

# HP Operations Orchestration Software Studio

ソフトウェアバージョン: 7.50

*Operations Orchestration フロー作成の手引き*

ドキュメントリリース: 2009 年 3 月

ソフトウェアリリース: 2009 年 3 月



## ご注意

### 保証

HP の製品およびサービスの保証は、かかる製品およびサービスに付属する明示的な保証の声明において定められている保証に限ります。本ドキュメントに記載されたいかなる内容も、追加の保証を構成するものではありません。当社は、本ドキュメントに技術上の誤り、編集上の誤り、記載漏れがあった場合でも責任を負わないものとします。本ドキュメントに記載した情報は、予告なしに変更することがあります。

### 制限付き権利

本ドキュメントで取り扱っているコンピューターソフトウェアは秘密情報であり、その保有、使用、または複製には、HP からの有効なライセンスが必要です。FAR 12.211 および 12.212 に従って、商業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資料、および商業用製品の技術データは、ベンダー標準の商業用ライセンス条件に基づいて米国政府にライセンスされています。

### 著作権

© Copyright 2009 Hewlett-Packard Development Company, L.P.

### 商標

本ドキュメントで言及されているすべての商標は、それぞれの所有者の財産です。

# インターネット上でのドキュメントの検索または更新

ヒューレット・パッカートのソフトウェアのドキュメントは継続的に拡充が図られています。HP OO のドキュメントセットやチュートリアルは、HP Software 製品マニュアルの Web サイトでいつでも入手または更新できます。この Web サイトにログインするには HP パスポートが必要です。

## HP OO のドキュメントおよびチュートリアルを入手するには

1. HP Software 製品マニュアルの Web サイト (<http://support.openview.hp.com/selfsolve/manuals>) に移動します。
2. HP パスポートのユーザー名とパスワードを入力してログインします。

もしくは

HP パスポートを取得していない場合は、[New users – please register] をクリックし、HP パスポートを作成してからこのページに戻ってログインします。

HP パスポートの取得についてご不明な点がある場合は、HP OO の窓口でご確認ください。

3. [製品] リストボックスで下にスクロールし、[Operations Orchestration] を選択します。
4. [製品バージョン] リストで、目的のマニュアルのバージョンをクリックします。
5. [オペレーティングシステム] リストで、該当するオペレーティングシステムをクリックします。
6. [検索] ボタンをクリックします。
7. [検索結果] リストで、必要なファイルのリンクをクリックします。

## ヘルプ、チュートリアルなどの場所

HP Operations Orchestration Software (HP OO) のドキュメントセットは次の内容で構成されています。

- Central のヘルプ

Central のヘルプには次の情報が記載されています。

- フローの検索と実行
- HP OO の機能の設定 (HP OO 管理者向け)
- フローの実行結果から利用可能な情報の生成と表示

Central のヘルプシステムは PDF ドキュメントとしても参照できます。このファイルは、HP OO のホームディレクトリ (¥Central¥docs サブディレクトリ内) にあります。

- Studio のヘルプ

Studio のヘルプは、プログラミングの初心者から上級者まで幅広く対応する、フロー作成のための手引きです。

Studio のヘルプシステムは PDF ドキュメントとしても参照できます。このファイルは、HP OO のホームディレクトリ (¥Studio¥docs サブディレクトリ内) にあります。

- Central 用および Studio 用の動画チュートリアル

HP OO のチュートリアルはどちらも 30 分未満で終わるもので、次の内容について基本的な説明をします。

- Central: フローに基づく情報の検索、実行、表示
- Studio: フローの編集

これらのチュートリアルは、HP OO のホームディレクトリ以下にある Central と Studio のサブディレクトリ内にあります。

- Accelerator Packs フォルダおよび ITIL フォルダ内にあるオペレーションやフローについてのセルフドキュメンテーション  
セルフドキュメンテーションは、フローに含まれているオペレーションやステップの説明で見ることができます。

## サポート

パッチ、トラブルシューティング情報、サポート契約管理、製品マニュアルなどのサポート情報については、次のいずれかのサイトを参照してください。

- <http://support.openview.hp.com>

# 目次

保証 .....	ii
制限付き権利 .....	ii
著作権 .....	ii
商標 .....	ii
インターネット上でのドキュメントの検索または更新.....	iii
ヘルプ、チュートリアルなどの場所 .....	iii
サポート .....	iv
このヘルプシステムおよびガイド.....	1
Studio 初心者向け: How Do I... フォルダー .....	1
フローの概要 .....	1
フローの主な構成要素 .....	2
有効なフローの要件.....	3
フロー作成の応用 .....	4
オペレーション: ステップのモデル.....	4
高度なフローアーキテクチャおよび概念 .....	6
ステップが実行される仕組み .....	7
オペレーション: アーキテクチャとデータの移動 .....	8
フロー、ステップ、オペレーションの高度な概念.....	10
オペレーション、ステップ、フローの入力におけるローカル変数とグローバル変数.....	10
ステップとオペレーション .....	10
フロー、ステップ、オペレーションの入力.....	11
出力と結果.....	11

複数の作成者とバージョン管理 .....	12
HP OO Studio の起動 .....	12
図解による Studio 概要 .....	13
リポジトリペイン .....	14
[My Changes/Checkouts] ペイン .....	16
作成ペイン .....	16
作成ペインのツールバー .....	16
[Bookmarks] ペイン .....	20
[Icons] ペイン .....	21
オブジェクトの [Properties] シート .....	22
Studio のキーボードショートカット .....	22
リポジトリペインのキーボードショートカット .....	22
作成ペインのキーボードショートカット .....	23
プロパティエディター / Inspector のキーボードショートカット .....	23
スクリプトレットペインのキーボードショートカット .....	23
[Bookmarks] ペインのキーボードショートカット .....	23
デバッガーのキーボードショートカット .....	23
ライブラリオブジェクトのバージョン管理 .....	24
ライブラリオブジェクトのチェックアウトおよびチェックイン .....	25
ライブラリオブジェクトのチェックアウト .....	26
ライブラリオブジェクトのチェックイン .....	27
2 者間で競合するオブジェクトバージョンの解決 .....	28
管理者による強制チェックイン .....	29
バージョンの管理 .....	30
変更の破棄 .....	30
オブジェクトの前のバージョンを開く .....	31
前のバージョンへのオブジェクトの復元 .....	32
フォルダーの作成 .....	32

<b>フローの作成</b> .....	<b>33</b>
フローの作成.....	34
テンプレートを使用したフローの作成.....	35
フローへのコールアウトの追加.....	36
<b>フローまたはオペレーションの検索</b> .....	<b>37</b>
フローまたはオペレーションの検索方法 .....	37
説明: 適切なオペレーションの判断 .....	39
フォルダーの説明を表示するには.....	41
フローの説明を表示するには.....	41
オペレーションの説明を表示するには .....	41
<b>複数のオペレーションやフローの説明の表示</b> .....	<b>41</b>
Central ユーザーによるドキュメンテーション生成の有効化 .....	45
カスタム形式のドキュメンテーションの生成.....	47
.vm テンプレートファイルの説明.....	48
Generate Documentation の出力の構成.....	50
Generate Documentation のテンプレートの編集 .....	51
<b>特定のオペレーションを使用するフローの検索</b> .....	<b>53</b>
<b>フローおよびオペレーションのコピー</b> .....	<b>54</b>
<b>フローまたはオペレーションの名前の変更</b> .....	<b>55</b>
<b>フローおよびオペレーションのブックマーク</b> .....	<b>55</b>
ブックマーク: フローおよびオペレーションの追加と削除.....	56
[Bookmarks] ペインのシェルフ: 追加、削除、名前の変更 .....	56
[Bookmarks] ペインのシェルフ: 表示、非表示、移動.....	57
<b>削除したライブラリオブジェクトの復元</b> .....	<b>57</b>

<b>ステップの作成</b> .....	<b>59</b>
フローへのステップの追加.....	59
使用頻度の高いフローおよびオペレーションを基にしたステップの作成.....	61
開始ステップの変更.....	62
ステップのコピー.....	62
ステップの編集.....	63
フロー図内のステップの位置変更.....	63
ステップ実行前に表示するユーザープロンプトの作成.....	63
<b>トランジション: ステップの接続</b> .....	<b>64</b>
トランジションの追加.....	66
トランジションの位置の変更.....	67
<b>リターンステップ</b> .....	<b>68</b>
<b>入力: オペレーションへのデータの提供</b> .....	<b>68</b>
入力の作成.....	69
入力の値の取得方法の指定.....	70
単一の固定入力値の作成.....	73
フローユーザーの入力内容を基にした単一入力値の作成.....	73
フローユーザーの選択内容を基にした単一入力値の作成.....	73
固定入力値リストの作成.....	74
フローユーザーの入力テキストを基にした入力値リストの作成.....	75
フローユーザーの選択内容を基にした入力値リストの作成.....	75
[Previous Step's Result] を基にした入力値の作成.....	76
入力値への資格情報の割り当て.....	76
入力とフロー実行のスケジュール.....	77
入力の削除.....	77
<b>フロー変数: 再利用可能なデータの作成</b> .....	<b>78</b>
フロー変数の作成.....	78
ローカルフロー変数の作成.....	79
予約済みフロー変数.....	79



<b>同時実行: 複数スレッドの並列実行</b> .....	<b>83</b>
並列分岐ステップ .....	84
並列分岐ステップとのデータの移動 .....	85
並列分岐ステップの作成 .....	85
並列分岐ステップとコンポーネントの移動またはコピー .....	86
レーンの追加 .....	87
レーンの複製 .....	87
レーンの削除 .....	87
レーンのサイズ変更 .....	87
レーンの名前変更 .....	88
レーンの開始ステップの変更 .....	88
並列分岐ステップのデバッグ .....	88
マルチインスタンスステップ .....	88
フローデザインでのマルチインスタンスステップの使用 .....	89
マルチインスタンスステップの作成 .....	91
マルチインスタンスステップとのデータの移動 .....	92
マルチインスタンスステップの制限 .....	93
フロー内のマルチインスタンスステップのデバッグ .....	94
ステップのノンブロッキング化 .....	94
<b>ステップのレスポンスの単一化</b> .....	<b>94</b>
<b>チェックポイント: リカバリ用のフロー実行進捗の保存</b> .....	<b>95</b>
<b>スクリプトレット</b> .....	<b>96</b>
スクリプトレットの作成 .....	97
スクリプトレットのデバッグ .....	98
ほかの用途を目的としたスクリプトレットの保存 .....	99
<b>出力、レスポンス、およびステップの結果</b> .....	<b>100</b>
オペレーション出力 .....	100
レスポンス .....	101
ステップの結果 .....	101

出力の追加と削除 .....	102
出力のソースの変更 .....	102
ステップの結果の追加と削除 .....	103
結果のソースの変更 .....	104
出力および結果のフィルタリング .....	104
フィルターを作成 .....	104
フィルターの詳細 .....	106
Diff Case .....	107
Extract Number .....	107
Format .....	107
Line Count .....	107
Regular Expression .....	107
Remove Duplicate Lines .....	108
Replace .....	108
Round Number .....	109
Scriptlet .....	109
Select Line .....	110
Select Range .....	110
Sort .....	111
Strip .....	111
Strip Whitespace .....	112
Table .....	112
フィルターの保存と再使用 .....	113
レスポンス: 結果の評価 .....	113
フローへのレスポンスの追加 .....	116
ステップのアイコンの変更 .....	116
システムプロパティの作成 .....	117
フローデザイン .....	118
サブフローを使用したフローデザインの単純化 .....	118

サブフローから親フローへのデータの引き渡し .....	119
ステップのベースとなるオペレーションの変更 .....	120
ステップの説明.....	121
フローの自動実行.....	122
フローのデバッグ .....	122
フローのデバッグ .....	123
デバッガーでのフロー変数の値の変更.....	126
ブレークポイントの使用.....	127
デバッグ実行でのレスポンスの上書き .....	128
別のユーザーとしてログオン .....	129
リポジトリ: フローおよびオブジェクト用のライブラリ.....	129
作成用にリポジトリを追加して開く .....	130
リポジトリ間のフロー要素の移動 .....	131
ターゲットリポジトリの設定 .....	132
パブリックリポジトリからのパブリッシュとパブリックリポジトリへの更新 .....	132
リポジトリのパブリッシュ: 方法.....	133
リポジトリからの更新: 方法.....	135
パブリッシュまたは更新のロールバック .....	138
リポジトリのエクスポート.....	138
リポジトリのインポート.....	140
リポジトリの検査 .....	142
リポジトリの暗号化.....	142
リポジトリの暗号化 .....	142
暗号化されたリポジトリを開く .....	143
リポジトリの復号化.....	143
2つ目の暗号化されたリポジトリのコピーの作成 .....	143

リポジトリのバックアップと復元.....	144
Web サービスからのオペレーションの作成 .....	145
リモートアクションサービスを使用した Central 外での 運用.....	147
Central での RAS に依存するオペレーションの実行方法 .....	149
RAS を使用できるかどうかの確認.....	149
既存の RAS の追加 .....	150
RAS 参照の追加 .....	150
既存の RAS 参照の再設定 .....	151
RAS 参照の変更.....	152
Web サービスにアクセスするオペレーションの作成.....	154
RAS に依存するオペレーションの実行のトラブルシューティング .....	154
Studio の例: 任意の複数ドメインで実行するフロー .....	154
RAS からのオペレーションの作成.....	155
RAS からのオペレーションの作成 .....	156
デフォルトの RAS にはないオペレーションの追加.....	156
RAS 参照の削除.....	157
RAS コンテンツのインポート .....	157
データが正しい形式かどうかの評価 .....	158
エバリュエーターの作成.....	158
エバリュエーターの編集.....	159
エバリュエーターの削除.....	159
ダッシュボードグラフでのレポート作成用の値の記録.....	160
ダッシュボードグラフ用のドメインターム .....	161
ドメインタームまたはドメインターム値の追加 .....	162

ユーザーへのメッセージ表示 .....	162
カテゴリ: フローの分類 .....	163
ユーザープロンプトの選択リスト .....	164
システムアカウント: 安全な資格情報 .....	165
システムアカウントの作成 .....	165
システムアカウントの編集 .....	166
システムアカウントの削除 .....	166
OO オブジェクトへのアクセス管理 .....	167
フォルダーと HP OO オブジェクトの権限の設定 .....	169
グループのアクセス権限のデフォルト設定 .....	171
新しいオペレーションを作成するには .....	171
オペレーションタイプ: プロパティの設定 .....	173
Cmd (コマンドライン) オペレーション .....	173
フロー実行サマリレポートオペレーション .....	175
HTTP オペレーション .....	176
Perl スクリプトオペレーション .....	177
RAS オペレーション: RAS のための IAction プログラミング .....	178
スクリプトレットオペレーション .....	180
Secure シェル (ssh) オペレーション .....	181
Telnet オペレーション .....	183
シェルオペレーション .....	184
Shell Wizard: シェルオペレーションを使用するフローの作成 .....	184
シェルオペレーションの使用 .....	189
正規表現での作業 .....	190
フローのテストと配布 .....	192

フロー使用の制限 .....	193
Central でフローの閲覧可能なユーザーを制限 .....	194
オペレーション用の IAction の作成 .....	194
IAction を実装するオペレーションの作成概要 .....	195
トラブルシューティング .....	195
同じパブリックリポジトリに接続した Studio の 2 つのインスタンスにログインしています。Studio の一方 のインスタンスの作業スペースで変更した内容がもう一方の Studio に反映されません。 .....	195
存在していることがわかっているフロー、オペレーション、またはシステムオブジェクトが、自分の Library に表示されません。 .....	195
オペレーションのスクリプトレットが正常に実行されません。 .....	195
パブリックリポジトリのロックを損失したというエラーがレポートされます。 .....	196
ステップを削除した後に削除を元に戻しましたが、変更内容が保存されていません。 .....	196
インデックス .....	197

# このヘルプシステムおよびガイド

HP OO Studio のヘルプ (Studio\_AuthorsGuide.pdf としても提供) では、Studio の概要やフローを作成する際の手順について説明します。

このヘルプは次の 3 つのセクションに分かれています。

- クイックビュー
- 基本的なフロー作成  
このセクションでは、Studio の概要、およびフローの作成に関連する次の基本的なタスクについて説明します。
  - フローおよび Accelerator Packs のインポート
  - フローおよびオペレーションの検索と使用
  - フローおよびフローステップの作成
  - フローのテスト
  - Central ユーザーへのフローの公開

基本的なフロー作成タスクについて学習するときは、フローの編集に関する Studio のチュートリアルを見てからこのヘルプシステムを参照すると効率的です。

- 高度なフロー作成  
このセクションでは、オペレーションやステップの作成 (入力の定義、フロー全体でのデータの移動など)、各ステップに基づきフローの経路を論理的に決定するルールの作成など、より高度な内容を取り上げます。また、オペレーションアーキテクチャとその内部でのデータの移動、さまざまな種類のオペレーションの使用方法についても説明します。
- オペレーション用の IAction の作成  
このセクションでは、IAction プログラミングの概要と、他のシステムとの統合やフローのリモート実行を行えるよう IAction を使ってフローの機能を拡張する方法を簡単に説明します。

## Studio 初心者向け: How Do I... フォルダー

Studio の使い方を習得する前に実用的なフローを作成して実行する必要がある場合は、**How Do I...** フォルダーを開いてください。

このフォルダー内のフローには、コールアウトによる注釈、[Description] タブに表示される説明、その他フローの使い方に関する情報が含まれています。

## フローの概要

フローとは、ヘルスチェックやトラブルシューティング、繰り返し実行する IT サポートタスクなどを自動化するために連結されたアクションの集まりです。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。

Operations Orchestration Software を使用していない場合、次のような方法が考えられます。

- Webサイトを15分おきにチェックする担当者を決め、必要な場合はコンテンツを手動でサイトにパブリッシュする。
- サイトをチェックし、コンテンツが正しくない場合にイベントまたはアラートを発行する監視システムをプログラミングする（コンテンツのパブリッシュは手動で行う）。必要に応じてテクニカルサポートスタッフが問題点を調べる。

一方、Studioを使用すれば、必要なタスクをすべて自動的に実行するフローを作成できます。必要なタスクとは、Webサイトの定期的なチェック、イベントまたはアラートの作成、サイトへのコンテンツのパブリッシュ、原因となる問題のトラブルシューティングと修復、実行されたステップの記録などです。

この例を基にして、フローの基本的な概念について説明します。こうしたサイトチェック要員向けのガイドや手順書には、次のような内容が書かれていると考えられます。

**手順 1:** `mysite.com/mypage.htm` を開きます。最新の更新を反映したテキストが表示されない場合は、手順 2 を実行します。

**手順 2:** `mypage.htm` を、サーバー `development1` からサーバー `production1` にコピーします。

**手順 3:** この Web ページに正しいテキストが表示される頻度と、問題の修復が必要となる割合を追跡記録します。

Studioでは、問題の選別、診断、解決の自動化、システムやアプリケーションのヘルスチェック、オペレーションやデータセンターにおける定期的なメンテナンス処理などを行うフローを作成できます。人手による作業と同じく、フローにもデータを収集するステップや適切なアクションを実行するステップがあります。各ステップはデータを受け取ってそれに対応しますが、検出された内容によってステップのレスポンスは異なります。また、各ステップで使用されたデータや結果が記録されるので、ステップのアクションをすべて追跡できます。

## フローの主な構成要素

ステップはフローの基本単位です。

フローには、ステップのほかに次のような主要要素があります。

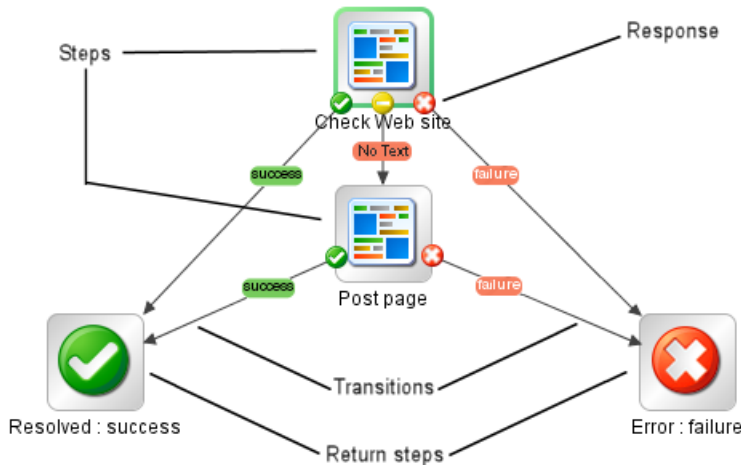
- **オペレーション:** フローの実際の作業を行います。前述の例の手順 1 では、Web ページに特定のテキストが含まれているかどうかをチェックするというオペレーションを用いています。手順 2 のオペレーションはファイルをコピーすることです。ステップはいくつかのオペレーションで成り立っています。つまり、オペレーションはステップのテンプレートです。したがって、オペレーションはどのフローでも構成要素として存在します。
- **入力:** アクションが発生するために必要なデータをオペレーションに渡します。Web ページをチェックするオペレーションでは、チェック対象のページ (`mysite.com/mypage.htm`) 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。ファイルコピーオペレーションでは、コピー元とコピー先の場所についての情報が必要です。次のようなデータを入力として使用できます。
  - フローの実行者が入力する。
  - 特定の値に設定する。
  - 別のステップで収集された情報から取得する。
- **レスポンス:** オペレーションにより起こり得る結果です。Web ページの取得オペレーションには、「page not found」、「text not found」、「success」（ページが存在し、かつ目的のテキストが表示される場合）という 3 種類のレスポンスがあります。ファイルコピーオペレーションの場合は、「success」か「failure」しかありません。



- **トランジション:** Web ページの読み取りオペレーションでは、Web ページが見つからない、ページは存在するがテキストが存在しない、または、ページが存在し目的のテキストも表示される、というそれぞれの結果に応じて異なるレスポンスが必要となります。トランジションは、あるオペレーションレスポンスからいずれかの後続ステップへの遷移を表す要素であり、これによってそのレスポンスの次のステップが決まります。

以上の内容を図として整理し、次のようにまとめました。

- フローステップのオペレーションは入力データを使ってタスクを実行し、そこから結果を取得する。
  - オペレーションに対するレスポンスは複数あり、オペレーションの結果に応じてそのうちの1つが選択される。
  - 各レスポンスは、トランジションによってフロー内のいずれかの後続ステップに接続されている。
- つまり、レスポンスの選択によって、フローの特定の実行処理の後続ステップが決まります。



**図 1 – フローの構成要素**

次に、オペレーションの主な要素について説明します。

## 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必要があります。

- ステップが1つ以上あり、そのうちの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、その最も基本となるオペレーションに対するそれぞれのレスポンスとその後続ステップがトランジションによって接続されている。ほとんどの場合オペレーションには複数のレスポンスがあるので、大部分のステップは1つ以上の後続ステップを接続しているトランジションが複数存在します。フロー図では、ステップのオペレーションのレスポンスは次の図のようにステップの傍らに名前とともに表示されます。



**図 2 – ステップおよびそのオペレーションの失敗と成功を表すレスポンス**

- 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。
- 値を戻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（定義）されている。

これらはフローの基本要素です。これらを設定することにより、有効なフローを作成できます。フローの有効性を向上させる方法については、次の「[フロー作成の応用](#)」およびこのトピックで紹介している他のトピックを参照してください。

## フロー作成の応用

フローの実用性を高めるための方法はいろいろあります。次に挙げるのは、フローの性能、堅牢性、より多くの情報をユーザーに返せる機能などを向上させることができる初歩的な方法の一部です。入力やトランジションなど一部の内容は前のトピックで取り上げていますが、その使い方によってはさらに応用が利きます。

- 入力へのデータの割り当て方法により、フローをリアルタイムで実行できるようにする。  
入力にデータを割り当てるさまざまな方法については、「[入力: オペレーションへのデータの提供](#)」を参照してください。
- フロー変数にデータを格納し、フロー内のステップ間で、またはサブフローから親フロー（サブフローがステップの1つであるフロー）にデータを渡せるようにする。  
フロー内の他のステップで使用できるフロー変数の作成方法については、「[フロー変数の作成](#)」を参照してください。  
フロー変数を他のフローに渡す方法については、「[サブフローから親フローへのデータの引き渡し](#)」を参照してください。
- 情報を取得し、ほかの場所でも利用できるように加工する。  
データを結果として取得する方法、およびフィルターを用いた取得データの絞り込み、加工、書式設定の方法については、「[出力、レスポンス、およびステップの結果](#)」を参照してください。
- フローの各ステップで発生したことを Central ユーザーに知らせ、フローが取得した情報を提供する。  
Central ユーザーへの各ステップの発生事項の伝達については、「[トランジションの追加](#)」を参照してください。
- 主要な情報を記録して、Central ユーザーが定義する Central のダッシュボードでグラフを作成できるようにする。  
ダッシュボードでレポートを作成するための情報の記録については、「[ダッシュボードグラフでのレポート作成用の値の記録](#)」を参照してください。

## オペレーション: ステップのモデル

オペレーションは、レスポンスのほかに次の要素で構成されています。

- **コマンド:** オペレーションによる実際の操作の大部分を決定するものです。（オペレーションには、データ処理を行うスクリプトレットが含まれていることもあります。）コマンドには、コマンドライン、HTTP 操作、実行するスクリプトなど、いくつかの種類があります。例を挙げると、ディレクトリ一覧の取得、サービスが実行中かどうかのチェック、Web サービスの実行、UNIX の `vmstat` コマンドの実行などがオペレーションで可能な操作です。
- **結果:** コマンドを実行して戻されたデータのことを、結果と呼びます。たとえば、`dir` コマンドによって戻される結果にはファイルリストが含まれています。`ps` コマンドの結果はプロセスのリストです。ほとんどのコマンドは、戻りコード、標準出力 (`stdout`)、エラー出力 (`stderr`) など複数のデータが結果として戻されます。HTTP ページ取得オペレーションの場合、HTTP 戻りコード (200、302、404 など) およびページ上のデータに関する結果が必要です。  
コマンドによっては、フロー内で後で必要になるさまざまなデータを戻すものもあります。コマンドを1つ使用して Linux のメモリ統計情報を取得するオペレーションでは、メモリの空き容量、使用可能なメモリの合計量、スワップメモリの使用量などがわかります。
- **フロー変数:** フロー変数を作成してそこに結果データを格納できます。このデータは、フロー内の他のステップに対する入力として後で利用できます。
- **フィルター:** コマンドから戻されたデータに基づく適切なレスポンスを判断するために、結果から主要なデータを絞り込む、あるいはデータを加工するスクリプトレットを作成することが必要になる場合があります。

- **ルール:** オペレーションの結果を評価して、ステップが実行されたときにとるべきオペレーションレスポンスを決定します。ルールの評価対象は、結果フィールド、データ文字列、戻りコード、エラーコードなどです。

オペレーションに対する特定のレスポンスは、結果フィールドに指定したデータ、またはいずれかの結果フィールドから抽出されたデータと、レスポンスのルールに記述された条件が一致したときに選択されます。ルールに記述した文字列とルール用に選択した結果フィールドとを比較する、または両者が一致するというルールを作成します。

前述の例での Web ページ取得オペレーションについて考えてみます。

- 「page not found」というレスポンスは、HTTP 戻りコードが 404 の場合に選択される。
- 「text not found」というレスポンスは、チェック対象のテキストがページ上のデータに含まれていない場合に選択される。
- **スクリプトレット:** スクリプトレット (JavaScript または Perl で作成) 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利用できるように加工する目的で使用します。

この仕組みを表したのが次の図です。

- オペレーションが、フロー、ステップ、またはオペレーションの入力から値を取得する。
- オペレーションの結果に含まれるデータがフロー変数に渡され、フローのほかの部分で利用できるようになる。

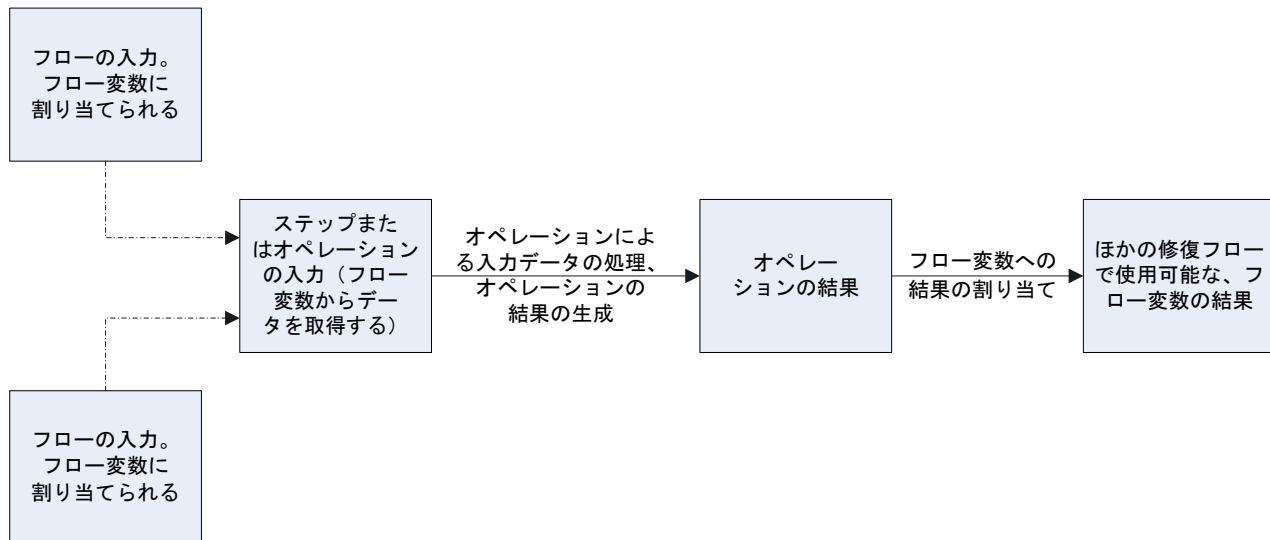


図 3 - オペレーションがデータを取得する仕組み

この図を、あるディレクトリに対して Windows コマンドラインの `dir` コマンドを実行するフローに当てはめてみます。

- フローの入力は、必要な 2 つのデータ、つまりホストコンピューター名と `dir` コマンドの実行対象のディレクトリ名です。
- この 2 つの入力データは、それぞれ「host」、「directory」という名前のフロー変数に格納されます。
- オペレーションの入力の値はフロー変数から取得します。
- 入力に基づきオペレーションがタスクを実行した後、このステップのオペレーション結果が別のフロー変数に代入され、フロー内の他のステップのオペレーションで利用できるようになります。

