

# VMware vRealize Orchestrator クライアントの 使用

vRealize Orchestrator 7.6



vmware®

最新の技術ドキュメントは、VMware の Web サイト (<https://docs.vmware.com/jp/>) でご確認ください。このドキュメントに関するご意見および感想は、[docfeedback@vmware.com](mailto:docfeedback@vmware.com) までお送りください。

VMware, Inc.  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

ヴィエムウェア株式会社  
105-0013 東京都港区浜松町 1-30-5  
浜松町スクエア 13F  
[www.vmware.com/jp](http://www.vmware.com/jp)

Copyright © 2008-2019 VMware, Inc. All rights reserved. [著作権および商標情報](#)。

# 目次

1	VMware vRealize Orchestrator クライアントの使用	4
2	VMware vRealize Orchestrator クライアント	5
	vRealize Orchestrator クライアントの API Explorer の使用	6
	vRealize Orchestrator オブジェクトのバージョン履歴	6
	vRealize Orchestrator クライアントを使用したワークフローの以前の状態への復帰	7
	vRealize Orchestrator クライアントの使用状況ダッシュボード	7
3	vRealize Orchestrator クライアントのロールおよびグループの管理	8
	vRealize Orchestrator Client でのロールの割り当て	10
	vRealize Orchestrator Client でのグループの作成	10
4	Orchestrator オブジェクトの管理	12
	vRealize Orchestrator クライアントでのワークフローの管理	12
	vRealize Orchestrator ワークフロー ライブラリ内の標準ワークフロー	12
	vRealize Orchestrator クライアントでのワークフローの作成	13
	vRealize Orchestrator 入力フォーム デザイナ	13
	vRealize Orchestrator クライアントでのユーザー操作の要求	17
	vRealize Orchestrator クライアントでのワークフローのスケジュール設定	17
	vRealize Orchestrator クライアントでのアクションの作成	18
	vRealize Orchestrator クライアントの構成要素	19
	vRealize Orchestrator クライアントでのポリシーの管理	20
	vRealize Orchestrator クライアントのリソース要素	22
	vRealize Orchestrator 監査ログの使用	22
5	vRealize Orchestrator パッケージの使用	24
	vRealize Orchestrator クライアントでのパッケージの作成	24
	vRealize Orchestrator クライアントでのパッケージのエクスポート	25
	vRealize Orchestrator クライアントでのパッケージのインポート	26
6	vRealize Orchestrator クライアントのトラブルシューティング	28
	vRealize Orchestrator クライアントのメトリック データ	28
	vRealize Orchestrator クライアントでのワークフローのプロファイル	28
	vRealize Orchestrator システム ダッシュボードの使用	29
	vRealize Orchestrator クライアントでのワークフロー トークンの再生の使用	30
	vRealize Orchestrator ワークフローの検証	30
	vRealize Orchestrator クライアントでのワークフローの検証と検証エラーの修正	31
	vRealize Orchestrator クライアントでのワークフロー スクリプトのデバッグ	31

# VMware vRealize Orchestrator クライアントの使用

1

VMware vRealize Orchestrator クライアントを使用すると、ワークフローの自動化機能および新しい HTML5 ベースの vRealize Orchestrator Client の機能に関する情報を取得できます。

## 対象者

この情報は、vRealize Orchestrator ワークフローの実行と管理に役立つツールをお探しの経験豊富なシステム管理者を対象としています。

---

**注:** Java ベースの Orchestrator レガシー クライアントは廃止され、今後の vRealize Orchestrator のリリースでは削除される予定です。Orchestrator レガシー クライアントの使用については、『VMware vRealize Orchestrator レガシー クライアントの使用』を参照してください。

---

# VMware vRealize Orchestrator クライアント

# 2

新しい HTML5 ベースの vRealize Orchestrator Client を使用して、vRealize Orchestrator サービスおよびオブジェクトを管理します。

vRealize Orchestrator オブジェクトは vRealize Orchestrator Client を使用して作成および管理できます。vRealize Orchestrator Client は `https://your_orchestrator_server_ip_or_dns_address:8283/vco-controlcenter/client` に配置されています。

REST API 通信	vRealize Orchestrator Client は、vRealize Orchestrator コントロール センター サーバで実行されます。クライアントは、REST プロキシを介して vRealize Orchestrator REST API と通信します。
ワークフロー管理	vRealize Orchestrator Client でワークフローを作成、編集、スケジュール設定、実行、および削除します。
アクション管理	vRealize Orchestrator Client のアクションを作成、編集、および削除します。アクション エディタは、vRealize Orchestrator API Explorer に含まれる一般的なスクリプト要素のオート コンプリートをサポートします。
ポリシー管理	vRealize Orchestrator Client でポリシーを作成、編集、実行、および削除します。
構成管理	vRealize Orchestrator Client で構成要素を作成、実行、および削除します。
リソース管理	vRealize Orchestrator Client でリソース要素をエクスポート、インポート、および更新します。
メトリック データ	vRealize Orchestrator Client システム ダッシュボードおよびプロファイル機能を使用して、vRealize Orchestrator 環境およびワークフローに関する有用なメトリック データを収集します。
パッケージの管理	vRealize Orchestrator Client を介して vRealize Orchestrator オブジェクトを含むパッケージを作成、削除、エクスポート、およびインポートします。
権限管理	管理者権限を持つユーザーは、vRealize Orchestrator Client のユーザーにロールを割り当て、グループに追加できます。
API Explorer	vRealize Orchestrator Client で使用できる API コマンドを検索できます。

この章には、次のトピックが含まれています。

- [vRealize Orchestrator クライアントの API Explorer の使用](#)

- [vRealize Orchestrator オブジェクトのバージョン履歴](#)
- [vRealize Orchestrator クライアントの使用状況ダッシュボード](#)

## vRealize Orchestrator クライアントの API Explorer の使用

vRealize Orchestrator API Explorer を利用すると、スクリプト化されたワークフロー要素で利用できる JavaScript オブジェクトのドキュメントを参照できます。

API Explorer は、vRealize Orchestrator 対応の REST オブジェクトの一覧を備えており、特定のオブジェクトに関連付けられている属性、コンストラクタ、および HTTP メソッドに関する情報を取得できます。API Explorer には、vRealize Orchestrator Client ダッシュボードの右上のメニューと、vRealize Orchestrator ワークフローおよびアクション エディタの [スクリプト] タブからアクセスできます。

## vRealize Orchestrator オブジェクトのバージョン履歴

vRealize Orchestrator Client は、各 vRealize Orchestrator オブジェクトのバージョン履歴レコードを保持します。バージョン履歴を使用すると、さまざまな vRealize Orchestrator オブジェクトのバージョンを比較して、以前のバージョンに戻すことができます。

vRealize Orchestrator は、オブジェクトを保存するときに、vRealize Orchestrator オブジェクトごとにバージョン履歴レコードを作成します。その後、vRealize Orchestrator オブジェクトを変更すると、新しいバージョン履歴レコードが作成されます。以前のバージョンの履歴レコードは保持され、オブジェクトに対する変更を追跡してオブジェクトを以前のバージョンに戻す際に使用できます。オブジェクトを以前のバージョンに戻すと、新しいバージョン履歴レコードが作成されます。

vRealize Orchestrator Client は、次の vRealize Orchestrator オブジェクトのバージョン履歴を追跡します。

- ワークフロー
- アクション
- パッケージ
- ポリシー
- 構成要素

オブジェクトのバージョン履歴には、オブジェクト エディタ ページの [バージョン履歴] タブからアクセスできます。別のユーザーと同時にオブジェクトを編集すると、マージの競合が発生する可能性があります。マージの競合を解決するには、エラー メッセージの右にある [解決] をクリックします。[競合の解決] ウィンドウには、次の 3 つのオプションがあります。

