、ディクショナリエントリが作成されるときに、既存のスキームに関連付けられます。ディクショナリエントリはキャッシュクエリーにも関連付けられ、キャッシュクエリーには SQL クエリーが含まれます。

スキームでは、スクリプトベースのデータソースタイプまたは SQL データソースタイプのいずれかを使用してデータを取得できます。スクリプトはデータソースタイプの作成時に定義され、SQL クエリはキャッシュクエリで定義されます。

WFA には次のスキームが含まれています。

- * 7-Mode (ストレージ) *

Active IQ Unified Manager を介して Data ONTAP 7-Mode からデータを取得するスキーム。

- * clustered Data ONTAP (cm_storage) *

clustered Data ONTAP から Active IQ Unified Manager 経由でデータを取得するスキーム。

- * 7-Mode のパフォーマンス (パフォーマンス) *

Performance Advisor から Data ONTAP 7-Mode のパフォーマンスデータを取得するスキーム。

- * clustered Data ONTAP のパフォーマンス (cm_performance) *

Performance Advisor から clustered Data ONTAP のパフォーマンスデータを取得するスキーム。

- * VMware vCenter (VC) *

VMware vCenter からデータを取得するスキーム。

- * プレイグラウンド (プレイグラウンド) *

データを直接取り込むことができるスキーム。

リモートシステムの種類

OnCommand Workflow Automation (WFA) はリモートシステムタイプと通信します。WFA が通信できるリモートシステムのタイプはリモートシステムです。WFA ではリモートシステムタイプを設定できます。たとえば、Data ONTAP システムをリモートシステムタイプとして設定できます。

リモートシステムタイプには、次の属性があります。

- 名前
- 説明

- ・バージョン
- ・プロトコル
- ・ポート
- ・タイムアウト

リモートシステムのクレデンシャルを検証するには、リモートシステムタイプごとに Perl スクリプトを使用します。WFA で設定されているリモートシステムのクレデンシャルを保存できます。新しいカスタムリモートシステムタイプを追加または編集できます。既存のリモートシステムタイプをクローニングすることもできます。リモートシステムタイプは、関連付けられているシステムがない場合にのみ削除できます。

テンプレートの使用方法

WFA テンプレートは、ワークフロー内で参照用として使用することも、利用ポリシーに従うために使用することもできます。

WFA テンプレートは、オブジェクト定義の青写真として機能します。テンプレートを定義するには、オブジェクトのプロパティとオブジェクトのプロパティの値を指定します。次に、テンプレートを使用してワークフロー内のオブジェクト定義のプロパティを入力します。

テンプレートを使用する場合、テンプレートから取得した値を含むフィールドは編集できません。したがって、テンプレートを使用して、使用ポリシーの設定やオブジェクトの作成を行うことができます。テンプレートを適用した後にテンプレートとワークフローの関連付けを解除した場合、テンプレートから値が入力されたままになりますが、フィールドは編集できます。

カテゴリの使用方法

ワークフローを分類して、ワークフローをよりよく整理したり、ワークフローにアクセス制御機能を適用したりできます。

WFA ポータルの特定のグループに表示されるようにワークフローを分類できます。ワークフローカテゴリにアクセス制御機能を適用することもできます。たとえば、一部のストレージオペレータや承認者のみがワークフローの特定のカテゴリを表示できます。ストレージのオペレータや承認者は、アクセス権が付与されているカテゴリ内のワークフローのみを実行できます。

Active Directory グループは、カテゴリへのアクセス制御にも使用できます。

エンティティのバージョン管理の仕組み

コマンドやワークフローなどの OnCommand Workflow Automation エンティティはバージョン管理されています。バージョン番号を使用すると、WFA エンティティに対する変更を簡単に管理できます。

各 WFA エンティティには、「`major.minor.revision`」形式のバージョン番号が含まれています。たとえば、`1.1.20` です。バージョン番号の各部分に最大 3 衔を含めることができます。

WFA エンティティのバージョン番号を変更する前に、次のルールを確認しておく必要があります。

- ・バージョン番号を現在のバージョンから以前のバージョンに変更することはできません。
- ・バージョンの各部分は、0~999 の数値である必要があります。

- 新しい WFA エンティティは、デフォルトでは 1.0 にバージョン管理されます。
- エンティティのバージョン番号は、クローン作成時、または * 名前を付けて保存 * を使用してエンティティのコピーを保存するときに保持されます。
- WFA インストールには、エンティティの複数のバージョンを存在させることはできません。

WFA エンティティのバージョンを更新すると、その親エンティティのバージョンが自動的に更新されます。たとえば、* Create Volume * コマンドのバージョンを更新すると、* Create an NFS Volume * ワークフローが * Create Volume * コマンドの直下の親エンティティであるため、* Create Volume * コマンドのバージョンが更新されます。バージョンの自動更新は、次のように適用されます。

- エンティティのメジャーバージョンを変更すると、その直後の親エンティティのマイナーバージョンが更新されます。
- エンティティのマイナーバージョンを変更すると、その直後の親エンティティのリビジョンバージョンが更新されます。
- エンティティのリビジョンバージョンを変更しても、その直後の親エンティティのバージョンの一部は更新されません。

次の表に、WFA のエンティティとそのすぐ上の親エンティティを示します。

エンティティ (Entity)	即時親エンティティ
キャッシュクエリ	<ul style="list-style-type: none"> データソースのタイプ
テンプレート	<ul style="list-style-type: none"> ワークフロー
機能	<ul style="list-style-type: none"> ワークフロー テンプレート <p> 関数に特殊文字または大文字と小文字が混在している場合、そのすぐ上の親エンティティのバージョンは更新されない可能性があります。</p>
辞書	<ul style="list-style-type: none"> テンプレート フィルタ キャッシュクエリ コマンドを実行します スクリプトメソッドを使用するデータソースのタイプ
コマンドを実行します	<ul style="list-style-type: none"> ワークフロー
フィルタ	<ul style="list-style-type: none"> ファインダ ワークフロー

エンティティ (Entity)	即時親エンティティ
ファインダ	• ワークフロー
データソースのタイプ	なし
ワークフロー	なし

WFA では、バージョン番号の一部または完全なバージョン番号を使用してエンティティを検索できます。

親エンティティを削除した場合、子エンティティは保持され、削除のためにそのバージョンは更新されません。

エンティティをインポートする際のバージョン管理の仕組み

Workflow Automation 2.2 より前のバージョンからエンティティをインポートする場合、エンティティのバージョンはデフォルトで 1.0.0 になります。インポートしたエンティティがすでに WFA サーバに存在する場合は、インポートしたエンティティで既存のエンティティが上書きされます。

インポート時に WFA エンティティに変更される可能性がある項目を次に示します。

- エンティティのアップグレード

エンティティは新しいバージョンで置き換えられます。

- エンティティのロールバック

エンティティは以前のバージョンで置き換えられます。



エンティティのロールバックを実行すると、そのすぐ上の親エンティティのバージョンが更新されます

- 新しいエンティティのインポート



.dar ファイルからエンティティを選択的にインポートすることはできません

新しいバージョンのエンティティをインポートすると、その直後の親エンティティのバージョンが更新されます。

インポートされた親エンティティに複数の子エンティティがある場合、子エンティティに対する最高レベルの変更（メジャー、マイナー、またはリビジョン）のみが親エンティティに適用されます。次の例では、このルールの仕組みについて説明します。

- インポートされた親エンティティの場合、マイナー変更のある子エンティティとリビジョン変更のある子エンティティが存在する場合、マイナー変更が親エンティティに適用されます。

親のバージョンのリビジョン部分が増分されます。

- インポートされた親エンティティの場合、メジャー変更を持つ子エンティティが 1 つ存在し、マイナー変更を持つ子エンティティが別の子エンティティである場合、親エンティティにメジャー変更が適用されます。

親のバージョンのマイナー部分が増分されます。

インポートされた子エンティティのバージョンが親のバージョンに与える影響の例

WFA で次のワークフローを考慮してください。 "Create Volume and export using NFS-Custom" 1.0.0"

ワークフローに含まれる既存のコマンドは次のとおりです。

- ・「エクスポートポリシーの作成 - カスタム」 1.0.0
- ・「ボリュームの作成 - カスタム」 1.0.0

インポートされる '.dar' ファイルに含まれるコマンドは次のとおりです

- ・「エクスポートポリシーの作成 - カスタム」 1.1.0
- ・「ボリュームの作成 - カスタム」 2.0.0

この '.dar' ファイルをインポートすると 'NFS-Custom' ワークフローを使用してボリュームを作成およびエクスポートするというマイナーバージョンが 1.1.0 に増分されます

プレイグラウンドデータベースとは

プレイグラウンドデータベースは MySQL データベースで、 Workflow Automation (WFA) サーバのインストールに含まれています。 プレイグラウンドデータベースにテーブルを追加して情報を含めることができます。 この情報は、 フィルタやユーザ入力用の SQL クエリで使用できます。

プレイグラウンドデータベースは、 WFA Web ポータルからアクセスできないスキーマです。 データベースにアクセスするには、 SQLyog 、 Toad for MySQL 、 MySQL Workbench 、またはコマンドラインインターフェイス (CLI) などの MySQL クライアントを使用します。

プレイグラウンドデータベースにアクセスするには、次の資格情報を使用する必要があります。

- ・ユーザ名： wfa
- ・パスワード： Wfa123

クレデンシャルに基づいて、 プレイグラウンドデータベースに完全にアクセスできるほか、 WFA の MySQL データベースに定義されている他のスキーマへの読み取り専用アクセスが可能になります。 プレイグラウンドデータベースに必要なテーブルを作成できます。

環境内のストレージオブジェクトに使用するタグやメタデータをプレイグラウンドデータベース内のテーブルに追加できます。 作成したタグやメタデータは、 WFA のフィルタやユーザ入力クエリによって、 WFA の他のキャッシュテーブル内の情報と一緒に使用できます。

たとえば、 次のようなユースケースでプレイグラウンドデータベースを使用できます。

- ・ビジネスユニット (BU) 名でアグリゲートにタグを付け、 タグに基づいてボリュームをバスに割り当ててください
- ・vFiler ユニットに BU 名をタグ付けしています

- ストレージオブジェクトに地理または場所の詳細を追加しています
- データベースへのデータベース管理者アクセスの定義

たとえば、アグリゲートや vFiler ユニットなどのストレージオブジェクトのタグとして BU の名前を使用している場合、BU の名前を含むプレイグラウンドデータベースにテーブルを作成できます。BU 名は、ワークフローのフィルタおよびユーザ入力クエリで使用できます。

次に、プレイグラウンドデータベーステーブル（遊び場 .volume_bu）の例を示します。

array_ip	volume_name を使用してください	バックアップ
10.225.126.23	DATA_11	マーケティング
10.225.126.28	ARC_11	時間

次に、BU 別にボリュームをフィルタリングできる SQL クエリの例を示します。

```

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}'

```

- 関連情報 *

"SQLyog 社のサービスです"

"MySQL ワークベンチ"

"toad for mysql のように入力します"

ワークフローの管理

ワークフローの管理の一環として、定義済みワークフローをカスタマイズしたり、新しいワークフローを作成したりできます。ワークフローの管理を開始する前に、関連する概念についても理解しておく必要があります。

定義済みのワークフローをカスタマイズする

要件に適した定義済みワークフローがない場合は、定義済みの Workflow Automation (WFA) ワークフローをカスタマイズできます。

必要なもの

事前定義されたワークフローに必要な変更を特定しておく必要があります。

このタスクについて

次の項目についての質問やサポートリクエストは、WFA コミュニティに送信される必要があります。

- WFA コミュニティからダウンロードされたすべてのコンテンツ
- 作成したカスタムの WFA コンテンツ
- 変更した WFA のコンテンツ

手順

1. [* ワークフローデザイン > ワークフロー *] をクリックします。
2. 要件に適した定義済みワークフローを選択し、をクリックします  をクリックします。
3. ワークフローデザインで、概要 の編集、コマンドの追加または削除、コマンドの詳細の変更、ユーザ入力の変更など、適切なタブで必要な変更を行います。
4. [* プレビュー *] をクリックして、ワークフローの実行をプレビューするために必要なユーザー入力を入力し、[* プレビュー *] をクリックしてワークフローの計画の詳細を表示します。
5. [OK] をクリックしてプレビューウィンドウを閉じます。
6. [保存 (Save)] をクリックします。

完了後

テスト環境で変更したワークフローをテストし、ワークフローを本番環境向けの準備としてマークできます。

ボリュームの作成ワークフローと CIFS 共有ワークフローをカスタマイズします

要件に応じてワークフローをカスタマイズできます。たとえば、事前定義されたボリュームの作成ワークフローと CIFS 共有ワークフローを変更して、重複排除と圧縮を含めることができます。

このタスクについて

このタスクのカスタマイズと図は一例です。要件に基づいて WFA ワークフローを変更できます。

手順

1. [* ワークフローデザイン > ワークフロー *] をクリックします。
2. Create a Volume and a CIFS Share * (ボリュームと CIFS 共有の作成) ワークフローを選択し、をクリックします  をクリックします。
3. [* 詳細 *] タブをクリックし、[* ワークフローネーム *] フィールドでワークフローの概要を編集します。

4. [* ワークフロー] タブをクリックし、【ストレージ * スキーマ】を展開し、* ボリュームの作成 * コマンドと * CIFS 共有の作成 * コマンドの間に * セットアップ重複排除と圧縮 * コマンドをドラッグ・アンド・ドロップします。
5. 1行目の * Setup deduplic なくなり compression* コマンドの下にマウスカーソルを置き、をクリックします 。
6. 「セットアップ重複排除と圧縮のパラメータ」 * ダイアログボックスの * ボリューム * タブで、以前に定義したボリューム * オプションを使用して * を選択し、* ボリュームの定義 * フィールドで * share_volume * オプションを選択します。これは、ワークフローで * Create Volume * コマンドで作成されるボリュームオブジェクト変数です。
7. [その他のパラメータ *] タブをクリックして、次の手順を実行します。
 - a. [StartNow] フィールドで [true] を選択します。
 - b. [* Compression] フィールドで [* Inline *] を選択します。
 - c. 「スケジュール *」 フィールドに「sun -sat@1」という式を入力します。この式は、毎週のすべての曜日の午前 1 時に重複排除と圧縮をスケジュールします
8. [OK] をクリックします。
9. [* プレビュー *] をクリックして、ワークフローの計画が正常に完了したことを確認し、[OK] をクリックします。
10. [保存 (Save)] をクリックします。

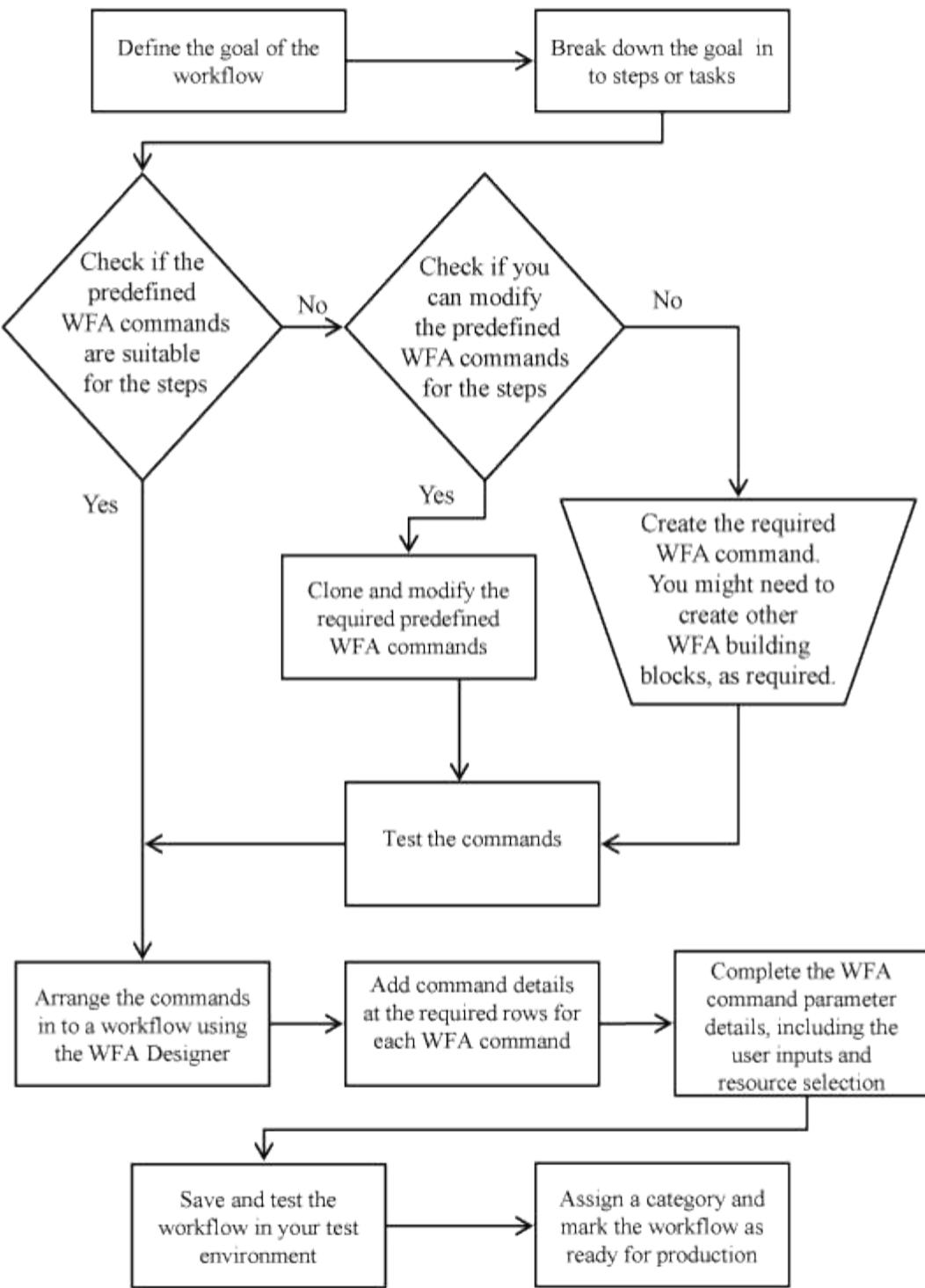
ワークフローの作成

定義済みのワークフローが要件に合わない場合は、必要なワークフローを作成できます。ワークフローを作成する前に、WFA 設計者が利用できる機能を理解し、ワークフローチェックリストを作成する必要があります。

ワークフローの作成に関するタスク

OnCommand Workflow Automation (WFA) でストレージ自動化ワークフローを作成するには、ワークフローで実行する手順を定義し、コマンド、ファインダ、フィルタ、ディクショナリエントリなどの WFA ビルディングブロックを使用してワークフローを作成します。

次のフローチャートは、ワークフローの作成プロセスを示しています。



ワークフローを定義する方法

ワークフローの目標を、ワークフローで実行する必要がある手順に分割する必要があります。その後、ワークフローを完了するための手順を並べ替えることができます。

ワークフローとは、エンドツーエンドのプロセスを実行するために必要な一連のステップを含むアルゴリズムです。プロセスの範囲は、ワークフローの目標によって異なる場合があります。ワークフローの目的は、ストレージの運用のみ、またはネットワーク、仮想化、ITシステム、その他のアプリケーションを1つのプロセスの一部として処理するなど、より複雑なプロセスを処理することにあります。OnCommand Workflow Automation (WFA) ワークフローはストレージアーキテクトが設計し、ストレージオペレータが実行しま

す。

ワークフローを定義するには、ワークフローの目標を一連の手順に分けます。たとえば、NFS ボリュームを作成するには、次の手順を実行します。

1. ボリュームオブジェクトの作成
2. 新しいエクスポートポリシーを作成し、そのポリシーをボリュームに関連付けます

ワークフローの各ステップには、WFA コマンドまたはワークフローを使用できます。WFA には、ストレージの一般的なユースケースに基づく、事前定義されたコマンドとワークフローが用意されています。特定の手順に使用できる定義済みのコマンドまたはワークフローが見つからない場合は、次のいずれかを実行できます。

- ・手順に最も近い事前定義されたコマンドまたはワークフローを選択し、要件に応じて事前定義されたコマンドまたはワークフローをクローニングして変更します。
- ・新しいコマンドまたはワークフローを作成します。

その場合は、新しいワークフローにコマンドやワークフローを配置して、目的を達成するためのワークフローを作成できます。

ワークフローの実行が開始されると、WFA は実行を計画し、ワークフローとコマンドの入力を使用してワークフローを実行できることを確認します。ワークフローを計画すると、すべてのリソース選択とユーザ入力が解決され、実行計画が作成されます。計画が完了すると、WFA は実行計画を実行します。実行計画は、一連の WFA コマンドと該当するパラメータで構成されます。