オペレーションからステップを作成する場合、オペレーションの各レスポンスは必ず別のステップへのトランジションの開始点になります。つまり、フローの特定の実行処理中に、オペレーションの結果を評価するルールによってオペレーションのレスポンスが決定され、さらにそのレスポンスによって次のトランジションおよびその後のステップのパスが決まります。

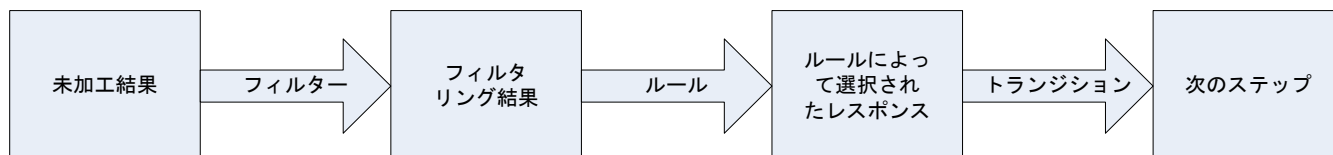


図 4 – オペレーションの未加工結果から次のステップが決まる仕組み

注: オペレーションから作成されるステップの結果にフィールド内の値を割り当てるようにオペレーションの結果を設定してから、フローの結果にフィールド内の同じ値を割り当てるように当該ステップ結果を設定することが可能です。フローの結果は、他のフローに値を渡す目的で使用できます。

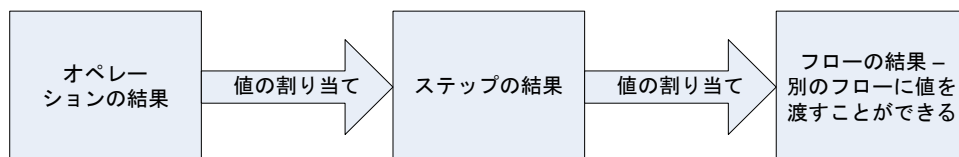


図 5 – オペレーションの結果がフローの結果になる仕組み

## 高度なフローアーキテクチャおよび概念

「[フローの概要](#)」では、フローの基本概念、および既存のフローに簡単な変更を加える方法について説明しました。この章では、次の概念についてさらに詳しく説明します。

- ステップ実行で行うアクションの順番
- オペレーションアーキテクチャと情報フロー
- 入力用データソースの定義やレスポンスの定義などの、ステップおよびオペレーションの要件
- HP OO オブジェクトとしてのフロー、ステップ、オペレーション
- オペレーションの出力データを加工できるようにするスクリプトレットの作成

スクリプトレットは、オペレーション内に含まれる JavaScript または Perl のスクリプトです。スクリプトレットを用いるとオペレーションのカスタマイズが比較的少なく済むため、プログラミングの時間を短縮できます。

高度な作成では、JavaScript や Sleep で作成するスクリプトレット以外にも次の知識が必要となります。

- 入力とフロー変数
- ステップとオペレーションの区別
- 結果、未加工結果、およびレスポンスの区別
- データフィルター

## ステップが実行される仕組み

ステップが実行されるときは、以下のアクションがこの順番で実行されます。

1. フロー変数およびグローバルデータ値の集合から入力値が取得されて入力に適用され、ステップのオペレーションで使用できるようになります。
2. 入力値に対する変更がある場合は、フロー変数およびグローバルデータ値の集合にその変更が適用されます。
3. ステップのオペレーションが実行されます。  
オペレーション内でのデータの取得、移動、変更の詳細については、「[オペレーション: 構成とデータの移動](#)」を参照してください。
4. オペレーションにスクリプトレットが含まれている場合は、そのスクリプトレットが実行されます。  
オペレーションのスクリプトレットでは次のことを実行できます。
  - a. オペレーションのレスポンスを選択する。
  - b. オペレーションのプライマリ出力を設定する。  
プライマリ出力とはある入力に値を提供する結果のことであり、その入力値の割り当てには [Previous Step's Result] を使用します。
  - c. ローカルおよびグローバルのフロー変数やデータ値に変更を加える。
  - d. RAS を使用する IAction がオペレーションに含まれているなど、読み取る値がある場合にオペレーションのレスポンス値を読み取る。この IAction にはレスポンスがあり、オペレーションのスクリプトレットで取得、読み取りができます。
5. オペレーションのスクリプトレットでオペレーションの主結果が設定されていない場合は、この時点で設定されます。
6. オペレーションのスクリプトレットでオペレーションのレスポンスが選択されていない場合は、レスポンスを選択するオペレーションの評価ルールを使って、この時点で選択されます。
7. ステップにスクリプトレットが含まれている場合は、そのスクリプトレットが実行されます。  
ステップのスクリプトレットでは次のことを実行できます。
  - a. オペレーションのレスポンスを選択する。
  - b. ローカルおよびグローバルのフロー変数に変更を加える。  
**注:** ステップのスクリプトレットでは主結果を設定できません。
8. 選択されたレスポンスに関連付けられているトランジションが選択されます。
9. 上記と同じ順序で次のステップが実行されます。

## オペレーション: アーキテクチャとデータの移動

オペレーションは、フローのステップに関連付けられている定義済みの一連のアクションです。あるフローが関連付けられているステップをサブフローとみなした場合、そのサブフローはオペレーションの一種と言えます。

オペレーションでのデータの移動を制御するために欠かせない概念の1つとしてコンテキストがあります。コンテキストは、さまざまな時点でステップと交換可能な各種の値を保持しているコンテナです（次の図を参照）。コンテキストには、ローカルとグローバルの2種類があります。ローカルコンテキストはステップの実行中に存在し、グローバルコンテキストはフローの実行中に存在します。ローカルコンテキストまたはグローバルコンテキストの間で値を受け渡すことができます。

次に、単独のオペレーション（つまりサブフローではないオペレーション）が機能する仕組みを具体的に説明します。

- コア機能（コアとも称される）：オペレーションのビジネスロジックをカプセル化したもの

Web オペレーションの場合、コア機能は **IAction** インターフェース実行メソッドおよびメソッドのパラメーターです。パラメーターから実行メソッドにデータが提供され、実行メソッドの動作が決まります。

オペレーションの実行前に、すべての入力値がローカルまたはグローバルのコンテキストにコピーされます。入力値はローカルまたはグローバルのコンテキストに結合されることもあります。

入力はコアによって未加工結果にマッピングされる場合があります。

- スクリプトレットによる未加工結果のさらなる処理（オプション）
- レスポンスの決定

次の図に示すように、オペレーション内でのデータ移動では以下の要素が関与します。

- 未加工結果

**IAction** インターフェースがコア機能に含まれている場合、データおよび状態が未加工結果となります。

- スクリプトレット

ステップの最後に実行される解釈プログラム（オプション）。スクリプトレットでは主に、オペレーションの未加工結果の評価や、オペレーションの出力データの生成が行われます。

- 出力データ

オペレーションによって生成されるデータ（該当する場合）。

- レスポンス

オペレーションの出力の評価、およびオペレーションのステップに最適なトランジションの判定。

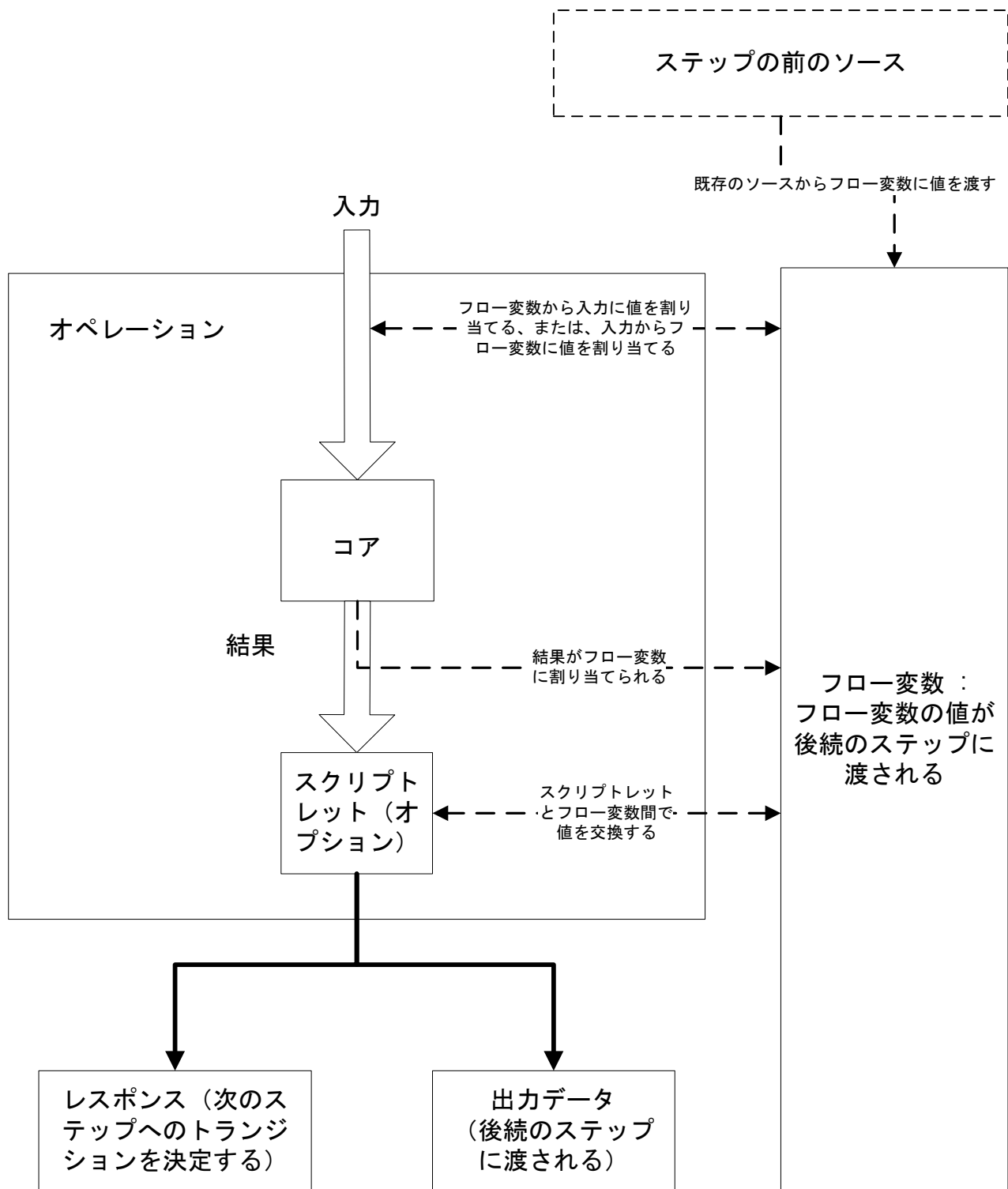


図 6 - オペレーション内でのデータの移動

注: Integrations、Operations、Utility Operations の各フォルダー内にあるオペレーションおよびフローはシールされています。つまり、入力に特定の固定値を指定する場合を除き、これらのオペレーションやフローを編集することはできません。



# フロー、ステップ、オペレーションの高度な概念

効率的なフローを作成するためには以下の概念が重要です。

## オペレーション、ステップ、フローの入力におけるローカル変数とグローバル変数

ローカル変数やグローバル変数を使用して、フロー内のステップ間でデータを移動できます。

- グローバル変数は、実行中のフローのすべてのコンポーネント（サブフロー、親フローも含む）で利用できます。サブフローでグローバル変数にある値を指定した場合、その実行中は親フローでも同じ値を利用できます。
- ローカル変数は、ローカル変数が定義されているフロー内でのみ利用できます。

## ステップとオペレーション

ステップとオペレーションの基本的な違いは、ステップはオペレーションのインスタンス（実体）であることです。

- HP OO による処理は、オペレーションおよびフローによって決まります。そのため、フローは一種のオペレーションであると言えます（フローを使用して別のフローのステップを作成するときは特にこのことに留意してください）。
- ステップは常にオペレーションまたはフローのインスタンスです。したがって、フローの作成キャンバスにオペレーションをドラッグした時点で、そのオペレーションはステップを作成するためのテンプレートとなります。

オペレーションとそれを基にしてできたステップには、次のような大きな違いがあります。

- ステップの編集（ステップ用の入力の定義など）は、ステップからオペレーションにアクセスして行うのではなく、ステップの **Inspector** で行います。  
オペレーションを編集する場合、そのオペレーションと関連付けられたすべてのステップのベースであるテンプレートに変更を加えることとなります。つまり、当該オペレーションと関連付けられているステップは、どのフローに含まれているかにかかわらず、すべて変更されるということです。そのため、オペレーションを編集する際は慎重に行わないと、そのオペレーションを使用しているフローを中断する可能性があります。
- ステップで作成されるスクリプトレットとオペレーションで作成されるスクリプトレットにも相違点があります。次に挙げるのがその例です。
  - オペレーションのスクリプトレットは、オペレーションのレスポンスの値を読み込みません。
  - オペレーションのスクリプトレットは、オペレーションから作成したステップの **[Scriptlet]** タブに表示されません。
  - ステップのスクリプトレットの結果をステップの結果に渡すことはできません。（ステップのスクリプトレット内ではなく、ステップの結果に適用するスクリプトレットフィルター内で必要なタスクを実行することにより、この制約を回避できます。）



## フロー、ステップ、オペレーションの入力

入力はいずれも1つの変数にマッピングされ、値の供給元として各種ソースを利用できます。

入力に追加する要素に応じて、入力の値が取得されるタイミングが決まります。

- フローの入力は、最初のステップが実行される前に値を取得します。  
フローに必要であり、かつフロー内での処理によりフローのプロパティとして生成されるデータではない入力を設定することをお勧めします。これらの入力値がフローで利用可能な状態にしてからフローの実行を開始できるためです。

- ステップの入力は、ステップのオペレーションが実行される前に値を取得します。

入力には各要素の違いも反映されます。

- ステップ用に（ステップの Inspector で）作成する入力は、ステップと関連付けられたオペレーション用の入力ではありません。この入力の値は、オペレーションの実行前に取得されます。
- ライブラリにあるオペレーション用の入力を（Library のオペレーションを右クリックして表示される [Properties] シートで）変更する場合、以降に作成するそのオペレーションのインスタンスはすべて、その変更が反映されます。

入力用のデータソースの定義の詳細については、「[入力: オペレーションへのデータの提供](#)」を参照してください。

## 出力と結果

オペレーション（およびフロー）により生成される出力を基にして、作成者はオペレーションやフローの出力を作成できます。作成者は、これらの出力から、ステップの結果を定義します。

オペレーションの作成者は、オペレーションの出力のいずれか1つをプライマリ出力に指定します。**プライマリ出力**とは、ステップの主結果として使用される出力です。特定の出力フィールドをプライマリ出力として指定する手順については、「[出力の追加と削除](#)」を参照してください。

ステップの結果には次の3種類があります。

- **未加工結果**。フローのコンテキスト内で実行されたオペレーションから戻された未加工データを表すキー値のペアの集合からステップが取得する結果です。

たとえば、Windows XP マシンで ping オペレーションを実行した場合、次の結果を取得できます。

```
{
  Code = "0"
  Error String = ""
  Output String =
"Pinging apple.com [17.254.3.183] with 32 bytes of data:

Reply from 17.254.3.183:bytes=32 time=24ms TTL=244
Reply from 17.254.3.183:bytes=32 time=24ms TTL=244
Reply from 17.254.3.183:bytes=32 time=25ms TTL=244
Reply from 17.254.3.183:bytes=32 time=26ms TTL=244

Ping statistics for 17.254.3.183:
    Packets:Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 24ms, Maximum = 26ms, Average = 24ms"
}
```

未加工結果の値にフィルターを適用することにより、結果を絞り込むことができます。上の結果を例にとると、**Ping** オペレーションの平均待ち時間だけが必要な場合に、オペレーションで [Primary Output From Field] を選択して、文字列の最後にある「24ms」トークンを抽出するフィルターを指定できます。詳細については、「[出力および結果のフィルタリング](#)」を参照してください。

- **その他の結果。** フロー作成者がステップのエディターの [Results] タブで作成した結果です。

一方、レスポンスは、それぞれが1つのトランジションによって次のステップに連結されている結果です。レスポンスは直後のステップに連結されることから、フローの次の処理を決定する要素と言えます。ただし、リターンステップの場合は例外です。リターンステップのレスポンスは、フロー全体の結果を戻します。

次の表は、オペレーション、フロー、およびステップと、上記の結果との対応関係を表したものです。ただし、シールされているオペレーションの出力の処理は変更できないことに注意してください。

	未加工出力	出力	結果	レスポンス
オペレーション (またはフロー)	✓	✓		✓
ステップ			✓	

## 複数の作成者とバージョン管理

Studio では、複数のインスタンスが同時に Central のパブリックリポジトリに接続することが可能です。複数の作成者が同時に Central リポジトリで作業しているときに変更を調整したり競合の発生を防止したりするのがチェックイン/チェックアウト機能です。また、この機能を使って作成チームのバージョン管理を行うこともできます。チェックイン/チェックアウト機能では、パブリックリポジトリ内のフロー、オペレーション、Configuration フォルダのオブジェクトなどを操作するときはあらかじめチェックアウトし、変更を加えた後は自分以外の作成者が参照できるように再度チェックインします。

リポジトリオブジェクトのチェックアウト、および実際の手順については、「[ライブラリオブジェクトのバージョン管理](#)」を参照してください。

## HP OO Studio の起動

### Studio を起動するには

- OO のホームディレクトリ (デフォルトでは C:\Program Files\Hewlett-Packard\Operations Orchestration) にある **Studio** サブディレクトリで、**Studio.exe** をダブルクリックします。

もしくは

Windows の [スタート] ボタンから Studio を起動します。プログラムの一覧では、[Hewlett-Packard] > [Operations Orchestration] の次に表示される [HP Operations Orchestration Studio] が Studio です。

もしくは

デスクトップ上、または [スタート] ボタンの上部に表示されるプログラムの中に Studio のショートカットがある場合は、そのショートカットをクリックします。

通常は、1 台のコンピューターでシングルユーザーとして Studio の特定のインスタンスを起動、ログインします。次の制限があることを確認してください。



### 鍵となる情報：

- 同一ユーザーとしてログインしたか別のユーザーとしてログインしたかにかかわらず、別のコンピューターから複数のリモートデスクトッププログラムを使用して、1 台のコンピューターで Studio の複数のインスタンスを起動しないでください。データの破損や損失、または Studio の異常終了などの想定外の動作が発生する可能性があります。
- 同一ユーザーとして Studio の 2 つのインスタンスにログインした場合、Central の同一インストールのパブリックリポジトリに接続しないでください。リポジトリ内にあるユーザーの作業スペースが壊れることがあります。また、Central がクラスタ化されたときに起こることもあります。

## 図解による Studio 概要

Studio は主に以下の要素で構成されています。

- 「**リポジトリペイン**」：作業中のリポジトリが表示されます。
- 「**My Changes/Checkouts ペイン**」：OO オブジェクトのチェックイン/チェックアウトの状態を確認できます。
- 「**作成ペイン**」：フロー図の作業をする場所。フロー、ステップ、オペレーション、トランジションの [Properties] シート、および多数のシステムオブジェクトやフローオブジェクト（選択リスト、ドメインターム、フィルター、スクリプトレットなど）用のエディターが用意されています。
- 「**Bookmarks**」 ペイン: 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。[Bookmarks] タブをクリックするとこのペインを表示できます。
- 「**Icons**」 ペイン: オペレーションやステップにドラッグできるアイコンが揃っています。[Icons] タブをクリックするとこのペインを表示できます。

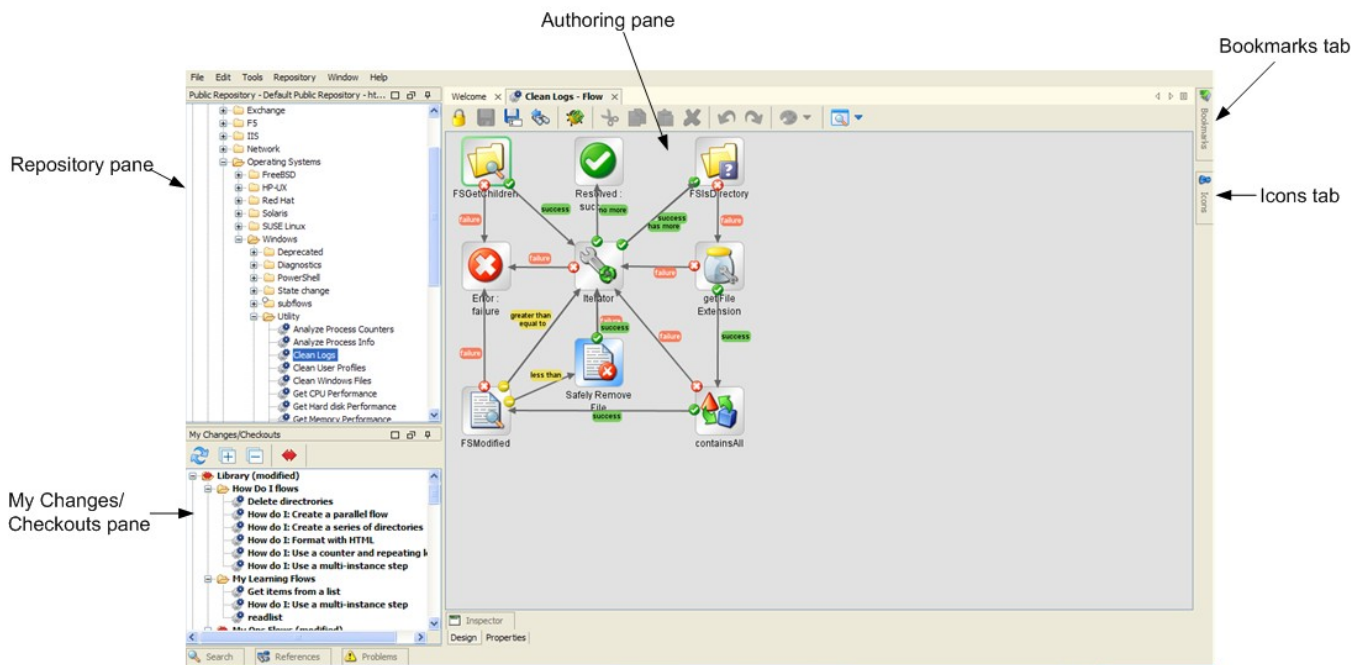


図 7 - Studio の主な構成要素

## リポジトリペイン

リポジトリペインの内容は、次の2つのフォルダーに分かれています。

- **Library** フォルダー: Studio のインストール時に取得したフローおよびオペレーションがすべて格納されています。
- **Configuration** フォルダー: その他の OO オブジェクト（ライブラリオブジェクトとも言います）が格納されています。OO オブジェクトは、オペレーション結果の処理や Central のダッシュボードグラフ用の結果データのレポートで使用するほか、Central の内部と外部、またはリモートでのフローの実行を容易にする目的でも使用します。

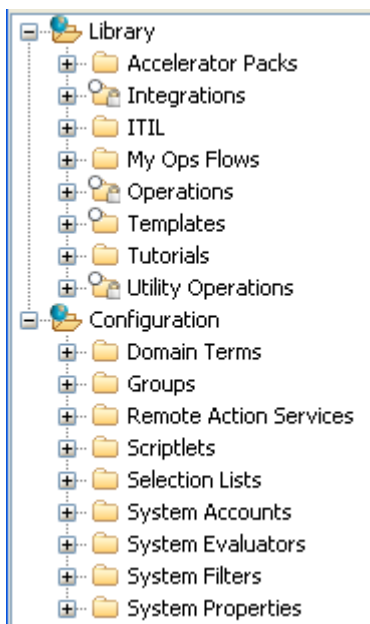


図 8 – リポジトリペインに表示されたライブラリ

Library フォルダーを展開し、さらにフローのフォルダーを展開すると、Studio のデフォルトのコンテンツ、つまり各種テクノロジーで動作するフローやオペレーションが表示されます。フロー作成時に利用できるフローやオペレーションについて参考になるよう、さまざまなコンテンツのフォルダーが展開された状態のスクリーンショットを次に示します。

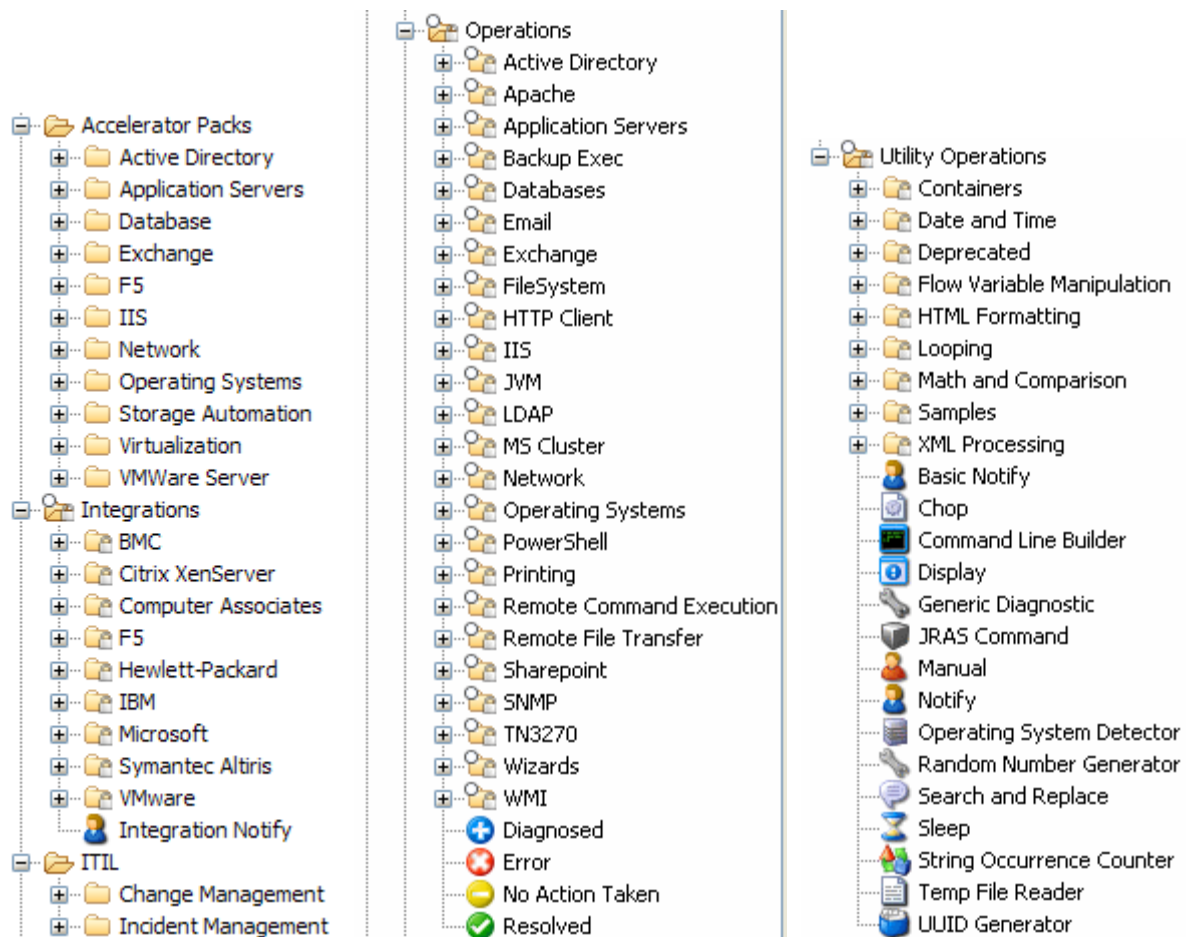


図 9 – デフォルトで用意されているコンテンツ（フローおよびオペレーション）とそのフォルダー

上の図では、デフォルトコンテンツが格納されているフォルダーがいずれも第 1 レベルまで展開されています。コンテンツは多数あり、必要なフローおよびオペレーションの選び方や使用方法を重点的に取り上げたドキュメントセットやチュートリアルもあります。以下に示すのは、最上位のフォルダーとその説明です。必要なフローやオペレーションを選択する参考にしてください。

- **Accelerator Packs**

これらのフローは、大部分のネットワークに次の機能を提供する目的で用意されたものです。

- 複雑なヘルスチェック、問題選別、診断、または修復フロー。
- 1 つ以上のデータを収集してユーザーに表示したり、簡単なアラート通知、一部データの収集、チケットへのデータの記録を行う単純なフロー。

Accelerator Pack の最上位にあるフローの多くは、詳細なヘルスチェック、問題選別、診断、修復機能を備えています。

- **Integrations**

OO とサードパーティ製のシステム管理製品との間でやりとりできるようにするためのオペレーションおよびフロー。

- **ITIL**

このフォルダーには、ITIL の仕様に準拠する他のエンタープライズレベルソフトウェアとの統合を自動化するフローが格納されています。

- **Operations**

このフォルダーには、一般的なテクノロジーで動作する汎用のオペレーションが格納されています。これらのオペレーションはシールされており、HP OO Central (Central) をインストールした後に変更することはできません。




**Operations** フォルダー内のオペレーションは変更できないため、入力用の固定値は設定されていません。入力はすべて、ユーザーによる入力となるか、割り当てないかのいずれかです。ただし、WMI コマンドを使用するような汎用性の高いオペレーションは、この規則の例外となります。

**Operations** フォルダーおよびそのサブフォルダー内のフローは、サブフローまたはオペレーションとして使用されることを前提としています。親フローとして実行するのに適したフローは、Accelerator Packs 内にあります。

- **Utility Operations**

このフォルダーには、データの収集と表示、単純なコマンドラインオペレーションの置換、データの加工と分析、フローの構造化、その他テクノロジーに依存しない機能などを実行するオペレーションやサブフローが格納されています。