- [相手の変更を使用]：他のユーザーが行った変更を使用してマージの競合を解決します。
- [自分の変更を使用]：自分が行った変更を使用してマージの競合を解決します。
- [解決]：表示された変更モデルを編集してマージの競合を解決します。指定されたモデルが無効の場合、このオプションは使用できません。

## vRealize Orchestrator クライアントを使用したワークフローの以前の状態への復帰

バージョン履歴を使用すると、ワークフローを以前に保存したときの状態に戻すことができます。ワークフローの状態は、ワークフロー バージョンより前のバージョンに戻すことも、より新しいバージョンに戻すこともできます。現在の状態のワークフローと保存されているバージョンのワークフローの違いを比較することもできます。

### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 ワークフローの編集集中に、[バージョン履歴] をクリックします。
- 3 ワークフロー バージョンを選択し、[差分の対象] ドロップダウン メニューからバージョンを選択して、差異を比較します。

現在のワークフロー バージョンと選択したワークフロー バージョンの違いがウィンドウに表示されます。

- 4 ワークフロー バージョンの [元に戻す] をクリックして、ワークフローの状態をリストアします。

ワークフローの状態は選択したバージョンの状態に戻ります。

## vRealize Orchestrator クライアントの使用状況ダッシュボード

vRealize Orchestrator Client ダッシュボードは、vRealize Orchestrator Client ワークフローの監視、管理、およびトラブルシューティングに役立つツールを提供します。

vRealize Orchestrator Client ダッシュボードの情報は、5 つのパネルに表示されます。

ウィンドウ	説明
ワークフローの実行	実行中、待機中、および失敗したワークフローの実行の数に関する視覚的なデータを提供します。
お気に入りワークフロー	お気に入りに追加されたワークフローを表示します。
入力を待機中	追加のユーザー操作が必要な保留中のワークフローの実行が表示されます。これらのワークフローは、ユーザー インターフェイスの右上隅の通知メニューにも表示されます。
最近のワークフローの実行	最近のワークフローの実行を管理します。ワークフローの実行の名前、状態、開始日、および終了日を示します。
注意	失敗したワークフローの実行と、ワークフロー実行のパフォーマンス メトリックを表示します。

# vRealize Orchestrator クライアントの ロールおよびグループの管理

## 3

管理者は、vRealize Orchestrator Client を使用して、vRealize Orchestrator の機能およびコンテンツに対するユーザー ロールとグループ権限を設定できます。

vRealize Orchestrator インスタンスが認証された後、管理者は機能とコンテンツへのアクセスを制御する権限を設定できます。vRealize Orchestrator Client の権限は、ロール管理とグループ権限に分かれています。ロール管理を使用して、ユーザーが表示および使用できる vRealize Orchestrator Client 機能を制御します。グループ権限を使用して、ユーザーが表示および使用できる vRealize Orchestrator Client コンテンツを制御します。グループ権限が適用されるコンテンツ アクセスには、ワークフロー、アクション、ポリシー、構成要素、およびリソース要素などがあります。グループを使用することで、ユーザーを共通のプロジェクトに編成できます。たとえば、カスタムの vRealize Orchestrator プラグインの開発を行っているユーザーを含むグループを作成できます。

---

**注:** 標準のワークフローやアクションなど、事前構成された vRealize Orchestrator コンテンツへのアクセスは、グループ権限を使用して構成されていない限り、すべてのユーザー間で共有されます。

---

### ロール

ロール管理は、vRealize Automation で認証された vRealize Orchestrator インスタンスでのみ使用できます。vSphere で認証された vRealize Orchestrator インスタンスの権限管理は、グループ権限に限定されます。



ロール	説明
[管理者]	<p>特定のグループによって作成されたコンテンツを含む、すべての vRealize Orchestrator Client 機能およびコンテンツにアクセスできます。ユーザー ロールの設定、グループの作成と削除、およびグループへのユーザーの追加を担当します。</p> <p><b>注:</b> vRealize Orchestrator の認証に使用される vRealize Automation 環境のテナント管理者には、デフォルトで [管理者] 権限が付与されます。</p>
[ワークフロー デザイナー]	<p>自分の vRealize Orchestrator Client コンテンツを作成、実行、編集、および削除できます。自分に割り当てられたグループに自分のコンテンツを追加できます。vRealize Orchestrator Client の管理およびトラブルシューティング機能にはアクセスできません。</p>

**注:** 事前定義済みロールを持っていない vRealize Automation ユーザーは、vRealize Orchestrator Client にログインできますが、クライアント機能へのアクセスは制限されます。グループに属している場合は、そのグループに関連付けられているコンテンツを表示および実行できます。

## グループ

vRealize Orchestrator Client のグループ権限は、Active Directory ユーザー グループに関連付けられていません。Active Directory ユーザー グループの詳細については、『VMware vRealize Orchestrator プラグインの使用』ドキュメントの「ユーザー グループ ワークフロー」を参照してください。

グループ ユーザー権限	説明
[実行および編集]	<p>vRealize Automation で認証された vRealize Orchestrator インスタンスでのみ使用できます。グループで使用する vRealize Orchestrator オブジェクトを作成、編集、追加、および実行できます。</p>
[実行]	<p>グループに含まれる vRealize Orchestrator オブジェクトを表示および実行できます。</p>

**注:** グループ権限は、vRealize Orchestrator Client のロール管理システムに関連付けられています。たとえば、事前定義されたロールを持っていないユーザーは、[実行および編集] 権限を持つことはできますが、自分のコンテンツまたはグループコンテンツの表示と実行のみを行うことができ、コンテンツの作成、編集、および追加を行うことはできません。

この章には、次のトピックが含まれています。

- [vRealize Orchestrator Client でのロールの割り当て](#)
- [vRealize Orchestrator Client でのグループの作成](#)

## vRealize Orchestrator Client でのロールの割り当て

管理者は、ユーザーを vRealize Orchestrator Client に追加して、ユーザーが表示および使用できる機能を設定できます。

ロール管理では、vRealize Orchestrator Client の機能へのユーザーのアクセスを vRealize Orchestrator の ID プロバイダから制御します。ロール管理は、vRealize Orchestrator Client ユーザー インターフェイスと API 機能の両方に適用されます。

---

**注:** ロール管理は、vRealize Automation で認証された vRealize Orchestrator インスタンスでのみ使用できます。vSphere で認証された vRealize Orchestrator インスタンスの権限管理は、グループ権限に限定されます。グループ権限の設定の詳細については、[vRealize Orchestrator Client でのグループの作成](#)を参照してください。