ユーザ入力の定義方法

OnCommand Workflow Automation (WFA) ユーザ入力は、ワークフローの実行中に使用できるデータ入力オプションです。ワークフローの柔軟性と使いやすさを高めるために、ワークフローにユーザ入力パラメータを定義する必要があります。

ユーザー入力は入力フィールドとして表示され、ワークフローのプレビューまたは実行時に関連データを入力できます。ワークフローでコマンドの詳細を指定するときに、ドル記号 (\$) でラベルまたは変数を事前に修正することによって、ユーザ入力フィールドを作成できます。たとえば、\$VolumeName と指定すると、Volume Name ユーザ入力フィールドが作成されます。WFA の [ワークフロー <workflow name>] ウィンドウの [ユーザ入力] タブに、作成したユーザ入力ラベルが自動的に入力されます。タイプ、表示名、デフォルト値、検証値などのユーザー入力属性を変更することにより、ユーザー入力のタイプを定義し、入力フィールドをカスタマイズすることもできます。

ユーザー入力タイプのオプション

- * 文字列 *

有効な値には正規表現 (A* など) を使用できます。

「0d」や「0f」などの文字列は、「0d` が double 型の 0 として評価される」のような数字として評価されます。

- * 番号 *

選択できる数値範囲を定義できます。たとえば、1 ~ 15 のように指定できます。

- * Enum *

列挙型を使用して、ユーザー入力フィールドに入力するときに選択できる列挙値を作成できます。必要に応じて、作成した列挙値をロックして、ユーザ入力に対して作成した値のみが選択されるようにすることができます。

- * クエリ *

クエリタイプは、WFA キャッシュの値からユーザ入力を選択するときに選択できます。たとえば、次のクエリを使用すると、ユーザ入力フィールドに WFA キャッシュの IP アドレスと名前の値が自動的に入力されます。 **select ip, name from storagegear.array** : クエリが取得した値をロックして、クエリが再試行した結果のみが選択されるようにすることもできます

- * 照会（複数選択）*

クエリ（複数選択）タイプはクエリタイプに似ており、ワークフローの実行中に複数の値を選択できます。たとえば、ユーザは、共有とエクスポートとともに複数のボリュームまたはボリュームを選択できます。複数の行を選択したり、選択を 1 行に制限したりできます。行を選択すると、選択した行のすべての列から値が選択されます。

ユーザー入力のクエリ（複数選択）タイプを使用する場合は、次の関数を使用できます。

- getSize の順にクリックします
- getValueAt
- getValueAt2D
- getValueFrom2DByRowKey
 - * ブール値 *

ブール型を使用して、ユーザー入力ダイアログボックスにチェックボックスを表示できます。ブール型は、「true」と「false」を持つユーザ入力に使用する必要があります。

- * 表 *

ユーザ入力のテーブルタイプを使用して、ワークフローの実行中に複数の値を入力するために使用できるテーブルの列ヘッダーを指定できます。たとえば、ノード名とポート名のリストを指定するためのテーブルなどです。列ヘッダーに次のいずれかのユーザ入力タイプを指定して、実行時に入力された値を検証することもできます。

- 文字列
- 番号
- 列挙（Enum）
- ブール値
- クエリ 'String' は 'カラム・ヘッダーのデフォルトのユーザー入力タイプです別のユーザー入力タイプを指定するには、[タイプ] 列をダブルクリックする必要があります。

Designer で SnapMirror ポリシーとルールの作成ワークフローを開いて、ユーザ入力タイプが「SnapMirrorPolicyRule」ユーザ入力でどのように使用されるかを確認できます。

テーブルタイプのユーザ入力を使用する場合は、次の関数を使用できます。

- getSize の順にクリックします
- getValueAt
- getValueAt2D
- getValueFrom2DByKey : デザイナで * Create を開き、 Infinite Volume を備えた Storage Virtual Machine * ワークフローを設定すると、テーブルタイプの使用方法を確認できます。
 - * パスワード *

パスワードの入力用のパスワードタイプをユーザ入力に使用できます。ユーザが入力したパスワードは暗号化され、 WFA アプリケーションとログファイルに一連のアスタリスク文字で表示されます。次の関数を使用してパスワードを復号化できます。このパスワードはコマンドで使用できます。

- Perl コマンドの場合 : WFAUtil::getWfaInputPassword (\$password)
- PowerShell コマンドの場合 : Get-WfaInputPassword-EncryptedPassword\$ password

ここで、 \$password は、 WFA からコマンドに渡される暗号化されたパスワードです。

- * 辞書 *

選択したディクショナリエントリのテーブルデータを追加できます。辞書エントリ属性は、返される属性を選択します。ワークフローの実行中に、単一の値または複数の値を選択できます。たとえば、1つまたは複数のボリュームを選択できます。デフォルトでは、単一の値が選択されています。フィルタ処理のルールを選択することもできます。ルールは、ディクショナリエントリ属性、演算子、および値で構成されます。属性には、その参照の属性も含めることができます。

たとえば、文字列「 aggr 」で始まる名前のすべてのアグリゲートを一覧表示し、使用可能なサイズが 5GB を超えるアグリゲートのルールを指定できます。グループの最初の規則は ' 属性名 ' で ' 演算子は 'starts-name' 、値は aggr です同じグループの 2 番目の規則は ' 属性 available_size_mb ' で ' 演算子は '>' で ' 値は '5000.' です

次の表に、ユーザ入力タイプに適用できるオプションを示します。

オプション	説明
検証中です	<p>ユーザ入力タイプを検証して、有効な値のみがユーザから入力されるようにすることができます。</p> <ul style="list-style-type: none"> • ユーザー入力の文字列および数値タイプは、ワークフローの実行時に入力した値で検証できます。 • 文字列タイプは正規表現で検証することもできます。 • 数値タイプは数値浮動小数点フィールドであり、指定した数値範囲を使用して検証できます。
ロック値	クエリーおよび列挙型の値をロックして、ユーザがドロップダウン値を上書きしないようにしたり、表示された値のみを選択できるようにしたりすることができます。

オプション	説明
必須としてマークしています	ワークフローの実行を続行するには、ユーザ入力を必須としてマークして、特定のユーザ入力を入力する必要があります。
グループ化	関連するユーザ入力をグループ化し、ユーザ入力グループの名前を指定できます。グループは、ユーザー入力ダイアログボックスで展開および縮小できます。デフォルトで展開するグループを選択できます。
条件の適用	条件付きユーザ入力機能を使用すると、別のユーザ入力に対して入力された値に基づいてユーザ入力の値を設定できます。たとえば 'NAS プロトコルを構成するワークフローでは 'Read/Write host lists' ユーザー入力を有効にするために ' プロトコルに必要なユーザー入力を nfs として指定できます

コマンドパラメータのマッピング方法

Workflow Automation (WFA) コマンドのパラメータは、特定のルールに基づいて特定の属性およびディクショナリエントリ参照にマッピングされます。WFA コマンドを作成または編集するときは、コマンドパラメータをマッピングするルールを理解しておく必要があります。

コマンドパラメータのマッピングは、ワークフローでコマンドの詳細を定義する方法を定義します。ワークフロー内のコマンドの詳細を指定する場合、コマンドのマッピングされたコマンドパラメータがタブに表示されます。タブの名前は、[パラメータマッピング (Parameters Mapping)] タブの [オブジェクト名 (Object Name)] 列で指定したグループ名に基づいて決まります。マッピングされていないパラメータは、ワークフローでコマンドの詳細を指定するときに [その他のパラメータ] タブに表示されます。

コマンドパラメータマッピングのルールは、コマンドカテゴリおよびワークフローエディタでのコマンドの表記方法に基づいています。

コマンドのカテゴリは次のとおりです。

- ・オブジェクトを作成するコマンド
- ・オブジェクトを更新するコマンド
- ・オブジェクトを削除するコマンド
- ・オプションの親オブジェクトおよび子オブジェクトを処理するコマンド
- ・オブジェクト間の関連付けを更新するコマンド

各カテゴリのルールは次のとおりです。

すべてのコマンドカテゴリ

コマンドパラメータをマッピングする場合は、ワークフローでのコマンドの使用方法に基づいたナチュラルパ

スを使用する必要があります。

次の例は、自然パスを定義する方法を示しています。

- ArrayIP パラメータでは、コマンドに応じて、array.ip 属性ではなく、ボリュームディクショナリエントリの aggregate.arr.ip 属性を使用する必要があります。

これは、ワークフローでボリュームを作成し、作成されたボリュームを参照して追加の手順を実行する場合に重要です。同様の例を次に示します。

- qtree ディクショナリエントリの volume-aggregate.arr.ip
 - LUN ディクショナリエントリの volume-aggregate.arr.ip
- コマンドで使用するクラスタの場合は、次のいずれかを使用する必要があります。
 - ボリュームディクショナリエントリの vserver.cluster.primary_address
 - qtree ディクショナリエントリの volume_vserver.cluster.primary_address

オブジェクトを作成するコマンド

このカテゴリのコマンドは、次のいずれかに使用されます。

- 親オブジェクトの検索と新規オブジェクトの定義
- オブジェクトを検索し、存在しない場合はオブジェクトを作成します

このカテゴリのコマンドには、次のパラメータマッピングルールを使用する必要があります。

- 作成されたオブジェクトの関連パラメータをオブジェクトのディクショナリエントリにマップします。
- 作成されたディクショナリエントリのリファレンスを使用して、親オブジェクトをマッピングします。
- 新しいパラメータを追加するときは、関連する属性がディクショナリエントリに存在することを確認します。

このルールの例外シナリオを次に示します。

- 作成されたオブジェクトの中には、対応するディクショナリエントリがなく、親オブジェクトだけが該当する親ディクショナリエントリにマッピングされているものがあります。たとえば、* VIF の作成 * コマンドなどです。この場合、アレイはアレイディクショナリエントリにのみマッピングできます。
- パラメータのマッピングは必要ありません

たとえば、* Create or resize aggregate * コマンドの ExecutionTimeout パラメータは、マッピングされていないパラメータです。

このカテゴリの証明済みコマンドの例を次に示します。

- ボリュームを作成します
- LUN を作成します

オブジェクトを更新するコマンド

このカテゴリのコマンドは、オブジェクトを検索し、属性を更新するために使用されます。

このカテゴリのコマンドには、次のパラメータマッピングルールを使用する必要があります。

- 更新されたオブジェクトをディクショナリエントリにマッピングします。
- オブジェクトに対して更新されたパラメータをマッピングしないでください。

例えば、 * Set Volume State * コマンドでは、 Volume パラメーターはマップされていますが、新しい状態はマップされていません。

オブジェクトを削除するコマンド

このカテゴリのコマンドは、オブジェクトを検索して削除するために使用されます。

コマンドによって削除されたオブジェクトをディクショナリエントリにマッピングする必要があります。例えば、 * ボリュームの削除 * コマンドでは、削除するボリュームは、ボリューム辞書エントリの関連する属性と参照にマップされます。

オプションの親オブジェクトおよび子オブジェクトを処理するコマンド

このカテゴリのコマンドには、次のパラメータマッピングルールを使用する必要があります。

- コマンドの必須パラメータを、オプションのパラメータからの参照としてマッピングしないでください。

このルールは、コマンドが特定の親オブジェクトのオプションの子オブジェクトを扱う場合に適しています。この場合、子オブジェクトと親オブジェクトを明示的にマッピングする必要があります。たとえば、 * 重複排除ジョブの停止 * コマンドでは、アレイまたは特定アレイのすべてのボリュームと一緒に指定した場合に、特定のボリュームで実行中の重複排除ジョブを停止します。この場合、配列パラメータは、 Volume.1 ではなく、アレイディクショナリエントリに直接マッピングする必要があります。 Array は、このコマンドのオプションパラメータであるためです。

- 親と子の関係が論理レベルでディクショナリエントリ間に存在するが、特定のコマンドの実際のインスタンス間に存在しない場合は、それらのオブジェクトを個別にマッピングする必要があります。

たとえば、 * ボリュームの移動 * コマンドでは、ボリュームが現在の親アグリゲートから新しいデスティネーションアグリゲートに移動されます。したがって、ボリュームパラメータはボリュームディクショナリエントリにマッピングされ、デスティネーションアグリゲートパラメータは、アグリゲートディクショナリエントリには個別にマッピングされますが、 volume.aggregate.name にはマッピングされません。

オブジェクト間の関連付けを更新するコマンド

このカテゴリのコマンドでは、関連付けとオブジェクトの両方を、関連するディクショナリエントリにマッピングする必要があります。たとえば、 Add Volume to vFiler コマンドでは、 Volume パラメータと vFiler パラメータが、ボリュームおよび vFiler ディクショナリエントリの関連属性にマッピングされます。

定数の定義方法

1 つのワークフローで使用できる値を定義するために、定数を作成して使用できます。定数はワークフローレベルで定義されます。

ワークフローで使用されている定数とその値は、計画および実行中のワークフローの監視ウィンドウに表示されます。定数には一意の名前を使用する必要があります。

定数を定義する際には、次の命名規則を使用できます。

- 各単語の最初の文字に大文字を使用します。単語間にアンダースコアやスペースは使用しません
すべての用語と略語で大文字を使用する必要があります。たとえば、「ActualVolumeSizeInMB」と入力します。
- すべての文字で大文字
アンダースコアを使用して単語を区切ることができますたとえば'aggregate_Used_space_threshold'のようにします

ワークフロー定数には、次の値を指定できます。

- 数字
- 文字列
- MVEL 式

式は、ワークフローの計画フェーズと実行フェーズで評価されます。式では、ループで定義されている変数を参照しないでください。

- ユーザ入力
- 変数 (variables)

行の繰り返しの仕組み

ワークフローには、コマンドとコマンドの詳細が行に表示されます。検索条件の結果に基づいて、一定のイテレーション数または繰り返し回数に対して繰り返されるコマンドを行に指定できます。

行のコマンドの詳細を指定して、特定の回数繰り返したり、ワークフローの設計時に繰り返したりできます。ワークフローは「ワークフローの実行時または実行スケジュール時に行を繰り返す必要のある回数を指定できるように設計することもできますオブジェクトの検索条件を指定し、行のコマンドを設定して、検索条件から返されたオブジェクトの数を繰り返すことができます。特定の条件が満たされたときに行を繰り返すように設定することもできます。

行の繰り返し変数

変数リストでは、行のイテレーション中に操作できる変数を指定できます。変数には、名前、変数の初期化に使用する値、および行の繰り返しのたびに評価される MVEL 式式を指定できます。

次の図は、行の繰り返しオプションと行の繰り返し変数の例を示しています。

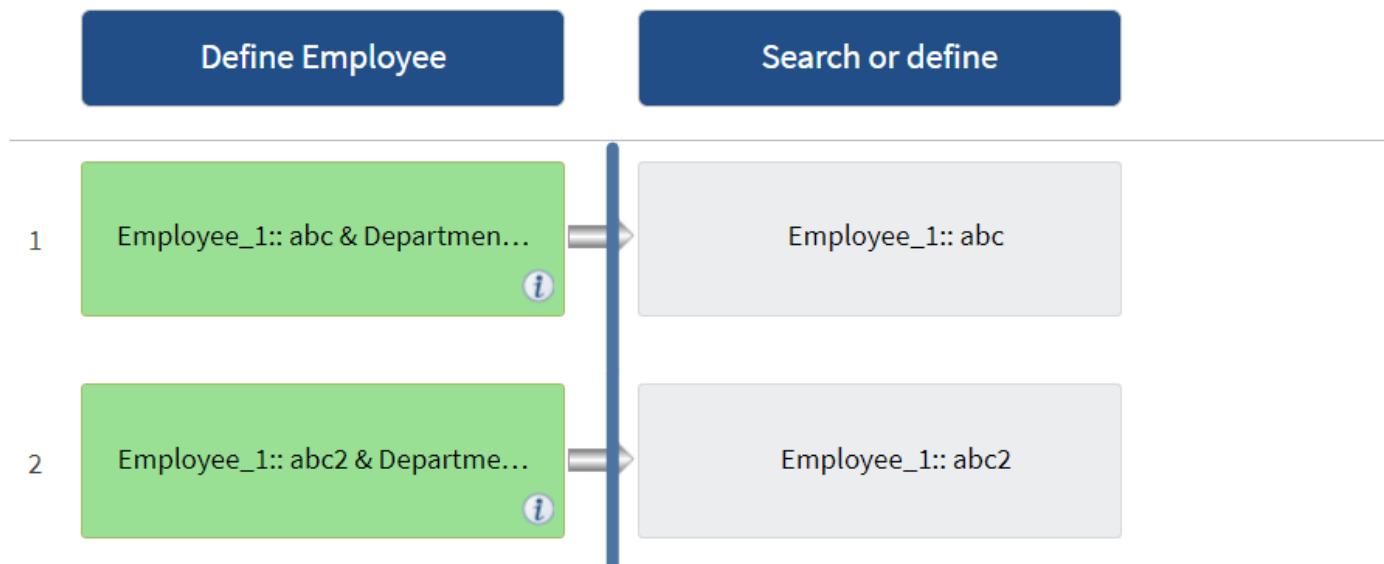
Row Repetition Details [?](#) X

Repeats*	Number of times						
Number of Times*	Number of times						
Index Variable*	Index1						
Variables	<table border="1"> <thead> <tr> <th>Name</th> <th>Initial Value</th> <th>Expression</th> </tr> </thead> <tbody> <tr> <td>size_to_alloc</td> <td>SIZE_MB</td> <td>(int)size_to_allocated - getData()</td> </tr> </tbody> </table>	Name	Initial Value	Expression	size_to_alloc	SIZE_MB	(int)size_to_allocated - getData()
Name	Initial Value	Expression					
size_to_alloc	SIZE_MB	(int)size_to_allocated - getData()					
<input type="button" value="Add"/> <input type="button" value="Remove"/>							
<input type="button" value="Cancel"/> <input type="button" value="OK"/>							

承認点を含む行の繰り返し

コマンドおよび含まれる承認ポイントに対して繰り返し行を指定すると、承認ポイントの前にコマンドのすべてのイテレーションが実行されます。承認ポイントを承認すると、次の承認ポイントまで、連続するすべてのコマンドの実行が続行されます。

次の図は、承認ポイントがワークフローに含まれている場合に繰り返し行の繰り返しがどのように実行されるかを示しています。



定義済みのワークフローで行の例を繰り返します

Designer で次の定義済みワークフローを開いて、リピート行の使用方法を理解できます。

- clustered Data ONTAP の NFS ボリュームを作成
- clustered Data ONTAP ストレージに VMware NFS データストアを作成します
- クラスタピアリングを確立する
- clustered Data ONTAP ボリュームを削除

リソース選択の仕組み

OnCommand Workflow Automation (WFA) では、検索アルゴリズムを使用して、ワークフローの実行に使用するストレージリソースを選択します。ワークフローを効率的に設計するには、リソースの選択の仕組みを理解しておく必要があります。