次の表は、Library 内で見られる一部のアイコンについて説明したものです。

アイコン	説明
	フローであることを示す。
	白い円は、このフォルダーおよびその中のすべてのフローが Central ユーザーに対して非表示である、つまり Central では表示されないことを表す (Central ではオペレーションは表示されない)。フォルダー内のフローは個別に非表示にできる。 ロックは、このフォルダーおよびその中のすべてのフローとオペレーションがシールされていることを表す。つまりこのフォルダー内にあるフローやオペレーションを編集することはできない。ただし、フローやオペレーションをコピーし、そのコピーに変更を加えることは可能。
	この警告マークは、未完成または無効のフローまたはオペレーションを表すアイコンに重なって表示される。

このように、Library では、オペレーションを表すアイコンによってそれぞれのタイプがわかります。

## [My Changes/Checkouts] ペイン

[My Changes/Checkouts] ペインはリポジトリペインの下にあり、チェックアウト中のライブラリオブジェクトを確認できます。チェックアウトされているオブジェクトは太字で [My Changes/Checkout] ペインに表示されます。

## 作成ペイン

作成ペインは Studio の右側にある広いエリアで、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接続を追加したり、フローやその各部の動作を決定するプロパティを設定したりします。

### 作成ペインのツールバー

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。次の図は、ツールバーの各ボタンの機能を示したものです。一部のボタンについては、図の下に詳しい説明があります。



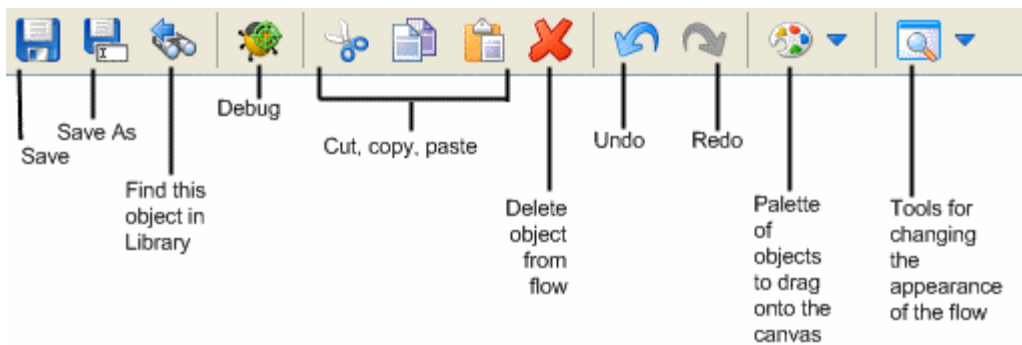





図 10 - フロー作成ツールバー



Studio で Central パブリックリポジトリを開いている場合は、ツールバーの左端 ([Save] アイコンの隣) に

[checkin/checkout] アイコン (  ) も表示されます。[checkin/checkout] アイコンの詳細については、「[ライブラリオブジェクトのバージョン管理](#)」を参照してください。

一部のツールバーアイコンは 2 種類のパレットに分類されています。これらのパレットは、それぞれの下向き矢印をクリックすると開きます。

- [Step Palette] アイコン (  )
- [View Options] アイコン (  )

その他のツールバーアイコンはよくあるアイコンですが、次の例外もあります。

- **Select in repository tree** (  ) : 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を展開します。
- **Debug Flow** (  ) : デバッガーを開き、現在のフローの実行を開始します。

## ステップパレット

ステップパレット (下の図を参照) には、4 種類のリターンステップ (**Success** (成功)、**Diagnosed** (診断済み)、**No Action Taken** (アクションなし)、**Failure** (失敗))、並列分岐ステップ、およびコールアウトのアイコンがあり、フローにドラッグできます。

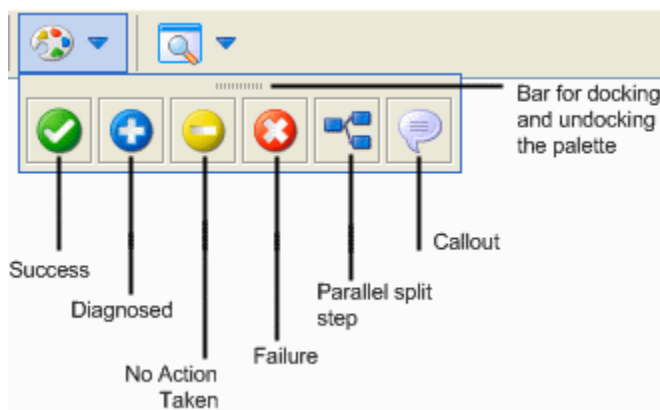


図 11 - ステップパレット

## オプション表示パレット

オプション表示パレット（下の図を参照）には、作成キャンバス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。

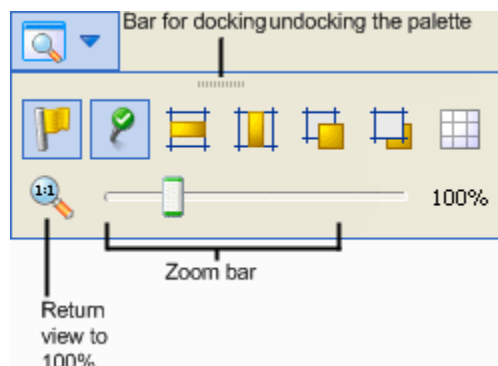


図 12 - オプション表示パレット

このパレットには次のアイコンがあります。



**Show/Hide Labels:** オブジェクト（レスポンスなど）のラベルを表示または非表示にします。



**Show/Hide Connected Response Icons:** レスポンスの各入力にフローのリターンステップと同様のアイコンを表示または非表示にします。



**Align selection horizontally:** 選択したステップを横に並べます。



**Align selection vertically:** 選択したステップを縦に並べます。



**Bring to Front:** 選択したオブジェクトを前面に移動します。



**Send to Back:** 選択したオブジェクトを背面に移動します。



**Show/Hide Grid:** ステップを配置するときに利用できる作成キャンバスのグリッドを表示します。グリッドを表示しているときは、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配置されます。

フロー図は、フローに対する作業のほとんどを直接行うことができる場です。



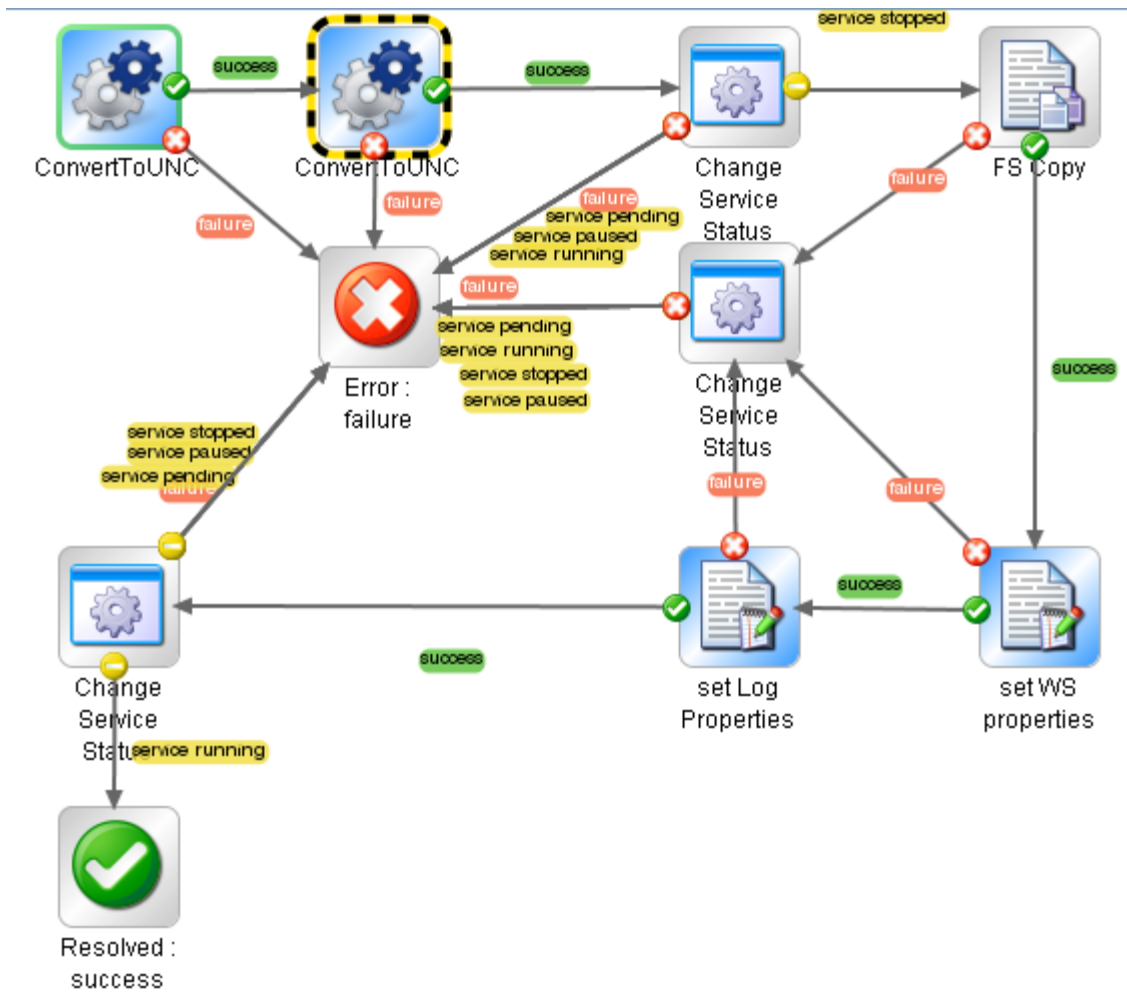


図 13 - フロー図

「[フローの主な構成要素](#)」で解説したフロー図の要素に加えて、上の図では次の要素が使用されています。

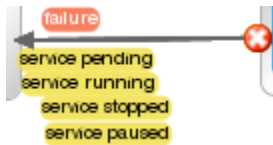
- フローの開始ステップは緑色の枠で囲まれます。



- デバッグ用のブレイクポイントが作成されているステップは、上の図のように黄色と黒の枠で囲まれます。デバッガーでフローを自動的に実行させると、フローは設計した各ブレイクポイントで停止します。



- 複数のトランジションが同じソースステップから開始して同一のステップに遷移する場合、図の上では次のように 1 本の線で表されます。ソースであるレスポンスの名前は積み重ねて表示されます。トランジションのレスポンス名をクリックすると、トランジションを移動、またはその Inspector を開くことができます。



## [Bookmarks] ペイン

[Bookmarks] ペイン（ウィンドウの右上にある [Bookmarks] タブをクリックすると開きます）では、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 Library からドラッグしてリストに追加することが可能です。詳細については、「[ブックマーク: フローおよびオペレーションの追加と削除](#)」を参照してください。

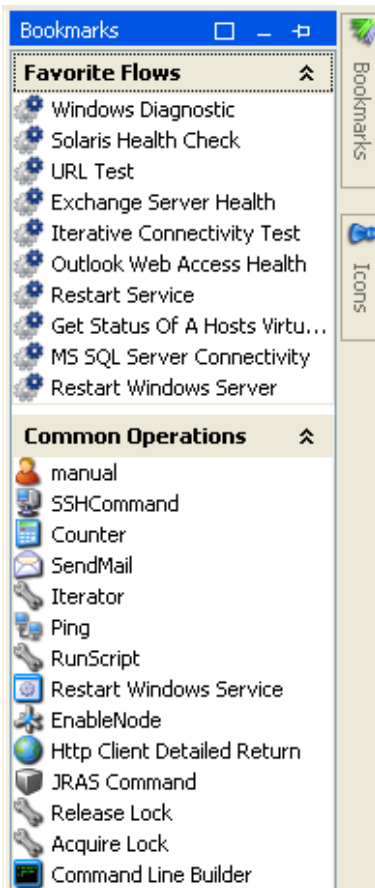


図 14 - [Bookmarks] ペイン

## [Icons] ペイン

[Icons] ペイン（ウィンドウの右上にある [Icons] タブをクリックして開きます）にはオペレーションのアイコンが格納された各種ライブラリがあります。これらを利用することにより、ステップのオペレーションが何をするかをより速くはっきりさせます。フロー、ステップ、オペレーション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。

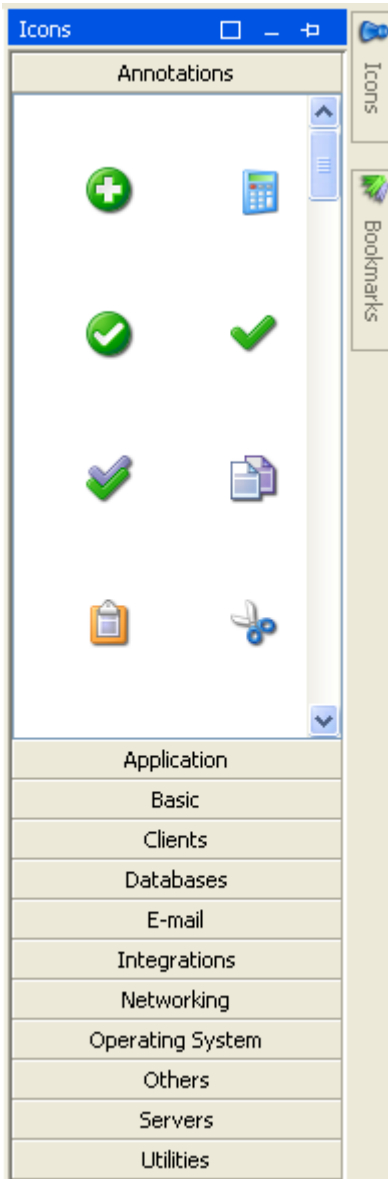


図 15 – フロー内で使用できるアイコン

## オブジェクトの [Properties] シート

フロー、オペレーション、**Configuration** フォルダ内のオブジェクトの [Properties] 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。Library に格納されている大部分のオブジェクトについては、[Properties] 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。ただし、フローの場合は、[Properties] シート以外に Design ビューも使用できます。

[Properties] シートでは編集ができるだけでなく、次の図のように [Properties] シートヘッダーに 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。

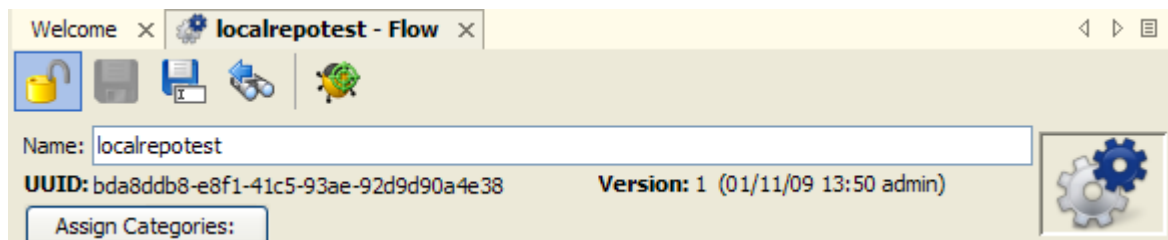


図 16 - [Properties] シートのヘッダー

## Studio のキーボードショートカット

以下のセクションでは、Studio で使用できるキーボードショートカットの一部について説明します。

[リポジトリペインのキーボードショートカット](#)

[作成ペインのキーボードショートカット](#)

[プロパティエディター/Inspector のキーボードショートカット](#)

[スクリプトレットペインのキーボードショートカット](#)

[\[Bookmarks\] ペインのキーボードショートカット](#)

[デバッガーのキーボードショートカット](#)

### リポジトリペインのキーボードショートカット

操作	キーボードショートカット
選択したオブジェクトのすべての下位オブジェクトを展開する	SHIFT + SPACE
選択したオブジェクトのエディターを開く	ENTER
選択したオブジェクトを削除する	DEL

## 作成ペインのキーボードショートカット

操作	キーボードショートカット
選択したステップの削除	DEL
切り取り	CTRL + x
コピー	CTRL + c
貼り付け	CTRL + v
元に戻す	CTRL + z
やり直し	CTRL + y
コールアウトの挿入	CTRL + t

## プロパティエディター / Inspector のキーボードショートカット

操作	キーボードショートカット
選択した行の編集	CTRL + <右矢印>

## スクリプトレットペインのキーボードショートカット

操作	キーボードショートカット
検索	CTRL + f

## [Bookmarks] ペインのキーボードショートカット

操作	キーボードショートカット
選択したブックマークの削除	DEL

## デバッガーのキーボードショートカット

操作	キーボードショートカット
実行	F11
一時停止	ALT + p
停止	ALT + c
ステップイン	F5

操作	キーボードショートカット
ステップオーバー	F6
ステップアウト	F7
リセット	F12

## ライブラリオブジェクトのバージョン管理

複数の作成者がいるチームでメンバーが同時に Central リポジトリで作業する場合、バージョン管理を行うことによって、それぞれの変更による競合が生じないようにすることができます。OO では、**チェックイン/チェックアウト**機能を使ってバージョン管理を行います。適用対象は、フロー、オペレーション、その他のライブラリオブジェクト（システムアカウントやフィルターなど）です。

オブジェクトを操作するには、そのオブジェクトをチェックアウトする必要があります。チェックアウトしたオブジェクトは自分用の作業スペースで作業できます。他の作成者が参照できるのは、前回チェックインされたときの状態のオブジェクトだけです。変更を保存すると、自分の作業スペース内に保存されます。変更を保存した後、オブジェクトをパブリックリポジトリに再度チェックインすることにより、別の作成者がその新しいバージョンを参照またはチェックアウトできるようになります。

**重要:** Central で複数の作成者が作業できるのは、Central を開発またはテスト用としてインストールした場合に限ります（つまり、本稼働環境にインストールした Central に複数の作成者がアクセスすることはできません）。

パブリックリポジトリに接続しているときは、オブジェクトを保存するごとに**バージョン番号**が 0.1 ずつ増えます。オブジェクトをチェックインすると、バージョン番号は次に大きい整数に切り上げられます。

次の例を考えてみます。グロリアという作成者が FlowA をチェックアウトしたとき、そのバージョンは 2 でした。

- グロリアが FlowA に対する変更を保存すると、バージョン番号は 2 から 2.1 に変わる。
- このとき別の作成者が Studio で FlowA を開くと、FlowA のバージョンは 2 と表示される。
- グロリアが FlowA を再度チェックインすると、バージョン番号は 3 に切り上げられる。他の作成者に対しては最新のバージョン（バージョン 3）が表示され、チェックアウトができるようになる。

**オブジェクトのチェックイン/チェックアウトの状態**は、状況に応じてそれぞれ次のように示されます。

- オブジェクトが自分以外の作成者によってチェックアウトされている場合
  - リポジトリペインのライブラリツリーで、オブジェクトの名前がイタリック体で表示される。
- オブジェクトがチェックアウトされていないか別の作成者によってチェックアウトされている場合
  - フローの場合、作成ペインのキャンバスが白ではなくグレーで表示され、キャンバスを使用できない。
  - すべてのオブジェクトで [Properties] シートが使用できない。
- 自分がオブジェクトをチェックアウトしている場合
  - フローの場合、作成ペインのキャンバスが白で表示され、キャンバスで作業ができる。
  - すべてのオブジェクトで [Properties] シートを使用できる。
  - リポジトリペインのライブラリツリーで、オブジェクトの名前が**太字**で表示される。

管理者（ADMINISTRATOR 機能を持つグループのメンバーであるユーザー）は、ある作成者がチェックアウトしたオブジェクトを（その作成者も管理者の 1 人である場合でも）強制的にチェックインできます。管理者がオブジェクトを強制チェックインすると、そのオブジェクトで作業していた作成者には警告メッセージが表示されます。

この機能の詳細については、以下のサブセクションで説明します。

[ライブラリオブジェクトのチェックインおよびチェックアウト](#)

[バージョンの管理](#)

## ライブラリオブジェクトのチェックアウトおよびチェックイン



パブリックリポジトリに接続して作業するときは、作業対象のフロー、オペレーション、または **Configuration** メニューのオブジェクトをチェックアウトする必要があります。オブジェクトをチェックアウトすると、他の作成者はそのオブジェクトを操作できません。オブジェクトに対する操作は自分の作業スペースで行うので、その操作内容は他の作成者には表示されません。



**鍵となる情報:** 同一ユーザーとして **Studio** の 2 つのインスタンスにログインした場合、**Central** の同一インストールのパブリックリポジトリに接続しないでください。リポジトリ内にあるユーザーの作業スペースが壊れることがあります。また、**Central** がクラスタ化されたときに起こることもあります。

チェックイン/チェックアウトとパブリッシュ/更新の違いは、**Studio** が直接パブリック (**Central**) リポジトリに接続しているかどうか、つまり、事前に **Studio** に追加されているパブリックリポジトリを **Studio** で開いているかどうかです。パブリックリポジトリに直接接続している場合は、ライブラリオブジェクトをチェックアウトしてパブリックリポジトリの作業スペースで作業を行い、再度チェックインすることでパブリックリポジトリに変更を反映させるとともに他の作成者がそのオブジェクトを利用できるようにします。

**Studio** の作成ペインには、オブジェクトのチェックイン/チェックアウト状態を見分けるためのいくつかの手がかりがあります。

- オブジェクトをチェックアウトした作成者の場合
  - オブジェクトのフィールドが編集可能な状態になっている。オブジェクトがフローの場合は、白いキャンバスが表示される。
  - オブジェクトが [My Changes/Checkouts] ペインに表示される。
  - リポジトリペインおよび [My Changes/Checkouts] ペインで、オブジェクト名が**太字**で表示される。
  - ツールバー上の [checkin/checkout] アイコンが、オブジェクトが編集可能であることを示すロック解除アイコン () として表示される。
  - 変更を保存するごとに、オブジェクトの [Properties] シートに表示されるバージョン番号が 0.1 ずつ増える。オブジェクトをチェックインすると、バージョン番号はそのたびに 1 ずつ増える。  
**注:** ローカルリポジトリで作業をしているときはチェックイン/チェックアウト機能はありません。オブジェクトを保存すると、バージョン番号は 1 ずつ増えます。
- オブジェクトをチェックアウトした作成者以外の作成者の場合
  - オブジェクトのフィールドが使用不可の状態、グレー表示されている。オブジェクトがフローの場合は、グレーのキャンバスが表示される。
  - ツールバー上の [checkin/checkout] アイコンが、オブジェクトが編集できないことを示すロックアイコン () として表示される。

チェックイン/チェックアウト機能には、以下の制限および考慮事項があります。

- オブジェクトをチェックアウトするには、そのオブジェクトに対する読み取りおよび書き込み権限が必要です。
- フォルダーはそのコンテンツ（サブフォルダーを含む）とともにチェックアウトできます。ただし、サブフォルダーは別の作成者によってチェックアウトされていない場合に限りです。
- オブジェクトをチェックアウトする場合、チェックアウト状態になるのはそのオブジェクトのみです（上記のようにフォルダーとそのコンテンツをチェックアウトする場合を除く）。当該オブジェクトによって参照されているオブジェクトはチェックアウトされません。例として、システムアカウントを使用するオペレーションが含まれるフローがある場合、そのフローをチェックアウトしても、すでにチェックアウト状態になっているため、オペレーションとシステムアカウントはいずれもチェックアウトされません。
- オブジェクトに対して次のいずれかの操作を行うと、その操作を行った作成者に自動的にチェックアウトされます。これらの変更をパブリックリポジトリに反映させるには、操作したオブジェクトをチェックインする必要があります。
  - 貼り付け
  - 切り取り
  - 名前の変更
  - 移動
  - 削除
  - 名前を付けて保存
  - リポジトリのインポート
- フォルダーの移動や削除、またはフォルダー名の変更を行うには、フォルダー内のすべてのオブジェクトをチェックアウトできる必要があります。つまり、すべてのオブジェクトに対して読み取り権限と書き込み権限が必要です。

このセクションには次のサブトピックがあります。

[ライブラリオブジェクトのチェックアウト](#)

[ライブラリオブジェクトのチェックイン](#)

[2者間で競合するオブジェクトバージョンの解決](#)


[管理者による強制チェックイン](#)

## ライブラリオブジェクトのチェックアウト

**注:** オブジェクトをチェックアウトした作成者が ADMINISTRATOR 機能を持つグループのメンバーでも、管理者によって強制的にそのオブジェクトがチェックインされる場合があります。その場合、チェックアウトが取り消されたことを知らせる警告メッセージが表示されます。

フォルダーはそのサブフォルダーも含め全体をチェックアウトできます。

### ライブラリオブジェクトをチェックアウトするには

- 作成ペインでオブジェクトを開いた状態で、[checkin/checkout] アイコン () をクリックします。あるいは、編集するオブジェクトを開いているかどうかにかかわらず、次の操作を行います。

リポジトリペインの **Library** フォルダーまたは **Configuration** フォルダーで、次の操作を行います。

- a. チェックアウトするオブジェクトを右クリックします。
- b. 表示されたコンテキストメニューで、[Repository] をポイントしてから [Check Out] をクリックします。



## フォルダーとそのコンテンツをチェックアウトするには

1. リポジトリペインで、目的のフォルダーを右クリックします。
2. 表示されたコンテキストメニューで、[Repository] をポイントしてから [Check Out Tree] をクリックします。

## ライブラリオブジェクトのチェックイン

オブジェクトをチェックインすると、個人の作業スペースで行ったそのオブジェクトに対する変更と新しいバージョン状態が、パブリックリポジトリに反映されます。パブリックリポジトリにあるオブジェクトの新しいバージョンは他の作成者に表示され、別のユーザーがチェックアウトできるようになります。


フォルダーをチェックインする場合、そのフォルダーをチェックアウトしたときにその中に含まれていたオブジェクトのみがチェックインされます。たとえば、flow5 と flow6 が格納されているフォルダーがあり、flow5 と flow6 をそれぞれ別のユーザーがチェックアウトしたとします。flow5 をチェックアウトしたユーザーが両方のフローが格納されているフォルダーをチェックインした場合、flow5 のみがチェックインされます。フォルダーのコンテンツ（サブフォルダーとそのコンテンツを含む）を再帰的にチェックインするには、[Check In Tree] コマンドを使用します。

オブジェクトに対して次のいずれかの操作を行うと、その操作を行った作成者に自動的にチェックアウトされます。これらの変更をパブリックリポジトリに反映させるには、操作したオブジェクトをチェックインする必要があります。

- コピー
- 貼り付け
- 切り取り
- 移動
- 削除
- 名前を付けて保存
- リポジトリのインポート

チェックインコメントを利用すると、各バージョンでオブジェクトに加えられた変更を追跡できます。前のバージョンに戻す必要がある場合、回復するバージョンがどれかを判断する際にコメントが役立ちます。

## ライブラリオブジェクトをチェックインするには

1. 作成ペインで編集するオブジェクトを開いた状態で、[checkin] アイコン () をクリックします。あるいは、編集するオブジェクトを開いているかどうかにかかわらず、次の操作を行います。

リポジトリペインの **Library** フォルダーまたは **Configuration** フォルダーで、次の操作を行います。

- a. チェックインするオブジェクトを右クリックします。
- b. 表示されたコンテキストメニューで、[Repository] をポイントしてから [Check In] をクリックします。[Check In] ダイアログボックスが表示されます。

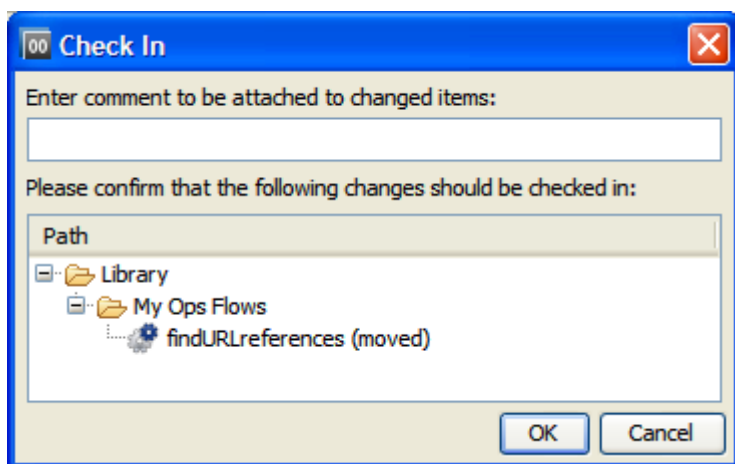


図 17 – [Check In] ダイアログボックス

2. [Enter Check In Comments] ボックスに適切なコメントを入力します。  
通常は、オブジェクトに対して行った変更の説明を入力します。
3. [OK] をクリックします。

#### フォルダーとそのコンテンツをチェックインするには

1. リポジトリペインで、目的のフォルダーを右クリックします。
2. 表示されたメニューで、[Repository] をポイントしてから [Check In Tree] をクリックします。

## 2 者間で競合するオブジェクトバージョンの解決

2 人の作成者がチェックインしたオブジェクトにバージョンの競合がある場合は、いずれかの作成者が Studio の [My Changes/Checkouts] パネルで競合を解決できます。

たとえば、エレインというユーザーが「foo」という名前のオペレーションを作成してチェックインしたとします。エレインは後でその名前を「safe call」に変更しましたが、チェックインはしませんでした。その後、同じフォルダーで作業していたデスモンドというユーザーが、「safe call」という別のオペレーションを作成しました。デスモンドが自分の「safe call」オペレーションをチェックインすると、エレインの画面では [My Changes/Checkouts] パネルに表示された同名のオペレーションのアイコン上に、競合を示すアイコン（赤い両向き矢印）が表示されます。

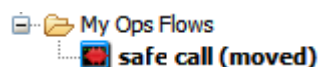


図 18 – 別の作成者のオブジェクトと競合するオブジェクト

このようなときは、両方のバージョンの [Properties] シートを確認して競合を解決できます。

#### オブジェクトの 2 つのバージョンの競合を解決するには

1. [My Changes/Checkouts] パネルで競合しているオブジェクトを右クリックし、[Resolve Conflict] をクリックします。  
オブジェクトの 2 つの競合バージョンが Studio の作成ペインに表示されます。