---

### 手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [管理] - [ロール管理] の順に移動します。
- 3 [追加] をクリックします。
- 4 vRealize Orchestrator Client に追加するユーザーまたはグループを検索します。
- 5 ユーザーのロールを選択します。ロールの詳細については、[3 章 vRealize Orchestrator クライアントのロールおよびグループの管理](#)を参照してください。
- 6 [保存] をクリックします。

## vRealize Orchestrator Client でのグループの作成

管理者は、グループを使用して、vRealize Orchestrator コンテンツ ユーザーが vRealize Orchestrator Client で表示およびアクセスできる内容を設定できます。

vRealize Orchestrator Client を使用して、vRealize Orchestrator ワークフロー、アクション、ポリシー、構成要素、リソース要素に対するグループ権限を設定できます。

---

**注:** vSphere で認証された vRealize Orchestrator インスタンスのユーザーの場合、[実行] グループ権限のみを設定できます。

---

### 手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [管理] - [グループ] に移動します。
- 3 [新規グループ] をクリックします。
- 4 [サマリ] タブでグループの名前と説明を追加します。

- 5 [ユーザー] タブで [追加] をクリックします。
  - a グループに追加するユーザーを検索します。
  - b ユーザーにグループ権限を割り当てます。
  - c [追加] をクリックします。
- 6 [項目] タブで、vRealize Orchestrator オブジェクトをグループに追加します。

---

**注:** vRealize Orchestrator Client でオブジェクトを作成する際に、そのオブジェクトを既存のグループに追加することもできます。オブジェクトを追加するには、オブジェクト エディタの [サマリ/全般] タブの [アクセス許可] ドロップダウン メニューからグループを選択します。

---

- 7 [保存] をクリックします。

# Orchestrator オブジェクトの管理

vRealize Orchestrator Client を使用して、ワークフロー、アクション、ポリシー、構成要素、リソース要素などの vRealize Orchestrator オブジェクトの作成、編集、実行、削除、およびトラブルシューティングを行うことができます。

この章には、次のトピックが含まれています。

- [vRealize Orchestrator クライアントでのワークフローの管理](#)

## vRealize Orchestrator クライアントでのワークフローの管理

ワークフローとは、順次実行する一連のアクションおよび決定のことです。vRealize Orchestrator は、一般的な管理タスクを実行するワークフローのライブラリを提供します。また vRealize Orchestrator には、ワークフローが実行する個々のアクションのライブラリも用意されています。

ワークフローは、特定の順で実行されたときに、仮想環境において特定のタスクまたは特定のプロセスを完了するアクション、決定、および結果を組み合わせます。ワークフローでは、仮想マシンのプロビジョニング、バックアップ、通常のメンテナンスの実行、メール送信、SSH 操作、物理インフラストラクチャの管理、および他の汎用ユーティリティ操作などのタスクを実行します。ワークフローは、入力を機能に応じて受け付けます。定義されたスケジュールに従って実行するワークフローや、特定の予測イベントが発生した場合に実行するワークフローを作成できます。情報は、ユーザー、他のユーザー、別のワークフローまたはアクションによって提供するか、アプリケーションからの Web サービス呼び出しなどの外部プロセスによって提供できます。ワークフローでは、実行する前に情報の検証およびフィルタリングを行います。

ワークフローは他のワークフローを呼び出すことができます。たとえば、いくつかの異なるワークフローにおいて、仮想マシンを開始するワークフローを再利用できます。

ワークフローは、vRealize Orchestrator Client インターフェイスの統合開発環境 (IDE) を使用して作成できます。この IDE では、ワークフロー ライブラリにアクセスでき、ワークフロー エンジンでワークフローを実行できる機能が用意されています。このワークフロー エンジンでは、vRealize Orchestrator にプラグインした外部ライブラリからのオブジェクトを使用することもできます。この機能を使用すると、プロセスのカスタマイズやサードパーティ アプリケーション提供の機能を実装できます。

## vRealize Orchestrator ワークフロー ライブラリ内の標準ワークフロー

vRealize Orchestrator には、ワークフローの標準ライブラリが付属しています。このライブラリを使用して、仮想インフラストラクチャでの操作を自動化することができます。標準ライブラリ内のワークフローは、読み取り専用の状態でロックされています。標準ワークフローをカスタマイズするには、そのワークフローを複製する必要があります。作成した複製ワークフローまたはカスタム ワークフローは完全に編集可能です。

ワークフロー ライブラリのコンテンツには、HTML5 ベースの vRealize Orchestrator Client の [ライブラリ] - [ワークフロー] メニューからアクセスできます。クライアントの標準およびカスタム ワークフローは、両方ともタグを使用して設定されます。たとえば、ワークフロー ライブラリの検索ボックスに **SSH** と入力して、[キー ペアの作成] ワークフローにアクセスすることができます。

---

**注:** ワークフローを複製しない限り、新しいタグを標準ワークフローに追加することはできません。

---

## vRealize Orchestrator クライアントでのワークフローの作成

vRealize Orchestrator Client を使用して、ワークフローを作成および編集できます。

vRealize Orchestrator Client は新しいプレゼンテーション エンジンを使用します。その結果、vRealize Orchestrator Client で作成されたワークフローは、vRealize Orchestrator Client でのみ実行および編集できます。Orchestrator レガシー クライアントで作成されたワークフローは vRealize Orchestrator Client で実行できますが、vRealize Orchestrator Client で編集することはできません。

### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動します。
- 3 [新しいワークフロー] をクリックします。
- 4 新しいワークフローの名前を入力し、[作成] をクリックします。
- 5 ワークフロー エディタを使用して、ワークフローの変数、入力と出力、スキーマ構造、およびプレゼンテーションを構成できます。
- 6 ワークフローの編集を終了するには、[保存] をクリックします。

---

**注:** [バージョン履歴] タブでワークフローへの変更を追跡できます。詳細については、『[vRealize Orchestrator オブジェクトのバージョン履歴](#)』を参照してください。

---

### 次のステップ

vRealize Orchestrator トークンの再生機能を使用して、ワークフローのパフォーマンスを最適化することができます。詳細については、『[vRealize Orchestrator クライアントでのワークフロー トークンの再生の使用](#)』を参照してください。

## vRealize Orchestrator 入力フォーム デザイナ

ワークフローで入力パラメータが必要な場合、ダイアログ ボックスが開かれ、ユーザーはここに必要な値を入力します。このダイアログ ボックスのコンテンツ、レイアウト、およびプレゼンテーションは、入力フォーム デザイナで調整できます。

入力フォーム デザイナは、ワークフロー エディタの [入力フォーム] タブにあります。このデザイナーは、ナビゲーション メニュー、デザイン キャンバス、およびプロパティ メニューで構成されています。左側のメニューからデザイン キャンバスに、入力と汎用要素をドラッグできます。キャンバスでは、入力パラメータの位置を設定し、個別の入力タブに編成して、入力パラメータのプロパティを構成できます。vRealize Orchestrator Client の入力フォーム デザイナは、Orchestrator レガシー クライアントの [プレゼンテーション] タブに相当します。