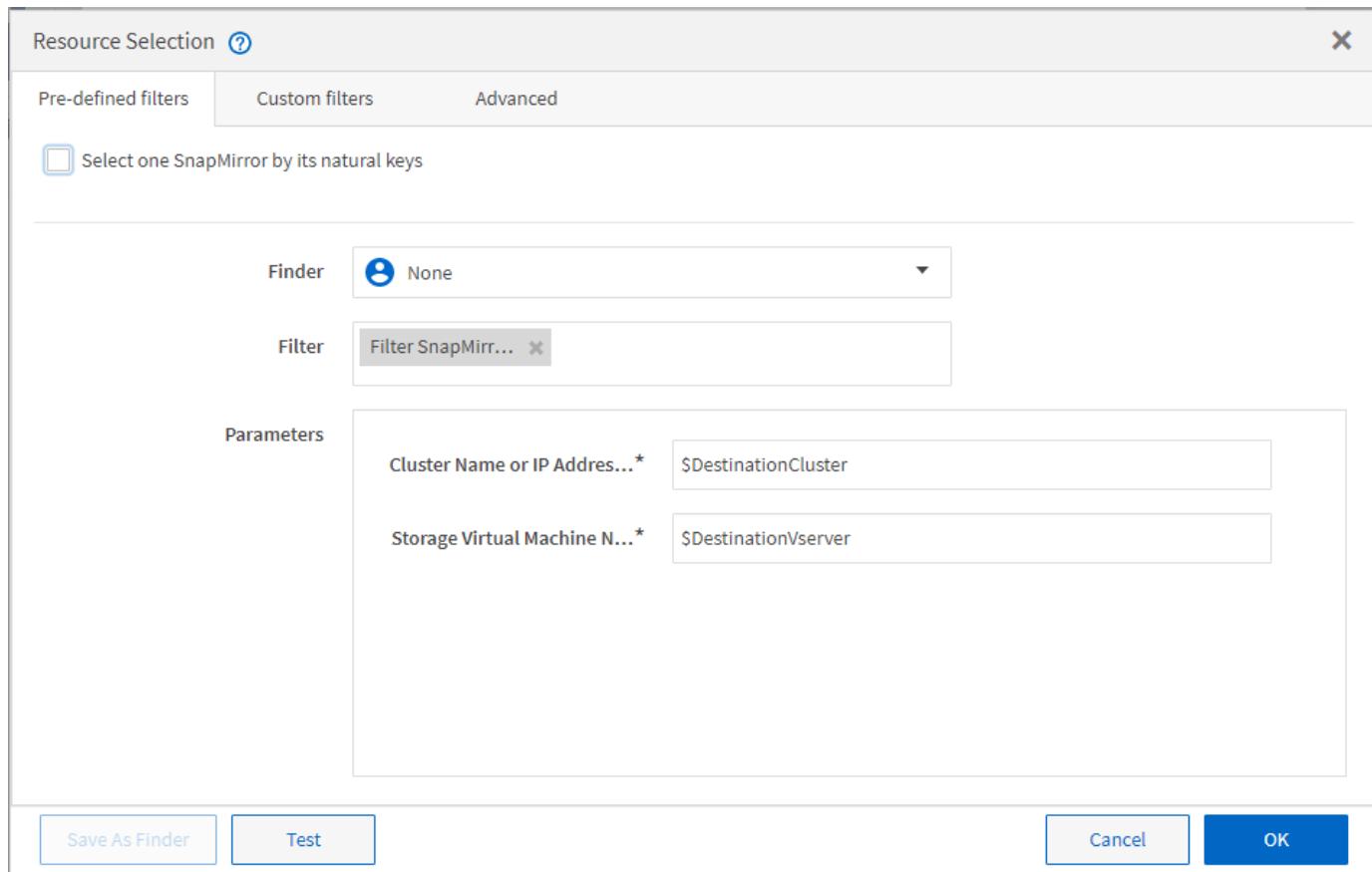
WFA では、検索アルゴリズムを使用して、vFiler ユニット、アグリゲート、仮想マシンなどのディクショナリエントリのリソースが選択されます。選択したリソースを使用してワークフローが実行されます。WFA 検索アルゴリズムは WFA ビルディングブロックの一部であり、ファインダとフィルタが含まれています。必要なリソースを特定して選択するために、検索アルゴリズムでは、Active IQ Unified Manager、VMware vCenter Server、データベースなど、さまざまなリポジトリからキャッシュされたデータを検索します。デフォルトでは、すべてのディクショナリエントリで、自然キーに基づいてリソースを検索するためのフィルタが使用できます。

ワークフロー内の各コマンドのリソース選択基準を定義する必要があります。また、Finder を使用して、ワークフローの各行にリソース選択条件を定義することもできます。たとえば、特定の量のストレージスペースを必要とするボリュームを作成する場合、「ボリュームの作成」コマンドで「使用可能な容量でアグリゲートを検索」ファインダを使用すると、指定した量の使用可能なスペースを持つアグリゲートを選択してボリュームを作成できます。

vFiler ユニット、アグリゲート、仮想マシンなど、ディクショナリエントリリソース用のフィルタルールのセットを定義できます。フィルタルールには、1 つ以上のルールグループを含めることができます。ルールは、ディクショナリエントリ属性、演算子、および値で構成されます。属性には、その参照の属性も含めることができます。たとえば、次のようにアグリゲートのルールを指定できます。List all aggregates that have names starting with the string "aggr>" and have more than 5GB available space 確保。グループの最初のルールは属性 "name" で、演算子 "starts-with", および値 "aggr" です。同じグループの 2 番目の規則は '属性 "available_size_MB" で '演算子 ">" と値 "5000" です' 一連のフィルタルールとパブリックフィルタを定義できます。Finder を選択した場合は、「フィルタルールを定義」オプションが無効になります。フィルタルールを定義 (Define filter rules) チェックボックスを選択した場合、Finder として保存 (Save As Finder) オプションは無効になります。

フィルタやファインダのほかに、検索コマンドや定義コマンドを使用して、使用可能なリソースを検索することができます。検索コマンドまたは定義コマンドは、No-op コマンドよりも推奨されます。検索および定義コマンドを使用して、証明済みディクショナリエントリタイプとカスタムディクショナリエントリタイプの両方のリソースを定義できます。検索コマンドまたは定義コマンドではリソースが検索されますが、リソースに対する操作は実行されません。ただし、リソースの検索に Finder を使用する場合は、コマンドのコンテキストで Finder を使用し、コマンドで定義されたアクションがリソースに対して実行されます。検索コマンドまたは定義コマンドによって返されるリソースは、ワークフロー内の他のコマンドの変数として使用されます。

次の図は、リソースの選択にフィルタが使用されていることを示しています。



定義済みワークフローでのリソース選択の例

Designer で次の定義済みワークフローのコマンドの詳細を開き、リソース選択オプションの使用方法を理解できます。

- clustered Data ONTAP の NFS ボリュームを作成
- クラスタピアリングを確立する
- clustered Data ONTAP ボリュームを削除

予約の仕組み

OnCommand Workflow Automation のリソースリザベーション機能では、ワークフローを正常に実行するために必要なリソースが確保されています。

WFA のコマンドを使用すると、WFA キャッシュデータベースにリソースを追加したあと、通常はキャッシュを取得したあとに、必要なリソースをリザーブしたり予約を解除したりできます。リザベーション機能を使用すると、WFA の設定で設定したリザベーションの有効期限まで、リザーブリソースをワークフローに確実に割り当てることができます。

リザベーション機能を使用すると、リソースの選択時に他のワークフローで予約されているリソースを除外できます。たとえば、あるアグリゲート上に 100GB のスペースをリザーブしたワークフローが 1 週間後に実行されるようにスケジュール設定されている場合、また、* Create Volume * コマンドを使用して別のワークフローを実行している場合、実行中のワークフローは、スケジュールされたワークフローによって予約されたスペースを消費して新しいボリュームを作成することはありません。また、リザベーション機能を使用すると、ワークフローを並行して実行することができます。

実行用のワークフローをプレビューするとき、WFA プランナーは、キャッシュデータベース内の既存のオブジェクトを含むすべてのリザーブオブジェクトを考慮します。リザベーションを有効にした場合、スケジュールされたワークフローおよび並行して実行されるワークフローの影響、およびワークフローを計画する際にストレージ要素の有無が考慮されます。

次の図の矢印は、ワークフローで予約が有効になっていることを示しています。

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					

事前定義されたワークフローでの予約の例

設計者で次の定義済みワークフローを開いて、予約の使用方法を理解できます。

- ・ クローン環境
- ・ clustered Data ONTAP ボリュームを作成
- ・ クラスタピアリングを確立する
- ・ clustered Data ONTAP ボリュームを削除

増分命名とは何ですか

増分命名とは、パラメータの検索結果に基づいてワークフロー内の属性に名前を付けるためのアルゴリズムです。属性には、増分値またはカスタム式に基づいて名前を付けることができます。命名機能が強化され、要件に基づいた命名規則を実装できます。

ワークフローを設計する際に増分の命名機能を使用すると、ワークフローで作成されたオブジェクトに動的に名前を付けることができます。この機能を使用すると、リソース選択フィーチャーを使用してオブジェクトの検索条件を指定でき、検索条件によって返される値がオブジェクトの属性に使用されます。また、指定された検索条件でオブジェクトが見つからなかった場合は、属性の値を指定できます。

属性に名前を付けるには、次のいずれかのオプションを使用します。

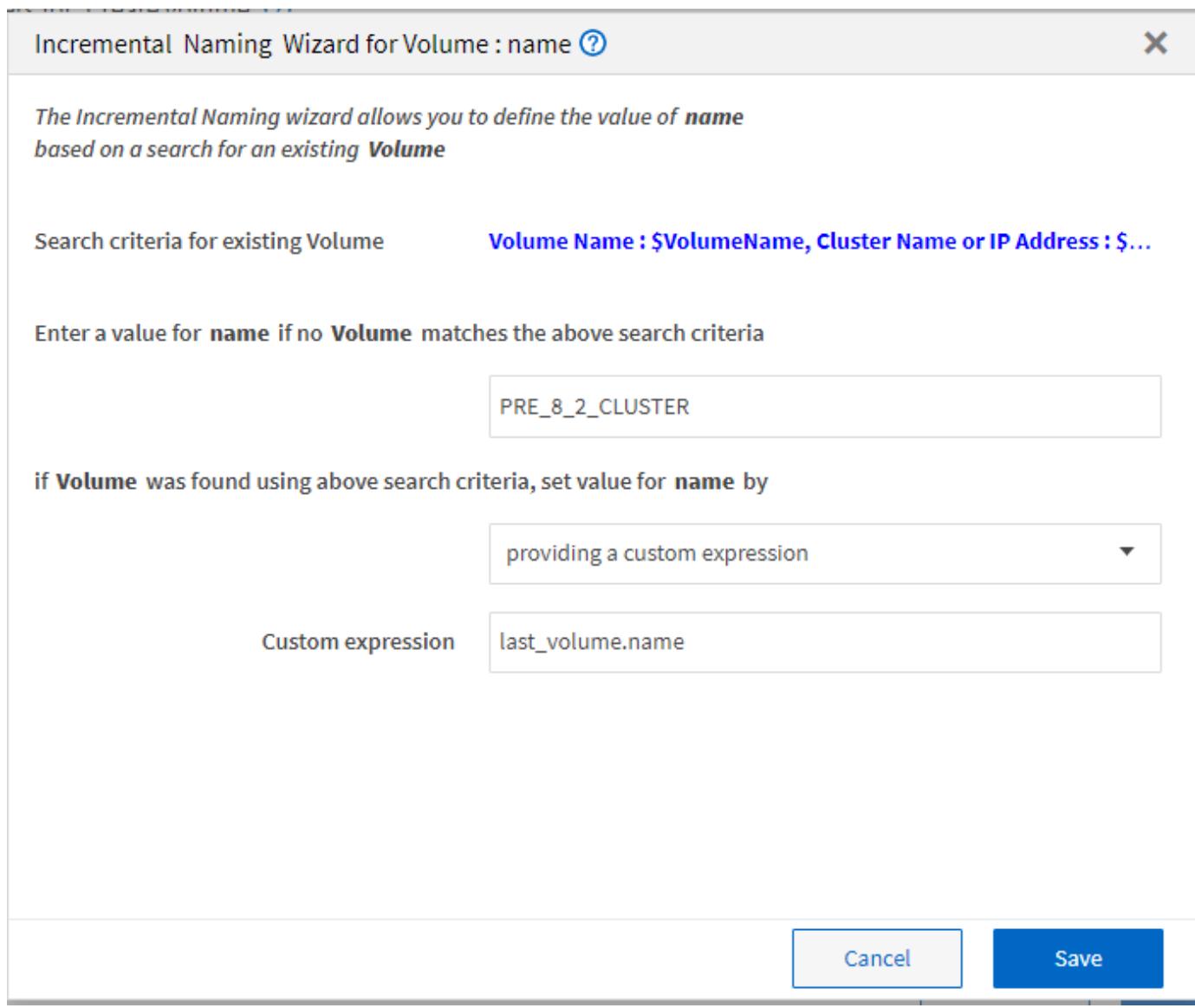
- 増分値とサフィックスを指定します

検索条件で検出されたオブジェクトの値とともに使用する値を指定し、指定した数だけ増分することができます。たとえば 'filer name_unique number_environment という命名規則を使用してボリュームを作成する場合 'Finder を使用して最後のボリュームを名前のプレフィックスで検索し '一意の番号を 1 ずつ増やして 'ボリューム名にサフィックス名を追加できます最後に見つかったボリューム名の接頭辞が _VF_023_prod で、3 つのボリュームを作成している場合、作成されるボリュームの名前は _VF_024_prod、VF_025_prod、および VF_026_prod です。

- カスタム式を指定する

検索条件で検出されたオブジェクトの値とともに使用する値を指定し、入力した式に基づいて値を追加できます。たとえば 'last volume name_environment name が 1 でパディングされたボリュームを作成する場合 " は 'last_volume.name +'' + nextName("lab1") という式を入力できます見つかった最後のボリューム名が VF_023 の場合、作成されたボリュームの名前は VF_023_lab2 です。

次の図に、命名規則を指定するカスタム式を指定する方法を示します。



条件付き実行とは

条件付き実行は、指定された条件が満たされたときにコマンドを実行できるワークフローを設計するのに役立ちます。

ワークフロー内のコマンドの実行は動的に実行できます。各コマンドの実行条件、またはワークフロー内のコマンド行を指定できます。たとえば '特定のデータセットが検出された場合にのみ 'Add volume to dataset' コマンドを実行し 'データセットが見つからない場合にワークフローを失敗させないようにすることができますこの場合は、「Add volume to dataset」コマンドを有効にして特定のデータセットを検索し、見つからない場合はワークフローでコマンドを無効にできます。

コマンドの条件付き実行のオプションは '辞書オブジェクトタブ' およびパラメータのコマンド用パラメータダイアログの詳細タブで使用できます

ワークフローを中止したり、ワークフロー内の特定のコマンドを無効にしたりできます。また、次のいずれかのオプションを使用して実行するようにコマンドを設定することもできます。

- ・条件なし
- ・指定した変数が見つかった場合
- ・指定した変数が見つからない場合
- ・指定した式が true の場合

また、特定の時間間隔を待機するようにコマンドを設定することもできます。

定義済みワークフローでの条件付き実行の例

Designer で次の定義済みワークフローのコマンド詳細を開き、コマンドの条件付き実行の使用方法を理解できます。

- ・基本的な clustered Data ONTAP ボリュームを作成
- ・clustered Data ONTAP の NFS ボリュームを作成

戻りパラメータの仕組み

戻りパラメータは、ワークフローの計画フェーズのあとに使用できるパラメータです。これらのパラメータから返される値は、ワークフローのデバッグに役立ちます。戻りパラメータの仕組みと、デバッグワークフローへの戻りパラメータとして使用できるパラメータについて理解しておく必要があります。

ワークフローでは、変数属性、式、ユーザ入力値などの一連のパラメータを戻りパラメータとして指定できます。ワークフローの実行中に、指定したパラメータの値が計画フェーズで入力され、ワークフローの実行が開始されます。これらのパラメータの値は、ワークフローのその特定の実行での計算方法で返されます。ワークフローをデバッグする場合は、パラメータから返された値を参照します。

ワークフロー内の必須の戻りパラメータを指定すると、これらのパラメータの計算値または選択した値を確認できます。たとえば、リソース選択ロジックを使用してワークフローでアグリゲートを選択する場合、戻りパラメータとしてアグリゲートを指定することで、ワークフローの計画時に選択されたアグリゲートを確認できます。

ワークフローをデバッグするための戻りパラメータの値を参照する前に、ワークフローの実行が完了している

ことを確認する必要があります。戻りパラメータの値は、ワークフローの実行ごとに設定されます。ワークフローをいくつか実行したあとに戻りパラメータを追加した場合、そのパラメータの値は、パラメータの追加後にのみ実行できます。

戻りパラメータとして使用できるパラメータ

パラメータを返します	例
スカラである変数属性	「ボリューム名」変数の属性である volume1.name、
定数	max_volume_size
ユーザ入力	\$clusterName の略
変数属性、定数、およびユーザー入力を含む MVEL 式	volume1.name+'-'+\$clusterName
コマンドの実行時に追加する戻りパラメータ	PowerShell コマンドで次の行を使用すると、\$volumeUUID パラメータが戻りパラメータとして追加されます。 Add-WfaWorkflowParameter -Name "VolumeUUID" -value "12345"-AddAsReturnParameter \$true

定義済みワークフローの戻りパラメータの例

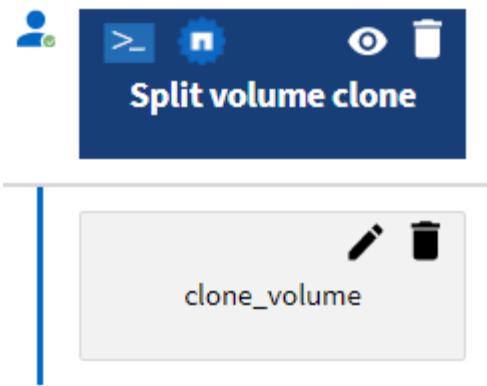
戻りパラメータの指定方法を理解するには、 Designer で次の定義済みワークフローを開き、指定した戻りパラメータを確認します。

- vFiler に NFS ボリュームを作成します
- vFiler に qtree CIFS 共有を作成します
- clustered Data ONTAP ボリュームの CIFS 共有を作成

どの承認ポイントがあるか

承認ポイントは、ワークフローでワークフローの実行を一時停止し、ユーザーの承認に基づいて再開するために使用されるチェックポイントです。

次の図に示す青色の垂直バーは承認ポイントです。



承認ポイントを使用して、ワークフローのセクションを特定の条件が満たされた後にのみ実行する必要がある場合に、ワークフローの段階的な実行を行うことができます。たとえば、次のセクションが承認される必要がある場合や、最初のセクションが正常に実行されたことが確認された場合などです。承認ポイントでは、ワークフローの一時停止と再開の間のプロセスは処理されません。E メール通知と SNMP 通知は、WFA 設定で指定されているように送信されます。ワークフローの一時停止通知を受信すると、ストレージオペレータに特定の操作を実行するよう求められます。たとえば、ストレージオペレータは、承認のために計画の詳細を管理者、承認者、オペレータに送信し、承認を受け取った時点でワークフローを再開できます。

承認が必要になることはありません。一部のシナリオでは、承認が必要になるのは、特定の条件が満たされ、承認ポイントが追加されたときに条件を設定できる場合だけです。たとえば、ボリュームのサイズを拡張するワークフローを考えてみましょう。ワークフローの開始時に承認ポイントを追加すると、ボリュームサイズの増加によってボリュームを含むアグリゲートのスペースが 85% 使用された場合に、ストレージオペレータが承認を得ることができます。ワークフローの実行中およびこの条件になるボリュームの選択中に、承認されるまで実行は停止されます。

承認ポイントに設定された条件には、次のいずれかのオプションがあります。

- 条件なし
- 指定した変数が見つかった場合
- 指定した変数が見つからない場合
- 指定した式が `true` と評価されます

ワークフロー内の承認ポイントの数に制限はありません。ワークフローのコマンドの前に承認ポイントを挿入し、承認ポイントの後にコマンドを設定して、実行前に承認を待つことができます。承認ポイントは、変更時刻、ユーザー、コメントなどの情報を提供します。これにより、ワークフローの実行が一時停止または再開された日時と理由を確認できます。承認ポイントのコメントには、MVEL 式を含めることができます。

定義済みワークフローの承認ポイントの例

Designer で次の定義済みワークフローを開いて、承認ポイントの使用方法を理解できます。

- clustered Data ONTAP ボリュームを削除
- HA ペアのコントローラとシェルフのアップグレード
- ボリュームをマイグレートする

カスタム REST エンドポイントの実行方法

OnCommand Workflow Automation (WFA) には、ワークフローを実行するためのカスタム REST エンドポイントを設定するためのメカニズムが用意されています。カスタム REST のエンドポイントは、アーキテクトがわかりやすく、わかりやすい Uniform Resource Identifier (URI) を設定してワークフローを実行するのに役立ちます。URI は、ワークフローのセマンティクスに基づいて、POST、PUT、DELETE の REST の規則に準拠します。これらの URI を使用すると、クライアント開発者はクライアントコードを簡単に開発できます。

WFA では、API呼び出しを介してワークフローを実行するためのカスタム URI パスを設定できます。URI パス内の各セグメントは、文字列、または括弧内のワークフローのユーザー入力の有効な名前にすることができます。たとえば、「/Dev/{ProjectName}/clone.」のようになります ワークフローは'

[https://WFAServer:HTTPS_port/rest/Devation/Project1/clone/jobs.'](https://WFAServer:HTTPS_port/rest/Devation/Project1/clone/jobs.) の呼び出しとして呼び出すことができます

URI パスの検証は次のとおりです。

- REST パスは「/」で始まる必要があります。
- 使用できる文字は、アルファベット、数字、アンダースコアです。
- ユーザー入力名は“{}”で囲む必要があります



"{}" で囲まれた値が有効なユーザー入力名であることを確認する必要があります。

- 空のパスセグメントがないようにしてください。たとえば、/、/ {} / のように指定します。
- HTTP メソッドの設定とカスタム URI パスの設定の両方を行うか、どちらも設定しないでください。