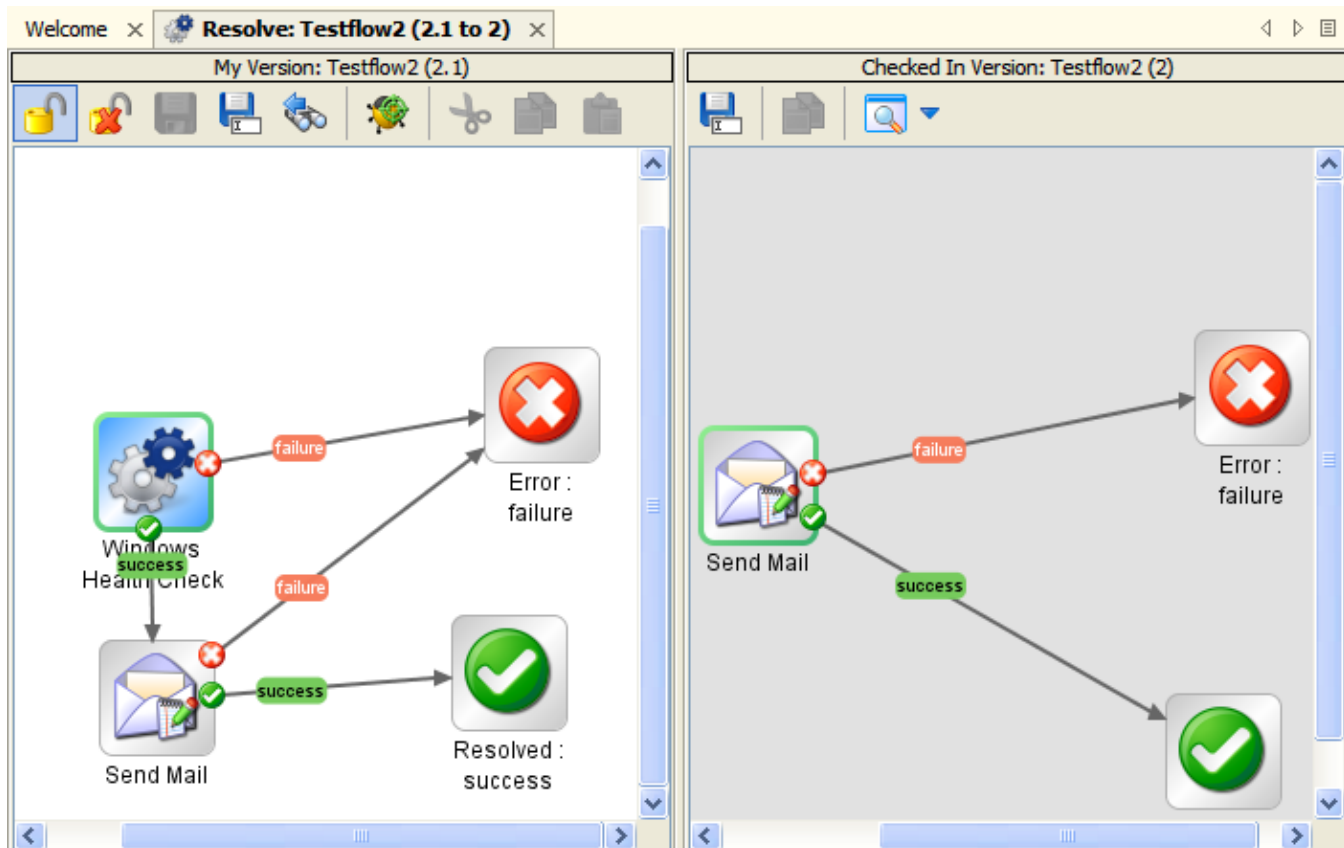




図 19 – フローの競合バージョンの解決

ソースバージョンのフロー図では、変更を加えて保存できます。ターゲットバージョン（チェックインされているバージョン）のフロー図では、[Save As]（オブジェクトを別名で保存）を実行できます。

2. 自分が作成したバージョンでは、オブジェクトのプロパティを変更できます。
3. 自分の変更を破棄して競合を解決する場合は、[Abandon Changes] アイコン（）をクリックします。  
もしくは

自分のオブジェクトバージョンをチェックインする場合は、[Checkin] アイコン（）をクリックします。

### 管理者による強制チェックイン

管理者は、別の作成者がチェックアウトしたオブジェクトを（当該作成者も ADMINISTRATOR 機能を持っている場合でも）強制的にチェックインできます。管理者がオブジェクトを強制チェックインすると、作業中の作成者がそのオブジェクトを閉じる前に別名で保存していた場合を除き、最終保存時以降に行われたオブジェクトに対する作業は無効になります。

作業中のオブジェクトが強制チェックインされる場合に備えて変更を保存しておくには、[Save As] コマンドを使用してオブジェクトに新しい名前を付けて保存します。その結果、オブジェクトには新しい UUID が割り当てられます。

## 別のユーザーが現在チェックアウトしているオブジェクトを強制チェックインするには

1. ADMINISTRATOR 機能を持つユーザーとして Studio にログインします。
2. 強制チェックインを実行するオブジェクトを右クリックし、[Repository] をポイントして [Force Check In] をクリックします。

## バージョンの管理

オブジェクトの前のバージョンを開いて現在のバージョンと比較できます（手順の詳細については、「[オブジェクトの前のバージョンを開く](#)」を参照してください）。バージョン管理では、目的に応じて以下のような操作が可能です。

- チェックアウト中にオブジェクトに加えた保存済みの変更を破棄する。  
変更を破棄することにより、オブジェクトが最後にチェックアウトされてから加えられた変更が無効になります。オブジェクトが再度チェックインされると元のチェックアウトはなくなります。オブジェクトを前の状態に戻す場合は、次のいずれかの操作を行います。
- オブジェクトの前のバージョンを開いて、別名で保存する。
- オブジェクトを前のバージョンに復元する。

上記の手順については以下のサブセクションで説明します。

### 変更の破棄

#### オブジェクトの前のバージョンを開く

#### 前のバージョンへのオブジェクトの復元

## 変更の破棄

あるオブジェクトにいくつか変更を加えて保存をした場合でも、その変更を保持する必要がないことがあります。このような場合は変更を破棄することが可能です。変更を破棄すると、オブジェクトが自動的にパブリックリポジトリにチェックインされます。チェックアウト中に加えた変更を破棄してオブジェクトにチェックインされることにより、パブリックリポジトリ内のオブジェクトはチェックイン状態に戻り、かつチェックアウトされる前の状態に戻ります。また、他の作成者がそのオブジェクトをチェックアウトできるようになります。

### 変更を破棄するには

1. リポジトリペインで、変更を保持する必要がないオブジェクトを右クリックします。
2. 表示されたコンテキストメニューで、[Repository] をポイントしてから [Abandon Change(s)] をクリックします。[Confirm] ダイアログボックスが表示されます。

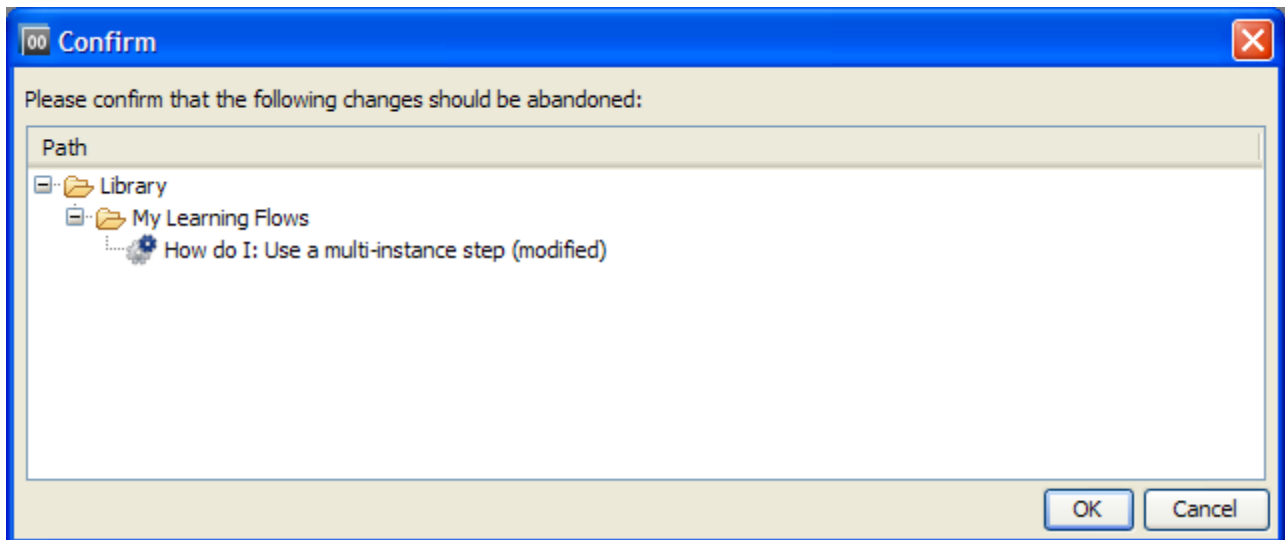


図 20 – [Confirm] ダイアログボックス

3. 変更を保持しないでチェックインしようとしているオブジェクトがダイアログボックスに表示されていることを確認し、[OK] をクリックします。

## オブジェクトの前のバージョンを開く

オブジェクトの前のバージョンを開いて表示したときに、そのバージョンに現在のバージョンと違う名前を付けて保存することが可能です。オブジェクトを前のバージョンに直接復元する手順については、「[前のバージョンへのオブジェクトの復元](#)」を参照してください。

### オブジェクトの前のバージョンを開くには

1. リポジトリペインで、前のバージョンを表示するオブジェクトを右クリックします。
2. 表示されたコンテキストメニューで、[Repository] をポイントしてから [Show History] をクリックします。

[Version History] ダイアログボックスが表示されます。

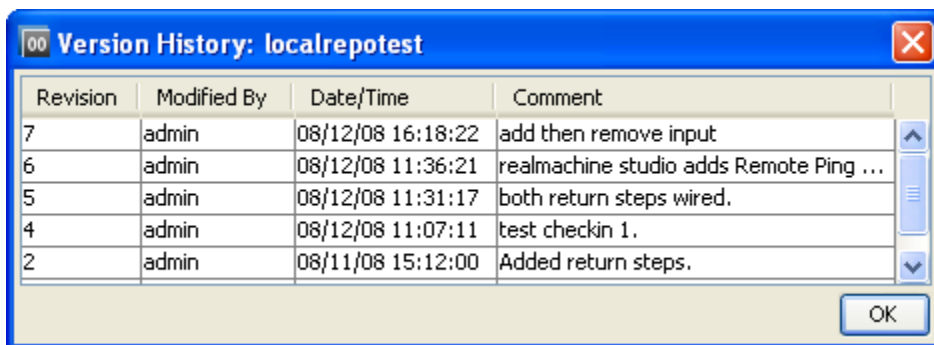



図 21 – バージョンの選択

3. バージョンの一覧で、必要なバージョンを右クリックします。  
または、必要なバージョンがない場合は [OK] をクリックします。

開いたバージョンが作業スペースに表示されます。このバージョンが Studio に表示されます。

この手順では、同じオブジェクト名でこのバージョンを保存することはできません（前のバージョンのオブジェクトを復元する手順については、「[前のバージョンへのオブジェクトの復元](#)」を参照してください）。

- 開いたバージョンを保存するには、[Save As] アイコン (  ) をクリックし、一意の名前を付けます。  
これで、この前のバージョンは新しいオブジェクトになります。
- この新しいオブジェクトをチェックインします。

## 前のバージョンへのオブジェクトの復元

この手順を実行すると、オブジェクトを前のバージョンに復元できます。オブジェクトの現在のバージョンと前のバージョンの両方を保持する手順については、「[オブジェクトの前のバージョンを開く](#)」を参照してください。

### オブジェクトを前のバージョンに復元するには

- リポジトリペインで、前のバージョンに復元するオブジェクトを右クリックします。
- 表示されたコンテキストメニューで、[Repository] をポイントしてから [Show History] をクリックします。  
[Version History] ダイアログボックスが表示されます。

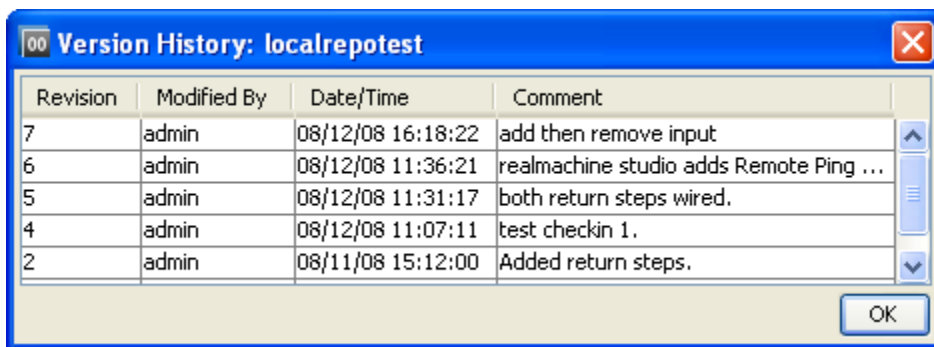


図 22 - バージョンの選択

- バージョンの一覧で、必要なバージョンを右クリックします。  
開いたバージョンが作業スペースに表示されます。このバージョンが Studio に表示されます。
- オブジェクトを保存してチェックインします。

## フォルダーの作成

フローを新規作成したか編集用としてコピーしたかにかかわらず、そのフローを格納する新しいフォルダーを必要に応じて作成できます。

### フォルダーを作成するには

- リポジトリペインで **Library** フォルダーを右クリックし、[New] をポイントしてから [Folder] をクリックします。
- 表示されたダイアログボックスで、テキストボックスに新しいフォルダーの名前を入力し、[OK] をクリックします。

#### 注:

- Studio では、名前の大文字と小文字は区別されません。
- 名前の長さは最大 128 文字までです。

## フローの作成

コンピューターとサーバー間のネットワーク接続をチェックする例を考えてみます。以下のトピックでは、サーバーに対して ping を実行し、コンピューターとサーバー間のネットワークで traceroute コマンドを実行するフローの作成および編集方法について説明します。

ユーザープロンプトからのフローの入力を特定の値に変更し、フロー全体を自動的に実行できるようにします。この作業は、通常次のような手順で行います。

1. フローおよびそのオペレーションを格納するフォルダーを作成します。
2. 次のいずれかの方法でフローの作成を開始します。
  - 使用するフローを検索し、そのコピーを作成する。詳細については、「[フローまたはオペレーションの検索](#)」を参照してください。  
必要なフローを取得する際、場合によってはリポジトリをインポートする必要があります。この例ではその必要はありませんが、リポジトリをインポートする手順については、「[リポジトリのインポート](#)」を参照してください。
  - テンプレートから新しいフローを開始する（「[テンプレートを使用したフローの作成](#)」を参照してください）。
3. 必要に応じて、入力に値を割り当てる方法を変更します。
4. 入力に対するレポートデータ（フローの状況を表す識別子）を記録します。
5. 各オペレーションのフィルターや結果を見て、オペレーションから必要なデータを取得できているかどうか確認します。
6. フロー変数がさらに必要な場合は作成します。
7. フロー、ステップ、オペレーション、トランジションの説明が Central ユーザーにとって実用的かどうかを確認します。



**ベストプラクティス:** Central ユーザー、およびオペレーションを使用してフローを作成する他の作成者にとってわかりやすくするためには、オペレーションの [Description] タブに次の情報を追加します。（同じテクノロジーを用いて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 [Description] タブにはこの情報を記載します。HP OO コンテンツにデフォルトで設定されたベストプラクティスです）。[Description] タブへの情報の追加に

より、他の作成者や Central ユーザーが Generate Documentation 機能を利用できるようにすることにご注意ください。Generate Documentation の詳細については、「[複数のオペレーションやフローの説明の表示](#)」を参照してください。

- 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ユーザーや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。
- 実装を追加するときの**注意点**として、次のようなものがあります。
  - 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。フローを実行するために RAS が必要な場合は特に重要（RAS オペレーションではフローの作成者やユーザーにこの情報が表示されない可能性があるため）。
  - その他の環境要件または使用要件。



フローの説明用としてコールアウトを使用することも可能です。詳細については、「[フローへのコールアウトの追加](#)」を参照してください。

8. フローをテストします。
9. Central ユーザーがフローを使用できるようにするには、リポジトリをパブリッシュします。

## フローの作成

フロー内で使用するオペレーションを作成またはインポートした後にフローを作成する際の主な手順は次のとおりです。

1. フローおよびそのオペレーションを格納するフォルダーを作成します。詳細については、「[フォルダーの作成](#)」を参照してください。
2. このセクションで説明する手順に従ってフローを作成します。
3. ステップとなるオペレーションを追加します。
4. ステップの各レスポンスからフロー内の別のステップに進めるように、ステップ間の接続（または「トランジション」）を作成します。
5. フローを終了するためのリターンステップを1つ以上作成し、それぞれのリターンステップにフローレスポンスを割り当てます。
6. フローを保存します。
7. フローのデバッグを実行します。

Studio でフローをデバッグする方法については、「[フローのデバッグ](#)」を参照してください。



**ヒント:** ステップの理解を助けるフローとして **Restart Service — Tutorial Flow** が参考になります。

**注:** 空のフローを作成する場合は、フォルダーを右クリックして、[New] をポイントしてから [Flow] を選択します。

### フローを作成するには

1. フローを作成するフォルダーを強調表示します。
2. [File] メニューから、[New] をポイントして [Flow] をクリックします。
3. 表示されたダイアログボックスで、通常の文字を使用してフローの名前を入力し、[OK] をクリックします。

**注:**

- 同一フォルダー内にフローが2つ存在する場合、同じ名前を付けることはできません。
- Studio では、名前の大文字と小文字は区別されません。
- 名前の長さは最大 128 文字までです。

新しいフロー図が**作成**ペインに表示されます。

頻繁に実行されるタスクの処理に特化したテンプレートの1つを使ってフローを作成することもできます。これらのテンプレートには共通のタスクを完了するために必要なステップが用意されているので、フローの作成をすぐに開始できます。テンプレートを基にしてフローを作成する方法については、「[テンプレートを使用したフローの作成](#)」を参照してください。

新しいフローが未完成の場合、Library ではそのフローの名前が赤で表示されます。未完成のフローの名前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。



## 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、フローでの使用頻度が高い特定のタスクを実行するステップ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テンプレート (**Restart Service**) を使用すれば、そのような処理を行うフローを作成できます。

また、サーバーに対して ping を実行し、コンピューターとサーバー間のネットワークをチェックするフロー用のテンプレート (**Network Check**) もあります。

あるいは、Success および Error のリターンステップ（フロー最後のステップであることが多い）のみで開始する必要がある場合には、**Blank Flow** テンプレートを利用できます。

### テンプレートからフローを作成するには

1. 作成ペインで、[Welcome] タブをクリックして Studio の [Welcome] ページを開き、[New Flow] アイコンをクリックします。

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、その説明が表示されます。

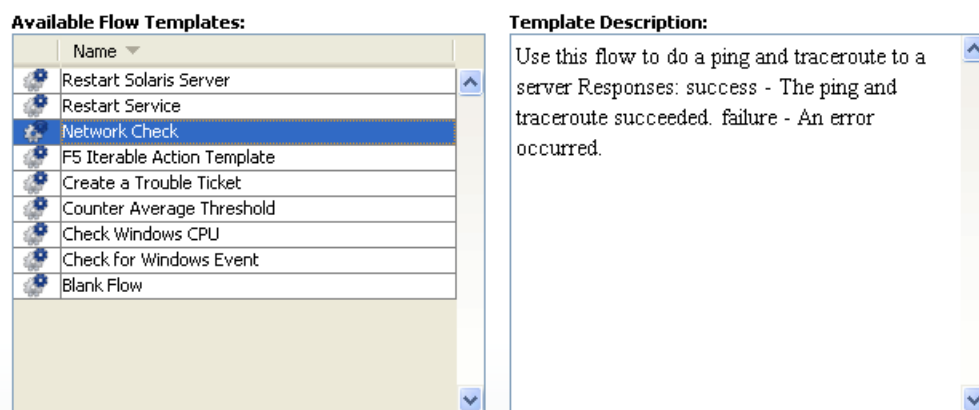


図 23 – 使用頻度の高いフローのテンプレート

2. 目的に合ったフローテンプレートを選択し、[Create] をクリックします。

作成ペインにフローが表示されます。

次は、オペレーションを作成するか、あるいはオペレーション（またはフロー）を基にしてフローのステップを作成します。



**ベストプラクティス:** フローを実行する Central ユーザーや、フローをサブフローとして利用し別のフローを作成する他の作成者にとってわかりやすくするためには、フローの [Description] タブに次の情報を追加します。（同じテクノロジーを用いて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 [Description] タブにはこの情報を記載します。HP OO コンテンツにデフォルトで設定されたベストプラクティスです）。[Description] タブへの情報の追加により、他の作成者や Central ユーザーが Generate Documentation 機能を利用できるようになることにご注意ください。Generate Documentation の詳細については、「[複数のオペレーションやフローの説明の表示](#)」を参照してください。

- フローの内容の説明。
- フローが必要とする入力。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。

- 実装を追加するときの**注意点**として、次のようなものがあります。
  - 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。フローを実行するために RAS が必要な場合は特に重要（RAS オペレーションではフローの作成者やユーザーにこの情報が表示されない可能性があるため）。
  - その他の環境要件または使用要件。

コールアウトを利用すると、他のフロー作成者や Central ユーザーにとってのフローの使い勝手が格段に向上します。他の作成者やユーザー向けにコールアウトに入力する内容は作成者が自由に決められますが、たとえば次のような情報を追加すると効果的です（これらは、Studio リポジトリの How Do I flows 内で使用されているコールアウトの一部です）。

- 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
- 入力に必要なデータに要求される形式

## フローへのコールアウトの追加

### フローにコールアウトを追加するには

1. 次のいずれかを実行します。
  - ステップとコールアウトのパレットで、[Callouts] アイコンをクリックしてフロー図にドラッグする。

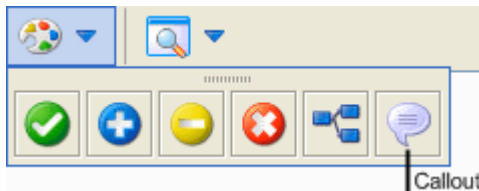


図 24 – ステップとコールアウトのパレット、およびコールアウトアイコン

- フロー図内で右クリックし、[Insert Callout] をクリックする。フロー図にコールアウトが表示されます。



図 25 – 追加されたコールアウト（テキスト入力前）

2. コールアウトのテキストを入力します。
3. 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。

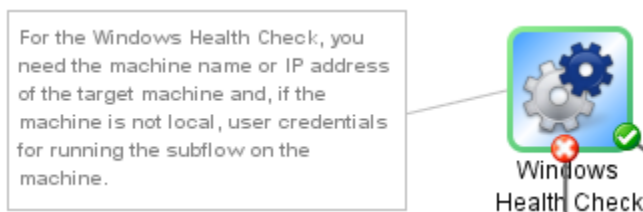


図 26 – テキストが入力されステップに接続されたコールアウト

4. 作業結果を保存します。

# 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は、Studio のリポジトリペインの **Library** フォルダのサブフォルダに格納されています。Library の主なフォルダは次のとおりです。

- **Accelerator Packs**


最もよく使用されるフローがテクノロジー別のサブフォルダにまとめられています。

- **My Ops Flows**

自分が作成したフローを格納するフォルダ。テンプレートから新規作成したフローは自動的にここに格納されます（テンプレートからフローを作成する方法については、「[フローの作成](#)」を参照してください）。

- **Operations**

このフォルダには、Accelerator Packs や自分で作成するフローの構成要素となる、事前設定済みの基本的なオペレーションやサブフローが格納されています。この中のフォルダの多くには **Samples** というフォルダがあり、こうしたオペレーションを使用する例として参考にできます。

**Accelerator Packs、Integrations、Operations、Utility Operations** の各フォルダはシールされています（ロック付きのフォルダアイコン（）で示されます）。つまり、シールされた（読み取り専用の）フローやオペレーションが格納されています。シールされているフローやオペレーションは実用性が高いだけでなく、ユーザーが編集できないという点で根本的に重要です。シールされたフローやオペレーションは、コピーを作成してからそのコピーを編集することが可能です。



**ヒント:** 使用頻度の高いフローやオペレーションにすぐにアクセスできるようにするには、そのフローまたはオペレーションを [Bookmarks] ペインにドラッグします。

フローやオペレーションを検索する際は、Library から検索するかわりに [Search] タブを使用すると便利です。Studio の検索エンジンは Apache Lucene の構文を使用しています。[Search] タブの使用方法、および Studio での Apache Lucene 構文の詳細については、「[フローまたはオペレーションの検索方法](#)」を参照してください。

**注:** ローカルの Studio リポジトリ、リモートの Central リポジトリのどちらにもないリポジトリに必要なフローが格納されている場合は、フローとその従属オブジェクト（オペレーション、システムアカウント、フローが使用するその他のシステムオブジェクト）の場所に応じて次のいずれかを実行することにより、目的のフローを Studio に取り込むことができます。

- Central リポジトリからローカルの Studio リポジトリを更新する
- リモートリポジトリをローカルの Studio リポジトリにインポートする

このセクションのトピック

[フローまたはオペレーションの検索方法](#)

[説明: 適切なオペレーションの判断](#)

## フローまたはオペレーションの検索方法

検索は、Apache Lucene 検索構文を使用した全文検索で、Library 全体が実行対象です。Apache Lucene 構文を使用した検索システムの構築については、Apache Software Foundation の Web サイトを参照してください。

Studio の画面左下にある [Search] タブをクリックする（またはキーボードの F3 を押す）と、[Search] ペインが開きます。ここで、フローやオペレーションの名前またはその他のフィールドプロパティを指定することにより、検索を実行できます。以下にいくつかの例を挙げて説明します。



図 27 - Studio のタブ

[Search] 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ピンのアイコン (📌) をクリックします。

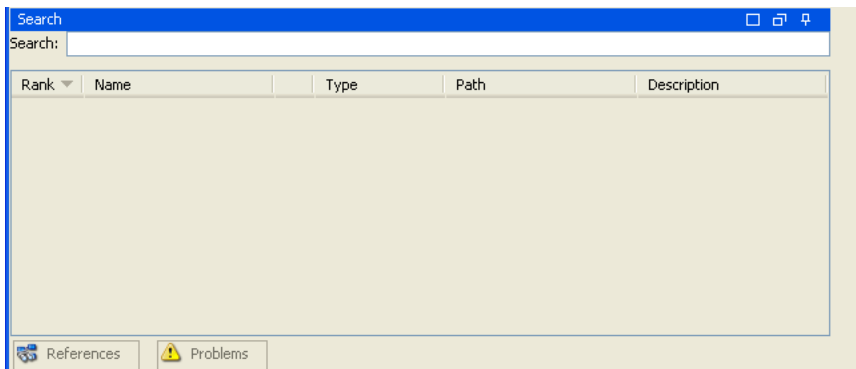


図 28 - [Search] ペイン

ほとんどの場合、適切な検索語を入力してキーボードの ENTER キーを押すだけで検索を実行できます。たとえば **Restart Service** フローを検索するには、[Search] テキストボックスに [Restart Service] または単に [service] と入力します。検索結果は関連性の高いものから順に表示されます。列見出しをクリックしてその列を基準に並べ替えることも可能です。[Type] フィールドは、検索結果がオペレーションなのかフローなのかを判断する上で役立ちます。

Rank	Name	Type	Path	Description
*****	Determine Results	other	/Library/iConclude/HTTPClient/Samples	Checks the outcome of the HttpGetData...
****	Http Client Get	WebExtension	/Library/iConclude/HTTPClient	<pre>Service to perform an HTTP GET ...
***	Check URL via Http Client Get	Flow	/Library/iConclude/HTTPClient/Samples	This sample flow uses the HttpClientDet...
***	Refresh	WebExtension	/Library/Integrations/SiteScope	<pre>Issues a refresh command to a Si...
***	Acknowledge	WebExtension	/Library/Integrations/SiteScope	<pre>Acknowledges a SiteScope monito...
**	Unacknowledge	WebExtension	/Library/Integrations/SiteScope	<pre>Clears an acknowledgement of a ...
**	Total Server Shutdown	WebExtension	/Library/iConclude/Application Servers/B...	<pre>Shuts down a server and waits fo...
**	Suspend Server	WebExtension	/Library/iConclude/Application Servers/B...	<pre>Suspends a server and waits for ...
**	Start Server	WebExtension	/Library/iConclude/Application Servers/B...	<pre>starts a server and waits for it to ...
**	Validate XML Document	WebExtension	/Library/iConclude/XML Processing	<pre>Service to validate an XML docum...

