# **HP Operations Orchestration**

Windows および Linux向け

ソフトウェアバージョン: 10.01

Studio オーサリングガイド

ドキュメントリリース日: 2013 年 8 月 (英語版) ソフトウェアリリース日: 2013 年 8 月 (英語版)





#### 保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載は、追加保証を提供 するものではありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

## 権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コン ピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政 府に使用許諾が付与されます。

著作権について

© Copyright 2013 Hewlett-Packard Development Company, L.P.

## 商標について

Adobe™は、Adobe Systems Incorporated (アドビシステムズ社)の登録商標です。

本製品には、'zlib' (汎用圧縮ライブラリ)のインタフェースが含まれています。'zlib': Copyright© 1995-2002 Jean-loup Gailly and Mark Adler.

AMDおよびAMD Arrowのシンボルは、Advanced Micro Devices, Inc.の登録商標です。

Google™およびGoogle Maps™は、Google Inc.の登録商標です。

Intel®、Itanium®、Pentium®、Intel®およびXeon®は、Intel Coporationの米国およびその他の国における登録商標です。

Javaは、Oracle Corporationおよびその関連会社の登録商標です。

Microsoft®, Windows®、Windows® XP、およびWindows Vista®は、米国におけるMicrosoft Corporationの登録商標です。

Oracleは、Oracle Corporationおよびその関連会社の登録商標です。

UNIX®は、The Open Groupの登録商標です。

## ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ソフトウェアバージョンの番号は、ソフトウェアのバージョンを示します。
- ドキュメントリリース日は、ドキュメントが更新されるたびに変更されます。
- ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

更新状況、およびご使用のドキュメントが最新版かどうかは、次のサイトで確認できます。http://support.openview.hp.com/selfsolve/manuals

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの登録は、次のWebサイトから行なうことができます。 http://h20229.www2.hp.com/passport-registration.html (**英語サイト**)

または、HP Passport のログインページの [New users - please register] リンクをクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

## サポート

HPソフトウェアサポートオンラインWebサイトを参照してください。http://support.openview.hp.com

このサイトでは、HPのお客様窓口のほか、HPソフトウェアが提供する製品、サービス、およびサポートに関する詳細情報をご覧いただけます。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様のビジネスを管理するのに必要な対話型の技術サポートツールに、素早く効率的にアクセスできます。HPソフトウェアサポートのWebサイトでは、次のようなことができます。

- 関心のあるナレッジドキュメントの検索
- サポートケースの登録とエンハンスメント要求のトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HPサポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

ー部のサポートを除き、サポートのご利用には、HP Passportユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDを登録するには、次のWebサイトにアクセスしてください。

http://h20229.www2.hp.com/passport-registration.html (英語サイト)

アクセスレベルの詳細については、次のWebサイトをご覧ください。

http://support.openview.hp.com/access\_level.jsp

HP Software Solutions Nowは、HPSWのソリューションと統合に関するボータルWebサイトです。このサイトでは、お客様のビジネスニーズを満たすHP製品ソリューション を検索したり、HP製品間の統合に関する詳細なリストやITILプロセスのリストを閲覧することができます。このサイトのURL はhttp://h20230.www2.hp.com/sc/solutions/index.jspです。

## このPDF版オンラインヘルプについて

本ドキュメントはPDF版のオンラインヘルプです。このPDFは、ヘルプ情報から複数のトピックを簡単に印刷したり、オンラインヘルプをPDF形式で閲覧できるようにするために提供されています。このコンテンツは本来、オンラインヘルプとしてWebブラウザーで閲覧することを想定して作成されているため、トピックによっては正しいフォーマットで表示されない場合があります。また、インタラクティブトピックの一部はこのPDF版では提供されません。これらのトピックは、オンラインヘルプから正しく印刷することができます。

# 第1章

# 目次

目次	4
『HP Operations Orchestration Studio オーサリングガイド』へようこそ	7
図 解による HP OO Studio 概要	7
HP OO Studio の開始 – ワークフローの主なステップ	17
HP OO Studio ウィンド ウの外 観 の調 整	18
ベストプラクティス	20
ー 般 的なベストプラクティス	20
共有コンテンツのベストプラクティス	21
命名のベストプラクティス	23
フローのベストプラクティス	25
オペレーションのベストプラクティス	26
ステップのベストプラクティス	28
トランジションのベストプラクティス	30
入力のベストプラクティス	30
デバッグのベストプラクティス	31
Studio の構成のベストプラクティス	31
説明のベストプラクティス	32
ソースコントロール管理のベストプラクティス	39
HP OO Studio での各 言 語 の使 用 - ローカライズ	39
Studio での現在のロケールの変更	40
Studio の表示言語	41
コンテンツパックのローカライズ	41
プロジェクトのローカライズ	41
新しいコンテンツパックの作成	42
プロジェクトの使用	43
プロジェクトの管理	43
[プロジェクト] ペインを使用したフォルダーの管理	49
ソースコントロールの使用	51

初期のソースコントロールリポジトリの作成	51
HP OO Studio のソースコントロールの使用	52
[SCM 変更] パネル	55
複数の作成者での作業	
参照情報	61
コンテンツパックの使用	66
コンテンツパックのプロジェクトへのインポート	66
プロジェクトのコンテンツパックの管理	68
構成アイテムの管理	
カテゴリの設 定	72
ドメインタームの設 定	74
グループエイリアスの設定	77
スクリプトレットの設定	
選択リストの設定	
システムアカウントの設定	
システムフィルターの設定	
システムプロパティの設定	
フロー作成 – 基本	97
フローの作成 – 手順	
新しいフローの作成	
フロー内 でのステップの作 成	
フローの外 観 の調 整	110
フローの変 更	
入力の作成	118
入 カソースの指定	125
入 カ データの評 価	
トランジションの作成	
レスポンスの設 定	
出力と結果の作成	147
オペレーションの出力の設定	
ステップの結果の設定	

出力および結果の	)フィルタリング	
変数の使用		
リターンステップの作 成		
高度な作成		
フロー内 でのサブフロー	−の作成	
並列分岐ステップのフ	ローの作成	
複数のインスタンスステ	テップのフローの作成	
フロー内 でのスクリプト	レットの使用	
フローでの正規表現の	D使用	
コンテンツの検査		
[問題] ペインでのフロ-	ーの検査	
フローのテストとデバック	זֿ	
複雑なフローのデバッ	ヷ	
Central の Studio での	デバッグ	
実行する操作		
コンテンツパックのエクスポー	<b>−</b> ト	
フローとオペレーションの管	理	
オペレーションの作 成		
フローまたはオペレーシ	ョンの検索	
フローおよびオペレーシ	′ョンの⊐ピー	
ハードコピーとソフト	→コピーの切り替え	
ハードコピー内のフ	プラグインの置き換え	
フローとオペレーションの	の使用方法の詳細	
フローおよびオペレーシ	ョンに関するドキュメントの生成	
フローとオペレーションの	のバージョン履歴管理	
フローおよびオペレーシ	ィョンのブックマーク	
HP OO 9.x からアップグレ・	ードする際のトラブルシューティン・	グ254
Studio のユーザーイン	ターフェースアイテムの場所	
HP OO 9.x と10.00の	バージョン比較	

# 『HP Operations Orchestration Studio オーサリングガ イド』へようこそ

HP OO Studio はスタンドアロンのオーサリングプログラムで、フローの作成、変更、およびテストに使用 します。

図解によるHP OO Studio概要	7
HP OO Studio の開始 – ワークフローの主なステップ	17
HP OO Studio ウィンド ウの外 観 の調 整	. 18
ベストプラクティス	. 20
HP OO Studio での各言語の使用 - ローカライズ	39

## 図解による HP 00 Studio 概要



Studio は主に以下の要素で構成されています。

- [プロジェクト] ペイン (左側)。作業中のプロジェクトと、編集可能なフロー、オペレーション、その他のプロジェクトで使用可能な HP OO オブジェクトが表示されます。
- [依存関係] ペイン (左側)。インポートされたコンテンツパックが表示されます。このペインでは、コンテンツパックをインポート、削除、閉じることができます。
- 作成ペイン(中央)。フローを作成ペインで開いている場合、次の3つのタブが、作成ペインの下 部で使用可能になります。
  - [設計]タブ。フロー図で作業できます。
  - [プロパティ] タブ。フロー、オペレーション、構成オブジェクトのプロパティを設定できます。
  - [インスペクター] タブ。個々のステップとトランジションのプロパティを設定できます([設計] タブを開いているときのみ使用可能)。
- [アイコン] ペイン (右側)。オペレーションまたはステップで使用するアイコンが集められています。[アイ コン] タブをクリックするとこのペインが開きます。
- [ブックマーク] ペイン (右側)。よく使用するオペレーションやフローのショートカットを保存できます。 [ブックマーク] タブをクリックするとこのペインが開きます。
- [**フロー変数**] ペイン (右側)。フローで使われているフロー変数と、各フロー変数の使用方法のリストと説明が表示されます。[**フロー変数**] タブをクリックするとこのペインが開きます。
- [参照] ペイン (下部)。既存のフローでフローやオペレーションがどのように使用されているかを表示 します。[参照] タブをクリックするとこのペインが開きます。
- [問題] ペイン (下部)。選択したフローやオペレーションで発生している問題を表示します。[問題] タブをクリックするとこのペインが開きます。
- [SCM メッセージ] ペイン (下部)。ソースコントロール関連のメッセージを表示します。[SCM メッセージ] タブをクリックするとこのペインが開きます。詳しくは、「ソースコントロールの使用」を参照してください。
- [SCM 変更] ペイン (下部)。ソースコントロールの最新の変更を表示します。[SCM 変更] タブをク リックするとこのペインが開きます。詳しくは、「ソースコントロールの使用」を参照してください。
- [検索] ペイン (下部)。 フローやオペレーションを検索 できます。 [検索] タブをクリックするとこのペイン が開きます。

## [プロジェクト] ペイン

[プロジェクト] ペインには、プロジェクトツリー (プロジェクトの編集可能なコンテンツが含まれる階層フォルダー構造)が表示されます。

- Library フォルダーには、フローとオペレーションが含まれます。
- Configuration フォルダーには、オペレーション結果の処理、レポートの作成、およびフローの実行 を容易にするために使用できるその他のHP OO オブジェクト (フィルター、スクリプトレット、システム プロパティなど)が格納されています。

プロジェクト	
+ 🧿 🧿  🕬 🗙 🍣	
WyProject1     WyProject1     Wy Ops Flows     Wy Ops Flows     Onfiguration     Oracle Categories     One Categories     One Operation     One of the comparises     Selection Lists     One System Accounts     One System Filters     One System Properties	

GUI ፖイテム	説明
プロジェクトの新規作成 🔸	新しいプロジェクトを作成します。
プロジェクトのインポート 🤌	別 のワークスペースの既存 のプロジェクトを参 照してインポート します。
コンテンツパックの作成 🥠	選択したプロジェクトのコンテンツパックを作成します。
削除 🗙	選択したプロジェクトをワークスペースから完全に削除します。
開< 蓉	現在選択されている、閉じたプロジェクトを開きます。
閉じる 🥬	現在選択されているプロジェクトを閉じて、グレー表示にしま す。
リフレッシュ ᇕ	現在選択しているプロジェクトのファイルをリフレッシュします。

プロジェクトの使用の詳細については、「プロジェクトの使用」(43ページ)を参照してください。

## [依存関係]ペイン

[依存関係] ペインには、使用可能なコンテンツパックが、オペレーションおよびフローが格納されたフォ ルダーとともに表示されます。



## 作成ペイン

作成ペインは Studio の中央にあるエリアで、フロー図の作業をする場所です。ここで、フロー図にス テップやステップ間のつながりを追加したり、フローの動作を決定するプロパティを設定したりします。

フローが作成ペインで開いている場合、次の3つのタブが、使用可能になります。

- [設計] タブ。フロー図で作業し、ステップやステップ間のつながりを追加します。
- [プロパティ] タブ。[プロパティ] シートが表示され、フローやオペレーションの選択リスト、フィルター、 スクリプトレットなどのプロパティを設定できます。
- [インスペクター] タブ。インスペクターが表示され、個々のステップとトランジションのプロパティを設定できます。



#### 作成ペインのツールバー

フローを作成ペインで開いている場合に、[設計]タブを開くと、作成ペインのツールバーが使用可能になります。

作成ペインのツールバーには、さまざまなタスクのショートカットとして機能するボタンが用意されています。

🗗 💾 🗒 🇞	🐲 🔉 🕒 📋 🗙 🦻 🥽 🥌 🐯 🖛
「 ロック リリーで選び 名前を付けて保存 保存 フロ・	R コピー 削除 やり直し 表示オデション F 切J取J 貼J付け 元に戻す ステップパレット ーのデバッグ
ボタン	機能
ステップ パレット 🧐 🔻	[ <b>ステップ</b> ] パレットが開きます。 ステップオブジェクトをキャンバスにド ラッグし ます。
表示オプション ころ  マ	[ <b>表 示 オプション</b> ] パレットの表 示
ツリーで選択 🇞	使用するフローまたはオペレーションを選択するためにライブラリツリーを 展開します。
フローのデバッグ 🎏	デバッガーを開き、現在のフローの実行を開始します。

作成ペインの使用の詳細については、「フロー作成 – 基本」(97ページ)を参照してください。

## [ステップ] パレット

•

ステップパレットには、リターンステップ、並列分岐ステップ、複数のインスタンスステップ、およびコールアウトをフローにドラッグするためのボタンがあります。 作成ペインのツールバーから [ステップパレット] ボタン

▼ をクリックして、[ステップ] パレットを表示します。



ボタン	説明
成功 🥝	<b>成功</b> リターンステップをフローにドラッグできます。
診断済み 📀	診断済みリターンステップをフローにドラッグできます。
操作なし	操作なしリターンステップをフローにド ラッグできます。
失敗 😳	<b>失敗</b> リターンステップをフローにドラッグできます。

ボタン	説明
並列分岐ステップ 🔧	並列分岐ステップをフローにドラッグできます。
複数のインスタンスステップ 🐚	複数のインスタンスステップをフローにド ラッグできます。
コールアウト 🤎	コールアウトをフローにドラッグして、ユーザーに情報を提 供 できます。
ドッキングバー	クリックして、パレットのドッキングとドッキング解除を行います。

リターンステップの使用の詳細については、「リターンステップの作成」(179ページ)を参照してください。

並列分岐ステップと複数のインスタンスステップの詳細については、「高度な作成」(183ページ)を参照してください。

## [表示オプション] パレット

[表示オプション] パレットには、作成ペイン上のフローの外観を変更するためのアイコンがあります。作

成ペインのツールバーから[表示オプション]ボタン をクリックして、[表示オプション]パレットを表示します。



ボタン	説明
ラベルの表示/非表示	オブジェクトのレスポンスラベルを表示または非表示にしま す。
接続されたレスポンスアイコンの表 示/非表示	オブジェクトのレスポンスアイコンを表示または非表示にしま す。
選択項目を水平方向に整列 🧮	選択項目を水平方向に整列します。

ボタン	説明
選択項目を垂直方向に整列 🛄	選択項目を垂直方向に整列します。
最前面へ移動 🖬	選択したオブジェクトを前面に移動します。
最背面へ移動 🛄	選択したオブジェクトを背面に移動します。
グリッドの表示/非表示	ステップを配置するときに利用できる作成ペインのグリッドを 表示します。ステップのドラッグ操作を止めると、グリッド上の 一番近い位置にそのステップがスナップされます。
ドッキングバー	クリックして、パレットのドッキングとドッキング解除を行います。

表示オプションの使用の詳細については、「フローの外観の調整」(110ページ)を参照してください。

## オブジェクトのプロパティシート

フロー、オペレーション、構成オブジェクトの[プロパティ]シートは、オブジェクトの値を追加、削除、変更するためのエディターです。 ライブラリに格納されている大部分のオブジェクトについては、[プロパティ] シートがオブジェクトを操作する際のインターフェースです。 [プロパティ]シートでは編集ができるだけで なく、UUID やオブジェクトのバージョンに関する情報が表示されます。

[プロパティ] シートでオペレーションのプロパティを変更した後、変更はこのオペレーション以前に作成されたステップを含めて、このオペレーションから作成されたすべてのステップに影響します。

- フローの [プロパティ] シートを表示するには、作成ペインでフローを開いて [プロパティ] タブをクリックします。
- オペレーションまたは構成オブジェクトの[プロパティ]シートを表示するには、ライブラリでオペレーションまたはオブジェクトを開いて[プロパティ]を選択します。

入力   結果   表 <sup>注</sup> ▲	示│説明│詳細│. ₽	スクリプトレット						$\leftrightarrow$ ×
	入力の追加	入力の削除		1		+	]	
	入力		必須	種類			ソース	
host			V	単一値	Ψ.	値:		<>
service			V	単一値	Ψ.	値:		<>
user				単一値	Ψ.	値:		4
password				単一値	-	値		<b>\$</b>

[プロパティ]シートの使用の詳細については、「入力の作成」(118ページ)および「オペレーションの出力の設定」(147ページ)を参照してください。

### ステップのインスペクター

ステップのインスペクターは、オペレーションに対する [プロパティ] シートと同様ですが、フローの1つのステップに関連します。 ステップのインスペクターでステップのプロパティを変更すると、変更はオペレーションのインスタンスであるこのステップのみに影響します。

インスペクター				_	Þ
ステップ名: Change Service Status					
入力 結果   表示   説明   詳細   スクリプトレット					
▲ ▼ 入力の概要				$\leftrightarrow \vdash \times$	
入力の追加 入力の削除		1	+		
入力	必須	種類	ソース		
host	$\overline{\checkmark}$	単一値 🔹	値:	4	
service	$\sim$	単一値 🗾 💌	値:	42	
requestedstatus	$\checkmark$	単一値 🗾	値: Running	4	
startdependants		単一値 📃 💌	値: true	4	
user		単一値 📃	値:	4	
password		単一値 🗾	値:	42	
serviceTimeout		単一値 🗾	値: 30	4	
operationTimeout		単一値 🗾	値: 30	4	

ステップのインスペクターの使用の詳細については、「入力の作成」(118ページ)を参照してください。

## トランジションのインスペクター

トランジションのインスペクターは、ステップ間のトランジションの設定に使用します。トランジションのイン スペクターを表示するには、2つのステップ間を結ぶ線を右クリックして [**プロパティ**]を選択します。

ተጋአለማቃ = =	,
トランジション名: Success	٦
「ゲート制御式トランジション」	
必須の役割エイリアス: <なし>	
□ このトランジション後にフロー実行をハンドオフ	
トランジションの ROI 値: 0.0	
<pre>\$[service] is now running</pre>	
- 一 インスペクター	
設計  プロパティ	

トランジションのインスペクターの使用の詳細については、「トランジションの作成」(133ページ)を参照 してください。

## [ブックマーク] ペイン

[ブックマーク] ペインは Studio ウィンド ウの右上にある [ブックマーク] タブをクリックすると開き、使用頻度の高いオペレーションやフローを簡単に見つけて使用できるようになっています。

よく使用するフローやオペレーションをライブラリからドラッグして [**ブックマーク**] ペインに追加することが可能です。 ブックマークの詳細については、「フローおよびオペレーションのブックマーク」(249ページ)を参照してください。



## [アイコン] ペイン

[アイコン] ペインは Studio ウィンド ウの右上 にある [アイコン] タブをクリックすると開き、ステップの内容を 分かりやすくするために使用できるアイコンのライブラリが格納されています。フローまたはステップのデ フォルトのアイコンにこの中のアイコンをドラッグして置き換えることができます。

アイコン					
ĵ	注釈				
o					
0	~	¥			
アプリケーション					
基本					
クラ	イアント				
デー	タベース				
電:	チメール				
ň	流合				
ネッ	トワーク				
オペレーテ	ィングシステレ	4			
7	の他				
<u> </u>	-//-				
-ב	ティリティ				

[アイコン] ペインの使用の詳細については、「フローの変更」(113ページ)を参照してください。

## [フロー変数]ペイン

[フロー変数] ペインは Studio ウィンドウの右上にある [フロー変数] タブをクリックすると開き、フローで

フロー変数 - Restart Service					
名前	#	\$	<b>&amp;</b>		
le raltpass	8	~	r		
📄 🔿 フロー入力	3	v	v		
フロー入力 "altpass" として宣言	-	v			
	-	v	v		
- 入力 "altpass" の値を "altpass" に割り当て				-	
📄 🚥 ユーザープロンプトを使用しないステップ入力	5				
"altpass" の値は "Change Service Status" の入力 "password" に割り当てるこ	-				
"altpass" の値は "Does Service Exist" の入力 "altpass" に割り当てることがで…	-				
- 入力 "altpass" の値を "Does Service Exist" の "altpass" に割り当て	-				
"altpass" の値は "ServiceStatus" の入力 "password" に割り当てることができま	-				
- 入力 "password" の値を "ServiceStatus" の "altpass" に割り当て	-				
≓-altuser	8	v	v		
↓ 📴 🔿 ר−ר ל	3	v	v	Ŧ	

使用されている変数のリストと、どのように使用されているかが説明されています。

[フロー変数] ペインの使用の詳細については、「変数の使用」(173ページ)を参照してください。

### [参照]ペイン

[参照] ペインは Studio ウィンド ウの下 端 にある [参照] タブをクリックすると開き、既存 のフローでオペレーションやフローがどのように使用されているかが表示されます。 このペインには、次の2種類の参照を表示できます。

- [これを使用する項目は?]-対象のオペレーションまたはフローから作成されたステップが含まれるフローを識別します。
- [これが使用する項目は?]-対象のオペレーションまたはフローが使用するオブジェクト(選択リスト、グループに割り当てられた権限、システムフィルターなど)を識別します。フローの場合は、フローのステップを作成したときのベースになったオペレーションおよびサブフローが含まれます。

参照	
/MyProject1/Library/My Ops Flows/Network Check からの参照	
オブジェクト	パス
⊡-Network Check	/MyProject1/Library/My Ops Flows/Network Check
Operation: Local Ping [Operation]	/Base [1.0.121]/Library/Operations/Network/Local Ping
-Operation: Local Traceroute [Operation]	/Base [1.0.121]/Library/Operations/Network/Local Traceroute
Operation: Manual [Operation]	/Base [1.0.121]/Library/Utility Operations/Manual

[参照] ペインの使用の詳細については、「フローとオペレーションの使用方法の詳細」(240ページ)を参照してください。

## [問題]ペイン

[問題] ペインは Studio ウィンド ウの下 端 にある [問題] タブをクリックすると開き、選択したフローやオペレーションが有効 かどうかを確認 できます。 このペインには、選択したフローまたはオペレーションの問

#### 題が、場所と説明とともに表示されます。

問題			
ソースの種類	名前	記明	場所
flow	testFlow	フローにリターンステップがありません	/MyProject1/Library/My Ops
flow	testFlow	フローに開始ステップがありません	/MyProject1/Library/My Ops
	ソースの種類 flow flow	ソースの種類 名前 flow testFlow flow testFlow	ソースの種類名前説明flowtestFlowフロー(にりターンステップがありませんflowtestFlowフロー(に開始ステップがありません

[問題] ペインの使用の詳細については、「[問題] ペインでのフローの検査」(207ページ)を参照してください。

## [検索]ペイン

[検索] ペインは Studio ウィンド ウの下 端 にある [検索] タブをクリックすると開き、フローやオペレーション を検索 できます。 Studio の検索 エンジンは Apache Lucene の構文を使用しています。

検索					
検索:	<i>&lt;すべてのフィールド&gt;</i> 💌 対象  SQL			▶ 完全 🔍	検索履歴
324 件					
ランクマ	名前		種類	1/2	
*****	SQL Script	1	operation	/Base [1.0.121]/Library/Operations/Databas	データベースに対して SQL 複数行コマンドを実行 🔺
*****	SQLScript	<u>9</u>	operation	/MyProject1/Library/My Ops Flows/oo-dotne	<pre>Runs the requested SQL commands se</pre>
*****	SQLScript	- 92	operation	/MyProject1/Library/My Ops Flows/oo-dotne	<pre>Runs the requested SQL commands se</pre>
****	SQLScript	2	operation	/MyProject1/Library/My Ops Flows/oo-dotne	<pre>Runs the requested SQL commands se</pre>
****	SQL Server Soft Reset	<b>*</b>	flow	/Databases [1.0.115]/Library/Accelerator Pa	MS SQL サービスを再起動し、MS SQL Server
****	Is An SQL Server	1	flow	/Databases [1.0.115]/Library/Operations/Dat	このサンブルは、指定されたホストが Microsoft S 💌

[検索] ペインの使用の詳細については、「フローまたはオペレーションの検索」(232ページ)を参照してください。

## HP 00 Studio の開始 - ワークフローの主なステップ

ここでは、HP OO Studio の作業での主なステップを簡単に説明します。各ステップの詳細な情報については、リンクをクリックして参照できます。



1. **新しいプロジェクトの作成**-業務で使用するフロー、オペレーション、フォルダー、構成アイテムを 含むプロジェクトを作成します。

「プロジェクトの管理」(43ページ)を参照してください。

2. コンテンツパックのインポート - 必要なコンテンツパックをインポートし、関連コンテンツをプロジェクト にコピーできるようにします。

**注**: 最初の2つのステップは、この順番で行わなくても構いません。コンテンツパックは、プロ ジェクトを作成する前にインポートすることもできます。

「コンテンツパックのプロジェクトへのインポート」(66ページ)を参照してください。

3. フローの作成 - オペレーション、入力、トランジション、レスポンス、リターンステップを配置してフ

ローを作成します。

「フローの作成 – 手順」(97ページ)と「高度な作成」(183ページ)を参照してください。

4. フローの実行とデバッグ-作成したフローをデバッガーで検証します。

「フローのテストとデバッグ」(208ページ)を参照してください。

5. コンテンツのリリースとコンテンツパックへのパッケージ - プロジェクトを、フロー、オペレーション、操作、構成アイテムを含めコンテンツパックにパッケージし、HP OO Central へ昇格します。

「コンテンツパックのエクスポート」(223ページ)を参照してください。

## HP 00 Studio ウィンドウの外観の調整

HP OO Studio のペインを以下の状態に設定できます。

- ドッキング Studio ウィンドウ内の標準の位置に設定する
- 移動可能 Studio ウィンドウ内で自由に再配置できるようにする
- **固定** Studio ウィンドウの脇に非表示にし、タブのみ表示してワークスペースにさらに広い領域を 使用できるようにする

## 実行する操作

#### ペインを移動可能にする

ペインを移動可能にすると、Studio ウィンドウ内の別の位置にペインを移動できます。

- 1. ドッキングされたペインの右上隅にある [移動可能] 🗖 ボタンをクリックします。
- 2. Studio ウィンドウ内の新しい位置にペインを移動します。

#### ペインをドッキングする

Studio ウィンドウ内の新しい位置にペインを移動した場合、ドッキングするとStudio ウィンドウ内の標準の位置にペインが戻ります。

移動可能なペインの右上隅にある [**ドッキング**] 🗳 ボタンをクリックします。 ドッキングされた位置にペインが戻ります。

#### ペインを最大化する

ペインを最大化して HP OO ウィンド ウ全体のサイズに拡大するには、 [最大化] 🗖 ボタンをクリックします。

#### ペインを元のサイズに復元する

最大化する前のサイズにペインを復元するには、[復元] 🗗 ボタンをクリックします。

## Studio ウィンドウの脇にペインを固定する

Studio ウィンド ウの脇 にペインを固 定してタブのみ表 示 するには、 [**固定**] 🛛 ボタンをクリックします。 ペインを表 示 するには、 タブをクリックします。

## ペインの固定を解除する

ペインを固定した後で、固定を解除するには、[**固定**] 🖳 ボタンを再度 クリックします。ペインの固定を 解除すると、ペインが開いて、Studio ウィンド ウ内 のドッキングされた位置 にペインが戻ります。

## ペインのサイズを調整する

ペインのサイズを大きくしたり小さくしたりするには、ペインの端をドラッグします。

### Studio ウィンドウをデフォルトのレイアウトにリセットする

Studio ウィンドウをデフォルトのレイアウトに戻すには、[ウィンドウ] > [ウィンドウレイアウトのリセット]を 選択します。

## ベストプラクティス

次のプラクティスは、複数の作成者がフローを作成している場合に特にお勧めします。

1
3
5
3
3
)
)
1
1
2
9
3 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

## 一般的なベストプラクティス

### フォルダー構造

ほかの作成者がフロー、オペレーション、およびユーティリティを見つけることができるように、フォルダー 構造が適切に定義されており、プロジェクト間で一貫性があるようにします。

### Studio 内での名前の変更

プロジェクト、フロー、またはほかの HP OO エンティティの名 前を変更する場合は、Studio で行います。 MS エクスプローラーなどのファイルブラウザーでエンティティの名 前を変更しないでください。

## ベストプラクティスのドキュメント

命名規則、フォルダー構造、フローの作成者が従うべきその他のガイドラインを記述するドキュメント を作成します。例については、「命名のベストプラクティス」(23ページ)を参照してください。

## 共有 コンテンツのベスト プラクティス

## 環境の識別

複数の作成者がプロジェクトで同時に作業するには、さまざまな方法があります。作業を始める前に、作成者がどのように作業を行うかについて考える必要があります。次に例を示します。

• 1人の作成者が1つまたは複数のプロジェクトで作業する



• 複数のプロジェクトが存在し、各作成者が1つのプロジェクトで作業する



• 1 つのプロジェクトで複数の作成者が作業する



• 複数のチームが存在し、各チームが1つのプロジェクトで作業する



• 複数の作成者が複数のプロジェクトを同時に進める



このドキュメントには、異なる環境ごとの推奨されるベストプラクティスを含めます。

#### コミュニティのやり取り

コミュニティのやり取りには、組織内でのコンテンツの共有と、ほかの組織とのコンテンツの共有があります。

作成時のコミュニティのやり取りには次のようなものが考えられます。

#### 1. コンテンツをコミュニティからダウンロードする

HP コンテンツや組織内の作成者により開発されたコンテンツパックのほかに、ほかの組織によりコ ミュニティに提供されたコンテンツをダウンロードすることができます。 このコンテンツは HPLN にあり ます。

詳細については、「コンテンツパックのプロジェクトへのインポート」(66ページ)を参照してください。

2. コミュニティで問題について話し合う

フロー開発の一部として、フロー内の問題、推奨コンテンツ、ベストプラクティスなどの問題に関してコミュニティで相談することができます。

このためには、関連するキーワードを使ってコミュニティにその話題がないか検索し、新しい話題 を作成し、経過をチェックします。

また、話し合いに積極的に参加したり、ベストプラクティスドキュメントを発行するなどしてコミュニ ティナレッジに貢献することもできます。このようなドキュメントでは、自由な編集を可能にするか、 読み取り専用にするかを決めることができます。

3. コンテンツをコミュニティに提供する

新しいコンテンツの開発と検査が終わり、生産環境に移行したら、これを、無料または有料コン テンツとしてコミュニティに提供することができます。コミュニティにおけるそのコンテンツの使用期間 内では、ほかのユーザーに更新するか、削除するか、公開するかを決定してコンテンツを更新す ることができます。

## 命名のベスト プラクティス

#### フローの種類

- さまざまなタイプのフローを種類ごとに分類すると、識別しやすくなります。
- フローの種類ごとに別のフォルダーに保存すると、探しやすくなります。
- また、フローの種類ごとに専用のアイコンを使用すると分かりやすくなります。

#### 命名規則:大文字と小文字

オブジェクトのタイプごとに、名前の大文字と小文字を揃えます。たとえば、作成したフローに名前を 付ける場合はタイトルケースにし、入力、出力、結果、フロー変数にはキャメルケースにします。

注: キャメルケースとは、最初の文字を小文字で、名前に含まれる次の語の最初の文字を大文字で表し、語間にスペースを入れない方法です。たとえば、serverNameのようになります。

タイトルケースとは、すべての語の最初の文字を大文字にします(「a」、「the」、「and」、「by」、「for」、「to」、「from」などの補助的な語を除く)。たとえば、**Reboot a Server**のようになります。

#### フローの種類別の命名規則: 接頭辞

フローの種類ごとに異なる命名規則を使用します。たとえば、フロー名に、フローの種類に基づいて次のように接頭辞を追加します。

- ユーザーインタフェースのフロー UI
- インフラストラクチャーのフロー IF

• ユーティリティのフロー - UT

#### 変数別の命名規則:接頭辞

フロー変数の種類ごとに異なる命名規則を使用します。たとえば、フロー名に、変数の種類に基づいて次のように接頭辞を追加します。

- フロー入力 FI
- ステップ入力 SI
- オペレーション入力 OI
- ローカル変数 LV
- グローバル変数 GV

#### 直感的な名前

- フローに、内容がわかるような名前を付け、フローの目的を説明するようにします。
- フローがより分かりやすくなる場合、ステップの名前を変更します。たとえば、String Comparator という名前のステップよりも、Validate <inputName> という名前のほうが分かりやすくなります。
- オペレーションやステップの名前を変更する場合、オペレーションやステップの目的がはっきりとわかる ような名前にしてください。
- フローがより分かりやすくなる場合、トランジションの名前を変更します。
- シングルタスクで実行するフローやオペレーションの場合、「<動詞><名前>」という形式で名前を 付けます。たとえば、Send Mail、Create Snapshotのようになります。
- サンプルのフローでは、名前に "Sample" という語を使用します。たとえば、Send Mail Sample、Create Snapshot Sample のようになります。
- 条件に一致するものがあるかどうかをチェックするフローの場合、問いを名前として使用します。たとえば、Is Computer Account Enabled のようになります。
- システムまたは環境に関する情報を収集するヘルスチェックフローの場合、フロー名に "Health check" を含めます (専用の Health Check フォルダーがある場合を除く)。たとえば、Solaris Health Check のようになります。

#### 単語の使用

ー部の入力名は、多くのオペレーションやステップで共通で使用します。次の入力名は、HP OO コン テンツで使用されています。

- host Windows の場合、host は、オペレーションが動作するマシン (パフォーマンスカウンターの取得先またはサービスの再起動を行うホストなど)を指します。セキュアシェル (SSH) オペレーションの場合、host は、コマンドを実行するマシンを指します。
- username マシンにログオンするために使用するアカウント名。
- password マシンにログオンするために使用するパスワード。

その他の共通の入力名には、次のようなものがあります。

- mailHost 電子メールの送信元のホストマシンです。
- target host が別のシステムにも影響を与える場合、host により影響を受けるシステムを target と呼びます。たとえば、server1へのSSHで server2への pingを実行する場合、host は server1 で target は server2です。

## フローのベストプラクティス

#### フローの設計

作成するフローの構造を、構築を始める前に設計します。

#### フローはシンプルに、1 画面に格納

フローは、Studio の最大の大きさである 1024 x 768 の画面上のキャンバスに、1:1 の表示倍率に合わせて作成します。これより大きいフローにすることも不可能ではありませんが、フローが大きくなる場合、そのステップのシーケンスの一部をサブフローに分割できないかをよく確認してください。

#### フローの再利用

フローを再利用できるように設計します。より複雑なフローでサブステップとして再利用可能な単純な フローのコレクションを作成します。

#### 変更の前にフロー/オペレーションが使用されている箇所のチェック

フローやオペレーションを変更する前に、[参照] > [これを使用する項目は?] を使用して、ほかのフローでこのフローやオペレーションを使用していないかを確認します。

#### 変更の前にフローをコピー

変更の前に、必ずフローのコピーを作成してください。両方のフローが必要ではなく、元のフローもほかで使用されていない場合でも、変更がうまくいかなかった場合に備えて元のフローをバックアップとして保存しておきます。コピーしたフローが完成したら、元のフローを削除してもかまいません。

#### 一貫性を持たせる

異なるフローを、同じ開始および終了位置を使用して設計します。

## オペレーションのベストプラクティス

## オペレーションを変更するときの注意

オペレーションのプロパティを変更する ([プロパティ] シート内) 場合、変更が、以前にこのオペレーション から作成されたステップを含めて、このオペレーションをステップとして使用しているすべてのフローに影響することに注意してください。オペレーションのプロパティを変更すると、そのオペレーションを使用して いるほかのフローが壊れる場合があります。オペレーションのコピーを作成し、そのコピーを変更すること をお勧めします。変更が1回限りの内容の場合、オペレーションではなくステップを変更します。

## オペレーションの変更が必要になりそうな場合はディープコピー

フローをコピーするときに、オペレーションのプロパティの変更が必要になる可能性がある場合は、 [ディープコピー] コマンドを使用することをお勧めします。このコマンドではオペレーションがコピーされるため、元のオペレーションに影響を与えずに変更することができます。

## ディープコピー用のフォルダーの作成

[ディープコピー] コマンドを使用してフローをコピーする場合、フローとそのオペレーション用の新しいフォ ルダーを作成することをお勧めします。

#### カスタマイズしない場合は元のオペレーションを使用

オペレーションをカスタマイズする必要がない場合は、コピーしたオペレーションではなく元のバージョンの オペレーションを使用します。フォルダー内が、不要なオペレーションのコピーでいっぱいにならないよう にしてください。

### 統合のために元の名前とメッセージを保持

統合には独自のルールとベストプラクティスがあります。統合で作業する場合、オペレーションを統合 する製品とできる限り同じにしてください。

- フロー、オペレーション、入力などで、使用されている API からの元の名前を保持してください。
- 統合する製品で使用されているエラー/情報/成功メッセージを変更しないでください。
- オペレーションで使用されている API バージョンをフォルダーレベルで表示するようにしてください。できれば、API の場所も表示してください。

### 封印されているオペレーションではなくステップをコピー

封印されているオペレーション (Operations フォルダー内のオペレーションなど) はコピーしないでください。 代わりに、 封印されたオペレーションから作成したステップを変更するようにします。

## 値をフロー変数に割り当てる

デフォルトでは、オペレーションは、特定のフローで繰り返し使用されている入力に対してフロー変数 を使用および設定します。たとえば、フロー内の複数オペレーションで、ホスト、ユーザー名、パスワードの入力を行って、サーバーまたはメールサーバーのポートから情報を入手する必要がある場合があります。これらの値を、このようなデータを必要とするさまざまなステップで使用されるフロー変数に割り 当てると、フローのメンテナンスが簡単になり、異なる状況にも簡単に適応できるようになります。

それに対して、電子メールの件名の行は、電子メールの件名の行を必要とするステップごとに異なる可能性があります。したがって、件名の行を、フロー変数で提供するのは良い方法ではないと言えます。

#### 複数のオペレーションで同じコマンドを実行するのを避ける

同じコマンドを実行する複数のオペレーションを作成しないようにします。 たとえば、パケットロスと最大 遅延の両方を ping オペレーションから取得できます。 ping コマンドを使用する複数のオペレーションを 作成するのではなく、両方の情報を、1つの ping オペレーションの複数出力を使用して1つのステッ プで取得する方法の方がベターです。

この考え方は、WMI コマンドを実行するオペレーションのような非常に汎用性の高いオペレーションの場合は例外です。非常に汎用的なWMI コマンドの入力と非常に汎用的な出力を持つ単一のオペレーションを作成するのではなく、特定の関数専用のWMI コマンドオペレーションを作成する方がお勧めです。

### スクリプトレットではなく結果フィルターを使用

コマンドの出力ストリームからデータを取得する場合、スクリプトレットではなく結果フィルターを使用する方がベターです。これには、次のような理由があります。

- スクリプトレットは [スクリプトレット] タブに個別に表示されますが、結果フィルターは、[結果] タブエ ディターでアクセスしすぐに表示できます。
- スクリプトレットの方が、プログラマーでない人にとっては管理が難しくなります。
- オペレーションの結果のいずれかが削除されると、結果フィルターは自動的に無効になります。作成者が、スクリプトレットが加工する結果を削除した後に削除し忘れたスクリプトレットはそのまま残り、フローのバグとなることがあります。
- 結果データの目的の処理を行うスクリプトレットが必要な場合、スクリプトレットフィルターを使用できます。

#### 必要なレスポンスのみを使用

ほとんどのオペレーションには、成功と失敗という、2つのレスポンスのみがあります。レスポンスの数を 少なくすると、フローの作成が簡単になり、分かりやすくなります。異なるタイプの失敗を基にした複数 のレスポンスは、明らかに異なるパスに進む場合や、結果がその状況のために失敗にしかならない場 合 (HTTP Get のリダイレクトレスポンスなど)でのみ使用してください。 ただし、この原理は、意味がない場合には当てはまりません。たとえば、データを取得してしきい値を チェックするオペレーションでは、失敗、しきい値越え、しきい値未満の3つのレスポンスが必要な場合があります (どれも成功ではないレスポンス)。

### デフォルトレスポンスは失敗

オペレーションのデフォルトのレスポンスは失敗にします。そうすれば、フローのデバッグ時に、不完全なオペレーションは失敗になり、フローが実稼働する前に問題を作成者に示すことができます。

## ステップのベスト プラクティス

#### 説明なし

通常、ステップに説明は不要です。ステップのレスポンスのトランジションの説明で、ステップで発生する 内容が説明されているためです。

#### コールアウト

コールアウトを使用してステップに関する情報を追加することもできます。コールアウトを使用すると、フローの使いやすさが向上します。

#### 開始ステップ

開始ステップはフローの左上隅に配置するべきです。ただし、次の場合は例外です。

- 開始ステップに複数のレスポンスがあり、それぞれが別のステップに進む場合。
- 開始ステップを左上隅に配置すると、トランジションが交差するなどして表示が複雑になりすぎる 場合。

#### リターンステップの名前の変更

フロー内の同じタイプのリターンステップが複数ある場合、リターンステップの名前を変更して原因を含めます。たとえば、Error: failure やError: threshold not met などになります。

#### 失敗と否定の結果とを区別する

オペレーションの失敗と否定の結果を混同しないようにします。たとえば、オペレーションで、答えが TRUE または FALSE のどちらかになる質問をした場合、FALSE の答えはオペレーションの失敗と同 じではありません。このような場合、結果が FALSE の場合とオペレーションの失敗の場合の、2つのエ ラーで結果を返す必要があります。



## 操作なしと情報の収集の成功とを区別する

次の状態を混同しないようにしてください。

- ・ 操作なし

   ○は、修復フローがデータを収集したが、診断または修復を決定できなかった場合に
   使用されます。
- データの収集のみを目的とするフローまたはオペレーションは、完了時に操作なし ではなく解 決済み ② を返すようにします。

## 結果を使用して値をフロー変数に割り当てる

フロー変数を情報に割り当てるには、ステップの[結果]タブを使用します。結果をフィルターすると、ステップ結果からのデータの取得が柔軟に行えるようになります。

#### フロー変数への注意

フロー変数には、フロー全体からアクセスできます。いずれかのステップでデータが誤って変更されると、 フローの後続のステップでそのデータの間違った値が使用されるため、フロー変数の操作に注意してく ださい。

#### 不要な結果を追加しない

ステップのベースとなるオペレーションまたはフローでは、複数の出力を提供することができます。ただし、ステップの結果を追加するときに、フローに必要な出力のみ使用していることを確認します。結果が多すぎると、不要なデータによりフロー速度が低下し、パフォーマンスに影響を与えます。

### エラーコードを取得する結果の作成

ステップまたはトランジションで、オペレーションから得られるそのままのエラーが必要な場合、エラーコードを取得するステップ結果を作成し、エラーコードをフロー変数に割り当てます。

## トランジションのベストプラクティス

トランジションの整理と簡潔さの保持

- 可能な限り、トランジション線が交差しないようにします。
- 可能な場合、直線のトランジションを使用します。曲線のトランジションは、フローのレイアウトに必要な場合のみ使用します。
- 可能な場合、トランジションが水平、垂直、または45度の斜線になるようにステップを配置します。
- 2つのステップ間の複数のトランジションを1つにまとめて、1本の線ですべてのトランジションを表す ようにします。
- トランジションのラベルがステップのラベルと重なったり、トランジションのラベル同士が重なったりしないように、トランジションのラベルを配置します。
- 可能な場合、トランジションのラベルをフローの外側に寄せて配置します。たとえば、2つのステップがフローキャンバスの上部にある場合、トランジションのラベルをトランジション線の上に配置します。ステップがキャンバスの下部にある場合、ラベルをトランジション線の下に配置します。

### グリッドの使用

グリッドを有効にして、ステップの整列を保持します。

## 入力のベスト プラクティス

### 不要な入力の削除

ステップ内のオプションの入力が必須でない場合、これらの入力を削除します。

#### オペレーションで必要な入力を無効にしない

オペレーションの必須の入力は無効にしないようにします。無効にすると、そのオペレーションのすべてのインスタンスで無効になるためです。個々のステップを編集する必要があります。

#### 順序ルールに従った入力の追加

順序ルールに従って、一貫した順序で入力を追加します。次に例を示します。

- 分かりやすい分類または論理的な分類に基づく順序
- 重要度に基づく順序(最初に必須の入力)
- アルファベット順

**注**:入力が別の入力を使用する場合、正しい順序で指定する必要があります。たとえば、次のような場合:

input\_a="The first input"

input\_b=\$(input\_a)+

input\_a を input\_b の前 に 定 義 してください。

## データをフロー入力に割り当てる

理想的には、フローステップで使用される入力値は、フロー入力により提供され、フロー変数によりス テップに渡されるようにします。

通常、フローの作成者は、ユーザーがフローを開始してから、そのフローを実行している間に別のタ スクを開始するということを前提にしてください。フロー入力に、できるだけ多くのデータを割り当てると、 フローへの変更を簡略化できます。

## デバッグのベストプラクティス

#### サブフローのデバッグ

親フローをデバッグする前に、サブフローをデバッグすることをお勧めします。

## Studio の構成のベスト プラクティス

### システムプロパティ

システムプロパティの作成または値の変更は慎重に行ってください。システムプロパティの適用範囲が グローバルであり、実行時に、フロー実行のコンテキストの一部となるためです。結果として、システム プロパティの値を変更すると、既存のオペレーションとフローが壊れる恐れがあります。

#### 構成アイテム

構成アイテム(システムフィルター、スクリプトレット、選択リストなど)を削除する前に、[これを使用する 項目は?]機能を使用して、ほかのアイテムが依存していないかをチェックすることをお勧めします。詳 細については、「フローとオペレーションの使用方法の詳細」(240ページ)を参照してください。

## 説明のベストプラクティス

#### すべての説明

- [説明] タブを使用して、オペレーション、ステップ、フローの目的の詳細な説明を入力し、ほかの作成者が、その目的を知り、どうしてこのように設計したかがわかるようにします。
- 説明には、アイテムを検索するときに役立つ検索語を含めるようにしてください。また、入力、レスポンス、出力についての説明を含めてください。
- もっと長い説明が必要な場合、各段落を1行空けて区切ります。
- 段落の最初にスペースは追加しないでください。
- 文は短く簡潔にします。
- 長いフレーズを使うことは避けます。代わりに、長いフレーズは短い文に分けます。
- 文はすべて、大文字から開始し、ピリオドで終わるようにします(英語の場合)。
- 現在形、能動態を使用します。たとえば、"If the input is not specified, it will default to 1534" ではなく、'If you do not specify a value for this input, it defaults to 1534" と記述します(英語の場合)。

### フォルダーの説明

同じテクノロジーと情報をやり取りする複数のフローまたはオペレーションを作成する場合、1つのフォ ルダーにまとめて、フォルダーの[説明] エリアにこの情報を記載します。[説明] 領域にアクセスするに は、フォルダーを右クリックして [プロパティ] を選択します。

#### フローの説明

- フローを自動的に実行するために必要な特殊な要件または変更 (スケジュールどおりなど)を含めます。
- 次のような、フローの使用に関する制限を含めます。

このフローは以下の場合にのみ動作します。

- Windows 2003 またはそれ以降
- Windows Telnet Service が有効である場合
- 必要なフローの入力を指定し、入力が必要なデータや適切なデータ形式を作成者が検索できる場所についての情報を含めます。

- フローのレスポンスと、各レスポンスの意味を含めます。
- 結果フィールドと、各結果フィールドで与えられるデータの説明を含めます。
- 次のように、実装に関する追加のメモがあれば含めます。
  - サポートされているプラットフォームまたはアプリケーションのサポート (バージョン情報を含む)。
  - フローが相互作用するアプリケーションまたは Web サービス API。
- 問題選別、診断、修復を行うフローは、最初に、問題が存在しているか確認を行います。
- ユーザーに通知を送信するフローでは、通知サブフローを使用し、フローの作成者がユーザーへの 複数の通知手段を選択できるようにします。たとえば、Web サイトヘルスチェックフローでは通知 サブフローを使用します。ユーザーがWeb サイトヘルスチェックフローを電子メールおよびチケットシ ステムに設定すると、このフローを使用するすべてのフローで、正しく通知が送信されます。
- サブフローの作成では、最初に説明を追加してください。フローの目的と入出力を説明することにより、フローデザインが明確になるので、実装が簡単になります。

最上位の親フロー内のすべてのトランジションに注釈を付けます(トランジションの説明)。これらのトラ ンジションの説明は、トランジションの先行ステップで発生する内容について説明します。

#### オペレーションの説明

- オペレーションの内容の説明を含めます。
- オペレーションが必要とする入力を列記します。入力に必要なデータや適切なデータ形式を作成 者が検索できる場所も含めます。
- 各レスポンスの名前とその意味を含めます。
- 結果フィールドと、各結果フィールドで与えられるデータの説明を含めます。
- 次のように、実装に関する追加のメモがあれば含めます。
- サポートされているプラットフォームまたはアプリケーションのサポート (バージョン情報を含む)。
- フローが相互作用するアプリケーションまたは Web サービス API。
- その他の環境要件または使用要件。
- オペレーションの説明のベースとして、次のテンプレートを使用してください。

オペレーションの内容の説明。 入力: 入力 1 - この入力の詳細 入力 2 - この入力の詳細
入力 3 - この入力の詳細
レスポンス:
レスポンス 1 - このレスポンスの詳細
レスポンス 2 - このレスポンスの詳細
結果:
フロー/オペレーションのプライマリ結果
その他の結果:
結果 1 - 最初の追加の結果
結果 2 - 2 番目の追加の結果

