



OnCommand Workflow Automation

でのビルディングブロックの操作

OnCommand Workflow Automation

NetApp
October 09, 2025

目次

OnCommand Workflow Automation でのビルディングブロックの操作	1
データソースとは	1
ディクショナリエントリとは	2
コマンドの仕組み	2
フィルタとは	4
ファインダとは	4
機能とは	4
方式は何ですか	4
リモートシステムの種類	5
テンプレートの使用方法	6
カテゴリの使用方法	6
エンティティのバージョン管理の仕組み	6
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	8
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	9

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、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ュします。WFAは、データソースからリソース情報を収集し、キャッシュ方式に対応するようにフォーマットします。キャッシュ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が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（MVFL）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、名前付き関数内の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トリ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。

スキーマには、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 桁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	• データソースのタイプ
テンプレート	• ワークフロー
機能	• ワークフロー • テンプレート  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。
辞書	• 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	• ワークフロー
フィルタ	• 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。

親のバージョンのレ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

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随時、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 5252.227-19（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权を有し、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权については、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。