図 29 - 検索結果

フローまたはオペレーションの説明を読み、最適な選択肢かどうかの判断材料にすることもできます。

### 検索結果の一覧から直接オペレーションまたはフローを使用するには

- 目的のオペレーションまたはフローの行をダブルクリックして、オペレーションの [Properties] シートまたはフロー図を開きます。  
もしくは  
オペレーションをフロー図にドラッグします。

### 特定のフィールドのみを対象にして検索するには

- 次の検索構文を使用します。  
<検索可能なフィールド名>:<検索対象の文字列>



**ヒント:** この検索はブール AND を使用します。つまり、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。

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：

```
categories:network
```

- 入力の名前

例：

```
inputs:server
```

- フローまたはオペレーションの ID

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
```

- オペレーションまたはフローが実行される RAS

例：

```
ras:RAS_Operator_Path
```

ヒント：

- 目的に合うオペレーションまたはフローを見つけた場合、その使用方法の詳細を調べることができます。該当するオペレーションまたはフローを Library で探し、右クリックして [References] をポイントしてから [What uses this?] または [What does this use?] をクリックします。詳細については、「[特定のオペレーションを使用するフローの検索](#)」を参照してください。
- ステップを作成する際のベースとなったオペレーションまたはフローを検索するときは、ステップの Inspector の [Advanced] タブを事前に調べておくとも時間を節約できます。[Advanced] タブには、ステップが関連しているオペレーションやフローとその場所が表示されるためです（ステップを右クリックして [Open Operation] をクリックすることにより、ステップのオペレーションを開くこともできます）。

## 説明: 適切なオペレーションの判断

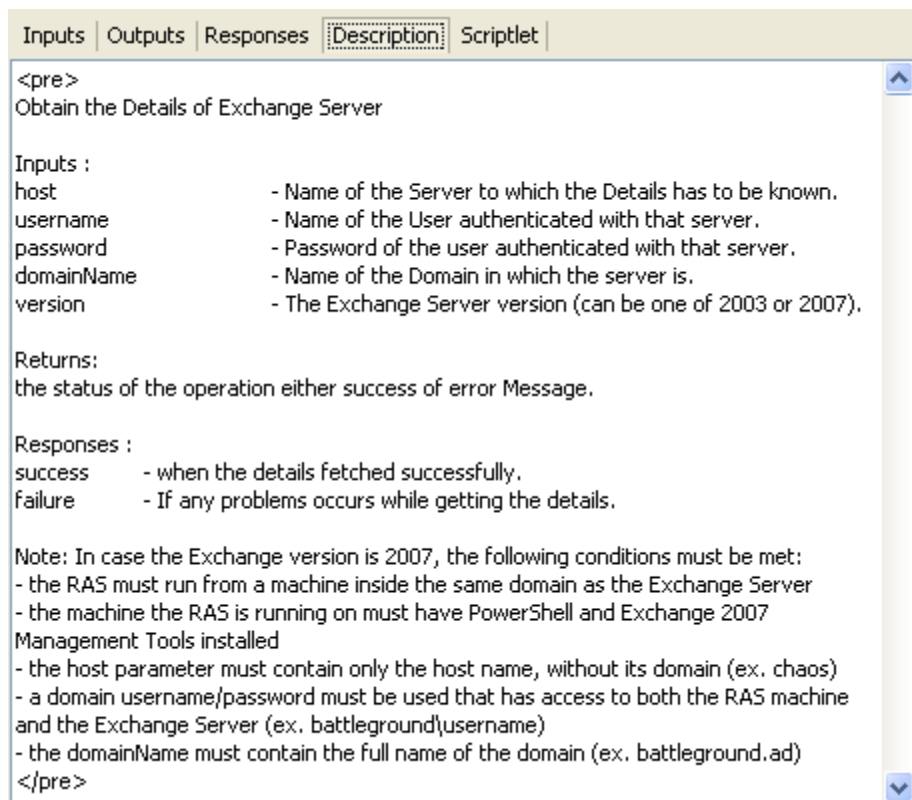
サーバーとの接続をテストする場合について考えてみます。[Search] タブで検索語「connectivity」を使って検索を実行した結果、**Connectivity Test** と **Iterative Connectivity Test** というフローが見つかりました。どちらが適切なフローか、それぞれの説明を調べることで判断できます。

同様に、オペレーションの説明や、デフォルトの OO コンテンツ内のオペレーションおよびフロー（Studio で提供されるオペレーションおよびフローのコレクション。Studio のリポジトリパネルで、**Accelerator Packs**、**Integrations**、**Operations**、**Utility Operations** の各フォルダー内にある）が格納されているフォルダーの説明についても調べることができます。

デフォルトの OO コンテンツ内にあるフォルダーの説明には、通常次のような情報が含まれています。フォルダー内のフローやオペレーションを有効に利用するためのヒントとなるのがこれらの情報です。

- フローまたはオペレーションの内容の説明。
- フローまたは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。
- 実装を追加するときの**注意点**として、次のようなものがあります。
  - 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。
  - フローまたはオペレーションが相互作用するアプリケーションまたは Web サービス API。フローまたはオペレーションを実行するために RAS が必要な場合は特に重要（RAS オペレーションではフローの作成者やユーザーにこの情報が表示されない可能性があるため）。
  - その他の環境要件または使用要件。

オペレーションについてさらに詳しく調べる場合は、目的のオペレーションのステップを強調表示し、ステップの Inspector を開いて、[Description] タブをクリックします。ステップはオペレーションを基にして作成されているので、ステップの説明はオペレーションの説明にもなります。この説明から、オペレーション（およびステップ）の入力、オペレーションのレスポンスや結果などがわかります。また、次に示す **Exchange Server Info** オペレーションの説明にあるように、制限事項に関するメモや使用方法についてのヒントが書かれている場合があります。



```
<pre>
Obtain the Details of Exchange Server

Inputs :
host           - Name of the Server to which the Details has to be known.
username       - Name of the User authenticated with that server.
password       - Password of the user authenticated with that server.
domainName    - Name of the Domain in which the server is.
version       - The Exchange Server version (can be one of 2003 or 2007).

Returns:
the status of the operation either success or error Message.

Responses :
success       - when the details fetched successfully.
failure       - If any problems occur while getting the details.