## 入力の説明

- 必要な入力に対する意味がある説明を含めます。
- 入力に必要なサンプルデータを正しい形式で含めます。
- オプション入力と必須入力を区別します。
- 説明で列記する入力の順番が、[入力]タブに表示される順番と同じであることを確認してください。
- 必要な場合、入力がとるデータの構文を記述します。
- 説明中のセクション見出しの構文は同じものを使用します(たとえば、"Example"や"ex:"ではなく "Examples"を使用するなど。英語の場合)。
  - セクション見出しの形式は、「値の形式」、「例」、「デフォルト値」、および「有効な値」です。これらは、該当する情報があり、使用可能な場合はいつでも必要です。
  - ■「値の形式」、「例」、「デフォルト値」、および「有効な値」からの値で、句読点のみがふくまれる 値は、アポストロフィで囲む(','、'.')必要があります。
- 各入力名の前に4つスペースを置きます。
- 入力の説明のベースとして、次のテンプレートを使用してください。

入力:

inputName - 最初の入力の名前

```
値の形式: domain\user
有効な値: us.east.1a、us.east.1b
デフォルト値: us.east.1b、','
例: valueString、31241423、one、two、three
secondInputName - 2 番目の入力の名前
値の形式: text
有効な値: one、two、three
デフォルト値: one
例:
```

### トランジションの説明

- 最上位の親フロー内のすべてのトランジションに説明を付けます。これらのトランジションの説明 は、トランジションの先行ステップで発生する内容について説明します。
- データが実行中に必ず表示されなければならない場合を除き、サブフローのトランジションの説明 を追加する必要はありません。

#### 出力の説明

- 説明で列記する出力の順番が、[出力]タブに表示される順番と同じであることを確認してください。
- 各出力名の前に4つスペースを置きます。
- オペレーションが実行可能な状況を制限するオペレーションの環境的な制限がある場合は含めます。

#### 結果の説明

- プライマリ出力は、[結果] リストの一番最初の結果にし、次のテキストを含めるようにしてください。「これはプライマリ出力です。」
- 見やすくするため、リストの項目と表の列は、別々の行に記載します。
- 各結果名の前に4つスペースを置きます。
- 結果の説明のベースとして、次のテンプレートを使用してください。

結果:

```
returnResult - これはプライマリ出力です。
firstResult - 次の列を持つテーブル内の EC2 インスタンス: (以下に
テーブル形式の結果の例を表示します。)
Instance Id
AMI ID Machine type
Status - 可能な値: 「パブリック DNS」、「キーの組の名前」、「RAM ディスク ID」
```

### レスポンスの説明

- 説明では、「レスポンス」という言葉を使用します。
- 最も良く使用されるレスポンスである "success" と "failure" では、次のフレーズを使用することをお 勧めします。
  - success 「オペレーションは正常に完了しました」ではなく「説明に記されているようにオペレーションが完了しました」を使用する。
  - failure -「何らかの問題が発生しました」ではなく「オペレーションは異常終了しました。トラブルシューティング情報については、注を参照してください」を使用する。
- 各レスポンス名の前に4つスペースを置きます。
- レスポンスの説明のベースとして、次のテンプレートを使用してください。

```
レスポンス:
```

```
success - 説明に記されているようにオペレーションが完了しました。
```

```
failure - オペレーションは異常終了しました。トラブルシューティング情報については、注を参照してください。
```

```
no more values -
```

## 注の説明

- •「注」のセクションは空白にしておくこともできます。
- より複雑なオペレーションでは、より多くの情報を記載する必要があるため、「注」セクションをサブ セクションに追加して、簡単に読めてわかるようにしておきます。
- 構成、設定、環境設定などの、オペレーションが正常に動作するために必要なすべてを「前提条件」サブセクションを使用します。
- "Extra settings" サブセクションを使用して、オペレーションやフローの特別なユースケース(セキュリ ティやネットワーク設定など)をすべて一覧にします。このセクションは、複数設定などの特殊な ユースケースのあるオペレーションで便利です。
- エラーの内容がわかるメッセージがないエラーの説明には "Troubleshooting" サブセクションを使用 して、考えられる解決策を提供します。
- 上記のいずれのセクションにも合わない情報は、"Other" サブセクションで説明します。
- •「注」のセクションのサブセクションの順番は、表示されたままにしておきます。
- これらのセクションはオプションです。"Other" サブセクションの見出しのみを含める必要がある場合、サブセクションの見出しは含めず、問題の通し番号だけを保持しておきます。つまり、見出しが "Notes" のままで、"Other" セクションを "Notes" に含めます。
- 「注」のセクションが複数のオペレーションまたはフローに適用される場合、注はフォルダーレベルに 配置してください。個々のオペレーションまたはフローで、フォルダーレベルで注を参照します。たと えば、「FolderName フォルダー内の「注」セクションを参照してください」などのように記述します。
- 注の説明のベースとして、次のテンプレートを使用してください。

前提条件:

- 1. 最初の前提条件
- 最初の前提条件の説明
  - 1.1. 最初のステップを実行します。(4 スペース)
  - 1.2. 次のいずれかのコメントを入力します: (4 スペース)
    - 最初のコメント。(11 スペース)
    - 2 番目のコメント。(11 スペース)
- 2.2 番目の前提条件
- 2 番目の前提条件の説明

その他の設定:

- 1. abc
- 2. def

トラブルシューティング:

- 1. abc
- 2. def

Studio オーサリングガイド

その他:

1. abc

2. def

#### データベースの可用性の取得グループを完全に説明する例

データベース可用性グループ (DAG) のメンバーであるサーバーのリストを取得します。これはまた、DAG に関する次のようなリアルタイムステータス情報の取得にも使用できます: PrimaryActiveManager 、OperationalServers、ReplicationPort、NetworkNames、StartedMailboxServers、Stop pedMailboxServers など。

入力:

host - 接続先の Exchange 2010 サーバーホスト。

username - サーバーへの接続時に使用するユーザー名。

password - サーバーへの接続時に使用するパスワード。

authType - ユーザーの資格情報の認証に使用されるメカニズムを指定します。

有効な値: Default、Basic、Credssp、Digest、Kerberos、Negotiate、NegotiateWithImpl icitCredential。

デフォルト 値: Default。

dagName - 停止する DAG の名前。

例: oodag。

delimiter - 結果でプロパティを区切るために使用される区切り文字。

デフォルト値: ','。

#### 結果:

returnResult - これはプライマリ出力です。DAG プロパティのリストを返します。各プロパティは 1 行に記述され、プロパティ名と値の間は

区切り文字で区切られます。

servers - DAG のメンバーであるサーバーのリスト。

operationalServers - DAG からの動作中サーバーのリスト。

primaryActiveManager - クラスターコアリソースグループを所有するノードを表します。

distinguishedName - Active Directory  $this on DAG \ on DN_{\circ}$ 

isValid - DAG が有効かどうか。

レスポンス:

success - DAG プロパティのリストが Exchange サーバーから取得されました。

failure - DAG 情報を取得できませんでした。サーバーに接続できません (資格情報に誤り があるか、authType がサポートされていない)。

クラスターが使用不可であり、クラスターサービスが動作していません。

注:

1. Powershell リモート接続に関する情報については、"Exchange 2010" フォルダーの説明を参照してください。

2. サポートされているバージョン: 2010。

## ソースコント ロール管 理 のベスト プラクティス

- ・ ほかの作成者との競合を防ぐため、ファイルで作業する前に必ずファイルをロックします。作業を開始してからコミットする前にファイルをロックするのではなく、ファイルをロックしてから編集するようにします。ファイルで作業している間に、ほかの作成者がファイルの編集を開始している可能性があります。一部のソースコントロール管理ツールでは、ファイルをロックしようとするまで、そのファイルがロックされていることが表示されません。
- アイテムを更新してから、アイテムをロック、編集、およびコミットするようにします。
- フォルダーをロックすると、フォルダー内のファイルもロックされます。
- 以下のアイテムのいずれかを編集したら、ツリーの一部をコミットできる場合でも、プロジェクト全体をコミットすることをお勧めします。
  - フロー、オペレーション、または構成アイテムの説明
  - 入力プロンプトのメッセージ ([プロンプト] が選択されている場合)
  - ステップのプロンプトのメッセージ([表示]タブ内)
  - @Action とIAction はプロジェクトに格納されません。作成者のStudio リポジトリに格納されて、これらのアクションを使用するオペレーションとの間でリンクされます。別の作成者が作成した@Action または IActionを使用するには、そのJar/DLLファイルを自身のStudio リポジトリに手動でコピーする必要があります。@Action および IActionを作成したら、ほかの作成者がインポートできるように、プラグインが含まれたコンテンツパックを作成することをお勧めします。

# HP 00 Studio での各言語の使用 - ローカライズ

ローカライズとは、ネイティブ以外の環境に合わせてソフトウェアを適合させることを指します。HP Operations Orchestration Studio は、次の言語へのローカライズが可能です。

- fr フランス語
- de ドイツ語
- ja 日本語
- es スペイン語
- zh 中国語

Studio では、次のテキスト文字列がローカライズされます。

- フローの説明とコールアウト
- トランジションの説明
- 構成アイテムの説明
- ステップの説明とプロンプトメッセージ([表示]タブ内)
- 入力プロンプト
- フォルダーの説明

### Studio での現在のロケールの変更

デフォルトでは、Studio はコンピューターのデフォルトの言語でインストールされます。 現在のロケールをオーバーライドするには、構成を変更します。

- 1. Studio\conf\studio.properties ファイルを開きます。
- 2. 次の行を検索します。
  - user.language=
  - user.region=
- 3. 上記の各行に、言語のロケールを追加します。使用可能なコードは次のとおりです。

言語のロケールコード	地域 (オプション)	言語
fr		フランス語
de		ドイツ語
ja		日本語
es		スペイン語
zh	CN	中国語

**注**:別の言語インタフェースを実装するには、user.language フィールドとuser.region フィールド (該当する場合)の両方に同じ言語のロケールを割り当てる必要があります。値が無効な場合 や空白の場合、Studio はデフォルトのシステムロケールを使用します。

## Studioの表示言語

Studio の表示言語には、user.language プロパティが有効であればその値 (user.region にも有効な値が必要)、無効であればシステムロケールが使用されます。

現在 Studio でサポートされている表示言語は次のとおりです。

言語	言語コード
フランス語	fr
ドイツ語	de
日本語	ja
スペイン語	es
中国語	zh

さらに、コンテンツパックのローカライズも可能です。詳細については、「コンテンツパックのローカライズ」 (41ページ)を参照してください。

## コンテンツパックのローカライズ

コンテンツパックのフローとオペレーションに含まれるテキスト文字列はすべて Studio の現在のロケール で表示されます。

ローカライズファイルは、cp\_<ロケール>.{<地域>.}propertiesという名前で保存されます。

(地域のパラメーターはオプションです)。

たとえば、コンテンツパックのデフォルトのプロパティファイルの名前は cp.properties、中国語のファイルの名前は cp\_zh\_CN.properties となります。

## プロジェクト のローカライズ

プロジェクトにはプロパティファイルが1つのみあり、プロジェクトをコンテンツパックにパッケージする際にデフォルトファイル(cp.properties)として保存されます。

次の2つの点に注意してください。

プロジェクトのテキストは実際にローカライズされるわけではありません。このテキストの読みとりと書き込みには、同じ cp.properties ファイルが使用されます。

 コンテンツパックを展開し、プロジェクトとして Studio にロードすると、デフォルトファイルのみが使用され、ローカライズしたプロパティファイルは使用されません。ファイルが存在しない場合は作成され、 すべての文字列は空白になります。

注:別の言語を使用する場合は、言語ごとにプロジェクトを作成することをお勧めします。

# 新しいコンテンツパックの作成

HP OO Studio でコンテンツパックをプロジェクトから作成すると、リソースバンドルフォルダーには cp.properties ファイルのみが格納されます。

注:コンテンツパックの構成を変更した後、Studioを再起動してください。

プロジェクトを作成して保存すると、cp.properties ファイルは \resource-bundles\cp.properties に あるプロジェクトファイルシステムに保存されます。

また resource-bundles ディレクトリには、ユーザーが作成したリソースファイルを追加することができます。追加したファイルは、ほかのリソースファイルと一緒にコンテンツパックの jar ファイルにバンド ルされます。

# プロジェクトの使用

プロジェクトは、業務で使用するフロー、オペレーション、フォルダー、選択リストなどのアイテムを含む 論理的な単位です。 プロジェクトは、1人の作成者によって、または共同作業プロジェクトとして開発 可能です。

Studio のプロジェクトは、オペレーティングシステムと同様のファイルディレクトリ構造を使用し、ソースコントロールアプリケーションと容易に統合できます。

# プロジェクトの管理



HP OO Studio でフローの作成を開始するには、プロジェクトで作業している必要があります。

**注**: プロジェクトの作成が、必ずしもワークフローの最初のステップである必要はありません。プロ ジェクトを作成する前に、コンテンツパックをインポートできます。

Studio 10.00 では、プロジェクトは作成者のファイルシステムにローカルに格納されます。 ソースコント ロール管理機能を使用するには、外部のソースコントロール管理ツールを使用する必要があります。 ソースコントロール管理の詳細については、「ソースコントロールの使用」を参照してください。

# 実行する操作

### ワークスペースを設定する

ワークスペースは、プロジェクトを格納するために作成するディレクトリです。 プロジェクトの作成時に、 作成したプロジェクトを Workspace ディレクトリに格納します。

このステップはオプションです。Workspace フォルダーを設定していない場合、プロジェクトの作成時に、Studioによってユーザーフォルダー内のデフォルトの場所にフォルダーが作成されます。

- 1. ソースコントロール管理ツールを使用している場合、共有リポジトリに Workspace フォルダーを 作成します。詳細については、「HP OO Studio のソースコントロールの使用」(52ページ)を参照 してください。
- 2. ファイル Users\[ユーザー名]\.oo\10.00\opstudiorc.xml をテキストエディターで開きます。
- 3. テキストに <entry key="workspace" value="C:\Users\[ユーザー名]\.oo\Workspace">と 表示されます。値を、Workspace フォルダーを配置するパスに置き換えます。

このステップで、新しいプロジェクトの作成時にこの場所をデフォルトで使用するように Studio に指定します。

- 4. Studio を1回以上開いて、Workspace フォルダーを作成します。
- 5. ソースコントロール管理ツールを使用している場合、ローカルの Workspace フォルダーに対して 共有の Workspace フォルダーをチェックアウトします。詳細については、「HP OO Studio のソース コントロールの使用」(52ページ)を参照してください。

#### プロジェクトを作成する

1. [ファイル] > [プロジェクトの新規作成]を選択します。

注: [プロジェクト] ペインで [プロジェクトの新規作成] 🕈 ボタンをクリックすることもできます。

- 2. [名前] ボックスにプロジェクトの名前を入力します。
- 3. [プロジェクトの作成] ダイアログボックスで、[場所] ボックス内のパスが前のタスクで設定した Workspace フォルダーへの正しいパスになっていることを確認します。パスが [Workspace] フォル ダーへの正しいパスでない場合、Workspace フォルダーを参照して選択します。
- 4. [OK] をクリックします。
- 5. ソースコントロール管理ツールを使用している場合、共有リポジトリにプロジェクトフォルダーを追加します。詳細については、「HP OO Studio のソースコントロールの使用」(52ページ)を参照してください。

#### プロジェクトをインポートする

別の場所に作成したプロジェクトを開くには、このプロジェクトを Studio にインポートする必要があります。 インポートすると、プロジェクトが [**プロジェクト**] ペインに表示されます。

ソースコントロール管理ツールを使用している場合、プロジェクトフォルダーをチェックアウトして、ローカ ルの作業コピーを作成してから、プロジェクトを Studio にインポートすることが必要になる場合がありま す。詳細については、「HP OO Studio のソースコントロールの使用」(52ページ)を参照してください。

1. [ファイル] > [プロジェクトのインポート]を選択します。

注: [プロジェクト] ペインで [プロジェクトのインポート] 🥙 ボタンをクリックすることもできます。

- 2. [プロジェクトディレクトリの選択]ダイアログボックスで、インポートするプロジェクトを参照して見つけます。
- 3. [OK] をクリックします。 プロジェクトが Studio の [プロジェクト] ペインに表示されます。

#### 開いたプロジェクトを閉じる

プロジェクトを閉じると、閉じたプロジェクトは [プロジェクト] ペインに表示されますが、 グレー表示されて 使用できなくなります。

- 1. 閉じるプロジェクトを選択します。
- 2. [**ファイル**] > [プロジェクトを閉じる] を選 択します。

注: [プロジェクト] ペインで [閉じる] 🧖 ボタンをクリックすることもできます。

#### 閉じたプロジェクトを開く

プロジェクトを閉じた後で、このプロジェクトを開いて再度作業することができます。

- 1. 開く必要がある、閉じた (グレー表示の) プロジェクトを選択します。
- 2. [**ファイル**] > [**プロジェクトを開く**]を選択します。

注: [プロジェクト] ペインで [開く] 🍼 ボタンをクリックすることもできます。

#### プロジェクトを削除する

プロジェクトを削除すると、プロジェクトを閉じる場合とは異なり、削除したプロジェクトはワークスペース から完全に削除されます。

**注**: プロジェクトを削除した場合、ワークスペースから削除されますが、ファイルシステムからは削除されません。必要に応じて、削除したプロジェクトを再度インポートできます。

- 1. 削除するプロジェクトを選択します。
- 2. [ファイル] > [プロジェクトの削除]を選択します。

注: [プロジェクト] ペインで [削除] 🎽 ボタンをクリックすることもできます。

3. 確認のダイアログボックスで [はい] をクリックします。

#### コンテンツパックのフローとオペレーションをプロジェクトにコピーする

[コンテンツパック] ペインのフローとオペレーションは読み取り専用です。これらのフローとオペレーションの 編集可能なコピーを作成するには、これらをプロジェクトにコピーします。

- 1. [コンテンツパック] ペインで、コピーするフローまたはオペレーションを選択します。
- 2. [編集] > [コピー]を選択します。
- 3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集] > [貼り付け]を選択します。フロー またはオペレーションは新しいオブジェクトとして扱われ、コピー元となったコンテンツパックからデタッ チされます。

**注**: フローを構成しているオペレーションもプロジェクトにコピーするには、[編集] > [ディープコ ピー]を選択します。詳細については、「フローおよびオペレーションのコピー」(237ページ)を 参照してください。

### プロジェクト のプロパティを表示する

プロジェクトを右 クリックして [プロパティ] を選択します。 [プロパティ] ウィンド ウにプロジェクトの情報が表示されます。

#### プロジェクト内のオフジェクトのプロパティを表示する

- [プロジェクト] ペインで、フロー、オペレーション、またはその他のオブジェクトをダブルクリックします。オ ブジェクトの [プロパティ] ウィンド ウが開きます。
- フローが作成キャンバスで開いている場合、フローの[プロパティ] ウィンドウを表示するには、作成 キャンバスの左下にある[プロパティ] タブをクリックします。

## 参照情報

[ファイル] メニュー

ファイル 編集 ツール SCM ウィ	ンドウーヘルプー	_
プロジェクトの新規作成		¢2€ × 🐼 ∗Restart
新規作成	•	フォルダー
プロジェクトのインポート		70-
プラグインのインポート		オペレーション
保存	Ctrl+S	
すべて保存	Ctrl+Shift+S	
閉じる	Ctrl+W	-
すべて閉じる		
最近開いた項目	•	
コンテンツパックの作成		
プロジェクトを閉じる		-
プロジェクトを開く		
プロジェクトの削除		
すべてのプロジェクトのリフレッシュ		
終了	Ctrl+Q	

メニューアイテム	説明
プロジェクトの新規作成	新しいプロジェクトを作成します。

新規作成	新しいファイル、フォルダー、またはオペレーションを作成します。
プロジェクトのインポート	別 のワークスペースの既存 のプロジェクトを参 照してインポート します。
プラグインのインポート	新しいオペレーションのベースとして使用する、アクションプラグイ ンを参照してインポートします。「オペレーションの作成」(225 ページ)を参照してください。
保存	選択された要素を保存します。
すべて保存	更新されたすべての要素を保存します。
閉じる	現在選択されているフローまたは [プロパティ] ウィンドウを閉じ ます。
すべて閉じる	開いているフローまたは [プロパティ] ウィンド ウをすべて閉じま す。
最近開いた項目	最近開いた項目を一覧表示します。これには、作成キャンバ スで開かれたフローとオペレーション、最近表示された [プロパ ティ] ウィンドウが含まれます。
コンテンツパックの作成	選択したプロジェクトのコンテンツパックを作成します。
プロジェクトを閉じる	現在選択されているプロジェクトを閉じて、グレー表示にしま す。
プロジェクトを開く	現在選択されている、閉じたプロジェクトを開きます。
プロジェクトの削除	選択したプロジェクトをワークスペースから完全に削除します。
終了	HP OO を終 了します。

## [プロジェクトの作成] ダイアログボックス

🕵 プロジェクトの作成		×
2 <del>2</del> 2		
治肌门		
場所: C:¥Users¥Administrator¥.oo¥Workspace		
	OK	キャンセル

GUI ፖイテム	説明
名前	新しいプロジェクトの名前を入力します。

場所	新しいプロジェクトを格納する、ワークスペースの場所を入力し ます。
	• 場所へのパスを入力します。
	• 場所を参照するには、参照ボタン・・・・ をクリックします。

### [プロジェクト] ペイン

[プロジェクト] ペインには、プロジェクトツリー (プロジェクトの編集可能なコンテンツが含まれる階層フォルダー構造)が表示されます。

- Library フォルダーには、フローとオペレーションが含まれます。
- Configuration フォルダーには、オペレーション結果の処理、レポートの作成、およびフローの実行 を容易にするために使用できるその他のHP OO オブジェクト (フィルター、スクリプトレット、システム プロパティなど)が格納されています。



GUI ፖイテム	説明
プロジェクトの新規作成 🔸	新しいプロジェクトを作成します。
プロジェクトのインポート 🤌	別 のワークスペースの既存 のプロジェクトを参 照してインポート します。

コンテンツパックの作成 🥠	選択したプロジェクトのコンテンツパックを作成します。
削除 🔀	選択したプロジェクトをワークスペースから完全に削除します。
開く 🎨	現在選択されている、閉じたプロジェクトを開きます。
閉じる 🥬	現在選択されているプロジェクトを閉じて、グレー表示にしま す。
リフレッシュ 🎅	現在選択しているプロジェクトのファイルをリフレッシュします。

# 【プロジェクト】 ペインを使用したフォルダーの管理

[プロジェクト] ペインで、フォルダーの追加、削除、コピー、名前変更といった、プロジェクトのフォルダー 管理を行います。

### ベストプラクティス

ほかの作成者がフロー、オペレーション、およびユーティリティを見つけることができるように、フォルダー 構造が適切に定義されており、プロジェクト間で一貫性があるようにします。

[ディープコピー] コマンドを使用してフローをコピーする場合、フローとそのオペレーション用の新しいフォ ルダーを作成することをお勧めします。

フロー作成者向けにフォルダー構造およびその他のガイドラインを記述しているドキュメントを参照してください。

### 実行する操作

#### フォルダーを作成する

- 1. [ファイル] > [新規] > [フォルダー] を選択します。
- 2. 新しいフォルダーの名前を入力して、[OK]をクリックします。

注:名前の長さは最大 128 文字までです。大文字と小文字は区別されません。

#### フォルダー名を変更する

プロジェクトツリーでフォルダーを右クリックし、[名前の変更]を選択します。

#### フォルダーを削除する

プロジェクトツリーでフォルダーを右クリックし、[削除]を選択します。

### フォルダーをコピーして貼り付ける

- 1. プロジェクトツリーでフォルダーを右クリックし、[編集]>[コピー]を選択します。
- 2. プロジェクトツリーで、フォルダーを貼り付ける位置を右クリックし、[編集] > [貼り付け]を選択します。

#### サブフォルダーにドリルダウンする

展開ボタン 🗄 をクリックしてフォルダーを展開し、フォルダー内のサブフォルダーを表示します。

#### フォルダーの説明を表示する

プロジェクトツリーでフォルダーを右クリックし、[プロパティ]を選択します。フォルダーのプロパティが表示 されます。

# ソースコントロールの使用

# HP 00 Studio のソースコントロールとは

HP OO Studio では、組み込みのソースコントロール管理ツールが提供されており、作成者はローカル 環境で作業した後で変更をパブリックバージョンと同期させることができます。

#### 作業コピー

SCM リポジトリはバージョン付きのデータをすべてソースコントロールサーバーに保持します。Studio の ソースコントロール管理ツールは、バージョン付きデータのローカルコピー (作業コピー)を管理しま す。SCM は、ネットワークを介して自身のリポジトリにアクセスします。 リポジトリには、複数のユーザー が同時にアクセス可能です。

#### チェックアウト

チェックアウトは、リポジトリから作業コピーにソースをダウンロードするために使用します。ソースコント ロールサーバーからファイルにアクセスする場合に、最初に実行するオペレーションがチェックアウトで す。チェックアウトを実行すると、作業コピーが作成され、その内容を編集、削除、追加できるように なります。ファイル、ディレクトリ、トランク、またはプロジェクト全体をチェックアウトすることができます。 チェックアウトを実行するには、チェックアウト対象コンポーネントのソースコントロールサーバーの URL が必要です。

#### コミット:リポジトリへの変更の保存

ローカルの作業コピーに変更を加えても、ソースコントロールサーバーには自動的に保存されません。 変更を確定するには、その変更をコミットする必要があります。

#### 追加: SVN リポジトリへの新しいファイルの追加

追加コマンドでは、新しいファイルまたはディレクトリをリポジトリに追加します。 リポジトリには、変更内容をコミットした場合にのみ、新しく追加されたファイルが表示されます。

#### 削除:リポジトリからのファイルの削除

**削除**コマンドでは、作業コピー (リポジトリ) からアイテムを削除 できます。 コミット すると、 ファイルがリポ ジトリから削除されます。

#### 移動:ファイルまたはディレクトリの名前変更

移動コマンドでは、ディレクトリ間でのファイルの移動や、ファイル名の変更を行います。ファイルはローカルのサンドボックス上で (コミット実行後はリポジトリ上で)直ちに移動されます。

#### 更新:作業コピーの更新

更新コマンドでは、変更内容がリポジトリから作業コピーに表示されます。リポジトリ内にある最新の 変更内容を作業コピー内でも使用できるように、作業コピーを更新してから作業を開始することをお 勧めします。

# 初期のソースコントロールリポジトリの作成

Studio でソースコントロールの使用を開始する前に、初期のリポジトリをセットアップする必要があります。

注:社内にすでにソースコントロールリポジトリがある場合、このセクションは必要ありません。

SVN リポジトリを作成するには、

- 1. Windows で共有フォルダーを作成し、リポジトリの作業を行う作成者のために、フォルダーに読み取り/書き込みアクセス許可を割り当てます。
- 2. <Studio インストールフォルダー>\studio でコマンドライン (cmd)を開きます。
- 3. コマンドウィンドウに、createFileSystemScmRepository.bat および共有フォルダーへの完全パ スを入力します。

例:

```
c:\<Studio インストールフォルダー>\Studio>
createFileSystemScmRepository.bat SHARED FOLDER FULL PATH
```

次のように表示されます。

Repository was successfully created at <SHARED FOLDER FULL PATH>

これで、共有リポジトリフォルダーをStudioで使用する準備が完了です。

## HP 00 Studio のソースコントロールの使用

このセクションでは、ソースコントロール管理ツールで使用する一般的なタスクと、複数の作成者でプロジェクトやアイテムを共有する場合に作成者が直面する可能性のある高度なタスクについて説明します。

### Studio での SCM の使用

ローカルリポジトリを使用して Studio をインストールした後、Studio で作成を開始し、ローカル環境で ソースコントロールを使用できるようになります。 ここでは、SCM で実行する一般的なタスクについて 説明します。

#### ソースコントロールサーバーでの認証設定

SCM を使用する際の最初のステップとして、ソースコントロールサーバーでのユーザー認証の設定を 行います。 1. [SCM] メニューから[設定]を選択します。

SCM 設定	×
SCM 資格情報	
マ Windows 認証の使用	
ユーザー名	
パスワード	
▶ サーバー証明書を信頼	
	保存 キャンセル

使用するソースコントロールサーバーに応じて以下のオプションから1つ選択します。

- Windows 認証の使用: このオプションは、デフォルトで選択されています。現在ログインしているユーザーを使用して認証を実行します。ファイルベースのソースコントロールリポジトリに適用されます。
- [Windows 認証の使用]オプションの選択を解除します。ソースコントロールサーバーで定義されているユーザー名とパスワードを入力します。セキュリティで保護されたサーバー (SSL/SSH)を使用している場合は、[サーバー証明書を信頼]を選択します。選択しないと、サーバーにアクセスできません。

SCM 設定 X	]
SCM 資格情報	
IT Windows 認証の使用	
ユーザー名	
パスワード	
□ サーバー証明書を信頼	
保存 キャンセル	

注:資格情報は、SCM自体へのアクセスのみに使用されます。システムが簡易ネット ワークファイル共有をベースにしている場合、HP OO は共有可能なネットワーク接続をオ ペレーティングシステムが生成するとみなすので、HP OO はネットワーク接続用の資格情報を OS に渡しません。

#### リポジトリのチェックアウト

リポジトリには自分のプロジェクトがすべて含まれ、複数の作成者と共有することができます。 リポジト リは、 ファイルシステムまたは Web サーバー上 でホストすることができます。 ファイルシステムベースのリポ ジトリ作成の詳細については、「ローカルリポジトリの作成」を参照してください。

リポジトリは Studio から直接 チェックアウトし、プロジェクトを追加してほかの作成者と共有することができます。

すべてのプロジェクトをチェックアウトするには、次の手順を実行します。

1. [SCM] メニューから[チェックアウト] を選択します。

SCM <del>チェッ</del> クアウト	×
SCM リポジドリの URL を入力してください	
接続解释余	
	チェックアウト キャンセル

[**チェックアウト**]を初めて選択した場合、ローカルリポジトリを設定した際に作成した共有フォルダーのURLを入力するよう求められます。ローカルリポジトリを使用している場合は、ファイルの URL スキームを使用して URL を入力します。

例:

■ リポジトリフォルダーを共有していない場合は、以下を使用して接続します。

file:///c:/temp/reop

リポジトリを共有していて、ユーザーにすべての権限が与えられている場合は、以下を使用します。

file://myshared/repo

2. [**チェックアウト**] をクリックします。これにより、URL からファイルがチェックアウトされ、Studio のワーク スペースのディレクトリにチェックインされます。

[SCM メッセージ] ウィンド ウに共有フォルダーとファイルのチェックアウト 先の場所が表示されます。 チェックアウトリポジトリにすでにプロジェクトがある場合は、これらを Studio にインポートして作業 することができます。 次のステップは、新しいプロジェクトの作成または既存プロジェクトのインポートのどちらかになります。[プロジェクト] ペインで、サアイコンをクリックして新しいプロジェクトを追加するか、 <sup>3</sup> をクリックして既存プロジェクトをインポートするかのどちらかを行います。

注:新しいプロジェクトを作成すると、自動的にSCMに追加されます。

#### ワークスペースの SVN からのデタッチ

リポジトリをチェックアウトした後、Studioのワークスペースをソースコントロールサーバーからデタッチできます。 これは、間違った処理をして正しくないリポジトリをチェックアウトした場合に便利です。 デタッチ を行ってもリポジトリからは何も削除されません。

- 1. [SCM] メニューから[チェックアウト] を選択します。
- 2. [接続解除]をクリックします。
- 3. [キャンセル]をクリックしてウィンドウを閉じます。

## [SCM 変更] パネル

[SCM 変更] パネルには、作業コピーのリビジョンと比較して、作業コピーのすべての変更が表示されます。たとえば、フローを編集すると、[SCM 変更] パネルにそのフローが表示されます。また、このパネルには、削除済みのプロジェクト (削除対象としてマークされたプロジェクト) があった場合、これらの一覧が表示されます。

### 変更の種類

次のスクリーンショットは、以下のものを示します。

SCM 変更	0 <del>0</del> - +
👯 🍝 🍄	
<ul> <li>□-  Project1 (説明の変更を含む)</li> <li>□-  De Ibrary </li> <li>□-  flows </li> <li>□-  flows </li> </ul>	
operations     deprecated     o-base-plugin	
@ MathOp	
Deleted Projects - 1	

- flow という名前の削除済みフロー (グレーで表示)。
- deprecated という名前で追加されたフォルダー(緑で表示)。
- Project2 という名前のバージョンなしプロジェクト (茶で表示)。

- 1つの削除済みプロジェクト。
- また、変更されたアイテム MathOp (青で表示)が、ロックインジケーター付きで表示されています。
   これは、このアイテムはユーザーが編集中のためロックされていることを示します。

#### 名前変更および移動されたアイテム

名前変更および移動されたアイテムには、アイテムの移動元または変更前の名前を持つ特別なラベルが付けられます。以下の例では、UUID Generator は UUID Generator 2 に名前が変更され、 フォルダー oo-base-plugin が移動されています。

SCM 変更	0 <i>0</i> - +
scm scm 🍐 🏠	
Project1     Project1     Description     Description	y\operations

- 以下のアイテムのいずれかを変更した後、プロジェクト全体をコミットすることを推奨します。変更はアイテム自体では発生しませんが、プロパティファイル内で発生します。そのため、このような変更により、アイテムが変更されたとは示されませんが、プロジェクトが変更されたと示されます。この場合、説明の変更を含むという接尾辞がプロジェクト名に追加されます。
  - フロー、オペレーション、または構成アイテムの説明
  - 入力のプロンプトメッセージ ([プロンプト] が選択されている場合)
  - ステップのプロンプトメッセージ([表示]タブ内)

SCM 変更	00-+
🐺 🏠 🍐	
😕 Project1 (説明の変更を含む)	

#### SCM 変更ツールバー

scm	すべて更新: Studioのワークスペース全体を更新します。
•	



### 色分け

Studioでは、以下の色で各アイテムを示します。



- 黒:変更なしの通常アイテム(変更パネルでは利用不可)
- 緑:追加済み
- グレー: 削除済み(プロジェクトツリーでは利用不可)
- 青:変更済み
- 茶:バージョンなし
- 下波線:エラーあり

### 複数の作成者での作業

#### アイテムのロックとロック解除

複数の作成者が共通プロジェクトで作業する場合、2人の作成者が同じアイテムを同時に変更す る可能性があります。Studioでは、競合が発生しないようにこれらの変更をすべてマージするよう試み ます。アイテムをロックすることにより、ほかの作成者が同時にそのアイテムで作業できないようにするこ とができます。ほかのユーザーによりロックされているアイテムは編集可能ですが、コミットできません。更 新時の競合を防ぐため、編集前にアイテムをロックすることを強くお勧めします。アイテムをロックできな い場合は、競合を避けるためにそのアイテムを編集しないようにすることをお勧めします。

**注:** アイテムのロック、変更、コミットを行う前に必ず更新を実行してください。ロックされたアイテム への変更をコミットすると、自動的にロックが解除されます。

アイテムをロックするには、アイテムの編集ウィンドウの左上隅でロックアイテムをクリックします。



ロックが成功した場合、アイコンがロックに変わり、ロックされたファイルへの完全パスとユーザー詳細を 示す SCM メッセージが表示されます。



13/09/19 11:29:37 - C:¥Users¥Administrator¥. oo¥Workspace¥MyProject2¥Content¥Library¥My Ops Flows¥oo-base-plugin¥MathOp.xml をロックしています 'MathOp.xml' はユーザ 'Administrator' にロックされました。			
•	ロックが成功		<u> </u>

ロック解除は上記と同じ方法で行えます。アイコンが変わり、SCMのロック解除メッセージが表示されます。



アイテムがロックできない場合、警告(黄色)メッセージが表示され、アイコンは変わりません。

ロックされたアイテムへの変更をコミットすると、自動的にロックが解除されます。

ロックボタンは追加されていてまだコミットされていないアイテム、またはStudioのワークスペースにない プロジェクトのアイテムでは無効です。

## リビジョン履歴

HP OO Studio では、バージョン履歴管理も提供されています。[**リビジョン履歴**] ウィンド ウでは SCM 履歴が4 つの列で表示され、各コミットが1 行ずつ表示されます。

QJビジョン履歴: flow 🛛 🔍			
リビジョン	日時	作成者	אינאב 📃
3	13/09/19 09:47	Administrator	Changed success respons
2*	13/09/19 09:46	Administrator	Changed flow layout
1	13/09/19 09:44	Administrator	Add Project1
			ОК

- ・ リビジョン: コミットのリビジョン。ローカルコピーのリビジョンにはアスタリスクが表示されます。チェックアウト後、アスタリスクは常に最新のリビジョンを示しますが、別のユーザーが変更をコミットした後かつ更新前の場合は、アスタリスクが示すリビジョンは最新のものではありません。上記のスクリーンショットは、ローカルコピーがリビジョン2であるが、リビジョン3でファイルへのコミットが行われたことを示します。
- 日時:リビジョンがコミットされた日付と時刻。
- 作成者:リビジョンをコミットした作成者。ユーザー名が表示されます。
- コメント: コミット時にユーザーが追加したコメント。

履歴アイテムを右クリックすると、コンテキストメニューに以下のオプションが表示されます。

• 更新先リビジョン: 選択したリビジョンへファイルを更新します。 このオプションを使用すると、プロジェクトの一部のアイテムがほかと異なるリビジョンに入る場合があります。 以下の例で、アスタリスク(\*)の上にあるリビジョンを右クリックします。

🕓リビジョン履歴:	flow			×
リビジョン	8	<b>寺</b>	作成者	אלאב 🔤
3 2* 1	13/09/19 09:47 13/09/19 09:46 13/09/19 09:44	マ このリビジョン 彩響されるす	に戻す たてのファイルの表示	Changed success response Changed flow layout Add Project1

• このリビジョンに戻す: 選択したリビジョンと同じ方法でローカルでオブジェクトを変更します。たとえば、リビジョン 333 で 2 つのステップを含み、リビジョン 337 で 3 つのステップを含むフローがある場

合、これをリビジョン 333 に戻すと、含まれるステップは再度 2 つになります。ほかの作成者が変更 を見ることができるようにするには、フローをコミットする必要があります。以下の例で、アスタリスク (\*)の下にあるリビジョンを右クリックします。

🕵リビジョン履歴: flow 🛛 🗙				
リビジョン	日時		作成者	אעאב
3	13/09/19 09:47		Administrator	Changed success response
2*	13/09/19 09:46		Administrator	Changed flow layout
1	13/09/19 09:44		Administrator	Add Project1
	13	このリビジョンに	:戻す	
		影響されるす/	べてのファイルの表示	

• 影響されるすべてのファイルの表示:選択したリビジョンで変更されたファイルをすべて表示します。

**注**: リビジョンを更新または戻すオプションは、アイテムに変更が適用されていない場合にのみ有効になります。

## [SCM メッセージ] パネル

SCM 操作の結果メッセージが表示されます。 すべての操作の結果がメッセージで表示され、 このメッセージには、実行内容と成功したかどうかが含まれています。

パネルの任意の場所を右クリックし、[**すべてクリア**] ボタンをクリックすることで、メッセージパネルをいつで もクリアすることができます。

以下の例は、適用した操作に応じて [SCM メッセージ]パネルに表示されるメッセージを示します。

#### 新しいオペレーションの追加

次の例では、MathOpという新しいオペレーションが追加されました。

それぞれを追加した後、新しいアイテムが自動的に追加としてマークされます。次のSCM メッセージには、oo-base-plugin フォルダーとオペレーションのXML ファイルの MathOp.xml が追加としてマーク されています。



### アイテムのコミット

コミットすると、次のメッセージが表示されます。このメッセージには、前回追加されたアイテムが、サーバーに正常にコミットされたことが示されます。



### アイテムの変更

たとえば新しい入力の追加といった変更を MathOp に対して行った後、変更を元に戻します。この メッセージには、MathOp.xml への変更が元に戻されたことが示されます。



#### エラーメッセージ

エラーが表示される場合もあります。エラーは赤で表示されます。この例では、変更のコミットが試行 されましたが、ファイルが期限切れになっています。

SCM メッセージ	
13/09/19 11:33:25 - コミットしています C:¥Users¥Administrator¥.oo¥Workspace¥MyProject1¥Content¥Library¥Flows¥ (Conv. 1) xml	flow
svn: E200009: つミッドに失敗しました (詳しい理由は以下のとわりです): svn: E200009: 'C:¥Users¥Administrator¥.oo¥Workspace¥MyProject1¥Content¥Library¥Flows' is out of date	
1 個のエラーが発生しました。	-
	Þ

## 参照情報

### [ソースコントロール] メニュー

ワークスペース全体に影響するオペレーションが含まれます。

SCM	ウィンドウ	ヘルプ		
すべて更新				
3	すべてコミット			
3	すべて元に戻	す		
Э	Fェックアウト			
Ī	殳定…			

オプション	説明
すべて更新	ワークスペースを直接更新しますが、Studio にプ ロジェクトがインポートされることはありません。すで にインポートされているプロジェクトは、サーバーか ら受信した更新を反映します。このコマンドは ワークスペース全体を更新するため、ほかの作成 者がコミットした新しいプロジェクトが、更新中に 取得される場合があります。これらのプロジェクト は Studio にインポートして作業することができま す。
すべてコミット	アクティブなプロジェクトの変更をすべてコミットしま す。 [ <b>コミット</b> ] をクリックすると、コミットのコメントを追 加できます。
すべて元に戻す	アクティブなプロジェクトの変更をすべて元に戻し ます。
チェックアウト	リポジトリをチェックアウトします。
設定	ソースコントロールサーバーでの認証の実行方法 を変更できます。 Windows 認証:現在ログインしているユーザーを 使用して認証を実行します。ファイルベースの ソースコントロールリポジトリに適用されます。 [ユーザー名]と[パスワード]の認証:入力された
	ユーザー名とパスワードを使用して認証を実行し ます。
	サーバー証明書を信頼: セキュリティで保護され たサーバー (SSL/SSH)を使用している場合は、 このオプションを選択します。選択しないと、サー バーにアクセスできません。

### [プロジェクト] ペイン

作業中のプロジェクトと、編集可能なフロー、オペレーション、プロジェクトで使用可能なその他のHP OO オブジェクトが表示されます。

[プロジェクト] パネルのコンテキストメニュー

コンテキストメニューには、選択したオブジェクトで使用可能なアクションがその状態に応じてすべて表示されます。

コミットされたフローを右クリックした際に使用可能なオプションを以下に示します。この場合、[更新] および [履歴] のみが使用可能です。

プロジェクト		
+ 🧿 🌒 🗶 🥹 –		
<ul> <li>→ → MyProject2</li> <li>→ → Library</li> <li>→ → My Ops Flows</li> <li< td=""><td>K G前の変更 JI除 編集 ・ M型 CM ・ CM ・ CM CM CM ・ CM CM ・ CM CM ・ CM CM ・ CM CM ・ CM CM ・ CM ・ CM ・ CM</td><td>追加 <b>更新</b> コミット 元に戻す <b>履歴</b></td></li<></ul>	K G前の変更 JI除 編集 ・ M型 CM ・ CM ・ CM CM CM ・ CM CM ・ CM CM ・ CM CM ・ CM CM ・ CM CM ・ CM ・ CM ・ CM	追加 <b>更新</b> コミット 元に戻す <b>履歴</b>

作成者がオブジェクトを変更する場合 (ステップを追加するなど)、ローカルで変更されます。この場合、[コミット]および [元に戻す]が使用可能です。

プロジェクト	
+ 🕸 🌒 🕷 🗙 🥹	
□・ → MyProject2 □・ → My Ops Flows □・ → My Ops Flows □・ → My Ops Flows □・ → My Ops Flows □・ → My Ops Flows ■ MathOr ■ Get CPU P ● Get CPU P ● Get Netwo ■ Get Proces ● Sample XP ● Windows H ● Configuration ■ Configuration ■ Configuration	追加 更新 コミット 元に戻す 履歴

オプション	説明
追加	ソースコントロールに追加するアイテム(フロー、オ ペレーション、構成アイテムまたはフォルダー)を マークします。Studioの[ <b>追加</b> ]では、先祖および 子孫オブジェクトがすべて含まれます。つまり、フォ ルダーが追加されると、子フローと親フォルダーも すべて追加されます。兄弟アイテムは追加されま せん。Studio内で作成されたアイテムは自動的 に追加され、次のコミットでコミットされます。
コミット	ローカルでの変更をサーバーにコミットします。この オプションは、変更されたアイテムや子アイテムが 変更されたフォルダーで使用できます。 コミットは 再帰的に行われるため、フォルダーをコミットする と、その子アイテムもすべてコミットされます。 コミッ ト後、コミットのコメントを追加できます。
更新	選択したアイテムを更新します。このオプション は、ローカルで追加され、まだコミットされていない アイテムに対しては使用できません。再帰的に 動作し、すべての子アイテムが更新されます。
元に戻す	選択したアイテムのローカルでの変更をすべて元 に戻します。フロー、オペレーション、構成アイテム での変更が元に戻ります。削除されたアイテムが 復元されます。
	<b>重要</b> : アイテムを追加すると自動的に追加済み のマークが付きます。それに対し変更を元に戻す と、追加マークが削除されますが、アイテムは削 除されず、Studio とファイルシステムに存在したま まになります。このアイテムは、メニューの[追加] を使用して再度追加したり削除したりすることが できます。
履歴	[ <b>リビジョン履歴</b> ] ウィンドウには、SCM 履歴が表示されます。詳細については、「リビジョン履歴」を参照してください。

### プロジェクトの削除

 削除するプロジェクトを選択し、[削除]をクリックします。[削除]をクリックすると、プロジェクトが Studio から削除 (インポート取り消し)されます。[ファイルシステムからプロジェクトを削除します (元 に戻すことはできません)] オプションを選択すると、プロジェクトはリポジトリから削除するようマークさ れ、次のコミットで削除が実行されます。

### トラブルシューティング

#### • 自分のプロジェクトを更新し、他ユーザーの変更を削除してしまいました。

以下のシナリオでは、更新中に競合が発生する可能性があります。

- a. ユーザー1がフロー1を編集し、コミットします。
- b. ユーザー2がフロー1を編集し、更新します。

ユーザー2はローカルでの変更を含むファイルへの更新を受け取るため、競合が発生します。

Studio では、このような競合をユーザーのコピーを更新することで解決します。つまり、この場合、 ユーザー1がフロー1に対して加えた変更は削除されます。

このような状況を回避するため、アイテムを編集する前に更新およびロックすることを推奨します。

#### • ファイルシステム内で直接変更を行ったところ、問題が発生しました。

ファイルシステム内で直接ディレクトリを作成したりファイル名を変更したりできますが、これは推奨されません。すべてのタスクをStudio内で実行することをお勧めします。

ファイルシステム内で変更を加え、変更結果に問題が発生した場合は、外部のSVN ツール (SlickSVN または TortoiseSVN)を使用してワークスペースをクリーンアップし、これらの変更を元に 戻すことを推奨します。

# コンテンツパックの使用

コンテンツパックは、オペレーション、フロー、操作、構成アイテム、リソースバンドルを含む jar ファイルです。コンテンツパックは、デプロイメント用に配信可能なエンティティで最も小さいものです。

HPLN から特定のドメインにコンテンツパックをダウンロードし、自分のプロジェクトにインポートすることができます。

# コンテンツパックのプロジェクト へのインポート



このセクションでは、Studioにコンテンツパックをインポートする方法について説明します。

**注**: Studio をインストールする際に、インストールウィザードで、既存のコンテンツパックをインポート するよう選択できます。コンテンツパックは、Studio を最初に起動した時に読み込まれます。

最初にベースコンテンツパックをインポートし、次に独自のドメインでカスタムフローの作成に必要な専用のコンテンツパックをインポートする必要があります。 これらのコンテンツパックは、 HPLN からダウン ロードできます。

コンテンツパックは、オペレーション、フロー、操作 (Java ベースまたは .NET ベース)、構成 アイテム (選択リスト、ドメインタームなど)、リソースバンド ルの集まりです。

コンテンツパックをインポートすると、ファイルは、読み取り専用で[コンテンツパック]ペインで使用できます。

注:ネットワークドライブからコンテンツパックをインポートできます。

コンテンツパックは、リモートホストサイトから直接インポートすることはできません。リモートコンテン ツパックにアクセスするには、[ファイルの選択] ダイアログボックスでその場所まで移動できるよう、 最初にローカルシステムにファイルのコピーをダウンロードするか、ネットワークドライブにマッピングします。

フローは別のプロジェクトから再利用することができます。そのフローで作成したコンテンツパックをイン ポートします。再利用するフローのコンテンツパックへのパッケージングについては、「コンテンツパックの エクスポート」(223ページ)を参照してください。

コンテンツパックは次の複数のソースからダウンロードすることができます。

- My content すでに Studio にあるプロジェクトとコンテンツパック。
- My organization 組織内のほかの作成者により開発されたコンテンツパックで、アーティファクトリポジトリに存在するコンテンツパック。
- HP content HP が定期的にリリースしている、HPLN に存在するコンテンツ。
- Community content ほかの組織によりコミュニティに提供されている、HPLN に存在するコンテンツ。

# 実行する操作

#### コンテンツパックのダウンロード

- 1. HPLN Operations Orchestration Content バージョン 10.00 のサイトを開きます。
- 2. 使用可能なベースコンテンツパックを、ネットワークドライブ上の場所にダウンロードします。
- 3. 指定したドメインで必要な別のコンテンツパックを、ネットワークドライブ上の場所にダウンロードします。

#### Studio の最初の起動時のベースコンテンツパックのインポート

- 1. [コンテンツパック] ペインで、[インポート] ボタン 🥺 を選択します。
- 2. コンテンツパックの場所を参照して[開く]をクリックします。
- 3. [OK] をクリックします。

注: ベースコンテンツパックをインポート するには数分 かかることがあります。

#### コンテンツパックのインポート

- 1. [コンテンツパック] ペインで、[インポート] ボタン 🎱 を選択します。
- 2. コンテンツパックの場所を参照して [開く]をクリックします。

注:必要に応じて、複数のコンテンツパックを一度にインポートできます。

3. [OK] をクリックします。インポートしたコンテンツパックは、[コンテンツパック] ペインに表示されます。

#### コンテンツパックを解凍してプロジェクトとして開く

旧 バージョンの HP OO からコンテンツをアップグレード する場合、アップグレード ツールでコンテンツをコン テンツパックに変換します。次の手順に従って、コンテンツパックを編集可能な プロジェクト として 開きます。

- 1. ファイルシステム (Windows エクスプローラーなど) で、コンテンツパックを解凍 (unzip) します。コンテンツパックは、プロジェクトフォルダーに解凍されます。
- 2. Studio で、[**ファイル**] > [**プロジェクトのインポート**]を選択します。
- 3. [プロジェクトディレクトリの選択] ダイアログボックスで、解凍されたコンテンツパックから作成された プロジェクトの場所を参照します。
- 4. [OK] をクリックします。これで、このプロジェクトを Studio で編集 できるようになります。