**注:** 入力フォーム デザイナのワークフロー エディタの [変数] タブのコンテンツを使用することはできません。[入力/出力] タブのパラメータのみを使用できます。

#### 汎用要素

ドロップダウン メニューやパスワード テキスト ボックスなどの汎用要素を入力フォーム デザイナに追加できます。汎用要素は実際の入力パラメータに対応していませんが、入力パラメータにバインドできます。

## vRealize Orchestrator クライアントでのワークフロー入力パラメータ ダイアログ ボックスの作成

入力フォーム デザイナを使用して、[ワークフロー入力パラメータ] ダイアログ ボックスを作成およびカスタマイズできます。

#### 前提条件

入力パラメータの定義済みリストがワークフローに存在することを確認します。

#### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動します。
- 3 カスタム ワークフローを選択します。
- 4 [入力フォーム] タブをクリックします。
- 5 (オプション) 入力ダイアログ ボックスで使用するタブを作成します。  
タブを使用して、ダイアログ ボックスの構造を整理できます。
- 6 入力パラメータを選択します。
- 7 入力パラメータのプロパティを編集します。  
入力パラメータのプロパティの詳細については、[vRealize Orchestrator クライアントの入力パラメータのプロパティ](#)を参照してください。
- 8 (オプション) キャンバスに汎用要素を追加し、入力パラメータにバインドします。
- 9 (オプション) 入力パラメータに外部検証を追加します。詳細については、『[アクションを使用した vRealize Orchestrator ワークフローの入力の検証](#)』を参照してください。
- 10 [保存] をクリックします。

ワークフロー ダイアログ ボックスのレイアウトを作成し、入力パラメータのプロパティを設定しました。

## vRealize Orchestrator クライアントの入力パラメータのプロパティ

パラメータのプロパティを設定すれば、ユーザーが vRealize Orchestrator ワークフローを実行する際に指定する入力パラメータに制約を加えることができます。

vRealize Orchestrator を使用すると、ワークフローで使用される入力パラメータの値を定量化するために使用するパラメータのプロパティを定義できます。定義したパラメータのプロパティは、ユーザーが vRealize Orchestrator ワークフローで指定できる入力パラメータのタイプや値に制限を課します。

パラメータのプロパティは入力パラメータを検証し、入力パラメータ ダイアログ ボックスに表示されるテキスト ボックスの表示を変更します。パラメータのプロパティには、パラメータ間の依存関係を作成できるものもあります。

パラメータのプロパティ	説明
[ラベル]	入力パラメータのラベルを設定します。
[表示タイプ]	入力テキスト ボックスの表示タイプを設定します。
[可視性]	入力パラメータの可視性を設定します。
[読み取り専用]	入力テキスト ボックスを読み取り専用として設定します。
[カスタム ヘルプ]	入力パラメータ Signpost の説明を設定します。
[デフォルト値]	入力パラメータのデフォルト値を設定します。
[ステップ]	数値タイプの入力に使用されます。1 回のクリックあたりの入力パラメータの値の増加量を設定します。
[必須項目]	入力パラメータの値が必須かどうかを設定します。
[正規表現]	正規表現を使用して入力を検証します。
[最小値]	パラメータの最小値または長さを設定します。
[最大値]	パラメータの最大値または長さを設定します。
[テキスト ボックスの一致]	入力パラメータの値を、別の入力パラメータの値と一致するように設定します。
[値のソース]	<p>[表示]、[値]、[制約] タブでパラメータ プロパティの値のソースを設定します。</p> <p><b>注:</b> [外部ソース] を使用して、外部アクションの値をインポートできます。使用可能なアクションのフィルタリングは、パラメータ タイプ別に実行されます。</p>

## アクションを使用した vRealize Orchestrator ワークフローの入力の検証

外部アクションを使用して、カスタム ワークフローの入力を検証します。

### 前提条件

入力パラメータを使用してカスタム ワークフローを作成します。詳細については、次を参照してください。

[vRealize Orchestrator クライアントでのワークフローの作成](#)。入力フォーム デザインを使用して、ワークフロー入力の外部検証を作成できます。外部検証では、入力パラメータ値にエラーが含まれている場合に文字列値を返すアクション スクリプトが使用されます。入力パラメータ値が有効の場合、外部検証は何も返しません。

## 手順

1 vRealize Orchestrator Client にログインします。

2 検証アクションを作成します。

- a [ライブラリ] - [アクション] の順に移動します。
- b [新規アクション] をクリックします。
- c [サマリ] タブに必要な情報を入力します。
- d 検証アクションの入力パラメータを入力します。

---

**注:** 検証アクションの入力パラメータの名前は、検証されるワークフローの入力パラメータの名前と同じである必要があります。

---

e [スクリプト] タブに検証アクションのスクリプトを入力します。

```
if (in_1=="invalid") {
    return "in_1 can't be invalid!";
}

if (in_2=="invalid") {
    return "in_2 can't be invalid!";
}

//inputs are valid, return nothing
```

---

**注:** 前のスクリプトは単純な例であり、使用可能な検証スクリプトのすべての範囲を表しているわけではありません。

---

f [保存] をクリックします。

3 外部検証を適用します。

- a [ライブラリ] - [ワークフロー] の順に移動します。
- b カスタム ワークフローを選択します。
- c [入力フォーム] タブを選択します。
- d 画面の左上にあるクリップボード アイコンを選択します。
- e vRealize Orchestrator 検証要素をキャンバスにドラッグします。
- f 検証要素を選択し、検証ラベルを入力して、検証アクションを選択します。
- g (オプション) 追加の検証要素を作成します。
- h [保存] をクリックします。

4 ワークフローを実行します。

検証でエラーが発生した場合は、文字列を返します。検証に成功すると、検証は何も返さず、ワークフローの実行が続行されます。

これで、カスタム vRealize Orchestrator ワークフローの外部検証が作成されました。



## vRealize Orchestrator クライアントでのユーザー操作の要求

ワークフローが完了する前に、追加のユーザー入力を要求できます。

追加のユーザー操作を必要とするワークフローは、ユーザーから必要な入力パラメータを受け取るまで処理を一時停止します。ワークフローは必要な情報を入力できるユーザーを定義し、その定義に応じて操作の要求を送信します。ユーザー入力を待機しているワークフローは、vRealize Orchestrator Client ダッシュボードの [最近のワークフローの実行] パネルおよび右上の通知メニューに表示されます。

## vRealize Orchestrator クライアントでのワークフローのスケジュール設定

スケジュール設定を利用すると vRealize Orchestrator ワークフローの実行を自動化できます。

ワークフローの実行をスケジューリングする際は、スケジュール設定タスクを実行する日付、時刻、および実行間隔を設定します。

### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [ライブラリ] メニューからワークフローを選択し、ワークフロー パネルで [スケジュール] をクリックします。
- 3 [全般]、[スケジュール設定]、および [ワークフロー] カテゴリでスケジュール設定されたタスクのパラメータを構成します。