Note: In case the Exchange version is 2007, the following conditions must be met:
- the RAS must run from a machine inside the same domain as the Exchange Server
- the machine the RAS is running on must have PowerShell and Exchange 2007
  Management Tools installed
- the host parameter must contain only the host name, without its domain (ex. chaos)
- a domain username/password must be used that has access to both the RAS machine
  and the Exchange Server (ex. battleground\username)
- the domainName must contain the full name of the domain (ex. battleground.ad)
</pre>
```

図 30 - オペレーションの説明

さらに、Studio の Generate Documentation 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。Generate Documentation の詳細については、「[複数のオペレーションやフローの説明の表示](#)」を参照してください。



**ヒント:** オペレーションやフローの説明は、[Search] タブの結果エリアで確認できます。オペレーションまたはフローの検索方法については、「[フローまたはオペレーションの検索方法](#)」を参照してください。

## フォルダーの説明を表示するには

フォルダーの説明に含まれている情報の詳細については、「[説明: 適切なオペレーションの判断](#)」を参照してください。

### フォルダーの説明を表示するには

1. リポジトリペインで目的のフォルダーを右クリックし、[Properties] をクリックします。
2. フォルダーの [Properties] シートで、[Description] タブをクリックします。

## フローの説明を表示するには

フローの説明に含まれている情報の詳細については、「[説明: 適切なオペレーションの判断](#)」を参照してください。

### フローの説明を表示するには

- 目的のフローを**作成**ペインで開いて [Properties] (ペインの下部) をクリックし、[Description] タブをクリックします。

**注:** フローの説明にはフローの入力に関する情報も含まれているので、入力として使用できる値の種類についてのヒントにもなります。

## オペレーションの説明を表示するには

フローの説明に含まれている情報の詳細については、「[説明: 適切なオペレーションの判断](#)」を参照してください。

### オペレーションの説明を表示するには

3. 目的のオペレーションを**作成**ペインで開きます。  
もしくは  
目的のオペレーションから作成されたステップを右クリックし、[Open Operation] をクリックします。
4. オペレーションの [Properties] シートで、[Description] タブをクリックします。

# 複数のオペレーションやフローの説明の表示

Central でフローを実行するユーザーや他のフロー作成者は、場合によってはフローおよびオペレーションについて次のような詳しい説明を必要とすることがあります。

- 入力として必要なデータの種類
- 結果として生成されるデータ
- レスポンスの種類
- フローが実行されたときのステップでの処理



Generate Documentation 機能は、フロー作成者がフローまたはオペレーションの [Description] タブに入力した内容からこれらの情報を収集し、生成されたセルフドキュメンテーションを一貫性のあるリンク付きの HTML ファイルの一覧として提供します。

個々のオペレーションやフロー、またはフォルダーのコンテンツに関するこれらの情報を生成する方法としては次の2種類があります。

- (右クリックから) [Generate Documentation] コマンドを使用して、出力される HTML ファイルを Central ユーザーに配布する。[Generate Documentation] コマンドの詳細については、次に説明する手順「右クリックコマンドを使用してフローのドキュメンテーションを生成するには」を参照してください。
- **Generate Documentation** オペレーションを使用してフローを作成し、そのフローを Central の本稼働インストールに移行する。**Generate Documentation** オペレーションの詳細については、「[Central ユーザーによるドキュメンテーション生成の有効化](#)」を参照してください。

生成したセルフドキュメンテーションの形式をカスタマイズすることも可能です。詳細については、「[カスタム形式のドキュメンテーションの生成](#)」を参照してください。

注: Central ユーザーに表示されないフォルダー (**Integrations** フォルダーや **Operations** フォルダーのサブフォルダーなど) やそのコンテンツに関するドキュメンテーションは生成できません。**Integrations** フォルダー、**Operations** フォルダー、**Utility Operations** フォルダーは Central では表示されないためです。



**ヒント:** 非表示フォルダーのコンテンツに関するドキュメンテーションを自分で生成する場合は、目的のフォルダーを Library 内の別の場所にコピーしてから、そのコピーに対してドキュメンテーションを生成します。

### 右クリックコマンドを使用してフローのドキュメンテーションを生成するには

1. コンテンツに関する情報が必要なフォルダーを右クリックします。
2. 表示されたコンテキストメニューで、[Generate Documentation] をポイントしてから [Standard Format] をクリックします。

もしくは

**Generate Documentation** オペレーションを使用するフローを作成します。**Generate Documentation** オペレーションの使用法については、「[Central ユーザーによるドキュメンテーション生成の有効化](#)」を参照してください。

3. [Choose an output directory] ダイアログボックスで HTML ファイルの出力先を指定し、[Save] をクリックします。

フローおよびオペレーションのセルフドキュメンテーションが生成されると、その結果が Web ブラウザーに表示されます。



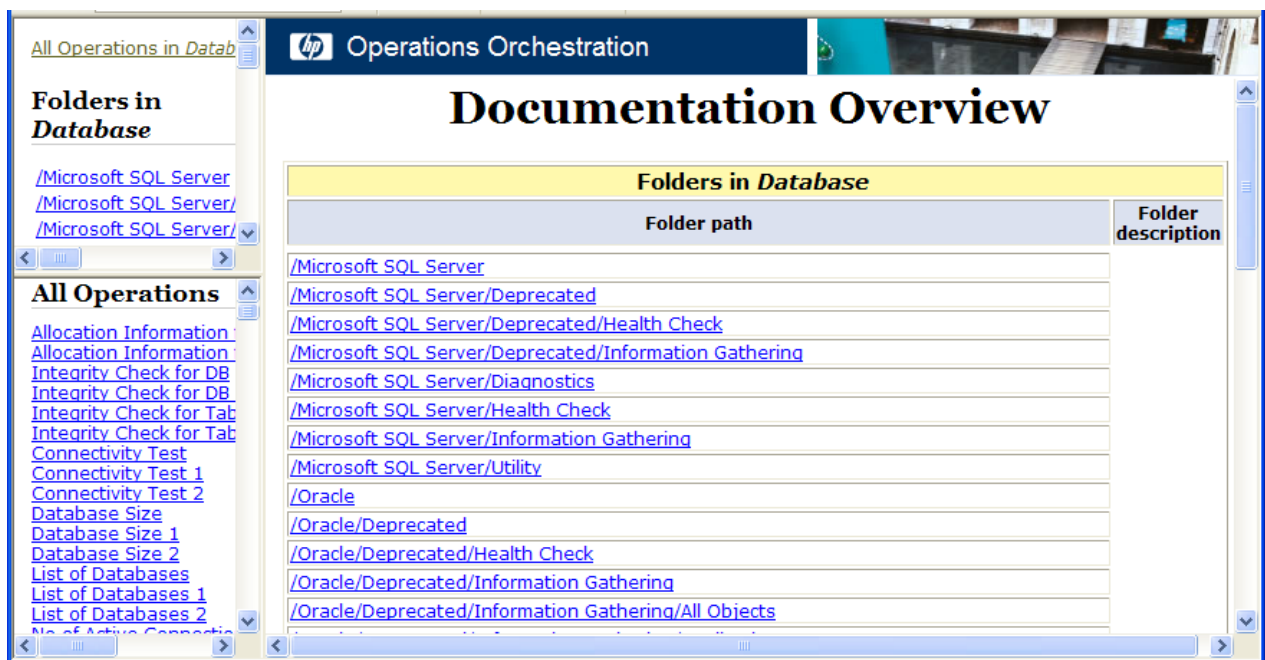


図 31 — Database Accelerator Pack に関して生成されたドキュメンテーション

- サブフォルダー内のオペレーションを表示するには、いずれかの [Folders in Database] ペインで目的のサブフォルダーをクリックします。選択したサブフォルダー内のオペレーションが左下のペインに表示されます。

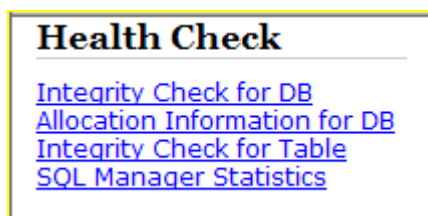


図 32 — Accelerator Packs/Database/Microsoft SQL Server/Health Check/ サブフォルダー内のオペレーション

- 単独のオペレーションまたはフローに関するドキュメンテーションを表示するには、次のいずれかの操作を行います。
  - [All Operations in <foldername>] をクリックしてから、左下のペインの一覧で目的のオペレーションまたはフローをクリックする。
  - 目的のオペレーションが格納されているフォルダーにドリルダウンしてから、左下のペインの一覧でそのオペレーションまたはフローをクリックする。

次に示すのは、**Integrity Check for DB** というフローに関するドキュメンテーションおよびフロー図のスクリーンショットです。

# Ops Flow: Exchange Server Health

## Description

Checks the health of an Exchange Server

### Inputs:

host - the host to check for events on.  
altuser - the username to use when authenticating to host.  
altpass - the password for altuser.  
mbuser - The user to check the mailbox of.  
mbpassword - The password for mbuser.

### Responses:

success - The operation succeeded, no problems found.  
failure - The operation failed or problems were found.

## Diagram

図 33 - Exchange Server Health フローの説明、およびその入力とレスポンスの情報

下にスクロールすると、フロー図、入力とレスポンスの一覧、さらにフローのステップの説明が表示されます。

## Diagram



図 34 - ヘルスチェックのフロー図

## 1. Is SQL Connected

### Operation: Is SQL Connected

#### Bindings:

- host : Prompt user
- database : Prompt user
- query : Value: sp\_help
- authtype : Prompt user from list
- save : Value: True
- timeout : Value: 30
- keyName : Value:
- user : Prompt user
- password : Prompt user

#### Transitions:

- rows returned : Is SQL Connected --> SQLCheck
- no rows returned : Is SQL Connected --> SQLCheck
- failure : Is SQL Connected --> Error : failure

#### Outputs:

## 2. SQLCheck

### Operation: Sql Check (1)

#### Bindings:

- password : Value:
- command : Value: dbcc checkdb(\${database})
- host : Value:
- username : Value:
- database : Value:
- databaseType : Prompt user

#### Transitions:

- failure : SQLCheck --> Error : failure
- success : SQLCheck --> Resolved : success

#### Outputs:

## 3. Resolved : success (return step)

## 4. Error : failure (return step)

図 35 – フローのステップに関する詳細情報

このセクションのトピック

[Central ユーザーによるドキュメンテーション生成の有効化](#)

[カスタム形式のドキュメンテーションの生成](#)

## Central ユーザーによるドキュメンテーション生成の有効化

**Generate Documentation** オペレーションを使用するフローを作成することにより、自動生成されるフロードキュメンテーションが提供する次のような有益な情報を Central ユーザーが活用できるようになります。

- フロー選択の指針
- ユーザープロンプトに対して入力として指定できる値
- ステップの未加工結果やレスポンスから得られる情報

Central ユーザーはこのフローをスケジュール化することもできます。ただし、Central の使用率が低いときは、選択したフォルダーに多数のフローがあると出力の生成に時間がかかる場合があります。

さらに、本稼働環境にある Central リポジトリが、ステージング環境から移行されたフローまたはフローの変更で周期的に更新される場合は、**Generate Documentation** フローをスケジュール化することで、Central ユーザーは常に最新のフロードキュメンテーションを見ることができます。

## Generate Documentation ステップを含むフローを作成するには

1. フローを作成します。  
フローの作成方法の詳細については、「[フローの作成](#)」を参照してください。
2. Library で、**Integrations** フォルダ、**Hewlett-Packard** フォルダ、**Operations Orchestration** フォルダの順に開き、**Generate Documentation** オペレーションを**作成**ペインのフローにドラッグします。
3. **Generate Documentation** オペレーションをダブルクリックしてオペレーションの Inspector を開きます。
4. [Inputs] タブの [doc\_TargetDir] 入力行で、右向き矢印 (→) をクリックします。  
入力エディターが開きます。

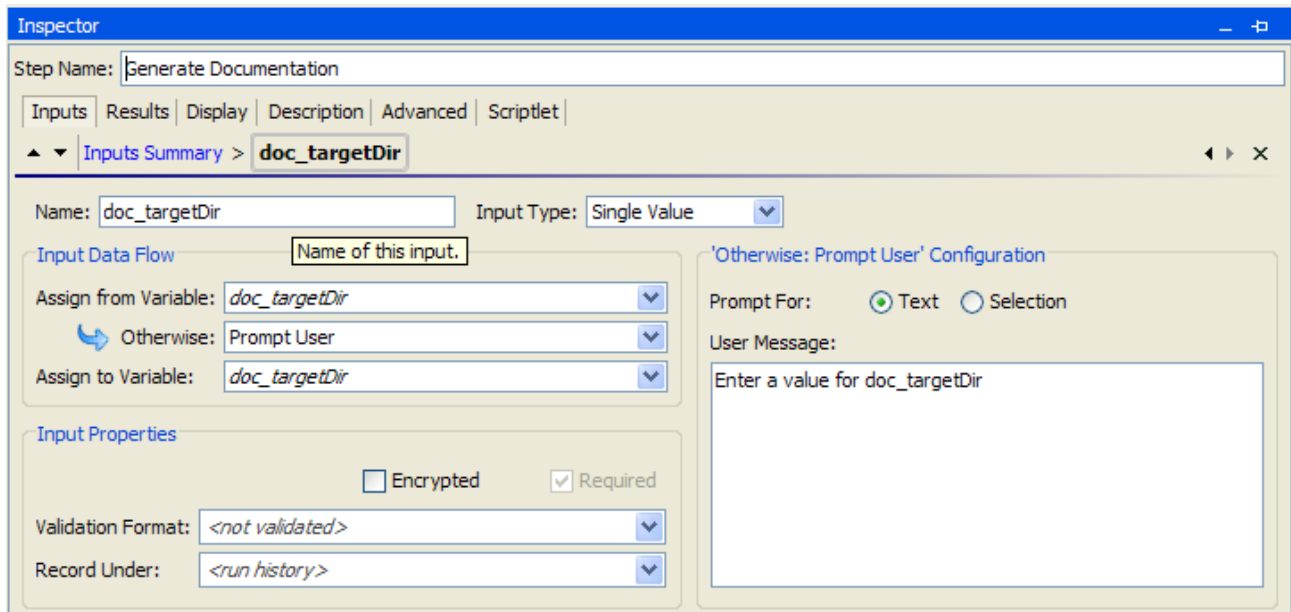


図 36 – Generate Documentation オペレーションの入力エディター

**doc\_targetDir** 入力には、ドキュメンテーションが生成される場所（ターゲットディレクトリ）を指定します。

5. [Assign from Variable] リストボックスには次の選択肢があります。
  - **doc\_targetDir**  
ユーザーがフローをスケジューリングするときに、**doc\_TargetDir** 入力に一定値を割り当てることができます。  
もしくは
  - **<not assigned>**  
[<not assigned>] を選択した場合は、[Otherwise] リストボックスで次のいずれかを選択します。
    - [Prompt User] を選択すると、ターゲットディレクトリの入力を促すメッセージがユーザーに表示されます。  
つまり、生成されたドキュメンテーションが出力されるフォルダはユーザーが選択できます。  
もしくは
    - [Constant] を選択した場合は、さらに [User Message] ボックスにターゲットディレクトリを入力します。
6. 入力エディターで、[Inputs Summary] の横にある下向き矢印 (▼) をクリックし、[libraryPath] 入力を開きます。  
この入力では、ドキュメンテーションの生成対象となる Library フォルダのパスを定義します。

7. [Assign from Variable] リストボックスには次の選択肢があります。

- **libraryPath**

ユーザーがフローをスケジューリングするときに、[libraryPath] 入力に一定値を割り当てることができます。

もしくは

- **<not assigned>**

[<not assigned>] を選択した場合は、[Otherwise] リストボックスで次のいずれかを選択します。

- [Prompt User] を選択すると、Library フォルダーのパスの入力を促すメッセージがユーザーに表示されます。

つまり、ドキュメンテーションの生成対象となるフォルダーはユーザーが選択できます。

もしくは

- [Constant] を選択した場合は、さらに [User Message] ボックスに Library フォルダーのパスを入力します。

**注:** Library パスを指定する場合、Library のフォルダー構造が変わったときはオペレーションでもそのパスを更新することが必要になる可能性があります。

### 注

- Library パスは必ず /Library/ から始まるようにします。
- Central では非表示のフォルダーを指定した場合、フローやオペレーションに関するドキュメンテーションは Central ユーザーには表示されません。

8. **Generate Documentation** ステップをフローの開始ステップにします。

9. **Success** と **Failure** のリターンステップ、および必要な機能を提供するその他のステップを追加してフローを完成させます。

## カスタム形式のドキュメンテーションの生成

Studio では、HP OO ホームディレクトリの **¥Studio¥extra¥template¥**内に .vm テンプレートのセットが標準で用意されています。これらのテンプレートには、生成されたドキュメンテーションを表示する際の HTML ファイルの外観やフレームが定義されています。

デフォルトテンプレートである DescribeFlows.vm では、デフォルトのドキュメンテーション形式が生成されます。フローまたはオペレーションについて生成されたドキュメンテーションの形式をカスタマイズするには、HTML ファイルおよびフレームの外観や動作を決定する .vm テンプレートのコピーを編集します。.vm テンプレートは、Apache Velocity テンプレート言語で作成されています。

このトピックのテンプレートの説明に登場するフレームの定義は次のとおりです。

- **overview-frame** – 左上のフレーム (概要フレーム)
- **folderFrame** – 左下のフレーム (フォルダーフレーム)
- **headerFrame** – 右上のフレーム (ヘッダーフレーム)
- **opFrame** – 右下のフレーム (オペレーションフレーム)

次の図は、HTML ページの基本的な構造 (フレームセット) と、DescribeFlows.vm での各フレーム定義を示したものです。

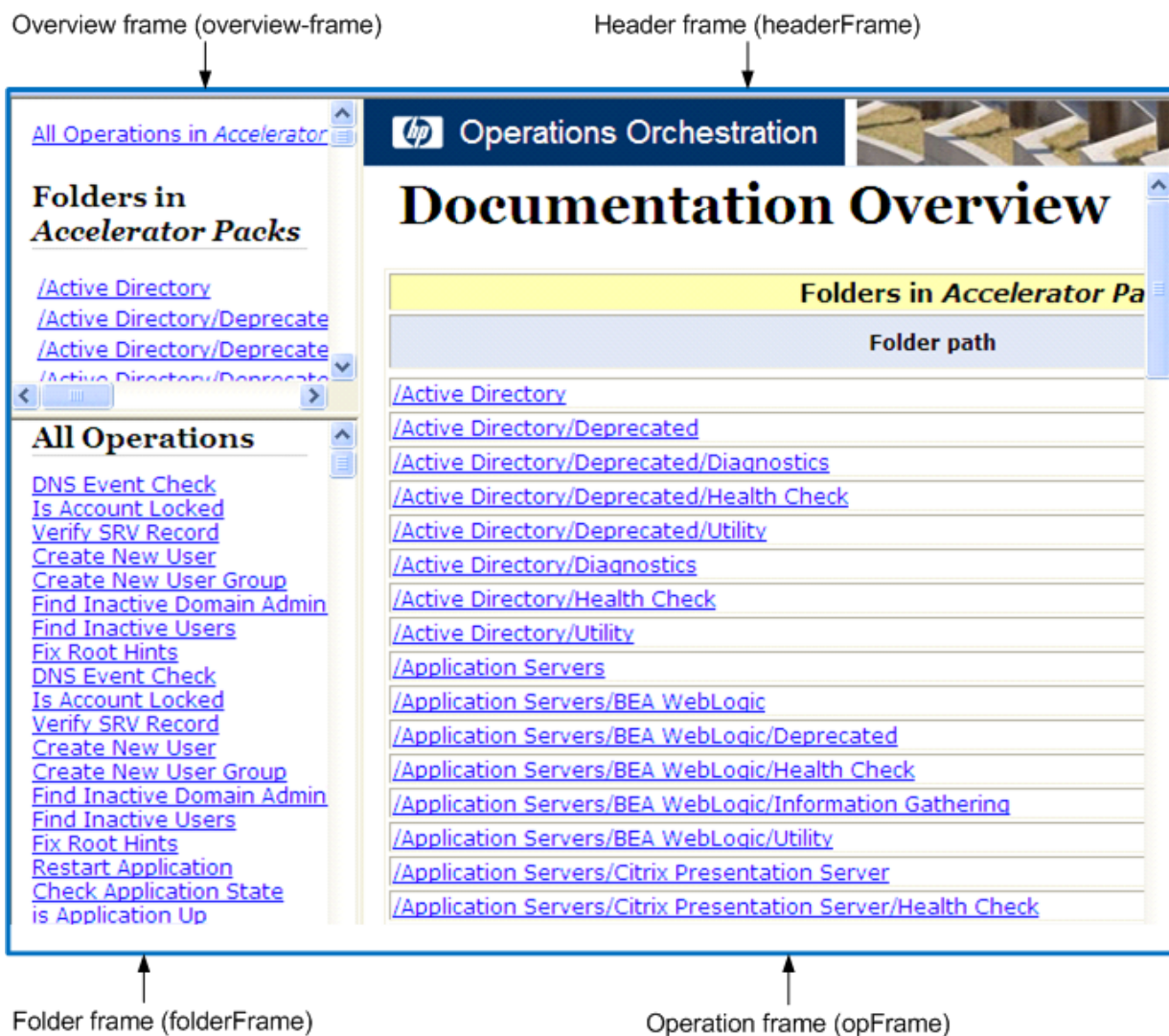


図 37 — Generate Documentation のフレームセット

このセクションのトピック

[.vm テンプレートファイルの説明](#)

[Generate Documentation の出力の構成](#)

[Generate Documentation のテンプレートの編集](#)

## .vm テンプレートファイルの説明

.vm テンプレートは、HP OO ホームディレクトリの `¥Studio¥extra¥template¥` 内にあります。これらのテンプレートは、フォルダーを右クリックして [Generate Documentation]、[Custom Format] の順にクリックしたときに、[Select Documentation Generation Template] ダイアログボックスには表示されません。テンプレートの編集方法については、「[Generate Documentation のテンプレートの編集](#)」を参照してください。

## Folder\_template.vm

ルートテンプレート。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  
後述の定義を参照してください。
- 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テーションの表を生成します。

- Header.css  
後述の定義を参照してください。
- 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  
後述の定義を参照してください。
- 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テーションを生成します。

- Header.css  
後述の定義を参照してください。
- 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テーションを生成します。

- Header.css  
後述の定義を参照してください。
- 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  
後述の定義を参照してください。
- 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・パッカーのバナー。

### 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。

### Hp\_rockwell.css

ヒューレット・パッカーのバナー用のスタイルシート。

### Hp\_steps\_307x39.jpg

ヒューレット・パッカーのバナー用の画像。

### Logo\_hp\_smallmasthead.gif

ヒューレット・パッカーのバナー用のロゴ。

## Generate Documentation の出力の構成

テンプレートの階層は次のように構成されていて、各テンプレートはそれぞれ下位の階層にあるテンプレートを呼び出します。

- Folder\_template.vm
  - All\_folders\_template.vm
    - All\_ops\_template.vm
      - Op\_template.vm
        - Folder\_template.vm
          - ...
        - Folder\_contents.vm
          - Op\_template.vm
            - ...



- Flow\_template.vm
  - Flow\_template.vm
    - ...
  - Folder\_contents.vm
    - ...
  - Op\_template.vm
    - ...
- Folder\_contents.vm
  - Op\_template.vm
    - ...
  - Flow\_template.vm
    - ...
- All\_ops\_template.vm
  - Op\_template.vm
    - ...
  - Flow\_template.vm
    - ...
- Folder\_overview\_template.vm
  - Folder\_contents.vm
    - ...

## Generate Documentation のテンプレートの編集

Generate Documentation 出力のカスタム表示を作成するための手順の概要は次のとおりです。

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。
2. 名前を変更したコピーに必要な編集を行います。

編集したコピーは、右クリックの [Generate Documentation] コマンドを使用して [Custom Format] を選択したときに、[Select Documentation Generation Template] ダイアログボックスに表示されます。

[Select Documentation Generation Template] ドロップダウンリストでは 1 つしか選択できないことに注意してください。

3. 複数の .vm テンプレートを使用するカスタム表示を作成する場合は、Folder\_template.vm のコピーを作成し、名前を変更してから、カスタム表示に追加する .vm テンプレートを参照するようにコピーを編集します。  
名前を変更した Folder\_template.vm のコピーは [Select Documentation Generation Template] ドロップダウンリストに表示されます。編集した .vm テンプレートを使用するカスタム表示をこのドロップダウンリストから取得するには、名前を変更して編集した Folder\_template.vm のコピーを選択します。

たとえば、全オペレーションのリストおよびフォルダーのコンテンツの説明の表示方法を変更したカスタム表示を使用するとします。この場合、次のような手順に従います。

1. 全オペレーションのリストの表示方法を変更するには、次の手順で行います。
  - All\_ops\_template.vm をコピーします。
  - コピーの名前を Better\_all\_ops\_template.vm に変更します。
  - Better\_all\_ops\_template.vm をカスタマイズします。
2. フォルダーのコンテンツの表示方法を変更するには、次の手順で行います。
  - Folder\_overview\_template.vm をコピーします。
  - コピーの名前を Better\_folder\_overview\_template.vm に変更します。
  - Better\_folder\_overview\_template.vm をカスタマイズします。

3. 表示に使用される .vm ファイルを変更するには、次の手順で行います。
  - Folder\_template.vm をコピーします。
  - コピーの名前を Better\_folder\_template.vm に変更します。
  - Better\_folder\_template.vm を次のように編集します。
    - All\_ops\_template.vm の参照を Better\_all\_ops\_template.vm を参照するように変更します。
    - Folder\_overview\_template.vm の参照を Better\_folder\_overview\_template.vm を参照するように変更します。

これで、[Generate Documentation] コマンドを使用したときに以下のテンプレートが [Select Documentation Generation Template] ドロップダウンリストに表示されるようになりました。

- Better\_all\_ops\_template.vm
  - Better\_folder\_overview\_template.vm
  - Better\_folder\_template.vm
4. ドロップダウンリストから **Better\_folder\_template.vm** を選択します。

.vm ファイルの構成、およびこれらのファイルで定義されているフレームの相互関係の詳細については、「[.vm テンプレートファイルの説明](#)」および「[Generate Documentation の出力の構成](#)」を参照してください。

### カスタム形式のドキュメンテーションを生成するには

1. HP OO ホームディレクトリにある ¥Studio¥extra¥template に移動します。
2. カスタマイズするテンプレートに対して次の操作を行います。
  - テンプレートのバックアップコピーを作成する。
  - テンプレートをコピーする。
  - コピーの名前を変更する。
  - テキストエディターでコピーに変更を加える。
  - テンプレートのカスタムバージョンをバックアップする。

Studio をアップグレードすると、テンプレートは上書きされます。変更したテンプレートのバックアップを使用して、アップグレード後のテンプレートに対する変更を元に戻すことができます。

3. Studio で、コンテンツの情報を確認する必要があるフォルダーを右クリックします。
4. 表示されたコンテキストメニューで、[Generate Documentation] をポイントしてから [Custom Format] をクリックします。
5. [Select Documentation Generation Template] ダイアログボックスが表示されたら、このドロップダウンリストから、使用するカスタムテンプレートを選択します。  
[Specify root file name] ダイアログボックスが表示されます。

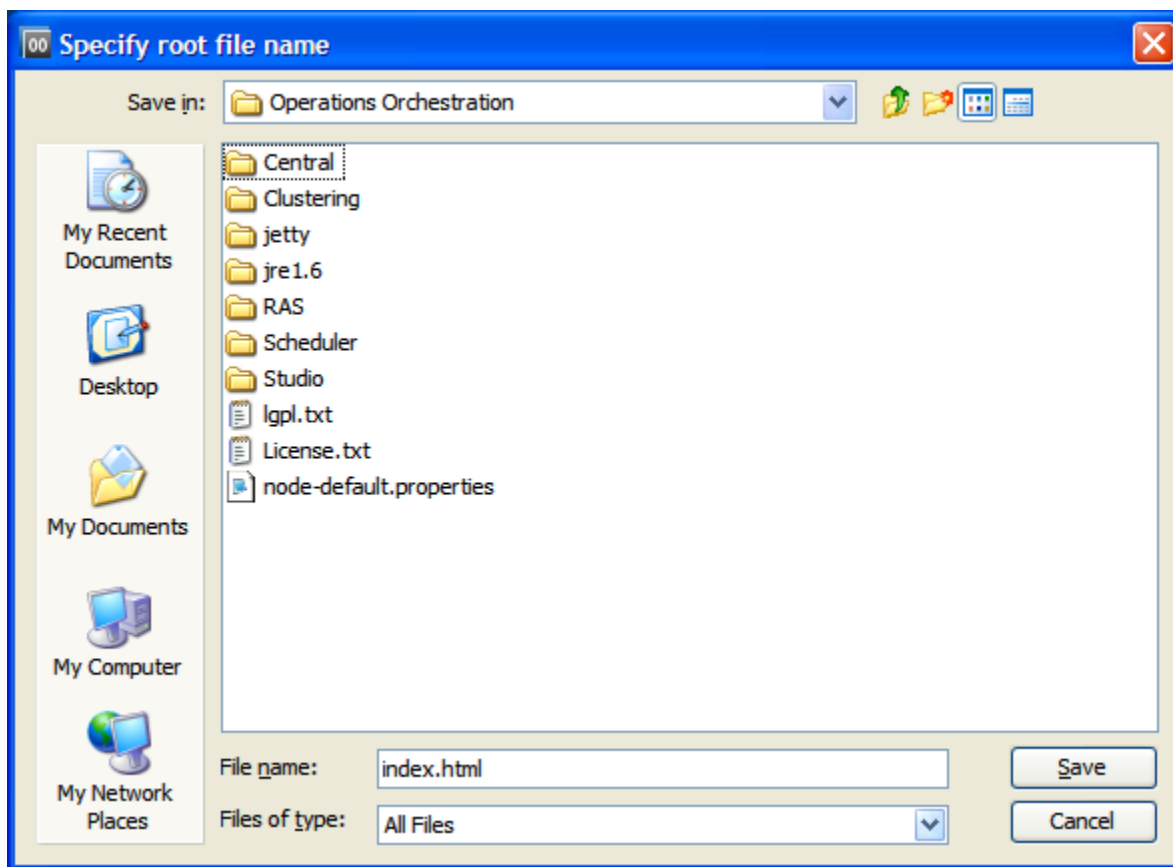


図 38 — 出力先の選択

6. 生成された HTML ファイルを保存する既存のフォルダーに移動します。
7. [File name] ボックスに開始ページのファイル名を入力して、[Save] をクリックします。  
デフォルトの開始ファイルの名前は index.html です。このファイルが自動的にデフォルトの Web ブラウザーに表示されます。別の名前や拡張子を指定した場合は、開始ページを開くためのプログラムを手動で選択する必要があります。

ドキュメンテーションが生成されます。

## 特定のオペレーションを使用するフローの検索

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で、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。

**重要:** 独自のオペレーションやフローを作成した後は、そのオペレーションまたはフローが別のどのフローで使用されているかを確認するまでは変更を加えないでください。

Studio には、特定のオペレーションを使用するフローの検索に関連する参照が 2 種類あります。

- オペレーションまたはフローを参照している **参照元**。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が該当します。
- オペレーションまたはフローが参照している **参照先**。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）が該当します。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。

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。

## オペレーションの参照を表示するには

1. Library で、目的のオペレーションまたはフローを右クリックします。
2. オペレーションまたはフローの参照元を表示するには、[References] をクリックしてから [What uses this?] をクリックします。

もしくは

オペレーションまたはフローの参照先を表示するには、[References] をクリックしてから [What does this use?] をクリックします。

[References] ペインが開いて、オペレーションまたはフローの参照元または参照先が表示されます。



ヒント: [References] ペインを閉じた後で、最後に取得した参照を再度表示するには、[References] タブをクリックします。

## フローおよびオペレーションのコピー

使用するフローまたはオペレーションを見つけたら、コピーを作成してからそのコピーで作業するようにしてください。Library の **Accelerator Packs**、**Integrations**、**Operations**、**Utility Operations** の各フォルダー内にあるフローおよびオペレーションはシールされていて編集できません。そのため、これらのフローまたはオペレーションの編集バージョンを作成するには、コピーを作成し、そのコピーに変更を加える方法しかありません。

例として、編集が必要なオペレーションに、InputA という入力があるとします。また、このオペレーションから作成したステップが、FlowAlpha というフローで使用されているとします。FlowAlpha には、InputA に値を提供するフロー入力があります。ここで、このオペレーションに InputB を追加する場合について考えてみます。オペレーションから作成したステップには InputB の値を取得する方法がありません。入力の値を取得する方法が定義されていないときは、ユーザーに値の入力を求めるプロンプトが表示されます。ただし、Central ユーザーが FlowAlpha を自動的に実行するスケジュールを作成していた場合、FlowAlpha は実行中に中断します。実行処理にユーザーの入力を必要としないことが、フローを完全に自動実行するための条件であるためです。

注: オペレーションを編集してもその利用機会が少なく、オペレーションから作成されたステップに対して同様の編集ができる場合は、オペレーションを編集するかわりに、ステップに必要な変更を加えることが可能です。ステップとオペレーションの違いについては、「[フロー、ステップ、オペレーションの高度な概念](#)」を参照してください。

ステップの編集については、次のトピックを参照してください。

- [入力: オペレーションへのデータの提供](#)
- [出力、レスポンス、およびステップの結果](#)
- [ステップのベースとなるオペレーションの変更](#)

コンテキスト (右クリック) メニューを使用して、フローやオペレーションをコピーまたは複製できます。コンテキストメニューで使用できるオプションは、対象のフローまたはオペレーションがシールされたフォルダー (**Integrations** など) 内にあるか、シールされていないフォルダー (**My Ops Flows** など) 内にあるかによって異なります。

シールされたフォルダー	シールされていないフォルダー
オペレーションのコピー	フローおよびオペレーションのコピーと複製
フローのコピーと複製	

## フローやオペレーションをコピーまたは複製するには

1. Library で、目的のフローまたはオペレーションを見つけて右クリックします。
2. [Edit] をポイントしてから、[Copy]、[Copy Deep]、[Duplicate] のいずれかをポイントします。
  - 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ます。
  - オペレーションをコピーする場合は、シール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 (Ctrl+V)。
  - [Copy Deep] をクリックすると、フローだけでなくそのフローが使用するすべてのオペレーションもコピーされます。
3. [Copy] をクリックした場合は、コピーを保存する場所に移動してから、Ctrl+V を押します。コピーの名前は変更可能です。

これで、フローで作業する準備ができました。

## フローまたはオペレーションの名前の変更

フローやオペレーションの名前はコンテキストメニューを使用して変更できます。コンテキストメニューで使用できるオプションは、対象のフローまたはオペレーションがシールされたフォルダー (**Integrations** など) 内にあるか、シールされていないフォルダー (**My Ops Flows** など) 内にあるかによって異なります。

シールされたフォルダー	シールされていないフォルダー
フローの名前変更	フローおよびオペレーションの名前変更

### フローまたはオペレーションの名前を変更するには

1. Library で、目的のフローまたはオペレーションを見つけて右クリックします。
2. [Rename] をクリックします。
3. 強調表示されたフィールドに新しい名前を入力し、ENTER を押して保存します。

## フローおよびオペレーションのブックマーク

フローまたはオペレーションを再度使用する可能性があるときは、[Bookmarks] ペインに追加しておくことでアクセスが簡単になります。追加したフローやオペレーションは、通常の Library 内の場所からも利用できます。また、フローやオペレーションを [Bookmarks] ペインに追加すると、フロー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。

[Bookmarks] ペインには、フロー用とオペレーション用の2つのセクション (シェルブ) があります。

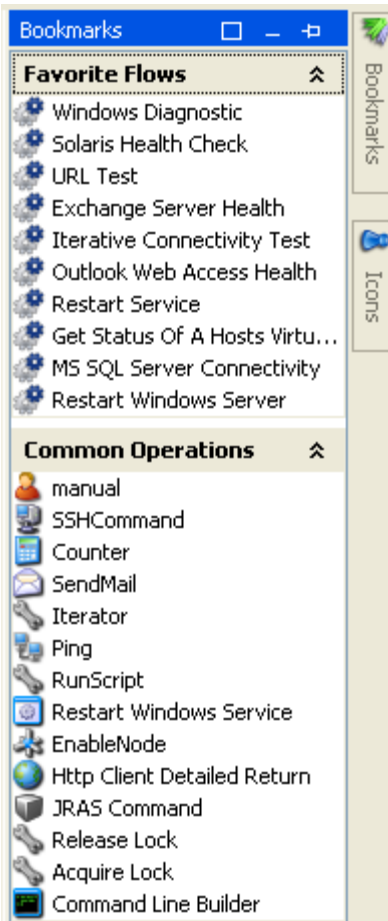


図 39 – [Bookmarks] ペイン — フローセクションとオペレーションセクション

## ブックマーク: フローおよびオペレーションの追加と削除

フローまたはオペレーションを [Bookmarks] ペインに追加するには

1. [Bookmarks] タブをクリックして [Bookmarks] ペインを開いてから、ペインを開いたままにするためにペインの右上隅にあるピンアイコン (📌) をクリックします。
2. Library または [Search] ボックスからフローまたはオペレーションを [Bookmarks] ペインにドラッグします。フローの場合は [Favorite Flows] セクション、オペレーションの場合は [Common Operations] セクションにドラッグします。

## [Bookmarks] ペインのシェルフ: 追加、削除、名前の変更

シェルフの削除、追加、名前の変更を行うには

1. [Bookmarks] ペインのシェルフのタイトルバーを右クリックし、実行する操作を選択します。
2. 必要に応じて、表示されたダイアログボックスに適切な情報を入力します。

## [Bookmarks] ペインのシェルフ: 表示、非表示、移動

### シェルフの非表示、表示、折りたたみを行うには

1. シェルフを非表示にするには、目的のシェルフのタイトルバーで右クリックし、[Hide] をクリックします。
2. 非表示にしたシェルフを表示するには、[Bookmarks] ペインの何もない場所で右クリックし、[Show All] をクリックします。  
もしくは  
[Show] をクリックし、非表示のシェルフのリストから、表示するシェルフを選択します。
3. 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。  
シェルフを再び展開するには、二重矢印を再度ダブルクリックします。

### シェルフを移動するには

- シェルフのタイトルバーで右クリックし、[Move Up] または [Move Down] をクリックします。

## 削除したライブラリオブジェクトの復元

Studio では、さまざまな操作に対して一般的な「元に戻す」機能を使用できるだけでなく、削除したライブラリオブジェクトを復元することも可能です。

ライブラリオブジェクト（フロー、オペレーション、または Library の **Configuration** フォルダ内のオブジェクト）を削除すると、そのオブジェクトは Library から削除されます。ただし、削除したオブジェクトの一覧を表示して、元に戻す必要があるオブジェクトを復元することが可能です。削除の記録は、Studio のインストール後に初めて削除を行ったときから開始されます。

注: 削除したオブジェクトを復元するには、そのオブジェクトに対する読み取りおよび書き込み権限が必要です。

### 削除したオブジェクトを復元するには

1. [Repository] メニューで、[View Delete History] をクリックします。  
削除の履歴が次のダイアログボックスに表示されます。

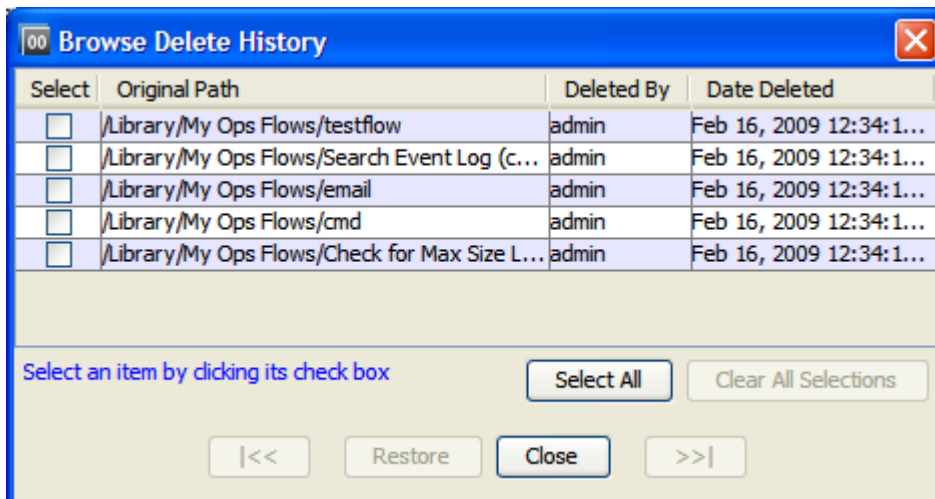


図 40 - 削除の履歴

削除したオブジェクトの一覧が長すぎてダイアログボックスに一度に表示できないときは、次に進む

( >>| ) ボタンまたは前に戻る ( |<< ) ボタンを使用して、1 ページずつ履歴を移動できます。



2. 復元するオブジェクトの横にある [Select] チェックボックスをオンにして、[Restore] をクリックします。  
次の操作が可能です。
  - 削除したオブジェクトを復元対象としてすべて選択する。
  - 選択をすべてクリアする。
  - 復元操作をキャンセルする。
3. 復元しようとしているオブジェクトが、すでに削除されたフォルダー内にあった場合は、次のダイアログボックスが表示されます。

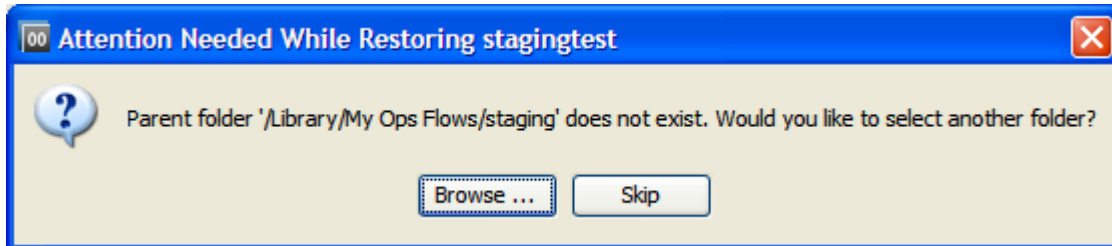


図 41 - 親フォルダーが削除されている場合のエラー

- オブジェクトの復元を中止する場合は、[Skip] をクリックします。  
もしくは  
復元したオブジェクトを格納する新しいフォルダーを指定する場合は、[Browse] をクリックし、新しいフォルダーに移動して選択してから、[OK] をクリックします。
4. 復元しようとしているオブジェクトが格納されていたフォルダーが、オブジェクトの削除後に移動されていた場合は、次のダイアログボックスが表示されます。

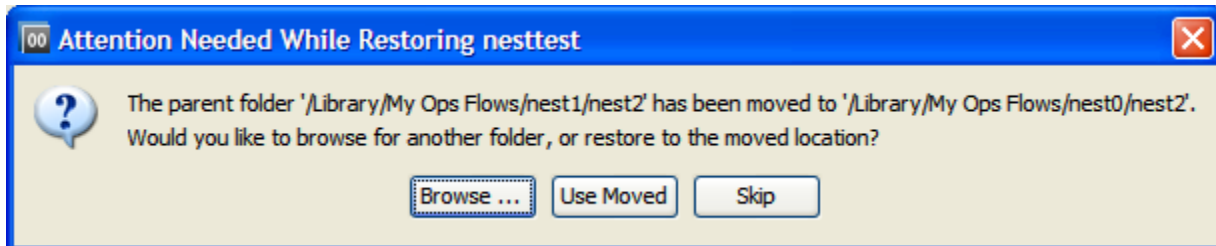


図 42 - 親フォルダーが移動されている場合のエラー

- オブジェクトの復元を中止する場合は、[Skip] をクリックします。  
もしくは  
オブジェクトを格納する新しいフォルダーを指定する場合は、[Browse] をクリックし、新しいフォルダーに移動して選択してから、[OK] をクリックします。  
もしくは  
移動後の場所にある親フォルダーにオブジェクトを復元する場合は、[Use Moved] をクリックします。



# ステップの作成

ステップの作成では、オペレーションまたはフローから作成する簡単な方法がいくつかあります（フローからステップを作成する場合、フローは一種のオペレーションとして扱われることに留意してください）。

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、結果、参照、その他の特性を継承します。ステップのこれらの要素は変更が可能であり、変更によりステップ作成のベースとなったオペレーションに影響することはありません。

**重要:** ステップの基になっているオペレーションの入力や結果を変更すると、そのオペレーションのインスタンスであるすべてのステップの入力や結果も変更されます。そのため、オペレーションに変更を加えると、オペレーションの入力や結果を変更する前に作成されたステップ（およびそのフロー）は中断する可能性があります。

各種ステップのベースとして機能できるように、オペレーションには高い汎用性があります。そのため、特定の状況に合わせてステップに変更を加えることが可能なのです。

[フローへのステップの追加](#)

[使用頻度の高いフローおよびオペレーションを基にしたステップの作成](#)

[開始ステップの変更](#)

[ステップのコピー](#)

[ステップの編集](#)

[フロー図内のステップの位置変更](#)

[ステップ実行前に表示するユーザープロンプトの作成](#)

## フローへのステップの追加

ステップの作成に利用できるオペレーションやフローを探すときは、まず **Library** のフォルダーを検索します。これらのフォルダーには、デフォルトコンテンツのフローやオペレーション（つまり、インストール時に提供されるフローやオペレーション）が格納されているためです。こうしたフォルダーのコンテンツを使用する場合、通常は編集する必要がありますが、これらのコンテンツはシールされていて、編集はそのコピーに対してしかできません。したがって、このようなフローまたはオペレーションを使用するには、コピーを作成してからそのコピーを編集し、それを基にしてステップを作成します。