障害発生時の続行方法

障害発生時に続行機能を使用すると、ワークフローのステップを設定して、そのステップが失敗した場合でもワークフローの実行を継続できるようにすることができます。失敗した手順に対処し、「wfa_log」ファイルにアクセスするか、またはをクリックして失敗の原因となった問題を解決できます をクリックします。

このような失敗したステップが 1 つ以上あるワークフローは、実行完了後に部分的に成功した状態になります。[パラメータ for <command_name>] ダイアログボックスの [詳細設定] タブで必要なオプションを選択して、ステップが失敗した場合でもワークフローの実行を続行するようにステップを設定できます。

失敗したときに続行するようにステップが設定されていない場合、そのステップが失敗するとワークフローの実行は中止されます。

失敗時に続行するように設定されているステップが失敗した場合は、次のいずれかのオプションを使用してワークフローを実行するように設定できます。

- ワークフローの実行を中止する (デフォルトオプション)
- 次の手順から実行を続行します
- 次の行から実行を続行します

ワークフロー要件チェックリストの例

ワークフロー要件チェックリストには、計画したワークフローのコマンド、ユーザ入力、リソースなどの詳細な要件が含まれます。このチェックリストを使用してワークフローを計画し、要件とのギャップを特定することができます。

要件チェックリストの例

次のワークフロー要件チェックリストの例は、クラスタ構成 Data ONTAP ボリュームの作成ワークフローの要件を示しています。このサンプルチェックリストをテンプレートとして使用して、ワークフローの要件をリストできます。

要件	説明
ワークフロー名	clustered Data ONTAP ボリュームを作成
カテゴリ	ストレージのプロビジョニング
説明	ワークフローは、特定の SVM に新しいボリュームを作成します。このワークフローは、あとで使用するためにボリュームをプロビジョニングして委譲するシナリオを想定しています。
ワークフローの仕組みを示す高レベルの概要	<ul style="list-style-type: none">ボリュームを含む SVM はユーザによって指定されます（クラスタ、SVM 名）。指定したサイズに基づいてボリュームが作成されます。ボリュームの構成については、テンプレートで説明します。

要件	説明
詳細	<ul style="list-style-type: none"> • Create CM Volume * コマンドを使用します • Create CM Volume * のコマンドの詳細： <ul style="list-style-type: none"> ◦ 実行は常に設定されます ◦ ボリュームの詳細は、ボリュームの属性を入力することで指定します ◦ ボリュームの構成には、 * Space ながらパフォーマンスの保証設定 * テンプレートを使用します ◦ ボリュームの名前とサイズはユーザが指定します。 <p>ボリュームは SVM ネームスペース内で「 /volname 」（ルートネームスペースの下）としてマウントされます。</p> <ul style="list-style-type: none"> ◦ スナップリザーブは 5% になるため、 * actualVolumeSize * 機能を使用します。 ◦ SVM 参照は、次のリソース選択ロジックで定義されます。 <ul style="list-style-type: none"> ▪ CM SVM - ユーザが指定した名前とクラスタで SVM を検索します ▪ タイプ別の CM SVM - データ SVM のみ (タイプ = クラスタ) ▪ SVM - 状態別 - (state = running) ◦ 集約基準は、リソース選択ロジックを定義済みのファインダ (スペースしきい値と RAID タイプ別の CM 集約) として定義されます。 <ul style="list-style-type: none"> ▪ CM で使用可能な容量別のアグリゲート (容量 = プロビジョニングするボリュームのサイズ、ユーザがクラスタを指定) ▪ CM で、 SVM に委譲されたアグリゲート ▪ RAID タイプ別の CM アグリゲート (RAID-DP) ▪ CM アグリゲート - aggr0 ではありません ▪ CM アグリゲート - 使用サイズ (しきい値 = 90 、 spaceToBeProvisioned = size provided 、これはギャランティが volume のためです) ▪ CM アグリゲート - オーバーコミット (しきい値 = 300 、 spaceToBeAllocated = プロビジョニングするボリュームのサイズ) ▪ 空きスペースが最大になるアグリゲートを選択します

名前	を入力します	概要 (データ値、検証など)
クラスタ	ロックされたクエリ (表形式)	<ul style="list-style-type: none"> SVM をホストするクラスタ 表形式のクエリでは、クラスタのプライマリアドレスと名前を表示できます 名前のアルファベット順で並べ替えます
SVM	ロックされたクエリ	<ul style="list-style-type: none"> ボリュームがプロビジョニングされている SVM クエリでは、前の入力で選択したクラスタに属する SVM 名のみが表示されます <p>管理者またはノードではなく、クラスタタイプの SVM のみを表示 (cm_storagegear.vserver のタイプ列)</p> <ul style="list-style-type: none"> アルファベット順に並べ替えます
ボリューム	文字列	<ul style="list-style-type: none"> 作成するボリュームの名前を指定します
サイズ (GB)	整数	<ul style="list-style-type: none"> プロビジョニングするボリュームのサイズ データサイズ (スナップリザーブを考慮する必要があります)

・コマンド *

名前	説明	ステータス
CM ボリュームを作成します	SVM にボリュームを作成します	既存

・戻りパラメータ *

名前	値
ボリューム名	プロビジョニングされたボリュームの名前
アグリゲート名	選択したアグリゲートの名前
ノード名	ノードの名前

名前	価値
クラスタ名	クラスタの名前

- ギャップおよび問題 *

1.	
2.	
3.	
4.	
5.	

ワークフローを作成します

Workflow Automation (WFA) を使用して、データベースやファイルシステムのストレージのプロビジョニング、移行、運用停止などのタスクのワークフローを作成できます。定義済みの WFA ワークフローが要件に合わない場合は、ワークフローを作成する必要があります。

必要なもの

- WFA のビルディングブロックの概念を理解しておく必要があります。
- ワークフローに必要な繰り返し行、承認ポイント、リソース選択などの機能を理解しておく必要があります。
- ワークフロー要件チェックリストなど、ワークフローに必要な計画を完了しておく必要があります。
- ワークフローに関する情報をストレージオペレータに提供するヘルプコンテンツを作成しておく必要があります。

このタスクについて

各ワークフローの構築は、ワークフローの目標と要件によって異なる場合があります。このタスクでは、特定のワークフローの手順については説明しませんが、ワークフローの作成に関する一般的な手順について説明します。

手順

- [* ワークフローデザイン > ワークフロー *] をクリックします。
- をクリックします  をクリックします。
- [* ワークフロー *] タブで、次の手順を実行します。
 - 必要なスキーマを展開し、目的のスキーマをダブルクリックします  (コマンド) または  (ワークフロー) を [使用可能なステップ (Available steps)] リストから選択します。

この手順は必要に応じて繰り返すことができます。ワークフローエディタでステップをドラッグアンド

ドドロップすると、ステップを並べ替えることができます。

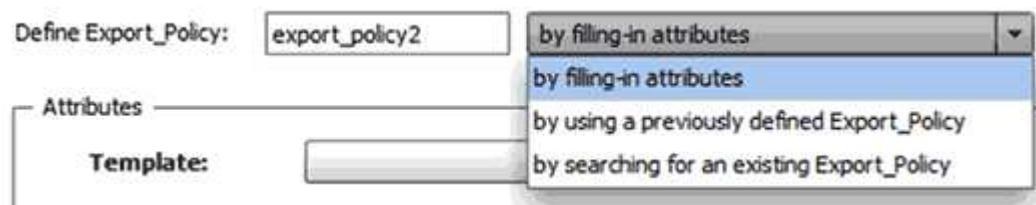
- b. * オプション： * クリック  必要な行数を追加します。これは、ステップの実行の詳細を指定するために使用されます。

各ステップは、指定された行と列で指定されたステップの詳細に基づいて実行されます。ステップは左から右、上から下の順に実行されます。

- c. 追加したステップの下にカーソルを置き、をクリックします  ステップ実行のステップの詳細を必要な行に追加します。

手順	手順
ワークフロー	[* ワークフロー * (* Workflow *)] タブで必要なユーザー入力を入力し、[* 詳細設定 * (* Advanced *)] タブで必要な条件を入力します。
コマンドを実行します	[< コマンド > のパラメータ] タブで、各オブジェクトタブをクリックし、必要なオプションを選択してオブジェクト属性を定義し、[詳細設定] タブと [その他のパラメータ] タブに必要な詳細を入力します。
検索または定義	検索または定義するディクショナリエンティオブジェクトを選択します。

次の図に、オブジェクト属性を定義するための使用可能なオプションを示します。



適切なアクションを選択します。

用途	手順
属性を入力します	<p>次のオプションを使用して、属性の値を入力します。</p> <ul style="list-style-type: none"> 式 変数（variables） ユーザ入力 リソースの選択 名前の増分 <p>属性フィールドにカーソルを合わせ、をクリックする必要があります [...] リソース選択機能または増分命名機能を使用するには、次の手順を実行します。</p>
以前に定義した「 <i>object</i> 」を使用します	オプションリストの前のボックスで'以前に定義したオブジェクトを選択します
既存の「 <i>object</i> 」を検索します	<p>i. [* 検索条件を入力してください*] をクリックし、リソース選択機能を使用してオブジェクトを検索します。</p> <p>ii. 必要なオブジェクトが見つからない場合は、実行に必要なオプションのいずれかを選択します。</p> <ul style="list-style-type: none"> ワークフローを中止する <p>特定のオブジェクトが見つからない場合は、ワークフローの実行を中止します。</p> <ul style="list-style-type: none"> このコマンドを無効にします <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>This option disables only the current step and executes the workflow. **** 「<i>object</i>」の属性を入力し、コマンドを実行します</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. 承認ポイントを挿入する場合は、をクリックします  をクリックし、承認ポイントに必要なコメントを

入力します。

承認ポイントのコメントには、MVEL式を含めることができます。

5. をクリックします ▾ 行番号の横に表示され、次の処理を実行できます。

- 行を挿入します。
- 行をコピーします。
- 行を繰り返します。

次のいずれかのオプションを使用して、コマンドパラメータの繰り返しを指定できます。

▪ 回数

このオプションを使用すると、指定した繰り返し回数に対してコマンドを繰り返し実行できます。たとえば、「Create Qtree」コマンドを3回繰り返して、3つのqtreeを作成するように指定できます。

このオプションは、コマンドの実行数を動的に指定する場合にも使用できます。たとえば、作成するLUN数に対するユーザ入力変数を作成し、ワークフローの実行時またはスケジュール時にストレージオペレータが指定した数を使用できます。

- グループ内のすべてのリソース

このオプションを使用して、オブジェクトの検索条件を指定できます。コマンドは、検索条件からオブジェクトが返される回数だけ繰り返し実行されます。たとえば「クラスタ内のノードを検索し」各ノードに対してCreate iSCSI Logical Interfaceコマンドを繰り返します

- 行を実行するための条件を追加します。
- 行を削除します。

6. [Details] タブで、次の手順を実行します。

- a. [ワークフローネーム *] フィールドと [ワークフロー概要 *] フィールドに必要な情報を指定します。

ワークフローネームと概要は、ワークフローごとに一意である必要があります。

- b. * オプション： * エンティティバージョンを指定します。
- c. * オプション：予約機能を使用しない場合は、* 予約済みエレメントを考慮 * チェックボックスをオフにします。
- d. * オプション： * 同じ名前のエレメントの検証を有効にしない場合は、* エレメントの存在検証を有効にする * チェックボックスをオフにします。

7. ユーザ入力を編集する場合は、次の手順を実行します。

- a. [ユーザー入力 * (User Inputs *)] タブをクリックします。
- b. 編集するユーザー入力をダブルクリックします。
- c. [変数の編集： <ユーザー入力 *>] ダイアログボックスで、ユーザー入力を編集します。

8. 定数を追加する場合は、次の手順を実行します

- a. [定数] タブをクリックし、[*Add] ボタンを使用してワークフローに必要な定数を追加します。

複数のコマンドのパラメーターを定義するために共通の値を使用している場合は、定数を定義できます。たとえば 'Create 'LUN with SnapVault ワークフローで使用される aggregate_OLIDE_THRESHOLD 定数を参照してください

- b. 各定数の名前、概要、および値を入力します。
9. [* 戻りパラメータ *] タブをクリックし、[* 追加 *] ボタンを使用してワークフローに必要なパラメータを追加します。

ワークフローの計画と実行で、計画中に計算値または選択した値を返す必要がある場合は、戻りパラメータを使用できます。ワークフローのプレビューまたはワークフローの実行が完了した後に、モニタリングウィンドウの [戻りパラメータ] タブで計算値または選択した値を表示できます。

Aggregate : 戻りパラメータとしてアグリゲートを指定すると、リソース選択ロジックで選択されたアグリゲートを確認できます。

ワークフローに子ワークフローが含まれていて、子ワークフローの戻りパラメータ名にスペース、ドル記号 (\$) が含まれている場合、または、親ワークフローで子ワークフローの戻りパラメータ値を表示するには、親ワークフローの角かっこ内に戻りパラメータ名を指定する必要があります。

パラメータ名	指定する形式
'ChildWorkflow1.abc\$values	「 ChildWorkflow1 ["abc\$"+" 値 "] 」
'ChildWorkflow1.\$values	「 ChildWorkflow1 ["\$"+" 値 "] 」
「 ChildWorkflow1.value \$ 」	「 ChildWorkflow1.value \$ 」
'ChildWorkflow1.P N	「 ChildWorkflow1 ["P N"] 」
'ChildWorkflow1.return_string("HW")'	'ChildWorkflow1["return_string(\"HW\")"]'

10. * オプション : * ヘルプコンテンツ * タブをクリックして、ワークフロー用に作成したヘルプコンテンツファイルを追加します。
11. [* プレビュー *] をクリックして、ワークフローの計画が正常に完了していることを確認します。
12. [OK] をクリックしてプレビューウィンドウを閉じます。
13. [保存 (Save)] をクリックします。

完了後

テスト環境でワークフローをテストしてから、ワークフローを「*_ワークフローネーム_*>*_詳細_*」で本番環境向けの準備完了としてマークします。

ワークフロー ヘルプコンテンツを作成します

ワークフローを設計する OnCommand Workflow Automation (WFA) の管理者およびアーキテクトは、ワークフローのヘルプコンテンツを作成してワークフローに含めることができます。

必要なもの

HTML を使用して Web ページを作成する方法を理解しておく必要があります。

このタスクについて

このヘルプでは、ワークフローのワークフローに関する情報と、ワークフローを実行するストレージオペレータへのワークフローのユーザ入力について説明します。

手順

1. 次の名前のフォルダを作成します。 workflow-help
2. HTML エディタまたはテキストエディタを使用してヘルプコンテンツを作成し、「 workflow-help 」 フォルダに「 index.htm 」 ファイルとして保存します。

JavaScript ファイルをヘルプコンテンツの一部として含めることはできません。サポートされるファイル拡張子は次のとおりです。

- .jpg
- .jpeg
- .gif
- .png
- .xml に保存されます
- .THMX
- .htm
- .html からのアクセスが可能です
- .css

Windows で作成された 'Thumbs.db' ファイルも含めることができます

3. 「 index.htm 」 ファイルと、イメージなどのヘルプコンテンツに関連するその他のファイルが「 workflow-help 」 フォルダにあることを確認します。
4. フォルダの「 .zip 」 ファイルを作成し、「 .zip 」 ファイルのサイズが 2 MB 以下であることを確認します。

「 NFS ボリュームの作成 - help.zip 」 を参照してください

5. ヘルプコンテンツを作成したワークフローを編集し、 **Setup>*Help Content*>*Browse*** をクリックして、「 .zip 」 ファイルをアップロードします。

WFA ワークフローパックを作成します

ストレージの自動化と統合の要件に対応するワークフローパックを OnCommand Workflow Automation (WFA) で作成できます。

手順

1. Web ブラウザを使用して WFA ウィンドウにログインします。
2. [* コンテンツ管理 * (Content Management *)] > [* パック (* Packs)]
3. [新しいパック * (New Pack *)] アイコンをクリックします。

4. [新しいパック * (* New Pack *)] ダイアログボックスで、[*名前 * (* Name *)]、[*作成者 * (* Author *)]、[*バージョン * (* Version *)]、および[*概要 * (**)] フィールド
5. [保存 (Save)] をクリックします。
6. 新しいパックが [*Packs] ウィンドウに作成されていることを確認します。

WFA ワークフローパックにエンティティを追加します

ストレージの自動化と統合の要件に応じて、OnCommand Workflow Automation (WFA) のワークフローパックに 1 つ以上のエンティティを追加できます。

このタスクについて

次のエンティティからパックを削除できます。

- ・ワークフロー
- ・ファインダ
- ・フィルタ
- ・コマンド
- ・機能
- ・テンプレート
- ・スキーム
- ・辞書
- ・データソースの種類
- ・リモートシステムタイプ
- ・キヤッショクエリ
- ・カテゴリ

手順

1. Web ブラウザから * WFA * ウィンドウにログインします。
2. * ワークフローデザインポータルで、追加するエンティティに移動し、* < エンティティ >* をクリックします。
3. エンティティー * ウィンドウで 'パックに追加するエンティティを選択します
4. 「* パックに追加 *」アイコンをクリックします。

「Add to Pack」は、証明書が *None.* に設定されているエンティティに対してのみ有効になります

5. パックに追加 < エンティティ >* (* Add to Pack < Entity >*) ダイアログボックスの * 使用可能なパック * (* Available Packs *) ドロップダウンリストから、エンティティを追加するパックを選択します。
6. [OK] をクリックします。

OnCommand Workflow Automation パックを削除します

不要になったパックは OnCommand Workflow Automation (WFA) から削除できま

す。パックを削除すると、パックに関連付けられているすべてのエンティティが削除されます。

このタスクについて

- パックの一部であるエンティティに依存関係がある場合は、パックを削除できません。

たとえば、カスタムワークフローの一部であるコマンドを含むパックを削除しようとすると、カスタムワークフローはパックに依存するため、削除処理が失敗します。パックを削除できるのは、カスタムワークフローを削除した後だけです。

- パックの一部であるエンティティを個別に削除することはできません。

パックの一部であるエンティティを削除するには、そのエンティティを含むパックを削除する必要があります。エンティティが複数のパックに含まれている場合、WFA サーバからそのエンティティを含むすべてのパックが削除されるまでエンティティは削除されません。

手順

- Web ブラウザから admin として WFA にログインします。
- [* コンテンツ管理 * (Content Management *)] > [* パック (* Packs)]
- 削除するパックを選択し、をクリックします 。
- [* パックの削除 * (* Delete Pack *)] 確認ダイアログボックスで、[OK] をクリックします。

OnCommand Workflow Automation コンテンツをエクスポートします

ユーザが作成した OnCommand Workflow Automation (WFA) のコンテンツを「.dar」ファイルとして保存し、他のユーザと内容を共有できます。WFA のコンテンツには、ユーザが作成したコンテンツ全体、またはワークフロー、ファインダ、コマンド、ディクショナリなどの特定の項目を含めることができます。

必要なもの

- エクスポートする WFA コンテンツへのアクセス権が必要です。
- エクスポートするコンテンツに認定コンテンツへの参照が含まれている場合、コンテンツのインポート時に、対応する認定コンテンツパックをシステムで使用できるようにする必要があります。

これらのパックは Storage Automation Store からダウンロードできます。

このタスクについて

- 次の種類の認定コンテンツはエクスポートできません。
 -  - ネットアップ認定コンテンツ
 -  - Professional Services (PS ; プロフェッショナルサービス) が開発したコンテンツ。 PS によるカスタムインストールでのみ利用可能です
 -  - ユーザが開発したパック
- エクスポートされたオブジェクトに依存するすべてのオブジェクトもエクスポートされます。

たとえば、ワークフローをエクスポートすると、ワークフローの依存コマンド、フィルタ、ファインダも

エクスポートされます。

- ロックされたオブジェクトをエクスポートできます。

オブジェクトは、他のユーザーによってインポートされるとロック状態のままになります。

手順

1. Web ブラウザを使用して WFA にログインします。
2. 必要なコンテンツをエクスポートします。

状況	手順
ユーザーが作成したすべてのコンテンツを 1 つの .dar ファイルとしてエクスポートします	<ol style="list-style-type: none">[* 設定 *] をクリックし、[* メンテナンス *] の下にある [すべてのワークフローをエクスポート *] をクリックします。「.dar」ファイルのファイル名を指定し、「* Export *」をクリックします。
特定のコンテンツをエクスポートします	<ol style="list-style-type: none">コンテンツをエクスポートするウィンドウに移動します。ウィンドウで 1 つ以上の項目を選択し、をクリックします .[名前を付けてエクスポート] ダイアログボックスで '.dar' ファイルのファイル名を指定し [Export] をクリックします