注: コンテンツパックからプロジェクトを作成する前に Studio でコンテンツパックを開いた場合、 Studio でコンテンツパックを閉じる必要があります。 プロジェクトとコンテンツパックの UUID は同じで あるため、 Studio で同時に開くことはできません。

# プロジェクト のコンテンツパックの管理

コンテンツパックをインポートすると、フローでコンテンツパックのオペレーションを使用することができますが、このオペレーションは読み取り専用です。 ただし、コンテンツパック内のフローも読み取り専用なので、 ステップとして使用することができます。

コンテンツパックのオペレーションを変更する場合、オペレーションをプロジェクトにコピーする必要があり ます。

**注**: この操作は、レスポンスまたは結果を追加する場合のみ行ってください。入力のみ使用する場合は、ステップ内でオペレーションを変更してください。

プロジェクトにコピーすると、オペレーションはコンテンツパックからデタッチされ、編集できるようになります。この新しいオペレーションをフローにドラッグアンドドロップして、オペレーションのプロパティを変更できます。

コンテンツパックからプロジェクトにコピーされたオペレーションは、「ソフトコピー」です。つまり、アクションプ ラグインをインポートして元のオペレーションを作成した場合、コピーされたオペレーションによって元の オペレーションが引き続き参照されます。アクションプラグインをアップグレードして、新しいバージョンを 呼び出すように元のオペレーションを更新すると、コピーされたオペレーションも自動的に更新されま す。詳細については、「オペレーションの作成」(225ページ)を参照してください。

# 実行する操作

### コンテンツパックのオブジェクトをプロジェクトにコピーする

- 1. [コンテンツパック] ペインで、コピーするオブジェクトを右クリックし、[編集]>[コピー]を選択します。 複数のオブジェクトを選択するには、Shift キーとCtrl キーを使用します。
- [プロジェクト] ペインで、オブジェクトを貼り付ける場所を右クリックし、[編集] > [貼り付け] を選択します。

**ヒント:** [**コンテンツパック**] ペインから [**プロジェクト**] ペインにオブジェクトをドラッグアンドドロップ することもできます。

#### コンテンツパックを削除する

コンテンツパックを削除すると、コンテンツパックを閉じる場合とは異なり、削除したコンテンツパックは ワークスペースから完全に削除されます。

**注:** コンテンツパックを削除した場合、ワークスペースから削除されますが、ファイルシステムからは 削除されません。必要に応じて、削除したコンテンツパックを再度インポートできます。

- 1. コンテンツパックを選択して、[削除] 🎽 ボタンをクリックします。
- 2. 確認のダイアログボックスで [はい] をクリックします。

#### コンテンツパックを閉じる

コンテンツパックを閉じると、閉じたコンテンツパックは [コンテンツパック] ペインに表示されますが、グレー 表示されて使用できなくなります。

以下の場合にコンテンツパックを閉じます。

- ワークスペースに2つのバージョンのコンテンツパックが配置されており、一方で作業するにはもう一方を閉じる必要がある場合。2つのコンテンツパックのUUIDは同じであるため、Studioで同時に開くことはできません。
- コンテンツパックからプロジェクトを作成し、プロジェクトで作業するには元のコンテンツパックを閉じる必要がある場合。プロジェクトとコンテンツパックのUUIDは同じであるため、Studioで同時に開くことはできません。
- 1. 閉じるコンテンツパックを選択します。
- 2. コンテンツパックを右クリックして [閉じる]を選択します。

注: [コンテンツパック] ペインで [閉じる] 🏴 ボタンをクリックすることもできます。

#### 閉じたコンテンツパックを開く

コンテンツパックを閉じた後で、このコンテンツパックを開いて再度作業することができます。

- 1. 開く必要がある、閉じた (グレー表示の) コンテンツパックを選択します。
- 2. コンテンツパックを右クリックして [開く]を選択します。

注: [コンテンツパック] ペインで [開く] 🧖 ボタンをクリックすることもできます。

### インポートしたコンテンツパックのプロパティを表示する

[コンテンツパック] ペインでコンテンツパックを右クリックし、[プロパティ] を選択します。コンテンツパックの [プロパティ] ウィンドウが読み取り専用モードで開きます。

#### コンテンツパック内のオフジェクトのプロパティを表示する

[コンテンツパック] ペインで、フロー、オペレーション、またはその他のオブジェクトをダブルクリックします。 オブジェクトの[プロパティ] ウィンドウが読み取り専用モードで開きます。

参照情報

#### [コンテンツパック] ペイン

[コンテンツパック] ペインに、それぞれが固有の階層ツリー構造を備えた、複数のコンテンツパックを表示できます。



GUI ፖイテム	説明
コンテンツパックのインポート	[コンテンツパックのインポート] ダイアログボックスを開くと、インポー トするコンテンツパックを選択できます。
削除 🗙	選択されたコンテンツパックを削除します。
開く 🍖	現在選択されている、閉じたコンテンツパックを開きます。
閉じる 救	現在選択されているコンテンツパックを閉じて、グレー表示にしま す。

### コンテンツパックのプロパティ

コンテンツパックを右クリックして [プロパティ]を選択すると、コンテンツパックのプロパティシートが開きます。

表示されるプロパティは読み取り専用です。この情報は、インポートしたコンテンツパック jar ファイルから取得されます。コンテンツパックを HP OO プロジェクト から作成した場合は、[コンテンツパックの作成] ダイアログボックスに入力した内容から情報が取得されます。

Base [1.0	).121] ×	٩	Þ∎
📄 🗞			
名前:	Base [1.0.121]		
UUID:	75b8b3d6-d260-43af-acf9-8a142b4feadf		
パブリッシャー:			
i兑8月:			
HP OO Bas	e Content Pack		

GUI アイテム	説明
名前	コンテンツパックの名前。コンテンツパックを HP OO プロジェクトから 作成した場合、この名前はプロジェクト名から取得されます。
UUID	コンテンツパックの一意のID。
パブリッシャー	コンテンツパックのパブリッシャー。コンテンツパックを HP OO プロ ジェクトから作成した場合、[コンテンツパックの作成] ダイアログ ボックスの [ <b>パブリッシャー</b> ] フィールドから取得されます。
説明	コンテンツパックの説明。コンテンツパックを HP OO プロジェクトから 作成した場合、[コンテンツパックの作成] ダイアログボックスの[説 明] フィールドから取得されます。

# 構成アイテムの管理

プロジェクトの Configuration フォルダーには、設定可能なアイテムが格納されています。ドメインターム、グループエイリアス、システムアカウント、システムプロパティなどです。

作成チームが異なるプロジェクトで作業しており、同じ構成アイテムが必要な場合、構成アイテムを 格納した共有プロジェクトを別に作成することをお勧めします。

カテゴリの設定	72
ドメインタームの設定	
グループエイリアスの設定	77
スクリプトレットの設定	80
選択リストの設定	83
システムアカウントの設定	
システムフィルターの設定	
システムプロパティの設定	93

# カテゴリの設定

カテゴリとは、フローに割り当 てることができる分 類 のことです。 Studio には多 数 のカテゴリがインストールされていますが、 独 自 のカテゴリの作 成 もできます。

ユーザーがカテゴリを使用し、重要なインフラコンポーネントの状態を示すレポートを作成することもあります。たとえば、サーバーのヘルスチェックを行うフローすべてにサーバーカテゴリを割り当てた場合、 サーバーカテゴリに割り当てられたフローを検出したときのみ、レポートにネットワーク上のサーバーの状態が強調表示されます。

また検索をフィルターするためにカテゴリを使用することもできます。たとえば、**セキュリティ**カテゴリのフ ローに対してのみ検索を実行するといったようなことができます。

カテゴリは Configuration Categories フォルダーに保存されます。

**6**  MyProject1 > Configuration > Categories > Application Server (.NET) 名前: Application Server (.NET) UUID: 045d282a-fc91-4d73-82f4-427a1fcea1e3 説明
## 実行する操作

#### カテゴリの作 成

- 1. [プロジェクト] ペインで、Categories フォルダーを展開します。
- 2. Categories フォルダーを右クリックし、[新規]をクリックします。
- 3. 表示されたダイアログボックスで、新しいカテゴリの名前を入力して[OK]をクリックします。
- 4. [説明] ボックスに、新しいカテゴリの説明を入力します。
- 5. [保存]をクリックします。

#### カテゴリの変更

- 1. [プロジェクト] ペインで Configuration フォルダーと Categories フォルダーを展開し、カテゴリをダ ブルクリックしてエディターを開きます。
- 2. 変更するカテゴリをダブルクリックして新しい値を入力します。

#### カテゴリの名前の変更

- 1. [プロジェクト] ペインで Configuration フォルダーと Categories フォルダーを展開し、カテゴリをダ ブルクリックしてエディターを開きます。
- 2. [名前] ボックスに、カテゴリの新しい名前を入力します。
- 3. [保存]をクリックします。

#### カテゴリの削除

カテゴリを削除する前に、[**これを使用する項目は?**] 機能を使用して、ほかのアイテムが依存してい ないかをチェックすることをお勧めします。詳細については、「フローとオペレーションの使用方法の詳 細」(240ページ)を参照してください。

- 1. [プロジェクト] ペインで、Configuration フォルダーとCategories フォルダーを展開します。
- 2. カテゴリを右クリックし、[削除]を選択します。
- 3. 確認 ウィンド ウで [はい] をクリックします。

## 参照情報

カテゴリエディター		
💣 🗄 🗒 🇞 🖛	Project1 > Configuration > Categories > Application Server (.NET)	
名前: Application Server (.NET)		
UUID: 045d282a-fc91-4d73-82f4-42	27a1fcea1e3	
記8月:		
GUI アイテム	説明	
名前	カテゴリの名前です。	

## ドメインタームの設定

ドメインタームは、フローや入力に割り当てることのできる属性です。たとえば、システム内のさまざまな 種類のサーバーに対してドメインタームを作成すると、特定の種類のサーバーのみに対してステップを 実行させることができます。

ドメインタームは、特定の選択リストに対し使用することができます。たとえば、それぞれ異なる種類の操作に対するドメインタームを作成できます。このドメインタームの値は Restart、Reboot、Open などにすることができます。

また、別の例として、特定の種類のサーバーに対してフローを実行し、その他の種類には実行しない ように指定する場合、システム内の各種サーバーのドメインタームを追加して、あるフローの実行対象にするサーバーの種類をユーザーが選択できるようなユーザープロンプトを作成することが可能です。

ドメインタームにはデフォルトで値が入っているもの、フローから入力値を取得するもの、ユーザーが値を指定するものがあります。

ドメインタームは Configuration\Domain Terms フォルダーに保存されます。



# 実行する操作

ドメインタームの作成

- 1. [プロジェクト] ペインで、Configuration フォルダーを展開します。
- 2. Domain Terms フォルダーを右 クリックし、[新規]をクリックします。
- 3. 表示されたダイアログボックスで、新しいドメインタームの名前を入力して[OK]をクリックします。
- 4. [説明] ボックスに、新しいドメインタームの説明を入力します。
- 5. [追加]をクリックして、新しいドメインタームの値を追加します。
- 6. [名前]列に、ドメインタームの値の名前を入力します。
- 7. (オプション)[説明]列に、ドメインタームの値の説明を入力します。
- 8. [保存]をクリックします。

#### ドメインタームの値の削除

- 1. [プロジェクト] ペインで Configuration フォルダーと Domain Terms フォルダーを展開し、ドメイン タームをダブルクリックしてエディターを開きます。
- 2. 値を強調表示して、[削除]をクリックします。

#### ドメインタームの値の変更

1. [**プロジェクト**] ペインで Configuration フォルダーと Domain Terms フォルダーを展開し、ドメイン タームをダブルクリックしてエディターを開きます。 2. 変更する値をダブルクリックして新しい値を入力します。

#### ドメインタームの名前の変更

- 1. [**プロジェクト**] ペインで Configuration フォルダーと Domain Terms フォルダーを展開し、ドメイン タームをダブルクリックしてエディターを開きます。
- 2. [名前] ボックスに、ドメインタームの新しい名前を入力します。
- 3. [保存]をクリックします。

#### ドメインタームの削除

ドメインタームを削除する前に、[これを使用する項目は?]機能を使用して、ほかのアイテムが依存 していないかをチェックすることをお勧めします。詳細については、「フローとオペレーションの使用方法 の詳細」(240ページ)を参照してください。

- 1. [プロジェクト] ペインで、Configuration フォルダーとDomain Term フォルダーを展開します。
- 2. ドメインタームを右クリックし、[削除]を選択します。
- 3. 確認ウィンドウで[はい]をクリックします。

### 参照情報

ドメインタームエディター

💼 📑 📑 🏀 – MyProject1 > Configuration > Domain Terms > Severity			
名前: Severity			
UUID: 428ccc34-1a3d-4656-add8- 訳明:	250acf304354		
Information, Warning, Error, Critica	1		
			追加 削除
名前			
Information			
Warning			
Error			
Critical			
GUI アイテム	説明		

GUI アイテム	説明
名前	ドメインタームの名前です。
説明	(オプション)ドメインタームの説明です。

追加	[追加]をクリックして、新しいアイテムをドメインタームリストに追加します。
削除	[ <b>削除</b> ]をクリックして、選択したアイテムをドメインタームリストから削除します。
[名前] 列	ドメインタームリストにあるアイテムの名前を入力します。
[説明] 列	(オプション)ドメインタームリストにあるアイテムの説明を入力します。

# グループエイリアスの設定

#### RAS グループ

RAS グループは、RAS の論理的集合です。デプロイメントにより、単一のRAS を特定の環境で使用するよりも多くの利点を得られます。たとえば、2つのRAS を操作実行負荷に耐えられるようにする必要があるリモートデータセンターを管理している場合や、データセンターのRAS の高い可用性を得たい場合などです。

RESTful API を使用して、RAS グループをサーバーで定義できます。詳細については、『HP OO Application Program Interface (API) Guide』を参照してください。

#### グループエイリアス

グループエイリアスで、作成時と実行環境でのオペレーションの RAS への割り当てを別にすることができます。

- 1. 作成時は、作成者はワーカーグループではなくグループエイリアスで実行するオペレーションを定 義します。
- 実行時は、管理者がエイリアスを実行環境のRAS グループに、Central RESTful APIを使用してマップします。管理者が、フローを開いて手動でRASの割り当てを変更する必要はありません。

フォールバックとして、グループエイリアスがグループ名と同一の場合、そのグループに自動的にマッピングされます。

オプションで、トリガー時 にグループエイリアスをオーバーライドしてオペレーションを異なる RAS グループ にマップできます。



たとえば、Oracle クライアントで実行する3つの RAS のグループがあるとします。Oracle でクエリを実行するオペレーションを作成します。このグループに対するエイリアスを使用することで、HP OO に、この オペレーションがこのグループのいずれかの RAS で実行する必要があることを知らせます。使用する RAS の選択は実行時に決定されるため、オペレーションで設定する必要はありません。

グループエイリアスは Configuration\Group Aliases フォルダーに保存されます。

In the second se
名前: Oracle
UUID: 2d21b507-6f8d-4200-8056-021e6d8842d7
Use the Oracle Group.

# 実行する操作

#### グループエイリアスの作成

たとえば、Oracle クライアントで実行する3つの RAS の RAS グループを作成したとします。オペレー ションにこのグループで作業するよう指示するグループエイリアスを作成する必要があるため、これらの RAS のいずれかを実行する必要があります。

- 1. [プロジェクト] ペインで、Configuration フォルダーを展開します。
- 2. Group Aliases フォルダーを右クリックし、[新規]をクリックします。
- 3. 表示されたダイアログボックスで、新しいグループエイリアスの名前を入力して [**OK**] をクリックします。
- 4. [説明] ボックスに、新しいグループエイリアスの説明を入力します。
- 5. [保存]をクリックします。

#### グループのグループエイリアスへのマッピング

Central RESTful APIを使用して、実行環境でグループエイリアスをグループにマッピングします。 詳細については、『HP OO Application Program Interface (API) Guide』を参照してください。

#### オペレーションでのグループエイリアスの使用

- 1. 「オペレーションの作成」(225ページ)の説明に従って、アクションプラグインから新しいオペレーションを作成します。
- 2. [入力] タブの [Operation フィールド] セクションで、[グループエイリアス] 🍥 ボタンをクリックします。

Operation フィールド ――	
グループ ID:	comhp.oo
アーティファクト ID:	oo-base-legacy-plugin
バージョン・	1.0.121
アクション名の	com.iconclude.content.actions.mail.SendMail
グループエイリアス:	/Base [1.0.121]/Configuration/Group Aliases/RAS_Operator_Path
グループのオーバーライド	\${overrideJRAS}

- 3. Configuration/Group Aliases フォルダーで設定済みのグループエイリアスを参照して選択します。
- 4. (オプション)現在のグループを別のグループにオーバーライドする必要がある場合、[グループのオー バーライド] ボックスに別のグループを入力します。
- 5. オペレーションを保存します。

### 参照情報

グループエイリアスのエディター

円 🗞 MyProject1 > Configuration > Group Aliases > Oracle 名前: Oracle UUID: 2d21b507-6f8d-4200-8056-021e6d8842d7 説明 Use the Oracle Group.

GUI ፖイテム	説明
名前	グループエイリアスの名前です。
説明	(オプション) グループエイリアスの説明です。

# スクリプトレットの設定

スクリプトレット (JavaScript で記述) は必要に応じて、データを加工する目的でオペレーションに追加 します。

スクリプトレットを使用すると、次のことを実行できます。

- オペレーション、フロー、ステップの結果をフィルタリングする
- オペレーションのレスポンスを決定する
- データを親フローに渡す前に、サブフローでデータを加工する

たとえば、divide by 1k スクリプトレットでは、入力した数字(被除数)を1024(除数)で数学的除算 をし、商を丸めて最も近い整数値を返します。



システムスクリプトレットを最初から、またはオペレーションの既存のスクリプトレットから作成し、共有シ ステムスクリプトレットとして保存できます。 結果のスクリプトレットは、作成したコンテキストに依存せ ず、ほかのオペレーションフローやステップで再使用できます。

システムスクリプトレットは Configuration\Scriptlets フォルダーに保存されます。

スクリプトレットの使用の詳細については、「フロー内でのスクリプトレットの使用」(198ページ)を参照 してください。

## 実行する操作

#### 既存のスクリプトレットをシステムスクリプトレットとして保存

1. [プロパティ] シートまたはステップのインスペクターの [スクリプトレット] タブで、システムスクリプトレッ

トとして保存するスクリプトレットを開きます。

- 2. [プロジェクト] ペインで、Configuration フォルダーとScriptlets フォルダーを展開します。
- 3. [プロパティ] シートまたはステップのインスペクターの [スクリプトレット] タブから [スクリプトレット] アイコンを Configuration\Scriptlets フォルダーにドラッグします。
- 4. 新しいシステムスクリプトレットの名前を変更するには、右クリックし、[名前の変更]をクリックして 名前を変更します。

#### オペレーション、フロー、ステップのシステムスクリプトレットのフィルタリング

- 1. システムスクリプトレットを使用するオペレーション、フロー、ステップについて、[**プロパティ**]シートまた はステップのインスペクターの[**スクリプトレット**] タブを開きます。
- 2. [プロジェクト] ペインで、Configuration フォルダーとScriptlets フォルダーを展開します。
- スクリプトレットを Scriptlets フォルダーから [プロパティ] シートまたはステップのインスペクターの [スクリプトレット] タブの [スクリプトレット] 「アイコンにドラッグします。

[スクリプトレット] タブに、共有のスクリプトレットへの参照があることが表示されます。

入力 | 出力 | レスポンス | 説明 スクリプトレット
 共有スクリプトレット への参照: /MyProject1/Configuration/Scriptlets/divide by 1k
 カスタムスクリプトレットに切り替え

#### システムスクリプトレットの作成

- 1. [プロジェクト] ペインで、Configuration フォルダーを展開します。
- 2. Scriptlets フォルダーを右 クリックし、[新規]をクリックします。
- 3. スクリプトレットの名前を入力して[OK]をクリックします。

スクリプトレットのエディターが開きます。



- 4. [説明] ボックスに、スクリプトレットの目的を入力します。
- 5. スクリプトレットを JavaScript で入力します。
- 6. (オプション)[**テンプレートの挿入**]をクリックして、テンプレートのガイドラインに従いスクリプトレットを 記述します。
- 7. [スクリプトの確認]をクリックしてスクリプトにエラーがないか確認します。「出力および結果のフィル タリング」(156ページ)
- 8. [保存]をクリックします。

スクリプトレットが Scriptlets フォルダーに保存され、任意のオペレーション、フロー、ステップで使用できるようになります。

#### システムスクリプトレットの編集

- 1. [プロジェクト] ペインで、Configuration フォルダーとScriptlets フォルダーを展開します。
- 2. 編集するシステムスクリプトレットをダブルクリックします。
- 3. スクリプトレットを変更して[保存]をクリックします。

#### システムスクリプトレットの削除

システムスクリプトレットを削除する前に、[これを使用する項目は?]機能を使用して、ほかのアイテム が依存していないかをチェックすることをお勧めします。詳細については、「フローとオペレーションの使 用方法の詳細」(240ページ)を参照してください。

- 1. [プロジェクト] ペインで、Configuration フォルダーとScriptlets フォルダーを展開します。
- 2. システムスクリプトレットを右クリックし、[削除]を選択します。
- 3. 確認ウィンドウで[はい]をクリックします。

## 参照情報

スクリプトレットエディター

MyProject1 > Configuration > Scriptlets > Unix Newline To Windows

名前: Unix Newline To Windows

UUID: c148c565-7a2c-4c56-b4b7-b98a21f5d6f2 説明:

Converts the line endings of the multiline string input from UNIX (line break) to DOS (carriage return, line break) format.

テンプレートの挿入 スクリプトの確認

٠

1 scriptletResult = scriptletInput.replace(/\n/g,"\r\n");

GUI ፖイテム	説明
名前	スクリプトレットの名前です。
説明	(オプション) スクリプトレットの目的の説明です。
テンプレートの挿入	[テンプレートの挿入]をクリックすると、スクリプトレットを書くのに役立つガイ ドラインが表示されます。
スクリプトの確認	[ <b>スクリプトの確認</b> ]をクリックすると、スクリプトレットにエラーがないか確認されます。

## 選択リストの設定

選択リストは、フローのユーザープロンプトで用られるアイテムを一覧にしたものです。

**重要**: 現在のバージョンでは、ユーザープロンプトはサポートされていません。 今後、ユーザープロンプトがサポートされるバージョンのため選択リストを設定して準備しておくことはできますが、このバージョンではフローで使用することはできません。

たとえば、フローユーザーがフローのステップにサービス状態を指定する必要がある場合、選択リストを データソースとした入力を作成し、選択リストの中からサービスステータスを指定できます(この選択リ ストは、実行中、停止、一時停止で構成されています)。

選択リストは、Configuration\Selection Lists フォルダーに保存されます。



## 実行する操作

#### 選択リストの作成

- 1. [プロジェクト] ペインで、Configuration フォルダーを展開します。
- 2. Selection Lists フォルダーを右 クリックし、[新規] をクリックします。
- 3. 表示されたダイアログボックスで、新しい選択リストの名前を入力して[OK]をクリックします。
- 4. [説明] ボックスに、新しい選択リストの説明を入力します。
- 5. [追加]をクリックして、新しい選択リストの値を追加します。
- 6. [値]列に、選択リストの値の名前を入力します。
- 7. [保存]をクリックします。

#### 選択リストの値の削除

- 1. [プロジェクト] ペインで Configuration フォルダーと Selection Lists フォルダーを展開し、選択リ ストをダブルクリックしてエディターを開きます。
- 2. 値を強調表示して、[削除]をクリックします。

#### 選択リストの値の変更

1. [プロジェクト] ペインで Configuration フォルダーと Selection Lists フォルダーを展開し、選択リ ストをダブルクリックしてエディターを開きます。 2. 変更する値をダブルクリックして新しい値を入力します。

#### 選択リストの名前の変更

- 1. [プロジェクト] ペインで Configuration フォルダーと Selection Lists フォルダーを展開し、選択リ ストをダブルクリックしてエディターを開きます。
- 2. [名前] ボックスに、選択リストの新しい名前を入力します。
- 3. [保存]をクリックします。

#### 選択リストの削除

選択リストを削除する前に、[これを使用する項目は?]機能を使用して、ほかのアイテムがこの選択 リストに依存していないかをチェックすることをお勧めします。詳細については、「フローとオペレーション の使用方法の詳細」(240ページ)を参照してください。

- 1. [プロジェクト] ペインで、Configuration フォルダーとSelection Lists フォルダーを展開します。
- 2. 選択リストを右クリックし、[削除]を選択します。
- 3. 確認ウィンドウで[はい]をクリックします。

## 参照情報

選択リストのエディター

🔐 🖪 🗒 🇞 🖬	/Project1 > Configuration > Selection Lists > Cha	racter Sets
名前: Character Sets		
UUID: c6ae688c-6271-4716-ac9b-   於8月:	933bec36c1c	
Different types of character sets		
		追加削除
	値	
UTF-8		
UTF-16		
UTF-32		
EUC-JP		
ISO-2022-JP		
Shift_JIS		
Windows-31J		
GUI アイテム	説明	

名前	選択リストの名前です。
説明	(オプション) 選択リストの目的の説明です。
追加	[追加]をクリックして、新しい選択リストの値を追加します。
削除	[ <b>削除</b> ]をクリックして、選択した値を選択リストから削除します。
値	選択リストの値を入力します。

# システムアカウントの設定

システムアカウントは、アカウントの資格情報 (ユーザー名とパスワード)を含むオブジェクトです。システムアカウントが作成される Studio のインストール時以外では資格情報が表示されないように保護されます。

フロー作成者は、フローの作成時にシステムアカウントを使用できます。たとえば、入力ソースを、シス テムアカウントからの資格情報に設定することができます。「入力ソースの指定」(125ページ)を参照し てください。

**注**: ここで定義するシステムアカウントは、Studio のみで使用されます。また、システムアカウントは 実行に設定する必要があります。これは、APIを使用して行います。詳細については、『HP OO Application Program Interface (API) Guide』を参照してください。

システムアカウント名はユーザーにー度も表示されることなく、リモートマシンへのアクセスのためユー ザーアカウント資格情報がフローに送られます。したがって、資格情報は復号化から保護され、シス テムアカウント名はユーザーに対して非表示となります。

システムアカウントは Configuration\System Accounts フォルダーに保存されます。

**注:** <>\\"/;%の文字はシステムアカウント名には使用できません。

In the MyProject1 > Configuration > System Acco	unts > John Citizen
名前: John Citizen	
UUID: 3843ccd4-3206-4f49-8550-c5925596b982 訳明:	
パスワード: ************************************	パスワードの割り当て

## 実行する操作

#### システムアカウントの作成

- 1. [プロジェクト] ペインで、Configuration フォルダーを展開します。
- 2. System Accounts フォルダーを右 クリックし、[新規]をクリックします。
- 3. 表示されたダイアログボックスで、新しいシステムアカウントの名前を入力して [**OK**] をクリックしま す。
- 4. (オプション)[説明] ボックスに、システムアカウントの説明を入力します。
- 5. [ユーザー名] ボックスに、次の構文を使用して、このシステムアカウントを表すアカウントのユー ザー名を入力します。

<ドメイン>\<ユーザー名 >

- 6. [パスワードの割り当て] ボタンをクリックします。
- 7. [パスワード] ボックスにパスワードを入力し、[確認パスワード] ボックスに再度入力します。
- 8. [保存]をクリックします。

#### システムアカウントの編集

- 1. [プロジェクト] ペインで、Configuration フォルダーとSystem Accounts フォルダーを展開します。
- 2. 編集するシステムアカウントをダブルクリックします。
- 3. エディターで変更し、[保存]をクリックします。

#### システムアカウントの削除

- 1. [プロジェクト] ペインで、Configuration フォルダーとSystem Accounts フォルダーを展開します。
- 2. システムアカウントを右クリックし、[削除]を選択します。
- 3. 確認 ウィンド ウで [はい] をクリックします。

**注**: システムアカウントをコンテンツパックから削除し、コンテンツパックを再デプロイすると、システムアカウントはデータベースから削除されません。この場合、rest API でシステムアカウントを削除する必要があります。

次の場所で DELETE を実行します。/oo/rest/system-accounts/<sa\_name>

rest API の使用の詳細については、『HP OO Application Program Interface (API) Guide』 を参照してください。

## 参照情報

システムアカウントのエディター

▼ ▲ MyProject1 > Configuration > System Accounts > John Citizen

名前: John Citizen UUID: 3843ccd4-3206-4f49-8550-c5925596b982 説明:

日 🗞

資格情報			
ユーザー名:	<ドメイン>¥<ユーザー名>		
パスワード	***************************************	パスワードの割り当て	

GUI アイテム	説明
名前	システムアカウントの名前です。
	<b>注:</b> <>\\"/;% の文字はシステムアカウント名には使用で きません。
説明	(オプション)システムアカウントの目的の説明です。
ユーザー名	<ドメイン>\<ユーザー名 > という構文を使用して、このシステム アカウントを表すアカウントのユーザー名を入力します。
パスワードの割り当て	クリックして [入力 パスワード] ダイアログボックスを開き、パスワー ドを2回入力します。

# システムフィルターの設定

フィルターを使用して、オペレーションの出力や、ステップの結果の抽出や編集ができます。システムフィルターは、システム全体で、複数のステップやオペレーションで使用できます。

たとえば、ある ping オペレーションで使用されるフィルターは、ほかの ping オペレーションで役に立つ可能性があります。

システムフィルターを最初から、またはオペレーションの既存のフィルターから作成し、システムフィルター として保存できます。結果のシステムフィルターは、作成したオペレーションに依存せず、ほかの出力や 結果で再使用できます。

システムフィルターは Configuration\System Filter フォルダーに保存されます。

Interest in the second	
名前: TableFilter	
UUID: 4b0f1926-57ad-4a62-a64c-01d9880fe815	
説明:	
I	
入力をテーブルとして解析し、指定された列でソートします	
列の区切り記号: スペース 💌 行の区切り記号: 改行 💌	
最初の行はヘッダー:  「 結果の最初の行を削除:  「	
ソート基準列: -1 昇順:	
選択行: 0 選択列: 0	
選択幅: 1 選択高: 1	
フィルターのテ	21 L

## 実行する操作

#### 既存のフィルターをシステムスフィルターとして保存

- 1. 該当のオペレーションを開いて、フィルターエディターで、システムフィルターとして保存するフィル ターを選択します。
- 2. [プロジェクト] ペインで、Configuration フォルダーとSystem Filters フォルダーを展開します。
- 3. オペレーションのフィルターエディターで、[**フィルター**] リストからフィルターを System Filters フォル ダーにドラッグします。



4. 新しいシステムフィルターの名前を変更するには、右クリックし、[名前の変更]をクリックして名前 を変更します。

#### 出力または結果でのシステムフィルターの使用

- 1. システムフィルターを使用する出力または結果のフィルターエディターを開きます。
- 2. [プロジェクト] ペインで、Configuration フォルダーとSystem Filters フォルダーを展開します。
- 3. 使用するフィルターを、System Filters フォルダーからフィルターエディターの [フィルター] リストにド ラッグします。

	追加	削除	1 🕂
🔆 共有フィノ	レター: Table	Filter	
J			

システムフィルターの作成

- 1. [プロジェクト] ペインで、Configuration フォルダーを展開します。
- 2. System Filters フォルダーを右クリックし、[新規]をクリックします。
- 3. [フィルターの選択] リストで、フィルターの種類を選択します。

フィルターの選択	×
追加するフィルターの選択	
形式	
形式	
行の選択	
(丁安又刀フント 間116余	
重複行の削除	
数値の丸め	
数値の抽出	
止現表現	<b>•</b>

4. フィルターの名前を入力して[OK]をクリックします。

フィルターエディターが開きます。フィルターエディターの外観は、選択したフィルターの種類によって異なります。

🔐 🗒 🗒 🇞 🗖	MyProject1 > Configuration > System Filters > Table				
名前: TableFilter					
UUID: 4b0f1926-57ad-4a62-a64c-01d9880fe815					
説明:					
スーカナニ、ゴルトレア32461 まわつう					
ヘリをナーノルどして解析し、指定の					
列の区切り記方:[スペース]]					
最初の行はヘッダー: Ⅰ_	結果の最初の行を削除:				
ソート基準列:  -1	昇順: □				
選択行:  0	選択列: 0				
選択幅:  1	選択高:  1				
	フィルターのテスト				
- 771 4- 3 70-72					

- 5. [説明] ボックスに、フィルターの目的を入力します。
- 出力または結果のフィルターに使用するテキスト、文字列、式の値、またはスクリプトレットを入力します。それぞれのフィルターオプションの詳細については、「出力および結果のフィルタリング」 (156ページ)を参照してください。
- 7. フィルターをテストします。
  - a. [クリア]をクリックして [フィルター入力のテスト] ボックスの内容をクリアします。
  - b. [**クイックコマンド**]をクリックします。
  - c. 目的のデータを生成するコマンドを入力します。
  - d. [OK] をクリックします。コマンドの出力が [フィルター入力のテスト] ボックスに表示されます。

フィルターのテストの詳細については、「出力および結果のフィルタリング」(156ページ)を参照してください。

8. [保存]をクリックします。

フィルターが System Filters フォルダーに保存され、入力エディターの[検査形式]リストで使用 できるようになります。

#### システムフィルターの編集

- 1. [プロジェクト] ペインで、Configuration フォルダーとSystem Filters フォルダーを展開します。
- 2. 編集するシステムフィルターをダブルクリックします。
- 3. フィルターを変更して[保存]をクリックします。

#### システムフィルターの削除

システムフィルターを削除する前に、[これを使用する項目は?]機能を使用して、ほかのアイテムが依存していないかをチェックすることをお勧めします。詳細については、「フローとオペレーションの使用方法の詳細」(240ページ)を参照してください。

- 1. [プロジェクト] ペインで、Configuration フォルダーとSystem Filters フォルダーを展開します。
- 2. システムフィルターを右クリックし、[削除]を選択します。
- 3. 確認 ウィンド ウで [はい] をクリックします。

### 参照情報

#### フィルターエディター

フィルターエディターの外観は、選択したフィルターの種類によって異なります。それぞれのオプションの詳細については、「出力および結果のフィルタリング」(156ページ)の「フィルターオプション」を参照してください。

💣 💾 🛃 🇞 🔹 🗠 MyProject 1 > Config	uration > System Filters > Table
名前: TableFilter	
UUID: 4b0f1926-57ad-4a62-a64c-01d9880fe815	
入力をテーブルとして解析し、指定された列でソートします	
列の区切り記号: スペース 💌 行の区切り記号	: 改行 💽
最初の行はヘッダー: 🔲 結果の最初の行を削除	i: 🗖
ソート基準列: -1 昇川	Ē: <b>Г</b>
選択行: 0 選択歹	l]: 0
選択幅: 1 選択語	5: 1
	フィルターのテスト
「フィルター入力のテスト	
	🖹 🖺 🔏 クリア タイックコマンド

GUI ፖイテム	説明
名前	システムフィルターの名前が表示されます。
説明	システムフィルターの説明を入力します。

# システムプロパティの設定

システムプロパティは、固定の値を持つ、どのようなフローでも使用可能なグローバルフロー変数で す。フロー変数を使用する必要が生じるたびにこの変数を再作成する手間を省けます。システムプロ パティの参照はすべて、システムプロパティの値を取得します。

たとえば、SQL Database Type システムプロパティは、さまざまな種類のSQL データベースを一覧表示します。



システムプロパティは Configuration\System Properties フォルダーに保存されます。

## ベストプラクティス

次の理由のため、システムプロパティの作成または値の変更は慎重に行ってください。

- 適用範囲がグローバルであり、実行時に、フロー実行のコンテキストの一部となるため。結果として、システムプロパティの値を変更すると、既存のオペレーションとフローが壊れる恐れがあります。
- 実行開始時にフローのコンテキストの一部となるため。
- すぐには表示されません。システムプロパティは、Studio Debugger ([システムプロパティ]の[コンテキ ストインスペクター])とConfiguration\System Properties フォルダーで表示されます。

さらに、フローを作成するときに入力を作成すると、フローの実行時に入力と同じ名前のフロー変数が自動的に作成されます。このため、システムプロパティと同じ名前の空のフロー変数が自動的に作成され、予期しない動作が得られることがあります。

同じ名前のフロー変数が優先されます。入力がフロー変数からの値を取得可能な場合、つまり、フロー変数が存在しており、それに割り当てられている値がある場合は、フロー変数は、入力値のソースとしてシステムプロパティよりも優先されます。

ー方、システムプロパティの値は、入力値の割り当てや、システムプロパティの結果によって変更する ことはできません。値を、これら2種類のソースからシステムプロパティに割り当てると、システムプロパ ティと同じ名前のフロー変数が作成され、値がそのフロー変数に割り当てられます。

## 実行する操作

#### システムプロパティの作成

- 1. [プロジェクト] ペインで、Configuration フォルダーを展開します。
- 2. System Properties フォルダーを右 クリックし、[新規]をクリックします。
- 3. システムプロパティの名前を入力して [OK] をクリックします。

システムプロパティのエディターが開きます。



- 4. (オプション)[説明] ボックスに、システムプロパティの説明を入力します。
- 5. [プロパティ値] ボックスに、システムプロパティの値を入力します。区切り記号として | を使用します。
- 6. [保存]をクリックします。

システムプロパティが System Properties フォルダーに保存され、フローで使用できるようになります。

#### フローでのシステムプロパティ値の使用

- 1. システムプロパティを、フローまたはステップ入力のデータソースとして指定します。
- 2. スクリプトレットで、適切なコマンドを使ってシステムプロパティの値を取得します。

注:オペレーションの[スクリプトレット] タブで必要なコマンドとその構文の詳細については、 [テンプレートの挿入]をクリックします。テンプレートで、グローバルコンテキストを使用するため に必要なコマンドが参照できます。詳細については、「フロー内でのスクリプトレットの使用」 (198ページ)を参照してください。

#### システムプロパティの値の変更

システムプロパティの値を変更するには、次のような方法があります。

- スクリプトレットでシステムプロパティを変更します。この変更により、スクリプトが実行する時点から 値が変わります。オペレーションの[スクリプトレット]タブで必要なコマンドとその構文の詳細につい ては、[テンプレートの挿入]をクリックします。
- システムプロパティの値を設定するオペレーションを作成します。
- Configurations\System Properties フォルダーのシステムプロパティを開き、値を変更します。

#### システムプロパティの削除

システムプロパティを削除する前に、[これを使用する項目は?]機能を使用して、ほかのアイテムが依存していないかをチェックすることをお勧めします。詳細については、「フローとオペレーションの使用方法の詳細」(240ページ)を参照してください。

- 1. [**プロジェクト**] ペインで、Configuration フォルダーとSystem Properties フォルダーを展開します。
- 2. システムプロパティを右クリックし、[削除]を選択します。
- 3. 確認 ウィンド ウで [はい] をクリックします。

## 参照情報

#### システムプロパティのエディター

💼 📑 🛃 🇞 🔹 All Superior System Properties SQL Database Type
名前: SQL Database Type
UUID: 5ed7faa2-c0c8-46e1-9627-8885c996c396 [於8月:
プロパティ値: Oracle  MSSQL  Sybase  Netcool  DB2

GUI ፖイテム	説明
名前	システムプロパティの名 前 です。
説明	(オプション)システムプロパティの説明です。
プロパティ値	システムプロパティの値を入力します。区切り記号として   を使用します。

# フロー作成 - 基本

フローとは、タスクを自動化するために意思決定ロジックにより連結された操作の集まりです。

たとえば、ある Web サイトのページに最新の正しいデータ(特定のテキストなど)が含まれているかどう かを確認するとします。対象のデータがそのWebページに表示されていない場合は、サイトに新しい コンテンツをプッシュする必要があります。これらのタスクを自動的に行うフローを作成できます。

この章では、単純なフローを作成するために必要な、すべての基本的なステップについて説明します。より複雑なフローの作成については、「高度な作成」(183ページ)を参照してください。

フローの作成 – 手順	97
新しいフローの作成	
フロー内 でのステップの作 成	104
フローの外 観 の調 整	110
フローの変更	113
入力の作成	118
トランジションの作成	
レスポンスの設定	
出 カと結果 の作 成	147
変数の使用	173
リターンステップの作成	

## フローの作成 - 手順



このトピックでは、フローの作成の主な手順を、1 つずつ説明します。 Web サイトのページに特定のテキストが含まれているかをチェックし、テキストが見つからなかった場合に Web サイトにページを投稿する簡単なフローを作成する方法を説明します。

これは Studio のワークフローの概要を説明するだけであり、ここで説明しないオプションもたくさんあります。 手順ごとの詳細情報については、リンク先の、詳細なフロー作成オプションを参照してください。

このトピックでは、推奨されている順番で手順を進めていますが、必ずしもこの手順で行う必要はあり ません。

フロー作成のベストプラクティスについては、「ベストプラクティス」(20ページ)を参照してください。

手順 1: フローの作成

フローの作成 / ステップとしてきか リターンステップの作成 トランジションの作成 / 入力の追加 / 結果の追加 / フローの保存
--

- 1. フローを作成するプロジェクトを開きます。
- 2. フローを作成するフォルダーを開きます。
- 3. [ファイル] > [新規] > [フロー] を選択します。
- 4. フローの分かりやすい名前を入力して[OK]をクリックします。

注:名前の長さは最大 128文字までです。大文字と小文字は区別されません。

- 5. このフローを作成ペインで開いて [プロパティ] (ペインの下部)をクリックし、[説明] タブをクリックします。
- 6. フローの説明を入力します。

詳細情報とオプションについては、「新しいフローの作成」(101ページ)を参照してください。

#### 手順 2:オペレーションをステップとして追加

オペレーションを作成ペインにドラッグし、フローのステップとして使用します。

オペレーションを編集する場合、コピーして [プロジェクト] ペインに貼り付けてから、作成ペインにそのオペレーションをドラッグします。

**注**: この操作は、レスポンスまたは結果を追加する場合のみ行ってください。入力のみ使用 する場合は、ステップ内でオペレーションを変更してください。

例では、2つのステップがあります。ステップ1では、Webページに特定のテキストが含まれているかどう かをチェックするという Check Web Site オペレーションを用いています。ステップ2は Post Page オペ レーションを使用して、ページを Web サイトに投稿します。

詳細情報とオプションについては、「フロー内でのステップの作成」(104ページ)を参照してください。

#### 手順 3: リターンステップの作成とフローの終了

1つまたは複数のリターンステップを作成してフローを終了します。リターンステップには、フローが取り 得る次の4つの終了状態があります。解決済み、診断済み、操作なし、失敗の4つです。

- 1. 作成ペインのツールバーで、[ステップパレット] ボタン <sup>50</sup> <sup>5</sup> をクリックします。
- 2. [ステップ] パレットから、使用するリターンステップのアイコンを作成キャンバスにドラッグします。
- 必要に応じて、ステップに割り当てられたフローレスポンスを変更します。

例のフローでは、[成功]と[失敗]の2つの終了状態があります。 デフォルトのフローレスポンスを変更 する必要はありません。

詳細情報とオプションについては、「リターンステップの作成」(179ページ)を参照してください。

#### 手順 4:トランジションの作成

ステップの各レスポンスからフロー内の別のステップまたは終了ステップに進めるように、ステップ間を接続します。

**注:** ステップ上のすべてのレスポンスアイコンが別のステップに接続されている必要があります。接続されていないアイコンがある場合、フローにエラーが表示されます。



- 2. トランジションを修正する必要がある場合は、線をダブルクリックして、トランジションのインスペク ターを開きます。
- 3. ステップの別のレスポンスアイコンに対し、同じ処理を繰り返します。

例では次のようになります。

- Web ページが見つからない場合、フローは失敗で終わります。赤い [failure] レスポンスアイコンから[失敗] エンドステップまでドラッグします。
- ページがあるのにテキストが存在しない場合は、別のページをWebサイトに投稿する2番目のステップに進みます。黄色いレスポンスアイコンから2番目のステップまでドラッグします。
- ページが存在し、目的のテキストが存在する場合は、フローは成功状態で終了します。どちらの ステップでも、緑の[success] レスポンスアイコンから[成功] エンドステップまでドラッグします。

トランジションの詳細については、「トランジションの作成」(133ページ)を参照してください。

オペレーションのレスポンスの設定の詳細については、「レスポンスの設定」(137ページ)を参照してください。

#### 手順 5: フローへの入力の追加

- 1. ステップをダブルクリックして、詳細をステップのインスペクターに表示します。
- 2. ステップのインスペクターで、[入力]タブをクリックします。
- 3. [入力の追加]をクリックして、入力名を入力します。
- 4. 目的の行の[種類]列をクリックし、リストから次のいずれかの値の割り当てタイプを選択します。
  - 単一値
  - 値のリスト
- 5. 入力のデータソースを定義するには、入力の行の最後にある矢印 😂 をクリックして、入力エディターを開きます。

例では、Check Web Site ステップで、チェック対象のページ (mysite.com/mypage.htm) および検索 するテキスト (「必要なテキスト」)を知る必要があります。この場合、単一固定値の入力を2つ作成 します。

詳細情報とオプションについては、「入力の作成」(118ページ)を参照してください。

#### 手順 6: フローへの結果の追加

結果を追加すると、ステップの出力を取得でき、これをフロー変数として保存(フロー内の別のステップ で使用する)するか、フロー出力フィールドとして保存(親フローに渡される)することができます。

- 1. ステップのインスペクターで [結果] タブをクリックします。
- 2. [結果の追加]をクリックします。
- 3. [名前]列に、結果の名前を入力し、キーボードのリターンキーを押します。この名前は、フロー 変数またはフロー出力フィールドの名前に使用されます。
- 4. [ソース] リストから、結果のソースを選択します。たとえば、プライマリ出力を選択します。
- 5. [割り当て先] リストで、値の保存先を指定します。フロー変数または出力フィールドのいずれかです。
- [割り当て操作]リストから、適切な操作を選択します。OVERWRITE、APPEND、PREPEND、または算術割り当て操作のいずれかです。

例 では、Check Web Site ステップを、見 つかったテキストをフロー変数 として保存 するよう設定 できます。この簡単な 2 ステップのフローでは、最初のステップが成功 として完了 すると、まっすぐに Resolved: Success エンドステップに進みます。しかし、フロー変数 データを電子 メールの本文に含める Send Email ステップなどの別のステップを追加することができます。

詳細情報とオプションについては、「ステップの結果の設定」(150ページ)を参照してください。

#### 手順 7:フローの保存

[保存] ボタンをクリックします。

Studio でフローが検証されます。フローが有効でない場合、保存が行われてエラーメッセージが表示 されます。

次の図は各種フロー要素を示しています。



# 新しいフローの作成

フローを作成するには2つの選択肢があります。

- フローを最初から作成する
- フローのベースとして定義済みのテンプレートを使用する

### ベストプラクティス

フローへの名前付け

- 作成するフローの構造を、構築する前に設計してあることを確認します。再使用ができるような、 複数の小さいフローに分割することが可能かどうかを考えます。
- フローに、フローの目的を説明する、内容がわかるような名前を付けます。
- フローの種類ごとに異なる命名規則を使用します。たとえば、フロー名に、フローの種類に基づいて接頭辞を追加します。フロー作成者向けの命名規則およびその他のガイドラインについてのドキュメントを使用することをお勧めします。
- 大文字と小文字に一貫性があるようにします。たとえば、フロー名はすべてタイトルケースを使用します。
- シングルタスクで実行するフローの場合、「<動詞><名前>」という形式で名前を付けます。たとえば、Send Mail、Create Snapshotのようになります。
- サンプルのフローでは、名前に "Sample" という語を使用します。たとえば、Send Mail Sample、Create Snapshot Sample のようになります。
- 条件に一致するものがあるかどうかをチェックするフローの場合、問いを名前として使用します。たとえば、Is Computer Account Enabled のようになります。

- システムまたは環境に関する情報を収集するヘルスチェックフローの場合、フロー名に "Health check" を含めます (専用の Health Check フォルダーがある場合を除く)。たとえば、Solaris Health Check のようになります。
- 統合を行っている場合、使用されている API からの元のフロー名を保持します。

#### フローの説明

- フローを作成する際、フローで行われる内容の説明を、[説明] タブに追加することをお勧めします。説明の中に検索語を含め、フローを検索する際に役立つようにします。また、ユーザー向けのフロー入力についての説明も含め、提供する値の種類についてわかるようにします。
- 説明で列記する入力および出力の順番は、[入力]タブおよび[出力]タブに表示される順番と同じでなければなりません。
- 説明のベストプラクティスに関する詳細については、「ベストプラクティス」(20ページ)を参照してください。

**注**:ドキュメントの生成機能を使用すると、説明に含まれる情報を作成者やユーザーが参照できるようになります。詳細については、「フローおよびオペレーションに関するドキュメントの生成」 (242ページ)を参照してください。

#### フローを最初から作成

- 1. フローを作成するプロジェクトを開きます。
- 2. フローを作成するフォルダーを開きます。
- 3. [ファイル] > [新規] > [フロー] を選択します。
- 4. フローの分かりやすい名前を入力して [OK] をクリックします。

注:名前の長さは最大 128 文字までです。大文字と小文字は区別されません。

#### テンプレートを使用したフローの作成

Studio で利用できるテンプレートには、使用頻度が高いいくつかのタスクを実行するフローが用意され ています。たとえば、サービスを再開するためのフローを作成するRestart Serviceテンプレートがありま す。

1. フローを作成するプロジェクトを開きます。



- 2. [ようこそ] 画面で、[新規フロー] 「ジョンをクリックします。
- 3. 表示されたテンプレートの一覧でいずれかのフローテンプレートを強調表示すると、その説明が表

利用	できるフローテンプレート		_	テンプレートの説明:
	名前 ▽	パス		このフローを使用して Windows サービスを再起動します。入力を 📥
1 (P	Restart Solaris Server	/Base [1.0.121]/Library/ 🔺		簡単に変更して、特定のサービスに固有のパージョンを作成する
- <b>4</b> 2	Restart Service	/Base [1.0.121]/Library/		か、セキュリティ資格情報の標準セットを使用するようにできま
1 (P	Network Check	/Base [1.0.121]/Library/		9。 人ガハフメーター:NOSt-サービスを再起動 9 の必要がめのホーフト comunico 再起動するが更があるサービス albusov ホフト
1	Counter Average Thresh	/Base [1.0.121]/Library/		スト。 servuice 再起動する必要がのなり「こス。 alluser - かスト に接続するためのコーザータ (個) domain/username), altnass - ホ
1 (P	Check Windows CPU	/Base [1.0.121]/Library/		スト接続用のパスワード。 レスポンス: failure -サービスが存在し
1	Check for Windows Event	/Base [1.0.121]/Library/		ないか、そのステータスを確認できないか、起動できませんでし
				た。 success -サービスが起動されました。
		-		×