**注:** [ワークフロー] パラメータ カテゴリは、入力パラメータが必要なワークフローにのみ表示されます。

パラメータ	説明
[名前]	スケジュール設定タスクの名前。
[説明]	スケジュール設定タスクの目的についての簡単な説明。
[開始]	ワークフローの実行がスケジュール設定されている最初の日付および時間。
[過去の場合は開始する]	スケジュール設定されている時間を過ぎてしまっている場合に、ワークフローを開始するかどうかを選択します。[はい] にすると、スケジュール設定されたワークフローはすぐに開始されます。[いいえ] にすると、繰り返しを設定されている次のスケジュールでワークフローが開始されます。
[スケジュール]	スケジュール設定タスクの繰り返しパターンとイベント トリガ エントリを設定します。
[終了日]	[繰り返しなし] が選択されている場合にのみ表示されます。スケジュール設定されたタスクが終了する日付および時間を設定します。
[ワークフロー]	ワークフローの入力パラメータを設定します。

- 4 [作成] をクリックします。

これでワークフローのスケジュール設定タスクが作成されました。[アクティビティ] - [スケジュール設定済み] に、スケジュール設定されたワークフローが表示されます。スケジュール設定タスクを削除するには、スケジュール パネルの [削除] をクリックします。

## vRealize Orchestrator クライアントでのスケジュール設定タスクの編集

スケジュール設定されたタスクを編集することで、スケジュール設定されたワークフローのパラメータ（日付、時刻、繰り返しなど）を変更できます。

### 前提条件

スケジュール設定されたワークフロー タスクを作成します。

### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [アクティビティ]-[スケジュール設定済み] からスケジュール設定タスクを選択します。
- 3 ワークフロー パネルで [編集] をクリックします。
- 4 スケジュールを編集し、[保存] をクリックします。

---

**注:** スケジュール設定タスクの作成時には、入力パラメータのセットは読み取り専用となっており、編集できません。これらのパラメータを変更するには、このワークフロー用に新しいスケジュール設定タスクを作成します。

---

## vRealize Orchestrator クライアントでのアクションの作成

vRealize Orchestrator Client を使用して、アクション スクリプトを作成、編集、および削除できます。

vRealize Orchestrator Client は、事前定義済みのアクションのライブラリとカスタム アクション スクリプトのアクション エディタを提供します。アクションは、ワークフローのビルディング ブロックとして使用する個々の機能を表します。

アクションは JavaScript 機能です。アクションでは複数の入力パラメータを使用でき、戻り値は 1 つになります。アクションでは vRealize Orchestrator API のすべてのオブジェクトのほか、プラグインを使用して vRealize Orchestrator にインポートしたすべての API のオブジェクトを呼び出すことができます。

ワークフローを実行すると、アクションはワークフローの変数から入力パラメータを取得します。これらの変数は、ワークフローの初期入力パラメータか、ワークフロー セット内の他の要素が実行時に設定した変数のいずれかになります。

アクション エディタには、スクリプトのオートコンプリート機能と、使用可能なスクリプト タイプとそのドキュメントを含む API エクスプローラが含まれています。

### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [ライブラリ]-[アクション] の順に移動します。
- 3 [新規アクション] をクリックします。
- 4 [全般] タブで、アクションの名前とモジュール名を入力します。

---

**注:** 名前とモジュール名は、アクションごとに一意である必要があります。アクション名は有効な JavaScript 関数である必要があります。アクション名は 1 つの単語とし、文字、数字、ドル記号 (\$) およびアンダースコア (\_) のみを使用できます。モジュール名は、ドット (.) で区切った単語で構成する必要があります。

---

- 5 (オプション) アクションの説明、バージョン番号、タグ、およびグループ権限を作成します。
- 6 [スクリプト] タブで、アクションの入力を追加し、出力の戻り値のタイプを選択して、スクリプトを書き込みます。
- 7 アクションの編集を終了するには、[保存] をクリックします。  
アクションが保存されたことがメッセージに示されます。

#### 次のステップ

これで、ワークフローで新しいカスタム アクションを使用できるようになりました。

## vRealize Orchestrator クライアントの構成要素

構成要素は、展開した vRealize Orchestrator サーバ全体で、定数の構成に使用できる変数のリストです。

構成要素を使用して、vRealize Orchestrator サーバで実行されているすべてのワークフロー、アクション、およびポリシーで変数を使用できるようにすることができます。

構成要素の変数を使用するワークフロー、アクション、またはポリシーを含むパッケージを作成すると、vRealize Orchestrator は、パッケージ内にその構成要素を自動的に含めます。構成要素を含むパッケージを別の vRealize Orchestrator サーバにインポートすると、構成要素の変数値もインポートすることができます。たとえば、実行元の vRealize Orchestrator サーバによって異なる変数値が必要なワークフローを作成する場合に、構成要素内でこれらの変数を設定すると、そのワークフローをエクスポートして、別の vRealize Orchestrator サーバで使用することができます。このため、構成要素を使用すると、サーバ間でワークフロー、アクション、ポリシーをより簡単に交換できます。

---

**注:** 5.1 以前の vRealize Orchestrator からエクスポートされた構成要素から、構成要素の変数値をインポートすることはできません。

---

## vRealize Orchestrator クライアントでの構成要素の作成

構成要素を使用すると、vRealize Orchestrator サーバ内で共通の変数を設定できます。サーバで実行中のすべての要素は、構成要素で設定した変数を使用することができます。

#### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [アセット] - [構成] の順に移動します。
- 3 [新しい構成] を選択します。
- 4 構成要素名を入力します。
- 5 [変数] タブを選択します。

6 ローカル変数を作成するには、[新規] をクリックします。

- a 変数名を入力します。
- b 変数タイプを選択します。

---

**注:** 構成変数の配列を作成するには、[配列] チェックボックスを選択します。

---

- c (オプション) 構成変数の値を入力します。
- d [保存] をクリックします。

7 構成要素の作成を完了するには、[保存] をクリックします。

#### 次のステップ

構成要素を使用して、ワークフロー、アクション、またはポリシーに変数を提供することができます。

## vRealize Orchestrator クライアントでのポリシーの管理

ポリシーは、システムのアクティビティを監視するイベントトリガです。ポリシーは、特定の vRealize Orchestrator オブジェクトのステータスまたはパフォーマンスの変更によって発行された、事前定義済みイベントに応答します。

ポリシーとは、特定の事前定義済みイベントが vRealize Orchestrator または vRealize Orchestrator がプラグインを介してアクセスするテクノロジーにおいて発生したときに、特定のワークフローまたはスクリプトを実行する一連のルール、ゲージ、しきい値、およびイベント フィルタです。vRealize Orchestrator は、ポリシーの実行中、常にポリシー ルールを評価します。たとえば、VC:HostSystem および VC:VirtualMachine タイプの vCenter Server オブジェクトの動作を監視するポリシー ゲージおよびしきい値を実装できます。

## vRealize Orchestrator クライアントでのポリシーの作成および適用

ポリシーを使用して、特定のイベントに対する vRealize Orchestrator システムの動作を監視することができます。

#### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [ライブラリ] - [ポリシー] の順に移動します。
- 3 [新規ポリシー] を選択します。  
空のポリシーが作成されました。
- 4 ポリシー名とバージョン番号を入力します。
- 5 [変数] タブを選択します。
- 6 ローカル変数を作成するには、[新規] をクリックします。
  - a 変数名を入力します。
  - b 変数タイプを選択します。