3. [名前を付けて保存 *] ダイアログボックスで '.dar' ファイルを保存する場所を指定し [保存 *] をクリックします

OnCommand Workflow Automation コンテンツをインポートします

ワークフロー、ファインダ、コマンドなど、ユーザが作成した OnCommand Workflow Automation (WFA) のコンテンツをインポートできます。また、別の WFA インストールからエクスポートしたコンテンツ、Storage Automation Store または WFA コミュニティからダウンロードしたコンテンツ、および Data ONTAP PowerShell ツールキットや Perl NMSDK ツールキットなどのパックをインポートすることもできます。

必要なもの

- インポートする WFA コンテンツへのアクセス権が必要です。
- インポートするコンテンツが、同じバージョンかそれ以前のバージョンの WFA を実行しているシステムに作成されている必要があります。
たとえば、WFA 2.2 を実行している場合、WFA 3.0 を使用して作成されたコンテンツをインポートすることはできません。
- N-2 バージョンの WFA で開発されたコンテンツは、WFA 5.1 にのみインポートできます。

- 「.dar」ファイルが NetApp 認定コンテンツを参照している場合は、NetApp 認定コンテンツ・パックをインポートする必要があります。

ネットアップ認定コンテンツパックは、Storage Automation Store からダウンロードできます。パックのドキュメントを参照して、すべての要件が満たされていることを確認する必要があります。

手順

1. Web ブラウザを使用して WFA にログインします。
2. [* 設定 *] をクリックし、[* メンテナンス *] の [* ワークフローのインポート *] をクリックします。
3. [ファイルの選択 *] をクリックして 'インポートする .dar ファイルを選択し [インポート *] をクリックします
4. [インポート成功 * (Import Success *)] ダイアログボックスで、[OK] をクリックします。

WFA ワークフローパックをインポート

ストレージの自動化と統合の要件に応じて、サーバから OnCommand Workflow Automation (WFA) にワークフローパックをインポートできます。

必要なもの

インポートするサーバ内の WFA コンテンツにアクセスできる必要があります。

手順

1. Web ブラウザから * WFA * ウィンドウにログインします。
2. [* コンテンツ管理 * (Content Management *)] > [* パック (* Packs)]
3. [サーバーからインポート (Import from Server)] アイコンをクリックします。
4. [サーバーフォルダからインポート *] ダイアログボックスの [サーバーシステムのフォルダの場所 *] フィールドに、サーバー内のパックの場所を文字列形式で入力します。たとえば、「C:\work\packs\test.」と入力します
5. [OK] をクリックします。
6. パックが * パック * ウィンドウにインポートされていることを確認します。

OnCommand Workflow Automation コンテンツをインポートする際の考慮事項

ユーザが作成したコンテンツ、別の OnCommand Workflow Automation (WFA) インストールからエクスポートされたコンテンツ、または Storage Automation Store または WFA コミュニティからダウンロードしたコンテンツをインポートする場合は、一定の考慮事項に注意する必要があります。

- WFA のコンテンツは「.dar」ファイルとして保存されます。また、ユーザが作成したコンテンツ全体を別のシステムや、ワークフロー、ファインダ、コマンド、ディクショナリなどの特定の項目に含めることができます。
- 既存のカテゴリが '.dar' ファイルからインポートされると 'インポートされたコンテンツ' がカテゴリ内の既存のコンテンツとマージされます

たとえば、WFA サーバのカテゴリ A には 2 つのワークフロー WF1 および WF2 があるとします。カテゴ

リ A のワークフロー WF3 および Wf4 を WFA サーバにインポートすると、カテゴリ A にはインポート後にワークフロー WF1、WF2、WF3、および Wf4 が含まれます。

- 「.dar」ファイルにディクショナリエントリが含まれている場合、ディクショナリエントリに対応するキヤッショテーブルが自動的に更新されます。

キヤッショテーブルが自動的に更新されない場合は、「wfa_log」ファイルにエラーメッセージが記録されます。

- WFA サーバに存在しないパックに依存する「.dar」ファイルをインポートすると、WFA は、エンティティに関連するすべての依存関係が満たされているかどうかを確認しようとします。

- 1つ以上のエンティティが見つからない場合や、エンティティの下位バージョンが見つかった場合、インポートは失敗し、エラーメッセージが表示されます。

エラーメッセージには、依存関係を満たすためにインストールする必要があるパックの詳細が表示されます。

- 上位バージョンのエンティティが見つかった場合や、証明書が変更された場合は、バージョン不一致に関する一般的なダイアログボックスが表示され、インポートが完了します。

バージョン不一致の詳細は 'wfa_log' ファイルに記録されます

- 次の項目についての質問やサポートリクエストは、WFA コミュニティに送信される必要があります。
 - WFA コミュニティからダウンロードされたすべてのコンテンツ
 - 作成したカスタムの WFA コンテンツ
 - 変更した WFA のコンテンツ

アップグレード中のパック ID

アップグレードプロセスの実行中、OnCommand Workflow Automation (WFA) はエンティティを識別してパックに分類します。アップグレード前にパックのエンティティを削除した場合、アップグレード中にパックは識別されません。

WFA はアップグレードプロセス中に、データベースのパックと Storage Automation Store でリリースされたパックのリストを比較し、アップグレード前にインストールされたパックを特定します。したがって、パック ID はデータベース内の既存のパックを分類します。

WFA は次のプロセスを実行して、パックを特定し、分類します。

- Storage Automation Store でリリースされたパックのリストを管理し、アップグレード前にインストールされたパックを比較して確認します。
- Storage Automation Store が有効になっている場合に、パック内のエンティティを Storage Automation Store の同期の一部として分類します。
- 更新されたリストを使用してエンティティをパックに分類します。

パック ID は、Storage Automation Store からダウンロードしたネットアップ認定パックにのみ適用されます。

アップグレード中にパックが特定されなかった場合は、パックを再インポートして WFA で特定できるように

することができます。wfa.log ファイルには、アップグレード時にパックとして識別されなかったエンティティに関する詳細が含まれています。

WFA ワークフローパックを SCM リポジトリと統合する

OnCommand Workflow Automation (WFA) パックは、ソース管理管理 (SCM) リポジトリと統合できます。

管理者またはアーキテクトのクレデンシャルが必要です。

GitHub、Perforce、SVNなどのSCMツールでは、SCMリポジトリサーバからコードをチェックアウトするためにローカルディレクトリをマッピングする必要があります。このローカルディレクトリマッピングは、_SCM クライアントの場所と呼ばれます。_SCM クライアントを設定して、ファイルシステムの場所をクライアント領域として指定する必要があります。

WFA サーバシステムに SCM クライアントを設定できます。SCM を使用するには、WFA サーバシステムにアクセスできる必要があります

SCM に新しいワークフローパックをチェックインします

OnCommand Workflow Automation (WFA) を使用して新しいワークフローパックを作成し、ソース管理 (SCM) にチェックインできます。

必要なもの

SCM をセットアップする必要があり、管理者またはアーキテクトのクレデンシャルが必要です。

手順

1. Web ブラウザから * WFA * ウィンドウにログインします。
2. 新しいワークフローパックを作成します。

Workflow Automation パックを作成します

3. 作成したパックにエンティティを追加します。

OnCommand Workflow Automation パックにエンティティを追加します

4. [サーバーにエクスポート (Export to Server)] アイコンをクリックします。
5. [Export to Server Folder] ダイアログ・ボックスの [Folder location at server system] フィールドに 'SCM クライアントを含むサーバにパックを保存するファイル・システムの場所を入力します

パックまたはコンテンツを編集または再エクスポートするには、* ロック解除 * アイコンをクリックします。

6. SCM クライアントの場所で、パックの内容を SCM サーバにチェックインします。

新しいバージョンの WFA ワークフローパックをチェックインします

OnCommand Workflow Automation (WFA) でパックのバージョンを更新し、ソース管理管理 (SCM) サーバの新しい場所に更新したパックをチェックインできます。

必要なもの

SCM をセットアップする必要があり、管理者またはアーキテクトのクレデンシャルが必要です。

手順

1. Web ブラウザから * WFA * ウィンドウにログインします。
2. [* コンテンツ管理 * (Content Management *)] > [* パック (* Packs)]
3. 「* パックの編集 *」アイコンをクリックします。
4. * パック < パック名 > * ダイアログボックスの * バージョン * フィールドで、パックのバージョンを更新します。
5. [保存 (Save)] をクリックします。
6. パックレベルで * サーバーにエクスポート * アイコンをクリックします。
7. サーバーフォルダへのエクスポート * (* Export to Server Folder *) ダイアログボックスのサーバーシステムでのフォルダの場所 * (* Folder location at server system *) フィールドに、新しいファイルシステムの場所を入力します。
パックが以前に「C:\p4.cd\1.0.0」ファイルシステムの場所に保存されていた場合は、「C:\p4.cd\2.0.0」の場所に保存します。
8. SCM クライアントの場所で、パックの内容を SCM サーバの新しい場所にチェックインします。
パックが SCM サーバ内の「/depot/wfa/packs/cdot /1.0.0」パスに保存されていた場合は、「/depot/wfa/packs/cdot /2.0.0」などの別の場所に保存できます

SCM サーバから WFA ワークフローパックを更新します

ソース管理 (SCM) サーバのパックを更新し、更新されたパックを OnCommand Workflow Automation (WFA) にインポートできます。

必要なもの

SCM をセットアップする必要があり、管理者またはアーキテクトのクレデンシャルが必要です

このタスクについて

SCM サーバのパックに変更や更新を加える場合、管理者やアーキテクトは SCM で提供される diff ツールを使用して競合を解決する必要があります。WFA は XML の差分ファイルを調整し、関連する変更のみを表示します。

パックをインポートする前に、WFA パックのコンテンツに加えられた変更について通知されます。

手順

1. Web ブラウザを使用して WFA ウィンドウにログインします。
2. 更新したパックを WFA にインポートします。

"WFA ワークフローパックをインポート"



WFA データベースに同じパックがすでに含まれている場合は、パックのコンテンツが上書きされます。

既存の **WFA** ワークフローパックを **SCM** サーバにチェックインします

既存のパックは OnCommand Workflow Automation (WFA) からソース管理管理 (SCM) サーバにチェックインできます。

必要なもの

SCM をセットアップする必要があり、管理者またはアーキテクトのクレデンシャルが必要です。

手順

1. Web ブラウザから * WFA * ウィンドウにログインします。
2. [* コンテンツ管理 * (Content Management *)] > [* パック (* Packs)]
3. [サーバーにエクスポート (Export to Server)] アイコンをクリックします。
4. サーバーフォルダへのエクスポート * (* Export to Server Folder *) ダイアログボックスのサーバーシステムでのフォルダの場所 * (* Folder location at server system *) フィールドに、サーバーにパックが保存されているサーバーフォルダの場所を入力します。

これにより、 SCM クライアントが作成されるファイルシステム内の展開形式でパックがエクスポートされます。

5. SCM クライアントの場所で、パックの内容を SCM サーバにチェックインします。
6. SCM で提供される diff ツールを使用して、 SCM 版と比較して変更内容を確認します。

WFA ワークフローパックをエンティティから削除します

OnCommand Workflow Automation (WFA) のエンティティからパックを削除し、更新されたパックをソース管理 (SCM) サーバにチェックインできます。

必要なもの

SCM をセットアップする必要があり、管理者またはアーキテクトのクレデンシャルが必要です。

このタスクについて

次のエンティティからパックを削除できます。

- ・ワークフロー
- ・ファインダ
- ・フィルタ
- ・コマンド
- ・機能
- ・テンプレート
- ・スキーム
- ・辞書
- ・データソースの種類
- ・リモートシステムタイプ

- ・キャッシュクエリ
- ・カテゴリ

手順

1. Web ブラウザから * WFA * ウィンドウにログインします。
2. * ワークフローデザインポータルで、削除するエンティティに移動し、* < エンティティ > * をクリックします。
3. [パックから削除 (Remove from Pack)] アイコンをクリックします。
4. パックから削除 (* Remove from Pack) < エンティティ > * (Entity > *) ダイアログボックスで、そのエンティティから削除するパックを選択します。
5. [OK] をクリックします。
6. [*Packs] タブをクリックします。
7. [サーバーにエクスポート (Export to Server)] アイコンをクリックします。
8. サーバーフォルダへのエクスポート * (* Export to Server Folder *) ダイアログボックスのサーバーシステムでのフォルダの場所 * (* Folder location at server system *) フィールドに、サーバーにパックが保存されているサーバーフォルダの場所を入力します。
- これにより、SCM クライアントが作成されるファイルシステム内に展開形式でパックがエクスポートされます。
9. SCM クライアントの場所で、パックの内容を SCM サーバにチェックインします。
10. SCM で提供される diff ツールを使用して、SCM 版と比較して変更内容を確認します。

WFA ワークフローパックを SCM 内の以前のバージョンにロールバックします

パックは、Source Control Management (SCM ; ソース管理管理) で以前のバージョンにロールバックして、OnCommand Workflow Automation (WFA) にインポートできます。

必要なもの

SCM をセットアップする必要があり、管理者またはアーキテクトのクレデンシャルが必要です。

手順

1. SCM クライアントの場所で、SCM ツールを使用して、ファイルシステムの場所にある以前のバージョンにパックをロールバックします。

SCM クライアントは、必要な変更番号と完全に同期されます。

2. Web ブラウザを使用して WFA ウィンドウにログインします。
3. 更新したパックを WFA にインポートします。

"WFA ワークフローパックをインポート"



これにより、WFA データベースが以前のバージョンにロールバックされます。

ワークフローのビルディングブロックの作成

Workflow Automation (WFA) には、ワークフローの構築に使用するビルディングブロックが複数含まれています。ワークフローに必要な WFA Buildings ブロックを作成できます。

データソースタイプを作成します

データソースからのデータ収集を有効にするには、データソースの種類を作成する必要があります。これは、OnCommand Workflow Automation (WFA) では事前定義されていません。

必要なもの

- WFA で事前定義されていないカスタムデータソースタイプを作成する場合は、必要なディクショナリエンティリとディクショナリ方式を作成しておく必要があります。
- スクリプトメソッドを使用するデータソースタイプを作成するために、PowerShell のスクリプト作成を理解しておく必要があります。

手順

- [* データソースデザイン > データソースタイプ*] をクリックします。
- をクリックします  をクリックします。
- [新しいデータソースタイプ*] ダイアログボックスで、[* データソース*]、[* データソースバージョン*]、および[* スキーム*] フィールドに必要な詳細を入力または選択します。

- [Default port] フィールドに、ポート番号を入力します。

「2638」

入力したポート番号は、データを取得するためにこのデータソースを追加するときに入力されます。WFA は、デフォルトでこのポートを使用してデータソースと通信します。データソースサーバでポートが開いている必要があります。

- 方法* のリストから、WFA がデータを取得するために使用する方法を選択します。

選択した項目	作業
SQL>	<p>[ドライバの種類*] リストから、データソースに適した次のいずれかのドライバを選択します。</p> <ul style="list-style-type: none">* MySQL Connector/J * を使用します* MySQL Server JDBC ドライバ 3.0** Oracle JDBC ドライバ 11.2.0.3*

選択した項目	作業
スクリプト	<p>[* スクリプト *] フィールドに、データソースとの接続およびデータ取得に使用する PowerShell スクリプトを入力します。</p> <p></p> <p>ディクショナリエントリの等価 CSV ファイルのデータには、フィールド区切り文字としてタブが含まれている必要があります。たとえば、VMware vCenter のデータソースタイプに対応した PowerShell スクリプトを参照してください。</p>

6. [保存 (Save)] をクリックします。

コマンドを作成します

タスクに適した定義済みの WFA コマンドがない場合は、WFA コマンドを作成してワークフロー内の特定のタスクを完了できます。

必要なもの

PowerShell または Perl を使用して、コマンドに必要なコードを記述する方法を理解しておく必要があります。

手順

1. [* ワークフローデザイン > コマンド *] をクリックします。
2. をクリックします  をクリックします。
3. [新規コマンド定義 * (New Command Definition)] ダイアログボックスの [* プロパティ * (* Properties *)] タブで、[* 名前 * (* Name *)]、[* 概要 * (* Timeout *)]、および [* タイムアウト * (* Timeout *)] フィールドに必要な詳細を入力
 - a. [*String Representation *] フィールドに、MVEL 構文を使用してコマンドの文字列表現を入力します。

「 + VolumeName + " ⇒ " + " SnapshotName + "」

コマンドの文字列表現を使用して、計画および実行時にワークフロー設計で表示する情報を表示します。コマンドの文字列表現では、コマンドのパラメータのみを使用する必要があります。

4. [コード *] タブで、次の手順を実行します。
 - a. コマンドに必要なスクリプト言語を * スクリプト言語 * リストから選択します。
 - [+] をクリックして、コマンドの追加言語を選択できます。
 - b. [選択した言語] タブにコマンドの適切なコードを入力します。

PowerShell スクリプトでユーザ入力にパスワードタイプを使用する場合は、パラメータのエイリアスを作成し、属性に「_Password」を含める必要があります。Perl スクリプトの場合は、* パラメータ一定義 * (Parameters Definition) タブで、タイプを * パスワード * として指定できます。



コマンドエディタでは、「<」、「&」、「XML タグ」<>」という特殊文字はサポートされていません。

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

5. [パラメータ一定義 (* Parameters Definition)] タブで、次の手順を実行します。

- パラメータ定義テーブルに値を入力するには ' パラメータ検出 (Discover Parameters) * をクリックします

パラメータとその属性はコードから抽出され、表に表示されます。たとえば、Array パラメータと VolumeName パラメータは次のコードから抽出されます。

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

- パラメータの概要 列をクリックして、概要 を編集します。

このタブで他のフィールドを編集することはできません。

6. [* パラメーターマッピング * (* Parameters Mapping *)] タブで、パラメーターごとに次の手順を実行します。

- [* タイプ * (* Type *)] 列から、適切な辞書オブジェクトを選択します。
- [* 属性 * (* Attribute *)] 列で、リストからディクショナリオブジェクトの適切な属性を入力または選択します。

属性を入力したら、ピリオド (.) を入力し、そのオブジェクトの別の属性を含めることができます。

type に「cm_storageVolume」を、AggregateName パラメータの属性に「aggregate.name」を入力します。

- [* オブジェクト名 * (* Object Name *)] 列に、オブジェクト名を入力します。

オブジェクト名は、ワークフローでコマンドの詳細を指定するときに、[<コマンド>のパラメータ] ダイアログボックスのタブの下にパラメータをグループ化するために使用されます。

ワークフローでコマンドの詳細を指定する場合、[コマンドパラメータ] ダイアログボックスの [他のパラメータ*] タブにマップされていないパラメータが表示されます。

7. 予約*タブで、SQL クエリを使用して予約スクリプトを入力し、スケジュールされたワークフローの実行中にコマンドで必要なリソースを予約します。

a. * オプション： * 予約リプレゼンテーション* フィールドに、MVEL 構文を使用して予約の文字列表現を入力します。

SnapMirror ラベルのルールを追加します

「`SnapMirrorLabel[.code]`」を SnapMirror ポリシー「」に割り当てます

"+PolicyName +""`

ストリング表現は、予約ウィンドウで予約されているリソースの詳細を表示するために使用されます。



データベースに対して、cm_storage、cm_performance、storage、performance、vc 以外の操作を実行しないでください。およびカスタムスキーム。

8. * オプション： * Verification* タブで、SQL クエリを入力して、コマンドがデータソースと WFA キャッシュに影響を及ぼしていないかどうかを確認し、予約を削除できるようにします。

入力する SQL クエリは 'SQL SELECT 文だけで構成できます

a. 検証スクリプトをテストするには、[検証のテスト] をクリックします。

b. [* Verification] ダイアログボックスで、必要なテストパラメータを入力します。

c. 予約データを使用して検証スクリプトをテストしない場合は、[テスト*で予約データを使用する] フィールドをクリアします。

d. [* テスト*] をクリックします。

e. テスト結果を確認したら、ダイアログボックスを閉じます。

9. * Test* をクリックしてコマンドをテストします。

10. * テストコマンド <コマンド名>* ダイアログボックスで、* テスト* をクリックします。

テストの結果は、ダイアログボックスのログメッセージセクションに表示されます。

11. [保存 (Save)] をクリックします。

予約スクリプトでコマンドをテストします

OnCommand Workflow Automation (WFA) コマンド用に開発した予約スクリプトをプレイグラウンドデータベースでテストすることで、スクリプトが正常に動作していて WFA データベーステーブルに影響していないことを確認できます。