示されます。必要に応じて、スクロールバーをドラッグしてテキストを表示します。

- 4. 目的に合ったフローテンプレートを選択し、[作成] 「バタンをクリックします。
- 5. 新しいフローが、選択したプロジェクトの Library フォルダーの My Ops Flows フォルダーに作成されます。別のフォルダーにフローを保存する場合は、フローを目的のフォルダーにドラッグするか、 [編集] > [切り取り] および [編集] > [貼り付け] コマンドを使用します。

**ヒント:** テンプレートを [コンテンツパック] ペインの OO-Base Content コンテンツパックの Templates フォルダーから入手 することもできます。テンプレートをダブルクリックして作成ペインで開きます。フ ローを修正する場合は、プロジェクトにコピーしてコピーを修正します。

**注**: テンプレートを別のフローから作成するには、プロジェクトの Library に Templates フォルダー を作成し、そのフォルダーにフローを保存します。これらのフローは、フローテンプレートのリストに表示されます。



#### フローの説明の追加

1. 作成ペイン上でフローを右クリックし、[プロパティ]をクリックします。

- 2. [説明]タブをクリックします。
- 3. フローの説明を入力して [OK] をクリックします。

## 参考資料

#### 利用できるフローテンプレート

HP OO の [ようこそ] 画 面 で [新規フロー] ボタンをクリックすると、 [利用できるフローテンプレート] リストが表示されます。



GUI ፖイテム	説明
名前	利用できるテンプレートの名前が表示されます。名前を強調表示 すると、説明が[ <b>テンプレートの説明</b> ] ボックスに表示されます。
パス	それぞれのテンプレートが保存されているフォルダーのパスが表示されます。
テンプレートの説明	選択したテンプレートの説明が表示されます。

## フロー内でのステップの作成

オペレーションからステップを作成する場合、ステップはオペレーションのインスタンスとなるため、オペレーションの入力、出力、参照、その他の特性を継承します。

作成ペインにオペレーションをドラッグし、オペレーションからステップを作成します。

- オペレーションを [コンテンツパック] ペインからドラッグすると、ステップを変更することができます。ただし、[コンテンツパック] ペインは読み取り専用であることに注意してください。
- ステップを作成する前にオペレーションを変更する場合、[コンテンツパック]ペインからコピーし、[プロ ジェクト]ペインに貼り付けてから、作成ペインにそのオペレーションをドラッグします。

### ベストプラクティス

- ステップを作成する際、ステップの作成元のオペレーションまたはフローの説明を、[説明]タブに追加することをお勧めします。説明には、ステップを検索するときに役立つ検索語を含めるようにし、ステップの入力、レスポンス、結果についての説明を含めます。説明のベストプラクティスに関する詳細については、「ベストプラクティス」(20ページ)を参照してください。
- 開始ステップはフローの左上隅に配置するべきです。ただし、次の場合は例外です。
  - 開始ステップに複数のレスポンスがあり、それぞれが別のステップに進む場合。
  - 開始ステップを左上隅に配置すると、トランジションが交差するなどして表示が複雑になりすぎる場合。
- ステップの名前を変更する場合、ステップの目的がはっきりとわかるような名前にしてください。
- コールアウトを使用してステップに関する情報を追加することもできます。
- オペレーションのプロパティをカスタマイズする必要がない場合、ステップのコピーではなく元の読み 取り専用バージョンを使用します。
- 統合を行っている場合、使用されているAPIからの元のオペレーション名を保持します。

### 実行する操作

#### オペレーションからのステップの作成

1. [プロジェクト] ペインまたは [コンテンツパック] ペインで、フローに追加 するオペレーションを選択します。

注: [コンテンツパック] ペインのオペレーションは読み取り専用です。

- 2. プロジェクトツリーから作成ペインにオペレーションをドラッグします。
- 3. 必要に応じて、フロー内での機能を表すような名前を付けます (オペレーションの名前は、総称 的すぎる場合 があります)。
  - a. 名前を変更するステップを右クリックし、[名前の変更]を選択します。
  - b. 強調表示されたフィールドに新しい名前を入力します。
- 4. 必要に応じて、ステップを編集します。詳細については、「フローの変更」(113ページ)を参照してください。

#### 読み取り専用オペレーションを編集可能にするためのプロジェクトへのコピー

1. [コンテンツパック] ペインで、コピーするオペレーションを選択します。

- 2. [編集] > [コピー]を選択します。
- 3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集] > [貼り付け]を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。
- 4. 必要に応じて、オペレーションを編集します。

**注**: オペレーションを編集する場合、このオペレーションから作成されたすべてのステップに、このプロパティへの変更が引き継がれます。ステップを編集する場合は、元のオペレーションには影響しません。

5. プロジェクトツリーから作成ペインにオペレーションをドラッグします。

注: コンテンツパックからプロジェクトにコピーされたオペレーションは、「ソフトコピー」です。つまり、オ ペレーションが元々、アクションプラグインからインポートされたものである場合、コピーされたオペ レーションは元のオペレーションへの参照を保持したままです。アクションプラグインをアップグレード して、新しいバージョンを呼び出すように元のオペレーションを更新すると、コピーされたオペレー ションも自動的に更新されます。詳細については、「オペレーションの作成」(225ページ)を参照 してください。

#### フロー内からのステップのコピー

ステップをコピーして貼り付けるには、次のいずれかのツールを使用します。

● 作成ペインツールバーの[コピー] → ボタンと[貼り付け] = ボタン

[編集] > [コピー] メニューコマンドと[編集] > [貼り付け] メニューコマンド

- 右クリックメニュー
- キーボードの組み合わせ(CTRL+C、CTRL+V)

#### ステップへの説明の追加

- 1. 作成ペイン上でステップを右クリックし、[プロパティ]をクリックします。
- 2. [説明] タブをクリックします。
- 3. ステップの説明を入力して[OK]をクリックします。

説明入力のベストプラクティスについては、「ベストプラクティス」(20ページ)を参照してください。

#### ステップへのコールアウトの追加

コールアウトには、ステップに関する情報を入力します。次のような情報を入力すると、フローの使い 勝手が格段に向上します。

- データの移動:ステップ間で情報を渡す方法
- データを格納するフロー変数の名前
- 入力データに必要な形式



- 1. ステップパレットを表示するには、作成ペインのツールバーで [ステップパレット] ボタン をクリックします。
- 2. [コールアウト] 🤍 ボタンをクリックして、選択したコールアウトを作成ペインにドラッグします。
- 3. コールアウトのテキストを入力します。
- 4. コールアウトをステップに接続するために、グレーの円 📍 からステップにドラッグします。
- 5. コールアウトのテキストエリアの角をドラッグしてサイズ変更します。

#### フローからステップを作成する (サブフロー)

サブフローとは、フロー内フローです。詳細については、「フロー内でのサブフローの作成」(183ページ) を参照してください。

- 1. 親フローを作成ペインで開きます。
- 2. ライブラリでステップ(またはサブフロー)として使用するフローを選択します。
- 3. ライブラリから作成ペインの親フローにフローをドラッグします。ドラッグしたフローは、親フローのス テップになります。

注: [Accelerator Packs] フォルダーには、親フローとして実行可能なフローが保存されています。

#### 手動ステップを作成する

手動ステップは、操作を選択できるステップです。ユーザーが、実行時に操作を選択する必要があり ます。 手動ステップを作成するには、手動オペレーションテンプレートを、ベースコンテンツからコピーして、ユー ザーが選択できる操作を定義します。

- 1. [コンテンツパック] ペインで、手動オペレーションテンプレートを選択します。
- 2. [編集] > [⊐ピー]を選択します。
- 3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集] > [貼り付け]を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。
- 4. プロジェクトツリーから作成ペインにオペレーションをドラッグします。
- 5. ステップで、ユーザーが選択できる操作を追加します。



**注**: ステップではなくオペレーションのプロパティで操作を追加することもできます。こうすると、別の フローでもこのオペレーションを使用することができます。

#### 表示ステップの作成

表示ステップとは、ポップアップのプロンプトメッセージに情報を表示しますが、ほかの操作が実行されないステップです。ユーザーは、実行時に[**続行**]を選択する必要があります。

表示ステップを作成するには、表示オペレーションテンプレートをベースコンテンツからコピーして、ユー ザーに表示する情報を定義します。
プロンプトメッセージに変数を含めることができます。たとえば、前のステップが終了した時間をユーザー に通知するよう、メッセージに日付/時刻の変数 (\${dateTime})を含めることができます。

- 1. [コンテンツパック] ペインで、表示オペレーションテンプレートを選択します。
- 2. [編集] > [コピー]を選択します。
- 3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集] > [貼り付け]を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。
- 4. プロジェクトツリーから作成ペインにオペレーションをドラッグします。
- 5. ステップのインスペクターを開いて、[表示]タブをクリックします。
- 6. [このステップの実行前に常にユーザーに確認する] チェックボックスをオンにします。
- 7. [プロンプトタイトル] ボックスに、プロンプトのラベルを入力します。
- 8. [プロンプトの幅] ボックスに、プロンプトの幅をピクセル単位で入力します。
- 9. [高さ] ボックスに、プロンプトの高さをピクセル単位で入力します。
- 10. [**プロンプトテキスト**] ボックスに、ユーザーへのメッセージを入力します。
- 11. [OK]をクリックし、変更を保存します。

**注**: ステップではなくオペレーションのプロパティで表示情報を追加することもできます。こうすると、別のフローでもこのオペレーションを使用することができます。

# 参照情報

#### ステップのインスペクター > [表示] タブ

ステップのインスペクターの [表示] タブで、ユーザーに表示するユーザープロンプトを作成します。

ステップ名: SQL Query				
入力  結果 表示 説明   詳細   スクリプトレット				
── このステップの実行前に常にユーザーに確認する				
ว้อมวัหรัสหมะ				
プロンプトの幅: 0	高さ <sup>0</sup>			
プロンプトテキスト				

GUI ፖイテム	説明			
このステップの実行前に常に ユーザーに確認する	ステップを実行 するたびにプロンプト ウィンド ウを表 示 するには、 チェックボックスをオンにします。			
プロンプトタイトル	プロンプトウィンドウのタイトルバーに表示するラベルを入力します。			
プロンプトの幅	プロンプトウィンドウの幅をピクセル単位で入力します。			
ち高	プロンプトウィンドウの高さをピクセル単位で入力します。			
プロンプトテキスト	プロンプトウィンドウの本文に表示するメッセージを入力します。メッ セージに変数を含めることができます。たとえば、\${dateTime}のようになります。			

# フローの外観の調整

フローを作成するときに、[表示オプション] パレットを使用して、作成ペインでフローを整理し、外観を調整できます。

作成ペインのツールバーから [表示オプション] ボタン をクリックして、 [表示オプション] パレットを表示します。



# 実行する操作

## ステップのグリッド へのスナップ

グリッドにオブジェクトをスナップすると、素早くオブジェクトを整列させ整えることができます。

- 作成ペインの背景にグリッドが表示されていない場合は、[表示オプション] パレットの[グリッドの 表示/非表示] ボタン をクリックします。作成ペインにオペレーションをドラッグすると、グリッドの 一番近い線にスナップされます。
- 2. グリッドのある線から別の線にステップを動かすには、ステップを少し動かしてマウスを放します。

### ステップの整列

- 1. 選択したステップを水平方向に整列するには、1つまたは複数のステップを選択してから[表示 オプション]の[選択項目を水平方向に整列] を選択します。
- 2. 選択したステップを垂直方向に整列するには、1つまたは複数のステップを選択してから[表示 オプション]の[選択項目を垂直方向に整列] を選択します。

#### レスポンスラベルおよびアイコンの表示または非表示

オペレーションのレスポンスラベルとアイコンがいっぱいあるためにフローが込みすぎている場合、ラベルやアイコンを選択して非表示にできます。

- 1. レスポンスラベルを表示または非表示にするには、[ラベルの表示/非表示] 「 ボタンをクリック」します。これにより、レスポンスラベルの表示と非表示を切り替えることができます。
- レスポンスアイコンを表示または非表示にするには、[接続されたレスポンスアイコンの表示/非表示]
   ボタンをクリックします。これにより、レスポンスアイコンの表示と非表示を切り替えることができます。

### オブジェクトの前面または背面への移動

フローでは、作成ペインでオブジェクトを重ねることができます。フロー内に多くのアイテムがある場合、

フローが長くなる場合があります。このような場合、最も重要なオブジェクトを前面に移動する必要があります。

- 1. オブジェクトをスタックの前面に移動するには、オブジェクトを選択してから[最前面へ移動] をクリックします。
- 2. オブジェクトをスタックの背面に移動するには、オブジェクトを選択してから[最背面へ移動] 4

参照情報

## [表示オプション] パレット

[表示オプション] パレットには、作成ペイン上のフローの外観を変更するためのアイコンがあります。

P 2 = 🔟 🖬 🖬	
	100%
100%表示 に戻す ズームを変更するスライド	<del>.</del>

ボタン	説明
ラベルの表示/非表示 🧜	オブジェクトのレスポンスラベルを表示または非表示にします。
接続されたレスポンスアイコンの 表示/非表示	オブジェクトのレスポンスアイコンを表示または非表示にします。
選択項目を水平方向に整列 	選択項目を水平方向に整列します。
選択項目を垂直方向に整列	選択項目を垂直方向に整列します。
最前面へ移動 描	選択したオブジェクトを前面に移動します。
最背面へ移動 🛄	選択したオブジェクトを背面に移動します。
グリッドの表示/非表示	ステップを配置するときに利用できる作成ペインのグリッドを表示 します。 ステップのドラッグ操作を止めると、 グリッド上の一番近い 位置にそのステップがスナップされます。

フローの変更

フローを作成したら、フローを変更できます。たとえば、前に作成したフローをコピーして、多少異なる 用途に適応できます。HP OO で提供されている、[Accelerator Packs] フォルダー内にあるようなすぐ に使えるフローを、ニーズに合わせて適応することもできます。

# ベストプラクティス

変更する前に、必ずフローをコピーしてください。

フローを変更する前に、[参照] > [これを使用する項目は?]を使用して、ほかのフローで使用されていないかどうかを確認します。

フローをコピーするときに、オペレーションのプロパティの変更が必要になる可能性がある場合は、 [ディープコピー] コマンドを使用することをお勧めします。このコマンドを使用すると、フローとともにオペレーションがコピーされるため、元の内容に影響を与えずに変更できます。「フローおよびオペレーションのコピー」(237ページ)を参照してください。

[**ディープコピー**] コマンドを使ってフローをコピーする場合、フローおよびフローのオペレーションに新しいフォルダーを作成します。

注意:ステップの変更とオペレーションの変更の違いを理解していることを確認してください。

- ステップのプロパティを変更する場合 (ステップのインスペクターを使用)、個々のステップにのみ 影響します。
- オペレーションのプロパティを変更する場合 ([プロパティ] シートを使用)、このオペレーションをステップとして使用するすべてのフローに影響します。オペレーションのプロパティを変更する場合、そのオペレーションを使用するほかのフローが破損する可能性があるため、非常に慎重に行う必要があります。

# 実行する操作

#### 開始ステップを変更する

フローの開始に新しいステップを追加すると、フローの開始ステップがまだ定義されていないため、その ステップは警告アイコンとともに表示されます。

フローを開始するのに使用するステップを右クリックし、[開始ステップの設定]を選択します。

## ステップの名前を変更する

- 1. 名前を変更するステップを右クリックし、[名前の変更]を選択します。
- 2. 強調表示されたフィールドに新しい名前を入力し、Enterキーを押して保存します。

ベストプラクティス:ステップの目的を明確に表す名前を指定します。

#### フローまたはオペレーションの名前を変更する

オペレーションの名前を変更する場合、このオペレーションがほかのフローで使用されていないかどうか を確認します。使用されている場合、オペレーションをコピーして、コピーの名前を変更することをお勧めします。

- 1. [プロジェクト] ペインで、名前を変更するフローまたはオペレーションを右クリックし、[名前の変更] を選択します。
- 2. 強調表示されたフィールドに新しい名前を入力し、Enterキーを押して保存します。

ベストプラクティス: フローまたはオペレーションの目的を明確に表す名前を指定します。

#### フロー図のステップのグループを移動する

- 1. Shift キーまたは Ctrl キーを押しながらステップのグループを選択します。
- 2. ステップをクリックし、グループとしてドラッグします。

### アイコンを変更する

ステップ、オペレーション、またはフローのアイコンを、その要素を分かりやすく表すものに変更できます。

1. [アイコン] ペインを開いて [アイコン] タブをクリックします。



- 2. 必要なアイコンを示す、アイコンのグループ名を選択します。
- 3. アイコンを選択し、ステップ、オペレーション、またはフローにドラッグします。

**注**: Ctrl キーを押しながらアイコンをドラッグし、アイコンを既存のアイコンの上に重ねて配置できます。

ベストプラクティス:種類に応じてフローを分類している場合、フローの各種類に特定のアイコンを使用します。

## ステップのインスペクターでステップを変更する

- 1. ステップを右クリックし、[プロパティ]を選択します。ステップのインスペクターが開きます。
- 2. ステップのインスペクターでステップを変更できます。
  - ステップの入力を追加または編集する。詳細については、「入力の作成」(118ページ)を参照してください。

- ステップの結果を追加または編集する。詳細については、「オペレーションの出力の設定」 (147ページ)を参照してください。
- ステップの説明を追加または編集する。詳細については、「フロー内でのステップの作成」(104 ページ)を参照してください。
- ステップのユーザープロンプトを追加または編集する。詳細については、以下の「ステップのユー ザープロンプトを表示する」を参照してください。
- ステップのスクリプトレットを追加または編集する。詳細については、「フロー内でのスクリプトレットの使用」(198ページ)を参照してください。

**ヒント**: ステップ間でフォーカスを移すときに、インスペクターを閉じてから再度開かずに済むように、 インスペクターを開いたままにするには、インスペクターのタイトルバーの右端にある[**固定**]ボタン ♀ をクリックします。

## ステップのユーザープロンプトを表示する

ステップの実行前に表示する、ユーザープロンプトを作成できます。 プロンプトメッセージに変数を含めることができます。 たとえば、 前のステップが終了した時間をユーザーに通知するよう、メッセージに日付/時刻の変数 (\${dateTime})を含めることができます。

- 1. ステップを右クリックし、[プロパティ]を選択します。
- 2. ステップのインスペクターで [表示] タブをクリックします。
- 3. [このステップの実行前に常にユーザーに確認する] チェックボックスをオンにします。
- 4. [プロンプトタイトル] ボックスに、プロンプトのラベルを入力します。
- 5. [プロンプトの幅] ボックスに、プロンプトの幅をピクセル単位で入力します。
- 6. [高さ] ボックスに、プロンプトの高さをピクセル単位で入力します。
- 7. [プロンプトテキスト] ボックスに、ユーザーへのメッセージを入力します。
- 8. [OK] をクリックし、変更を保存します。このステップに、表示プロンプトを示す青い矢印が付与されます。



### ステップのベースとなるオペレーションを変更する

たとえば、既存のフローステップを別のオペレーションに関連付けながら、そのステップの既存のトランジ ション (ステップからとステップへのトランジション)を保持する必要がある場合があります。

- 1. ステップを右クリックし、[プロパティ]を選択します。
- 2. ステップのインスペクターで [詳細] タブをクリックします。
- 3. [ソースオペレーション] で、[選択] ボタンをクリックします。
- 4. [オペレーションの選択]ダイアログボックスで、ステップのベースとなるオペレーションに移動して選択し、[OK]をクリックします。
- 5. オペレーションが変更されたことを表すよう、ステップの名前を変更します。
- 6. 入力の値の割り当てに必要な変更を確認して実行し、以前のオペレーションの入力と新しいオペレーションの入力の違いが反映されるようにします。

# 参照情報

### ステップのインスペクター > [表示] タブ

ステップのインスペクターの [**表 示**] タブで、ステップの実行前に表示するユーザープロンプトを作成できます。

 ステップ名:
 SQL Query

 入力 |結果 表示 説明|詳細 | スクリプトレット |

 このステップの実行前に常にユーザーに確認する

 プロンプトタイトル:

 プロンプトの幅:
 0

 アロンプトテキスト:

GUI ፖイテム	説明
このステップの実行前に常に ユーザーに確認する	ステップを実行するたびにプロンプトウィンドウを表示するには、 チェックボックスをオンにします。
プロンプトタイトル	プロンプトウィンドウのタイトルバーに表示するラベルを入力します。
プロンプトの幅	プロンプトウィンドウの幅をピクセル単位で入力します。

ち高	プロンプトウィンドウの高さをピクセル単位で入力します。
プロンプトテキスト	プロンプトウィンドウの本文に表示するメッセージを入力します。メッ セージに変数を含めることができます。たとえば、\${dateTime}のようになります。

# 入力の作成

入力では、フロー内のステップが必要なデータを取得する方法とタイミングを指定します。たとえば、 [Network Check] フローでは、最初のステップはサーバーに対して ping を実行するため、 ping 対象の サーバーの IP アドレスを必要とします。 IP アドレスは入力によって提供されます。

各入力は変数にマッピングされます。その値は以下の方法で設定できます。

- ユーザープロンプトを作成して、フローを実行する人がフローの開始時に値を入力するようにする。
- 入力の値を特定の固定の値に設定する。
- 値を別のステップから取得するように設定する。
- フロー変数を入力に割り当てる。フロー変数は、フロー全体で使用できる変数とデータ値の集合 に含まれます。

フロー、オペレーション、またはステップに入力を作成できます。

注意:ステップの変更とオペレーションの変更の違いを理解していることを確認してください。

- ステップのプロパティを変更する場合 (ステップのインスペクターを使用)、個々のステップにのみ 影響します。
- オペレーションのプロパティを変更する場合 ([プロパティ] シートを使用)、このオペレーションをステップとして使用するすべてのフローに影響します。オペレーションのプロパティを変更する場合、そのオペレーションを使用するほかのフローが破損する可能性があるため、非常に慎重に行う必要があります。

# ベストプラクティス

- オプションの入力と必須の入力を区別しながら、可能なすべての入力をフローの説明で定義します。
- ステップ内のオプションの入力が必須でない場合、これらの入力を削除します。
- 可能な限り、入力のユーザー選択リストを作成します。これにより、入力ミスが原因で生じるエラーを防ぐことができます。
- 大文字と小文字に一貫性があるようにします。たとえば、すべての入力名にキャメルケースを使用します。

- 統合を行っている場合、使用されている API からの元の入力名を保持します。
- オペレーションの必須の入力は無効にしないようにします。無効にすると、そのオペレーションのすべてのインスタンスで無効になるためです。個々のステップを編集する必要があります。
- 順序ルールに従って、一貫した順序で入力を追加します。次に例を示します。
  - 分かりやすい分類または論理的な分類に基づく順序
  - 重要度に基づく順序(最初に必須の入力)
  - アルファベット 順

# 実行する操作

#### 入力を作成する

- 1. [プロパティ] シートまたはステップのインスペクターを開きます。
  - オペレーションに入力を追加するには、[プロジェクト]ペインでオペレーションを右クリックし、[プロ パティ]を選択します。
  - フローに入力を追加するには、[プロジェクト]ペインでフローを右クリックし、[プロパティ]を選択します。
  - ステップに入力を追加するには、作成ペインでステップをダブルクリックします。
- 2. [入力] タブを選択し、[入力の追加] ボタンをクリックします。

入力 結果 表示 説明 詳細 スクリプトレット  ▲ ▼   入力の概要 (1)							$\leftrightarrow$ ×	
	入力の追加	入力の削除		•		+	]	
	入力		必須	種類			ソース	
host				単一値	Ψ.	値:		<b>4</b>
service			V	単一値	-	値:		<b>\$</b>
user				単一値	Ψ.	値:		<u></u>
password				単一値	*	値:		4

3. 新しい入力の名前を入力して、[OK]をクリックします。入力が新しい行に表示されます。

**注**:入力の名前に "service" または "sp" は使用しないでください。これらの名前を使用すると、特定の状況でフローを実行した場合にエラーが発生する可能性があります。

- 4. (オプション) ステップが機能する上で入力を必須にするには、新しい行で[必須] チェックボックスを オンにします。
- 5. [種類] リストで、入力の値を取得する方法を指定します。

- 単一値
- 値のリスト 複数のターゲットに対してオペレーションを実行できます。
- 6. 該当する行の端にある右向き矢印 🔯 をクリックして、入力エディターを開きます。

入力  結果  表示  説明  詳細  スクリプトレット	
▲ ▼ 入力の概要 > filesToAdd	↓ ► ×
名前: filesToAdd 入力の種類: 単	→値
「入力データフロー	「未指定時: ユーザーへのプロンプト」構成
変数から割り当で files ToAdd  マ	入力項目: ③ テキスト 〇 選択
⇒ 未指定時: ユーザーへのプロンプト	ユーザーメッセージ:
変数への割り当て: files TaAdd 💌	
「入力のプロパティーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー	
□ 暗号化 □ 必須	
検査形式:         <未検証>	
次の名前で記録:「〈実行履歴〉	
	,

- 7. 入力エディターで入力ソースを指定します。
  - 入力と同じ名前の変数から値を割り当てるには、[変数から割り当て]ボックスに表示される デフォルト名を適用します。
  - 別のフロー変数から値を割り当てるには、[変数から割り当て]ボックスに変数名を入力します。

たとえば、最初のステップで情報を検索し、この情報をフロー変数として保存してから、2 番目のステップでその情報を表示できます。2番目のステップでは、最初のステップで作成したフロー変数が使用されます。

- 8. [未指定時] リストから、[変数から割り当て] ボックスに指定したフロー変数が存在しない場合、 またはこのフロー変数に値が格納されていない場合に発生するアクションを選択します。使用で きるオプションは、次のとおりです。
  - ユーザーへのプロンプト: フローの開始時に、情報を入力するかリストから選択して情報を指定する、ユーザーへのプロンプトを設定します。

**注:** [ユーザーへのプロンプト] オプションは、フローの中間のステップには選択しないでください。フローの途中のユーザープロンプトは、このバージョンでは入力ソースとしてサポートさ

れていません。

- 定数を使用する:入力に使用する定数値を入力します。たとえば、常に使用する IP アドレスを入力します。
- 前のステップの結果を使用する:入力に値がない場合に使用する、前のステップの結果を選択します。
- システムアカウント:システムアカウント名と資格情報の種類を入力します。
- **ログインしたユーザーの資格情報**: ログインしたユーザーのユーザー名またはパスワードを入力します。

さまざまなタイプの入力ソースの詳細については、「入力ソースの指定」(125ページ)の関連する タスクを選択してください。

9. デフォルトでは、Studio によって入力と同じ名前のフロー変数が作成されます。この変数をフロー 内の後続のステップで使用できます。[**変数への割り当て**] ボックスでこの名前を変更できます。

たとえば、ユーザーによるパスワード入力が必要なステップがある場合、簡単に識別できるように、変数に password という名前を指定できます。

10. (オプション)入力の値を暗号化するには、[暗号化] チェックボックスをオンにします。フローの実行時に、入力はアスタリスクの並びとして表示されます。

### 入力を削除する

- 1. [プロパティ] シート (フローまたはオペレーションの場合) またはステップのインスペクター (ステップの 場合)を開きます。
- 2. [入力] タブで、削除する入力を選択して [入力の削除] をクリックします。削除したデフォルトの入力は、グレーのイタリック体で表示されます。

#### 削除したデフォルトの入力を復元する

デフォルトの入力を[**入力**] タブから削除した場合、この入力を復元できます。 デフォルトの入力は、 ステップのベースとなるオペレーションの一部として作成された入力です。 削除したデフォルトの入力 は、入力のリストにグレーのイタリック体で表示されます。

- 1. [入力]タブで、[入力の追加]をクリックします。
- 2. 復元する入力の正確な名前を入力します。
- 3. [OK] をクリックします。

#### 必須の入力を無効にする

ー 部 のオペレーションには必 須 の入 力 があり、[必須] チェックボックスをオフにすることはできません。た

だし、ステップで入力が必要ない場合、この入力を無効にできます。

**注意:** オペレーションの必須の入力は無効にしないようにします。無効にすると、そのオペレーションのすべてのインスタンスで無効になるためです。個々のステップを編集する必要があります。

- 1. ステップで、必須の入力の入力エディターを開きます。
- 2. [未指定時] リストで、[定数を使用する] を選択します。ただし、[「未指定時: 定数を使用する」 構成] セクションは空白のままにします。入力が無効になります。

# 参照情報

## ステップのインスペクター > [入力] タブ

ステップのインスペクターの[入力]タブで、フロー内のステップが必要なデータを取得する方法とタイミングを指定します。

	インスペクター								-	÷Þ
ľ	ステップ名: Change	Service Status								
l	入力 結果 表	示 説明 詳細	スクリプトレット							
l	▲ ▼ 入力の概要	Ē							$\leftrightarrow$ ×	
ľ		入力の追加	入力の削除		1		+			
l		入力		必須	種類			ソース		
L	host			V	単一値	Ψ.	値:		42	
L	service			V	単一値	Ψ.	値:		4	
L	requestedstatus			<b>V</b>	単一値	Ψ.	値: Running		42	
L	startdependants			V	単一値	-	値: true		4	
L	user				単一値	Ψ.	値		42	
L	password				単一値	Ψ.	値		42	
	serviceTimeout				単一値	Ψ.	値: 30		42	
L	operationTimeout	t			単一値	Ψ.	値: 30		4	
L										

GUI アイテム	説明
名前	ステップの名前を表示します(読み取り専用)。
UUID	フローまたはオペレーションの一 意の ID を表示します (読み取り専用)。
入力の追加	新しい入力行を追加します。
入力の削除	選択した入力行を削除します。
上に移動 🚹	実行で先に処理されるように、選択した入力行をリスト内の上位 の位置に移動します。
下に移動 🤳	実行で後に処理されるように、選択した入力行をリスト内の下位 の位置に移動します。
入	その行の入力に対して入力エディターを表示します。

入力列	入力名を表示します。		
必須列	この入力を必須にします。		
種類列	[ <b>種類</b> ]リストで、入力の値を取得する方法を指定します。次のオ プションを選択できます。		
	• 単一値		
	• 値のリスト - 複数のターゲットに対してオペレーションを実行できます。		
	• 資格情報 - システムの資格情報またはユーザーの資格情報 から入力データを取得できます。		
ソース列	入力の値の取得先を指定します。		

# [プロパティ] シート > [入力] タブ

[プロパティ] シートの[入力] タブで、フローまたはオペレーションが必要なデータを取得する方法とタイミングを指定します。

名前: Restart Service		
UUID: bf3b0bce-f043-42a6-8a12-9d67378fc9f4		
カテゴリの割り当て: Windows OS (Base [1.0.121])	5	
入力 出力   レスポンス   説明   スクリプトレット		
▲ ▼ 入力の概要	$\leftrightarrow \vdash \times$	
እ		

		入力の追加		入力の削除		1	+	
	入力		必須	種類		ソース		
host				単一値	-	ユーザーへのプロンフ	ንト	4
service				単一値	-	ユーザーへのプロンプ	ንト	4
altuser				単一値	-	ユーザーへのプロンス	ንト	$\Leftrightarrow$
altpass				単一値	•	ユーザーへのプロンプ	ንト	$\Leftrightarrow$

GUI アイテム	説明
名前	フロー名またはオペレーション名を表示します(読み取り専用)。
UUID	フローまたはオペレーションの一意のIDを表示します(読み取り 専用)。
カテゴリの割り当て	[カテゴリの割り当 て] ダイアログボックスを開くと、フローにカテゴリを 割り当 てできます。

入力の追加	新しい入力行を追加します。
入力の削除	選択した入力行を削除します。
上に移動 🚹	実行で先に処理されるように、選択した入力行をリスト内の上 位の位置に移動します。
下に移動 🤸	実行で後に処理されるように、選択した入力行をリスト内の下 位の位置に移動します。
5	その行の入力に対して入力エディターを表示します。
入力列	入力名を表示します。
必須列	この入力を必須にします。
種類列	[種類]リストで、入力の値を取得する方法を指定します。次の オプションを選択できます。
	• 単一値
	• 値のリスト - 複数のターゲットに対してオペレーションを実行できます。
テンプレート列	入力の値の取得先を指定します。

## 入力エディター

ステップのインスペクター (ステップの場合) または [プロパティ] シート (フローまたはオペレーションの場合) の[入力] タブで入力の基本 パラメーターを設定した後で、入力エディターを使って入力ソースの詳細 を指定します。

入力エディターを表示するには、入力の行の端にある右向き矢印 🖾 をクリックします。

入力は課日表	示   説明   詳細   スクリプトレッ > <b>files Ts Add</b>	۲I		
<ul> <li>名前: filesToAdd</li> <li>入力データフローー</li> <li>変数から割り当て:</li> <li>◆ 未指定時:</li> </ul>	filesToAdd ユーザーへのプロンプト	入力の種類: 単- マ マ	→値 「未指定時: ユ・ 入力項目: ユーザーメッセー	ーザーへのプロンプトJ構成 ・デキスト C 選択 ・ジ:
2.55×の害的当じ、 →入力のプロパティー 検査形式: 次の名前で記録:	■ mes / 0A33 □ 暗号化 <未検証> <実行履歴>	·   必須 ·		

GUI ፖイテム	説明
名前	入力名を表示します。ここで修正できます。
入力の種類	入力の種類を表示します。ここで修正できます。
変数から割り当て	入力のソースとなるフロー変数の名前を入力または選択します。
未指定時	[変数から割り当て]ボックスに指定したフロー変数が存在しない場合、またはこのフロー変数に値が格納されていない場合に発生する アクションを選択します。
「未指定時 : <action>」 構成</action>	[変数から割り当て] ボックスに指定したフロー変数が存在しない場合、またはこのフロー変数に値が格納されていない場合に発生する アクションの詳細を構成します。このセクションは、[未指定時] リストで 選択するアクションによって異なります。
変数への割り当て	入力の値を割り当てるフロー変数を選択します。
暗号化	フローの実行時にアスタリスクの並びとして表示されるように、入力の 値を暗号化します。
必須	この入力を必須にします。
検査形式	入力の値をシステム評価子で検証します。この機能は現在サポート されていません。
次の名前で記録	診断または監査に値を使用できるようにします。この機能は現在サポートされていません。
矢印ボタン▲▼	入力エディターで複数の入力を開いている場合、入力間を移動する には、縦の矢印ボタンをクリックします。
矢印ボタン 😱	入力エディターと[入力の概要]の間を移動するには、横の矢印ボタン をクリックします。

# 入力ソースの指定

フローやオペレーション、ステップでの入力設定時、選択できる入力ソースの指定方法がいくつかあります。入力エディターは、選択内容によって変わります。

このセクションのタスクは、入力を設定するために行うタスクにおける、オプションのサブタスクです。「入力の作成」(118ページ)を参照してください。

注: フローの途中のユーザープロンプトは、このバージョンでは入力ソースとして使用できません。

# 実行する操作

## 単一の固定値を入力ソースに指定

入力ソースを単一の固定値として指定します。たとえば、単一の固定値は、常にステップで使用される IP アドレスなどです。

- 1.入力を作成して、種類を[単一値]に設定します。
- 2. 行の端にある右向き矢印 (😂)をクリックして、その行の入力エディターを開きます。
- 3. デフォルトで、入力に付けた名前が[変数から割り当て]ボックスに表示されます。これは、入力 値のソースとなる同じ名前の変数があるとStudioで見なされるためです。
  - 入力と同じ名前の変数から値を割り当てるには、[変数から割り当て]ボックスに表示される デフォルト名を適用します。
  - 入力とは別の名前の変数から値を割り当てるには、[変数から割り当て] ボックスに変数の名前を入力します。
- 4. [未指定時] リストボックスで [定数を使用する] を選択します。これにより、[変数から割り当て] ボックスで指定したフロー変数が存在しなかったり、その変数に値が保持されていない場合にど うするかを定義します。
- 5. [「未指定時: 定数を使用する」構成] セクションで、入力の値 (false など)を入力します。また、テキストと変数の参照を組み合わせて使用することもできます。 \${variablename}の形式で 指定します。 たとえば、 Ping of \${targethost} succeeded のようになります。

資格情報を入力ソースとして使用し、フローで、システムアカウントの資格情報が必要なタスクを実行することもできます。

入力を作成します。 行の端にある右向き矢印 ()をクリックして、その行の入力エディターを開きます。 [**未指定時**] リストボックスで、次のいずれかを行います。

- 1. [**ログインしたユーザーの資格情報**]を選択する。ログインユーザーが、フローを開始するユーザー アカウントと見なされます。
- [システムアカウント]を選択し、[アカウント名]リストからオペレーションの資格情報で使用するシステムアカウントと、システムアカウントから抽出するプロパティ(パスワードやユーザー名など)を選択します。これにより、こういったアカウントの資格情報が必要なタスクをフローで実行できるだけでなく、資格情報をシステムアカウント名で隠しておけるので資格情報の漏洩を防ぐことにもなります。システムアカウントの詳細については、「システムアカウントの設定」(86ページ)を参照してください。

**注:**[ユーザーへのプロンプト]オプションは選択しないでください。ユーザーへのプロンプトは、このバージョンではサポートされていません。

### 単一のユーザー入力テキストを入力ソースに指定

フローが動作するために必要な情報をユーザーが提供する必要がある場合に、入力ソースを、ユー ザー入力テキストとして指定します。たとえば、ユーザーが使用しているサーバーの IP アドレスを、フ ローの開始時に指定するようにすることができます。

**注:** フローの途中のステップで入力ソースをユーザー入力テキストとして指定しないでください。フ ローの途中のユーザーへのプロンプトは、このバージョンではサポートされていません。

- 1. 入力を作成して、種類を[単一値]に設定します。
- 2. 行の端にある右向き矢印 (😂)をクリックして、その行の入力エディターを開きます。
- 3. [未指定時] リストボックスで [ユーザーへのプロンプト] を選択します。
- 4. [「未指定時: ユーザーへのプロンプト」構成] セクションで [テキスト] を選択します。
- 5. [ユーザーメッセージ] ボックスに、ユーザーに入力してもらうデータの内容を知らせるプロンプトメッ セージを入力します。

#### 単一のユーザー選択を入力ソースに指定

ユーザーに入力を提供するもう1つの方法に、ユーザーが選択できるリストを提示する方法があります。たとえば、ユーザーは場所の選択をフローの開始時に行うようにすることができます。

**注:** フローの途中のステップで入力ソースをユーザー選択として指定しないでください。フローの途中のユーザーへのプロンプトは、このバージョンではサポートされていません。

- 1. 入力を作成して、種類を[単一値]に設定します。
- 2. 行の端にある右向き矢印 (🔤)をクリックして、その行の入力エディターを開きます。
- 3. [未指定時]リストボックスで[ユーザーへのプロンプト]を選択します。
- 4. [「未指定時: ユーザーへのプロンプト」構成] セクションで [選択] を選択します。
- 5. [リストソース] リストで、次のいずれかを選択します。
  - [選択リスト] あらかじめ定義されているリストのセットから選択する。

[名前]リストで、ユーザーに提示するリストを選択します。

**ヒント**: リストを作成し、事前定義済みリストのセットに追加することもできます。 リストの作成方法については、ユーザープロンプトの選択リストの作成を参照してください。

■ [ドメインターム] - ドメインタームは、専用の選択リストです。例として、特定の種類のサーバー に対してフローを実行するように指定する場合、システム内の各種サーバーのドメインタームを 追加して、あるフローの実行対象にするサーバーの種類をユーザーが選択できるようなユー ザープロンプトを作成することが可能です。

[名前]リストで、ユーザーに提示するドメインタームリストを選択します。

■ [フロー変数]-フロー変数の内容からリストを作成します。

[名前] リストで、リストに含むフロー変数を入力または選択します。

[ソース区切り記号]ボックスに、リスト内の各要素を区切る記号を入力します。

6. [**ユーザーメッセージ**] ボックスに、ユーザーに入力してもらうデータの内容を知らせるプロンプトメッ セージを入力します。

#### 単一の入力値のリストを入力ソースに指定

固 定 値 のリストを作 成 することにより、1 つのステップを複 数 のターゲット で実 行 できるようになります。 たとえば、 複 数 のマシンを対 象 にしたオペレーティングシステムのヘルスチェックやソフト ウェアアップデート のインストールを実 行 することができます。

- 1. 入力を作成して、種類を[値のリスト]に設定します。
- 2. 行の端にある右向き矢印 (🖾)をクリックして、その行の入力エディターを開きます。
- [入力のプロパティ]の下にある[入力区切り記号]ボックスに、リスト内の各要素を区切る記号を 入力します。
- 4. [未指定時] リストボックスで [定数を使用する] を選択します。
- 5. [「未指定時: 定数を使用する」構成] セクションの [定数値] ボックスで次のいずれかの操作を行います。
  - 入力の値を、[入力区切り記号]ボックスで入力した記号で区切って入力します。
  - 1つ以上のフロー変数参照を次の形式で入力します。

\${flowvariablename1}<区切り記号>\${flowvariablename2}

**注:**入力した値と変数を同じリストに含めることが可能です。\${flowvariableA} |\${flowvariableB}|10.2.0.200|18.35.100.7のようにします。

この例では[flowvariableA]には220.220.3.9、[flowvariableB]には10.51.110.12が入力され、区切り記号は"|"に設定されています。2つの変数名を入力し、別の1つのIPアドレスを手動で[定数値]ボックスに入力すると、オペレーションは、220.220.3.9、10.51.110、1210.2.0.200、18.35.100.7の4つのマシンすべてで実行されます。

### ユーザーが入力したテキストから取得する入力値のリストを入力ソースに指定

この種類の入力では、ユーザーは、区切り記号で区切った値のリストを入力する必要があります。た とえば、フローの開始時に、フローでターゲットへの複数のホストアドレスをユーザーが入力するようにす ることができます。 **注:** フローの途中のステップで入力ソースをユーザー入力テキストとして指定しないでください。フ ローの途中のユーザーへのプロンプトは、このバージョンではサポートされていません。

- 1. 入力を作成して、種類を[値のリスト]に設定します。
- 2. 行の端にある右向き矢印 (📴)をクリックして、その行の入力エディターを開きます。
- 3. [入力のプロパティ]の下にある[入力区切り記号]ボックスに、リスト内の各要素を区切る記号または記号列を入力します。
- 4. [未指定時] リストボックスで [ユーザーへのプロンプト] を選択します。
- 5. [「未指定時: ユーザーへのプロンプト」構成] セクションの[入力項目] で[テキスト] を選択します。
- 6. [ユーザーメッセージ] ボックスに、フローユーザーに対してオペレーションが必要とするデータの種類 を説明するプロンプトのテキストを入力します。

注: プロンプトのテキストは、ユーザーにリストの正しい入力方法を説明する文にします。さらに、正しい区切り記号または記号列を使用するよう注意書きも付け加えてください。区切り記号として使用されていない場合に空白を挿入するとオペレーションの失敗の原因となるので、注意が必要です。

## ユーザーの選択から取得する入力値のリストを入力ソースに指定

ユーザーへのプロンプトで、ユーザーに、複数アイテムを選択可能なリストを提示します。たとえば、フ ローの開始時に、フローでターゲットへのマシンのリストをユーザーが選択する必要がある場合などで す。

**注:** フローの途中のステップで入力ソースをユーザー選択として指定しないでください。フローの途中のユーザーへのプロンプトは、このバージョンではサポートされていません。

- 1. 入力を作成して、種類を[値のリスト]に設定します。
- 2. 行の端にある右向き矢印 (🔄)をクリックして、その行の入力エディターを開きます。
- [入力のプロパティ]の下にある[入力区切り記号]ボックスに、リスト内の各要素を区切る記号を 入力します。
- 4. [未指定時] リストボックスで [ユーザーへのプロンプト] を選択します。
- 5. [「未指定時: ユーザーへのプロンプト」構成] セクションの[入力項目] で[選択]を選択します。

- 6. [リストソース]リストで、次のいずれかを選択します。
  - [選択リスト] あらかじめ定義されているリストのセットから選択する。

[名前] リストで、ユーザーに提示するリストを選択します。

**ヒント**: リストを作成し、事前定義済みリストのセットに追加することもできます。 リストの 作成方法については、ユーザープロンプトの選択リストの作成を参照してください。

[ドメインターム] -ドメインタームは、専用の選択リストです。例として、特定の種類のサーバーに対してフローを実行するように指定する場合、システム内の各種サーバーのドメインタームを追加して、あるフローの実行対象にするサーバーの種類をユーザーが選択できるようなユーザープロンプトを作成することが可能です。

[名前] リストで、ユーザーに提示するドメインタームリストを選択します。

■ [フロー変数]-フロー変数の内容からリストを作成します。

[名前]リストで、リストに含むフロー変数を入力または選択します。

[ソース区切り記号]ボックスに、リスト内の各要素を区切る記号を入力します。

7. [ユーザーメッセージ] ボックスに、ユーザーに入力してもらうデータの内容を知らせるプロンプトメッセージを入力します。

### 前のステップの結果を入力ソースに指定

たとえば、前のステップでは、プロセスが動作するかどうかをテストし、現在のステップの入力はテストの結果を表示するようにすることができます。

- 1. 入力を作成し、入力で複数の値を許可するかどうかによって種類を[単一値]または[値のリスト]のいずれかに設定します。
- 2. [未指定時] リストで[前のステップの結果を使用する]を選択します。
- 3. 入力値が複数値の場合は、[入力のプロパティ]の下にある[入力区切り記号]ボックスに、リスト内の各要素を区切る記号を入力します。

**注**:前のステップの結果に複数のアイテムが含まれている場合、指定した入力の区切り記号が結果の区切り記号と同じでなければなりません。

## 参照情報

## 入力エディター

ステップのインスペクター (ステップの場合) または [プロパティ] シート (フローまたはオペレーションの場合) の [入力] タブで入力の基本 パラメーターを設定した後で、入力エディターを使って入力ソースの詳細 を指定します。

# 入力エディターを表示するには、入力の行の端にある右向き矢印 😂 をクリックします。

入力 結果 | 表示 | 説明 | 詳細 | スクリプトレット |

▲ ▼ 入力の概要 > filesToAdd

名前: <mark>filesToAdd</mark>	 入力の種類:	単一	•値
入力データフローー			「「未指定時: ユーザーへのプロンプト」構成 ―――
変数から割り当て	filesTaAdd	-	入力項目: ⓒ テキスト 〇 選択
🤟 未指定時:	ユーザーへのプロンプト	-	ユーザーメッセージ
変数への割り当て	files ToAdd	-	
入力のプロパティー			
	🗌 暗号化 🗌 必須	湏	
検査形式: [	〈未検証〉	-	
次の名前で記録	〈実行履歴〉	-	
			1

GUI ፖイテム	説明
名前	入力名を表示します。ここで修正できます。
入力の種類	入力の種類を表示します。ここで修正できます。
変数から割り当て	入力のソースとなるフロー変数の名前を入力または選択します。
未指定時	[変数から割り当て]ボックスに指定したフロー変数が存在しない場合、またはこのフロー変数に値が格納されていない場合に発生するアクションを選択します。
「未指定時: <action>」構成</action>	[変数から割り当て]ボックスに指定したフロー変数が存在しない場合、またはこのフロー変数に値が格納されていない場合に発生するアクションの詳細を構成します。このセクションは、[未指定時]リストで選択するアクションによって異なります。
変数への割り当て	入力の値を割り当てるフロー変数を選択します。
暗号化	入力の値を暗号化し、フローを実行するときに値がアスタリスクで表示され るようにします。
必須	この入力を必須にします。
検査形式	入力の値をシステム評価子で検証します。この機能は現在サポートされて いません。
次の名前で記録	診断または監査に値を使用できるようにします。この機能は現在サポートされていません。
矢印ボタン▲▼	入力の概要に複数の入力がある場合、矢印ボタンをクリックして、入力間 を入力エディターで切り替えられます。

 $\bullet \vdash \mathsf{X}$ 

# 入力データの評価

入力の検証には、評価子を使用します。次に例を示します。

- 入力が電子メールアドレスの場合、評価子を使って、入力が正しい電子メールの形式であるか を確認できます。
- 入力が1以上の数値である必要がある場合、評価子を使ってこの場合が当てはまるかを確認できます。

Studioでは、次の値を検証する標準のシステム評価子を用意しています。

- 英数字
- 電子メール
- ファイル名
- IP アドレス
- 空白なし
- 数値
- UUID
- 電話番号

注: デフォルトの電話番号用データ評価子は、北アメリカ内からかける場合の北アメリカの電話 番号形式 (1-nnn-nnnn)のみをサポートします。地域の電話番号形式を検証するには、そ のためのシステム評価子を作成する必要があります。

評価子では次が使用されます。

- =、!=、Begins with、Contains、Match All Words、Match At Least One Wordなどの簡単 な演算子。
- 正規表現(詳細については「フローでの正規表現の使用」(202ページ)を参照)
- スクリプトレット (詳細については、「フロー内でのスクリプトレットの使用」(198ページ)を参照)

# 実行する操作

#### 評価子を使用した入力の検証

入力エディターで入力を作成している時に、[検査形式]リストから評価子を選択して入力値を検 証できます。

<ul> <li>入力 出力   レスポンス   説明   スクリプトレット  </li> <li>▲ ▼ 入力の概要 &gt; excelFileName</li> </ul>	<b>∢</b>
名前: excelFileName	入力の種類: 単一値
「人力データフロー」	
変数から割り当て:excelFileName	🗾 入力項目: 💿 テキスト 🔿 選択
└∽ 未指定時: ユーザーへのプロンプト	■ ユーザーメッセージ:
変数への割り当て: excelFileName	
入力のプロパティーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー	
□ 暗号化	☑ 必須
検査形式: MyEvaluator	
次の名前で記録:〈実行履歴〉	

**次の名前で記録**:診断または監査に値を使用できるようにします。この機能は現在サポートされていません。

入力の作成については、「入力の作成」(118ページ)を参照してください。

# トランジションの作成

トランジションを使用して、フロー内の2つのステップを接続します。トランジションの起点はステップのいずれかのレスポンス (success、failureなどのレスポンスアイコンで示される) で、そこから別のステップに 遷移します。フロー内のすべてのレスポンスに、後続のステップへのトランジション、またはフロー全体の 結果を戻してフローを終了するリターンステップへのトランジションが必要です。