**重要:** 最良の方法は、作成するフローやオペレーションと同じ手順に従うことです。ステップをコピーする手順については、「[ステップのコピー](#)」を参照してください。

デフォルトコンテンツが格納されているのは以下のフォルダーです。

- **Accelerator Packs**

このフォルダーには、大部分のネットワークに対して次の機能を提供することにより、一般的な IT 問題を解決する目的で設計されたフローが格納されています。

- 複雑なヘルスチェック、問題選別、診断、または修復フロー。
- 1 つ以上のデータを収集してユーザーに表示したり、簡単なアラート通知、一部データの収集、チケットへのデータの記録を行う単純なフロー。

Accelerator Pack の最上位にあるフローの多くは、詳細なヘルスチェック、問題選別、診断、修復機能を備えています。

- **Integrations**

このフォルダーには、HP OO と他のエンタープライズソフトウェア製品（Hewlett-Packard Network Node Manager、BMC Remedy など）を統合するためのオペレーションや、その使用方法を示すサンプルが格納されています。ある一定のデータセンターで使用されるエンタープライズソフトウェア製品に特有のカスタマイズレベルがある場合は、これらのコンテンツを使用する際にカスタムフローの作成が必要になります。

- **ITIL**

このフォルダーには、ITIL の仕様に準拠する他のエンタープライズレベルソフトウェアとの統合を自動化するフローが格納されています。

- **Operations**

このフォルダーには、LDAP や Linux など、一般的なプロトコルやソフトウェアシステムで動作する汎用のオペレーションが格納されています。

これらのオペレーションはシールされており、変更はできません。そのため、これらのオペレーションに入力用の固定値は設定されていません。入力はずべて、割り当て値を持たない（割り当てなし）か、またはユーザーによる入力となります。ただし、WMI コマンドを使用するような汎用性の高いオペレーションは、この規則の例外となります。

**Operations** フォルダおよびそのサブフォルダ内のフローは、サブフローとして使用されることを前提としています。親フローとして実行するのに適したフローは、**Accelerator Packs** フォルダ内のフローです。

- **Utility Operations**

このフォルダには、ほぼすべてのテクノロジーで利用できる基本的な機能を備えたフローおよびオペレーションが格納されています。これらのフローやオペレーションでは、日付と時間の計算やフィルタリング、数学的操作や比較などを実行できます。

**注:** 現在開いているフローキャンバスに **Library** からフローをドラッグしてフローのステップを作成する場合、ドラッグしたフローはドラッグ先のフローのサブフローになります。

フローのキャンバスにドラッグした最初のステップは自動的にフローの開始ステップになり、緑色の枠で示されます。

**重要:** 作成したステップと、そのステップが関連しているオペレーションまたはサブフローを混同しないようにしてください。ステップは、オペレーションまたはフローの具体的なインスタンスです。現在のフローの要件に特化した効果を生むようにオペレーションを変更する必要がある場合は、ステップに対して適宜変更を加えます。

例として、オペレーションの実行対象となるホストマシンを指定する際は、オペレーションに対する入力ではなくステップに対する入力に対象のホストを指定します。オペレーションの入力に特定のマシンを指定した場合、オペレーションが属するフローや、別のマシンに対するオペレーションの実行処理に依存するフローが中断する原因になります。

### フローにステップを追加するには

1. 次のいずれかから、オペレーションまたはフローを作成キャンバスに追加します。

- **リポジトリペイン**
- **検索結果**
- **[Bookmarks] ペイン**

必要に応じて、フロー内での機能を表すような名前をステップに付けます（オペレーションの名前は、特定のステップでの用途に比べると総称的です）。

2. ステップの名前を変更するには、フローキャンバス内でステップを右クリックして **[Rename]** をクリックし、強調表示されたフィールドに新しい名前を入力してから **ENTER** を押します。

3. 必要に応じてステップを構成します。
  - 入力用のデータソースを追加、定義する。  
詳細については、「[入力: オペレーションへのデータの提供](#)」を参照してください。
  - 結果の追加、データソースの定義とフィルタリング、フロー変数への値の格納、フロー内の次のステップを決めるレスポンスの追加を行う。  
これらの項目の詳細については、「[出力、レスポンス、およびステップの結果](#)」を参照してください。
  - トランジションを追加する。  
詳細については、「[トランジション: ステップの接続](#)」を参照してください。

フローの作成キャンバスにオペレーションまたはフローをドラッグする方法のほかに、既存のステップをコピーしてステップを作成することもできます。

## 使用頻度の高いフローおよびオペレーションを基にしたステップの作成

右クリックメニューから、使用頻度の高いオペレーションやフローを基にしてステップを作成できます。利用できるオペレーションおよびフローは次のとおりです。

- Favorite Flows
  - Windows Diagnostic
  - Solaris Health Check
  - URL Test
  - Exchange Server Health
  - Iterative Connectivity Test
  - Outlook Web Access Health
  - Restart Service
  - Get Status of a Hosts Virtual Machines
  - MS SQL Server Connectivity
  - Restart Windows Server
- Common Operations
  - manual
  - SSHCommand
  - Counter
  - SendMail
  - Iterator
  - Ping
  - RunScript
  - Restart Windows Service
  - EnableNode
  - Http Client Detailed Return
  - JRAS Command
  - Release Lock
  - Acquire Lock

## Favorite Flow または Common Operation からステップを作成するには

1. Studio のフローキャンバスでフローを開きます。
2. 次のいずれかを実行します。
  - フローの作成キャンバスの何もない場所で次の操作を行います。
    - 右クリックして [Insert] をポイントしてから、[Favorite Flows] または [Common Operations] をポイントする。
    - 表示されたメニューで、フローに追加するオペレーションまたはフローをクリックする。
  - もしくは
  - [Bookmarks] タブをクリックし、表示された [Bookmarks] ペインで、[Favorite Flows] 内のフローまたは [Common Operations] 内のオペレーションをフロー図にドラッグします。
3. 必要に応じてステップを編集し、フロー内の先行ステップと後続ステップに接続します。

## 開始ステップの変更

### 開始ステップとなるステップを変更するには

- フローの開始ステップにするステップを右クリックし、表示されたコンテキストメニューから [Set Start Step] を選択します。

## ステップのコピー

同一のフローまたは別のフロー内に新しいステップを作成するときに、既存のステップをベースとして利用できます。その際は、目的のステップをコピーし、名前を変更してから編集します。

この方法では、入力を定義したり、フィルター、結果、その他のフローオブジェクトを作成する時間を節約できます。



**ヒント:** 以下の 2 つの方法以外に、Ctrl を押しながらステップをコピー先にドラッグすることで、ステップをコピーすることも可能です。

### ステップをコピーするには

1. キャンバス上でステップを右クリックし、[Copy] をクリックします。
2. キャンバス上の任意の場所で右クリックし、[Paste] をクリックします。

### フロー間でステップをコピーするには

1. コピーするステップが含まれているフローの [Design] タブを開きます。
2. 目的のステップを強調表示し、右クリックメニュー、または Windows の標準のコピー用アクセラレータキー (Ctrl+C) を使用してコピーします。
3. コピーしたステップを追加するフローの [Design] タブを開きます。
4. 必要な場所にステップを貼り付けます。
5. 作業結果を保存します。

## ステップの編集

入力や結果など、ステップに関する設定（トランジション先を除く）を変更するには、ステップの Inspector を開きます。

### ステップの Inspector を開くには

1. ステップをダブルクリックします。  
もしくは  
ステップを右クリックし、右クリックメニューから [Properties] をクリックします。  
Inspector が次のような状態で開きます。

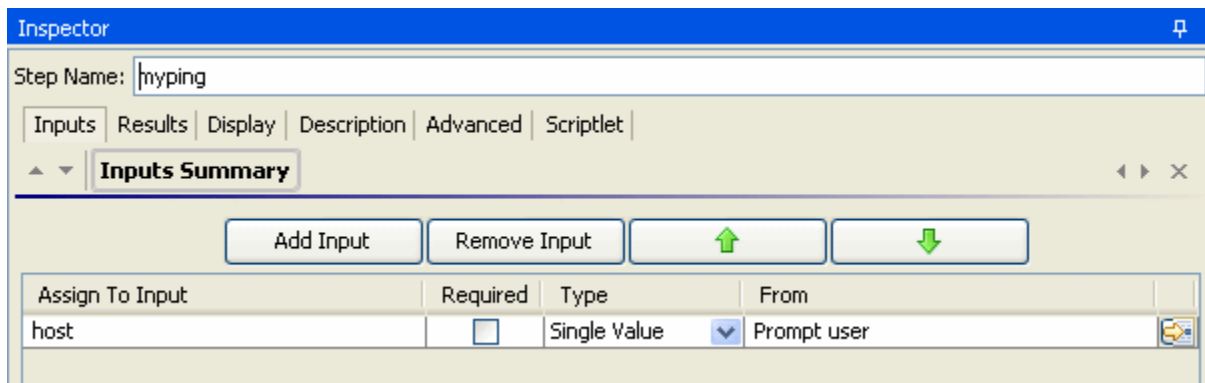


図 43 – ステップの Inspector

2. ステップ間で、またはステップからトランジションへフォーカスを移すときに Inspector を閉じたり開いたりする必要がないように Inspector を開いたままにするには、Inspector のタイトルバーの右端にあるピンアイコン (📌) をクリックします。

## フロー図内のステップの位置変更

ステップをドラッグするだけでなく、複数のステップを 1 つのグループとして移動することもできます。Shift キーまたは Ctrl キーを押しながら移動するステップをクリックし、それらのステップをまとめてドラッグできます。

## ステップ実行前に表示するユーザープロンプトの作成

ステップにユーザーからの入力が必要ない場合でも、ステップを実行する前にユーザーの承認を求めるプロンプトを作成することが可能です。



**ヒント:** Central ユーザーの情報を要求するステップの場合、このステップについてユーザーに必要な情報を知らせるプロンプトを作成することにより、フローの説明の補足となり、ユーザーにとってわかりやすくなります。

### ステップのユーザープロンプトを作成するには



1. ステップの Inspector を開いて、[Display] タブをクリックします。
2. ユーザーにプロンプトが表示されるようにするには、[Always prompt user before executing this step] チェックボックスをオンにします。

3. 次の各ボックスを使用してプロンプトを作成します。
  - [Prompt Title] ボックスでプロンプトのラベルを作成します。
  - [Prompt Width] ボックスと [Height] ボックスでプロンプトのサイズをピクセル単位で指定します。
  - [Prompt Text] ボックスにユーザーへのメッセージを入力します。
4. [OK] をクリックして変更を保存します。

## トランジション: ステップの接続

2つのステップを接続するには、1つ以上のトランジションを使用します。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（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、別のステップへのトランジション、または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。

レスポンスアイコンには次の種類があります。

- success (成功) 
- failure (失敗) 
- diagnosed (診断済み) 
- no action taken (アクションなし) 

1つのステップに複数のレスポンスを接続することが可能です。たとえば、失敗のレスポンスが複数ある場合、単独の **failure** リターンステップに接続されることがほとんどです。

次の図の中の線がトランジションを表しています。デフォルトでは、レスポンスの名前が、そのレスポンスを次のステップに接続しているトランジションの名前になります。例を挙げると、**success** レスポンスを起点とするトランジションの名前は **success** になります。



図 44 - フロー内のトランジション

トランジションは、単にステップを接続するだけでなく、以下の点で有用です。

- トランジション後にフローを続行できるユーザーを制御する。  
セキュリティ上の理由から、あるいはフローの続行に一定の知識を要することから、特定のトランジション後にフローの実行を続行できるユーザーを制限することが必要になる場合があります。  
この制御は、**ゲートトランジション**（フローを実行するアカウントについて特定の **OO** ロールのメンバーシップを必要とするトランジション）を使用して実行できます。  
ゲートトランジションは、**Studio** と **Central** のどちらのフロー図でも赤で表示されます。
- フローを別のユーザーにハンドオフする。この処理が必要となるのは、次のステップでは別のユーザーからの情報が必要である場合、またはトランジションがゲートトランジションで、自分のアカウントには設定されていない権限が必要である場合です。  
ハンドオフの設定をすると、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てからメッセージを送信できます。
- フローの実行の価値を計算するベースとなる。  
トランジションに一定の価値を割り当てると、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、当該実行フローの価値にトランジションの価値が追加されます。この実行フロー全体の価値は、発生したトランジションに割り当てられた価値の合計です。  
**Central** ユーザーは、この値を **Central** のダッシュボードで確認できます。
- フローのユーザーに、ステップでどのような処理が行われたかを知らせる。  
発生するトランジションはステップの結果に応じて決まります。そのため、トランジションの説明は、各ステップで行われた処理の説明として **Central** の **[Results Summary]** エリアの **[Message]** に表示されます。  
次に示すのは、4つのトランジションの説明が表示されている **Central** の **[Results Summary]** エリアのスクリーンショットです。これらの説明は、トランジションの発生源となったステップでの処理内容を表しています。

Step	Response	Message
Get Stopped Services	✓	Retrieved a list of services which are currently stopped.
Select a Service	✓	Selected Adobe LM Service to restart.
Restart Service	✓	Restarted service Adobe LM Service
Resolved : success	✓	Return step - <b>Restart Service - Tutorial Flow</b>

図 45 – Central の [Results Summary]

トランジションの説明には、ステップのオペレーションから、または実行フローの任意の場所から取得される、動的に変化するデータを含めることができます。その場合、フロー変数に対象のデータを格納してから、そのフロー変数への参照を説明に追加します。参照の形式は、**#{フロー変数名}** です。

例として、**ping** コマンドを実行するステップでは、対象のホストマシンの名前を **host** というフロー変数に割り当ててあるとします。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。すると、**success** レスポンスの説明はたとえば「**Successfully pinged #{host}**」のように表示されます。「**server1**」という名前のホストに対して **Central** で実行した場合、概要説明には「**Successfully pinged server1**」と表示されます。

[トランジションの追加](#)

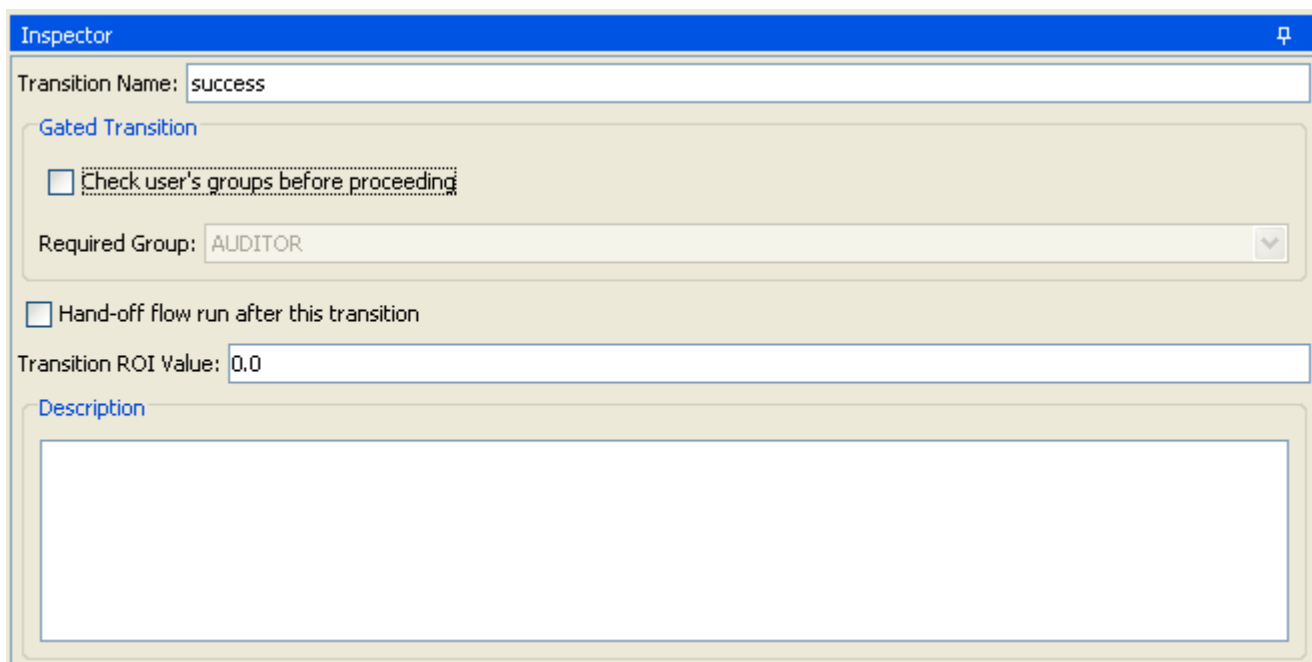
[トランジションの位置の変更](#)



## トランジションの追加

### 2つのステップ間にトランジションを追加するには

1. Studio の作成キャンバスでフローを開いた状態にして、次のステップに接続するステップ上でいずれかのレスポンスを表すアイコンをクリックし、そのレスポンスの結果発生するステップにドラッグします。  
作成したトランジションには、その発生源であるレスポンス名がラベル付けされます。
2. トランジションの Inspector を開くために、トランジションまたはその名前をシングルクリックで選択してフローキャンバスの下部にある [Inspector] タブをクリックします（またはトランジションをダブルクリックします）。
3. トランジションの Inspector が表示されます。次の例は、**Select a Service** ステップの **success** レスポンスを次のステップに接続する **success** トランジションの Inspector です。



The image shows the Inspector window for a transition named "success". The window has a blue title bar with the text "Inspector" and a maximize icon. Below the title bar, there is a text field for "Transition Name:" containing the value "success". Underneath, there is a section titled "Gated Transition" with a checkbox labeled "[Check user's groups before proceeding]" which is currently unchecked. Below this checkbox is a dropdown menu for "Required Group:" with the value "AUDITOR" selected. Further down, there is another checkbox labeled "[Hand-off flow run after this transition]" which is also unchecked. Below this is a text field for "Transition ROI Value:" containing the value "0.0". At the bottom, there is a section titled "Description" with a large empty text area for input.

図 46 – トランジションの Inspector

デフォルトでは、トランジション名はその発生源であるレスポンスの名前と同じですが、必要に応じて変更できます。

4. トランジションの名前を変更する場合は、[Name] ボックスに新しい名前を入力します。
5. 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場合は、[Gated Transition] エリアで次の操作を行います。
  - [Check user's groups before proceeding] チェックボックスをオンにします。
  - [Required Group] ドロップダウンリストから、ユーザーがフローの実行を続けるためにメンバーとして所属する必要があるグループを選択します。
6. 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場合は、[Hand-off flow run after this transition] チェックボックスをオンにします。
7. フローの価値においてトランジションの発生を考慮する場合は、[Transition ROI Value] の横に値を入力します。
8. [Description] ボックスに説明を入力します。

注: ここには、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の処理の説明を入力します。入力した内容は、Central の [Flow Library] タブの [Results Summary] エリアに表示されます。



説明には、可変情報を格納するフロー変数を使用できます。たとえば、**servername** フロー変数に名前が格納されるサーバーを指定する場合、「Server \${servername} is available for connection」と入力できます。

9. [OK] をクリックし、作業結果を保存します。

## トランジションの位置の変更

フローを整理するときや積み重なっているトランジションを分けるときなど、トランジションの移動や形状変更が必要になる場合があります。クリックとドラッグの2つの操作で、トランジションの移動と形状変更、およびトランジション名の移動を行うことができます。

### 曲線定義点を追加してトランジションの形状を変更するには

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。

次のフローを例にして、2番目の **Iterator** ステップの **success** トランジションをほかのトランジションと交差しないように移動するとします。

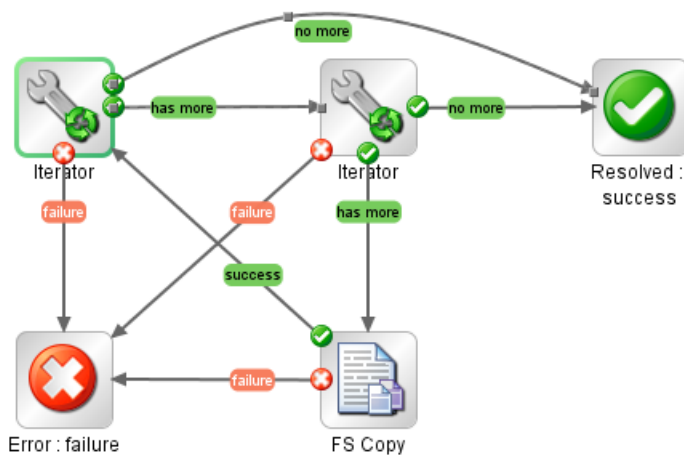


図 47 - SHIFT を押しながらトランジションをドラッグする前

上記の手順に従って、トランジションを新しい位置にドラッグします。曲線定義点が追加されていることに注意してください。

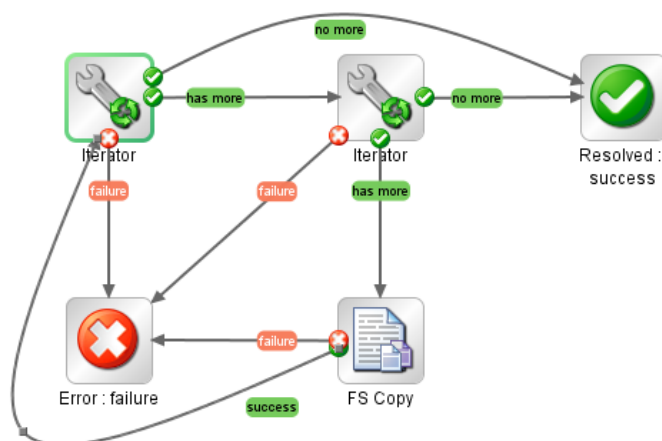


図 48 - SHIFT を押しながらトランジションをドラッグした後

### 曲線定義点を削除するには





- 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には

- 名前をクリックし、適切な場所にドラッグします。

## リターンステップ

リターンステップは、フローのレスポンスをステップとして表したものです。リターンステップには4つの種類があり、フローがとり得る主な4つの終了状態を表します。

- Resolved: success 
- Diagnosed: diagnosed 
- No Action Taken: 
- Error: failure 

これらのリターンステップはシールされており、編集はできません。

### リターンステップをフローに追加するには

1. フロー図のツールバーから、使用するリターンステップのアイコンをフローキャンバスにドラッグします。



図 49 - リターンステップのアイコン

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。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、**Error: failure** 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、**Error: failure** ステップに対してこの結果を反映するような新しいレスポンス（たとえば **Error: threshold not met**）を作成することが可能です。

2. リターンステップのレスポンスを変更する場合は、フロー図内でリターンステップを右クリックし、[Select Response] をポイントしてから目的のレスポンスをクリックします。

もしくは

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場合は、対象のリターンステップを右クリックし、[Select Response] をポイントして [Add New Response] をクリックしてから新しいレスポンスの名前を指定します。

## 入力: オペレーションへのデータの提供

入力は、フローのステップが必要なデータを取得する方法、およびデータが取得されるタイミングを、フロー作成者が指定するための手段です。

**Network Check** フローを例にとると、開始ステップのオペレーションはサーバーに対して ping を実行するので、ping 対象のサーバーの IP アドレスを必要とします。そのため、入力には IP アドレスを指定します。

IP アドレスを提供するための入力は、次の要素に対して作成できます。

- フロー（フローのプロパティで指定する）。

入力値をフロー変数に割り当てることができます。フロー変数は、フロー全体を通じて参照可能です。この方法は、フローの外部から提供されるデータに適しています。

フローの入力は、Central のグラフに記録するためのデータをレポートするステップに対してデータを提供する場合に大変便利です。このようなステップは主にリターンステップで、通常はフロー変数からデータを取得します。フロー変数には、フローの入力またはステップの出力から値を割り当てることができます。

- オペレーションと関連付けられているステップ。

ステップの入力は、ステップのオペレーションの特定の用途に対して値を提供します。ステップの入力の値には、フローの先行ステップから取得したデータを割り当てることができます。

- オペレーション自体。

オペレーションはステップを作成する際のベースである一方、各ステップはそのベースとなったオペレーションを実行します。したがって、ステップの入力は、ステップ作成のベースとなったオペレーションの入力を基にして作成されます。そのため、オペレーションの入力に割り当てられたデータは、そのオペレーションから作成されたステップで使用できます。ただしフロー内では、特定の場合にオペレーションの入力に使用される値は、ステップの入力が提供します。ステップの入力の値の取得先は変更することが可能です。たとえば、**flowvariable1** から値を取得するオペレーションの入力があるとし、このオペレーションからフローのステップを作成するときに、ステップの値の取得先として **flowvariable1** ではなく **flowvariable2** を指定できます。このように、ステップの入力は各フローの必要性に合わせて変更できます。

そこで、再度 **Network Check** フローを例にしてみます。ping オペレーションは、ターゲットの IP アドレスを **host** 入力から取得します（この入力は、オペレーションではユーザープロンプトとして定義されています）。このオペレーションからフローにステップを作成するときに、ステップの **host** 入力へのデータ割り当てを特定の値、つまり当該フローのターゲットの IP アドレスに変更できます。

フローの入力の値をフロー変数に割り当てて、ステップの入力の値をそのフロー変数から取得するように設定できます。ステップの入力のデータソースとなるフロー変数が定義されていない場合は、別のデータソース（ユーザープロンプト、固定値、ユーザー資格情報など）を設定することも可能です。

**注:** ステップの入力に対して有効なデータソースを指定していない場合、デフォルトではユーザープロンプトが使用されます。この場合は当然、フローが処理を実行するために必要な情報をユーザーが提供できることが必要です。

#### 入力の作成

#### 入力の値の取得方法の指定

#### 入力とフロー実行のスケジュール

#### 入力の削除

## 入力の作成

入力を作成する対象がフロー、ステップ、オペレーションのどれであるかにかかわらず、作成手順は基本的に同じです。

### 入力を作成するには

1. フローまたはオペレーションの [Properties] シートを開くために、**リポジトリ** ペインで目的のフローに移動してそのフロー名をダブルクリックし、**作成** ペインの下部にある [Properties] タブをクリックします。
2. [Inputs] タブで、[Add Input] をクリックし、入力の名前を指定します。

**警告:** 入力の名前として「service」および「sp」は使用しないでください。これらの名前を付けると、特定の状況で実行されるフローでエラーが発生する可能性があります。詳細については、『HP Operations Orchestration Software Central ユーザーガイド』（Central\_UsersGuide.pdf）または『HP OO SDK Guide』（SDKGuide.pdf）を参照してください。

新規作成した入力、それぞれ個別の行に表示されます。

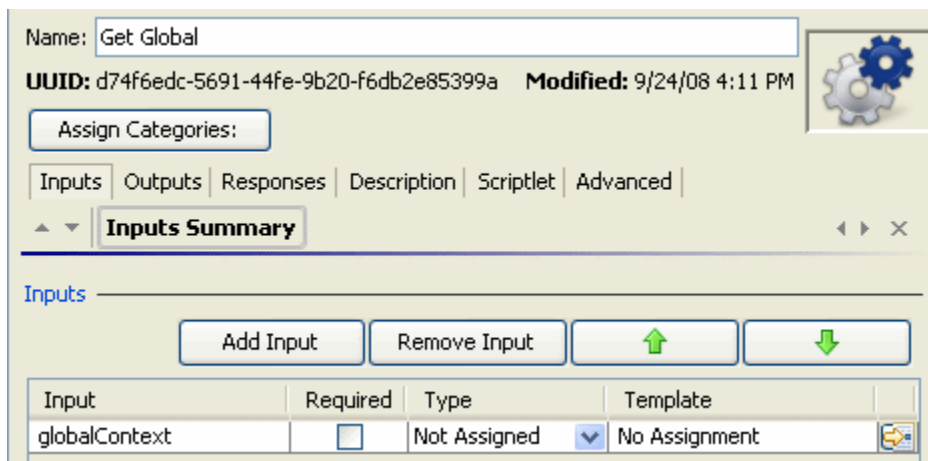


図 50 – フローの [Properties] シートと新しい入力

最後の欄の列見出しはステップの Inspector とは多少異なりますが、入力の同じ要素を表しています。オペレーションとステップのどちらの場合も、最後の列には入力用の値のソースが表示されます（フロー変数から値が割り当てられない場合）。

3. 入力の名前を変更する場合は、目的の行の [Input] 列をダブルクリックし、新しい名前を入力します。
4. ステップ（またはフロー内の後続のステップ）が機能するためにその入力のデータが必須である場合、[Required] チェックボックスをオンにします。

入力を作成する際に最も考慮すべき項目の 1 つが、その値の割り当ての **タイプ**、つまりどのような種類の値を割り当てるかです。値の割り当てのタイプはここで変更できますが、入力エディターで変更することも可能です。

5. 値の割り当てのタイプを変更するには、目的の行の [Type] 列をクリックし、ドロップダウンリストから次のいずれかの割り当てタイプを選択します。

- **Single Value**
- **List of Values**
- **Credentials**
- **Not Assigned**

**Not Assigned** タイプは、オペレーションまたはフローに使用する場合のみ、この列で選択できます。

入力の値のソースを変更する場合は、入力エディターを開きます。

6. 入力エディターを開いて入力がデータを取得する方法やその他の入力の動作を定義する場合は、目的の入力の行の端にある右向き矢印 (👉) をクリックします。

入力のデータソースを指定する方法の詳細については、「[入力の値の取得方法の指定](#)」を参照してください。

## 入力の値の取得方法の指定

フローまたはオペレーションの [Properties] シート、あるいはステップの Inspector の [Inputs] タブで、入力のタイプについて最小限の定義を行いました。そのほかに入力の値の取得方法を定義しなかった場合、オペレーションまたはステップは、入力と同じ名前のフロー変数（定義されている場合）か、ユーザープロンプトに対するフローユーザーのレスポンスのいずれかから入力の値を取得します。ただし、フローの使い方によっては、フローの入力（およびそのステップの入力）の取得先としてユーザープロンプトを利用できない場合も数多くあります。次に示すのはその一例です。

- フローを自動的に実行する
- システムアカウントから、またはフローを実行するシステムにログインしているユーザーの資格情報から、資格情報を取得する
- 先行ステップの結果から、またはフロー変数から入力値を取得する
- フローをスケジュール化する
- Central の外部からフローを実行する

入力エディターでは、入力がどのようにデータを取得するかをさまざまな方法で指定できます。

### 入力のデータソースを定義するには

1. 入力エディターを開いて入力がデータを取得する方法やその他の入力の動作を定義するには、[Inputs] タブで、目的の入力の行にある右向き矢印 (→) をクリックします。  
入力エディターが次のような状態で開きます。

図 51 – 入力エディター

ここでは [Inputs] タブと同じく次の操作ができます。

- 入力の名前とタイプを変更する。
- 入力を必須（または任意）として設定する。

入力に使用できる値は、基本的に次の 2 種類です。

- 値 (Single Value または List of Values のいずれか)
- Credentials

入力タイプを選択した後に、入力の値の取得先を指定します。入力タイプに応じて使用できるデータソースが異なるので、入力エディターに表示されるオプションも変わります。

2. 入力が値を取得する方法を選択するには、[Input Type] リストから次のいずれかを選択します。

- **Single Value**
- **List of Values**

マルチインスタンスステップまたは Iterator ステップに複数の値を提供する入力の場合はこの入力タイプを選択します。

- **Credentials**
- **Not Assigned**

次に、入力の値の取得先を指定します。

3. 入力にフロー変数の値を割り当てる場合は、[Assign from Variable] リストボックスで、適切なフロー変数を選択するか、またはフロー変数の名前を入力します。

デフォルトでは、入力の名前がフロー変数の名前としてボックスに表示されます。

もしくは

フロー変数を入力の名前の取得先として使用しないことを明示的に指定する場合は、[<not assigned>] を選択します。

注:

- 入力に割り当て可能な値を持つフロー変数の中に、予約済みフロー変数があります。予約済みフロー変数の値は、フロー内のどの場所からでも常に参照できます。予約済みフロー変数の詳細については、「[予約済みフロー変数](#)」を参照してください。
- [Input Type] が [Not Assigned] のときに、[Assign from Variable] ボックスで [Not Assigned] を選択すると、オペレーションが失敗します。



**ヒント:** 入力に特定の値を割り当てたときに [Assign from Variable] ボックスの入力の名前をそのままにしておいた場合、入力はそれと同名のフロー変数から値を取得しようとします。その結果、フローのスケジュールを作成する Central ユーザーが、([Scheduler] ダイアログボックスで) 入力に別の値を割り当てられるようになります。入力が同名のフロー変数から値を取得できないように設定すれば、Central ユーザーがフローをスケジュールリングするときに入力の名前に別の値を指定した場合でも、スケジュールされたフローの実行時には、フロー作成者が Studio で入力に割り当てた特定の値が使用されます。

[Assign from Variable] ボックスで [<not assigned>] を選択した場合、またはここで指定したフロー変数が存在しない、あるいはそのフロー変数に格納されている値がない場合は、[Otherwise] リストボックスで選択した方法で入力の値が取得されます。

4. 入力の値を暗号化する場合は、[Input Properties] の下にある [Encrypted] チェックボックスをオンにします。入力の値を暗号化すると、Central でフローを実行するときに値がアスタリスクで表示されます。

**重要:** 入力の割り当てが [Logged-in User Credentials] のときは、入力の値は常に暗号化されます。

5. 入力を必須にする場合は、[Required] チェックボックスをオンにします。

固定値、ユーザープロンプト、および [Use Previous Step Result] を指定した場合は、選択したドメインタームで入力の値を記録することが可能です。この値は Central でのフローの実行履歴に記録できます。つまり、診断や監査の目的でフローを使用する際にこの入力の値を利用できます。

値は特定の形式（電子メール、ファイル名、IP アドレス、英数字、電話番号など）に合わせる必要があります。

6. 入力の値の形式を検証する場合は、[Validation Format] ドロップダウンリストから目的の形式を選択します。
7. Central でのフローの実行履歴に入力の値を記録する場合は、[Record Under] ドロップダウンリストから、入力の値を記録する際のドメインタームを選択します。

入力エディターの [Input Data Flow] セクションと [Configuration] セクションは、選択した入力タイプに応じて変わります。入力のデータソースを指定するには、次のいずれかの手順に従います。

- [単一の固定入力値の作成](#)
- [フローユーザーの入力内容を基にした単一入力値の作成](#)
- [フローユーザーの選択内容を基にした単一入力値の作成](#)
- [固定入力値リストの作成](#)
- [フローユーザーの入力テキストを基にした入力値リストの作成](#)
- [フローユーザーの選択内容を基にした入力値リストの作成](#)
- [\[Previous Step's Result\] を基にした入力値の作成](#)
- [入力値への資格情報の割り当て](#)



8. 作業結果を保存します。
9. 入力エディターを閉じずに別の入力を編集するときは、[Inputs Summary] の横にある上向きまたは下向きの矢印 (▲ ▼) をクリックして、ステップの入力のリストをスクロールします。

## 単一の固定入力値の作成

### 単一の固定入力値を指定するには

1. 「[入力のデータソースを定義するには](#)」の手順が完了した後、[Otherwise] リストボックスで [Use Constant] を選択します。
2. [「Otherwise: Use Constant」 Configuration] セクションで、[Constant Value] ボックスに入力の値を入力します。  
もしくは  
次の形式でフロー変数参照を入力します。

`${flowvariablename}`



ヒント: null 値を指定するには、[Otherwise] リストボックスで [Use Constant] を選択した後、[Constant Value] ボックスをクリックしますが、何も入力しないでください。

## フローユーザーの入力内容を基にした単一入力値の作成

### ユーザーが入力したテキストから取得する固定入力値を指定するには

1. 「[入力のデータソースを定義するには](#)」の手順が完了した後、[Input Type] リストボックスで [Single Value] を選択します。
2. [Otherwise] リストボックスで [Prompt User] を選択します。
3. [「Otherwise: Prompt Use」 Configuration] セクションの [Prompt For] で [Text] を選択します。
4. [User Message] ボックスに、フローユーザーに対して入力するデータの種類の説明するプロンプトのテキストを入力します。

## フローユーザーの選択内容を基にした単一入力値の作成

### ユーザーがリストから選択した内容から単一入力値を指定するには

1. 「[入力のデータソースを定義するには](#)」の手順が完了した後、[Input Type] リストボックスで [Single Value] を選択します。
2. [Otherwise] リストボックスで [Prompt User] を選択します。
3. [「Otherwise: Prompt User」 Configuration] セクションの [Prompt For] で [Selection] を選択します。
4. [List Source] リストボックスで、次のいずれかを選択します。

- **Selection List**

ユーザーに提示する事前定義済みリストを選択できます。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](#)」を参照してください。

- [Named] リストボックスで、ユーザーに提示するリストを選択します。

- **Domain Term**

ドメインタームは、手動で作成した選択リストに比べてなんらかの利点を持つ特殊な選択リストです。たとえば、ダッシュボードでのレポート作成を容易にする、より厳密にフローを監査できる、などのメリットがあります。

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。

ドメインタームの詳細については、「[ダッシュボードグラフ用のドメインターム](#)」を参照してください。

- [Named] リストボックスで、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。

- **Flow Variable**

フロー変数のコンテンツをリストとしてユーザーに提示します。

1. [「Otherwise: Prompt User」 Configuration] エリアの [List Source] リストでも [Flow Variable] を選択します。
  2. [Named] 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。
  3. [Source Delimiter] ボックスに、リスト内の各要素を区切る文字または文字列を入力します。
5. [User Message] ボックスに、フローユーザーに対して入力するデータの種類の説明するプロンプトのテキストを入力します。

## 固定入力値リストの作成

単一入力値のリストを作成することにより、1つのオペレーションを複数のターゲットに対して実行できるようになります。たとえば、マルチインスタンスステップとして構成したステップでこの方法を用いて、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可能です。

### 固定入力値リストを指定するには

1. 「[入力のデータソースを定義するには](#)」の手順が完了した後、[Input Type] リストボックスで [List of Values] を選択します。
2. [Input Properties] の下にある [Input Delimiter] ボックスに、リスト内の各要素を区切る文字または文字列を入力します。
3. [Otherwise] リストボックスで [Use Constant] を選択します。
4. [「Otherwise: Use Constant」 Configuration] セクションの [Constant Value] ボックスで次の操作を行います。

- 入力に使用する値を入力します ([Input Delimiter] ボックスで指定した文字または文字列で各値を区切ります)。

この入力を使用するオペレーションが想定どおりに実行されるようにするため、リストの各値を区切る文字は [Input Delimiter] に入力したとおりの文字（空白の有無を含む）を使用してください。

もしくは

1式以上のフロー変数参照を次の形式で入力します。