このタスクについて

WFA のデフォルトのインストールパスは、この手順で使用されます。インストール時にデフォルトの場所を変更した場合は、変更した WFA のインストールパスを使用する必要があります。

手順

1. WFA サーバでコマンドプロンプトを開き、ディレクトリを「c : \Program Files\NetApp\WFA\mysql\mysql\bin」に変更します
2. 次のコマンドを使用して、WFA データベースのダンプを作成します。 `mysqldump -u wfa -pWfa123 --single-transaction --skip-add-drop-table database-tables > dump_location`
cm_storage データベーステーブルのダンプを作成するコマンド： `mysqldump -u wfa -pWfa123 --single-transaction --skip-add-drop-table cm_storage > c : /tmp/cmSt2.sql`
3. 次のコマンドを使用して、作成したダンプを WFA のプレイグラウンドデータベースにリストアします。「mysql -u wfa -pWfa123 遊び場 <dump_location``
「mysql -u wfa -pWfa123 プレイグラウンド」 <c : /tmp/cmSt2.sql`
4. WFA コマンドを作成または編集し、予約スクリプトを「* Reservation *」タブに書き込みます。

予約および検証スクリプトでプレイグラウンドデータベースのみが使用されていることを確認する必要があります。

5. ワークフローを作成または編集し、ワークフローにコマンドを含めてから、ワークフローを実行します。
6. リザベーションスクリプトと検証スクリプトが想定どおりに動作していることを確認します。

WFA のデータソースの取得プロセスでプレイグラウンドデータベースが更新されない。コマンドで作成した予約は手動で削除する必要があります。

Finder を作成します

WFA Finder を作成し、必要なリソースの検索に適した定義済みの WFA Finder がない場合にリソースを検索することができます。

必要なもの

Finder で使用する必要なフィルタを作成しておく必要があります。

手順

1. [* ワークフローデザイン > ファインダ *] をクリックします。
2. をクリックします  をクリックします。
3. 「新規 Finder *」ダイアログボックスの「* プロパティ *」タブで、「* 名前 *」、「* タイプ *」、および「* 概要 *」フィールドに必要な詳細を入力または選択します。
4. [* フィルタ * (* Filters *)] タブで、[使用可能なフィルタ * (* Available Filters *)] リストから必要なフィルタを選択し、をクリックする 。

要件に応じてフィルタを追加または削除できます。

5. [返された属性 (Returned Attributes)] タブで、[使用可能な属性 (*Available *)] リストからフィル

タに必要な属性を選択し、をクリックする 。

6. * オプション： * Test * をクリックして、 Finder をテストします。

- [* テストファインダ <検索名 >*] ダイアログボックスで、必要なテストパラメータを入力します。
- Finder のテストに予約データを使用しない場合は、 [テストで予約データを使用する *] チェックボックスをオフにします。
- [* テスト *] をクリックします。

テストの結果が表示されます。

- ダイアログボックスを閉じます。

7. [保存 (Save)] をクリックします。

フィルタを作成します

タスクに適した定義済みの WFA フィルタがない場合にリソースを検索できる WFA フィルタを作成できます。

必要なもの

フィルタを作成するには、適切な SQL 構文を知っている必要があります。

手順

- [* ワークフローデザイン >] > [フィルタ *] をクリックします。
- をクリックします  をクリックします。
- [新しいフィルタ *] ダイアログボックスの [* プロパティ *] タブで、 [* 名前 *] 、 [* 辞書タイプ *] 、および [* 概要 *] フィールドに必要な詳細を入力または選択します。
- [* クエリ *] タブで、フィルタの適切な SQL クエリを入力します。

1 つの SQL クエリを入力する必要があります、必要に応じて入力パラメータを使用できます。入力パラメータ「+\$ { ParameterName } +'」を使用するには、次の構文を使用する必要があります。

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

5. [* 更新 * (Refresh *)] をクリックして、 [* 入力パラメーター * (* Input Parameters *)] テーブルと [* 戻り属性 * (* Returned Attributes *)]

この情報は、入力した SQL クエリから取得されます。たとえば、前の手順で SQL クエリの例を使用すると、返される属性に IP が表示され、入力パラメータに ArrayName が表示されます。エントリは、 * Label * および * 概要 * 列で編集できます。

6. * オプション : * Test * をクリックして、フィルタをテストします。
 - a. [* テストフィルタ <フィルタ名> *] (* Test Filter <FilterName> *) ダイアログボックスで、必要なテストパラメータを入力します。
 - b. 予約データをフィルタのテストに使用しない場合は、[テストで予約データを使用する *] チェックボックスをオフにします。
 - c. [* テスト *] をクリックします。

テスト結果が表示されます。

 - d. ダイアログボックスを閉じます。
7. [保存 (Save)] をクリックします。

ディクショナリエントリを作成します

ストレージ環境で新しいオブジェクトタイプとその関係を定義する場合は、WFA ディクショナリエントリを作成します。

手順

1. [* データソースデザイン > 辞書 *] をクリックします。
2. をクリックします  をクリックします。
3. [* 新しい辞書エントリ *] ダイアログボックスで、必要な詳細を [* オブジェクトタイプの名前 *] フィールドと [* 概要 *] フィールドに入力します。
4. [* Scheme * (スキーム *)] フィールドで、次のいずれかの操作を実行します。
 - リストから使用可能なスキームのいずれかを選択します。
 - [新しいスキームの追加] をクリックし、[新しいスキーム] ダイアログボックスに必要な * スキーム名 * を入力し、[* 追加] をクリックします。
5. [* 行の追加 *] をクリックし、次の手順を実行して属性を説明します。
 - a. [* 名前 *] 列をクリックして、属性の名前を入力します。
 - b. [* タイプ *] 列で、必要なタイプを選択します。

文字列の長さ * 列が入力され、文字列をタイプとして選択した場合は編集可能になります。また、タイプとして * enum * を選択した場合は、* 値 * 列を編集できます。
 - c. 「ナチュラルキー」からキャッシュする属性の適切なチェックボックスを選択し、「ヌルカラム」にすることができます。
 - [* ナチュラルキー *] チェックボックスをオンにした場合、[* は NULL にできます *] チェックボックスはオンにできません。
 - d. ディクショナリオブジェクトに必要な属性を追加します。
 - e. * オプション : * ナチュラルキーを大文字と小文字を区別する場合は、* ナチュラルキーカラムの * 値を大文字と小文字を区別する * チェックボックスを選択します。
6. [保存 (Save)] をクリックします。

関数を作成します

タスクに適した WFA の事前定義された機能がない場合は、ユーティリティとして使用できる WFA の機能を作成できます。

必要なもの

関数を作成するには、MVEL 構文を知っている必要があります。

このタスクについて

関数の定義には、次のものを含める必要があります。

- name : 機能の名前

MVEL 構文では予約語を使用しないでください。各関数には一意の名前を付ける必要があります。

- MVEL 定義 : 関数定義の MVEL 構文を指定する文字列

手順

1. [* ワークフローデザイン > 関数 *] をクリックします。
2. をクリックします  をクリックします。
3. [関数 * (New Function *)] ダイアログボックスの [関数の定義 * (概要 *)] および [関数の定義 * (* Function definition *)] フィールドに必要な詳細を入力または選択します。

```
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);
}
```

関数名 * フィールドには、MVEL 構文で使用されるデータが入力されます。

4. * オプション : * Test * をクリックして、機能をテストします。
 - a. [* テスト *] ダイアログボックスの [式 *] セクションで、必要な関数式を入力します。

'actualVolumeSize(600,1)

- b. [* テスト *] をクリックします。

テスト結果が表示されます。

- c. ダイアログボックスを閉じます。

5. [保存 (Save)] をクリックします。

テンプレートを作成します

コマンドの詳細に属性を入力するための青写真として使用できるテンプレートを作成できます。

手順

1. [* ワークフローデザイン > テンプレート *] をクリックします。
 2. をクリックします  をクリックします。
 3. [新しいテンプレート *] ダイアログボックスで、[名前 *]、[* タイプ *]、および [* 概要 *] フィールドに必要な詳細を入力または選択します。
- 属性テーブルは、* タイプ * フィールドで選択した辞書オブジェクトに基づいて入力されます。
4. 各属性の値列をクリックし、次のいずれかを実行します。
 - ° リストから必要な値を入力または選択します。
 - ° ユーザー入力エントリーを入力しますたとえば 'サイズユーザー入力には '\$size' と入力します
 5. [保存 (Save)] をクリックします。

キャッシュクエリを作成します

データソースの種類から WFA データベース内のディクショナリオブジェクトに関する情報をキャッシュする場合は、キャッシュクエリを定義できます。キャッシュクエリを作成して、ディクショナリエントリと、Active IQ Unified Manager 6.1 などの 1 つ以上のデータソースタイプに関連付けることができます。

必要なもの

キャッシュクエリを作成するには、適切な SQL 構文を知っている必要があります。

手順

1. [* データソースデザイン > キャッシュクエリ *] をクリックします。
2. をクリックします  をクリックします。
3. [キャッシュクエリの追加 *] ダイアログボックスで、必要なディクショナリエントリとデータソースタイプを選択します。
4. [SQL SELECT QUERY] セクションで '適切な SQL クエリを入力します

次の SQL クエリは、Active IQ Unified Manager 6.1 データソースタイプからディスクディクショナリオブジェクトに関する情報をキャッシュします。

```

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

```

5. SQL クエリをテストする場合は、 * Test * をクリックします。

複数のデータソースタイプを選択した場合は、 [キャッシュクエリのテスト] ダイアログボックスが開き、必要なデータソースタイプを選択できます。

テスト結果が表示されます。

6. ダイアログボックスを閉じます。

7. [保存 (Save)] をクリックします。

定期的なスケジュールを作成

OnCommand Workflow Automation (WFA) には、ワークフロー用の 2 つのスケジュール設定オプションがあります。ワークフローを特定の時間に 1 回実行するようにスケジュールを設定することも、定期的なスケジュールを作成してワークフローにスケジュールを関連付けることもできます。

このタスクについて

作成したスケジュールは、再利用して複数のワークフローに関連付けることができます。

手順

1. [* 実行 > スケジュール *] をクリックします。
2. をクリックします  をクリックします。
3. [新しいスケジュール *] ダイアログ・ボックスで 'スケジュールの名前' '概要' '頻度' を入力または選択しま

す

頻度には、24時間形式で時刻を入力する必要があります。WFAサーバの時間がスケジュールに適用されます。

4. [OK]をクリックします。

完了後

- ワークフローを実行するときに、*再帰的に実行*オプションを使用してスケジュールをワークフローに関連付けることができます。
- ワークフローの詳細とそのスケジュールとの関連付けを表示するには、*Execution* > *Recurring Schedules*をクリックします。

一度実行されるようにスケジュールされたワークフローのリソースと実行の計画は、ワークフローがスケジュール設定されるとすぐに実行されます。ただし、繰り返しスケジュールが設定されたワークフローのリソースと実行計画は、スケジュールされた時刻に実行され、スケジュールがワークフローに関連付けられている時刻には実行されません。

フィルタルールを定義します

vFilerユニット、アグリゲート、仮想マシンなどのディクショナリエントリリソースをフィルタリングするための一連のルールを定義できます。既存のワークフローおよび新しいワークフローの作成時に、それらのワークフローのルールをカスタマイズできます。

手順

1. WebブラウザからadminとしてWFAにログインします。
2. [*ワークフローデザイン>ワークフロー*]をクリックします。
3. [*ワークフロー*]ウィンドウで、変更するワークフローをダブルクリックします。

ワークフロー <ワークフローナイム> ウィンドウが表示されます。

4. 次のいずれかのオプションを選択して、一連のルールを定義します。

状況	操作
行のコマンドが繰り返される場合は、リソースを検索する	<ol style="list-style-type: none">行番号をクリックし、*行の繰り返し*を選択します。[行の繰り返し]ダイアログボックスの[*リピート*]ドロップダウンリストから、グループ*の各リソースに対して*を選択します。リソースタイプを選択します。[検索条件の入力*]リンクをクリックします。

状況	操作
コマンド入力で必要なリソースを検索します	<p>a. をクリックします .</p> <p>b. [<コマンド名> のパラメータ (Parameters for <command_name>)] ダイアログボックスで、[定義 (Define)] <辞書オブジェクト*>* ドロップダウンリストから既存の <辞書オブジェクト*>* オプションを検索して * を選択します。</p> <p>c. [検索条件の入力*] リンクをクリックします。</p>
コマンド入力の変数で参照されているリソースを検索します	<p>a. をクリックします .</p> <p>b. [<コマンド名> のパラメータ (Parameters for <command_name>)] ダイアログボックスで、[属性* (attributes*)] オプションを [* 定義 <辞書オブジェクト*>* (* define <dictionary object*>*)] ドロップダウンリストから入力して * を選択します。</p> <p>c. をクリックします  をクリックします .</p>
コマンド名の文字列タイプを入力します	<p>a. をクリックします .</p> <p>b. [<コマンド名> のパラメータ (Parameters for <command_name>)] ダイアログボックスで、[属性* (attributes*)] オプションを [* 定義 <辞書オブジェクト*>* (* define <dictionary object*>*)] ドロップダウンリストから入力して * を選択します。</p> <p>c. をクリックします  文字列フィールド。</p>

5. [* リソースを選択* (* Resource Selection*)] ダイアログボックスで、[* フィルタルールを定義* (Define filter rules*)] チェックボックスを選択する。

[リソースを選択] ダイアログボックスの [ファインダ] ドロップダウンからいずれかのオプションを選択した場合、[フィルタルールを定義] チェックボックスは無効になります。フィルタルールの定義を有効にするには、Finder の値を「なし」に設定する必要があります。

6. ルールの属性、演算子、および値を入力します。

値は単一引用符で囲む必要があります。フィルタルールには 1 つ以上のグループを含めることができます。

7. [OK] をクリックします。

承認ポイントを追加します

ワークフローでは、承認ポイントをチェックポイントとして追加して、ワークフローの実行を一時停止し、承認に基づいて再開することができます。承認ポイントは、ワークフローの段階的な実行に使用できます。ワークフローのセクションは、特定の条件が満

たされた後にのみ実行されます。たとえば、次のセクションを承認する必要がある場合や、最初のセクションが正常に実行された場合などです。

手順

1. Web ブラウザから、アーキテクトまたは管理者として WFA にログインします。
2. [* ワークフローデザイン > ワークフロー *] をクリックします。
3. [* ワークフロー *] ウィンドウで、変更するワークフローをダブルクリックします。
4. [* ワークフロー < ワークフローネーム >] ウィンドウで、をクリックします  承認ポイントを追加するステップの左側にあるアイコン。

1 つ以上のステップの承認ポイントを追加できます。

5. [新しい承認ポイント * (* New Approval Point *)] ダイアログボックスで、コメントおよび条件の詳細を入力します。
6. [OK] をクリックします。

WFA のコーディングガイドライン

フィルタ、機能、コマンド、ワークフローなど、さまざまなビルディングブロックの作成に関する OnCommand Workflow Automation (WFA) のコーディングに関する一般的なガイドライン、命名規則、および推奨事項を理解しておく必要があります。

変数のガイドライン

コマンドまたはデータソースの種類を作成するときは、OnCommand Workflow Automation (WFA) で PowerShell 変数と Perl 変数のガイドラインに注意する必要があります。

PowerShell の変数

ガイドライン	例
スクリプト入力パラメータの場合： <ul style="list-style-type: none">• Pascal の事例を使用してください。• アンダースコアは使用しないでください。• 省略形は使用しないでください。	「\$VolumeName」 \$AutoDeleteOptions' 「\$Size」
スクリプト内部変数の場合： <ul style="list-style-type: none">• Camel case を使用します。• アンダースコアは使用しないでください。• 省略形は使用しないでください。	「\$newVolume」 「\$qtreeName」 「\$TIME」

ガイドライン	例
関数の場合： <ul style="list-style-type: none">Pascal の事例を使用してください。アンダースコアは使用しないでください。省略形は使用しないでください。	「GetVolumeSize」
変数名では大文字と小文字は区別されません。ただし、読みやすくするために、同じ名前に異なる大文字と小文字を使用しないでください。	「\$VARIABLE」は「\$VARIABLE」と同じです
変数名は、プレーン英語で記述され、スクリプトの機能に関連した名前にする必要があります。	「\$a」ではなく「\$name」を使用してください
各変数のデータ型を明示的に宣言します。	[string] name [int] サイズ
特殊文字 (! @ # & % 、 .) とスペース。	なし
PowerShell の予約キーワードは使用しないでください。	なし
入力パラメータをグループ化するには、まず必須パラメータを配置し、続けてオプションパラメータを配置します。	<pre>param([parameter(Mandatory=\$true)] [string]\$Type, [parameter(Mandatory=\$true)] [string]\$Ip, [parameter(Mandatory=\$false)] [string]\$VolumeName)</pre>
すべての入力変数には 'HelpMessage' 注釈と意味のあるヘルプ・メッセージを使用してコメントを付けています	<pre>[parameter(Mandatory=\$false,HelpMessage="LUN to map")] [string]\$LUNName</pre>
変数名として「ファイル」を使用しないでください。代わりに「アレイ」を使用してください。	なし

ガイドライン	例
<p>引数が列挙値を取得する場合は 'ValidateSet' 注釈を使用しますこれにより、パラメータの Enum データ型に自動的に変換されます。</p>	<pre>[parameter (Mandatory=\$false, HelpMessage="Volume state")] [ValidateSet ("online", "offline", "restricted")] [string]\$State</pre>
<p>パラメータの末尾に「_Capacity」が付いたエイリアスを追加して、パラメータが容量タイプであることを示します。</p>	<p>「Create Volume」コマンドでは、次のようにエイリアスを使用します。</p> <pre>[parameter (Mandatory=\$false, HelpMessage="Volume increment size in MB")] [Alias("AutosizeIncrementSize_Capacity")] [int]\$AutosizeIncrementSize</pre>
<p>パラメータがパスワードタイプであることを示すために、エイリアスを "_Password" で終わるパラメータに追加します。</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>

Perl 変数

ガイドライン	例
<p>スクリプト入力パラメータの場合：</p> <ul style="list-style-type: none"> • Pascal の事例を使用してください。 • アンダースコアは使用しないでください。 • 省略形は使用しないでください。 	<pre>「\$VolumeName」 \$AutoDeleteOptions' 「\$Size」</pre>

ガイドライン	例
スクリプトの内部変数には省略形を使用しないでください。	\$new_volume 「\$qtree_name」のようになります 「\$TIME」
関数には省略形を使用しないでください。	'get_volume_size'
変数名では大文字と小文字が区別されます。読みやすくするために、同じ名前に異なる大文字と小文字を使用しないでください。	「\$VARIABLE」は「\$VARIABLE」と同じではありません
変数名は、プレーン英語で記述され、スクリプトの機能に関連した名前にする必要があります。	「\$a」ではなく「\$name」を使用してください
入力パラメータをグループ化するには、まず必須パラメータを配置し、続けてオプションパラメータを配置します。	なし
GetOptions 関数で、入力パラメータの各変数のデータ型を明示的に宣言します。	<pre>GetOptions ("Name=s"=>\\$Name, "Size=i"=>\\$Size)</pre>
変数名として「ファイル」を使用しないでください。代わりに「アレイ」を使用してください。	なし
Perl には '列挙値の 'ValidateSet' 注釈は含まれません 引数が列挙値を取得する場合は '明示的な if 文を使用します	<pre>if (defined\$SpaceGuarantee && !(\$SpaceGuarantee eq 'none')) { die'Illegal SpaceGuarantee argument: \".\$SpaceGuarantee.\"';</pre>
	\$SpaceGuarantee eq 'volume' \$SpaceGuarantee eq 'file') { die'Illegal SpaceGuarantee argument: \".\$SpaceGuarantee.\"'; } ----

ガイドライン	例
<p>すべての Perl WFA コマンドでは、変数、参照、サブルーチンに安全でない構成要素を使用しないようにするために、"strict" プラグマを使用する必要があります。</p>	<pre>use strict; # the above is equivalent to use strictvars; use strictsubs; use strictrefs;</pre>
<p>すべての Perl WFA コマンドでは、次の Perl モジュールを使用する必要があります。</p> <ul style="list-style-type: none"> • getopt <p>これは、入力パラメータの指定に使用されます。</p> <ul style="list-style-type: none"> • WFAUtil のようになります <p>コマンドロギング、コマンドの進捗状況の報告、アレイコントローラへの接続などに使用されるユーティリティ機能に使用されます。</p>	<pre>use Getopt::Long; use NaServer; use WFAUtil;</pre>

インデントのガイドライン

OnCommand Workflow Automation (WFA) 用の PowerShell または Perl スクリプトを作成する場合は、インデント設定のガイドラインに注意する必要があります。