1つのステップに複数のレスポンスを接続できます。たとえば、失敗のレスポンスが複数ある場合、1つの失敗リターンステップに接続されることがほとんどです。

オペレーションのレスポンスの設定の詳細については、「レスポンスの設定」(137ページ)を参照してください。

# ベストプラクティス

- 可能な限り、トランジション線が交差しないようにします。
- 可能な場合、直線のトランジションを使用します。曲線のトランジションは、フローのレイアウトに必要な場合のみ使用します。

- 可能な場合、トランジションが水平、垂直、または45度の斜線になるようにステップを配置します。
- 2 つのステップ間の複数のトランジションを1つにまとめて、1本の線ですべてのトランジションを表す ようにします。
- トランジションのラベルがステップのラベルと重なったり、トランジションのラベル同士が重なったりしないように、トランジションのラベルを配置します。
- ほかのユーザーに対してフローが明確になる場合は、トランジションのラベル名を変更します。
- 可能な場合、トランジションのラベルをフローの外側に寄せて配置します。たとえば、2つのステップがフローキャンバスの上部にある場合、トランジションのラベルをトランジション線の上に配置します。ステップがキャンバスの下部にある場合、ラベルをトランジション線の下に配置します。

# 実行する操作

### ステップ間にトランジションを追加する

- 1. Studio の作成ペインでフローを開きます。
- 2. 次のステップに接続する必要があるステップで、レスポンス名をクリックするか、いずれかのレスポン スを表すアイコンをクリックし、そのレスポンスのターゲットステップにドラッグします。
- 3. トランジションをダブルクリックします。トランジションのインスペクターが開きます。
- 4. (オプション)トランジションの名前を変更するには、[トランジション名] ボックスに新しい名前を入力 します。
- 5. [説明] ボックスに、このトランジションが発生する原因となった先行ステップで起こったことについて の説明を入力します。この説明は、フローの実行時に HP OO Central の [結果の概要] エリアに 表示されます。

**注**:トランジションの説明は、トランジションの起点となるステップに関連します。たとえば、メッ セージ「localhost に対する ping は成功しました」は、ステップ「ターゲットシステムに対する ping の実行」に続くトランジション用に作成されましたが、このステップで起こったことについて の説明です。

#### フロー変数を含む説明を作成する

説明内にフロー変数を使用して、可変の情報を格納できます。たとえば、サーバー名がフロー変数 servernameに格納されている場合、このサーバーを識別するために、「サーバー \${servername}に接 続可能です」と入力できます。

- 1. 2つのステップ間にトランジションを作成します。
- 2. トランジションをダブルクリックしてトランジションのインスペクターを開きます。

3. [説明] ボックスに、ステップのオペレーションまたは実行されたフローの別の場所から取得された データを格納しているフロー変数を含めて、説明を入力します。参照の形式は、\${フロー変数 名}を使用する必要があります。

たとえば、ping コマンドを実行するステップで、ホストマシンの名前をhostというフロー変数に保存できるとします。トランジションの説明でこの値を使用する際は、\${host}という構文を使用して値を参照できます。成功レスポンスのトランジションの説明は、たとえば「\${host}に対するpingは成功しました」のように表示されます。"server 1"という名前のホストに対して HP OO Centralで実行した場合、概要説明には「server 1 に対する ping は成功しました」と表示されます。

## トランジション後のステップを実行できるユーザーを制限する (ゲート制御式トランジ ション)

ゲート制御式トランジションを使用すると、特定の役割に所属するユーザーに対して次のステップへの アクセスを制限して、トランジションの後のフローを続行するユーザーを制御できます。 この役割グルー プのメンバーでないユーザーがフローを実行しようとすると、フローが停止して、ユーザーはフローを別の ユーザーにハンドオフするのか、フローをキャンセルするのかを選択します。

ゲート制御式トランジションは赤で表示されます。

- 1. 2 つのステップ間にトランジションを作成します。
- 2. トランジションをダブルクリックしてトランジションのインスペクターを開きます。
- 3. [続行前にユーザーのグループを確認] チェックボックスをオンにします。
- 4. [必須の役割エイリアス] リストから、フローの実行を続けるためにユーザーに割り当てる必要がある役割を選択します。

### トランジション後に実行がハンドオフされるようにする

フローを別のユーザーにハンドオフするように、トランジションを設定できます。この処理が必要となるのは、次のステップで別のユーザーからの情報が必要である場合です。

ハンドオフトランジションを設定すると、フローの実行中に、フローのURLがメッセージ本文に記載された新規の電子メールメッセージが開きます。フローの実行者は、フローを引き継ぐユーザーを電子メールメッセージの宛先に指定し、メッセージを送信できます。宛先のユーザーがメッセージを受信した後で、フローを引き続き実行できるようになります。

- 1. 2 つのステップ間にトランジションを作成します。
- 2. トランジションをダブルクリックしてトランジションのインスペクターを開きます。
- 3. [このトランジション後にフロー実行をハンドオフ] チェックボックスをオンにします。

#### トランジションの完了をフローの ROI 値 でカウントする

フローのトランジションに値を関連付けることができます。これらの値は、各トランジションの投資収益 率 (ROI)の値を表します。フローが実行されると、これらの値は、実際に発生したトランジションに基 づいて記録されます。管理者は、Central でレポートを表示してフローのROI 値を確認し、ビジネスの 重要な統計情報として使用できます。

- 1. 2つのステップ間にトランジションを作成します。
- 2. トランジションをダブルクリックしてトランジションのインスペクターを開きます。
- 3. [トランジションの ROI 値] ボックスに、トランジションの数 値を入力します。

## 曲線定義点を追加して曲線のトランジションを作成する

曲線定義点を追加して、トランジションの形状を直線から曲線に変更できます。これにより、フローを 整理したり、積み重なっているトランジションを分けることができます。



- 1. 曲線定義点を配置するトランジションの上にマウスカーソルを合わせます。
- 2. 定義点を作成するには、Shiftを押しながらマウスをクリックします。
- 3. トランジションが適切な曲線を描くように点をドラッグします。

### 曲線定義点を削除する

曲線定義点の上にカーソルを合わせ、Shiftを押しながらマウスをクリックします。

#### トランジション名を移動する

トランジション名をクリックして別の場所にドラッグします。

### 2 つのステップ間 のトランジションを削除する

2 つのステップ間の既存のトランジションを削除するには、トランジションを選択し、キーボードの Delete キーを押します。

# 参照情報

#### トランジションのインスペクター

トランジションのインスペクターで、トランジションの詳細を指定します。

1215 522
----------

トランジション名: success 「ゲート制御式トランジション

•

д

必須の役割エイリアス: LEVEL\_THREE (Base [1.0.121])

🔲 このトランジション後にフロー実行をハンドオフ

トランジションの ROI 値: 0.0 <sub>「</sub>説明

GUI アイテム	説明
トランジション名	デフォルトでは、トランジション名は起点となるレスポンスの名前 (success、failureなど)と同じになりますが、トランジション名は変更することができます。
続行前にユーザーのグループ を確認	ゲート制御式トランジションを作成するには、このチェックボックスを オンにします。必須の役割エイリアスが割り当てられている場合の み、ユーザーは次のステップに進むことができます。
必須の役割エイリアス	フローの実行を続けるためにユーザーに割り当てる必要がある、 役割エイリアスを選択します。
このトランジション後にフロー 実行をハンドオフ	ハンドオフトランジションは、現在のバージョンではサポートされてい ません。
	トランジション後 に別 のユーザーにフローをハンドオフするには、この チェックボックスをオンにします。
トランジションの ROI 値	トランジションの値を入力します。これにより、フローの実行中にそのトランジションが発生した場合に、このフローの実行の値にトランジションの値が加算されます。
説明	このトランジションが発生する原因となった先行ステップで起こった ことの説明を入力します。この説明は、HP OO Centralの[ <b>結果</b> <b>の概要</b> ] エリアに表示されます。

# レスポンスの設定

レスポンスは、オペレーションまたはフローの、発生する可能性があるさまざまな結果の1つです。



レスポンスの種類には次の4つがあります。

- 解決済み 🖉 正常に実行されているオペレーションまたはフローの標準のレスポンスです。
- 診断済み C→ 何が問題なのかを特定して、通知以外の操作を実行しないことをオペレーション またはフローが選択したことを示します。
- 操作なし □ データを収集したが、診断か修復かをオペレーションまたはフローが決定できない 場合に使用されます。

**注**: データ収集のみを目的としたオペレーションでは、オペレーションが完了すると、操作なし ○ ではなく解決済み ② を返します。

• エラー〇 - ステップまたはフローの実行が失敗した場合に使用されます。たとえば、入力が正しくない場合や、システムに到達できない場合に使用されます。

オペレーションまたはフローに、同じ種類のレスポンスが複数設定されている場合があります。たとえば、SQL クエリオペレーションに次の結果が設定されている場合があります。

- さらにアイテムが存在する 🝚
- これ以上アイテムが存在しない 의
- 失敗 😳

オペレーションまたはフローのレスポンスを追加、削除、または編集できます。 リターンステップを除き、 ステップのレスポンスを編集することはできません。 リターンステップの詳細については、「リターンステップ の作成」(179ページ)を参照してください。

## レスポンスルール

ルールを使用すると、オペレーション結果の特定の条件が該当する場合のみ発生するように、レスポンスを制限できます。 ルールによって、指定した値と、オペレーションの未加工結果のフィールドの値が 比較されます。

たとえば、結果に1より大きい値が含まれる場合は成功レスポンスのみ返すルールを作成できます。

1つのレスポンスに複数のルールを作成すると、そのレスポンスのすべてのルールで、選択されるレスポンスに該当するかどうかを評価する必要があります。

レスポンスは、オペレーションの [レスポンス] タブに表示される順序で評価されます。ルールが該当する ことが最初に評価されたレスポンスが、選択されるレスポンスになります。そのため、port open レスポン スのルールが該当した場合、port listening のルールが同様に該当しても、port open レスポンスが選 択されます。フローについて最も役立つ結果を得るには、レスポンスの順序が非常に重要になります。

# ベストプラクティス

 オペレーションの失敗と否定の結果を混同しないようにします。たとえば、回答がTRUEまたは FALSE となる質問をオペレーションがした場合、回答 FALSE は失敗と同じことではありません。このような場合、結果がFALSEの場合とオペレーションの失敗の場合の、2つのエラーで結果を返 す必要があります。



# 実行する操作

オペレーションにレスポンスを追加する

1. オペレーションの [レスポンス] タブで、[レスポンスの追加] をクリックして新しいレスポンスの名前を 入力します。

入力 出力 レスポンス 説明 スクリプトレット 詳細							
▲ ▼ レスポンスの概要							
		レスポンジ	スの追加	1/7	ポンスの削除	1	÷
レスポンス	デフォルト	失敗時	種類	į		ルール	
success			🕑 😋 🤇	0	1個のルール D	ース: returnCode, フィル	ターなし完全ー 🕺
failure			000	) 😳	1個のルール D	ース: returnCode, フィル	ターなし完全一 🐼

- 2. オペレーションの実行が失敗した場合に選択されるレスポンスを指定するには、[失敗時]列でそのレスポンスのチェックボックスをオンにします。
- 3. レスポンスをデフォルトのレスポンスとして指定するには、[デフォルト]列にあるチェックボックスをオンにします。レスポンスのルールで該当するものがない場合、デフォルトのレスポンスが、選択される レスポンスになります。
- 4. [種類]列で、レスポンスの種類を選択します。
  - 解決済み: 🔮
  - 診断済み: 🔂
  - 操作なし: ◯
  - 失敗: 😳

これにより、ステップの作成に使用するときに、オペレーションに表示されるレスポンスアイコンが 決まります。

5. レスポンスのルールを作成するには、レスポンスの行の右端にある右向き矢印 😂 をクリックします。詳細については、以下の「レスポンスのルールを作成する」を参照してください。

#### レスポンスのルールを作成する

ルールを使用すると、結果の特定の条件が該当する場合のみ発生するように、レスポンスを制限できます。

- 1. オペレーションに新しいレスポンスを作成します。
- 2. レスポンスルールエディターで、[追加]をクリックします。

新しいルールが表示されます。

入力   出力   レスポンス   説明   スクリプト	レット   詳細	
▲ マ レスポンスの概要 > success		$\leftrightarrow x$
以下のルールのすべてが真であるとき、'succe	ss' レスポンスが選択されます	追加 削除
フィールドにルールを適用	ルールの種類	ルールテキスト
returnCode	完全一致 📃	0 式
	後方一致 🔺	
	完全一致	
	すべての単語に一致	
	少なくとも1単語に一致	
	どの単語にも一致しない	
	1 単語にのみ一致	
	非完全一致	
	参照 <u></u>	

3. [フィールドにルールを適用]列で、ルールをテストする値を持つ結果フィールドを選択します。

テストできる結果フィールドには、結果の終了コード、出力文字列、エラー文字列、失敗メッセージ、およびタイムアウトになった true または false が含まれます。

注:これらの結果フィールドの詳細を表示するには、[説明]タブをクリックします。

これらのフィールドの値を確認するには、Studio Debuggerを使ってフローのオペレーションをテストします。Debuggerでフローを実行すると、ステップ結果インスペクターにステップの結果が 表示されます。Debuggerの詳細については、「フローのテストとデバッグ」(208ページ)を参照 してください。

4. [ルールの種類]列で、フィールド値でテストを行う比較または一致を選択します。

ルールの種類	
!=	۲.
後方一致	
完全一致	
すべての単語に一致	
少なくとも1単語に一致	
どの単語にも一致しない	
1 単語にのみ一致	
非完全一致	
参照	Ŧ

- [=]、[!=]、[前方一致]、[部分一致]、[すべての単語に一致]、[少なくとも1単語に一致]な どの単純な演算子を選択します。
- 正規表現を作成するには、[**正規表現**]を選択します。
- スクリプトレットを作成するには、[スクリプトレット]を選択します。
- 共有ルールへの参照を作成するには、[参照]を選択します。
- 5. [ルールテキスト]列に、テストで使用するテキストを入力します。

#### レスポンスルールをフィルタリングおよびテストする

ルール詳細エディターで、次の操作が可能です。

- ルール、フィルター、正規表現、またはスクリプトレットの使用を含め、ルールを詳しく指定する。
- ルールの作成時にルールをテストする。
- 1. ルール詳細エディターを開くには、該当するルールの行の右端にある右向き矢印 😂 をクリックします。

入力   出力   レスポンス   説明   スクリプトレッ	ト 言義田
▲ ▼ レスポンスの概要 > success > ルール	(1 ≠ 1) ↔ ×
入力に値が含まれるかどうかを確認するテスト	
大文字と小文字を区別しない: 🔽	
テキスト 0	
	unuu
「人力(一致が短調表示されます)	
パターンが一致します	יאראבקקארא אנארא 🗗 🔁 🔁
この領域に、照合させる入力文字を入力	
初の文字には下線も引かれるため、2 つ	の一致が連続している場合でも容易に判断できます (例: 強調表示強
調表示)。	
どちらかのテキスト領域を編集すると、・	一致の強調表示が即应に軍新されます。入力された正規表現に構立上
の問題がある場合は、入力中に一時的に	問題をなっているだけであっても、 編集によって正しい表現に戻るま
で 挿入力ニンルが去色で表示されます。	
	•
	•
	•

注: ルールの種類として [**スクリプトレット**]を選択している場合、ルール詳細エディターにスク リプトレットエディターが含まれます。 スクリプトレットの作成の詳細については、「フロー内での スクリプトレットの使用」(198ページ)を参照してください。

ルールの種類として [**正規表現**]を選択している場合、ルール詳細エディターに正規表現エディターが含まれます。正規表現の作成の詳細については、「フローでの正規表現の使用」(202ページ)を参照してください。

- 2. 別のルールの種類を使用するには、[ルールの種類]リストでほかの種類を選択します。
- 3. ルールを適用する前に結果をフィルタリングするには、[ルールを適用する前に結果をフィルター] をクリックし、フィルターエディターでフィルターを作成します。

レスポンスルールのフィルターを作成するのは、出力または結果のフィルターを作成するのと同じです。「出力および結果のフィルタリング」(156ページ)を参照してください。

ほとんどのルールの種類については、[テキスト]ボックスに、比較をテストするテキストを入力します。大文字小文字を区別しない場合は、[大文字と小文字を区別しない]チェックボックスをオンにします。

**正規表現**ルールでは、オペレーション結果の**正規表現**フィルターを作成するのと同じように、正 規表現とその用途を指定します。詳細については、「フローでの正規表現の使用」(202ページ) を参照してください。 結果ボックスにテストの結果が表示されます。[パターンが一致します] または [パターンが一致しま せん] が表示されて、一致するテキストが強調表示されます。

5. オペレーションのレスポンスの別のルールを使用するには、[レスポンスの概要]の横にある上向き 矢印または下向き矢印をクリックします。

**注**: ルールでは、数字で始まる文字列の評価に比較演算子 (=、 !=、<、>など)を使用する と、比較されるのは文字列の数字部分のみです。たとえば、!=(等しくない)を使用して "123" と "123Test"を比較する場合、"123" は明らかに "123Test" と同じではありませんので、評価は "false" となります。ただし、**部分一致**評価子で文字列を比較すると、この問題に対処できま す。

#### フローにレスポンスを追加する

フローにレスポンスを作成したら、作成したレスポンスをフローのリターンステップで使用できるようにします。

たとえば、**エラー**リターンステップをもたらした結果が、オペレーションの失敗ではなく、要求されるしきい 値に到達しない結果だった場合、この結果を反映して「**エラー: しきい値に到達していない**」と表示 される新しいレスポンスを**エラー**リターンステップに作成できます。.

- 1. フローの[プロパティ]シートを開きます。
- 2. [レスポンス] タブをクリックします。
- 3. [レスポンスの追加]をクリックし、表示されたテキストボックスに、レスポンスの名前を入力します。 たとえば、「しきい値に到達していない」と入力します。
- 4. [OK] をクリックします。

フローに**エラー**リターンステップを作成するときに、レスポンスとして「**しきい値に到達していない**」を選択 できます。リターンステップの詳細については、「リターンステップの作成」(179ページ)を参照してください。

#### オペレーションまたはフローからレスポンスを削除する

- 1. オペレーションまたはフローの [**プロパティ**] シートを開きます。
- 2. [レスポンス] タブをクリックします。
- 3. レスポンスを選択し、[レスポンスの削除]をクリックします。

# 参照情報

#### フローの [プロパティ] シート > [レスポンス] タブ

フローの [**プロパティ**] シートの [レスポンス] タブで、フローのリターンステップに使用 できるようにする、発生する可能性があるレスポンスを指定します。たとえば、「エラー: しきい値に到達していない」を指定します。

入力 出力 レスポンス	説明 ス	カリプトレット	詳細					
▲ ▼ レスポンスの概要	)							$\times$
		レスポンスの	)追加	レスポ	ンスの削除	Û	÷	
レスポンス	デフォルト	失敗時	種	類		ルール		
failure			<b>O</b>	0	1 個のルール	レース: returnCode, フィ	アルターなし 完全…	<≻
success			🕑 🔁	<u> </u>	1個のルール	シース: returnCode, フィ	ルターなし 常に…	$\Leftrightarrow$

GUI ፖイテム	説明
レスポンスの追加	新しいレスポンス行を追加します。
レスポンスの削除	選択したレスポンス行を削除します。
<b>4 4</b>	クリックして、選択したレスポンスをリスト内で上または下に移動します。

# オペレーションの [プロパティ] シート > [レスポンス] タブ

オペレーションの[プロパティ] シートの[レスポンス] タブで、オペレーションで発生する可能性があるレスポンスを指定します。

入力 | 出力 | レスポンス | 説明 | スクリプトレット | 詳細 | ▲ ▼ レスポンスの概要  $\leftrightarrow \times$ レスポンスの追加 レスポンスの削除 1 4 失敗時 種類 デフォルト レスポンス ルール 📝 😌 😂 1 個のルール [ソース: returnCode, フィルターなし, 完全一... 🤡 😌 😂 🔯 1 個のルール [ソース: returnCode, フィルターなし, 完全一... success Γ  $\Box$ 7 Г failure

GUI アイテム	説明
レスポンスの追加	新しいレスポンス行を追加します。
レスポンスの削除	選択したレスポンス行を削除します。
<b>1</b>	クリックして、選択したレスポンスをリスト内で上または下に移動します。
デフォルト	オンにして、レスポンスをデフォルトのレスポンスとして指定します。 レスポンス のルールで該当するものがない場合、 デフォルトのレスポンスが、 選択される レスポンスになります。
失敗時	オンにして、オペレーションの実行が失敗した場合に選択されるレスポンスを 指定します。
種類	レスポンスの種類を選択します。
-----------	--
	● 成功/解決済み: 🤡
	● 診断済み: 🔂
	• 操作なし: 😑
	• 失敗: 😳
ルール	レスポンスに作成したルールを表示します。
<b>\$</b>	クリックすると、ルールエディターが表示されて、レスポンスのルールを作成でき るようになります。

## オペレーションの [プロパティ] シート > [レスポンス] タブ > ルールエディター

ルールエディターで、結果の特定の条件が該当する場合のみ発生するように、レスポンスを制限します。

たとえば、結果に1より大きい値が含まれる場合は成功レスポンスのみ返すルールを作成できます。

<ul> <li>入力   出力   レスポンス   説明   スクリプト</li> <li>▲ ▼ レスポンスの概要 &gt; success</li> </ul>	レット   詳細		<b>↓</b> ⊢ ×
ー 以下のルールのすべてが真であるとき、'succe	ss' レスポンスが選択されます	追加	削除
フィールドにルールを適用	ルールの種類	ルールテキスト	
returnCode	完全一致 🗾	0	
	後方一致 🔺		
	完全一致		
	すべての単語に一致		
	少なくとも1単語に一致		
	どの単語にも一致しない		
	1 単語にのみ一致		
	非完全一致	]	
	参照 🗾 🗾		

GUI アイテム	説明
レスポンスの追加	新しいレスポンス行を追加します。
レスポンスの削除	選択したレスポンス行を削除します。
<b>1</b>	クリックして、選択したレスポンスをリスト内で上または下に移動します。
フィールド にルールを 適用	ルールをテストする値を持つ結果フィールドを選択します。テストできる結果 フィールドには、結果の終了コード、出力文字列、エラー文字列、失敗 メッセージ、およびタイムアウトになった true または false が含まれます。
ルールの種類	フィールド値でテストを行う比較または一致を選択します。

ルールテキスト	テストに使用するテキストを入力します。
52	クリックすると、ルール詳細エディターが開いて、ルールをテストおよびフィルタリ ングできるようになります。

# オペレーションの [プロパティ] シート > [レスポンス] タブ > ルールエディター > ルール詳細 エディター

ルール詳細エディターで、フィルターをテストしてルールに適用できます。

ルールを使用すると、結果の特定の条件が該当する場合のみ発生するように、レスポンスを制限できます。

入力 出力 レスポンス 瓢	郑明 スクリプトレット 詳細		
▲ ▼ レスポンスの概要 > su	ccess >		$\bullet \vdash X$
「ルール詳細」			
▶ ルールの種類: 常に一致	<u>۲</u>	ルールを適用する前に結果をフィルター	
入力に値が含まれるかどうか	を確認するテスト		
- 人又子と小又子を区別しない テキフ	,: 💌		
144	te la		
∟ □入力 (一致が強調表示され)	±व)	101001	
パターンが一致します		₽ 🗎 🌡	2 DUR DAVDIR
この領域に、照合させる	入力文字を入力します。すべ	にの一致 (重複なし) が強調表示され	います。各一致の最 🛋
初の文字には下線も引か	れるため、2 つの一致が連続	もしている場合でも容易に判断できます	' (例: 強調表示強
調表示)。			
	大石生于了。 动不动词主		
こちらかのテキスト領域 の問題がある場合は 入	を編集すると、一致の短調衣 カロニ→時約二門頭をなって	(示小叫)坐に更新されます。 八月された コンスだけであっても、 須生に トップ正	止視表現に備义上
で、挿入力ー・ハルが赤色	が中に、時間に問題となりで で表示されます。		OV WORLD A
GUI アイテム	説明		

GUI アイテム	説明
ルールの種類	ルールエディターで選択した、ルールの種類を表示します。別のルールの種 類を選択することもできます。
ルールを適用する前 に結果をフィルター	クリックすると、フィルターエディターが表示されて、ルールを適用する前に結 果をフィルタリングできるようになります。
テキスト	比較をテストするテキストを入力します。

大文字と小文字を 区別しない	オンにすると、テキストが大文字であるのか小文字であるのか区別されなく なります。
結果ボックス	テストの結果を表示します。[ <b>パターンが一致します</b> ] または [ <b>パターンが一致</b> しません] が表示されて、一致するテキストが強調表示されます。
コピー	結果ボックス内のデータをコピーします。
貼り付け	結果ボックスにデータを貼り付けます。
切り取り	結果ボックス内のデータを切り取ります。
クリア	結果ボックス内のデータをクリアします。
クイックコマンド	フィルターをテスト するデータを生成 するコマンドを入力します。 コマンドの出力 は結果 ボックスに表示 されます。

## 出力と結果の作成

フロー内で使用するデータを取得する方法の1つに、ステップ結果を使用する方法があります。この データを割り当てる方法は2つあります。

- 結果内の出力をフロー変数に割り当てるときに、この出力をデータとしてフロー内のほかのステップに渡すことができます。
- 結果内の出力をフロー出力フィールドに割り当てるときに、この出力をデータとして親フローに渡すことができます。

このプロセスには、次のようなさまざまなステップがあります。

1. プライマリ出力を含め、オペレーションの出力を設定します。

「オペレーションの出力の設定」(147ページ)を参照してください。

 フロー内のステップにオペレーションを使用するときに、どのオペレーション出力をステップ結果として 使用するのか、つまり、どのオペレーション出力をフロー変数またはフロー出力フィールドに割り当 てるのかを決定します。

「ステップの結果の設定」(150ページ)を参照してください。

3. (オプション) フィルターを作成して、出力または結果の選択の範囲をさらに絞り込むことができます。

「出力および結果のフィルタリング」(156ページ)を参照してください。

## オペレーションの出力の設定

フロー出力を設定するには、まず、オペレーションの出力を設定します。これを行うと、フロー作成者 (およびほかのフロー作成者)がこのオペレーションをフロー内で使用するときに、出力をフロー変数に 割り当てできるようになります。

### オペレーション出力のタイプ

次のようなさまざまなタイプのオペレーション出力があります。

• 未加工結果は、すべてのオペレーションのリターンコード、データ出力、およびエラー文字列です。

Studio では、オペレーションから作成されたステップの未加工結果を除き、未加工出力は直接表示されません。

- プライマリ出力とその他の出力は未加工出力の一部で、出力として指定したサクセスコード、出力文字列、エラー文字列、障害メッセージなどです。
  - プライマリ出力とは、ステップのプライマリ結果として使用される出力です。プライマリ出力は入力に値を提供し、その入力値の割り当てには[前のステップの結果]を使用します。
  - オペレーションのセカンダリ出力とは、プライマリ出力に追加する別の出力です。

**ヒント**: 出力にフィルターを1つ以上作成すると、出力の選択の範囲をさらに絞り込むことができます。「出力および結果のフィルタリング」(156ページ)を参照してください。

#### オペレーション出力の例

オペレーションには通常、そのオペレーションに固有の出力が含まれます。ただし、Libraryの Accelerator Packs、Integrations、Operationsの各フォルダーでオペレーションを使用する場合、 次の出力がよく見られます。

• returnResult

名前付きのフィールドがなく"returns:"が表示された場合、通常これがプライマリ出力です。また、 プライマリ出力は、大文字 R(汎用)の **Result** でアクセス可能です。

• response (または returnCode)

オペレーションのレスポンスの決定に使用するコードまたは文字列。

• failureMessage

インフラストラクチャーの内部出力。オペレーションが失敗を返す場合、この出力は例外となります。多くのオペレーションはこの出力を使用しません。

### ベストプラクティス

- 大文字と小文字に一貫性があるようにします。たとえば、すべての出力名にキャメルケースを使用します。
- 統合を行っている場合、使用されている API からの元の出力名を保持します。

## 実行する操作

#### オペレーションのプライマリ出力を指定する

オペレーションを設定する際に、プライマリ出力を指定できます。 プライマリ出力を作成すると、その ソースを変更できますが、プライマリ出力なしの状態に戻すことはできません。

- 1. [プロジェクト] ペインでオペレーションを右クリックし、[プロパティ]を選択します。
- 2. [出力]タブを選択します。
- 3. [フィールドからプライマリ出力を抽出]リストから、ソースフィールドを選択します。たとえば、 [FailureMessage]を選択します。

**ヒント:** 各出力フィールドに表示されるデータの詳細については、オペレーションの[説明]タブをクリックします。

#### オペレーションにセカンダリ出力を追加する

オペレーションのセカンダリ出力とは、プライマリ出力に追加する別の出力です。

- 1. [プロジェクト] ペインでオペレーションを右クリックし、[プロパティ]を選択します。
- 2. [出力] タブを選択します。
- 3. [出力の追加]をクリックします。
- 4. 出力の名前を入力します。
- 5. [出力フィールド]リストから、出力データを取得するフィールドを選択します。

フィルターの作成の詳細については、「出力および結果のフィルタリング」(156ページ)を参照してください。

#### オペレーションから出力を削除する

- 1. [プロジェクト] ペインでオペレーションを右クリックし、[プロパティ]を選択します。
- 2. [出力]タブを選択します。
- 3. 削除する出力を選択して、[出力の削除]をクリックします。

#### 出力データを取得するフィールドを変更する

1. オペレーションの[プロパティ]シートを開いて、[出力]タブを選択します。

- 2. プライマリ出力のフィールドを変更するには、[フィールドからプライマリ出力を抽出]ボックスの右に ある下向き矢印をクリックし、リストから目的のフィールドを選択します。
- 3. セカンダリ出力のフィールドを変更するには、出力の行の[出力フィールド]列をクリックして、リストから目的のフィールドを選択します。

## 参照情報

## [プロパティ] シート > [出力] タブ

[プロパティ]シートの[出力]タブで、オペレーションのプライマリ出力とセカンダリ出力を指定します。

入力 田力 レスポンス   説明   スクリプ	トレット 詳細	
▲▼ 出力の概要		* * ×
フィールドからプライマリ出力を抽出: Failure 利用できる出力	Message	▼ フィルターの編集
		出力の追加 出力の削除
名前	出力フィールド	フィルター
TimeOut	TimedOut 74	///ターなし 🗠
GUI ፖイテム	説明	
フィールド からプライマリ出 力を 抽出	プライマリ出力のデータを取得す	るフィールドを選択します。
フィルターの編集	プライマリ出力のフィルターエディタ	ターを表示します。
出力の追加	新しい出力行を追加します。	
出力の削除	選択した出力行を削除します。	
出力フィールド	セカンダリ出力のデータを取得す	るフィールドを選択します。

## € その行の出力に対してフィルターエディターを表示します。

## ステップの結果の設定

オペレーションではさまざまな出力が生成されますが、出力はフローに自動的には保持されません。 保持した場合、不要なデータでフローが低速になり、パフォーマンスに悪影響を与える可能性があり ます。

ステップのインスペクターの[結果]タブで、必要な結果を指定します。結果には、オペレーションの結果が含まれます。次の2つの方法で結果を格納できます。

• *同じフロー*内のオペレーション、トランジション、およびプロンプトがアクセスできるフロー変数を作成 する。詳細については、「変数の使用」(173ページ)を参照してください。

例: ステップ LocalPing は、ターゲットホストが使用可能かどうかを判断し、結果 PingOutput に ping オペレーションの出力を格納します。これにより、後続のステップで使用で きるフロー変数 PingOutput を作成します。

次のステップ Display では、変数 PingOutput をユーザーに表示します。このステップのプロンプトのテキストは、ping の結果: {PingOutput}として設定されています。

フローがサブフロー(別のフロー内のステップ)として使用される場合、親フロー内のオペレーション、トランジション、およびプロンプトがアクセスできる出力フィールドを作成する。詳細については、「フロー内でのサブフローの作成」(183ページ)を参照してください。

**例**:親フローには、Windows Health Checkフローがサブフローとして含まれているステップがあります。Windows Health Checkフローの結果は出力フィールド HealthCheckOutputの値として格納されて、メインフローで使用できるようになります。

メインフローには、出力フィールド HealthCheckOutput の値を電子メールの本文に表示する、メールの送信オペレーションが含まれています。

ステップの結果には次の2種類があります。

- 未加工結果は、フローのコンテキストで実行されたオペレーションから戻された、すべての未加工データです。ステップの未加工結果とプライマリ結果は、元のオペレーションの未加工出力とプライマリ出力から取得されます。
- ステップエディターの[結果]タブで作成する、その他の結果。ステップのインスペクターで、セカンダリ結果を作成して指定できます。

ステップにフローの結果を設定する前に、関連するオペレーションにプライマリ出力が設定されていることを確認します。「オペレーションの出力の設定」(147ページ)を参照してください。

**ヒント**: 出力にフィルターを1つ以上作成すると、結果の選択の範囲をさらに絞り込むことができます。「出力および結果のフィルタリング」(156ページ)を参照してください。

### ベストプラクティス

- 大文字と小文字に一貫性があるようにします。たとえば、すべての結果名にキャメルケースを使用します。
- ステップのベースとなるオペレーションまたはフローでは、複数の出力を提供することができます。ただし、ステップの結果を追加するときに、フローに必要な出力のみ使用していることを確認します。

## 実行する操作

#### ステップにプライマリ結果を作成する

オペレーションにプライマリ出力が設定されています。プライマリ出力は入力に値を提供し、その入力値の割り当てには[前のステップの結果]を使用します。

ステップで、このプライマリ出 力をフロー変 数 で取 得 する (フロー内 のほかのステップで使 用 する) のか、 フロー出 カ フィールド で取 得 する (親 フローに渡 す) のかを指 定 できます。

- 1. 作成ペインでステップをダブルクリックします。
- 2. [結果] タブを選択し、[結果の追加]をクリックします。
- 3. [**名前**] 列に結果の名前を入力します。キーボードのReturn キーを押します。この名前は、フロー変数またはフロー出力フィールドの名前に使用されます。

注:結果名に「Result」を使用しないでください。

4. [ソース] 列のリストから、結果のソースとしてプライマリ出力を選択します。

たとえば、そのオペレーションのプライマリ出力である[結果フィールド: returnResult]を選択します。

プライマリ出力の設定の詳細については、「オペレーションの出力の設定」(147ページ)を参照してください。

#### ステップにセカンダリ結果を作成する

- 1. 作成ペインでステップをダブルクリックします。
- 2. [結果] タブを選択し、[結果の追加] をクリックします。
- 3. [名前]列に結果の名前を入力します。キーボードのReturn キーを押します。この名前は、フロー変数またはフロー出力フィールドの名前に使用されます。
- 4. [ソース] リストから、結果のソースを選択します。
- 5. [割り当て先]リストで、値の保存先を指定します。
  - フロー変数に値を格納するには、[Flow Variable]を選択します。
  - 親フローで値を使用できるようにするには、[Flow Output Field]を選択します。
- 6. [割り当て操作]リストから、適切な操作を選択します。

- OVERWRITE フロー変数またはフロー出力フィールドの現在の値をこの値に置き換えます。
- APPEND フロー変数またはフロー出力フィールドの現在の値の末尾にこの値を追加します。
- PREPEND フロー変数またはフロー出力フィールドの現在の値の先頭にこの値を追加します。
- 4 つの算術割り当て操作 ADD、SUB、MULTIPLY、および DIVIDE のいずれかを使用して、フロー変数またはフロー出力フィールドの現在の値を算術的に変更します。

たとえば、ステップの結果が3.14の場合、MULTIPLYを選択すると、フロー変数またはフロー出力フィールドの現在の値と3.14が乗算されます。

フィルターの作成の詳細については、「出力および結果のフィルタリング」(156ページ)を参照してください。

#### ステップから結果を削除する

- 1. 作成ペインでステップをダブルクリックします。
- 2. [結果] タブを選択し、[結果の削除] をクリックします。
- 3. ステップを保存します。

#### 結果データを取得するフィールドを変更する

- 1. 作成ペインでステップをダブルクリックします。
- 2. [結果] タブを選択します。
- 3. 出力の行の[ソース]列をクリックし、リストから目的のフィールドを選択します。
- 4. ステップを保存します。

### 参照情報

#### ステップのインスペクター > [結果] タブ

ステップのインスペクターの[結果]タブで、フロー変数に保存する出力を指定したり、親フローで使用 できるようにする出力を指定します。

インスペクター								_ +Þ
ステップ名: Traceroute								
入力 結果 表示	説明 詳細 スクリプトレ	ット						
▲ ▼ ステップ結果								$\leftrightarrow \vdash \times$
			結果	の追加	結果の	削除	1	÷
名前	א-ע	割り当て先		割り当る	て操作		フィルター	
traceroute	Result Field: returnR	Flow Variable	0	VERWRIT	E	フィルタ	ーなし	42
■ インスペクター								
設計 プロパティー								

GUI ፖイテム	説明
結果の追加	新しい結果行を追加します。
結果の削除	選択した結果行を削除します。
名前	結果の名前を入力します。この名前は、フロー変数またはフロー出力 フィールドの名前に使用されます。
ソース	結果のソースを選択します。
割り当て先	結果の値を格納する場所を選択します。
	• フロー変数に値を格納するには、[Flow Variable]を選択します。
	<ul> <li>親フローで値を使用できるようにするには、[Flow Output Field]を選択します。</li> </ul>

割り当て操作	[割り当て操作]リストから、適切な操作を選択します。
	• OVERWRITE – フロー変数またはフロー出力フィールドの現在の値をこの値に置き換えます。
	• APPEND - フロー変数またはフロー出力フィールドの現在の値の末尾に この値を追加します。
	• PREPEND - フロー変数またはフロー出力フィールドの現在の値の先頭にこの値を追加します。
	<ul> <li>4つの算術割り当て操作 ADD、SUB、MULTIPLY、および DIVIDE を 使用して、フロー変数またはフロー出力フィールドの現在の値を算術的 に変更します。</li> </ul>
	たとえば、ステップの結果が3.14の場合、MULTIPLYを選択すると、フロー変数またはフロー出力フィールドの現在の値と3.14が乗算されます。
8	その行の結果のフィルターエディターを表示します。

## ステップのインスペクター > [詳細] タブ

ステップのインスペクターの[詳細]タブで、ステップのベースとなるソースオペレーションを変更できる場所 を指定します。

インスペクター	- +
ステップ名: Traceroute	
入力   結果   表示   説明   詳細   スクリプトレット	
実行	
▶ 並列実行を、この同時インスタンス数にまで絞ります:	
オペレーション	
リンク先: Local Traceroute [Operation]	
場所: //Base [10121]/Library/Operations/Network/Local Traceroute	¥#R
	<b>E</b> 1/(

GUI アイテム	説明
リンク先	ステップのベースとするソースオペレーションを表示します。
場所	ステップのベースとするソースオペレーションの場所を表示します。
選択	[オペレーションの選択] ダイアログボックスを開きます。 このダイアログボックス で、 ステップのベースとなるオペレーションに移動して、 そのオペレーションを選 択できます。

L

## 出力および結果のフィルタリング

オペレーションの出力や、ステップの結果の抽出や編集を行う際にフィルターを作成することができます。

たとえば、サーバーに対する ping オペレーションについて、最大、最小、平均の往復時間のみが必要だとします。未加工出力を3つの出力にフィルタリングして、これらの情報を未加工出力から分離して抽出できます。

## 実行する操作

#### フィルターの作成

- 1. フィルターエディターを開きます。この手順は、フィルタリング内容によって異なります。
  - オペレーションのプライマリ出力に1つ以上のフィルターを作成するには、オペレーションの[プロ パティ] シートを開き、[出力] タブをクリックして、[フィルターの編集] ボタンをクリックします。
  - オペレーションのセカンダリ出力にフィルターを作成するには、オペレーションの[プロパティ]シートを開き、[出力]タブをクリックして、出力行の端にある右向きの矢印 <sup>[2]</sup>をクリックします。
  - ステップのセカンダリ結果にフィルターを作成するには、作成ペインでステップをダブルクリックし、
     [結果] タブをクリックして、結果行の端にある右向きの矢印 😂 をクリックします。
- 2. フィルターエディターで、[追加]ボタンをクリックします。

注:出力または結果には複数のフィルターを追加できます。

3. [フィルターの選択]リストで、フィルターの種類を選択します。

フィルターの選択	×
追加するフィルターの選択	
形式	
形式	
行の選択	
   1000000000000000000000000000000000000	
重複行の削除	
数値の丸め	
数値の抽出	
止現表現	<b>•</b>

4. フィルターエディターの右上にあるエリアの [詳細] で、フィルターを設定します。 それぞれのフィル ターのオプションの詳細については、フィルターオプションを参照してください。

#### コマンドラインからのデータを用いたフィルターのテスト

テストを行うには、一部のデータを[フィルター入力のテスト]ボックスに貼り付けます。このデータがローカルのコマンドラインコマンドで生成できる場合、次のようにします。

- 1. 出力または結果のフィルターエディターを開きます。
- 2. [クリア]をクリックして [フィルター入力のテスト] ボックスの内容をクリアします。
- 3. [**クイックコマンド**]をクリックします。
- 4. 目的のデータを生成するコマンドを入力します。
- 5. [OK] をクリックします。コマンドの出力が [フィルター入力のテスト] ボックスに表示されます。
- 6. 次のいずれかを実行します。
  - [**すべてのフィルターのテスト**]をクリックする
  - テストするフィルターを選択して[選択したフィルターのテスト]をクリックする

フィルターが (上から順番に) [フィルター入力のテスト] ボックスのデータに適用され、フィルタリング結果が [出力のテスト] ボックスに表示されます。

#### デバッガーからのデータを用いたフィルターのテスト

必要なデータが単純なコマンドラインコマンドで複製できない手段で生成されている場合、次のよう にデータをデバッガーからコピーして、[フィルター入力のテスト] ボックスに貼り付けます。

- 1. 出力または結果のフィルターエディターを開きます。
- 2. [クリア]をクリックして [フィルター入力のテスト] ボックスの内容をクリアします。
- 3. デバッガーでフローを実行します。
- 4. 該当するステップを強調表示します。
- 5. [ステップ結果 インスペクター] ペインで、[未加工結果] タブのコンテンツをコピーします。
- 6. フィルターエディターで、コンテンツを [フィルター入力のテスト] ボックスに貼り付けます。
- 7. 次のいずれかを実行します。
  - [**すべてのフィルターのテスト**]をクリックする
  - テストするフィルターを選択して [選択したフィルターのテスト]をクリックする

フィルターが (上から順番に) [フィルター入力のテスト] ボックスのデータに適用され、フィルタリング結果が [出力のテスト] ボックスに表示されます。

#### さまざまな出力や結果のフィルタリング

フィルターエディターを開いている間、[出力の概要]の横の上向き矢印または下向き矢印をクリックして、別の出力または結果のフィルターを作成できます。

#### 出力または結果でのシステムフィルターの使用

- 1. システムフィルターを使用する出力または結果のフィルターエディターを開きます。
- 2. [プロジェクト] ペインで、Configuration フォルダーとSystem Filters フォルダーを展開します。
- 3. 使用するフィルターを、System Filters フォルダーからフィルターエディターの [フィルター] リストにド ラッグします。



#### フィルターをシステムスフィルターとして保存

オペレーションの既存のフィルターを引用し、システムフィルターとして保存できます。結果のシステム フィルターは、作成したオペレーションに依存せず、ほかの出力や結果で再使用できます。

詳細については、「システムフィルターの設定」(88ページ)を参照してください。

## [表示オプション]

#### 大文字と小文字の変換

[大文字と小文字の変換]フィルターは、文字列に含まれるすべての文字を大文字または小文字に 変更します。[大文字に変換]チェックボックスを選択しないでそのままにした場合、このフィルターによ りすべての文字が小文字に変更されます。

#### 数値の抽出

[**数値の抽出**] フィルターは、結果内の最初の数字を抽出します。連続した整数は1つの数と見な されます。たとえば、"123Test" や"Test123" という文字列からは、[**数値の抽出**] フィルターにより"123" という数が抽出されます。

#### 形式

[形式] フィルターは、結果や出力にテキストを付加したり、結果や出力の元のコンテンツを指定したテキストに置換します。

「詳細: 形式-	
入力の先頭や	の末尾に解決された表現を追加したり、入力を解決された表現で置
き換えることで	、人力の形式を調整します
テキスト	¥n
入力の配置:	末尾に追加
	先頭に追加
	末尾に追加
	置換

- 1. [テキスト] ボックスに、結果に付加するテキスト、または結果と置換するテキストを入力します。
- 2. [入力の配置] リストで、次を行います。
  - テキストを既存のテキストの前に付加する場合は、[先頭に追加]を選択する
  - テキストを既存のテキストの後ろに付加する場合は、[末尾に追加]を選択する
  - 出力をテキストに置換するには、[置換]を選択する

#### 行数カウント

[行数カウント] フィルターは、結果に含まれる行の総数を出力します。

#### 正規表現

[正規表現] フィルターは、正規表現 (regex)を使用して未加工結果をフィルタリングします。

「詳細・止視表現 正規表現と一致する部分を抽出して、入力文字列をフィルターします。正規表現の詳細な説明については、ヘルプマニュアルを 参照してください。			
正規表現の種類	Java スタイル		
正規表現の値	*		
フィルターのスタイル:	⑦ 入力全体をフィルター		
	◎ 行単位でフィルター		
大文字と小文字を区別しない			

- 1. [**正規表現の種類**] リストで、[**Java スタイル**] を選択します。ほかのスタイルは使用しないでください。 推奨されていません。
- 2. [**正規表現の値**] ボックスに、正規表現を入力します。
- 3. [フィルターのスタイル] で、未加工結果にどのようにフィルターを適用するかに応じて、[入力全体 をフィルター] または [行単位でフィルター] を選択します。
- 4. 正規表現の大文字小文字を区別しない場合は、[大文字と小文字を区別しない]を選択します。

正規表現の使用の詳細については、「フローでの正規表現の使用」(202ページ)を参照してください。

#### 重複行の削除

このフィルターは同一の行を検出し、そのうち1つを残してすべて削除します。

連続している重複行のみにこのフィルターを適用するには、[連続]を選択します。

#### 置換

このフィルターは文字列の最初または最後のインスタンスか、すべてのインスタンスを別の文字列と置換します。

詳細: 置換 入力に含まれる文字のインスタンスを置き換えます	
検索:	
対象: すべて 🔽	
置換:	
大文字を小文字を区別しない: 🗖	

- 1. [検索]ボックスに、検索して置換する対象の文字列を入力します。
- 置換する対象文字列のインスタンスに応じて、[置換]から[先頭]、[すべて]、または[末尾]を選択します。
- 3. [置換] ボックスに、対象文字列と置換する文字列を入力します。
- 4. 検索時に大文字小文字を区別しない場合は、[大文字と小文字を区別しない] チェックボック スを選択します。

#### 数値の丸め

このフィルターは、数値を切り捨てまたは切り上げて丸めます。

- 詳細: 数値の丸め一 入力が数値として解釈	Rできる場合に、その数値を丸めます。
小数点以下の桁数:	0
丸めの種類	<ul> <li>切り捨て</li> </ul>
	⊙ 丸め
	○ 切り上げ

- 1. 丸めの精度を指定するには、数値を丸める小数位を[小数点以下の桁数] ボックスに入力します。
- 2. [丸めの種類]で、数値を丸める方向を指定します。
  - [切り捨て]は常に数値を切り捨てる
  - [切り上げ]は常に数値を切り上げる
  - [丸め]は、末尾の数字が5以上の場合に切り上げ、4以下の場合は切り捨てる

### スクリプトレット

このフィルターは、作成したスクリプトレットを使用してデータをフィルタリングします。

「詳細: スクリプトレット JavaScript スクリプトレットを実行してデータをフィルターします		
	テンプレートの挿入 スクリプトの確認	
<pre>1 scriptletResult = "Timestamp:"+new jage</pre>	ava.util.Date()+"\n"+scrip	
	•	
	F	

1. フィルターとして使用するのに必要となるスクリプトレットの行を取得するには、[テンプレートの挿入]をクリックします。

挿入されたテンプレートは、選択した言語に固有のもので、フロー変数、オペレーション結果、および入力へのアクセスや、フロー変数および結果の設定と加工に最もよく使用するコマンドを含んでいます。

2. スクリプトレットをデバッグするには、[スクリプトの確認]をクリックします。

スクリプトレットの詳細については、「フロー内でのスクリプトレットの使用」(198ページ)を参照してください。

#### 行の選択

このフィルターは未加工結果から抽出する行を指定します。

- 詳細: 行の選択 入力がら 1 行を選択します			
行の選択方法:	対象文字列が先頭にある最初の行		
パラメーター:			
行末の種類	○ 自動		
	O Unix		
	C Windows		

- 1. [行の選択方法]リストから、対象とする行の基準を選択します。
- 2. [パラメーター]ボックスに、対象文字列を含む文字列を入力します。
- 3. [行末の種類] グループで、次のいずれかを選択します。
  - フィルタリングしているテキストが、Unix オペレーティングシステム (LF で行を終了)で生成されている場合は、[Unix]を選択します。
  - フィルタリングしているテキストが、Windows オペレーティングシステム (CR/LF で行を終了)で生成されている場合は、[Windows]を選択します。

■ フィルターで、どちらの種類の行末でも処理できるようにするには、[自動]を選択します。

デフォルトでは[自動]が選択されています。

#### 範囲の選択

このフィルターは、入力データから抽出する文字列を指定します。指定する際に使用する2つの条件は、文字列の長さ(文字数)と、入力データの先頭から見た文字列の最初の文字の位置です。

-詳細: 範囲の選択 入力内の選択範囲を指定します(最初の文字は 0)	
開始: 0	
長さ 0	

- 1. [開始]ボックスに、文字列の開始位置をゼロを基準として入力します。
- 2. [長さ] ボックスに、文字列の文字数を入力します。

改行は、フィルタリングしているデータの取得元のオペレーティングシステムに応じて、1文字また は2文字としてカウントされます。

ソート

このフィルターは、各行の最初の文字を使用して入力データの行を並べ替えます。



- 1. 次のようにして、並べ替えの向きを指定します。
  - 昇順で並べ替える場合、[昇順]チェックボックスを選択します。
  - 降順で並べ替える場合、[昇順]チェックボックスのチェックを外します。
- 2. ASCII 順でデータを並べ替える場合、[数値として扱う] チェックボックスを選択します。