---

**注:** ポリシー変数の配列を作成するには、[配列] チェックボックスを選択します。

---

- c 変数の値を入力します。

**注:** 構成要素の変数の値をインポートするには、[構成にバインド]を使用できます。

- d [保存] をクリックします。

- 7 [定義] タブでは、ポリシー要素の追加とイベント ハンドラの設定を行うことができます。

ポリシー要素の詳細については、[vRealize Orchestrator クライアントのポリシー要素](#)を参照してください。

- 8 [保存] をクリックします。

ポリシーが構成されました。

#### 次のステップ

ポリシーを開始するには、ポリシーを選択し、[実行] をクリックします。ポリシーの実行名と、必要な入力パラメータ（要求された場合）を入力します。

ポリシーのステータスを表示するには、[アクティビティ] - [ポリシーの実行] の順に移動します。

### vRealize Orchestrator クライアントのポリシー要素

ポリシー要素を使用して、イベントが発生したときに事前定義された vRealize Orchestrator ワークフローまたはスクリプトを実行できます。

オブジェクトによってトリガされたイベントに対する応答としてワークフローまたはスクリプト実行をトリガするためのポリシー要素を追加できます。定期イベント要素を使用して、ワークフローまたはスクリプト実行をスケジュール設定できます。ルート要素を使用して、ポリシーの開始または停止の動作を設定できます。ポリシー要素には、ポリシー要素を実行する必要があるタイミングを定義するイベント ハンドラを含めることができます。

**注:** ポリシー要素を有効にするイベント ハンドラは、ワークフローまたはアクション スクリプトのいずれかになります。ワークフローとスクリプトの両方をイベント ハンドラに追加すると、ポリシーによってスクリプト トリガが無視され、ワークフロー トリガのみが使用されます。

イベント ハンドラ	説明
[OnInit]	ポリシー要素は、ポリシーを開始するたびにトリガされます。
[OnExit]	ポリシー要素は、ポリシーを停止するたびにトリガされます。
[OnExecute]	定期イベント要素によって使用されます。定期イベント要素で指定された時間内にポリシー要素をトリガします。

**注:** vRealize Orchestrator データベースにプラグインされたテクノロジーは、一意のイベント ハンドラを保持できます。たとえば、SNMP プラグインを使用すると、SNMP ベースのポリシー要素を作成するときに、[OnTrap] イベント ハンドラを使用できます。

ポリシー要素は、[ポリシー編集] ウィンドウの [定義] タブで構成されます。

### vRealize Orchestrator クライアントでのポリシーの実行の管理

vRealize Orchestrator Client を使用して、ポリシーの優先度と、vRealize Orchestrator サーバが再起動されたときのポリシーのサーバ起動時の動作を管理できます。

## 前提条件

ポリシーを作成して実行します。詳細については、『[vRealize Orchestrator クライアントでのポリシーの作成および適用](#)』を参照してください。

## 手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [アクティビティ]-[ポリシー実行] に移動します。
- 3 管理するポリシー実行をクリックします。
- 4 [停止] をクリックします。  
ポリシーの状態が [停止] に変更されます。
- 5 [全般] タブで、ポリシーの優先度とサーバ起動時の動作を設定します。
- 6 ポリシーを再起動するには、[実行] をクリックします。  
ポリシーの状態が [実行中] に変更されます。

## vRealize Orchestrator クライアントのリソース要素

ワークフローでは、vRealize Orchestrator とは独立して作成したオブジェクトを属性として使用することができます。ワークフロー内で外部オブジェクトを属性として使用するには、そのオブジェクトをリソース要素としてサーバにインポートします。

vRealize Orchestrator ワークフロー内でリソース要素として使用できるオブジェクトには、イメージ ファイル、スクリプト、XML テンプレート、HTML ファイルなどがあります。vRealize Orchestrator サーバ上で実行されるワークフローの場合、vRealize Orchestrator にインポートされたすべてのリソース要素を使用することができます。

オブジェクトをリソース要素として vRealize Orchestrator にインポートすることにより、そのオブジェクトを 1 か所で変更し、そのリソース要素を使用しているすべてのワークフローに対して変更内容を自動的に反映させることができます。

1 つのリソース要素の最大サイズは 16 MB です。

リソース要素には、インポート、エクスポート、リストア、更新、および削除の操作を行うことができます。

## vRealize Orchestrator 監査ログの使用

vRealize Orchestrator オブジェクトの監査ログとその実行を表示できます。また、vRealize Orchestrator Client のすべてのイベントのログを表示することもできます。

## 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 特定の vRealize Orchestrator オブジェクトのログを表示するには、オブジェクトを開き、[監査] タブをクリックします。

- 3 vRealize Orchestrator Client で発生したすべてのイベントのログを表示するには、クライアント ナビゲーションメニューの [監査ログ] をクリックします。

---

**注:** 重要度レベル別に監査ログをフィルタリングすると、選択した重要度レベル以上のすべてのログ エントリが表示されます。たとえば、[情報] によるフィルタリングでは、[警告] と [エラー] のログ エントリも表示されます。

---

# vRealize Orchestrator パッケージの使用

# 5

vRealize Orchestrator Client を使用してパッケージを作成、エクスポート、およびインポートします。パッケージを利用すれば、ワークフロー オブジェクトをエクスポートして他の vRealize Orchestrator インスタンスで使用できます。

パッケージには、ワークフロー、アクション、ポリシー、構成要素、またはリソース要素を含めることができます。

パッケージに要素を追加すると、vRealize Orchestrator によって依存関係がチェックされ、従属要素がパッケージに追加されます。たとえば、アクションまたは他のワークフローが使用されるワークフローを追加した場合は、vRealize Orchestrator によってそのアクションと他のワークフローがパッケージに追加されます。

パッケージをインポートすると、サーバによって、コンテンツのさまざまな要素のバージョンと、対応するローカル要素のバージョンが比較されます。この比較により、ローカル要素とインポートされた要素のバージョンの違いが明確になります。ユーザーは、パッケージをインポートするか、特定の要素を選択してインポートするかを判断できます。

vRealize Orchestrator Client で作成されるほとんどのオブジェクト（リソース要素を除く）の場合、パッケージがこれらのオブジェクトをエクスポートおよびインポートする唯一の方法です。

パッケージはデジタル著作権管理を使用して、受信側サーバでのパッケージのコンテンツの使用方法を制御できます。パッケージは vRealize Orchestrator によって署名され、データ保護のために暗号化されます。パッケージは、X509 証明書を使用して、度のユーザーが要素のエクスポートや再配布を行うかを追跡できます。

---

**注:** vRealize Orchestrator Client は、暗号化パッケージの作成、インポート、またはエクスポートをサポートしていません。vRealize Orchestrator Client で暗号化パッケージを使用するには、Orchestrator Java クライアントにインポートして、パッケージ暗号化を削除する必要があります。