ガイドライン	例
タブは、4つの空白スペースに等しい。	

ガイドライン	例
ブロックの先頭と末尾を表示するには、タブと波っこを使用します。	PowerShell スクリプト <pre> if (\$pair.length-ne 2) { throw "Got wrong input data" } </pre> Perl スクリプト <pre> if defined \$MaxDirectorySize { # convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024; } </pre>
オペレーションのセットまたはコードのチャンク間に空白行を追加します。	<pre> \$options=\$option.trim(); \$pair=\$option.split(" "); Get-WFALogger -Info -messages \$("split options: "+ \$Pair) </pre>

コメントのガイドライン

OnCommand Workflow Automation (WFA) 用スクリプトでの PowerShell と Perl のコメントに関するガイドラインに注意する必要があります。

PowerShell のコメントを表示します

ガイドライン	例
1 行のコメントには # 文字を使用します。	<pre> # Single line comment \$options=\$option.trim(); </pre>

ガイドライン	例
行末のコメントには # 文字を使用します。	<pre>\$options=\$option.trim(); # End of line comment</pre>
ブロックコメントには、との文字を使用します。	<pre><# This is a block comment #> \$options=\$option.trim();</pre>

Perl のコメント

ガイドライン	例
1行のコメントには # 文字を使用します。	<pre># convert from MBytes to Bytes my \$MaxDirectorySizeBytes = \$MaxDirectorySize * 1024 * 1024;</pre>
行末のコメントには # 文字を使用します。	<pre>my \$MaxDirectorySizeBytes = \$MaxDirect orySize * 1024 * 1024; # convert to Bytes</pre>
先頭と末尾に空白の # を含む各行に # 文字を使用し て、複数行コメントのコメント境界を作成します。	<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>

ガイドライン	例
WFA コマンドには、コメント化したコードやデッドコードを含めないでください。ただし、テスト目的では、Plain Old Documentation (PoD) メカニズムを使用してコードをコメントアウトできます。	<pre>=begin comment # Set deduplication if(defined \$Deduplication && \$Deduplication eq "enabled") { \$wfaUtil- >sendLog ("Enabling Deduplication"); } =end comment =cut</pre>

ロギングのガイドライン

OnCommand Workflow Automation (WFA) 用の PowerShell スクリプトまたは Perl スクリプトを作成する際には、ログ記録に関するガイドラインに注意する必要があります。

PowerShell のロギング

ガイドライン	例
ログには Get-WFALogger コマンドレットを使用します。	<pre>Get-WFALogger -Info -message "Creating volume"</pre>
Data ONTAP、VMware、PowerCLI などの内部パッケージとの対話を必要とするすべてのアクションをログに記録します。すべてのログメッセージは、ワークフローの実行ステータス履歴の実行ログで使用できます。	なし
内部パッケージに渡される関連するすべての引数を記録します。	なし
使用状況に応じて、Get-WFALogger コマンドレットを使用する場合は、適切なログレベルを使用してください。-INFO、-Error、-Warn、-Debug は、使用可能なさまざまなログレベルです。ログレベルが指定されていない場合、デフォルトのログレベルは Debug です。	なし

Perl のロギング

ガイドライン	例
ログには WFAUtil sendLog を使用します。	<pre>my wfa_util = WFAUtil->new(); eval { \$wfa_util->sendLog('INFO', "Connecting to the cluster: \$DestinationCluster"); }</pre>
Data ONTAP、VMware、WFA など、コマンドの外部にある処理とのやり取りが必要なすべてのアクションをログに記録します。WFAUtil sendLog ルーチンを使用して作成するログメッセージは、すべて WFA データベースに格納されます。これらのログメッセージは、実行されるワークフローとコマンドで使用できます。	なし
呼び出されたルーチンに渡されたすべての関連引数を記録します。	なし
適切なログレベルを使用します。-情報、-エラー、-警告、-デバッグは、使用可能なさまざまなログレベルです。	なし
-Info レベルでロギングする場合は、正確かつ簡潔にしてください。ログメッセージにクラス名や関数名などの実装の詳細を指定しないでください。正確な手順またはエラーの正確な説明を英語で入力してください。	<p>次のコードスニペットは、正常なメッセージと不正なメッセージの例を示しています。</p> <pre>\$wfa_util->sendLog('WARN', 'Removing volume: ' . \$VolumeName); # Good Message</pre> <pre>\$wfa_util->sendLog('WARN', 'Invoking volume- destroy ZAPI: ' . \$VolumeName); # Bad message</pre>

エラー処理のガイドライン

OnCommand Workflow Automation (WFA) 用の PowerShell または Perl スクリプトを作成する際に、エラー処理に関するガイドラインに注意する必要があります。

PowerShell のエラー処理

ガイドライン	例
<p>PowerShell ランタイムによってコマンドレットに追加される共通パラメータには、 ErrorAction や WarningAction などのエラー処理パラメータがあります。</p> <ul style="list-style-type: none">• ErrorAction パラメータは、コマンドレットが終了しないエラーに応答する方法を決定します。• WarningAction パラメータは、コマンドレットがコマンドレットの警告にどのように対応するかを決定します。• Stop 、 SilentlyContinue 、 inquire 、 Continue は、 ErrorAction パラメータおよび WarningAction パラメータの有効な値です。 <p>詳細については、 PowerShell CLI の「 Get-Help About_CommonParameters 」コマンドを使用してください。</p>	<p>ErrorAction :次の例は、終了しないエラーを終了エラーとして処理する方法を示しています。</p> <pre>New-NcIgroup-Name \$IgroupName-Protocol \$Protocol-Type\$OSType-ErrorActionstop</pre>
<p>着信例外のタイプが不明な場合は、一般的な "try/catch" ステートメントを使用します。</p>	<p>警告アクション</p> <pre>New-VM-Name \$VMName-VM \$SourceVM-DataStore\$DataStoreName-VMHost\$VMHost-WarningActionSilentlyContinue</pre>
<p>着信例外のタイプがわかっている場合は、特定の「 try/catch 」ステートメントを使用します。</p>	<pre>try { "In Try/catch block" } catch { "Got exception" }</pre>
	<pre>try { "In Try/catch block" } catch[System.Net.WebException], [System.IO. IOException] { "Got exception" }</pre>

ガイドライン	例
<p>「finally」文を使ってリソースを解放します。</p>	<pre data-bbox="845 196 1209 671"> try { "In Try/catch block" } catch { "Got exception" } finally { "Release resources" }</pre>
<p>PowerShell の自動変数を使用して、例外に関する情報にアクセスします。</p>	<pre data-bbox="845 798 1416 1294"> try { Get-WFALogger -Info -message \$(\$("Creating Ipspace: " + \$Ipspace) New-NaNetIpspace-Name \$Ipspace } catch { Throw "Failed to create Ipspace. Message: " + \$_.Exception.Message; }</pre>

Perl エラー処理

ガイドライン	例
<p>Perl には、 try/catch ブロックに対するネイティブ言語サポートは含まれていません。 eval ブロックを使用して、エラーの確認と処理を行います。評価ブロックはできるだけ小さくしてください。</p>	<pre data-bbox="845 192 1454 1262"> 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>

WFA での PowerShell と Perl の一般的な規則

既存のスクリプトと整合性のあるスクリプトを作成するために、 WFA で使用される PowerShell と Perl の特定の規則を理解しておく必要があります。

- ・スクリプトで何を実行するかを明確にするために役立つ変数を使用します。
- ・コメントなしで理解できる読み取り可能なコードを記述してください。
- ・スクリプトとコマンドはできるだけシンプルにしてください。
- ・PowerShell スクリプトの場合：
 - 可能なかぎりコマンドレットを使用してください。
 - 使用可能なコマンドレットがない場合は、 .NET コードを呼び出します。
- ・Perl スクリプトの場合：

- 改行文字を含む "die ``" ステートメントは必ず終了してください。

改行文字が含まれていない場合は、スクリプトの行番号が出力されます。これは、WFA で実行する Perl コマンドのデバッグには役立ちません。

- 「getopt」モジュールで、文字列引数をコマンドに必須にします。

Windows にバンドルされている Perl モジュール

一部の Perl モジュールは、OnCommand Workflow Automation (WFA) 用の Windows Active 状態 Perl ディストリビューションにバンドルされています。これらの Perl モジュールは、Windows に付属している場合にのみ、コマンドの記述に Perl コードで使用できます。

次の表に、Windows for WFA にバンドルされている Perl データベースモジュールを示します。

データベースモジュール	説明
DBD::mysql	MySQL データベースへの接続を可能にする Perl5 データベースインターフェースドライバ。
試してみましょう	評価ブロックを使用して一般的なミスを最小限に抑えます
XML::libxml	DOM、SAX、XMLReader インターフェイスを持つ XML および HTML パーサーを提供する libxml2 へのインターフェイス。
DBD : Cassandra	CQL3 クエリ言語を使用する Cassandra 用の Perl5 データベースインターフェースドライバ。

カスタムの PowerShell モジュールと Perl モジュールを追加する場合の考慮事項

OnCommand Workflow Automation (WFA) に PowerShell および Perl のカスタムモジュールを追加する前に、一定の考慮事項について理解しておく必要があります。カスタムの PowerShell モジュールと Perl モジュールを使用すると、ワークフローを作成するためのカスタムコマンドを使用できます。

- WFA コマンドの実行中に、すべてのカスタム PowerShell モジュールが WFA インストールディレクトリ「/posh/modules」に自動的にインポートされます。
- 「wfa/perl」ディレクトリに追加されたすべてのカスタム Perl モジュールは、_@INC_library に含まれています。
- カスタムの PowerShell モジュールと Perl モジュールは、WFA のバックアップ処理の一環としてバックアップされません。
- カスタムの PowerShell モジュールおよび Perl モジュールは、WFA のリストア処理中にリストアされません。

新しい WFA にコピーするには、カスタムの PowerShell モジュールと Perl モジュールを手動でバックアップする必要があります。

modules ディレクトリ内のフォルダ名は ' モジュール名と同じである必要があります

WFA のコマンドレットと機能

OnCommand Workflow Automation (WFA) には、複数の PowerShell コマンドレットと、WFA コマンドで使用できる PowerShell および Perl の機能が用意されています。

WFA サーバが提供するすべての PowerShell コマンドレットと機能を表示するには、次の PowerShell コマンドを使用します。

- `get-command - モジュール WFAWrapper`
- 「`getcommand - Module WFA`」

WFA サーバが提供する Perl の機能は、すべて「`WFAUtil.pm`」モジュールで確認できます。WFA ヘルプモジュールのサポートリンクにあるヘルプセクション、WFA PowerShell コマンドレットヘルプと WFA Perl メソッドを使用すると、PowerShell のすべてのコマンドレットと機能、および Perl の機能にアクセスできます。

PowerShell および Perl WFA モジュール

ワークフローのスクリプトを作成するには、OnCommand Workflow Automation (WFA) 用の PowerShell または Perl モジュールを理解しておく必要があります。

PowerShell モジュール

ガイドライン	例
Data ONTAP PS Toolkit を使用して、Toolkit が使用可能になったときにいつでも API を呼び出すことができます。	[Add VLAN] コマンドでは ' 次のようにツールキットを使用します 「 <code>Add-NaNetVlan-Interface \$Interface-VLANs \$VlanID</code> 」
Data ONTAP PS Toolkit で使用できるコマンドレットがない場合は、「 <code>Invoke-NaSSH</code> 」コマンドを使用して、Data ONTAP で CLI を呼び出します。	<code>Invoke-NaSSH -Name\$ArrayName</code> コマンド「 <code>ifconfig -a -Credential \$Credentials</code> 」が実行されます

Perl モジュール

NaServer モジュールは WFA のコマンドで使用されます。NaServer モジュールを使用すると、Data ONTAP システムのアクティブ管理で使用される Data ONTAP API の呼び出しが可能になります。

ガイドライン	例
NetApp Manageability SDK が使用可能な場合は、NaServer モジュールを使用して API を呼び出します。	<p>以下に、SnapMirror の再開処理に NaServer モジュールを使用する例を示します。</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>

ガイドライン	例
<p>Data ONTAP API が使用できない場合は、<code>executeSystemCli</code> ユーティリティメソッドを使用して Data ONTAP CLI を呼び出します。</p> <p> <code>executeSystemCli</code> はサポートされておらず、現在は 7-Mode の Data ONTAP でのみ使用できます。</p>	なし

PowerShell コマンドを Perl に変換する際の考慮事項

PowerShell と Perl の機能は異なるため、PowerShell コマンドを Perl に変換する場合は、いくつかの重要な考慮事項に注意する必要があります。

コマンド入力タイプ

OnCommand Workflow Automation (WFA) を使用すると、ワークフローの設計者は、コマンドを定義する際に、コマンドの入力としてアレイとハッシュを使用できます。これらの入力タイプは、Perl を使用してコマンドを定義する場合には使用できません。Perl コマンドで配列とハッシュの入力を受け入れる場合は、デザイナで文字列として入力を定義できます。コマンド定義では、入力を解析できます。この入力は必要に応じて配列またはハッシュを作成するために渡されます。入力の概要は、入力が想定される形式です。

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

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

PowerShell ステートメント

次の例は、PowerShell および Perl へのアレイ入力の受け渡し方法を示しています。例では、cron ジョブの実行がスケジュールされている月を指定する、CronMonth 入力について説明しています。有効な値は、整数 -1 ~ 11 です。値 -1 は、スケジュールが毎月実行されることを示します。その他の値は特定の月を表し、0 は 1 月、11 は 12 月を表します。

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

Perl ステートメント

```

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
}

```

コマンド定義

PowerShell でパイプ演算子を使用する 1 行式を、Perl で複数のステートメントブロックに拡張して、同じ機能を実現しなければならない場合があります。次の表に、待機コマンドの例を示します。

PowerShell ステートメント	Perl ステートメント
<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>

WFA のビルディングブロックに関するガイドライン

Workflow Automation ビルディングブロックの使用に関するガイドラインを確認しておく必要があります。

WFA の SQL に関するガイドライン

OnCommand Workflow Automation (WFA) で SQL を使用して WFA 用の SQL クエリを記述する際のガイドラインに注意する必要があります。

WFA では、SQL は次の場所で使用されます。

- 選択のためのユーザ入力を入力する SQL クエリ
- 特定のディクショナリエントリタイプのオブジェクトをフィルタリングするフィルタを作成するための SQL クエリ
- プレイグラウンドデータベース内のテーブル内の静的データ
- カスタム構成管理データベース (CMDB) などの外部データソースからデータを抽出する必要がある SQL タイプのカスタムデータソース。
- 予約および検証スクリプトを照会する SQL です

ガイドライン	例
SQL の予約キーワードは大文字で入力する必要があります。	<pre>SELECT vserver.name FROM cm_storage.vserver vserver</pre>
テーブル名と列名は小文字で入力する必要があります。	表：アグリゲート 列：Used_space_MB
単語はアンダースコア (_) 文字で区切れます。スペースは使用できません。	array_performance
テーブル名が単数で定義されています。テーブルは 1 つ以上のエントリの集合です。	「関数」ではなく「関数」

ガイドライン	例
SELECT クエリで意味のある名前を持つテーブルエイリアスを使用します	<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>

ガイドライン	例
<p>フィルタクエリまたはユーザクエリでフィルタ入力パラメータまたはユーザ入力パラメータを参照する必要がある場合は、構文を「<code> \${inputVariableName}</code>」として使用します。予約スクリプトおよび検証スクリプトでコマンド定義パラメータを参照する場合は、構文を使用することもできます。</p>	<pre data-bbox="845 192 1437 1284"> 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>複雑なクエリにはコメントを使用します。クエリでサポートされているコメントスタイルの一部を次に示します。</p> <ul style="list-style-type: none"> ```-- 行の最後まで <p>このコメントスタイルでは、2番目のハイフンの後にスペースは必要です。</p> <ul style="list-style-type: none"> 行の末尾までの「#」文字から "/" から次の "/" シーケンスに移動します 	<pre data-bbox="845 1389 1421 1826"> /* multi-line comment */ --line comment SELECT ip as ip, # comment till end of this line NAME as name FROM --end of line comment storage.array </pre>

WFA の機能に関するガイドラインを参照してください

よく使用されるより複雑なロジックを名前付き関数にカプセル化する関数を作成し、その関数をコマンドパラメータ値として再利用したり、 OnCommand Workflow Automation (WFA) でパラメータ値をフィルタリングしたりすることができます。

ガイドライン	例
関数名には Camel case を使用します。	計算ボリュームサイズ
変数名は、標準的な英語で、関数の機能に関連している必要があります。	splitByDelimiter
省略形は使用しないでください。	calculateVolumeSize、_not_calcVolSize
関数は、 MVEL (MVEL) を使用して定義します。	なし
関数定義は、公式の Java プログラミング言語のガイドラインに従って指定する必要があります。	なし

WFA ディクショナリエントリのガイドライン

OnCommand Workflow Automation (WFA) でディクショナリエントリを作成するためのガイドラインを確認しておく必要があります。

ガイドライン	例
ディクショナリエントリ名には、英数字とアンダースコアのみを使用する必要があります。	Cluster License (クラスタライセンス) switch_23
辞書エントリ名の先頭は大文字にする必要があります。名前のすべての単語の先頭には大文字を入力し、各単語をアンダースコア (_) で区切ります。	ボリューム array_License
ディクショナリエントリの属性名にディクショナリエントリの名前を含めることはできません。	なし
ディクショナリエントリ内の属性と参照は、小文字で記述する必要があります。	アグリゲート、 size_MB
単語はアンダースコアで区切ります。スペースは使用できません。	resource_pool を指定します

ガイドライン	例
辞書エントリには、別のスキームからの参照を含めることはできません。ディクショナリエントリが別のスキームのオブジェクトへの相互参照を必要とする場合は、参照されるオブジェクトのすべての自然キーがディクショナリエントリに存在することを確認します。	array_Performance ディクショナリエントリでは、Array ディクショナリエントリのすべての自然キーが直接属性として必要になります。
属性に適切なデータ型を使用します。	なし
サイズやスペースに関連する属性には、長いデータ型を使用します。	storage.Volume 辞書エントリの size_mb と available_size MB です
属性の値が固定されている場合は、Enum を使用します。	storage.1 の raid_type。ボリューム辞書エントリ
データソースがその属性または参照の値を提供する場合は、属性または参照の「キャッシュする」を true に設定します。Active IQ Unified Manager データソースの場合、データソースがその属性に値を提供できる場合は、キャッシュ可能な属性を追加します。	なし
この属性または参照の値を提供するデータソースが NULL を返す可能性がある場合は " " を null にすることができます	なし
各属性と参照に意味のある概要を指定します。ワークフローを設計する際には、コマンドの詳細に概要が表示されます。	なし
ディクショナリエントリ内の属性の名前として「id」を使用しないでください。WFA の内部使用のために予約されています。	なし

- ・関連情報 *

[学習資料への参照](#)

コマンドのガイドライン

OnCommand Workflow Automation (WFA) でコマンドを作成する際のガイドラインに注意する必要があります。

ガイドライン	例
コマンドには、わかりやすい名前を使用します。	qtree を作成します

ガイドライン	例
単語を区切るにはスペースを使用します。各単語は大文字で始まる必要があります。	「ボリュームを作成」
オプションパラメータで想定される結果など、コマンドの機能について説明する概要を提供します。	なし
デフォルトでは、標準コマンドのタイムアウトは 600 秒です。コマンドの作成時にデフォルトのタイムアウトが設定されます。デフォルト値を変更するのには、コマンドの完了に時間がかかる場合だけにしてください。	[Create Volume] コマンドを使用します
長時間の処理の場合は、2つのコマンドを作成します。1つは長時間の処理を実行するコマンドで、もう1つは処理の進捗状況を定期的に報告するコマンドです。最初のコマンドは「標準実行」コマンドタイプで、2番目のコマンドは「条件を待機」コマンドタイプである必要があります。	'Create VSM' および 'Wait for VSM' コマンド
容易に識別できるように 'wait for condition' コマンド名の前には 'wait' を付けます	「CM ボリューム移動の待機」
「条件の待機」コマンドに適切な待機間隔を使用します。指定した値は、長時間の処理が完了したかどうかを確認するためにポーリングコマンドが実行される間隔を制御します。	'Wait for VSM' コマンドの 60 秒のサンプリング間隔
「条件待機」コマンドでは、長時間実行動作が完了するまでの予想時間に基づいて適切なタイムアウトを使用します。ネットワーク経由のデータ転送を実行する場合は、想定時間が大幅に長くなる可能性があります。	VSM ベースライン転送が完了するまでに数日かかることがあります。したがって、指定されたタイムアウトは 6 日です。