昇順のASCII順は、英語の文字の場合おおまかに次のようになります。

- 空白
- ∎ 記号

- ∎ 数字
- アルファベット

削除

このフィルターは、未加工結果の最初または最後から文字を削除します。

**注**: このフィルターがほかのフィルターの後に適用されると、前のフィルター処理により取得された 未加工結果のサブセットの最初または最後から文字を削除します。

- 詳細: 削除 入力の先頭または末尾から、一致した文字列を削除します		
削除方法:	対象文字より前(対象文字も含む)のすべての文字	
削除対象文字:		

- 1. [削除方法] リストから、フィルターで未加工結果から文字を削除する方法を選択します。[削除 対象文字] テキストボックスで指定した文字列を削除する際の次のようなオプションを指定でき ます。
  - [対象文字より前のすべての文字]の文字列
  - [対象文字より前 (対象文字も含む)のすべての文字]の文字列
  - [対象文字より後のすべての文字]の文字列
  - [対象文字より後 (対象文字も含む)のすべての文字]の文字列
- 2. [削除対象文字] テキストボックスに、検索する文字列を入力します。

#### スペースの削除

このフィルターは未加工結果の先頭および末尾から空白文字をすべて削除します。

テーブル

テーブルフィルターでは、未加工結果の表への変換は行いませんが、列の並べ替えや、列、行、ブロックの選択など、未加工結果を表として操作できます。

- 言关シロ・ニューゴル		
入力をテーブルとして解析し、指定:	された列でソートします	
列の区切り記号: スペース 💌	行の区切り記号:	改行 💌
最初の行はヘッダー: 🔲	結果の最初の行を削除:	
ソート基準列: -1	昇順	
選択行: 0	選択列:	0
選択幅: 1	選択高:	1

**注:** 行の番号設定は0ベース(0[0]から始まる)、列の番号設定は1ベースです。

- 1. [列の区切り記号]リストでは、データを列に分割するための記号を選択します。
- 2. [行の区切り記号]リストでは、データを行に分割するための記号を選択します。

注: 複数の連続した空白は1つの空白と見なされるため、列の右にあると想定していた データで列が占められることがあります。たとえば、列の区切り記号に空白を指定して dir コ マンドラインコマンドの出力にこのフィルターを適用する場合です。

- 3. 最初の行を列ヘッダーとするには、[最初の行はヘッダー]を選択します。
- 4. 最初の行を削除するには、[結果の最初の行を削除]を選択します。
- 5. 列で並べ替えを行うには、[ソート基準列]ボックスに列番号を(1ベースで)入力します。

ヒント: 値-1 は列で並べ替えを行わないことを意味します。

6. 昇順を指定するには、[昇順]ボックスを選択します。

デフォルトの並べ替え順序は、降順です。

- 7. フィルタリングして抽出する行を選択するには、次の操作を行います。
  - [選択行] ボックスに、行番号 (0ベース)を入力します。

**ヒント: -1**を選択すると、データ内のすべての行が選択されます。

■ [選択幅] ボックスに、抽出する行の列数を入力します。

**ヒント: -1**を選択すると、[選択列]に指定した列の右にある、データ内の残りの列がすべて選択されます。

- 8. フィルタリングして抽出する列を選択するには、次の操作を行います。
  - [選択列]に列番号を入力します。

ヒント:-1を選択すると、データ内のすべての列が選択されます。

■ [選択高]ボックスに、抽出する列の行数を入力します。

**ヒント: -1**を選択すると、[選択行]に指定した行の下にある、データ内の残りの行がすべて選択されます。

たとえば、2~4列目の最初の5行を抽出するには、次のように指定します。次に示す設定のうち、 最初の2つでは選択する行を指定し、残りの2つでは選択する列を指定します。

- [選択行]:0
- [選択高]:5
- [選択列]: 2
- [**選択幅**]:3

#### XML フィルター

XML フィルターを使用すると、ステップ内の XML を解析できます。 XML は HP OO デフォルトコンテンツ の XML 処理オペレーションのいずれかに XML を渡すフローを作成 せずに、 ステップの入力または結果から XML を取得することができます。

オペレーションでの XML フィルターの使用と、デフォルトコンテンツでの XML 処理オペレーションの使用は、いくつかの点で異なります。

- オペレーション内でのタスクの完了と、タスクの完了までのフローのインフラストラクチャーの仕様が異なる
- オペレーション内のフィルターには、XML処理オペレーションにはない制限がいくつかあります。これらの制限については、該当のフィルターについて次のセクションで説明します。入力 XMLのフィルタリングにフィルターとオペレーションのどちらを選ぶかは、XMLの取得方法によって決定します。

XML フィルターは次のとおりです。

- XML 属性の取得
- XML 要素の取得
- XML 要素値の取得
- XPath クエリ

XML フィルターを説明するため、例では次の XML 例を参照することとします。

```
<?xml version="1.0" encoding="utf-8"?>
<tickets>
<ticket id="1448" severity="3">
<customer firstName="John" lastName="Doe">
<volume>30000</volume>
<company>myOrg</company>
```

```
<position>CIO</position>
      <contactInfo>
        <email>jdoe@myorg.com</email>
        <email>johnsSecondEmail@myorg.com</email>
        <mobile>12065551212</mobile>
        <description internal="1">Private contact info</description>
        <description>Partial contact info</description>
      </contactInfo>
      <description>Our best customer</description>
    </customer>
    <details>
      <description>A simple Test xml</description>
      <comment user="john"> Initially raising ticket</comment>
      <comment user="frank"> Problem diagnosed, not a real issue</comment>
      <comment user="albert">ok, I'm going to close it.</comment>
      <state>Closed</state>
    </details>
  </ticket>
  <ticket id="1886" severity="5">
    <customer firstName="Elaine" lastName="Benson">
      <volume>50000</volume>
      <company>herCompany</company>
      <position>CEO</position>
      <contactInfo>
        <email>ebenson@herco.com</email>
        <mobile>011445551212</mobile>
        <description internal="1">Private contact info</description>
        <description>Partial contact info</description>
      </contactInfo>
      <description>Our other best customer</description>
    </customer>
    <details>
      <description>datastream bug</description>
      <comment user="jack">Customer found bug.</comment>
      <comment user="elsbeth">It is a third-party supplier bug.</comment>
      <state>Closed</state>
    </details>
  </ticket>
</tickets>
```

#### XML 属性の取得

[XML 属性の取得] フィルターでは、指定する属性の1つまたは複数のインスタンスそれぞれの値を 抽出します。フィルターエディターで、属性への要素のパスを指定してフィルターが適用される属性のイ ンスタンスを管理することができます。

属性の単一インスタンスに対する値を取得することも、複数のインスタンスに対する値を表形式で取得することもできます。この表では、列はコンマ区切りとなり、行は改行区切りとなります。

−詳細: XML 属性 XML 文書から、勇 使用します。	の取得 要求された属性値をフィルターします。高度な XML フィルター処理には XPath フィルターを	
要素パス:		
サブ要素を含む		
属性名:		
結果:	● 1つの一致	
	○ テーブル	

1. [要素パス] ボックスに、抽出する値の属性を含む要素のパスを指定します。要素へのパスを部分ごとに区切るには、フォワードスラッシュ(/)を使用します。

フィルターの属性値の取得元の要素のインスタンスを管理するには、[2] や[3] のように指定を追加します。要素の番号付けは、1 ベース ([1] から開始)です。したがって、要素の2 つ目のインスタンスを指定するには、[2] を使用します。

- 2. 指定した要素の子要素を検索するには、[サブ要素を含む]チェックボックスを選択します。
- 3. [属性名] ボックスに、値を取得する属性の名前を入力します。
- 4. [結果] で次のいずれかを選択します。
  - 抽出結果を属性の単一インスタンスの値に制限するには、[1 つの一致]を選択します。
  - 指定した属性のすべてのインスタンスの値を抽出するには、[テーブル]を選択します。

**例**: コメントのいずれかでユーザー名を検索するには (*XML フィルター*のトピックで使用した XML の 例を使用します)

[要素パス] ボックスに /ticket/details/commentと入力します。

例:特定のコメント (この例では2つ目のコメント)のユーザー名を取得するには、

- 1. [要素パス] ボックスに/ticket/details/comment[2] と入力します。
- 2. [属性名] ボックスに、user と入力します。
- 3. [結果]の横の[1つの一致]を選択します。

john と出力されます。

例: 各コメントのユーザー名を検索するには

- 1. [要素パス] ボックスに /ticket/details/comment と入力します。
- 2. [属性名] ボックスに、user と入力します。

3. [結果]の横の[テーブル]を選択します。

#### 次のように出力されます。

Path, user

```
/ticket/details/comment[1],john
```

/ticket/details/comment[2],frank

```
/ticket/details/comment[3],albert
```

#### XML 要素の取得

[XML 要素の取得] フィルターで、次のいずれかの方法で記述することで、要素全体 (子要素、値、 属性を含む)を抽出できます。

- 相対パスまたは絶対パスを指定する。
- 抽出する要素の子要素を指定する。子要素の特定の値を検索することもできます。
- 抽出する要素の属性を指定する。属性の特定の値を検索することもできます。

「詳細: XML XML 文書☆ を使用します	」要素の取得 から指定されたパスの要素をフィルターします。高 す。	高度な XML フィルター処理には XPath フィルタ	<u>9</u> —
要素パス:			
子の名前:		値:	
属性名:		値:	

次の手順で、テキストボックスを1つまたは組み合わせて使用して詳細を入力できます。

1. [要素パス] ボックスに要素の絶対パスを入力します。

パス内の相対パスインジケーターは、相対パスインジケーターの前の要素の相対位置を表します。

- ../は、最後に指定した要素の親を示します。
- ./は、最後に指定した要素です。

**例**:例のXMLでは、<volume>および <company>は兄弟要素で、両方とも <customer> 要素の子です。<company> 要素は、次のような相対パスで指定することもできます。

/tickets/ticket/customer/volume/../company

特定した要素に複数のインスタンスがある場合は、上記の例のようにパスを指定するだけで、要素のすべてのインスタンスが返されます。

パスに角括弧で囲んだ整数を使用して、要素の特定のインスタンスを指定することができます。

#### 例:

/tickets/ticket/details/comment とすると、すべてのチケットのすべてのコメントの詳細が指定されます。

/tickets/ticket/details/comment[2] とすると、各チケットの2つ目のコメントが指定されます。

/tickets/ticket[2]/details/comment とすると、2つ目のチケットのすべてのコメントが指定されます。

- 2. [子の名前] ボックスに、抽出する要素の子である要素の名前を入力します。子要素に値がある場合、[値] ボックスに値を入力することで結果を絞り込めます。
  - [**子の名前**] ボックスは、単一レベルの子要素に対してのみ動作します。フィルタリングの結果、指定した子要素の直接の親が返されます。
  - [値] ボックスは、簡略化した値を処理します。ここに入力する値は、抽出する要素の子要素の値と正確に一致している必要があります。
- 3. [属性名] ボックスに、抽出する要素に一意の属性名を入力します。さらに結果を絞り込むには、[値] ボックスに属性の値を入力します。

例:例のXMLでは、customer要素とそのコンテンツを抽出する複数の方法があります。

- [要素パス] ボックスに /ticket/customer と入力する
- [子の名前] ボックスに、次のように customer の子要素のいずれかを入力する。

company

position

contactInfo

[子] ボックスに company と入力した場合、付属の[値] ボックスに myOrg と入力することができます。

• [属性名] ボックスに、次のいずれかを入力します。

名

姓

付属の[値] ボックスに、次のようにこれらの属性に対応する値を入力できます。

John

Doe

これらのフィルターのそれぞれで、出力は次のような customer 要素になります。

#### XML 要素値の取得

[XML 要素値の取得] フィルターで、特定の要素の値を取得できます。



[要素パス] ボックスに、対象の値の要素へのパスを入力します。

ほかのフィルターと同様、要素に複数のインスタンスがある場合、別のインスタンスを指定しない限り、 フィルターは最初のインスタンスを返します。

**例**: サンプル XML の場合

電子メール要素の値を取得するには、/tickets/ticket/customer/contactInfo/email と入力します。

出力は指定した2つの電子メールのいずれかです。

jdoe@myorg.com

johnsSecondEmail@myorg.com

電子メール要素の特定のインスタンスを指定するには、/ticket/customer/contactInfo/email[2] と入力します。

次のように出力されます。johnsSecondEmail@myorg.com

#### XPath クエリ

[XPath クエリ] フィルターで、 [XPath クエリ] ボックスに入力した標準の XPath 構文を使用したクエリで 結果からデータを抽出することができます。

「詳細: XPath クエリー XPath クエリに基づいて XML 文書をフィルターし、 クエリの結果を返します。				
XPath クエリ:				

[XPath クエリ] ボックスに、XPath 構文を使用してクエリを入力します。

- 角括弧の前のパスで、結果を絞り込むクエリの範囲を指定します。
- 角括弧内には、クエリのフィルタリング部分を入れます。クエリに、複数のフィルターのセットを含めることができます。

**例**: サンプル XML の場合

次のいずれかのクエリを使用して、40,000単位を超えるボリュームの顧客を抽出できます。

この XPath クエリでは、40,000 を超えるボリュームの顧客を抱える企業がすべて検索されます。

/tickets/ticket/customer/company[../volume>40000]

<volume> 要素は、<company> タグの兄弟のため、要素 <volume> を探すためには、角括弧内のシーケンスを次のようにして、<company> に対するパスを記述します。

../

• この XPath クエリでは、40,000 単位を超えるボリュームの顧客がすべて検索されます。

/tickets/ticket/customer[volume>40000]

<volume>は <company>の子であるため、その相対パスを指定する必要はありません。

## 参照情報

フィルターエディター

左上のフィルターリストに作成したフィルターのリストが表示されます。

インスペク	<b>ター</b>								- +
ステップ名: Get Performance Counter Value									
入力	入力 結果 表示   説明   詳細   スクリプトレット								
<b>▲ ▼</b>   <b>λ</b>	テップ結果:	value							$\bullet \models \mathbf{x}$
また。 また。 また。 また。 また。 また。 また。 また。 また。 また。	<ul> <li>▲ ▼ XF97編集 &gt; value</li> <li>道加 削除 ① ↓</li> <li>詳細:形式</li> <li>入力の先頭や末尾に解決された表現を追加したり、入力を解決された表現で置 き換えることで、入力の形式を調整します     テキスト ¥n     入力の配置: 末尾に追加     </li> </ul>						を解決された表現で置		
   フィルタ・   すべ	ー入力のテス てのフィルタ・	₹► −ወ <del>,</del> ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	選択した	7ィルターのう	גר			₽ E %	אעדבלעיאל קונל
	PID	PPID	PGID	WINPID	TTY	UID	STIME	COMMAND	
	3016	1	3016	3016	0	1003	09:19:50	/usr/bin/rxvt	
	1424	3016	1424	3964	1	1003	09:19:50	/usr/bin/bash	
	2056	1424	2056	3080	1	1003	09:19:56	/usr/bin/ps	-
」ピート 「出力の <sup>5</sup>	テスト				I				
	PID	PPID	PGID	WINPID	TTY	UID	STIME	COMMAND	
	3016	1	3016	3016	0	1003	09:19:50	/usr/bin/rxvt	
	1424	3016	1424	3964	1	1003	09:19:50	/usr/bin/bash	
	2056	1424	2056	3080	1	1003	09:19:56	/usr/bin/ps\n	V V
	いぐりター								
設計  ブ	) ロバティ								

フィルターを作成して、フィルターの種類を選択する場合は、右上の[詳細:] セクションは、選択したフィルターの種類に従って、フィルターを変更するコントロールが表示されるように変更されます。

GUI ፖイテム	説明
追加	クリックすると新しいフィルターが追加されます。
削除	クリックすると選択したフィルター(複数も可)が削除されます。
<b>1</b>	クリックすると、選択したフィルターがリストの上または下に移動しま す。フィルターは、リストに表示されている順番で処理されます。
フィルター入力のテスト	フィルターが想定どおりに動作するかテストするためのデータを配置 する場所です。
すべてのフィルターのテスト	テストを、出力または結果のすべてのフィルターに適用します。

選択したフィルターのテスト	テストを選択したフィルターに適用します。
コピー	[フィルター入力のテスト] ボックスのデータをコピーします。
貼り付け	[フィルター入力のテスト]ボックスにデータを貼り付けます。
切り取り	[フィルター入力のテスト]ボックスのデータを切り取ります。
クリア	[フィルター入力のテスト] ボックスのデータを消去します。
クイックコマンド	フィルターをテストするデータを生成するコマンドを入力します。 コマ ンドの出力が [ <b>フィルター入力のテスト</b> ] ボックスに表示されます。
出力のテスト	フィルターが [ <b>フィルター入力 のテスト</b> ] ボックスのテスト データに適用 され、フィルタリング結果が [出力のテスト] ボックスに表示されま す。

## 変数の使用

変数を使用すると、フロー内やフロー間でデータを移動できます。

たとえば、サーバーで実行する複数のステップがある場合、最初のステップでサーバーの IP アドレスを 取得し、その値をフロー変数に割り当てることができます。以降、同じ名前の入力を持つ後続のス テップではすべて、自動的にそのサーバー名が使用されます。

#### フロー変数

フロー変数は、フロー変数が定義されているフロー内でのみ使用できます。

### フロー変数への値の割り当て

次の値をフロー変数に割り当てることができます。

- ステップの結果 たとえば、該当項目をカウントするオペレーションが含まれたステップによって、結果をフロー変数に格納します。
- 入力値 たとえば、IP アドレスを入力値として取得するステップによって、アドレスをフロー変数として格納します。
- スクリプトレット たとえば、ステップのオペレーションから戻されたデータを評価するスクリプトレットによって、データをフロー変数に格納します。

## フロー変数の使用

フロー変数およびフロー変数に格納されているデータを、次の場所で参照できます。

- 同じフロー内の**別のステップ**
- 並列分岐ステップのレーン内 同じレーン内の、または並列分岐ステップより前の先行ステップによってフロー変数に値が書き込まれた場合、レーンステップでフロー変数の値を使用できます。ただし、フロー変数の値が別のレーン内のステップによって変数に書き込まれた場合は、レーン内のステップでこの値を使用することはできません。
- オペレーションの入力
- フロー、ステップ、トランジションの説明 たとえば、ping の遅延オペレーションは、ping の平均応答時間をフィルターにより除外します。このオペレーションに関連付けられているステップはこの平均応答時間をフロー変数 latency として保存し、さらにこのステップに続くトランジションがその値をユーザーにレポートすることが可能です。
- レスポンスルールを使用してテストするデータの一部 たとえば、フロー変数に格納した値が出力 文字列またはエラー文字列に含まれているかどうかを確認します。
- スクリプトレット内 スクリプトレットの結果をステップの外部で利用できるようにするには、スクリプトレットでフロー変数を作成(必要なフロー変数が存在しない場合)し、このフロー変数に結果を割り当てる必要があります。
- オペレーションのパラメーター内 オペレーションのパラメーターが値をとる場合、その値が格納されているフロー変数を参照すると、値にアクセスできます。

[フロー変数]ペインを使用すると、作成したフロー変数を追跡できます。

### グローバル変数

グローバル変数は、グローバルコンテキストに含まれているキー名と値のペアであるため、常にすべての フロー実行で使用または参照できます。

フロー変数とグローバル変数の名前が同じである場合、その変数名の参照は、グローバル変数ではなく、その名前の(ローカルの)フロー変数にアクセスします。これは、変数に値を割り当てる場合、または変数の値を取得する場合の、いずれの場合にも当てはまります。

入力の値をグローバル変数から取得するように指定している場合、フロー変数はそのグローバル変数の値を使って作成され、この値がフロー変数から入力に提供されます。

## ベストプラクティス

- 大文字と小文字に一貫性があるようにします。たとえば、すべてのフロー変数名にキャメルケース を使用します。
- フロー変数の種類ごとに異なる命名規則を使用します。たとえば、フローの入力にはFI、ステップの入力にはSI、オペレーションの入力にはOIなど、変数のタイプに基づいて変数の名前に接頭辞を追加します。

フロー変数には、フロー全体からアクセスできます。いずれかのステップでデータが誤って変更されると、フローの後続のステップでそのデータの間違った値が使用されるため、フロー変数の操作に注意してください。

## 実行する操作

#### 入力の値をフロー変数に割り当てる

デフォルトでは、入力の値は、この入力と同じ名前を持つフロー変数に割り当てられます。

- 1. [プロパティ] シート (オペレーションの場合) またはステップのインスペクター (ステップの場合)を開きます。
- 2. [入力]タブで、入力を選択するか、または新規作成します。
- 3. 該当する行の端にある右向き矢印 🚱 をクリックして、入力エディターを開きます。
- 4. [変数への割り当て]ボックスで、値を割り当てる変数の名前を指定します。
- 5. 保存します。

入力の作成の詳細については、「入力の作成」(118ページ)を参照してください。

#### 結果の値をフロー変数に割り当てる

- 1. [プロパティ] シート (オペレーションの場合) またはステップのインスペクター (ステップの場合)を開きます。
- 2. [結果]タブで、該当する結果の行を選択します。
- 3. [割り当て先] リストから [Flow Variable] を選択します。
- 4. [名前] で、フロー変数の名前を指定します。
- 5. [ソース] で、値のソースを指定します。

結果の作成の詳細については、「出力と結果の作成」(147ページ)を参照してください。

必要に応じて、厳密な結果を取得するには、結果に1つまたは複数のフィルターを作成します。 「出力および結果のフィルタリング」(156ページ)を参照してください。

#### スクリプトレットの値をフロー変数に割り当てる

スクリプトレットで値を作成してフロー変数に割り当てることもできます。

スクリプトレットに、次の構文のコマンドを含めます。

scriptletContext.putLocal("<localflowvariablename>", <value>);

<value>には、スクリプトレット内で作成されるオブジェクトまたは変数を指定できます。

#### [フロー変数]ペインに情報を表示する

[フロー変数] ペインを使用すると、フロー変数へのデータの格納について、次のことを追跡できます。

- 必要なときにデータを使用できるように、フローでフロー変数を使用する仕組み
- フロー変数がデータを取得する場所

[**フロー変数**] ペインに、この情報がソリー構造で表示されます。[**フロー変数**] ペインには、現在のフローで使用されているすべてのフロー変数が表示され、各フロー変数の作成と使用が一覧表示されます。フロー内のフロー変数に行った変更は、[**フロー変数**] ペインに自動的に反映されます。

[フロー変数] ペインを開くには、Studio ウィンドウの右上にある[フロー変数] タブをクリックします。

フロー変数 - Restart Service ロロ				
🔶 🔿 🔽				
名前	#	•	<b>&amp;</b>	
- altpass	8	v	v	
□ → フロー入力		v	v	
		v		
		v	v	
入力 "altpass" の値を "altpass" に割り当て		v	v	-
🖻 ≔ ユーザープロンプトを使用しないステップ入力				
"altpass"の値は "Does Service Exist"の入力 "altpass" に割り当てることがで…				
入力 "altpass" の値を "Does Service Exist" の "altpass" に割り当て				
- 入力 "password" の値を "ServiceStatus" の "altpass" に割り当て				
- altuser		v	v	
□ → フロー入力	3	v	v	-

### [フロー変数] ペインの情報をフィルタリングする

最も関心のあるフロー変数の使用に焦点を絞るため、ペインに表示するフロー変数の使用を選択 できます。

- 1. [フロー変数] ツールバーの [フィルター] ボタン ママ をクリックして、フィルターボタンを表示します。
- 表示されたボタンの行で、ボタンをクリックして、各フィルタータイプのオンとオフを切り替えます。各 データソースタイプを切り替えると、そのタイプのデータソースが表示されたり、表示から削除されま す。
  - ■フロー入力
  - ユーザープロンプトのステップ入力

注: ユーザープロンプトは、このバージョンの HP OO ではサポートされていません。

- ユーザープロンプトが設定されていないステップ入力
- 結果
- スクリプトレット

#### フロー変数のリストの参照先の入力を見つける

特定の使用を表示して、その使用を定義するエディターを開くには、[フロー変数]ペインで使用インスタンスを選択します。

- 使用がフロー入力の場合、[入力]タブにフローの[プロパティ]シートが開いて、関連する入力の入力エディターが開きます。
- 使用がステップ入力または結果の場合、選択したステップが含まれたフロー図が開きます。フロー 図の下に、入力または結果のエディターが開きます。

**ヒント**:使用のリストを上に移動するには、[前へ] < ボタンを使用し、使用のリストを下に移動するには、[次へ] < ボタンを使用します。

#### グローバル変数を表示する

フロー内のすべてのグローバル変数を表示するには、フローをデバッグします。 グローバル変数 (フロー 変数も同様) および変数の現在の値が、コンテキストインスペクターに表示されます。

コンテキストインスペクターの詳細については、「コンテンツの検査」(207ページ)を参照してください。

#### グローバル変数を変更する

**重要**: グローバル変数の値を変更する前に、グローバル変数はすべてのフローのすべての実行 に使用できることに留意してください。グローバル変数の値を変更すると、そのグローバル変数を 使用するほかのフローとオペレーションに影響します。

グローバル変数を変更するには、タスク入力の値をフロー変数に割り当てるを完了します。[変数への割り当て] ボックスで、値を割り当てるグローバル変数の名前を入力します。

#### フロー変数をデフォルトでグローバルにする

サブフローで発生する場合を除き、フロー内のすべてのフロー変数がデフォルトでグローバルになるよう に、フローを設定できます。

- 1. フローの[フロープロパティ] シートを表示します。
- 2. [詳細] タブを選択します。
- 3. [サブフローではない場合に、フロー変数をグローバルにする] チェックボックスをオンにします。

## 参照情報

## [フロー変数]ペイン

作成ペインでフローを開いている場合、[フロー変数]ペインに各フロー変数がアルファベット順に表示 され、フローの各使用の説明(フロー変数を使用できるフロー内の各場所)が表示されます。

フロー変数 - Restart Service 🛛 🖸 🖸				
名前	#	=>	<b>&amp;</b>	
- altpass	8	v	v	
□□-入力		v	v	
		v		
		v	v	
入力 "altpass" の値を "altpass" に割り当て		v	v	
□ 🛥 ユーザープロンプトを使用しないステップ入力				
入力 "altpass" の値を "Does Service Exist"の "altpass" に割り当て				
- 入力 "password"の値を "ServiceStatus"の "altpass" (ご割り当て				
≓ altuser		v	v	
📴 🔿 フロー入力			v	-

GUI アイテム	説明
4	クリックすると、使用のリスト内を上に移動します。
	クリックすると、使用のリスト内を下に移動します。
	クリックすると、フィルターボタンが表示されて、[ <b>フロー変数</b> ] ペインに表示されている 情報をフィルタリングできるようになります。
#	フロー内でこのフロー変数を使用した回数を表示します。
•	フロー変数の特定の使用がフロー入力で発生した場合、チェックされます。
<u>2</u>	フロー変数の特定の使用がユーザー入力から値を取得している場合、チェックされます。

## [フロー変数] ペインのフィルターボタン

[フロー変数] ツールバーの[フィルター] ボタン ママ をクリックして、フィルターボタンを表示します。

# フィルターボタンをクリックして、各フィルタータイプのオンとオフを切り替えます。各データソースタイプを切り替えると、そのタイプのデータソースが表示されたり、表示から削除されます。

フィルターボタ ン	説明
	フロー入力 - フロー変数が入力内で参照されます。
	ユーザープロンプトを使用したステップ入力
	ユーザープロンプトを使用しないステップ入力
	<b>注</b> : ユーザープロンプトは、このバージョンの HP OO ではサポートされていません。
<b></b>	結果 - フロー変数がステップ結果に関連付けられます。
Ī	スクリプトレット - フロー変数 がスクリプトレット で参照されます。

## リターンステップの作成

フローには、フローを終了する1つ以上のリターンステップが必要です。



リターンステップの種類には次の4つがあります。

- 解決済み 🔮 正常に実行されているフローの標準のリターンステップです。
- 診断済み 3 何が問題なのかを特定して、通知以外の操作を実行しないことをフローが選択したことを示します。

 ・操作なし □ - データを収集したが、診断か修復かを修復フローが決定できない場合に使用されます。

**注:** データ収集のみを目的としたフローでは、フローが完了すると、操作なし ○ ではなく解決 済み ◇ を返します。

エラー ○ – フローの実行が最後まで失敗した場合に使用されます。たとえば、入力が正しくない場合や、システムに到達できない場合や、フローに問題がある場合に使用されます。

各リターンステップ名で、コロンの後にあるのがリターンステップのレスポンスです。たとえば、エラー: 失敗 のようになります。このレスポンスを編集できます。たとえば、エラー: 失敗をもたらした結果が、オペ レーションの失敗ではなく、要求されるしきい値に到達しない結果だった場合、この結果を反映す る、エラー: しきい値に到達していないなどの新しいレスポンスをエラー: 失敗ステップに作成できます。

## ベストプラクティス

- フロー内に同じ種類の終了ステップが複数ある場合(複数のエラー終了ステップなど)、失敗の原因を含めるように終了ステップの名前を変更します。
- オペレーションの失敗と否定の結果を混同しないようにします。たとえば、回答がTRUEまたは FALSEとなる質問をオペレーションがした場合、回答 FALSE は失敗と同じことではありません。このような場合、結果がFALSEの場合とオペレーションの失敗の場合の、2つのエラーで結果を返 す必要があります。


## 実行する操作

### フローにリターンステップを追加する

- 1. 作成ペインのツールバーで、[ステップパレット]ボタン <sup>6</sup> マ をクリックして、ステップパレットを表示します。
- 2. ステップパレットから、使用するリターンステップのアイコンを作成キャンバスにドラッグします。
- 3. フローステップからリターンステップへのトランジションを作成します。

### リターンステップのレスポンスを変更する

リターンステップのレスポンスは、そのリターンステップに至った結果をより具体的に表すように変更できます。たとえば、フローに複数のエラーレスポンス (エラー: 失敗とエラー: しきい値に到達していない) がある場合、[エラー] 〇 アイコンを作成 キャンバスにドラッグしたときに、必要なレスポンスがエラーリターンステップに含まれない可能性があります。

- 1. 作成ペインでリターンステップを右クリックし、[レスポンスの選択]を選択します。
- 2. リターンステップに必要なレスポンスを選択します。たとえば、「エラー: しきい値に到達していない」を選択します。

### 新しいレスポンスを作成してリターンステップに割り当てる

使用可能なレスポンスのリストに必要なレスポンスが含まれていない場合、カスタムのレスポンスを作成できます。

- 1. 作成ペインでリターンステップを右クリックし、[レスポンスの選択]を選択します。
- 2. [新規レスポンスの追加]を選択します。
- 3. ダイアログボックスで、新しいレスポンスの名前を入力し、[OK]をクリックします。

### 参照情報

### ステップパレット

ステップパレットには、リターンステップ、並列分岐ステップ、複数のインスタンスステップ、およびコールアウトをフローにドラッグするためのボタンがあります。 ステップパレットを表示 するには、作成ペインのツー

をクリックします。



ルバーで [**ステップパレット**] ボタン

ボタン	説明
成功 🥝	<b>成功</b> リターンステップをフローにドラッグできます。
診断済み 📀	診断済みリターンステップをフローにドラッグできます。
操作なし	操作なし リターンステップをフローにド ラッグできます。
失敗 😳	<b>失敗</b> リターンステップをフローにドラッグできます。
並列分岐ステップ 🔧	並 列 分 岐 ステップをフローにド ラッグできます。
複数のインスタンスステップ 💽	複数のインスタンスステップをフローにドラッグできます。
コールアウト 🤎	コールアウトをフローにド ラッグして、ユーザーに情報を提 供 できます。
ドッキングバー	クリックして、パレットのドッキングとドッキング解除を行いま す。

# 高度な作成

この章では、より複雑なフローの作成について説明します。単純なフローの作成については、「フロー 作成 – 基本」(97ページ)を参照してください。

フローの作成時、メモリを無制限に増大させるようなフローは作成しないでください。フローが休止し、 ー部のタスクを実行した後にまた休止する、といった無限のループを実行するフローなどは作成しな いでください。この場合、システムでメモリがなくなるまで実行履歴が増大します。

フロー内 でのサブフローの作 成	
並列分岐ステップのフローの作成	
複数のインスタンスステップのフローの作成	
フロー内 でのスクリプトレットの使用	
フローでの正規表現の使用	

# フロー内でのサブフローの作成

サブフローからステップを作成して、フローを単純化できます。次のことが可能です。

- プログラミングタスクを、管理しやすいように小さく分割する
- フローの部分ごとにテストを行う
- 作成したステップを再使用する

たとえば、以下のフローにおいて Windows Health Checkのステップはサブフローです。



サブフローは、複数のオペレーションが含まれている場合でも1つのステップとして処理されます。

サブフローでは主に、親フローのステップがアクセスする必要のあるデータを生成します。フロー内で作成したフロー変数は、そのフロー外では参照できません。ただし、サブフローの結果をフロー出力フィー ルビとして保存することにより、サブフローから親フローに値を渡すことが可能です。

### ベストプラクティス

- フローは、Studioの最大の大きさである1024 x 768の画面上のキャンバスに、1:1の表示倍率に合わせて作成します。これより大きいフローにすることも不可能ではありませんが、フローが大きくなる場合、そのステップのシーケンスの一部をサブフローに分割できないかをよく確認してください。
- 最上位の親フロー内のすべてのトランジションに説明と名前を付けます。これらのトランジションの 説明は、トランジションの先行ステップで発生する内容について説明します。データが実行中に必 ず表示されなければならない場合を除き、サブフローのトランジションの説明を追加する必要はあ りません。

## 実行する操作

### サブフローのあるフローの作成

- 1. まずフローを作成し、保存します。
- 2. 親フローとして機能する新しいフローを作成します。
- 3. [**プロジェクト**] ペインからこのサブフローを親フローにドラッグし、サブフローからステップを作成します。

### サブフローから親フローへのデータの引き渡し

- 1. 作成 キャンバスでサブフローを開き、親フローでデータを使用可能にするステップのインスペクター を開きます。
- 2. [結果] タブをクリックし、結果を追加します (詳細については、「ステップの結果の設定」(150ページ)を参照してください)。
- 3. 結果データがフロー出力フィールドに保存されるように結果を設定します。これによって、サブフ ロー外でデータが使用可能になります。

入力 結果 表示   説	明 詳細 スクリプトレット							
▲ ▼ ステップ結果							${}^{+}{}^{+}$	×
			の追加	結果の消	除	1	Ŧ	r
名前	ソース	割り当て先	割り当	行操作		フィルター		
FailureMessage	結果フィールド: Result	Flow Output Field	OVERWR	ITE	フィルタ	いなし	(	<>

- a. [名前] で、フロー出力フィールドの名前を入力します。
- b. [ソース] で、[結果フィールド: Result]を選択します。

- c. [割り当て先] で、[Flow Output Field] を選択します。
- d. 必要に応じてフィルターを作成し、結果をフィルター処理します(詳細については、「出力お よび結果のフィルタリング」(156ページ)を参照してください)。
- 4. 親フローの作成キャンバスで、サブフローから作成したステップのインスペクターを開きます。
- 5. [結果] タブをクリックし、ステップの結果を作成します。 デフォルトでは、この新しい結果により以下が行われます。
  - サブフローのフロー出力フィールドの名前が付いた結果フィールドから値を取得する
  - サブフローのフロー出力フィールドと同じ名前を持つ
  - デフォルトで結果と同じ名前を持ち、このステップに続くトランジションとステップで使用可能になった、フロー変数に割り当てられる

#### 例

- 1. コマンド操作をコピーし、"dir C:\"を実行するようにします。これに dir という名前を付けます。
- 2. flowdir という名前のフローを作成します。
- 3. flowdir フローで、オペレーション dir を使用してステップを作成します。
- 4. dir ステップで、オペレーションの出力文字列から得られる結果を追加します。
- 5. 結果をフロー出力フィールドに割り当て、結果に foo という名前を付けます。これで、フローにも foo という名前のフロー出力フィールドが追加されました。
- 6. parentflow という名前の別のフローを作成します。
- 7. parentflow で、flowdir からステップを作成します。
- 8. flowdir ステップに結果を追加します。

新しい結果の名前は、デフォルトでは foo です。これは、[結果フィールド: foo] から値を取得し、 その値が foo という同名のフロー変数に割り当てられます。これで、サブフローのステップ dir の結 果 foo は、親フローの flowdir ステップの後に続くトランジションとステップで使用可能になります。

- 9. これをテストするには、flowdir ステップの後に基本通知オペレーションから作成されたステップを追加します。
- 10. この新しいステップで、以下を行います。
  - a. 入力 notifyData を定数を使用する単一値として定義し、定数に \${foo} を指定します。
  - b. 入力 notifyMethod を定数を使用する単一値として定義し、定数に Display を指定します。

- c. 入力 subject を定数を使用する単一値として定義し、定数を次のように指定します。正常に動作した場合、フロー出力フィールドに「fooという outputString の内容」と表示。
- d. フローをデバッグします。

# 並列分岐ステップのフローの作成

並列分岐ステップは、同時に実行される一連の連続したステップです。各連続ステップは、フロー図内にレーンとして視覚的に表されます。各レーンに含まれるステップを「レーンステップ」と呼びます。フローを実行すると、レーンが同時に開始されます。

並列分岐ステップは、それぞれに独立した異なる処理を同時に実行する場合に最適です。これに 対して、複数のインスタンスステップでは、複数の異なる単一入力を使ってインスタンスが同じ処理を 実行します。

たとえば、並列分岐ステップを使用すると、XML ファイルの記述と検証を実行すると同時に、担当者にこの記述と検証について電子メールを送信できます。

- 1番目のレーンに、ファイルの記述と検証のステップが含まれます。
- 2番目のレーンによって電子メールが送信されます。



# 実行する操作

並列分岐ステップを作成する

1. 作成ペインのツールバーで、[ステップパレット] ボタン <sup>きゅ</sup> <sup>を</sup>をクリックして、ステップパレットを表示します。

- 2. ステップパレットから、[並列分岐ステップ] <sup>1</sup> アイコンを作成キャンバスにドラッグします。 デフォルトでは、ステップには2つのレーンがあります。
- 3. 各レーンに含めるステップシーケンスを作成します。
  - a. レーンにステップ (フローまたはオペレーション)を追加します。
  - b. 各レーンのステップを接続します。
  - c. レーン内の最後のステップを[レーンエンド] アイコン
- 4. 複数のインスタンスステップを残りのフローに接続します。
  - a. 並列分岐ステップが開始ステップではない場合、この並列分岐ステップに先行するステップを [並列分岐ステップ] - アイコンに接続します。
  - b. 並列分岐ステップの[完了] 🥝 レスポンスをフローの次のステップに接続します。

#### レーンの表示順序を変更する

フロー図内のレーンの表示順序を変更できますが、フローを実行すると、すべてのレーンが同時に開始されます。フロー図内の表示順序は、処理の順序には影響しません。

- 1. 移動するレーンを右クリックします。
- 2. ドロップダウンメニューから、[レーンを上に移動] または [レーンを下に移動] を選択します。

### 並列分岐ステップまたはコンポーネントを移動する

- 並列分岐ステップを移動するには、フロー図内の[並列分岐ステップ] <sup>1</sup> アイコンをクリックして ドラッグします。
- 個々のレーンステップを移動するには、ステップを選択し、レーン内または別のレーンにドラッグします。

#### 並列分岐ステップをコピーする

- 1. フロー図内で[**並列分岐ステップ**] <sup>1</sup> アイコンを右クリックし、[**コピー**]を選択します。
- 2. 作成キャンバスを右クリックし、[貼り付け]を選択します。

### 並列分岐ステップのコンポーネントをコピーする

並列分岐ステップのコンポーネントをコピーするには、次のいずれかのツールを使用します。

- [編集] > [コピー] メニューコマンドと[編集] > [貼り付け] メニューコマンド
- 右 クリックメニュー
- キーボードの組み合わせCtrl+C、Ctrl+V
- 作成ペインツールバーの[コピー] → ボタンと[貼り付け] = ボタン

**注:** レーンをコピーする場合、[貼り付け] コマンドを実行するときに、カーソルをレーン内に置きます。

### レーンを追加する

- 1. 既存のレーンを右クリックします。
- 2. ドロップダウンメニューから、[レーンの追加]を選択します。

現在選択しているレーンの下に、新しい空のレーンが追加されます。

### レーンを複製する

- 1. 既存のレーンを右クリックします。
- 2. ドロップダウンメニューから、[レーンの複製]を選択します。

コピーしたレーンのすぐ下に、同じタイトルを持つ新しいレーンが表示されます。

#### レーンを削除する

レーンを削除するには、次のいずれかのツールを使用します。

- [編集] > [レーンの削除] メニューコマンド
- 右 クリックメニュー
- キーボードの組み合わせCtrl+X
- 作成ペインツールバーの[削除] 样 ボタン

### レーンのサイズを変更する

- 1. レーンの空白部分をクリックして選択します。辺と角にハンドルが表示されます。
- 2. 辺または角のハンドルをドラッグします。

### レーンの名前を変更する

デフォルトでは、レーンの名前はレーン1、レーン2などとなります。

- 1. レーンを右クリックして [名前の変更]を選択します。
- 2. 表示されたテキストボックスに、レーンの新しい名前を入力します。

### レーンの開始ステップを変更する

レーンの開始ステップには、フローの開始ステップのような緑色の枠がありません。

[**レーン開始**] アイコン 😳 コネクターを、現在の対象のレーンステップから、レーンの開始ステップにする ステップにドラッグします。



#### 並列分岐ステップとの間でデータを移動する

並列分岐ステップが開始すると、ステップの各レーンはグローバルコンテキストのフロー変数のコピー、 ローカルコンテキストの変数のコピー、および並列分岐ステップ自体の入力を取得します。各レーンで は、通常のフロールールに従って、ほかのレーンとは関係なく、これらの変数を使用したり、これらの変 数を作成、変更、または削除できます。

1 つのレーン内のステップから別のレーン内のステップに値を渡すことはできません。 各レーン内のステッ プには、 並列分岐ステップ (およびステップのレーン) が開始されたときに利用できるようになった値のみ 含まれています。

レーンで実行が終了すると、各コンテキストのフロー変数が、呼び出し元のフロー (並列分岐ステップ が含まれているフロー)のコンテキストに戻されてマージされます。マージの順序は、レーンが終了した 順序です。このため、2つのレーンが同じフロー変数に書き込む場合、後に終了したレーンによって 変数の最後の値が書き込まれます。

並列分岐ステップのレーン内のステップでは、ローカルコンテキストとグローバルコンテキストからデータを 取得して、ローカルコンテキストに保存できます。レーンステップは、スクリプトレットで scriptletcontext.putGlobal()メソッドを使用する場合のみ、グローバルコンテキストに書き込み できます。scriptletcontext.putGlobal()を使用する構文については、オペレーションまたはステッ プの[**スクリプトレット**]タブで、**JavaScript**テンプレートを挿入します。

### 並列分岐ステップをデバッグする

実際の実行時には、フローを実行するとレーンは同時に開始されて実行されますが、デバッガーでテ ストすると連続して実行されます。デバッガーでのレーンの実行順序は制御できませんが、一意の名 前を指定すると、実行された順序を確認できます。 この方法では、デバッガーで実稼働環境でのフローの動作が正確には再現されませんが、デバッガーで並列分岐ステップを連続して実行すると、さまざまな条件を制御しながらテストできます。詳細については、「複雑なフローのデバッグ」(220ページ)を参照してください。

# 複数のインスタンスステップのフローの作成

複数のインスタンスステップは、複数のターゲットで同時に実行されるステップです。たとえば、100台の サーバーで Windows 診断フローを実行する場合、100台のすべてのサーバーで同時にフローを実行 する、複数のインスタンスステップを作成できます。

オペレーションのターゲット (この例 では 100 台 のサーバー) は、 複数 のインスタンスステップの入 カリスト で定 義 されます。



複数のインスタンスステップ内に、1 つまたは複数のオペレーションまたはサブフローを含めることができ ます。 複数のインスタンスステップ内のオペレーションまたはサブフローは、ターゲット ごとに 1 回実 行され ます。 この実行を*インスタンス*と言います。

各インスタンスは、インスタンスの開始時に、 グローバルコンテキストとローカルコンテキストの複製を取 得します。 実行時に、インスタンスのステップごとに、 複数のインスタンスステップ内のグローバル変数、 フロー変数、およびフロー出カフィールドを変更できます。

**注**: いずれかのインスタンスで例外が発生した場合、そのインスタンスは停止されます。 ほかのインスタンスは並列で実行されているため、引き続き実行されます。

### 複数のインスタンスステップと並列分岐ステップの相違点

複数のインスタンスステップでは、各インスタンスが別々のターゲットで同じタスクを実行しますが、並列 分岐ステップでは、異なるタスクを実行するように各並列ステップを設定できます。

複数のインスタンスステップでは、実行時にインスタンス数が変化する可能性がありますが、並列分 岐ステップでは、並列ステップの数は一定です。

### フローデータの保存

複数のインスタンスステップのインスタンスで作成されたフロー変数、グローバル変数、およびフロー出 カフィールドは、作成および割り当て元のインスタンスに対してローカルです。これらの変数とフロー出 カフィールドの変数は、以下のいずれかの方法でこのデータを残りのフローで使用できるようにしない場合、レーンの終了時に消去されます。

- 複数のインスタンスステップの結果にデータをバインドする
- データを保存するように、複数のインスタンスステップにスクリプトレットを作成する

### 結果を使用したデータの保存

複数のインスタンスステップの終了後にフロー変数のデータを使用できるようにするため、ステップの結果を複数のインスタンスステップに定義して、インスタンスで作成されたフロー変数から値を取得できます。 複数のインスタンスステップのステップのインスペクターの[結果] タブで、[ソース] 列の[結果 <result>]を選択して、インスタンスで作成されたフロー変数を選択します。

同様の方法で、インスタンスのサブフローで作成されたフロー出力フィールドのデータを保存することもできます。 複数のインスタンスステップのステップのインスペクターの [結果] タブで、[ソース] 列の [結果 フィールド <result>]を選択して、インスタンスで作成されたフロー出力フィールドを選択します。

[割り当て操作]フィールドを設定し、収集した値を使って別の処理を実行できます。たとえば、異なるインスタンスの結果を追加したり、異なるインスタンスの結果を追加したり、異なるインスタンスの結果をすべて足し合わせたり、より新しいインスタンスで古いインスタンスを上書きしたりできます。

以下の例では、複数のインスタンスステップの結果に5つの変数が設定されています。最初の3つの 変数はフロー変数から値を取得し、残りの2つの変数はフロー出力フィールドから値を取得します。

<ul> <li>入力 [結果] 表示   説明   詳細   スクリプトレット  </li> <li>▲ ▼   ステップ結果</li> </ul>						
			結果の追加	結果		
名前	ソース	割り当て先	割り当て挑	樂作		
var1	結果: Result	Flow Variable	OVERWRITE			
var2	結果: Result	Flow Variable	OVERWRITE			
var3	結果: Result	Flow Variable	APPEND			
var4	結果フィールド Result	Flow Variable	OVERWRITE			
var5	結果フィールド: Result	Flow Variable	APPEND			

Instance1とInstance2の2 つのインスタンスが存在し、メインフローのコンテキストは空であり、 Instance2 は Instance1 の後で終了するとします。これらのインスタンスで以下の変数を指定するとします。

- Instance1
  - フロー変数

var1 = xvar2 = y

var3 = w

■ フロー出力フィールド

var4 = z

- Instance2
  - フロー変数

```
var2 = t
```

```
var3 = v
```

■ フロー出力フィールド

var5 = u

複数のインスタンスステップが終了したときに、変数の値は次のようになります。

- var1 = NULL (Instance2 では、この変数の値は存在せず、アクションが上書きされるため)
- var2 = t (Instance2 の値によって Instance1 の値 が上書きされます)
- var3 = wv (Instance2 の値がInstance1 の値に追加されています)
- var4 = NULL (Instance2 では、この変数の値は存在せず、アクションが上書きされるため)

var5 = u

### スクリプトレットを使用したデータの保存

ステップで生成したデータを残りのフローで使用できるようにするには、データを収集して、インスタンスの実行が完了した後も引き続き存在する変数として保存する、スクリプトレットを作成するという方法もあります。

ここに示す例では、スクリプトレットを使用して、実行された各インスタンスが成功したのか失敗したのかを追跡し、このデータを累計し、グローバルコンテキストで使用可能な変数として保存しています。



このスクリプトレットは、インスタンスごとに1回ずつ、複数回実行されます。毎回、現在のインスタンス (scriptletBranchContext)のScriptletContextにアクセスでき、scriptletContextにアクセス することで親フローコンテキストを変更できます。

scriptletBranchContext は scriptletContext と同じアクセスメソッドを備えています。

スクリプトレットの詳細については、「フロー内でのスクリプトレットの使用」(198ページ)を参照してください。

### アップグレード後のマージ

前のバージョンの HP OO からアップグレード すると、[複数インスタンスの切り替え] オプションを使って作成された複数のインスタンスステップがフローに含まれている場合、古いインスタンスを上書きする新しいインスタンスによって、ステップで作成されたグローバル変数が更新されます。

### 実行する操作

### 複数のインスタンスステップを作成する

- 1. 作成ペインのツールバーで、[ステップパレット]ボタン <sup>50</sup> をクリックして、ステップパレットを表示します。
- 2. ステップパレットから、 [複数のインスタンス] 💟 アイコンを作成 キャンバスにドラッグします。
- 3. [プロジェクト] ペインから、フローまたはオペレーションを複数のインスタンスレーンにドラッグします。

注: 複数のフローおよびオペレーションを複数のインスタンスレーンに追加できます。

- 4. 入力を複数の値のリストとして作成して、複数のインスタンスステップのターゲットのリストを設定します。たとえば、フローを実行するサーバーのリストの場合、次の操作を実行します。
  - a. ステップの開始時に[複数のインスタンス] Pイコンをダブルクリックし、複数のインスタンスステップのステップのインスペクターを開きます。
  - b. 入力を作成します。この例では、入力の名前は servers です。
  - c. [必須] チェックボックスをオンにして、種類を[値のリスト]に設定します。