---

## vRealize Orchestrator クライアントでのパッケージの作成

パッケージ内のワークフロー、ポリシー、アクション、プラグイン参照、リソース要素、および構成要素をエクスポートおよびインポートできます。パッケージ オブジェクトに関連するすべての依存要素は、パッケージに自動的に追加され、バージョン間の互換性が確保されます。依存要素を削除するには、最初に関連するパッケージ オブジェクトを削除する必要があります。

vRealize Orchestrator Client で作成されるほとんどのオブジェクト（リソース要素を除く）の場合、パッケージがこれらのオブジェクトをエクスポートおよびインポートする唯一の方法です。



### 前提条件

パッケージに追加することができるワークフロー、アクション、ポリシーなどのオブジェクトが vRealize Orchestrator サーバに含まれていることを確認します。

### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [アセット] - [パッケージ] に移動します。
- 3 [新しいパッケージ] をクリックします。
- 4 [全般] タブでパッケージの名前と説明を入力します。

---

**注:** vRealize Orchestrator Client でのパッケージ名には特殊文字を使用することはできません。

---

- 5 [コンテンツ] タブで [追加] をクリックします。
- 6 パッケージに追加するオブジェクトを選択し、[追加] をクリックします。

---

**注:** 依存要素はパッケージに自動的に追加されますが、パッケージの作成中に [コンテンツ] タブには表示されません。依存要素を表示するには、パッケージの作成後に [コンテンツ] タブを選択します。

---

- 7 パッケージの作成を完了するには、[作成] をクリックします。

## vRealize Orchestrator クライアントでのパッケージのエクスポート

vRealize Orchestrator Client を使用して、パッケージを別の vRealize Orchestrator 環境にエクスポートできます。

### 前提条件

エクスポートする vRealize Orchestrator オブジェクトを含むパッケージを作成します。詳細については、『[vRealize Orchestrator クライアントでのパッケージの作成](#)』を参照してください。

### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [アセット] - [パッケージ] に移動します。
- 3 パッケージで、[エクスポート] をクリックします。
- 4 (オプション) 他のエクスポート オプションを選択します。

オプション	説明
パッケージに構成の属性値を追加する	構成要素の属性値をエクスポートします。
パッケージに構成の SecureString 属性値を追加する	SecureString 構成属性値をエクスポートします。
パッケージにグローバル タグを追加する	グローバル タグをエクスポートします。

## 5 パッケージをインポートするユーザーのアクセス権限を設定します。

オプション	説明
コンテンツの表示	ユーザーは、パッケージのコンテンツを表示できます。
パッケージへの追加	ユーザーは、インポートされたパッケージのコンテンツを他のパッケージに追加できます。
コンテンツの編集	ユーザーは、パッケージのコンテンツを編集できます。

## 6 [OK] をクリックします。

**注:** 拡張子が `.package` のファイルは、ローカル マシン上のデフォルト フォルダに保存されます。カスタム フォルダを設定するには、ブラウザのストレージ設定を変更します。

パッケージがエクスポートされます。エクスポートされたオブジェクトを別の vRealize Orchestrator 環境で使用できるようになりました。

# vRealize Orchestrator クライアントでのパッケージのインポート

vRealize Orchestrator Client を使用して、ワークフロー パッケージをインポートできます。パッケージをインポートすることで、ある vRealize Orchestrator サーバのオブジェクトを別のサーバで再利用できます。

### 前提条件

- 変更した標準の vRealize Orchestrator オブジェクトをすべてバックアップします。
- リモート サーバ上で、インポートするオブジェクトが含まれるパッケージを作成し、エクスポートします。

### 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [アセット] - [パッケージ] に移動します。
- 3 [インポート] をクリックします。インポートする `.package` ファイルを参照し、[開く] をクリックします。
- 4 インポートされたパッケージの情報を確認します。
  - a [全般] タブには、名前、説明、パッケージに含まれる項目数などの、インポートされるパッケージに関する情報と、証明書の情報が表示されます。  
  
ファイルをインポートする前に、移行元 vRealize Orchestrator インスタンスの発行者証明書を信頼するよう求められる場合があります。
  - b [パッケージ要素] タブには、インポート ファイルに含まれるオブジェクトが一覧で表示されます。パッケージ内のオブジェクトのバージョンが、サーバ上のバージョンより新しい場合は、インポート対象としてそのオブジェクトのバージョンが自動的に選択されます。前のバージョンの vRealize Orchestrator 要素は、手動で選択する必要があります。
  - c パッケージから構成要素の属性値をインポートしない場合は、[構成の属性値をインポートする] の選択を解除します。
  - d ドロップダウン メニューから、タグをインポートするかどうかを選択します。

5 [インポート] をクリックします。

# vRealize Orchestrator クライアントの トラブルシューティング

# 6

メトリック、トークンの再生、検証、およびデバッグを使用して vRealize Orchestrator インスタンスのトラブルシューティングおよび監視を行うことができます。

この章には、次のトピックが含まれています。

- [vRealize Orchestrator クライアントのメトリック データ](#)
- [vRealize Orchestrator クライアントでのワークフロー トークンの再生の使用](#)
- [vRealize Orchestrator ワークフローの検証](#)
- [vRealize Orchestrator クライアントでのワークフロー スクリプトのデバッグ](#)

## vRealize Orchestrator クライアントのメトリック データ

vRealize Orchestrator 管理者は、ワークフロー プロファイルおよびシステム ダッシュボードのメトリックを使用して、vRealize Orchestrator システムとワークフローのトラブルシューティングを行うことができます。

プロファイル機能は、ワークフローの実行に関するメトリック データを収集します。ワークフローのプロファイルはデフォルトで有効になっています。自動プロファイルは、[コントロール センター] - [拡張機能プロパティ] - [profiler-7.6.0] で無効にできます。

vRealize Orchestrator Client のメトリック データのその他のソースは、システム レベルのメトリックを提供するシステム ダッシュボードです。詳細については、『[vRealize Orchestrator システム ダッシュボードの使用](#)』を参照してください。

## vRealize Orchestrator クライアントでのワークフローのプロファイル

ワークフローの実行をプロファイルして、vRealize Orchestrator のトラブルシューティングや最適化を行うことができます。

vRealize Orchestrator Client のプロファイル機能を使用すると、ワークフローの実行に関する有用なメトリック データを収集できます。このデータを使用してワークフローのパフォーマンスを最適化できます。デフォルトでは、ワークフローの実行は自動的にプロファイルされます。vRealize Orchestrator コントロール センターの [拡張機能プロパティ] ページで自動プロファイルを無効にして、手動でプロファイラを実行できます。手動でプロファイルを実行するには、ライブラリ内でワークフローを見つけ、[アクション] - [プロファイル] を選択します。

### 前提条件

ワークフローの実行

## 手順

- 1 vRealize Orchestrator Client にログインします。
- 2 [アクティビティ] - [ワークフローの実行] の順に移動します。
- 3 ワークフローの実行を選択します。

ワークフローの実行スキーマでは、個々のワークフロー アイテムに関するデータを表示できます。データには、合計実行時間、最大実行時間、およびアイテムの実行数が含まれます。この情報は、ページの右上にあるドロップダウン メニューからフィルタできます。