文字列表現

コマンドの文字列表現は、計画および実行中にワークフロー設計内のコマンドの詳細を表示します。コマンドの文字列表現で使用できるのは、コマンドパラメータだけです。

ガイドライン	例
値のない属性は使用しないでください。値のない属性は NA と表示されます。	volname 10.68.66.212 [NA]aggr1 / testVol7
文字列表現では、[],/ の区切り記号を使用して異なるエントリを区切ります。	ArrayName [ArrayIp]

ガイドライン	例
文字列表現のすべての値に意味のあるラベルを指定します。	<code>Volume name=VolumeName</code>

コマンド定義言語

コマンドは、次のサポートされているスクリプト言語を使用して記述できます。

- PowerShell
- Perl の場合

コマンドパラメータの定義

コマンドパラメータは、名前（Name）、パラメータ（概要）、タイプ（Type）、パラメータのデフォルト値、およびパラメータが必須かどうかによって記述されます。パラメータタイプは、String、Boolean、Integer、Long、Double、列挙、日時、容量、アレイ、ハッシュテーブル、パスワード、または XmlDocument。ほとんどの型の値は直感的ですが、Array と Hashtable の値は次の表に示す特定の形式にする必要があります。

ガイドライン	例
Array 入力タイプの値がカンマで区切った値のリストであることを確認します。	<pre>[parameter(Mandatory=\$false, HelpMessage="Months in which the schedule executes.")] [array]\$CronMonths</pre> <p>入力は 0、3、6、9 のように渡されます</p>
Hashtable 入力型の値が、セミコロンで区切られた key= 値のペアのリストであることを確認します。	<pre>[parameter(Mandatory=\$false, HelpMessage="Volume names and size (in MB)")] [hashtable]\$VolumeNamesAndSize</pre> <p>指定するパスは、次のとおりです。ボリューム 1 = 100、ボリューム 2 = 250、ボリューム 3 = 50</p>

ワークフローのガイドライン

OnCommand Workflow Automation（WFA）の事前定義されたワークフローの作成または変更に関するガイドラインを確認しておく必要があります。

一般的なガイドライン

ガイドライン	例
ストレージオペレータが実行した処理が反映されるように、ワークフローに名前を付けます。	CIFS 共有を作成します
ワークフロー名では、最初の単語の最初の文字とオブジェクトであるすべての単語を大文字にします。略語や頭字語の文字を大文字にします。	ボリューム qtree clustered Data ONTAP の qtree CIFS 共有を作成
ワークフローの説明には、ワークフローの前提条件、ワークフローの結果、または条件付きの実行など、ワークフローのすべての重要な手順を含めます。	前提条件を含むサンプルワークフロー「clustered Data ONTAP ストレージでの VMware NFS データストアの作成」の概要 を参照してください。
ワークフローがプロダクションの準備ができていて、ポータルページに表示できる場合にのみ、「Ready for Production」を「true」に設定します。	なし
デフォルトでは'予約済み要素の考慮を true に設定します実行用のワークフローをプレビューするとき、WFA プランナーは、キャッシュデータベース内の既存のオブジェクトと一緒に予約されているすべてのオブジェクトを考慮します。このオプションを「true」に設定すると、特定のワークフローを計画するときに、他のスケジュールされたワークフローまたは並列実行ワークフローの影響が考慮されます	<ul style="list-style-type: none"> シナリオ 1 <p>ワークフロー 1 ではボリュームを作成し、1 週間後に実行するようにスケジュール設定します。ワークフロー 2 では、検索対象のボリュームに qtree または LUN が作成されます。ワークフロー 2 が 1 日以内に実行される場合、ワークフロー 2 で「予約済み要素の考慮」をオフにして、1 週間に作成するボリュームを考慮しないようにする必要があります。</p> <ul style="list-style-type: none"> シナリオ 2 <p>ワークフロー 1 では 'Create Volume' コマンドを使用しますスケジュール設定されたワークフロー 2 でアグリゲートの 100GB を消費する場合は、計画段階でワークフロー 2 の要件を考慮する必要があります。</p>

ガイドライン	例
<p>デフォルトでは、「エレメントの存在検証を有効にする」は「真」に設定されています</p>	<ul style="list-style-type: none"> シナリオ 1 ボリュームが存在する場合にのみ 'ボリュームを削除' コマンドを使用してボリュームを名前で削除するワークフローを作成し、そのボリュームを作成ボリュームまたはクローンボリュームなどの別のコマンドを使用して再作成する場合、'ワークフロー' ではこのフラグを使用しないでください。ボリュームを削除した場合の効果は 'Create volume' コマンドでは使用できないため、'ワークフロー' は失敗します シナリオ 2 Create Volume コマンドは 'vol198' という名前のワークフローで使用されます このオプションが true に設定されている場合、WFA プランナーは、計画時に、その名前を使用してボリュームが特定のアレイに存在するかどうかを確認します。計画中にボリュームが存在する場合、ワークフローが失敗します。
<p>ワークフローで同じコマンドを複数回選択する場合は、コマンドインスタンスに適切な表示名を指定します。</p>	<p>サンプル・ワークフローでは 'Create Volume' コマンドを 2 回使用して 'LUN の作成' マッピング、保護を SnapVault で行います。ただし、'プライマリ・ボリュームとミラー・デスティネーション・ボリューム' には、'プライマリ・ボリュームの作成' と 'セカンダリ・ボリュームの作成' が適切に使用されます。</p>

ユーザ入力

ガイドライン	例
<p>名前：</p> <ul style="list-style-type: none"> 名前の先頭には「\$」文字を付けます。 各単語の先頭に大文字を使用します。 すべての用語や略語に大文字を使用します。 アンダースコアは使用しないでください。 	<p>「\$Array」</p> <p>「\$VolumeName」</p>

ガイドライン	例
表示名：	<p>「ボリューム名」</p> <p>「ボリューム・サイズ（MB）」</p>
説明：	<p>「igroup」に追加するイニシエータたとえば、イニシエータの IQN や WWPN などです。</p> <p>・ユーザ入力ごとにわかりやすい概要を指定します。</p> <p>・必要に応じて例を挙げてください。</p> <p>これは、特にユーザ入力が特定の形式であると想定される場合に実行してください。</p>
ユーザー入力の説明は、ワークフローの実行中にユーザー入力のツールチップとして表示されます。	
Type：入力を特定の値セットに制限する場合は、タイプとして Enum を選択します。	プロトコル："iSCSI\"", "FCP\"", "MIXED\""
Type：ユーザが WFA キャッシュの値から選択できる場合は、タイプとして Query を選択します。	<p>\$Array: クエリのクエリタイプ：</p> <pre>SELECT ip, name FROM storage.array</pre>
[タイプ]：ユーザー入力がクエリから取得した値に制限されているか、サポートされている列挙型のみに制限されている必要がある場合に、ユーザー入力をロック済みとしてマークします。	\$Array：ロックされたクエリタイプ：キャッシュ内のアレイのみ選択できます。\$Protocol：有効な値が iSCSI、FCP、mixed のロックされた Enum タイプです。有効な値以外の値はサポートされません。
タイプ：クエリタイプクエリ演算子がユーザ入力を適切に選択できるようになると、クエリに戻り値として列を追加できます。	\$ Aggregate：アグリゲートを選択する前に属性を確認できるように、名前と合計サイズ、使用可能なサイズを指定します。

ガイドライン	例
<p>タイプ：クエリータイプユーザ入力の SQL クエリーは、その前にある他のユーザ入力を参照できます。この機能を使用すると、アレイの vFiler ユニット、アグリゲートのボリューム、Storage Virtual Machine (SVM) の LUN など、他のユーザ入力に基づいてクエリの結果を制限できます。</p>	<p>サンプル・ワークフローでは 'Create a Clustered Data ONTAP Volume] で 'VserverName のクエリは次のようにになります</p> <pre data-bbox="850 333 1416 846"> 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>クエリは \${\{ clustername \}} を参照します。 \$clustername は、\$VserverName ユーザ入力の前に 入力されたユーザ名です。</p>
<p>type：ブール型を使用し、ブール型の値を "true"、"false" として使用します。これにより、ユーザ入力を直接使用して、ワークフロー設計で内部式を記述できます。たとえば、\$UserInputName='Yes' ではなく \$UserInputName とします。</p>	<p>\$CreateCIFSShare: 有効な値が「true」または「false」のブール型</p>
<p>[タイプ]: 文字列および数値型の場合、特定の形式で値を検証するときは、[値] 列で正規表現を使用します。</p> <p>IP アドレスとネットワークマスクの入力には正規表現を使用します。</p>	<p>場所に固有のユーザ入力は、「[A-Z][A-Z]-0[1-9]」と表現できます。このユーザー入力は "US-01"、"NB-02" などの値を受け入れますが、"nb-00" は受け入れません。</p>
<p>[タイプ]: 数値タイプの場合、[値] 列で範囲ベースの検証を指定できます。</p>	<p>作成する LUN の数については、「値」列のエントリは 1~20 です。</p>
<p>グループ：グループに関連するユーザが該当するバケットに入力し、グループに名前を付けます。</p>	<p>ストレージ関連のすべてのユーザー入力用の「ストレージの詳細」。VMware 関連のすべてのユーザー入力の「データストアの詳細」。</p>

ガイドライン	例
必須：ワークフローを実行するためにユーザ入力の値が必要な場合は、ユーザ入力を必須としてマークします。これにより、ユーザ入力画面がユーザからの入力を受け入れられるようになります。	「Create NFS Volume」ワークフローの「\$VolumeName」
デフォルト値：ユーザ入力にデフォルト値があり、ほとんどのワークフロー実行で有効な場合は、値を指定します。これにより、デフォルトで目的が達成された場合に、実行中に入力を減らすことができます。	なし

定数、変数、および戻りパラメータ

ガイドライン	例
定数：複数のコマンドにパラメータを定義するために共通の値を使用する場合は、定数を定義します。	SnapVault サンプル・ワークフローでの LUN の作成 'マッピング' 保護については 'aggregate_Oオーバーコミットメント_threshold' を参照してください
定数：名前	<p><i>aggregate_Used_space_threshold</i></p> <p><i>ActualVolumeSizeInMB</i></p>
変数：コマンドパラメータのいずれかのボックスで定義されたオブジェクトに名前を指定します。変数は自動的に生成される名前で、変更できます。	なし
変数：変数名には小文字を使用します。	<p>ボリューム 1</p> <p>cifs_share</p>
戻りパラメータ：ワークフローの計画と実行で、計画中に計算値または選択した値が返される場合は、戻りパラメータを使用します。ワークフローが Web サービスから実行されたときにも、プレビューモードで値が使用可能になります。	アグリゲート：リソース選択ロジックを使用してアグリゲートを選択した場合、選択した実際のアグリゲートを戻りパラメータとして定義できます。

リモートシステムタイプの検証スクリプトを作成する際のガイドライン

OnCommand Workflow Automation (WFA) で定義したリモートシステムタイプをテストするための検証スクリプトを作成する際のガイドラインを確認しておく必要があります。

- 作成する Perl スクリプトは、[検証スクリプト] ウィンドウに表示されるサンプルスクリプトに似ている必要があります。
- 検証スクリプトの出力は、サンプルスクリプトの出力と同様である必要があります。

サンプルの検証スクリプト

```
# 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;
}
```

データソースタイプの作成に関するガイドライン

OnCommand Workflow Automation (WFA) のカスタムデータソースの定義に使用するデータソースタイプの作成に関するガイドラインを確認しておく必要があります。

データソースタイプは、次のいずれかの方法で定義できます。

- SQL : WFA の SQL ガイドラインを使用して、外部データベースに基づいてデータソースからの選択クエリを定義できます。
- スクリプト : 特定のディクショナリ方式にデータを提供する PowerShell スクリプトを記述できます。

データソースタイプの作成に関するガイドラインは次のとおりです。

- スクリプトの作成には PowerShell の言語を使用する必要があります。
- PowerShell スクリプトは、現在の作業ディレクトリ内の各ディクショナリエントリの出力を提供する必要があります。
- データ・ファイルには 'dictionary_entry_csv' という名前を付ける必要がありますこの場合 'ディクショナリ・エントリの名前は小文字にする必要があります

Performance Advisor から情報を収集する事前定義されたデータソースのタイプでは、スクリプトベースのデータソースのタイプを使用します。出力ファイルの名前は 'array_performe.csv' および 'aggregate_performe.csv' です

- 「.csv」ファイルには、ディクショナリエントリ属性と同じ順序でコンテンツが含まれている必要があります。

ディクショナリエントリには、array_ip、date、day、hour、cpu_busy、total_ops_per_sec、disk_per_sec

PowerShell スクリプトは '.csv' ファイルに同じ順序でデータを追加します

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

- スクリプトからのデータ出力が WFA キャッシュに正確にロードされるようにするには、エンコーディングを使用します。
- “.csv”ファイルにヌル値を入力するときは、“N”を使用する必要があります。

予約語

OnCommand Workflow Automation (WFA) には予約語が含まれています。変数名、ユーザ入力、定数、戻りパラメータなどの属性やパラメータには、ワークフローで予約語を使用しないでください。

WFA での予約語のリストを次に示します。

• および	• 浮動小数点	• Proto
• 配列	• 浮動小数点	• 戻ります
• アサート	• の場合	• 実行時
• ブール値	• foreach	• SecurityManager
• ブール値	• 機能	• 短い
• バイト	• 状況	• 短い
• バイト	• インポート	• サウンド好き
• 特性	• IMPORT STATIC	• StrictMath
• を押します	• インチ	• 文字列
• CharSequence (シャルシーケンス	• instanceof	• StringBuffer
• クラス	• 整数	• StringBuilder
• クラスローダー	• 整数	• strsim
• コンパイラ	• はです	• スイッチ
• が含まれます	• isdef	• システム
• convertable_to	• 長	• ねじ切り (Thread)
• DEF	• 長	• ThreadLocal を選択します
• する	• 数学	• 正しいです
• ダブル	• 新規	• まで
• ダブル	• null	• VAR
• それ以外	• 番号	• 無効です
• 空です	• オブジェクト	• 間
• いいえ	• または	• を使用

REST API の使用方法

Workflow Automation (WFA) の REST API を使用して、外部ポータルやデータセンター、オーケストレーションソフトウェアからワークフローを呼び出すことができます。WFA では、すべての REST API について XML および JSON コンテンツタイプがサポートされます。

WFA を使用すると、外部サービスからワークフロー、ユーザ、フィルタ、ファインダなどのさまざまなリソース収集にアクセスできます。URI パスを使用する。外部サービスでは、GET、PUT、POST、DELETE などの HTTP メソッドを使用できます。これらの URI を使用してリソースに対して CRUD 操作を実行します。

WFA REST API を使用して、次のような操作を実行できます。

- ワークフローの定義とメタデータにアクセスします。

- ・ワークフローを実行し、その実行を監視します。
- ・ユーザとロールを表示し、パスワードを変更する。
- ・リソース選択フィルタの実行とテスト
- ・リソースファインダの実行とテスト
- ・ストレージやその他のデータセンター オブジェクトのクレデンシャルを管理する。
- ・データソースとデータソースのタイプを表示します。

REST API の詳細については、_REST ドキュメントを参照してください。

https://wfa_server_ip:port/rest/docs

「wfa_server_ip」は WFA サーバの IP アドレスで、インストール時に WFA サーバに使用した TCP ポート番号です。



WFA は、Web UI からの要求について、クロスサイトリクエストフォージェリ (CSRF) トークンをチェックします。ただし、WFA では、REST クライアントまたはオーケストレーション ソフトウェアからの REST 要求の受信については、CSRF トークンをチェックしません。

学習資料への参照

高度な Workflow Automation (WFA) ワークフローを作成するためには、スクリプト作成とプログラミングに関するいくつかの手順を理解しておく必要があります。WFA のビルディングブロックまたはワークフローを作成する前に、参考資料を使用して必要なオプションを確認できます。

Windows PowerShell の場合

WFA では、ワークフローの処理に PowerShell スクリプトを使用します。次の表に、PowerShell の学習資料への参照資料を示します。

Windows PowerShell を使用する前に	http://msdn.microsoft.com/en-us/library/windows/desktop/aa973757(v=vs.85).aspx
PowerShell 開発—統合スクリプト環境 (ISE)	https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/ise/introducing-the-windows-powershell-ise?view=powershell-7.2
_.NET フレームワークの命名ガイドライン _	http://msdn.microsoft.com/en-us/library/xzf533w0%28v=vs.71%29.aspx
PowerShell コード形式	http://get-powershell.com/post/2011/04/13/Extra-Points-for-Style-when-writing-PowerShell-Code.aspx
PowerShell の try/catch が最後に行われました	http://technet.microsoft.com/en-us/library/dd315350.aspx

PowerShell の自動変数	http://technet.microsoft.com/en-us/library/dd347675.aspx
PowerShell エラーレポート機能	https://docs.microsoft.com/en-us/powershell/scripting/developer/cmdlet/error-reporting-concepts?view=powershell-7.2
PowerShell の共通パラメータ	https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_commonparameters?view=powershell-7.2

Data ONTAP PowerShell ツールキット

Data ONTAP PowerShell ツールキットには WFA がバンドルされています。PowerShell Toolkit のコマンドレットを使用して、PowerShell スクリプトから Data ONTAP コマンドを呼び出すことができます。詳細については、「`wfa_install_location \wfa\posh\Modules\DataONTAP\webhelp\index.html`」の形式でアクセスできる Data ONTAP PowerShell Toolkit Help_ を参照してください

「`_wfa_install_location _`」は WFA のインストールディレクトリ、「`C:\Program Files\NetApp`」はデフォルトのインストールディレクトリです。

次の表に、Data ONTAP PowerShell ツールキットに関する情報の参照先を示します。

ONTAP PowerShell Toolkit の記事	https://community.netapp.com/t5/Tech-OnTap-Articles/The-Data-ONTAP-PowerShell-Toolkit/ta-p/85933
ONTAP PowerShell Toolkit ネットアップコミュニティ	https://community.netapp.com/t5/forums/filteredbylabelpage/board-id/microsoft-cloud-and-virtualization-discussions/label-name/powershell%20toolkit

Perl の場合

WFA では、ワークフロー処理用の Perl コマンドがサポートされます。WFA をインストールすると、必要な Perl モジュールと Perl モジュールが WFA サーバにインストールされます。

"ActivePerl ユーザガイド"

`_ActivePerl User Guide_` には、「`wfa_install_location \WFA\Perl64\HTML\index.html`」からもアクセスできます

「`_wfa_install_location _`」は WFA のインストールディレクトリ、「`C:\Program Files\NetApp`」はデフォルトのインストールディレクトリです WFA では「ワークフローの操作に Perl スクリプトを使用します」次の表に、Perl の学習資料の参照先を示します。

最新の Perl : 2014	http://modernperlbooks.com/books/modern_perl_2014/index.html
-----------------	---

Perl プログラミングのドキュメント	http://perldoc.perl.org/
Perl プログラミング言語	http://www.perl.org/

NetApp Manageability SDK の使用

NetApp Manageability SDK に必要な Perl モジュールは、WFA にバンドルされています。これらの Perl モジュールは、WFA で Perl コマンドを使用するため必要です。詳細については、NetApp Manageability SDK のドキュメントを参照してください。このドキュメントには、「`wfa_install_location \wfa\perl\NMSDK\html.`」からアクセスできます

「`wfa_install_location`」は WFA のインストールディレクトリ、「`C:\Program Files\NetApp`」はデフォルトのインストールディレクトリです。

Structured Query Language (SQL ; 構造化クエリ言語)

SQL SELECT 構文は、フィルタおよびユーザー入力の入力に使用されます。

["MySQL Select の構文"](#)

MVFLEX 表現言語 (MVEL)

WFA ワークフローの MVEL 式の構文を関数や変数などで使用できます。

詳細については、[_ MVEL 言語ガイド _](#) を参照してください。

正規表現

WFA では正規表現 (regex) を使用できます。

["ActionScript 3.0 では正規表現を使用します"](#)

OnCommand Workflow Automation の関連ドキュメント

ここでは、OnCommand Workflow Automation (WFA) サーバをより高度に設定する方法を学ぶのに役立つ、その他のドキュメントとツールを示します。

その他の参照

ネットアップコミュニティの Workflow Automation のスペースでは、次のような追加のラーニングリソースを提供しています。

• * ネットアップコミュニティ *

["ネットアップコミュニティ：Workflow Automation \(WFA \) "](#)

ツール参照

- * Interoperability Matrix *

に、ハードウェアコンポーネントとソフトウェアバージョンのサポートされる組み合わせを示します。

["互換性マトリックス"](#)

著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を隨時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5225.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。