入力  結果  表示  説明  詳細  スクリプトレット							
▲ ▼ 入力の概要	Į						$\leftrightarrow x$
	入力の追加	入力の削除	1		÷		
	入力	必须	頁 種類			ソース	
servers			値のリスト	-	値: server1,ser	ver2,server3	<b></b>

- d. 行の端にある右向き矢印 (🔤)をクリックして、その行の入力エディターを開きます。
- e. [入力区切り記号]ボックスに、区切り記号(リスト内の要素を区切る記号)を入力します。
- f. 値のリストを入力する方法を指定します。たとえば、複数のインスタンスステップを多数のサーバーで実行するには、[定数を使用する]を選択し、[定数値]ボックスでサーバー名を指定し

ます。値のリストを作成する方法には、前のステップの結果を使用する方法や、別のプログ ラムと統合する方法もあります。

<ul> <li>入力 結果   表示   説明   詳細   スクリプトレット  </li> <li>▲ ▼ 入力の概要 &gt; servers</li> </ul>	↓ ► X
名前: servers 入力データフロー 変数から割り当て: servers を指定時: 定数を使用する 変数への割り当て: servers マ	ト 「未指定時: 定数を使用する」構成 定数値: server1,server2,server3
入力のプロパティー 入力区切り文字: 「暗号化 ▼必須 検査形式: 〈未検証〉 ▼ 次の名前で記録 〈実行履歴〉 ▼	

入力の値のリストを作成するオプションの詳細については、「入力ソースの指定」(125ページ)を参照してください。

- 5. 複数のインスタンスステップの複数の異なる部分を接続します。
  - a. [レーン開始] アイコン 🙋 を複数のインスタンスレーンの最初のステップに接続します。
  - b. 複数のインスタンスステップに複数のステップが含まれている場合は、これらのステップを接続 します。
  - C. レーン内の最後のステップのすべてのレスポンス行を[レーン エンド] アイコン



- 6. ターゲットのリストを各内部ステップに適用します。
  - a. 複数のインスタンスレーン内の内部ステップごとに、ステップのインスペクターを開いて入力を追加します。

- b. 入力エディターを開いて、[変数から割り当て]リストで、ターゲットのリストを保持するために 作成した変数を選択します。この例では、この変数は servers です。
- 7. 複数のインスタンスステップのさまざまなインスタンスで収集されたデータを保存するには、結果を 保存するフロー変数を作成します。
  - a. [マルチステップ] Pイコンをダブルクリックし、複数のインスタンスステップのステップのインスペクターを開きます。
  - b. [結果] タブをクリックして結果を追加します。
  - c. [割り当て先]列で、結果をフロー変数に割り当てます。
  - d. データを保持するフロー変数に名前を指定します。たとえば、outcomeと指定します。
  - e. データをどのように保存するのかを決定します。この例では、各サーバーの結果を保存するため、割り当て操作は APPEND になっています。詳細については、以下の「複数のインスタン スステップの出力を保存する」を参照してください。

入力 結果 表示   説	明 詳細 スクリプトレット						
▲ ▼ ステップ結果							$\leftrightarrow \vdash \times$
			結果の追加	結果	の削収余	1	
名前	X	割り当て先	割り当て操	作	-	ィルター	
outcome	結果: outcome	Flow Variable	APPEND		フィルターた	36	<b>\$</b>

- 8. グローバルコンテキストで作成できるよう、複数のインスタンスステップのさまざまなインスタンスで収 集されたデータを保存するには、結果を保存するスクリプトレットを作成します。
  - a. 複数のインスタンスステップのステップのインスペクターで、[スクリプトレット] タブをクリックします。
  - b. scriptletBranchContextのデータを収集するスクリプトレットを作成して、 scriptletContextで使用できるようにします。

次の例では、スクリプトレットを使用して、outcome 変数のすべての値を累計するようにフローに指示しています。これは、前の手順で選択した APPEND アクションと似ています。



- 9. 複数のインスタンスステップを残りのフローに接続します。
  - a. 複数のインスタンスステップが開始ステップではない場合、この複数のインスタンスステップに

先行するステップを「複数のインスタンス」 🔊 アイコンに接続します。

b. 複数のインスタンスステップの[完了] 🔮 レスポンスをフローの次のステップに接続します。

#### 複数のインスタンスステップの出力を保存する

インスタンスのフロー変数とフロー出力フィールドのデータは、複数のインスタンスステップが完了すると 消去されます。このデータを保存するために、データを複数のインスタンスステップの結果にバインドで きます。

- 1. 前述の手順に従って、複数のインスタンスステップを作成します。
- 2. ステップのインスペクターで [結果] タブをクリックします。
- 3. 保存する各フロー変数の結果行を追加します。

入力  結果   表示   説明   詳細   スクリプトレット   ▲ ▼   <b>ステップ結果</b>						
			結果の追加	結果		
名前	ע-ג	割り当て先	割り当て挑	全		
var 1	結果: Result	Flow Variable	OVERWRITE			
var2	結果: Result	Flow Variable	OVERWRITE			
var3	結果: Result	Flow Variable	APPEND			
var4	結果フィールド: Result	Flow Variable	OVERWRITE			
var5	結果フィールド: Result	Flow Variable	APPEND			

- 4. [名前]列に、データを保存するフロー変数の名前を入力します。
- 5. [**ソース**] 列で、保存するデータのソースであるフロー変数またはフロー出力フィールドを選択しま す。
  - インスタンスで作成されたフロー変数を選択するには、[ソース]列で[結果 <result>]を選択します。
  - インスタンスで作成されたフロー出力フィールドを選択するには、[ソース]列で[結果フィールド
     <result>]を選択します。
- 6. [割り当て操作]列で、どのようにデータを収集するのかを表すアクションを選択します。

たとえば、すべてのインスタンスを実行するのにかかる時間を計算する場合、[追加]を選択します。 複数のインスタンスステップでチェックされたすべてのサーバーのリストを収集する場合は、[追加]を選択します。

7. ステップを保存します。複数のインスタンスステップの実行が終了した後で、作成したフロー変数 を残りのフローに使用できるようになります。

#### 複数のインスタンスステップの出力をグローバル変数として保存する

複数のインスタンスステップの出力を保存してフローの外部で使用できるようにするため、この出力を グローバル変数として保存するスクリプトレットを作成できます。

- 1. ステップの開始時に[複数のインスタンス] 2 アイコンをダブルクリックし、複数のインスタンスステップのステップのインスペクターを開きます。
- 2. [スクリプトレット] タブをクリックします。
- 3. scriptletBranchContext のデータを収集するスクリプトレットを作成して、scriptletContext で使用できるようにします。例:



### 複数のインスタンスステップを移動する

- 1. レーンの開始時に、ステップ全体を表す[複数のインスタンス] 「 アイコンを選択します。
- 2. 作成キャンバスでステップをドラッグします。

### 複数のインスタンスステップのサイズを変更する

- 1. レーンの空白部分をクリックして選択します。辺と角にハンドルが表示されます。
- 2. 辺または角のハンドルをドラッグしてレーンのサイズを変更します。

### 複数のインスタンスステップの名前を変更する

- 1. レーンの開始時に、[複数のインスタンス] 🔊 アイコンを選択します。
- 2. 右クリックして [名前の変更]を選択します。
- 3. テキストボックスに新しい名前を入力します。

### 複数のインスタンスステップをデバッグする

実際の実行時には、複数のインスタンスは同時に実行されますが、デバッガーでテストすると連続し て実行されます。つまり、実際の条件下でテストが行われないということですが、各インスタンスが終了 するまでにかかる時間を調べることができます。

詳細については、「複雑なフローのデバッグ」(220ページ)を参照してください。

# フロー内でのスクリプトレットの使用

スクリプトレット (Rhino JavaScript で記述) は必要に応じてオペレーションに追加します。オペレーションの入力データまたは結果データのいずれかを、オペレーションまたはフローのほかの部分で使用できるように加工する目的で使用します。

スクリプトレットを使用すると、結果の特定のデータをテスト、フォーマット、加工、または抽出できます。

スクリプトレットを使用すると、次のことを実行できます。

- オペレーション、フロー、ステップの結果をフィルタリングする
- オペレーションのレスポンスを決定する
- データを親フローに渡す前に、サブフローでデータを加工する

### スクリプトレットを書くのに役立つリソース

- スクリプトレットテンプレート (Rhino JavaScript) がスクリプトレットエディターに用意されています。
- デフォルトのスクリプトレットが Configuration\Scriptlets フォルダーに用意されています。
- デフォルトコンテンツの既存のスクリプトレットをコピーできます。

## 実行する操作

テンプレートからスクリプトレットを作成する

- 1. [プロパティ] シートまたはステップのインスペクターを開きます。
  - オペレーションにスクリプトレットを追加するには、[プロジェクト]ペインでオペレーションを右ク リックし、[プロパティ]を選択します。
  - フローにスクリプトレットを追加するには、[プロジェクト]ペインでフローを右クリックし、[プロパティ] を選択します。
  - ステップにスクリプトレットを追加するには、作成ペインでステップをダブルクリックします。
- 2. [スクリプトレット] タブを選択します。

```
入力 結果 |表示 |説明 |詳細 スクリプトレット
I var split = scriptletInput.split("\n");
2 var out = "";
3 for (count = 0;count < split.length;count++) {</p>
4 var curr = split[count];
5 var index = curr.indexOf("--More--");
6 if (index <0 || index >1)
7 out += "\n"+split[count];
8 }
9 scriptletResult = out;
```

- 3. [テンプレートの挿入]をクリックします。
- 4. テンプレートのガイドラインに従って、スクリプトを作成します。
- 5. [スクリプトの確認]をクリックしてエラーを確認します。
- 6. 保存します。

### 既存のスクリプトレットを使用する

- 1. [プロパティ] シートまたはステップのインスペクターを開きます。
  - オペレーションにスクリプトレットを追加するには、[プロジェクト]ペインでオペレーションを右ク リックし、[プロパティ]を選択します。
  - フローにスクリプトレットを追加するには、[プロジェクト]ペインでフローを右クリックし、[プロパティ] を選択します。
  - ステップにスクリプトレットを追加するには、作成ペインでステップをダブルクリックします。
- 2. [スクリプトレット]タブを選択します。
- 3. 既存のスクリプトレットを別のウィンドウで開きます。
  - Configuration\Scriptlets フォルダーのスクリプトレットをダブルクリックします。
  - スクリプトレットが含まれているオペレーションを開きます(たとえば、Operations\Operating Systems\Linux\Red Hat フォルダーのオペレーション)。
- 4. スクリプトレットのテキストをコピーして、オペレーション、フロー、またはステップの[**スクリプトレット**]テキストボックスに貼り付けます。
- 5. 必要に応じて、スクリプトレットを変更します。
- 6. [スクリプトの確認]をクリックしてエラーを確認します。
- 7. 保存します。

テンプレートの挿入 スクリプトの確認

### スクリプトレットでステップまたはフローの結果をフィルタリングする

スクリプトレットを使用して、ステップまたはフローの結果をフィルタリングできます。

- 1. 作成ペインでステップをダブルクリックします。
- 2. [結果] タブを選択し、フィルタリングする結果を選択します。
- 3. 結果の行の端にある右向き矢印 📴 をクリックして、フィルターエディターを開きます。
- 4. フィルターエディターで、[追加]ボタンをクリックします。
- 5. [フィルターの選択]リストで、[スクリプトレット]を選択します。
- 6. 次のいずれかの方法で、データをフィルタリングするスクリプトレットを作成します。
  - [テンプレートの挿入]をクリックし、スクリプトレットテンプレートをベースとして使用します。
  - 別のオペレーションまたは Configuration Scriptlets フォルダーの既存のスクリプトレットのテキ ストをコピーして貼り付けます。

フィルターの作成の詳細については、「出力および結果のフィルタリング」(156ページ)を参照してください。

- 7. [スクリプトの確認]をクリックしてエラーを確認します。
- 8. フィルターをテストして作業結果を保存します。

**ヒント:** スクリプトレットオペレーションを作成する場合、スクリプトレットで、スクリプトレットレスポンスとして success を指定します。続いて、オペレーションの [レスポンス] タブで、デフォルトのレスポンスとして failure を選択します。

### オペレーションレスポンスのスクリプトレット ルールを作成する

スクリプトレットを使用して、オペレーションのレスポンスを制御できます。

- 1. オペレーションの [レスポンス] タブを開いて、レスポンスを選択します。
- 2. レスポンスの行の右端にある右向き矢印 😂 をクリックして、ルールエディターを開きます。
- 3. [ルールの種類]リストで、[スクリプトレット]を選択します。
- 4. ルールの行の右端にある右向き矢印 🐼 をクリックして、ルール詳細エディターを開きます。

```
入力 結果 |表示 |説明 |詳細 スクリプトレット
I var split = scriptletInput.split("\n");
2 var out = "";
3 for (count = 0;count < split.length;count++) {</p>
4 var curr = split[count];
5 var index = curr.indexOf("--More--");
6 if (index <0 || index >1)
7 out += "\n"+split[count];
8 }
9 scriptletResult = out;
```

- 5. 次のいずれかの方法で、スクリプトレットを作成します。
  - [テンプレートの挿入]をクリックし、スクリプトレットテンプレートをベースとして使用します。
  - 別のオペレーションまたは Configuration\Scriptlets フォルダーの既存のスクリプトレットのテキ ストをコピーして貼り付けます。
- 6. スクリプトレットを作成し、[スクリプトの確認]をクリックしてエラーを確認します。
- 7. 作業結果を保存します。

### オペレーション、フロー、ステップでシステムスクリプトレットを使用する

- 1. システムスクリプトレットを使用するオペレーション、フロー、ステップについて、[プロパティ]シートまた はステップのインスペクターの [**スクリプトレット**] タブを開きます。
- 2. [プロジェクト] ペインで、Configuration フォルダーとScriptlets フォルダーを展開します。
- スクリプトレットを Scriptlets フォルダーから [プロパティ] シートまたはステップのインスペクターの [スクリプトレット] タブの [スクリプトレット] 「アイコンにドラッグします。

[スクリプトレット] タブに、共有のスクリプトレットへの参照があることが表示されます。

入力 | 出力 | レスボンス | 説明 | スクリプトレット | 共有スクリプトレット への参照: /MyProject1/Configuration/Scriptlets/divide by 1k カスタムスクリプトレットに切り替え

### スクリプトレットを Configuration \Scriptlets フォルダーに保存する

- 1. [プロパティ] シートまたはステップのインスペクターの [スクリプトレット] タブで、保存 するスクリプトレットを開きます。
- 2. [スクリプトレット] CALL アイコンを [プロジェクト] ペインの Configuration\Scriptlets フォルダーにド ラッグします。
- 3. スクリプトレットの名前を入力します。

テンプレートの挿入 スクリプトの確認

### 参照情報

### スクリプトレットエディター

スクリプトレットエディターの外観は、[プロパティ]シートまたはステップのインスペクターの[スクリプトレット] タブで開く場合でも、Configuration\Scriptlets フォルダーでスクリプトレットをダブルクリックして開く場合でも同じです。

	入力	結果  表示  説明  詳細  スクリプトレット
	1	
1	var	<pre>split = scriptletInput.split("\n");</pre>
2	var	out = "";
3	for	<pre>(count = 0;count &lt; split.length;count++) {</pre>
4		<pre>var curr = split[count];</pre>
5		<pre>var index = curr.indexOf("More");</pre>
6		if (index <0    index >1)
7		<pre>out += "\n"+split[count];</pre>
8	}	
۵.	acri	intletResult = out:

GUI ፖイテム	説明
スクリプトレットアイコ ン <u>「</u>	このアイコンを Configuration\Scriptlets フォルダーにドラッグすると、スクリ プトレットをフォルダーに保存して再利用できます。
テンプレートの挿入	[ <b>テンプレートの挿入</b> ]をクリックすると、スクリプトレットを書くのに役立つガイ ドラインが表示されます。
スクリプトの確認	[ <b>スクリプトの確認</b> ]をクリックすると、スクリプトレットにエラーがないか確認されます。

# フローでの正規表現の使用

正規表現を使用すると、一致するテキストだけでなく、文字の種類も検索できます。たとえば、ワイル ドカード \dを使用すると、任意の数字を検索できます。

正規表現を使用すると、次のことを実行できます。

- 次のような目的で、データの一部を抽出する結果/出力フィルターを作成する。
  - 後のオペレーションで使用するためにデータを変数に保存する。
  - ステップのレスポンスを決定するためにデータをテストする。

### 正規表現のワイルドカードと修飾子

正規表現の主なワイルドカードは次のとおりです。

テンプレートの挿入 スクリプトの確認

ワイルドカード	用途
٨	文字列の先頭に一致
\$	文字列の末尾に一致
•	改行以外の任意の文字
/b	単語境界
\В	単語境界以外
\d	任意の数字 0 ~ 9
١D	数字以外
\n	改行
١r	キャリッジリターン
ls	空白文字
\S	非空白文字
\t	タブ
\w	任意のアルファベット、数字、またはアンダーバー
\W	アルファベット、数字、アンダーバー以外の任意の文字

正規表現の修飾子は次のとおりです。

修飾子	結果
*	ゼロ(0)回以上の繰り返しに一致
+	1回以上の繰り返しに一致
?	ゼロ(0)回または1回の繰り返しに一致
{n}	n回の繰り返しに完全一致
{n,}	n回以上の繰り返しに一致
{n,m}	n ~ m 回 の繰り返しに一 致
[abc]	a、b、c のいずれかに一致
[^abc]	a、b、c以外の任意の文字に一致
[a-c]	a~cの間の任意の文字に一致
alb	a または b に一 致
1	特殊文字をエスケープする(たとえば\.の場合、'.'がそ れ自身以外のいずれの文字にも一致しないことを意 味します)

### 実行する操作

### 正規表現を使用してテスト出力をフィルタリングする

- 1. 出力または結果のフィルターエディターを開いて、新しいフィルターを作成します。詳細について は、「出力および結果のフィルタリング」(156ページ)を参照してください。
- 2. [フィルターの選択] リストから、フィルターの種類として [正規表現] を選択します。右上の [詳細:] セクションに、正規表現を作成するコントロールが表示されます。

詳細・正規表現 正規表現と一致する部分を抽出して、入力文字列をフィルターします。正規表現の詳細な説明については、ヘルプマニュアルを 参照してください。						
正規表現の種類	Java スタイル					
正規表現の値	*					
フィルターのスタイル:	① 入力全体をフィルター					
	○ 行単位でフィルター					
大文字と小文字を区別しない:						

- 3. [正規表現の種類] リストから [Java スタイル] を選択します。ほかのスタイルは使用しないでください。 推奨されていません。
- 4. [**正規表現の値**] ボックスに正規表現を入力します。

**例**: パケット損失の数を抽出するには、正規表現 Lost = \d を使用します。

この表現で、文字列 "Lost = " とその後に続く任意の数字を検索するよう、 HP OO に指 定します。

ワイルドカード \dを使用して、任意の文字を検索するよう、HP OO に指定します。

- 5. [フィルターのスタイル] で、未加工結果にどのようにフィルターを適用するかに応じて、[入力全体 をフィルター] または [行単位でフィルター] を選択します。
- 6. 正規表現の大文字小文字を区別しない場合は、[大文字と小文字を区別しない]を選択します。
- 7. [選択したフィルターのテスト]をクリックしてフィルターをテストします。
- 8. フィルターを保存します。

#### 複数の正規表現を組み合わせて値を抽出する

複数の正規表現を組み合わせて、フィルターで値を抽出できます。

たとえば、Unixのpsコマンドの出力でpsの時間を抽出するには、2つの正規表現が必要です。1 つ目で出力をpsの行まで絞り込んで、2つ目で時間を抽出します。

F	S	UID	PID	PPID	С	PRI	NI	ADD R	SZ	WCHA N	TTY	TIME	CMD
0	S	512	2160 4	2160 3	0	75	0	-	1096	wait	pts/1	00:00:0 0	Bash
0	R	512	2659	2160 4	0	76	0	-	1110	-	pts/1	00:00:0 0	Ps

1. 出力または結果のフィルターエディターを開きます。

2. 正規表現のフィルターを新しく追加します。

3. [正規表現の値] ボックスに、1つ目の正規表現を入力します。

この例では、.\*psと入力しています。これにより、"ps"で終わる任意の文字列が抽出されます。

注: 先頭のピリオド [.] を省略しないでください。

- 4. [行単位でフィルター] チェックボックスをオンにします。
- 5. [選択したフィルターのテスト]をクリックします。

[出力のテスト] ボックスで出力されるのは、"ps"の文字が入った行だけとなります。

- 6. 正規表現の2つ目のフィルターを追加します。
- 7. [正規表現の値] ボックスに \d\*:\d\*:\d\* と入力します。

この表現は、コロンで区切られた3セットの数字を表します。この例では、この表現によって、行から時間が抽出されます。

- 8. [選択したフィルターのテスト]をクリックします。
- 9. 保存します。

出力のテストに ps 行の時間だけが表示されます。これで、この値を変数に割り当てることができます。

### 参照情報

#### フィルターエディター>詳細:正規表現

[正規表現]をフィルターの種類として選択すると、右上の[詳細:]セクションに正規表現を作成および変更するコントロールが表示されます。

- 詳細: 正規表現- 正規表現と一致する部分を抽出して、入力文字列をフィルターします。正規表現の詳細な説明については、ヘルプマニュアルを 参照してください。					
正規表現の種類	Java スタイル				
正規表現の値	*				
フィルターのスタイル	<ul> <li>入力全体をフィルター</li> </ul>				
○ 行単位でフィルター					
大文字と小文字を区別しない。					

GUI ፖイテム	説明
正規表現の種類	データをフィルタリングする正規表現の種類として [ <b>Java スタイル</b> ] を選択します。ほかのスタイルは使用しないでください。推奨され ていません。
正規表現の値	正規表現を入力します。
フィルターのスタイル > 入力 全体をフィルター	未加工結果全体にフィルターを適用するには、これを選択しま す。
フィルターのスタイル > 行単 位でフィルター	各行に個別にフィルターを適用するには、これを選択します。
大文字と小文字を区別しな い	正規表現の大文字小文字を区別しない場合は、これを選択します。

# コンテンツの検査

コンテンツをリリースする前に、プロジェクトのフローをテストし検査することが重要です。 Studio では、これを行う際に役立つ次のツールを用意しています。

- [問題] ペイン。問題が、発生した場所と内容説明とともに一覧され、この問題を解決するための 指針となります
- デバッガー。フローで見つかったエラーや予期しない動作の原因を突き止めるのに役立ちます

# [問題]ペインでのフローの検査

フローを実行するには、フローそのもの、そのオペレーション、そしてそのフローで使用するすべてのシステムアカウントが有効である必要があります。

[問題] ペインを使用して、個別のフローやオペレーションに問題がないかチェックしたり、プロジェクト全体を検査したりできます。この場合、プロジェクトのすべてのフロー、オペレーション、システムアカウントが検査されます。

問題						
$\nabla$	ソースの種類	名前	記明	場所		
	flow	testFlow	フローにリターンステップがありません	/MyProject1/Library/My Ops		
	flow	testFlow	フローに開始ステップがありません	/MyProject1/Library/My Ops		

### 有効なフローの要件

フローを有効にするには次の条件を満たすことが必要です。

- 少なくとも1つのステップがある
- ステップの1つが開始ステップとして指定されている
- 各ステップについて、各レスポンスから、ステップが後続ステップにトランジションによって接続されている
- フローの各ステップに何らかの実行処理に至る経路がある
- 値を返してフローを終了するリターンステップがある
- 各入力の値の取得方法が割り当てられている

## 実行する操作

### フローまたはオペレーションの検査

- 1. [プロジェクト] ペインでフローまたはオペレーションを選択します。
- 2. [問題] タブをクリックし、[問題] ペインを表示します。

#### プロジェクトのすべてのフローとオペレーションの検査

- 1. 検査するプロジェクトを開きます。
- 2. [ツール] メニューから[フローとオペレーションの検査]を選択します。

問題がある場合、その場所と説明が一覧表示され、問題を修復する方法が示されます。

### 参照情報

#### [問題]ペイン

[問題] ペインは Studio ウィンド ウの下 端 にある [問題] タブをクリックすると開き、選択したフローやオペレーションが有効 かどうかを確認 できます。

問題					
$\nabla$	ソースの種類	名前	記明	場所	
	flow	testFlow	フローにリターンステップがありません	/MyProject1/Library/My Ops	
	flow	testFlow	フローに開始ステップがありません	/MyProject1/Library/My Ops	

GUI ፖイテム	説明
ソースの種類	問題がある要素の種類が表示されます。
名前	問題がある要素の名前が表示されます。
説明	問題の説明と、修復する方法が示されます。
場所	問題がある要素の場所が表示されます。

# フローのテストとデバッグ

ブロジェクトの作成	コンテンツパックの インボート	フローの作成	フローのデバッグ	コンテンツのリリース

デバッガーは、以下の情報を表示し、フローで見つかったエラーや予期しない動作の原因を突き止めるのに役立ちます。

- 実行したステップを表示するツリー
- 各ステップで生成されたステップ結果とオペレーション出力
- 各 ステップのさまざまなコンテキストのフロー変数の値
- 追跡した各トランジションの説明

ようこそ 🗙 🎏 Windows Health Check - デバッグ	x	4 ▷ 🗉
D 🛛 🗶 📥 🛛 🦓		
実行ツー 🛠	トランジション履歴	*
🐲 実行 [Windows Health Check - Debug]	[◇ [11:28:01.389] Display?: <空のトランジションの説明>	
- 🐌 ステップ [Ping Target System] (完了)		
	📗 🙋 [11:28:02.398] Write to File?: <空のトランジションの説問	虏>
白 🕹 ステップ [Notify] (完了)	│ 【爻 [11:28:02.420] None?: <空のトランジションの説明>	
🌺 ステップ [Display?] (完了)	【 2011:28:02.443] リターンステップ:Notify	<b>.</b>
「 100 人丁ツノ [Write to File?] (元子)	コニップ約4月 パリコ ぶちちょ	
■ ステップ [Nothing displayed : success] (完1	スリック福米インスペンター	
マステップ [Resolved : success] (完了)	tresponse=success;FailureMessage=;Result=\${in1};Tim	edOut=false;returnRest
		•
	未加工結果「ブライマリ結果」その他の結果」	
	コンテキストインスペクター	*
فالمساع	ロ ステップ入力	A
ブレークポイントとレスポンスオーバーライド 🗙	Field_1	
	Field_2 \${in2}	
	Field_3	
	Field_4	
	ignoreCase true	
	string 1 None	
	String2 Display	
·	destination	
すべて有効/無効にする	from	
	htmlEmail false	

また、デバッガーのブレークポイントを設定し、テスト対象の動作に集中するためにレスポンスを強制 的に選択することもできます。

# ベストプラクティス

親フローをデバッグする前に、サブフローをデバッグすることをお勧めします。

### 実行する操作

### フローのデバッグ

1. [プロジェクト] ペインでフローを右クリックし、[デバッグ]をクリックします。

注:または、作成ペインでフローを開き、[デバッグ] 🌮 ボタンをクリックします。

- 2. フローを最後まで実行するには、デバッガーのツールバーの[再生] ▶ ボタンをクリックします。
- 3. 完了したステップの情報を確認するには、[実行ツリー]ペインで目的のステップをクリックします。



4. [ステップ結果 インスペクター] ペインでは、ステップの未加工結果、プライマリ結果またはフィルタリング済み結果を表示できます。



5. グローバル変数、フロー変数、ステップの入力でのこれらの変数の値、ステップおよびグローバルコンテキストを確認するには、[コンテキストインスペクター]の該当セクションに移動します。

コンテキストインスペクター	\$
▝▋▕₽↓▕₽\$₽↓│¶	
🗆 ステップ入力	
packetCount	
packetSize	
targetHost	localhost
□ ステップコンテキスト	
host	localhost
notifyMethod	None
output	C:¥HP¥OO¥studio>ping localhost
targetHost	localhost
vitalProcesses	svchost.exe,Isass.exe,smss.exe,Syst
ログローバルコンテキスト	
CENTRAL_URL	

### フローのデバッグ - ステップごと

- 1. デバッガーでフローを開きます。
- 2. フローをステップごとに実行するには、[ステップオーバー] 🔤 ボタンをクリックします。

### サブフローへのステップインとサブフローからのステップアウト

親フローをデバッグする前に、サブフローをデバッグすることをお勧めします。

- ステップのサブフローにステップインするには、 [ステップイン 📩 ボタンをクリックします。
- サブフローからステップアウトするには、 [ステップアウト 🤷 ボタンをクリックします。

### デバッガーでのペインの折りたたみ/展開

ほかのペインにより多くのスペースを使用 できるようにするために、 デバッガーの一部 のペインを折りたた む必要 がある場合 があります。

- ペインを折りたたむには、パネルの右上にある上向きの二重矢印 素をクリックします。
- 折りたたんだペインを元に戻すには、下向きの二重矢印 ¥をクリックします。

### デバッガーでのフローのリセットと再開

フローのリセットと再開を行うと、フロー変数の値がデバッガーを開いたときの値にリセットされます。

- 1. デバッガーのツールバーで、[リセット] 🖓 ボタンをクリックします。
- 2. [再生] 🏓 ボタンをクリックします。

### デバッガーでのフロー変数の値の変更

フロー変数の値を変更したときにフローがどのように動作するかを確認する場合、ステップを実行する前にフロー変数の値を変更できます。

- 1. デバッガーでフローを開きます。
- 2. 目的のステップが保留中になるまで、[ステップオーバー 📩 をクリックします。

**注:** ステップの前にブレークポイントを設定済みである場合、[再生] ▶ をクリックし、ステップが一時停止するまでフローを実行します。

[コンテキストインスペクター] ペインに、ステップが保留中になっているポイントの[ステップ入力] お よび [ステップコンテキスト] の現在の値が表示されます。

コンテキストインスペクター		*
\$ \$↓   ₽\$ ₽\$   ¶		
□ ステップ入力		
host	10.51.0.36,10.51.0.37,10.51.0.38	
□ ステップコンテキスト		
execution_thread_id	AWT-EventQueue-1	
□ グローバルコンテキスト		
execution_userid	admin	

- [ステップ入力] セクションの値は、ステップ開始前に入力に割り当てられた値です。
- [ステップコンテキスト] セクションの値は、ステップ開始後に更新された値です。
- 3. このステップで使用されるフロー変数の値を変更するには、[ステップ入力]で目的のフロー変数のリストを見つけ、その値を強調表示し、新しい値を入力して置き換えます。上記の例では、ステップは複数のインスタンスステップです。host フロー変数のリストに別の IP アドレスを追加できます。

- 4. このステップでアクセス可能だが後続のステップで使用されるフロー変数の値を変更するには、[ス テップコンテキスト]に表示される該当フロー変数のリストの値を変更します。
- 5. フローの実行を続行します。
- 6. 変更したフロー変数の値を、前回のフロー保存時に設定した値にリセットするには、[リセット マンボタンをクリックします。

#### フローのブレークポイントの設定

ブレークポイントは、デバッガーでのフローの実行を自動的に一時停止します。これは、たとえば次の ような処理を行う際に便利です。

- フロー変数の値を調べる
- フロー変数の値を変更して、残りの実行でフローが受ける影響を確認する

ブレークポイントはフロー図で設定しますが、デバッガー内から設定済みブレークポイントを有効または 無効にできます。

- 1. 作成ペインでフローを開き、ブレークポイントを設定するステップを右クリックします。
- 2. [デバッグ] > [ブレークポイントの設定]を選択します。

フロー図では、ブレークポイントは黄と黒の枠で囲まれたステップで表されます。



3. デバッガーでフローを開きます。

デバッガーの[**ブレークポイントとレスポンスオーバーライド**]ペインの[**ブレークポイント**]タブに、既存のブレークポイントが表示されます。



- 4. 次のいずれかを実行します。
  - 1つのブレークポイントを有効にするには、ブレークポイントのチェックボックスをオンにします。
  - 1つのブレークポイントを無効にするには、ブレークポイントのチェックボックスをクリアします。
  - すべてのブレークポイントを有効または無効にするには、[**すべてを有効/無効にする**]をクリックします。
  - すべてのブレークポイントをクリアするには、[ツール]メニューから[すべてのブレークポイントの削除]を選択します。

### 単一ステップのデバッグ実行でレスポンスをオーバーライドする

レスポンスのオーバーライドにより、オペレーションが失敗した場合でも、選択したレスポンスを強制的に指定できます。

レスポンスをオーバーライドすると、デバッガーを終了して入力値を変更せずに、フローの特定のパスを テストできます。

たとえば、フローのステップについて必要な情報がない場合に、このステップが失敗したかどうかに関わらず、フローの残りのステップについてテストが必要な場合があります。ユーザーが介入しない場合に発生する failure レスポンスではなく、目的のレスポンスとトランジションをたどるように強制的に実行を続けることができます。

- 1. 作成ペインでフローを開き、レスポンスをオーバーライドするステップを右クリックします。
- 2. [デバッグ]>[レスポンスのオーバーライド]を選択し、ステップに強制的に指定するレスポンスをクリックします。
  - ∎ なし
  - SUCCESS
  - failure
  - プロンプト

レスポンスのオーバーライドを作成した後、デバッガーでオーバーライドを有効または無効にしたり、別のレスポンスを選択したりできます。

3. デバッガーでフローを開きます。

デバッガーの [**ブレークポイントとレスポンスオーバーライド**] ペインの [**レスポンスオーバーライド**] タブ に、既存のレスポンスのオーバーライドが表示されます。

ブレークポイントとレスポンスオーバーライド	*
✓ Notify	
success	<b>T</b>
すべて有効/無効にする	
ブレークポイント レスポンスオーバーライド	

- 4. 目的のレスポンスオーバーライドにスクロールします。
- 5. 次のいずれかを実行します。
  - オーバーライドに別のレスポンスを選択するには、下向き矢印をクリックしてレスポンスを選択します。
  - 1つのレスポンスオーバーライドを有効にするには、チェックボックスをオンにします。
  - 1つのレスポンスオーバーライドを無効にするには、チェックボックスをクリアします。
  - すべてのレスポンスオーバーライドを有効または無効にするには、[**すべてを有効/無効にする**] をクリックします。
  - すべてのレスポンスオーバーライドをクリアするには、[ツール]メニューから[すべてのレスポンス オーバーライドの削除]を選択します。
  - ステップごとにレスポンスをオーバーライドする場合は、[すべてのレスポンスのオーバーライド]
     チェックボックスをオンにします。

すべてのレスポンスをオーバーライドした後にデバッガーでフローを実行すると、ステップごとに該当ステップのレスポンスを手動で選択するよう求められます。

レスポンスのオーバーライド 🗙			
1	ステップ <b>Create System Health Report</b> によりレスポンス <b>"success"</b> が生成されました。 新規レスポンスを選択してください。		
	success		
	OK キャンセル		

### 参照情報

[実行ツリー] ペイン



[実行ッリー] ペインには、フローのサブフロー内のステップも含めて、実行する各ステップが表示されます。

実際の実行では、同時に実行されるステップは、デバッガーで順に実行されます。

### [ブレークポイントとレスポンスオーバーライド]ペイン

ブレークポイントとレスポンスオーバーライド 🕿	ブレークポイントとレスポンスオーバーライド 🕿
<ul> <li>✓ Windows Health Check:Get FileSystem Information</li> <li>✓ Windows Health Check:Get System Uptime</li> </ul>	Votify success
すべて有効/無効にする ブレークポイント レスポンスオーバーライド	すべて有効/無効にする ブレークポイント レスポンスオーバーライド

ブレークポイントは、特定のステップで実行を自動的に一時停止し、停止ポイントでのフロー変数の値、実行のパス、または結果を調べるためのフラグです。
• レスポンスのオーバーライドにより、オペレーションの結果に関係なく、選択したレスポンスを強制的 に指定できます。

[ブレークポイントとレスポンスオーバーライド]ペインでは、ブレークポイントとレスポンスオーバーライドが 表示され、該当の実行について各要素の削除、有効化、または無効化が可能です。

### [トランジション履歴]ペイン

	ŀЭ	ンジジョン履歴	*
R	0	[13:40:34.543] WMIQuery: <空のトランジションの説明>	
K	9	[13:40:34.582] リターンステップ:Get Task List	
K	9	[13:40:38.133] Get Task List: task list retrieved	
K	Ð	[13:40:52.048] Analyze Table Columns: outliers were found in the data	
K	9	[13:40:53.265] Display?: <空のトランジションの説明>	
K	9	[13:40:54.586] Email?: <空のトランジションの説明>	-

[トランジション履歴]ペインでは、実行内で発生したトランジションとその説明が一覧表示されます。

## [ステップ結果 インスペクター] ペイン

ステップ結果インスペクター	*
{response=success;FailureMessage=;Result=\${in1};TimedOut=false;returnF	Resi
	$\mathbf{F}$
未加工結果「プライマリ結果」その他の結果」	

[ステップ結果 インスペクター] ペインには、選択したステップの結果が表示されます。

- 未加工結果 (ステップのオペレーションの結果)を表示するには、[未加工結果] タブをクリックします。
- ステップのプライマリ結果を表示するには、[プライマリ結果]タブをクリックします。
- 作成した結果がほかにもある場合、そうした結果を表示するには、[その他の結果]をクリックします。

## [コンテキストインスペクター] ペイン

コンテキストインスペクター	\$
🗆 ステップ入力	*
packetCount	
packetSize	
targetHost	localhost
□ ステップコンテキスト	
host	localhost
notifyMethod	None
output	C:¥HP¥OO¥studio>ping localhost
targetHost	localhost
vitalProcesses	svchost.exe,lsass.exe,smss.exe,Syst
ログローバルコンテキスト	
CENTRAL_URL	
12 11 121	<b>_</b>

[コンテキストインスペクター] ペインには、各ステップについてフロー変数 (グローバルおよびローカル)の 現在の値が表示されます。

グローバル変数、フロー変数、ステップの入力でのこれらの変数の値、ステップおよびグローバルコンテキストを確認するには、[コンテキストインスペクター]の該当セクションに移動します。

• [ステップ入力] セクションの値は、ステップ開始前に入力に割り当てられた値です。フロー変数の 値を含むテキストボックスは、色分けして表示されます。

コンテキストインスペクター		
₩ 2↓   ₽\$ ₽\$   ¶		
日 ステップ入力		
host 10.51.0.36,10.51.0.37		
□ ステップコンテキスト		
RS_Previous_Response success		
RS_Previous_Transition	success	
RS_Previous_Transitio	Failure email succeeded	
execution_thread_id	debug-runpool-1-thread-1	

• [ステップコンテキスト] セクションの値は、ステップ開始後に更新された値です。

ステップのコンテキストは、ステップのフローと親フローのローカルコンテキストに含まれるフロー変数と 値の割り当ての集合です(フローが別のフローのステップである場合、2つのフローの関係はサブフ ローと親フローになります)。

• [グローバルコンテキスト] セクションの値には、システムプロパティおよび作成されたすべてのグローバ

ル変数の値が表示されます。

コンテキストインスペクター *				
闇 2↓   ₽\$ 만\$   ¶				
rs.ws.session@https://jtho	iconclude-5002317708056339283			
user	admin			
wqlQuery	Select SystemUpTime from Win			
ロ グローバルコンテキスト				
RS_Previous_Response	success			
RS_Previous_Transition	success			
RS_Previous_Transition_An	localhost successfully pinged.			
execution_thread_id	runpool-1-thread-3			
execution_userid	admin			
host	localhost			
namespace	root\cimv2			
notifyMethod	Display			
output	C:\Program Files\Hewlett-Packa			

## デバッガーのツールバー



GUI ፖイ <del>テ</del> ム	説明	キーボードショート カット
再生 📂	フローを最後まで実行します。	F11
一時停止	デバッガーで実行中のフローを一時停 止します。 再生 ボタンをクリックする と、一時停止された時点からフローの 実行が再開されます。	ALT + P
停止	デバッガーで実行中のフローを停止しま す。	ALT + C
ステップオーバー 📩	フローをステップごとに実行します。	F5
ステップイン ᡖ	ステップのサブフローにステップインしま す。	F6
ステップアウト 🚹	ステップのサブフローからステップアウトし ます。	F7

リセット 祋	フロー変数の値をデバッガーを開いたと きの値にリセットします。	F12

# 複雑なフローのデバッグ

#### 並列処理ステップを使用したフローのデバッグ

Studio では、並列処理を使用したステップ、複数のインスタンス、並列分岐ステップをデバッグします。並列処理を使用するステップを持つフローが実行時にどのように動作するかについて知るには、Studio デバッガーでフローをテストした後にステージング環境でフローを実行する以外の方法はありません。

並列分岐ステップ、複数のインスタンスステップを含むフローは、こうしたステップを含まないフローをデ バッグするのと同じ方法でデバッグします。ただし、デバッガーで実行すると異なる動作となることを考 慮する必要があります。

## 実行する操作

#### フローの並列分岐ステップのデバッグ

フロー実行では、デバッガーは並列分岐ステップをすべて同時に開始し、終了の順序は、Studioで 予測不能な変動要因によって決まります。そのため、デバッガーは、同一のフロー変数への書き込み が競合した場合に、どのレーンが最後にフロー変数に書き込むかなどの考慮事項を予測できません。

ー方、Studio では、制御された方法でさまざまなシナリオをテストするために、デバッガーでのレーンの 終了順序を変更できます。

並列分岐ステップの詳細については、「並列分岐ステップのフローの作成」(186ページ)を参照してく ださい。

### フロー内の複数のインスタンスステップのデバッグ

フロー実行では、複数のインスタンスが並列で実行され、フローは、ほかのインスタンスの処理中に、1つのインスタンスのレスポンスに続くステップを続行します。

つまり、実際の条件下でテストが行われないということですが、各インスタンスが終了するまでにかかる 時間を調べることができます。

複数のインスタンスステップの詳細については、「複数のインスタンスステップのフローの作成」(190ページ)を参照してください。

# Central の Studio でのデバッグ

Central リモート デバッグ機能を使用して、Studio の作成者が、Central または外部 RAS およびデー タベースとして構成された Central でフローを実行することができます。

Studio では、ユーザーがリモートまたはローカルの Central で接続 およびデバッグできるよう設定する必要があるプロパティのセットが公開されています。

Studio では、HTTP とHTTPS の2種類の接続プロトコルをサポートしています。また、認証もサポートしています。認証が有効の場合、実行するすべてのフローで権限のあるユーザーであることを確認してください。

# Studio 構成ファイル

Studio では、[STUDIO\_HOME]\conf\studio.properties にある Studio 構成 ファイルを使用して、Central 接続設定を公開します。

studio.properties ファイルには、次のプロパティが格納されています。

- engine.connection.protocol: HTTP および HTTPS プロトコル
- engine.connection.host: Central ホスト名または IP
- engine.connection.port: Central のポート
- engine.connection.authenticated: Central が認証を使用しているかどうか (False または True)

# 実行する操作

注: Central で、Studio のフローをデバッグするには、フローが使用するコンテンツ (ベースコンテンツ パックなど)をすべて Central にデプロイする必要があります。

### 認証 HTTP 接続を使用した Central でのデバッグ

注: 実行されるフローは、選択したユーザーの権限でデプロイする必要があります。

- 1. STUDIO\_HOME/conf/studio.properties ファイルを開きます。
- 2. 次のように接続プロパティを変更します。
  - engine.connection.protocol: http
  - engine.connection.host: localhost
  - engine.connection.port: 8080
  - engine.connection.authenticated: true
- 3. Central を開きます。
- 4. 内部ユーザーを作成するか、セキュアユーザー向けにLDAPを構成します。Central でのLDAP の構成の詳細については、『Central ユーザーガイド』の「セキュリティのセットアップ – LDAP 認証」 を参照してください。

5. Central でセキュリティ設定を有効にします (Central で [認証を有効にする] を選択しま す。Central でのLDAP の構成の詳細については、『Central ユーザーガイド』の「セキュリティの セットアップ – LDAP 認証」を参照)。

次の手順で、Centralでフローを構築しデバッグします。

- 1. インポートしたコンテンツパックまたは既存のコンテンツパックからのオペレーションとフローを使用してフローを構築します。
- 2. フローを実行してデバッグします。

Operations Orchestration	Studio	
ファイル 編集 ツール <u>S</u> CM	ウィンドウ・ヘルプ・	
プロジェクト	ㅁ 。 ₽	ようこそ 🗙 🟈 allInOne 🚬 📜 🗕 🗙
-+ 🤌 🌒 👘 🗙 🥹		🚽 🗎 🖪 🐁 🌘 🗶
<ul> <li>→ MyProject 1</li> <li>→ → Library</li> <li>→ → My Ops Flows</li> <li>→ → fol 1</li> <li>→ → Configuration</li> </ul>		

3. ポップアップウィンドウが表示され、Central 認証用のユーザー名とパスワードを入力するよう促されます。

	Lane 1
	エンジン接続 "localhost:8080" の認証設定 ×
Success UUID Generator	エンジン接続資格情報 ユーザー名 user パスワード
	•••••
	保存 キャンセル

- 4. Central で認証する認証ユーザーのユーザー名とパスワードを入力し、[保存]をクリックします。
- 5. フローを実行します。フローが実行し正常に完了します。

# コンテンツパックのエクスポート

プロジェクトの作成 コンテンツバックの フローの作成 フローのデバッグ コンテンツのリリース

フローの検査が終了したら、フローをデプロイして実行するよう、コンテンツパックにリリースできるようになります。

コンテンツパックは、プロジェクトの成果です。これには、プロジェクトのエンティティと参照 ID が含まれます。コンテンツパックには、フローやオペレーションだけではなく、操作や構成アイテムも含まれています。

コンテンツパックは、デプロイメントのためにリリースする要素です。

注: 無効なフローやオペレーションは、コンテンツパックには含められません。

プロジェクトからコンテンツパックを作成すると、このコンテンツパックには、プロジェクトと同じ固有識別 番号 (UUID)が付けられます。つまり、プロジェクトから次のバージョンのコンテンツパックを作成でき、そ れをデプロイすると、同じUUIDを持つ古いバージョンを新しいバージョンで上書きすることができます。 古いバージョンを上書きしない新しいコンテンツパックを作成する場合は、新しいUUIDを持つプロ ジェクトのコピーを作成してそのコピーからコンテンツパックを作成します。

# 推奨されるベストプラクティス

- コンテンツパックの新しいバージョンを作成する場合、前のバージョンのコンテンツパックと区別するために更新されたバージョン番号を付けます。
- 新しいバージョンのコンテンツパックを Studio にインポート する場合、Studio で前のバージョンを閉じるか削除して、重複に関する問題が生じないようにすることをお勧めします。
- ・複数のコンテンツパックを作成している場合、同じ名前の構成アイテムが含まれていないことを確認してください。どのような場合でも、構成アイテムの名前を重複させないことをお勧めします。共有構成アイテムごとに別の共有プロジェクトにすることをお勧めします。
- あるコンテンツパックから別のコンテンツパックにコンテンツを移動した場合、フロー1とフロー2を含むコンテンツパックAをデプロイすると、CentralではコンテンツパックAに割り当てられたフロー1、フロー2が含められます。次に、フロー1を含むコンテンツパックBをデプロイすると、Centralでは、コンテンツパックAに割り当てられたフロー2と、コンテンツパックBに割り当てられたフロー1が含められます。

# 実行する操作

### コンテンツパックの作成

- 1. [プロジェクト] ペインで、コンテンツパックを作成するプロジェクトを選択します。
- 2. [ファイル] > [新しいコンテンツパックの作成]を選択します。

注:または、[プロジェクト] ペインの [コンテンツパックの作 成] <sup>(3)</sup> ボタンを選択しても構いません。

- 3. [コンテンツパックの作成]ダイアログボックスに、コンテンツパックの詳細を次のように入力します。
  - 場所 コンテンツパックを保存する場所を入力または参照します。 デフォルトで、プロジェクトの ワークスペースへのパスが選択されています。
  - バージョン 同じコンテンツパックを複数繰り返して作成する場合、それぞれにバージョン番号 を付けることをお勧めします。
  - パブリッシャー
  - 説明 (オプション)

[名前] フィールドに、プロジェクトの新しい名前が表示されます。これは読み取り専用です。

4. [OK] をクリックします。新しいコンテンツパックが、指定した場所に作成されます。

このコンテンツパックは、デプロイ、実行、別のプロジェクトへのインポートができます。

# 参照情報

[コンテンツパックの作成] ダイアログボックス

GUI アイテム	説明
名前	コンテンツパックの名前は、プロジェクト名から取得されます。この フィールドは読み取り専用です。
場所	コンテンツパックを保存する場所を入力または参照します。 デフォ ルトで、 プロジェクトのワークスペースへのパスが選択されていま す。
バージョン	コンテンツパックのバージョンを入力します。 この情報は、コンテンツ パックの [プロパティ] シートに表示されます。
パブリッシャー	コンテンツパックのパブリッシャーを入力します。 この情報は、コンテンツパックの [プロパティ] シートに表示されます。
説明	コンテンツパックの説明を入力します。この情報は、コンテンツ パックの [プロパティ] シートに表示されます。

# フローとオペレーションの管理

プロジェクトのライブラリに多数のフローとオペレーションを格納できます。この章では、アイテムを検索、 コピーする方法、アイテムをブックマークする方法、アイテムの使用状況を確認する方法、および新し いオペレーションを作成する方法といった、このライブラリを管理する方法について説明します。

オペレーションの作成	225
フローまたはオペレーションの検索	232
フローおよびオペレーションのコピー	.237
フローとオペレーションの使用方法の詳細	240
フローおよびオペレーションに関 するド キュメントの生 成	.242
フローとオペレーションのバージョン履歴管理	.247
フローおよびオペレーションのブックマーク	249

# オペレーションの作成

Studio でオペレーションを作成する方法には次の3つがあります。

- 既存オペレーションをコピーおよび編集する。
- オペレーションを既存のプラグインからインポートする。
- Java でアクションプラグインを作成し、そのアクションプラグインを Studio にインポートする。

# アクションプラグインからのオペレーションの作成

アクションプラグインは、IActions または @Actions を含む jar ファイルです。 アクションプラグインを Studio にインポートし、 アクションプラグインのいずれかの操作 からオペレーションを作成 することができま す。

アクションプラグインは、複数の操作を含めることができ、それぞれの操作から1つずつオペレーションを 作成することができます。

アクションプラグインの開発の詳細については、『拡張開発者ガイド』を参照してください。

## アクションプラグインから作成されたオペレーションのコピー

アクションプラグインをインポートして作成したオペレーションをコピーした場合、コピーされたオペレーションは元のオペレーションの参照を保持したままです。アクションプラグインがアップグレードされた場合、元のオペレーションを、新しいバージョンを呼び出すよう更新すると、コピーされたオペレーションもすべて自動的に更新されます。これを「ソフトコピー」といいます。