- 4 [パフォーマンス] タブを選択します。

このタブには、ワークフロー実行の CPU 時間、実行時間、トークン サイズ、およびワークフロー アイテム データに関するメトリック データが表示されます。

---

**注:** ワークフローが追加入力を待機しているときなど、ワークフローの実行が一時停止している場合、CPU 時間メトリックは完了前に発生するランタイム スレッドのみを取得します。

---

## 次のステップ

プロファイルから収集したデータを使用して、ワークフローを最適化します。

## vRealize Orchestrator システム ダッシュボードの使用

管理者は、vRealize Orchestrator Client システム ダッシュボードを使用して、vRealize Orchestrator 環境のノードに関する有用なメトリック データを収集できます。

システム ダッシュボードには、vRealize Orchestrator Client ダッシュボード ページの上部にある [システム] タブをクリックしてアクセスできます。提供されるデータは次のとおりです。

- ノードのステータス
- ノードのプロパティ
- クラスタ設定 クラスタ設定は、システム ダッシュボードからのみ表示できます。これらの設定を変更するには、vRealize Orchestrator コントロール センターの [Orchestrator クラスタ管理] ページに移動します。
- スレッド情報
- ヒープ メモリ
- 非ヒープ メモリ
- ファイル システムの使用状況
- 認証データ
- Orchestrator データベース接続 プール
- プロセスの入力引数

このデータは、vRealize Orchestrator 環境の個々のノードの状態を監視し、問題のトラブルシューティングに使用できます。個々のノード間を移動するには、システム ダッシュボードの上部にある、ノードに関連付けられているタブをクリックします。

## vRealize Orchestrator クライアントでのワークフロー トークンの再生の使用

トークンの再生機能を使用して、ワークフロー実行中のアイテムの移行を表示できます。

トークンの再生機能は、ワークフロー アイテム間のそれぞれの移行のコンテキスト情報を記録します。ワークフロー アイテムごとに、トークンの再生機能はワークフローの実行の開始と終了、およびワークフロー アイテムの実行の終了時に変更された変数を記録します。トークンの再生では、各ワークフロー アイテムに対して生成されたスクリプト ログ メッセージも参照されます。

---

**注:** ワークフロー アイテムの移行に関するデータは、vRealize Orchestrator PostgreSQL データベースに保存されます。このデータは、ワークフローの実行が削除されるとデータベースから削除されます。

---

### 前提条件

ワークフローの実行

### 手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [アクティビティ] - [ワークフローの実行] の順に移動します。
- 3 ワークフローの実行を選択します。
- 4 左側のメニューからワークフローの実行項目を選択します。

[変数] と [ログ] のタブには、そのワークフロー アイテムに固有の情報が表示されるようになりました。

## vRealize Orchestrator ワークフローの検証

vRealize Orchestrator には、ワークフローの検証ツールが用意されています。ワークフローを検証することで、ワークフローのエラーを特定したり、要素間のデータの流が正しいかどうか確認したりすることができます。

vRealize Orchestrator ではデフォルトで、ワークフローの実行時に必ずワークフロー検証が実行されます。

ワークフローの検証を実行すると、検証ツールによってエラーや警告のリストが作成されます。このリストのエラーをクリックすると、エラーが含まれるワークフロー要素が強調表示されます。

ワークフロー エディタで検証ツールを実行すると、検出されたエラーを修正するためのクイック修正アクションが提示されます。クイック修正アクションによっては、追加情報や入力パラメータが必要になるものもあります。また、そのままエラーを解決できるクイック修正アクションもあります。

ワークフローの検証では、要素間のデータのバインドと接続がチェックされます。ワークフローの検証ではワークフロー内の各要素が実行するデータ処理がチェックされません。そのため、スキーマ要素の関数が正しくない場合、有効なワークフローが正しく実行できず、誤った結果が生成されることがあります。

## vRealize Orchestrator クライアントでのワークフローの検証と検証エラーの修正

ワークフローを実行するには、事前にワークフローを検証しておく必要があります。編集のためにワークフローを開いている場合は、検証エラーの修正のみ可能です。

### 前提条件

スキーマ要素がリンクされ、バインドが定義された、検証すべき完成したワークフローがあることを確認します。

### 手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、検証するワークフローを選択します。
- 3 [編集] をクリックします。
- 4 上部メニューから [検証] をクリックします。

ワークフローが有効な場合は、確認メッセージが表示されます。ワークフローが無効な場合は、エラー リストが表示されます。

- 5 無効なワークフローの場合は、エラー メッセージをクリックし、問題を解決するための適切な手順を実行します。

検証ツールによって、エラーが発生しているスキーマ要素が強調表示されます（赤色のアイコンが追加されます）。クイック修正アクションが表示される場合もあります。

- 提示されたクイック修正アクションに同意する場合は、そのアクションをクリックして実行します。
- 提示されたクイック修正アクションに同意しない場合は、[ワークフローの検証] ダイアログ ボックスを開じ、手動でスキーマ要素を修正します。

---

**重要:** vRealize Orchestrator によって提示された修正方法が適切かどうか必ず確認してください。

---

たとえば、未使用の属性を削除するように提示されることがありますが、実は属性が正しくバインドされていないことが原因の場合もあります。

- 6 検証エラーがすべてなくなるまで前述の手順を繰り返します。

ワークフローが検証され、検証エラーが修正されました。

### 次のステップ

ワークフローを実行できます。

## vRealize Orchestrator クライアントでのワークフロー スクリプトのデバッグ

ワークフロー アイテムのスクリプトにブレークポイントを挿入して、ワークフローの実行をデバッグできます。

ブレークポイントに達した場合は、いくつかの方法でデバッグ プロセスを継続することができます。ワークフロー スキーマから要素をデバッグする場合は、ワークフローの実行に関する一般的な情報の表示、ワークフローの変数の変更、ウォッチ式の追加、ログ メッセージの確認を行うことができます。

**注:** 非本番環境ですべてのスクリプトのデバッグを実行します。

#### 手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 ライブラリから、ワークフローを選択します。
- 3 ワークフロー スキーマを開き、ワークフロー要素を選択して、[スクリプト] タブをクリックします。
- 4 ブレークポイントを挿入するには、行番号の左側にある赤い円をクリックします。

**注:** ブレークポイントは、スクリプトを使用したワークフロー要素にのみ挿入可能です。

- 5 デバッグ モードでワークフローを実行するには、[デバッグ] をクリックします。  
ワークフローで入力パラメータが必要な場合は、そのパラメータを指定します。
- 6 ブレークポイントに到達してワークフローの実行が停止したら、以下に示すいずれかのオプションを選択します。

オプション	説明
続行	別のブレークポイントに到達するか、ワークフローの実行が完了するまで、ワークフローの実行を再開します。
ステップイン	このオプションを使用して、ワークフロー要素にステップインできます。ワークフロー エディタでデバッグしている場合は、ネストされたワークフロー要素にステップインすることはできません。
ステップオーバー	スキーマ内の現在の要素をスキップし、次の要素でワークフローの実行を停止します。

**注:** デバッガで、現在のブレークポイントをクリックして無視するように指示できます。これにより、ブレークポイントのシンボルが緑色の三角形に変更されます。

- 7 (オプション) [デバッガ] タブで、ウォッチ式を挿入します。  
式を使用して、特定の変数の完了に従うことができます。
- 8 (オプション) [デバッガ] タブで、変数の値を変更します。