ソースオペレーション (このオペレーションのコピー元)は、[詳細] タブに表示されます。

コピーされたオペレーションで自動的に更新されるアイテムは、アクションプラグインへのリンクのみです。 元のオペレーションの入力、出力、変数、スクリプトレットなどは、コピーでは更新されません。

# 有効なオペレーション

有効なオペレーションには次の条件があります。

- UUID ジェネレーターなどの入力を必要としないオペレーションがあること。
- オペレーション結果を表す有効な式へマッピングされるレスポンスが少なくとも1つあること。

新規オペレーションが無効または未完成の場合は、オペレーションの名前が[**プロジェクト**] ペインに波線の下線付きで表示されます。未完成のオペレーション名の上にカーソルを合わせると、未完成の状態を示すツールチップが表示されます。

# ベストプラクティス

作成したオペレーションを使用してフローを作成するほかの作成者にそのオペレーションをわかりやすくするには、オペレーションの[説明]タブに次の情報を追加します。

- オペレーションの内容の説明。
- オペレーションが必要とする入力。入力に必要なデータや適切なデータ形式を作成者が検索できる場所も含む。
- レスポンス。各レスポンスの意味を含む。
- 結果フィールド。各結果フィールドで与えられるデータの説明を含む。
- 次のような、実装に関する追加のメモすべて。
  - プラットフォームまたはアプリケーションのサポート。バージョン情報を含む。
  - フローが相互作用するアプリケーションまたは Web サービス API。
  - その他の環境要件または使用要件

## 実行する操作

#### Studio でのオペレーションのコピーと編集

- 1. [コンテンツパック] ペインまたは [プロジェクト] ペインで、コピーするオペレーションを選択します。
- 2. [編集] > [コピー]を選択します。
- 3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集] > [貼り付け]を選択します。オペレーションは、新しいオブジェクトとして扱われます。

- 4. オペレーションを検索用途のためにカテゴリに割り当てるには、[カテゴリの割り当て]をクリックし、リストからカテゴリを選択します。
- 5. 入力を作成するには、[入力]タブをクリックし、[入力の追加]をクリックします。
- 6. 表示されたダイアログボックスに名前を入力し、[OK]をクリックします。
  - 入力の内容と使用方法の詳細については、「入力の作成」(118ページ)を参照してください。
  - 入力用のデータソースの定義の詳細については、「入力ソースの指定」(125ページ)を参照してください。
- 7. 出力データを追加し、定義します。

出力データの追加と使用の詳細については、「オペレーションの出力の設定」(147ページ)を参照してください。

8. 必要なレスポンスを作成し、結果をそのレスポンスにマッピングします。

オペレーションにどのレスポンスが選択されるのかを定めるルールの定義については、「レスポンスの設定」(137ページ)を参照してください。

- 9. [説明] タブをクリックして、テキストボックスに説明を入力します。
- 10. [OK]をクリックします。

#### オペレーションがベースとするソースオペレーションの変更

既存のオペレーションをコピーしてオペレーションを作成した場合、ソースオペレーションを変更できます。コピーしたオペレーションは、ソースとして選択したオペレーションのコピーになります。

- 1. コピーしたオペレーションを開き、[プロパティ]を選択します。
- 2. フォルダーの[プロパティ]シートで、[詳細]タブをクリックします。
- 3. [ソースオペレーション] で、[選択] ボタンをクリックします。
- 4. [オペレーションの選択] ダイアログボックスで、コピー元のソースオペレーションに移動して選択し、 [OK] をクリックします。
- 5. 必要に応じて、変更内容に合わせてオペレーションの名前を変更します。
- 6. 入力の値の割り当てに必要な変更を確認して実行し、以前のオペレーションの入力と新しいオペレーションの入力の違いが反映されるようにします。

#### アクションプラグインのインポートによるオペレーションの作成

アクションプラグインからオペレーションを作成する最も直接的な方法は、アクションプラグインのインポートと同時にオペレーションを作成する方法です。

- 1. アクションプラグインを作成してパックし、開発したアクションを含めます。アクションプラグインの開発方法の詳細については、『拡張開発者ガイド』を参照してください。
- 2. Studio で新しいオペレーションの作成先のフォルダーを右クリックして [プラグインのインポート]を選択します。
- 3. [プラグインのインポート] ダイアログボックスで、インポートする HP OO プラグインを参照して選択します。

🕵 ブラグインのインボート		×
* プラグインの場所:		
☑ 新規オペレーションのインポートと作成		
	OK	キャンセル

- 4. [新規オペレーションのインポートと作成] チェックボックスをオンにして [OK] をクリックします。
- 5. [オペレーションの作成]ダイアログボックスで、必要なアクションを含むプラグインを展開し、オペレーションの作成元として使用するアクションを選択します。

🕵オペレーションの作成	×
インボートするオペレーションの選択:	
🗈 🗆 comhp.oc:oo-dotnet-web-actions-sccm-legacy-plugin:1.0.113	
🗄 🗆 🗖 com hp.po:po-dotnet-web-actions-scom-legacy-plugin:1.0.113	
🗄 🗆 🗖 com hp.oo:oo-dotnet-web-actions-scvmm-legacy-plugin:1.0.111	
🗄 🗆 🗖 com hp.po:oo-dotnet-web-actions-xen-app-legacy-plugin:1.0.113	
🗄 🗆 🗖 com hp.oo:oo-ec2-legacy-plugin:1.0.116	
🗄 🗂 🗖 com hp.oc:oo-ec2-plugin:1.0.116	
🗄 🗹 com hp.oc:oo-excel-legacy-plugin: 1.0.121	
🗄 🗆 🗖 comhp.co:co-fs-plugin:1.0.121	
🗄 🗆 🗖 com hp.oo:oo-groovy-legacy-plugin: 1.0.121	
🗄 🗆 🗖 com hpilocio-hp-cda-legacy-plugin: 1.0.117	
🗄 🗆 🗖 com hp.oo:oo-hp-om-i-legacy-plugin:1.0.117	
🗄 🗆 🗖 com hp.oo:oo-hp-om-legacy-plugin:1.0.117	-
リー・ーー アクションの共通プレフィックス "com" を次の値に置換:	
com	
	OK キャンセル

**注**: プラグインに複数のアクションが含まれている場合、複数のアクションを選択して複数の オペレーションを作成することができます。

選択したそれぞれのアクションに対する新しいオペレーションが、右クリックしたフォルダーに作成されます。

それぞれのオペレーションで、アクションプラグインについての情報が、オペレーションの[プロパティ] ウィンドウの[**入力**] タブの上部の[**Operation フィールド**] セクションに表示されます。

Operation フィールド ――	
グループ ID:	comhp.oo
アーティファクト ID:	oo-excel-legacy-plugin
バージョン	1.0.121
アクション名	com.iconclude.content.actions.excel.ops.ModifyExcelData

6. オペレーションを保存します。

### インポートしたアクションプラグインからのオペレーションの作成

アクションプラグインを Studio のリポジトリにインポートしたら、その中 のアクションからオペレーションを作成 できます。

- 1. Studio で新しいオペレーションの作成先のフォルダーを右クリックして [新規] > [オペレーション] を 選択します。
- 2. Studio のリポジトリからプラグインの場所を参照し、[OK] をクリックします。
- 3. [オペレーションの作成]ダイアログボックスで、オペレーションの作成元のアクションを選択します。

🕵オペレーションの作成	×
インポートするオペレーションの選択	
🗄 🗆 🗖 com hp.po:oo-dotnet-web-actions-sccm-legacy-plugin:1.0.113	
😟 🗖 com hp.oc:oo-dotnet-web-actions-scom-legacy-plugin:1.0.113	
😟 🗆 🗖 com hp.oc:oo-dotnet-web-actions-scvmm-legacy-plugin:1.0.111	
😟 🗖 com hp.oo:oo-dotnet-web-actions-xen-app-legacy-plugin:1.0.113	
😟 🗁 🗔 comhploc:oo-ec2-legacy-plugin:1.0.116	
🗄 🖳 com hp.oc:oo-ec2-plugin: 1.0.116	
😟 🗹 comhploc:oo-excel-legacy-plugin:1.0.121	
🗄 🗁 🗔 com hp.oc:oo-fs-plugin: 1.0.121	
👾 🗖 comhplocio-groovy-legacy-plugin:1.0.121	
👾 🗖 comhplocio-hp-cda-legacy-plugin:1.0.117	
🗄 🖷 🗖 com hp.oc:oo-hp-om-i-legacy-plugin: 1.0.117	
🗄 🗠 🔲 com hplocico-hp-om-legacy-plugin: 1.0.117	<b>T</b>
リー・ーー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	
com	
	OK キャンセル

**注**: プラグインに複数のアクションが含まれている場合、複数のアクションを選択して複数の オペレーションを作成することができます。

選択したそれぞれのアクションに対する新しいオペレーションが、右クリックしたフォルダーに作成されます。

それぞれのオペレーションで、アクションプラグインについての情報が、オペレーションの[プロパティ] ウィンドウの[**入力**] タブの上部に表示されます。

### アクションプラグインのインポート

Studio のリポジトリにアクションプラグインをインポートだけし、後から自分またはほかの作成者がその アクションプラグインからオペレーションを作成することも可能です。

- 1. Studio で、[**ファイル**] > [**プラグインのインポート**]を選択します。
- 2. [プラグインのインポート] ダイアログボックスで、Studio のローカルの Maven リポジトリにインポート する HP OO プラグインを参照して選択します。
- 3. [OK] をクリックします。 作成者はこのプラグインからオペレーションを作成できます。 詳細については、「インポートしたアクションプラグインからのオペレーションの作成」を参照してください。

### 手動オペレーションの作成

手動オペレーションは、操作を選択できるオペレーションです。 ユーザーが、実行時に操作を選択する 必要があります。

手動オペレーションを作成するには、手動オペレーションテンプレートをベースコンテンツからコピーして、 ユーザーが選択できる操作を定義します。

- 1. [コンテンツパック] ペインで、コンテンツパック内にある手動オペレーションテンプレートを選択します (base-cp/Library/Utility Operations/Manual など)。
- 2. [編集] > [コピー]を選択します。
- 3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集] > [貼り付け]を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。
- 4. オペレーションのプロパティで、ユーザーが選択できる操作を追加します。

### 表示オペレーションの作成

表示オペレーションとは、ポップアップのプロンプトメッセージに情報を表示しますが、ほかの操作が実行されないオペレーションです。ユーザーは、実行時に[続行]を選択する必要があります。

表示オペレーションを作成するには、ベースコンテンツにある表示オペレーションテンプレート Library/Utility Operations/Display Messageをコピーして、ユーザーに表示する情報を定義 します。

プロンプトメッセージに変数を含めることができます。たとえば、前のステップが終了した時間をユーザー に通知するよう、メッセージに日付/時刻の変数 (\${dateTime})を含めることができます。

- 1. [コンテンツパック] ペインで、表示オペレーションテンプレートを選択します。
- 2. [編集] > [コピー]を選択します。
- 3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集] > [貼り付け]を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。

- 4. オペレーションのプロパティシートで、[詳細]タブを選択します。
- 5. ステップのインスペクターで [表示] タブをクリックします。
- 6. [このステップの実行前に常にユーザーに確認する] チェックボックスをオンにします。
- 7. [プロンプトタイトル] ボックスに、プロンプトのラベルを入力します。
- 8. [プロンプトの幅] ボックスに、プロンプトの幅をピクセル単位で入力します。
- 9. [高さ] ボックスに、プロンプトの高さをピクセル単位で入力します。
- 10. [プロンプトテキスト] ボックスに、ユーザーへのメッセージを入力します。
- 11. [OK]をクリックし、変更を保存します。

## 参照情報

### ステップのインスペクター > [表示] タブ

[オペレーションのプロパティ] シートの [**表 示**] タブで、ユーザーに表 示 するユーザープロンプトを作 成 でき ます。

ステップ名: SQL Query	
入力 結果 表示 説明 詳細 スクリプトレット	
🔲 このステップの実行前に常にユーザーに確認する	
プロンプトタイトル	
プロンプトの幅: 0	高さ 0
プロンプトテキスト	
1	

GUI ፖイテム	説明
このステップの実行前に常に ユーザーに確認する	ステップを実行するたびにプロンプトウィンドウを表示するには、 チェックボックスをオンにします。
プロンプトタイトル	プロンプトウィンドウのタイトルバーに表示するラベルを入力します。

プロンプトの幅	プロンプトウィンドウの幅をピクセル単位で入力します。
高さ	プロンプトウィンドウの高さをピクセル単位で入力します。
プロンプトテキスト	プロンプトウィンドウの本文に表示するメッセージを入力します。メッ セージに変数を含めることができます。たとえば、\${dateTime}のようになります。

### プラグインのインポート

🕵 ブラグインのインボート		×
* プラグインの場所:		
☑ 新規オペレーションのインボートと作成		
	ОК	キャンセル

GUI アイテム	説明
プラグインの場所	インポートする HP OO プラグインを参照して選択します。
新規オペレーションのインポート と作成	このチェックボックスを選択して、インポートしたプラグインから新 しいオペレーションを作成します。

## オペレーションのプロパティ: [詳細] タブ

ソースオペレーション		
リンク先:  Get Performance Counter Value [Operation]		
場所:  /Base [1.0.121]/Library/Operations/Operating Systems/Windows/Get Performance C	厭	接続解除

GUI ፖイテム	説明
リンク先	ソースオペレーション (選択したオペレーションのコピー元)が表示されます。
場所	ソースオペレーションの名前が表示されます。
開く	ソースオペレーションの [プロパティ] シート が開きます。
接続解除	オペレーションを親 プラグインからデタッチします。

# フローまたはオペレーションの検索

目的のフローまたはオペレーションを見つけるにはいくつかの方法があります。

- [プロジェクト] ペインおよび [コンテンツパック] ペインでフォルダーを参照する
- フォルダー、フロー、オペレーションの説明を表示する

検索を実行する

## 実行する操作

#### フォルダーを参照してフローまたはオペレーションを見つける

フローやオペレーションを探す最も簡単な方法は、フォルダーを参照することです。

フォルダーに適切な名前が付けられ、適切な構造になっている場合は、目的のフローやオペレーションが探しやすくなります。

#### 説明を使用してフローまたはオペレーションを見つける

オペレーションやフローの説明を表示して、目的のものかどうかを確認できます。

- オペレーションの説明を表示するには、作成ペインでオペレーションを開いて[説明]タブをクリックします。
- フローの説明を表示するには、作成ペインでフローを開いて [プロパティ] (ペインの下部)をクリックし、[説明] タブをクリックします。

**注**: または、[プロジェクト] ペインまたは [コンテンツパック] ペインで、フローまたはオペレーションを 右 クリックして [プロパティ] を選択します。

#### ドキュメントを生成してフローまたはオペレーションを見つける

ドキュメントの生成機能では、複数のフローやオペレーションについてのこれらの情報を1つにまとめる ことができます。ドキュメントの生成機能の詳細については、「フローおよびオペレーションに関するド キュメントの生成」(242ページ)を参照してください。

#### オペレーションの検索

[検索] ペインを使用して、ライブラリ全体をフルテキスト検索できます。 名前 やそれ以外のフィールドプ ロパティを検索することにより、フローやオペレーションを検索できます。

- 1. Studio ウィンドウの下部にある[検索]タブをクリックして、[検索]ペインを開きます。
- 2. [検索]リストで、検索対象のフィールドを選択します。

注:検索にすべてのフィールドを含めるには、[検索]リストを[<すべてのフィールド>]に設定したままにします。

3. [対象] テキストボックスに、検索するテキストを入力します。

検索文字列に空白が含まれている場合、完全一致検索または空白を含む検索のいずれかを 指定できます。完全一致検索では、文字列全体を、空白がある場合はそれも含めて、1つの 検索値として扱います。空白を挿入すると、空白で区切られた文字列すべてを検索します。 注:入力した検索文字列の検索は、大文字と小文字を区別しません。

- 4. 検索での空白の扱い方を定義するには、次のようにします。
  - 空白を1つの検索文字列の一部として扱うよう指定するには、[完全] チェックボックスを選択します。
  - 空白を、検索文字列を順に区切るセパレーターとして扱うには、[完全] チェックボックスを選 択解除します。
- 5. 検索ボタン 🔍 をクリックします。

検索					
検索:	<すべてのフィールド>	▼ 対象		☑ 完全	0,
該当項目がありません					
ランク マ	名前		種類	R	र

6. [説明] タブのテキストを確認し、目的のオペレーションを指定します。

検索結果には、オペレーションの[説明]タブから説明が取得され、オペレーションを最大限に活用するための、次のような重要な情報が表示されます。

- オペレーションの入力が必要な種類の情報。
- 結果に含まれる情報。
- オペレーションの要件と前提条件。

説明全体を確認する場合は、オペレーションの[プロパティ]シートの[説明]タブを参照してください。

#### 前回の検索コマンドを使用した検索

1. [検索]ペインで、[検索履歴]ボタンをクリックします。

[以前に入力した検索文字列] ウィンドウが開きます。

以前に入力した検索文字列:	
<すべてのフィールド> 対象 solaris <すべてのフィールド> 対象 形式 <すべてのフィールド> 対象 SQL	

このウィンドウには、以前に使用した検索コマンドのリストが最大 25 個表示されます。

リストの検索コマンドの形式は、「対象<検索テキスト>」です。

2. リストから検索コマンドをダブルクリックして実行します。そのコマンドは、リストの最上位に追加されます。

### 検索結果の並べ替え

いずれかの列の列見出しをクリックすると、そのパラメーターを基に検索結果が並べ替えられます。

### Lucene 構文を使用した検索

より詳細な検索結果を得たい場合、Apache Lucene構文を使用して検索を行うことができま す。Lucene検索構文については、Apache Software FoundationのWebサイトを参照してください。

- 1. Studio ウィンドウの下部にある[検索]タブをクリックして、[検索]ペインを開きます。
- 2. [検索]ドロップダウンリストで、[<Lucene クエリ>]を選択します。
- 3. [対象] テキストボックスに、Lucene 検索構文を使用してクエリを入力し、検索ボタン 🤽 をクリックします。

最も簡単な Lucene 検索構文は次のようなものです。

<検索可能なフィールド名>:<検索対象の文字列>

検索のヒント

- この検索はブールANDを使用します。ANDを用いて検索語を2つ入力すると、両方の検索語が含まれるオペレーションまたはフローのみが検索結果として戻されます。ANDを使わずに2つの検索語を入力すると、どちらかの検索語を含む検索結果が得られます。
- category: database server の検索のように、スペースを含む文字列のみを検索結果として得たい場合は、category: "database server"のように文字列を引用符で囲みます。

検索可能なフィールド名は以下のとおりです。なお、ここで使用している検索文字列は一例です。

■ フローまたはオペレーションの名前

例:

name:Get Temp Dir
name:Clear Temp Dir

オペレーションのタイプ

例:

type:cmd

■ カテゴリ

例:

category:network

■ 入力の名前

例:

inputs:server

■ フローまたはオペレーション UUID

例:

id:1234-3453-3242-32423

■ フローまたはオペレーションの説明に含まれている文字列

例:

description:clear

#### [検索] ペインからのオペレーションへのアクセス

検索結果から直接、オペレーションやフローを操作することができます。オペレーションやフローを開いて 編集したり、作成ペインで開いているフローに追加することができます。

- 検索結果でオペレーションの行をダブルクリックして、オペレーションの[プロパティ]シートまたはフロー図を開きます。
- [検索] ペインからフロー図にオペレーションをドラッグし、検索結果のオペレーションからステップを作成します。

# 参照情報

[検索] ペイン

検索	
検索:	▼ 対象 🔽 🔽 🔍
該当項目がありません	
_ ランク マ 名前	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
GUI アイテム	説明
検索	<ul> <li>1つのフィールドで検索を実行する場合は、検索を実行する フィールドを選択します。</li> </ul>
	<ul> <li>検索にすべてのフィールドを含めるには、[&lt;すべてのフィール ド&gt;]を選択します。</li> </ul>
	<ul> <li>Lucene クエリを使用して検索する場合は、[<lucene クエリ="">]を選択します。</lucene></li> </ul>
対象	検索する文字列を入力します。
完全	<ul> <li>空白を1つの検索文字列の一部として扱うよう指定するには、[完全]チェックボックスを選択する</li> </ul>
	<ul> <li>空白を、検索文字列を順に区切るセパレーターとして扱うには、[完全]チェックボックスを選択解除します。</li> </ul>
ランク	各検索結果のランクが表示されます。星が多いほどランクが高く なります。
種類	フローなどの見つかったアイテムの種類が表示されます。
パス	アイテムの保存先が表示されます。
説明	アイテムの[ <b>説明</b> ]タブから取得されたアイテムの説明が表示され ます。
検索履歴	クリックすると、[以前に入力した検索文字列] ウィンドウが表示され、検索コマンドを再使用できます。

# フローおよびオペレーションのコピー

フローやオペレーションをコピーする方法は3つあります。

フローやオペレーションをコピーする場合は、封印されていないいずれかのフォルダーにコピーを貼り付けることができます。フローをコピーした場合、フローのみがコピーされ、フローを構成するオペレーションはコピーされません。

- [ディープコピー]をクリックすると、フローだけでなくそのフローを構成しているすべてのオペレーションも コピーされます。これは、新しいフローでオペレーションを変更する必要があり、かつ元のオペレーショ ンに影響を与えたくない場合に使用します。
- フローやオペレーションを複製する場合、複製は自動的に元のオペレーションと同じフォルダーに配置され、[<名前> (Copy 1)] という名前になります。
- フローやオペレーションを切り取ると、そのフローやオペレーションは現在の場所から削除され、別の場所に貼り付けられます。

## アクションプラグインから作成されたオペレーションのコピー

ソフトコピー

アクションプラグインの jar ファイルにリンクされたオペレーションをコピーした場合、コピーされたオペレー ションは元のオペレーションへの参照を保持したままです。 アクションプラグインの jar ファイルがアップグ レードされた場合、元のオペレーションを、新しいバージョンを呼び出すよう更新すると、コピーされたオ ペレーションもすべて自動的に更新されます。 これを、**ソフトコピー**と呼びます。

コピーされたオペレーションで自動的に更新されるアイテムは、アクションプラグイン jar ファイルへのリンクのみです。 元のオペレーションの入力、出力、変数、スクリプトレットなどは、コピーでは更新されません。

コンテンツパックからオペレーションをコピーできますが、参照は親のコンテンツパックに保持されます。親 のプラグインが今後修正された場合、ソフトコピーにもその修正が入ります。場合によっては、オペレー ションへの修正を行わないようにすることもできます。この場合、オペレーションを親プラグインからデタッ チすることができます。デタッチされたオペレーションに対しては、手動で修正を実施する必要がありま す。

ソフトコピーには次のような長所と短所があります。

- 短所は、元のオペレーションが削除された場合に、コピーが親をなくし、プラグインへのリンクも損失してしまうことです。その場合、新しい親を手動で選択する必要があります。
- 長所は、元のオペレーションが別のバージョンのプラグインに更新された場合、ソフトコピーも更新されることです。

アクションプラグイン jar ファイルからのオペレーションの作成の詳細については、「オペレーションの作成」 (225ページ)を参照してください。

#### ハードコピー

10.00 以前のバージョンの HP OO では、アクションプラグインにリンクされたオペレーションをコピーする と、ハードコピーが作成されました。これは、コピーが、元のオペレーションと同様にアクションプラグイン に直接リンクしています。アクションが更新された場合 (たとえば JAR の名前またはクラスが変更され た場合)、ハードコピーされたオペレーションもすべて更新する必要がありました。

HP OO 10.00 では、新しいオペレーションを作成し関連プラグインを選択することで、ハードコピーを作成することができます。 この方法では、IAction getTemplate または @Action メタデータに基づいて新しいオペレーションが作成されます。 入力や出力の複製も行う、オペレーションのハードコピーを作成することはできません。

ハードコピーには次のような長所と短所があります。

- 長所は、元のオペレーションが削除された場合、コピーには影響せず、親を失うことがありません。
- 短所は、元のオペレーションが別のバージョンのプラグインに更新されても、ハードコピーは更新されないことです。

## ハードコピーとソフトコピーの切り替え

ソフトコピーを親からデタッチし、ハードコピーにすることができます。フローの[詳細]タブで、[接続解除] ボタンをクリックしてオペレーションをデタッチします。 確認 メッセージが表 示されます。 プラグインの GAV パラメーターが元 の親 から取 得されます。

[**選択**] ボタンをクリックし、親オペレーションを選択してハードコピーをソフトコピーに変換することができます。

## ハードコピー内のプラグインの置き換え

特定のプラグインを使用している、ハードコピーされたすべてのオペレーションを検索できます。また、そのオペレーションを選択し、プラグイン GAV パラメーターを変更することができます。

# ベストプラクティス

フローをコピーしている時に、オペレーションのプロパティの変更が必要になった場合は、[ディープコ ピー] コマンドを使用して、オペレーションと一緒にフローもコピーする方法が最適です。

[ディープコピー] コマンドを使用してフローをコピーする場合、フローとそのオペレーション用の新しいフォ ルダーを作成することをお勧めします。

## 実行する操作

#### フローまたはオペレーションのコピー

- 1. [コンテンツパック] ペインまたは [プロジェクト] ペインで、コピーするフローまたはオペレーションを右ク リックします。
- 2. [編集] > [コピー]を選択します。
- 3. コピーを保存するフォルダーに移動し、[編集] > [貼り付け]を選択します。

#### フローまたはオペレーションの複製

フローやオペレーションを複製する場合、複製は自動的に元のオペレーションと同じフォルダーに配置 され、[<名前> (Copy 1)] という名前になります。

- 1. [コンテンツパック] ペインまたは [プロジェクト] ペインで、コピーするフローまたはオペレーションを右ク リックします。
- 2. [編集]>[複製]を選択します。

### フローまたはオペレーションのディープコピー

フローをディープコピーした場合、フローのみではなく、フローを構成するオペレーションもコピーされます。

- 1. [コンテンツパック] ペインまたは [プロジェクト] ペインで、コピーするフローを右 クリックします。
- 2. [編集] > [ディープコピー]を選択します。
- 3. フローとそのオペレーションを保存するフォルダーに移動し、[編集] > [貼り付け]を選択します。

#### フローまたはオペレーションの切り取り

- 1. [コンテンツパック] ペインまたは [プロジェクト] ペインで、移動するフローまたはオペレーションを右ク リックします。
- 2. [編集] > [切り取り]を選択します。
- 3. フローやオペレーションを保存するフォルダーに移動し、[編集] > [貼り付け]を選択します。

# フローとオペレーションの使用方法の詳細

オペレーションやフローが既存のフローでどのように使用されているかを調べることにより、その使用方 法や実装方法について理解を深めることができます。これは、[参照] ペインでできます。

参照	
/MyProject1/Library/My Ops Flows/Network Check からの参照	
オブジェクト	パス
⊡-Network Check	/MyProject1/Library/MyOps Flows/Network Check
Operation: Local Ping [Operation]	/Base [1.0.121]/Library/Operations/Network/Local Ping
Operation: Local Traceroute [Operation]	/Base [1.0.121]/Library/Operations/Network/Local Traceroute
Operation: Manual [Operation]	/Base [1.0.121]/Library/Utility Operations/Manual

Studioには、次の2種類の参照があります。

- オペレーションまたはフローへの参照 選択したオペレーションまたはフローから作成されたステップのあるフローの一覧を示します。
- オペレーションまたはフローからの参照 選択したオペレーションまたはフローが使用するオブジェクト(選択リスト、グループに割り当てられた権限、システムフィルターなど)の一覧を示します。フローの場合は、フローのステップを作成したときのベースになったオペレーション(サブフローを含む)です。

# ベストプラクティス

オペレーションまたはフローを変更すると、それを使用しているほかのフローが壊れる場合があります。 フローやオペレーションを変更する前に、[参照] > [これを使用する項目は?]を使用して、ほかのフ ローでこのフローやオペレーションを使用していないかを確認します。

# 実行する操作

### フローまたはオペレーションを使用しているものを識別

- 1. [プロジェクト] ペインで、オペレーションまたはフローを右クリックします。
- 2. [参照] > [これを使用する項目は?]を選択します。

[参照]ペインが開き、オペレーションまたはフローの参照元が表示されます。

Operations Orche	stration Studio			
ファイル 編集 ツール	SCM ウィンドウ ヘルプ			
プロジェクト			084	4328 ×
+ 🔌 🌒 👘 🎗	4 P			<b>P B</b>
MyProject1     Dibrary     Dibrary	Flows 開く デバッグ 名前の変更 削除 編集	•		Get Statu for Runs failure
	参照	►	これを使用する功	頁目は?
	SCM	►	これが使用する巧	頁目は?
	プロパティ			
	ローカル履歴			
	エクスプローラーで表示			

#### フローまたはオペレーションが使用しているものを識別

- 1. [プロジェクト] ペインで、オペレーションまたはフローを右クリックします。
- 2. [参照] > [これが使用する項目は?]を選択します。

[参照] ペインが開き、オペレーションまたはフローの参照先が表示されます。

**ヒント:**参照元であるフローやオペレーションは、コピー、貼り付け、編集が可能なサンプルとして役立ちます。

## 参照情報

### [参照]ペイン

[参照] ペインでは、既存のフローでのオペレーションまたはフローの使用方法を参照できます。

参照	
/MyProject1/Library/My Ops Flows/Network Check からの参照	
オブジェクト	パス
⊡-Network Check	/MyProject1/Library/MyOps Flows/Network Check
Operation: Local Ping [Operation]	/Base [1.0.121]/Library/Operations/Network/Local Ping
Operation: Local Traceroute [Operation]	/Base [1.0.121]/Library/Operations/Network/Local Traceroute
Operation: Manual [Operation]	/Base [1.0.121]/Library/Utility Operations/Manual

GUI アイテム	説明
オブジェクト	選択したフローまたはオペレーションで使用されている、またはこれらが使用 するオブジェクトが表示されます。
パス	選択したフローまたはオペレーションで使用されている、またはこれらが使用 する場所が表示されます。

# フローおよびオペレーションに関するドキュメントの生成

**重要**:現在のバージョンでは、実行時のドキュメントの生成機能はサポートされていません。Studio内からドキュメントを生成できますが、ドキュメントの生成ステップを含むフローを作成しても、このステップは実行時には動作しません。

フローやオペレーションをドキュメントにし、ほかのユーザーに詳細情報を提供することができます。

- フローを PNG 画像としてエクスポート する。
- ドキュメントの生成機能を使用して、フローやオペレーションに関する情報を含む HTML ページを 作成する。

## 生成されたドキュメントの構造

ドキュメントを生成すると、index.html という HTML ページが作成されます。このページには、次のフレームがあります。

- [概要] フレーム 左上の[概要] フレームには、ドキュメントを生成したフォルダーに含まれるサブ フォルダーのリストが表示されます。[フォルダー] フレームに内容を表示するフォルダーを選択しま す。
- [フォルダー] フレーム 左下の [フォルダー] フレームには、[概要] フレームで選択したフォルダー内のフローやオペレーションのリストが表示されます。
- [ヘッダー] フレーム 右上の [ヘッダー] フレームには、HP OO バナーが表示されます。

• [オペレーション] フレーム – 右下の[オペレーション] フレームには、フローまたはオペレーションの説明 が表示されます。[プロパティ] シートの[説明] タブで入力した情報です。



# 実行する操作

### フローのPNG画像でのエクスポート

- 1. フローを作成ペインで開きます。
- 2. 作成ペインを右クリックし、[PNG にエクスポート]を選択します。
- 3. 画像の保存先を参照し、[保存]をクリックします。

#### 標準形式でのドキュメントの生成

特定のフローやオペレーションを含むフォルダーのドキュメントや、Library フォルダー全体のドキュメント を生成することができます。

- 1. ドキュメントを作成するフォルダーを右クリックします。
- 2. [ドキュメントの生成] > [標準形式] を選択します。
- 3. ドキュメントファイルの保存先を参照し、[保存]をクリックします。HTML ファイルの index.html が Web ブラウザーで開きます。
- 4. 旧バージョンの index.html を上書きする場合は、[すべてはい] をクリックします。

注: 旧バージョンのドキュメントを上書きしない場合は、[キャンセル]をクリックして手順を繰り返し、ファイルを別の場所に保存します。

#### カスタム形式のドキュメントの生成

- 1. ドキュメントを作成するフォルダーを右クリックします。
- 2. [ドキュメントの生成] > [カスタム形式]を選択します。
- 3. [ドキュメント生成テンプレートの選択]ダイアログボックスで、ドキュメントの生成に使用するテンプ レートを選択します。
- 4. ドキュメントファイルの保存先を参照し、[保存]をクリックします。HTML ファイルの index.html が Web ブラウザーで開きます。

#### カスタマイズしたドキュメントの生成テンプレートの作成

ドキュメントのテンプレートは Studio\template フォルダーに保存されます。 テンプレートの末尾には .vm が付いていて、 テキストエディターで編集できます。 テンプレートの詳細については、以降の「参照 情報」のセクションを参照してください。

Studio\template フォルダーに作成した新しい.vm ファイルが、[ドキュメント生成テンプレートの選択] ダイアログボックスのテンプレートリストに表示されます。



1. 必要な .vm テンプレートのコピーを作成し、そのコピーの名前を変更します。

注意:元の.vmテンプレートは編集したり、名前を変更しないでください。

- 2. テキストエディターで、新しいテンプレートに変更を加え、保存します。
- 3. Studio で、ドキュメントを作成するフォルダーを右クリックして、[ドキュメントの生成]>[カスタム形式]を選択します。
- 4. [ドキュメント生成テンプレートの選択] ダイアログボックスで、作成したカスタムテンプレートを選択します。

参照情報

.vm テンプレートファイル

#### Folder\_template.vm

ルートのテンプレート。フレームセットを生成し、以下のテンプレートを呼び出してそのフレームセットに組み込みます。

- All\_folders\_template.vm 対象のフォルダーのサブフォルダーのリストを生成し、overview-frame (左上)に配置します。
- All\_ops\_template.vm すべてのオペレーションのリストを生成し、folderFrame (左下) に配置し ます。
- Header.html ヘッダーを headerFrame (右上) に配置します。
- Folder\_overview\_template.vm 1 つ以上のオペレーションに関する情報を生成し、opFrame (右下)に配置します。

#### All\_folders\_template.vm

フォルダーのコンテンツの表を生成します。

- Header.css 全体的なフォント、色などを定義しているスタイルシート。
- All\_ops\_template.vm すべてのオペレーションのリストを生成し、リンクを作成して folderFrame (左下)に表示します。
- Folder\_contents.vm 選択したフォルダーのコンテンツのリストを生成し、リンクを作成して folderFrame (左下)に表示します。

### All\_ops\_template.vm

すべてのオペレーションのコンテンツおよびすべての子オペレーションのドキュメントの表を生成します。

- Header.css 全体的なフォント、色などを定義しているスタイルシート。
- Op\_template.vm リンクを作成して opFrame (右下)に表示します。

#### Folder\_overview\_template.vm

フォルダーのコンテンツについての概要を表形式で生成します。

- Header.css 全体的なフォント、色などを定義しているスタイルシート。
- Folder\_contents.vm リンクを作成して folderFrame (左下) に表示します。

#### Op\_template.vm

単独のオペレーションのドキュメントを生成します。

- Header.css 全体的なフォント、色などを定義しているスタイルシート。
- Folder\_template.vm リンクを作成して同じフレームに表示します(親フォルダー以下)。
- Folder\_contents.vm フォルダーのコンテンツを folderFrame に表示します。

#### Flow\_template.vm

単独のフローのドキュメントを生成します。

- Header.css 全体的なフォント、色などを定義しているスタイルシート。
- Flow\_template.vm リンクを作成して同じフレームに表示します(親フォルダー以下)。
- Folder\_contents.vm フォルダーのコンテンツのリストを生成し、リンクを作成して folderFrame (左下)に表示します。
- Op\_template.vm リンクを作成して opFrame (右下) に表示します。

#### Folder\_contents.vm

単独のフォルダーのコンテンツの表を生成します。

- Header.css 全体的なフォント、色などを定義しているスタイルシート。
- **Op\_template.vm** リンクを作成して **opFrame** (右下)に表示します。

#### Header.html

ヒューレット・パッカードのバナー。

#### Hp\_rockwell.css

ヒューレット・パッカードのバナー用のスタイルシート。

### Hp\_steps\_307x39.jpg

ヒューレット・パッカードのバナー用の画像。

#### Logo\_hp\_smallmasthead.gif

ヒューレット・パッカードのバナー用のロゴ。

# フローとオペレーションのバージョン履歴管理

構成アイテム、フロー、またはオペレーションが保存されるたびに、そのアイテムの新しいバージョンが作成されます。[バージョン履歴]ダイアログボックスにこれらのバージョンが一覧表示され、以下の操作を実行できます。

- アイテムの前のバージョンを表示する
- 前のバージョンを別の名前で保存する
- アイテムを前のバージョンへ復元する
- 2つのバージョンの差異を表示する

注: バージョン履歴機能は、Studio でのローカルな作成作業にのみ適用され、ソースコントロール管理リポジトリに保存されたバージョンには適用されません。バージョン履歴は、作業内容を ソースコントロールリポジトリにコミットする前に使用します。リポジトリで共有するためにフローをコ ミットした場合、HP OO のローカルのバージョン履歴機能を前のバージョンに戻す作業に使用す ることは推奨されません。

🕵バージョン履歴: Network Check				×
バージョン	日時		コメント	
4*	13/09/18 11:25	変更されました		
3	13/09/18 11:24	変更されました		
2	13/09/17 14:15	変更されました	開く	
1	13/09/13 15:23	/Base [1.0.121]/	復元	Network C
		l	FE車X	ОК

# 実行する操作

### 構成アイテム、フロー、オペレーションの前のバージョンを開く

アイテムの前のバージョンを開いて表示したときに、そのバージョンに別の名前を付けて保存することができます。

- 1. 構成アイテム、フロー、またはオペレーションを右クリックし、[履歴の表示]を選択します。
- 2. 必要なバージョンを右クリックし、[**開く**]をクリックします。選択したバージョンが作成ペインで開きます。

- 3. 開いたバージョンを保存するには、[名前を付けて保存] 🛃 ボタンをクリックし、一意の名前を 付けます。プロジェクトの2つのバージョンが個別に保存されます。
- 4. [OK]をクリックして [バージョン履歴] ダイアログボックスを閉じます。

#### 構成アイテム、フロー、オペレーションの以前のバージョンへの復元

この手順により、構成アイテム、フロー、またはオペレーションを前のバージョンに復元できます。現在のバージョンと前のバージョンの両方を保持するには、「構成アイテム、フロー、オペレーションの前の バージョンを開く」を参照してください。

- 1. 構成アイテム、フロー、またはオペレーションを右クリックし、[履歴の表示]を選択します。
- 2. 復元するバージョンを右クリックし、[**復元**]をクリックします。復元するバージョンが作成ペインで開きます。
- 3. [OK] をクリックして [バージョン履歴] ダイアログボックスを閉じます。
- 4. プロジェクトを保存します。開いたバージョンが現在のバージョンを上書きして保存されます。

#### バージョンの比較

[バージョン履歴]ダイアログボックスでは、構成アイテム、フロー、オペレーションのバージョンを比較する こともできます。現在のバージョンが片側に表示され、前のバージョンが反対側に表示されます。

- 1. 構成アイテム、フロー、またはオペレーションを右クリックし、[履歴の表示]を選択します。
- 2. Ctrl キーを押しながら現在のバージョン(最上行)と前のバージョンの両方を選択します。
- 3. 右クリックし、[比較]を選択します。アイテムの現在の状態と前のバージョンとの差異が表示されます。
- 4. [OK] をクリックして [バージョン履歴] ダイアログボックスを閉じます。

## 参照情報

### [バージョン履歴] ダイアログボックス

🔍バージョン履歴: Network Check				×
バージョン	日時		コメント	
4*	13/09/18 11:25	変更されました		
3	13/09/18 11:24	変更されました		
2	13/09/17 14:15	変更されました	開く	
1	13/09/13 15:23	/Base [1.0.121]/	復元	Network C
			しし連立	
			FC#X	ОК

メニューアイテム	説明
バージョン	自動的に生成された、プロジェクトのバージョン番号。

日時	バージョンが変更された日付と時刻。
コメント	加えられた変更の種類。

# フローおよびオペレーションのブックマーク

[**ブックマーク**] ペインを使用すると、使用頻度の高いオペレーションやフローを簡単に見つけて使用できます。 ブックマークしたフローやオペレーションは、通常のライブラリ内の場所からも利用できます。

フローやオペレーションを [ブックマーク] ペインに追加すると、作成キャンバスの右クリックメニューにそのフローまたはオペレーションが表示されるようになります。

インストールした Studio から別の Studio へ、ブックマークをエクスポートおよびインポートできます。



# 実行する操作

## ブックマークの追加

- 1. Studio ウィンド ウの右上にある [ブックマーク] タブをクリックして、[ブックマーク] ペインを開きます。
- 2. ペインを開いたままにしておくには、ペインの右上隅にある[固定] アイコンをクリックします。
- 3. ライブラリまたは [検索] ペインの結果 からフローまたはオペレーションを [ブックマーク] ペインの適切 なシェルフにドラッグします。

### シェルフの [ブックマーク] ペインへの追加

[ブックマーク] ペインには、デフォルトでフロー用とオペレーション用の2つのシェルフがあります。ですが、 カスタマイズしたシェルフを追加してブックマークを整理できます。

- 1. [ブックマーク] ペインのシェルフのタイトルバーを右クリックし、[追加]を選択します。
- 2. シェルフの名前を入力して [OK] をクリックします。

### シェルフの名前の変更

- 1. 名前を変更するシェルフのタイトルバーを右クリックし、[名前の変更]を選択します。
- 2. シェルフの新しい名前を入力して [OK] をクリックします。

#### シェルフの削除

- 1. 削除するシェルフのタイトルバーを右クリックし、[削除]を選択します。
- 2. 確認 ウィンド ウで [はい] をクリックします。

#### シェルフの表示/非表示

- 1. 非表示にするシェルフのタイトルバーを右クリックし、[非表示]を選択します。そのシェルフは [ブックマーク]ペインに表示されなくなります。
- 2. 非表示にしたシェルフを表示するには、[ブックマーク]ペインを右クリックし、[表示]をクリックしてから、非表示にしたシェルフ名を選択します。
- 3. 非表示にしたシェルフをすべて表示するには、[ブックマーク]ペインを右クリックし、[すべて表示]を 選択します。

#### シェルフの上下移動

上下に移動するシェルフのタイトルバーを右クリックし、[上に移動]または[下に移動]を選択します。

#### シェルフの折りたたみ/展開

- 1. シェルフを折りたたむには、シェルフのタイトルバー上にある二 重 矢 印 (☆)をクリックします。シェルフ のタイトルは表示されますが、シェルフ内のブックマークは非表示になります。
- 2. シェルフを展開するには、二重矢印を再度クリックします。

### ブックマークのエクスポート

インストールした Studio から別の Studio へ、ブックマークをエクスポートできます。

- 1. [ブックマーク] ペイン内を右クリックし、[エクスポート]をクリックします。
- 2. [ブックマークのエクスポート] ダイアログボックスで、エクスポート するシェルフを選択します。

🕵 ブックマークのエクスポート 🛛 🛛
エクスポートするブックシェルフの選択:
□ お気に入りフロー
□ お気に入りオペレーション
OK キャンセル

- 3. [OK] をクリックします。[エクスポートファイルの選択] ダイアログボックスが表示されます。
- 4. ブックマークの保存先を参照し、ファイルの名前を入力します。
- 5. [保存]をクリックします。

### ブックマークのインポート

- 1. [ブックマーク] ペイン内を右クリックし、[インポート]を選択します。
- 2. [インポートファイルの選択] ダイアログボックスで、ブックマークファイルを探して選択し、[開く]をクリックします。
- 3. [ブックマークのインポート] ダイアログボックスで、インポート するシェルフを選択して [**OK**] をクリックします。

🕵ブックマークのインボート 🛛 🛛
インポートするブックシェルフの選択:
□ お気に入りフロー
OK キャンセル

インポート中、同じシェルフにあるブックマークは UUID を基に統合されます。既存のブックマークは保持され、複製は作成されず、新しいエントリがシェルフに追加されます。

# 参照情報

[ブックマーク] メニュー

[ブックマーク] ペインで右クリックすると、[ブックマーク] メニューが表示されます。メニューに表示されるアイテムは、右クリックした時に選択されているアイテムによって異なります。

メニューアイテム	説明
追加	新しいシェルフを [ <b>ブックマーク</b> ] ペインに追加します。
削除	選択したシェルフを [ <b>ブックマーク</b> ] ペインから削除します。
名前の変更	[ <b>ブックマーク</b> ] ペインで選択したシェルフの名前を変更します。
上に移動	[ <b>ブックマーク</b> ] ペインで選択したシェルフを上に移動します。
下に移動	[ <b>ブックマーク</b> ] ペインで選択したシェルフを下に移動します。
非表示	[ <b>ブックマーク</b> ] ペインで選択したシェルフを非表示にします。
表示	[ <b>ブックマーク</b> ] ペインで表示する、非表示のシェルフを選択できます。
すべて表示	[ <b>ブックマーク</b> ] ペインで選択したすべてのシェルフを非表示にします。
折りたたみ ∞	[ <b>ブックマーク</b> ] ペインのシェルフを折りたたみます。 シェルフのタイト ルは表示されますが、 ブックマークは非表示になります。
インポート	
--------	--
エクスポート	

## HP 00 9.x からアップグレードする際のトラブルシュー ティング

## Studio のユーザーインターフェースアイテムの場所

HP OO 9.x での作業に慣れているユーザーが、Studio のユーザーインターフェースアイテムを見つけれられない場合、以下のトラブルシューティングのヒントが対象アイテムを見つけるのに役立ちます。

### リポジトリはどこにありますか?

HP OO ではリポジトリは使用されなくなりました。ファイルはユーザーのファイルシステムにローカルで保存されるため、共同作業ではソースコントロールアプリケーションを使用することをお勧めします。

### [チェックイン] ボタン、[チェックアウト] ボタン、[自分の変更/チェックアウト] ペインはどこ にありますか?

ユーザーがソースコントロールに接続している場合、変更パネルからコミットとチェックアウトを実行可能 です。「ソースコントロールの使用」を参照してください。

# [プロジェクト] ペインと[コンテンツパック] ペインに同じアイテムが含まれているように見えるのはなぜですか?

[プロジェクト] ペインと[コンテンツパック] ペインは以下の点で異なります。

- [プロジェクト] ペインには、プロジェクトで使用できる編集可能なフロー、オペレーション、その他の HP OO が含まれます。
- [コンテンツパック] ペインには、 読み取り専用のフロー、オペレーション、その他の HP OO が含まれ ます。 プロジェクト内 でこれらのオブジェクトを使用できますが、 編集 はできません。 これらのオブ ジェクトを編集 する場合は、 [プロジェクト] ペインにコピーします。

### オペレーションを作成 できないのはなぜですか?

この機能は存在し、オペレーションを作成することは可能です。削除されたのは、組み込みオペレー ション (HTTP、SSH、コマンドラインなど)を作成するオプションです。 これらは既存のテンプレートオペレーションからコピーする必要があります。

HTTP、SSH、コマンドラインなどの組み込みオペレーションを作成することはできません。既存のテンプレートオペレーションからこれらをコピーし、プラグイン内のアクションから新しいオペレーションを作成する必要があります。詳細については、「オペレーションの作成」(225ページ)を参照してください。

### sleep スクリプトレットを作成できないのはなぜですか?

sleep スクリプトレットは廃止されました。HP OO 10.00 では、 スクリプトレットは Rhino で記述する必要があります。

### カテゴリというドメインタームはどこにありますか?

Configuration\Categories フォルダーが追加され、カテゴリを保存してフローを分類することができるようになりました。これは、カテゴリドメインタームの代わりに使用します。

## HP 00 9.x と 10.00 のバージョン比較

タスク	HP OO 9.x での処理	HP OO 10.00 での処理方法
オペレーションの作成	[ <b>新規</b> ] > [オペレーション] メ ニューオプションを使用し、オペ レーションの種類を選択しま す。	アクションプラグインをインポートする か、インポート済みのアクションプラ グインからオペレーションを作成しま す。
		「オペレーションの作成」(225ペー ジ)を参照してください。
共有リポジトリへのフロー のチェックイン	Studio で [ <b>チェックイン</b> ] ボタンを 使 用します。	プロジェクトをファイルシステムにロー カルに保存し、ソースコントロール管 理ツールを使用して、共有リポジト リにコミットします。
		「HP OO Studio のソースコントロー ルの使用」(52ページ)を参照してく ださい。
フローのデプロイと実行	Central のアプリケーションでフ ローを開き、フローを実行しま す。	フローをコンテンツパックとしてリリー スし、API 経由で HP OO にデプロ イします。
		『HP OO インストールガイド』および 『HP OO Application Program Interface (API) Guide』を参照してく ださい。
複数 のインスタンスステップ の作 成	ステップを右クリックし、[複数イ ンスタンスの切り替え]オプション を選択して、そのステップを複 数のインスタンスステップに変更 します。次に、ステップの異なる ターゲットについて複数のルー プを作成します。	[ステップ] パレットの[複数のインスタ ンス] アイコンを作成キャンバスにド ラッグします。1つまたは複数のサブ フローまたはオペレーションを複数の インスタンスレーンに追加し、値の 入力リストでステップの複数のター ゲットを設定します。
		「複数のインスタンスステップのフローの作成」(190ページ)を参照してください。
オペレーションのアクション の作 成	IAction 実装クラスを作成し、 コンパイルして.dll または.jar ファイルを作成します。この.dll または.jar ファイルを自分の Web サービスにコピーし、その Web サービスを Studio にイン ポートします。	アクションプラグインを作成してパッ ケージ化し、Studio にインポートし て、そこから新しいオペレーションを 作成します。 「オペレーションの作成」(225ペー ジ)を参照してください。

タスク	HP OO 9.x での処理	HP OO 10.00 での処理方法
フローの分類に使用する カテゴリの作成	Configuration\Domain Terms フォルダー内のカテゴリと よばれるドメインタームで、新し いカテゴリ用に新しい行を追加 します。	Configuration\Categories フォル ダー内に新しいカテゴリを作成しま す。