```
${flowvariablename1}<delimiter>${flowvariablename2}<delimiter>${flowvariablename3}
```

それぞれに複数のリストが含まれるフロー変数参照のリストから、つまり値とフロー変数（単一値または値リストが含まれる）の組み合わせのリストから、特定のリストを指定できます。唯一の条件は、リスト項目を区切る文字として、フロー変数の中か外かにかかわらず必ず [Input Delimiter] ボックスで定義した区切り文字を使用することです。



例として、次の場合を考えてみます。

- 入力 **host** にはパイプ文字 (|) を区切り文字として使用する
- **flowvariableA** の内容: 10.51.0.5|12.225.8.71
- **flowvariableB** の内容: 220.220.3.9|10.51.110.12

このとき、[Constant Value] ボックスには次のように入力できます。

```
10.2.0.200|18.35.100.7|${flowvariableA}|${flowvariableB}
```

この場合、この **host** 入力を使用するオペレーションは、次の IP アドレスを持つマシンで実行できます。

- 10.2.0.200
- 18.35.100.7
- 10.51.0.5
- 12.225.8.71
- 220.220.3.9
- 10.51.110.12

マルチインスタンスステップのリストの入力値をフロー変数に割り当てる方法については、「[マルチインスタンスステップとのデータの移動](#)」を参照してください。

## フローユーザーの入力テキストを基にした入力値リストの作成

入力用の値リストを入力するよう求めるユーザープロンプトを使用する場合、ユーザーの値リストでは入力で指定した区切り文字を使用する必要があります。

### ユーザーが入力したテキストから取得する固定入力値リストを指定するには

1. 「[入力のデータソースを定義するには](#)」の手順が完了した後、[Input Type] リストボックスで [List of Values] を選択します。
2. [Input Properties] の下にある [Input Delimiter] ボックスに、リスト内の各要素を区切る文字または文字列を入力します。
3. [Otherwise] リストボックスで [Prompt User] を選択します。
4. 「[Otherwise: Prompt User] Configuration」セクションの [Prompt For] で [Text] を選択します。
5. [User Message] ボックスに、フローユーザーに対して入力するデータの種別を説明するプロンプトのテキストを入力します。

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、正しい区切り文字または文字列を使用するよう注意書きも付け加えてください。区切り文字として使用されていない場合に空白を挿入すると失敗の原因となるので、特に注意が必要です。

## フローユーザーの選択内容を基にした入力値リストの作成

入力をリスト値として指定した場合、ユーザープロンプトでユーザーに提示するリストは複数選択リストになります。この方法は、フローユーザーがフローのターゲットとなるマシンを選択するときなどに、リストの中から選択できるようにするのに便利です。

### ユーザーがリストから選択した内容を基にした入力値リストを指定するには

1. 「[入力のデータソースを定義するには](#)」の手順が完了した後、[Input Type] リストボックスで [List of Values] を選択します。
2. [Input Properties] の下にある [Input Delimiter] ボックスに、リスト内の各要素を区切る文字または文字列を入力します。
3. [Otherwise] リストボックスで [Prompt User] を選択します。
4. 「[Otherwise: Prompt User] Configuration」セクションの [Prompt For] で [Selection] を選択します。
5. [List Source] リストボックスで、次のいずれかを選択します。

- **Selection List**

ユーザーに提示する事前定義済みリストを選択できます。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](#)」を参照してください。

[Named] リストボックスで、ユーザーに提示するリストを選択します。

- **Domain Term**

ドメインタームは、手動で作成した選択リストに比べてなんらかの利点を持つ特殊な選択リストです。たとえば、ダッシュボードでのレポート作成を容易にする、より厳密にフローを監査できる、などのメリットがあります。

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。

ドメインタームの詳細については、「[ダッシュボードグラフ用のドメインターム](#)」を参照してください。

[Named] リストボックスで、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。

- **Flow Variable**

フロー変数のコンテンツをリストとしてユーザーに提示します。

1. [「Otherwise: Prompt User」 Configuration] エリアの [List Source] リストでも [Flow Variable] を選択します。
2. [Named] 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。
3. [Source Delimiter] ボックスに、リスト内の各要素を区切る文字または文字列を入力します。
6. [User Message] ボックスに、フローユーザーに対して入力するデータの種類の説明するプロンプトのテキストを入力します。または、リストを含むプロンプトの場合は、（可能であれば）複数の項目を選択できることを説明するテキストを入力します。

## **[Previous Step's Result] を基にした入力値の作成**

この場合のデータソースは、先行ステップに対して作成したフィルターが適用された後の当該ステップの結果です。このデータソースタイプには、詳しい設定はありません。

### **[Previous Step's Result] を入力値のソースとして指定するには**

1. 「[入力のデータソースを定義するには](#)」の手順が完了した後、[Input Type] リストボックスで、入力に複数の値を使用するかどうかに応じて [Single Value] または [List of Values] を選択します。
2. [Otherwise] リストボックスで [Use Previous Step Result] を選択します。
3. 入力値が複数值の場合は、[Input Properties] の下にある [Input Delimiter] ボックスに、リスト内の各要素を区切る文字または文字列を入力します。

指定した入力の区切り文字が [Previous Step's Result] で使用されている区切り文字と一致しない場合、オペレーションは失敗します。

## **入力値への資格情報の割り当て**

ユーザープロンプト以外にも、ログインユーザーの資格情報またはシステムアカウントを使用して、入力値に資格情報のセットを割り当てることができます。

ログインユーザーの資格情報は、フローを実行しようとしているログインアカウントにフローのタスクを実行するために必要なアクセス権がある場合に、オペレーションに対する資格情報として機能します。

システムアカウントは、一連のユーザー資格情報への参照です。システムアカウントを利用すると、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作成方法については、「[システムアカウント: 安全な資格情報](#)」を参照してください。

### ユーザー資格情報を入力値のソースとして指定するには

1. 「[入力データのソースを定義するには](#)」の手順が完了した後、[Input Type] リストボックスで [Credentials] を選択します。
2. [Otherwise] リストボックスで、次のいずれかを選択します。
  - **Prompt User**  
[Prompt User] を選択したときは、[「Otherwise: Prompt User」 Configuration] セクションの [User Name and Password Message] ボックスに、ユーザーに表示するプロンプトのメッセージを入力します。
  - **System Account**  
[System Account] を選択したときは、[「Otherwise: System Account」 Configuration] セクションの [Account Named] リストボックスで、オペレーションの資格情報に使用するシステムアカウントを選択します。
  - **Logged-in User Credentials**  
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。

## 入力とフロー実行のスケジュール

フローの実行を自動的に開始するスケジュールを作成するには、開始や入力でユーザーが介入することなくフローを実行できることが必要です。ユーザープロンプトによる入力がある場合は、特定の値に変更するか、または特定の値が割り当てられているフロー変数から値を取得できるようにする必要があります。

さらに、フローのスケジュールを複数作成した場合、Central のスケジュールボックスで各スケジュールのフロー変数に異なる値を格納できます。たとえば、**Network Check** フローに対して複数のスケジュールを作成し、各スケジュールの host 入力に個別のサーバー IP アドレスを指定することにより、スケジュールがそれぞれ異なるサーバーで実行されるようにすることが可能です。

フローのスケジュールを作成する方法については、Central のヘルプを参照してください。

### スケジュールごとに個別の入力値を定義できるようにするには

- 入力エディターで、入力データのソースとして [Use Constant] を選択し、[Constant Value] ボックスに次の構文を入力してフロー変数を参照します。  
`${flowvariablename}`  
入力にデータソースを割り当てる方法については、「[入力の値の取得方法の指定](#)」を参照してください。

## 入力の削除

### ステップから入力を削除するには

1. 目的のステップの Inspector、あるいはフローまたはオペレーションの [Properties] シートを開きます。
2. [Inputs] タブの入力のリストで、削除する入力を強調表示して、[Remove Input] をクリックします。

## フロー変数: 再利用可能なデータの作成

フロー変数は、フロー内やフロー間でデータを移動するときに利用できます。その場合、必要なデータを格納しているフロー変数を以下の要素内で参照します。

- フロー内の個々のステップ、またはフロー変数を作成したフローとは別のフロー。  
フロー変数を作成すると、そのフロー変数（および格納されるデータ）を別のフロー、ステップ、またはオペレーションの入力で参照できます。
- フロー、ステップ、トランジションの説明。  
たとえば、**Ping Latency** 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。
- 並列分岐ステップのレーン内。  
レーンステップは、同一レーン上の、または並列分岐ステップより前の先行ステップによりフロー変数に書き込まれた値を使用できます。  
ただし、あるレーンのステップによってフロー変数に書き込まれた値を別のレーンのステップが使用することはできません。
- レスポンスルールを使用してテストするデータの一部。  
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し、そ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、その格納先のフロー変数を参照することによりその値にアクセスできます。

## フロー変数の作成


デフォルトでは、ステップまたはフローの入力を作成すると同名のフロー変数が作成され、入力の値はそのフロー変数に格納されます。ただし、入力の値の格納先または取得先として別のフロー変数を指定することも可能です。この機能は、あるサーバーを実行対象とする複数のオペレーションを追加し、最初のオペレーションの実行時にユーザーにサーバーの入力を求める場合に利用できます。同じ名前を入力を持つ後続のオペレーションではすべて、自動的にそのサーバー名が使用されます。

フロー変数は、オペレーション、ステップ、フローの入力または結果から作成できます。どの場合でも作成方法は同じです。

**注:** 以下の手順は、入力およびステップの結果の作成方法について基本的な知識があることを前提にしています。

- 入力の作成方法については、「[入力: オペレーションへのデータの提供](#)」を参照してください。
- オペレーションの出力の作成方法については、「[オペレーションの出力](#)」および「[出力および結果のフィルタリング](#)」を参照してください。
- ステップの結果の作成方法については、「[ステップの結果](#)」および「[出力および結果のフィルタリング](#)」を参照してください。

### 入力からフロー変数を作成するには

1. オペレーションまたはフローの [Properties] シート、またはステップの Inspector を開きます。
2. [Inputs] タブで、入力を選択するか、または新規作成します。
3. 入力の行にある右向き矢印 (  ) をクリックします。  
入力エディターが開きます。
4. エディターの [Assign to Variable] ボックスで、値を割り当てる変数を指定します。
5. 作業結果を保存します。

### 結果からフロー変数を作成するには

1. オペレーションの [Properties] シート、またはステップの Inspector を開きます。
2. [Results] タブで、結果を新規作成します。
3. 新しい結果の行で、[Assign To] の下にある [Flow Variable] を選択します。
4. [Name] で、新しいフロー変数の名前を指定します。
5. [From] で、フロー変数の値のソースを指定します。
6. 必要に応じて、結果に使用するフィルターを 1 個以上、作成します。  
フィルターの作成方法については、「[出力および結果のフィルタリング](#)」を参照してください。
7. 作業結果を保存します。

## ローカルフロー変数の作成

スクリプトレットタイプのオペレーションでは、スクリプトレット内で次の構文のコマンドを使用してローカルフロー変数を作成します。

```
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。

スクリプトレットの作成方法については、「[スクリプトレット](#)」を参照してください。

## 予約済みフロー変数

自分が作成したフロー変数に加えて、次に示す予約済みのグローバルフロー変数をすべてのフローや実行処理でいつでも利用できます。これらのフロー変数は、特にスクリプトレット内で使用する場合に便利です。

### 予約済みフロー変数

予約済みフロー変数	設定可能な値、初期値
Boolean	True または False、一般的なブール型の入力
Boolean – False Default	True または False
CA Unicenter Log Type	CA Unicenter で使用されるログのタイプ: CB、RS、または LOG
DCDiag Tests	DCDiag テストの名前: All、Intersite、Connectivity、Replications、NCSecDesc、CutoffServers、Advertising、KnowsOfRoleHolders、FsмоCheck、RidManager、MachineAccount、OutboundSecureChannels、ObjectsReplicated、frssysvol、systemlog、Kccevent

予約済みフロー変数	設定可能な値、初期値
Delete or Zip	Delete または Zip
Event Log Searcher	*, System、Security、Application
Event Severity	Error、AttentionRequired、Warning、Information、All
Exchange Mailbox Permissions	Exchange メールボックスのアクセス許可: Delete、Read permissions、Change permissions、Take ownership、Full access
FTP Type	Binary または Ascii
Host Type	Windows Log Rotator (Local または Remote) で使用: Local Host または Remote Host
Host Type (Deprecated)	local または remote
HP Service Manager Resolution Fix Types	Permanent または temporary
HP Service Manager Ticket Types	Change、Problem、Incident
Hyper-V Boot Order	Hyper-V マシンを起動できるデバイス: Floppy、CD、IDE、PXE
Hyper-V Controller Position	Hyper-V VM の VHD 用 IDE コントローラの位置: 0 または 1
Hyper-V Controllers	Hyper-V 仮想マシン用 IDE コントローラ: IDE コントローラ 0 または IDE コントローラ 1
IIS Site Status	Running または Stopped
Job History Status	Backup Exec Log Analyzer 用: All、Successful、Failed、Completed With Exceptions
Linux Log Severity	DEBUG、INFO、NOTICE、WARNING、ERR、CRIT、ALERT、EMERG
LinuxLogRotatorOption	Archive または Delete
List Type	BackupExec Log Analyzer 用 (CAL または AAL) : ClearedAlertList または ActiveAlertList
Local Or Remote	Local Host または Remote Machine
Mail Body Type	html または text
Microsoft Exchange Version	3003 または 2007
NAS Diagnostics	BackupExec Log Analyzer 用 (CAL または AAL) : Memory Troubleshooting、NAS Detect Device Boot、NAS Device File System、NAS Duplex Data Gathering、NAS Flash Storage Space、NAS Interfaces、NAS Module Status、NAS OSPF Neighbors、NAS Routing Table、NAS Topology Data Gathering、Hardware Information
NetDiag Tests	All、Autonet、Bindings、Browser、DcList、DefGw、DNS、DsGetDc、IpConfig、IpLoopBk、IPSec、IPX、Kerberos、Ldap、Member、Modem、NbtNm、Ndis、NetBTTransports、Netstat、NetWare、Route、Trust、WAN、WINS、Winsock

予約済みフロー変数	設定可能な値、初期値
NNM – Lifecycle State	Network Node Manager でサポートされるライフサイクル状態: Registered、In Progress、Completed、Closed
NNM – Node Management Modes	HP Network Node Manager でノードに設定できる管理モード: Out of Service、Managed、Not Managed
NNM – Priority	HP Network Node Manager でサポートされる優先度: None、Low、Medium、High、Top
Notification Option	ヘルスフローの通知オプション: Email、Display、Write to File、None
Protocols	Remote コマンド実行オペレーションでサポートされるプロトコル: ssh、ROSH、Telnet、GlobalShell、RSH、local、WMI
Radia Priorities	Must、Should、May、MayNot、ShouldNot、MustNot
Remedy Case Statuses	New、Assigned、Work In Progress、Pending、Resolved、Closed
Remedy Urgency	High、Medium、Low、Urgent
Remote Or Local	remote または local
RemoteCopyProtocols	2つのリモートホスト間でファイルをコピーする際に使用するプロトコル: local、scp、ftp、sftp
SAS Version	sas7 または sas6
Scp copy Action	To または From
Script Languages	vbscript または jscript
SE-AddOrRemove	Add または Remove
SE-VisibilityOptions	ホスト HBA ポートに対するアレイ / ボリュームの可視性: All、Visible、NonVisible
Selection List Delete Or Zip	Windows Log Rotator フローでの選択に使用: LocalDelete、LocalZip、RemoteDelete、RemoteZip
Service Desk Object	Servicecall、Incident、Change、Workorder、Problem、Person、Workgroup、Organization、Service、ServiceLevelAgreement、Project、Configuration、MainCon
Service Status	Running、Stopped、Paused
SFTP Operations	cd、ls、pwd、lpwd、lls、chgrp、chmod、chown、compression、symlink、get、lcd、lstat、mkdir、put、rm、stat、exit
Site Scope Acknowledgement Type	Enable または Disable
SMTP Greetings	HELO または EHLO
SNMP Varbind Type	TIMETICKS、COUNTER、GAUGE、INTEGER、OID、STRING、ADDRESS、HEX

予約済みフロー変数	設定可能な値、初期値
Sql Authentication	Windows または Sql
SQL Database Type	Oracle、MSSQL、Sybase、Netcool、DB2
SQL Reporting Server Format	HTML 3.2、HTML 4.0、HTMLLOWC、MHTML、CSV、XML
Storage Redundancy Options	Redundant Parity Based Storage または Redundant Mirrored Storage
String Comparator Match Type	Exact Match、Contains、Contains Once、Does Not Contain、Match All Words、Match No Words、RegEx
Tomcat Actions	Start、Stop、reload、undeploy
Transport Layer Protocols	TCP または UDP
User Session Control Levels	Active Directory のユーザーのセッションの制御レベル: EnableInputNotify、EnableInputNoNotify、EnableNoInputNotify、 EnableNoInputNoNotify
VHD Type	仮想ハードディスクのタイプ: Fixed または Dynamic
Virtualization Technologies	Virtualization Accelerator Pack のフローを実行する際にサーバーで 実行する仮想化テクノロジーをユーザーが選択できる選択リス ト: VMWare Virtual Infrastructure、Microsoft Hyper-V、Citrix XenServer
VMReference Type	VMWare ESX/VC 統合の仮想マシンを参照する方法: INVENTORYPATH、NAME、IP、HOSTNAME、UUID、VMID
VMWare Actions	Start、Stop、Enumerate、Status
VMWare Managed Object Types	VirtualMachine、Network、ComputeResource、ResourcePool、 HostSystem、DataStore
VMWare Virtual Machine Power States	poweredOn、poweredOff、suspended
Windows Access Control	Allow または Deny
Windows Clustering States	Online または Offline
Windows File System Inheritance Propagation	This folder、subfolders and files、This folder and subfolders、This folder and files、This folder only、Subfolders only、Files only、This object only
Windows File System Permissions	Append Data、Change Permissions、Create Directories、CreateFiles、 Delete、Delete Subdirectories and Files、Execute File、Full Control、 List Directory、Modify、Read、Read and Execute、Read Attributes、 Read Data、Read Extended Attributes、Read Permissions、Take Ownership、Traverse、Write、Write Attributes、Write Data、Write Extended Attributes
Windows Registry Value Type	REG_SZ、REG_EXPAND_SZ、REG_BINARY、REG_DWORD、 REG_MULTI_SZ
Windows Script Languages	VBScript または Jscript



予約済みフロー変数	設定可能な値、初期値
Windows Service Startup Mode	Boot、System、Auto、Manual、Disabled
Windows Service Status	Running、Stopped、Paused
WMI Query Format	csv または text
Yes No	Yes または No

## 同時実行: 複数スレッドの並列実行

フロー内では、*同時実行*（*並列実行*とも呼びます）を有効にできます。その効果について、いくつか例を挙げて説明します。

- ある問題を診断するために、データベースサーバー、アプリケーションサーバー、Web サーバー、ルーターのそれぞれにヘルスチェックを実行する必要があります。この 4 つのヘルスチェックを同時に実行できれば、診断にかかる時間を短縮することが可能です。
- ソフトウェアアップグレードを 100 台のサーバーにインストールしようとしています。25 台のサーバーに同時にインストールすることができれば、一度に 1 台ずつアップグレードする場合と比べて 4 分の 1 の時間で全サーバーをアップグレードできます。
- 診断フローのステップの 1 つでは、トラブルチケットの作成、または電子メールの作成と送信を行います。このステップがチケットまたは電子メールを作成している間にフローの後続ステップを中断することなく実行できれば、解決時間を短縮できます。

HP OO では、複数の方法でこのような並列実行を実現できます。

- フロー全体を並列に実行するには、次のいずれかの操作を行います。
  - Central の [Schedule] タブで、フローの同時実行をスケジューリングします。
  - URL でフローの並列同時実行を指定します。
- フロー内で、並列処理を行うステップを使用できます。

並列処理を行うステップは、次の 3 種類です。

- 並列分岐ステップ**。同時に実行される連続ステップが複数含まれます。

並列分岐ステップは、種類の異なる個々の連続ステップを同時に実行することを目的としています。それぞれの連続ステップをレーンと呼び、フロー図でも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言います。複数のレーンステップで構成されるレーンはサブフローによく似ているため同じようなものと考えられそうですが、データの受け渡しなどいくつかの点で、レーンはサブフローまたはフローとは大きく異なります。

種類の異なる複数のヘルスチェックを同時に実行しようとする場合、たとえば 4 つのレーンを持つ並列分岐ステップを作成します。あるレーンではデータベースサーバーでヘルスチェックを実行するために必要なレーンステップを作成し、次のレーンではアプリケーションサーバーでヘルスチェックを実行するために必要なステップを作成する、という方法が可能です。

詳細については、「[並列分岐ステップ](#)」を参照してください。

- **マルチインスタンスステップ**。ステップのオペレーションの入力の値として提供された値リストのメンバーを同時に実行します。マルチインスタンスステップには**スロットル**を適用できます。つまり、同時に処理できる値の数を制限することが可能です。

多数のサーバーを同時にアップグレードする例について、アップグレードを実行するサブフローに関連付けられているステップがあり、このステップは当該サブフローの実行対象のサーバーを特定するターゲット入力を受け取るとします。そこで、このステップをマルチインスタンスステップに変換してから、サブフローによりアップグレードを実行するサーバーのリストを提供します。すると、このステップはこれらすべてのサーバーに対して同時にチェックを実行します。

サブフローを同時に実行する対象のサーバーが多すぎてインフラストラクチャの機能低下が生じる場合は、たとえば同時に処理するターゲット入力を 25 のみに指定することで、マルチインスタンスステップを制限できます。

詳細については、「[マルチインスタンスステップ](#)」を参照してください。

- **ノンブロッキングステップ**。ノンブロッキングステップが実行中でも、フローを停止せずにその後続のステップを実行できます。

トラブルチケットをオープンするステップの場合、このステップをノンブロッキングステップにすることで、ステップがチケットを作成している間もフローを続行できます。

詳細については、「[ステップのノンブロッキング化](#)」を参照してください。

## 並列分岐ステップ

並列分岐ステップは、同時に実行されるステップ配列のセットです。それぞれの配列を**レーン**と呼び、レーン内のステップを**レーンステップ**と呼びます。特定のレーンに使用されるフロー変数の集合をまとめて**レーンコンテキスト**と呼びます。

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。対照的に、マルチインスタンスステップでは単一入力に複数のバリエーションがある場合も、各インスタンスが同じ処理を実行することに注意してください。

たとえば、パッチをインストールし、適切な IT 担当者にこのインストールについての電子メールを送信する場合に、並列分岐ステップを使用するとします。

- 一方のレーンには、パッチをインストールするフローから作成したステップを含めます。
- もう一方のレーンでは電子メールを送信します。

並列分岐ステップを作成する際には、次のことにも注意します。

- レーン間でデータを交換することはできません。各レーンのステップが使用できる値は、並列分岐ステップ（のすべてのレーン）の開始時に使用可能だった値のみです。
- 並列分岐ステップの出力 / レスポンスは、**すべてのレーンが完了したことを示す 1 種類しかありません**。
- 2 つのレーンのステップが 1 つのフロー変数に値を書き込んだ場合、後に完了したレーンにより書き込まれた値が、並列分岐ステップが完了した後にフロー変数に残る値になります。

**注：**この値は、書き込みオペレーションが遅かった方の値と一致するとは限りません。

たとえば、以下のイベントが次の順序で発生するとします。

- レーン A で、ステップ A1 が「42」を **threshold** に渡す。
- レーン B で、ステップ B2 が「34」を **threshold** に渡す。
- レーン B が完了する。
- レーン A が完了する。

この場合、ステップ A1 が **threshold** に値（「42」）を渡した後にステップ B2 が **threshold** に値（「34」）を渡していますが、**threshold** の値は「42」になります。

- 並列分岐ステップの場合、フローやオペレーションを実行する権限、およびその他の要素（システムアカウントなど）を使用する権限は、親フローに付与された権限によって決まります。
- レーン内のゲートトランジションはハンドオフできません。ユーザーが必要なグループメンバーシップを持っていない場合は実行が失敗し、必要なグループのメンバーでないことを知らせるエラーメッセージが表示されます。
- 同様に、レーン内のトランジションをハンドオフする設定はできません。
- 並列分岐ステップとレーンステップにはすべてチェックポイントが設定されており、このチェックポイントを削除することはできません。つまり、並列分岐ステップの実行中に実行が中断された場合、その実行を復元すると、並列分岐ステップが開始される直前のポイントまで復元されます。チェックポイントの詳細については、「[チェックポイント: リカバリ用のフロー実行進捗の保存](#)」を参照してください。

## レーンの順序: 開始と終了

レーンは、**Central** では同時に開始および実行されますが、デバッガーでは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、一意の名前を付けることにより、実行された順序を確認できます。

## 並列分岐ステップとのデータの移動

並列分岐ステップを開始すると、各レーンでグローバルコンテキストのフロー変数のコピーが取得されるため、フロー変数に関連付けられた値が各レーンのすべてのステップで使用できるようになります。

並列分岐ステップのレーン内の各ステップでは、ローカルコンテキストとグローバルコンテキストの両方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、`scriptletcontext.putGlobal()` メソッドを使用するスクリプトレットによってのみ、グローバルコンテキストへの書き込みが可能です。レーンステップでグローバルコンテキストに書き込まれるデータは、すべてのレーンが完了するまで、グローバルコンテキストに再びマージされません (`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 [Scriptlet] タブで、**JavaScript** テンプレートを挿入します)。

**重要:**


- 複数のレーンのステップで、同一のフロー変数に値が書き込まれる場合、最後のレーンの完了時にフロー変数に書き込まれる値が、並列分岐ステップの完了後に変数に含まれる値になります。
- 並列分岐ステップでは、レーン内のステップは別のレーンのステップに値を渡すことはできません。

## 並列分岐ステップの作成

並列分岐ステップを作成する際には、次のことに注意します。


- レーン間でデータを渡すことはできません（2つのレーン内の各ステップが実行される相対的な順序について、制御や予測はできません）。
- 並列分岐ステップのレスポンスは、**Success**（すべてのレーンが完了したことを示す）の1つのみです。
- 並列分岐ステップには、別の並列分岐ステップを含めることはできませんが、並列分岐ステップを含むサブフローを含めることは可能です。

**並列分岐ステップを作成するには**

1. Studio のツールバーで、[Parallel Split step] アイコン () をクリックしてフローデザインにドラッグします。デフォルトでは最小数（2つ）のレーンとともにステップが表示されます。
2. 各レーンに含めるステップシーケンスを作成します。  
レーンのステップシーケンスを作成するには、フローの作成と同じ方法を使用して同じルールに従い、各レーン内のステップを作成します。その際、次のことに注意します。

- レーンにはリターンステップは含まれません。すべてのステップのレスポンスを、次のステップまたはレーンの終了ポイントに接続します。success や failure などのレスポンス、デバッガーでのレーンステップの結果に関する情報が表示されます。Central ユーザーは、フローの実行履歴レポートでこうした情報を確認できます。
- レーン外のステップをレーンに追加する場合、そのステップをグループとして選択し、レーンにドラッグします。

3. 各レーンのステップを接続します。

前述のとおり、すべてのステップのレスポンスは、後続のステップまたは [Lane-end] アイコン (  ) に接続する必要があります。

4. 並列分岐ステップをフローの残りのステップに接続するには、次の手順で行います。

- 並列分岐ステップが開始ステップではない場合、その並列分岐ステップに先行するステップを [Parallel Split step] アイコンに接続します。
- 並列分岐ステップの done レスポンスをフローの次のステップに接続します。

ここで、非常に単純な並列分岐ステップのフローは次のようになります。

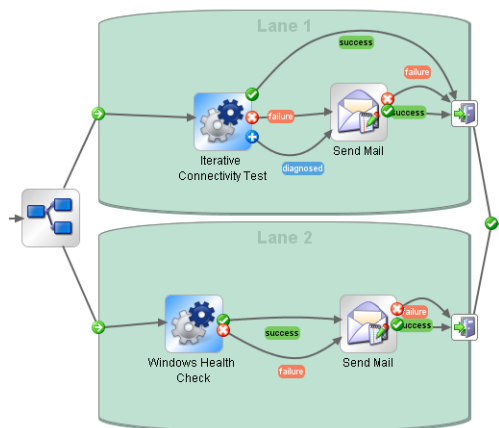



図 52 – 並列分岐ステップ

## 並列分岐ステップとコンポーネントの移動またはコピー

### 並列分岐ステップまたはコンポーネントを移動するには





- **Design** ビューでフローを開いて、次のいずれかを実行します。
  - 並列分岐ステップを移動するには、ステップ全体を表す [Parallel Split step] アイコン (  ) をクリックしてドラッグします。
  - 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。

ほかにも、フローキャンバスでレーンを上や下に移動する方法があります。ただし、すべてのレーンは同時に開始し、グラフィックの順序は処理の順序には影響しません。

### レーンの表示順序を変更するには

- 移動するレーンを右クリックして、ドロップダウンメニューで [Move Lane Up] または [Move Lane Down] をクリックします。

## 並列分岐ステップまたはコンポーネントをコピーするには

- **Design** ビューでフローを開いて、次のいずれかを実行します。
  - コピーするレーンまたはステップを右クリックするか、ステップ全体をコピーする並列分岐ステップの [Parallel Split step] アイコン () を右クリックし、ドロップダウンメニューから [Copy] をクリックします。レーン、ステップ、または並列分岐ステップ全体を貼り付ける場所にカーソルを移動し、右クリックしてドロップダウンメニューから [Paste] をクリックします。
  - コピーするレーンまたはステップをクリックするか、ステップ全体をコピーする並列分岐ステップの [Parallel Split step] アイコン () をクリックし、次のいずれかを実行します。
    - CTRL+C を押して、レーン、ステップ、または並列分岐ステップ全体を貼り付ける場所にカーソルを移動し、CTRL+V を押します。
    - 作成ペインのツールバーの [Copy] アイコン () をクリックします。レーン、ステップ、または並列分岐ステップ全体を貼り付ける場所にカーソルを移動し、ツールバーの [Paste] アイコン () をクリックします。
    - [Edit] メニューをクリックして [Copy] をクリックします。レーン、ステップ、または並列分岐ステップ全体を貼り付ける場所にカーソルを移動し、[Edit] メニューをクリックして [Paste] をクリックします。

## レーンの追加

### 新しいレーンを追加するには

- 既存のレーンを右クリックし、ドロップダウンメニューから [Add Lane] をクリックします。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。



## レーンの複製

### レーンを複製するには

- レーンを右クリックし、ドロップダウンメニューの [Duplicate Lane] をクリックします。コピーしたレーンの直下に、同じタイトルを持つ新しい空のレーンが表示されます。

## レーンの削除

### レーンを削除するには

- 次のいずれかを実行します。
  - 削除するレーンを右クリックし、ドロップダウンメニューから [Remove Lane] をクリックします。
  - 削除するレーンをクリックし、次のいずれかを実行します。
    - 作成ペインのツールバーの [Remove] アイコン () または [Cut] アイコン () をクリックします。
    - [Edit] メニューをクリックして [Cut] をクリックします。

## レーンのサイズ変更

### レーンのサイズを変更するには

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。

## レーンの名前変更

### レーンの名前を変更するには

1. レーンを右クリックし、ドロップダウンメニューの [Rename] をクリックします。
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の変更

### レーンの開始ステップを変更するには

- 次のいずれかを実行します。
  - 開始ステップに変更するレーンステップを右クリックし、ドロップダウンメニューから [Set Start Step] をクリックします。
  - レーン開始アイコンコネクタを、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。

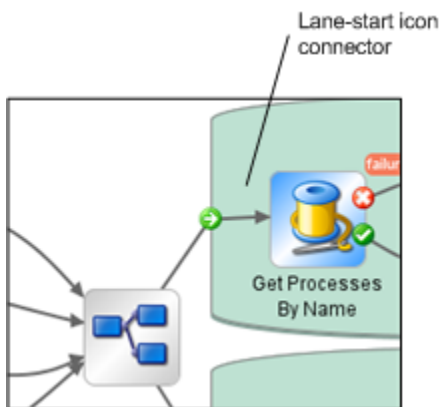


図 53 - レーン開始アイコンコネクタ

## 並列分岐ステップのデバッグ

並列分岐ステップのデバッグ時、デバッガーは、並列分岐ステップをパラレル実行ではなくシリアルで処理します。そのため、デバッガーでは本稼動環境でのフローの動作が正確に再現されません。一方、デバッガーでの並列分岐ステップのシリアル実行により、さまざまな状況の制御テストを実行できます。

並列分岐ステップ、マルチインスタンスステップ、またはノンブロッキング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。

フローのデバッグ方法については、「[フローのデバッグ](#)」を参照してください。

## マルチインスタンスステップ

100 台のサーバーで **Windows Diagnostic** フローを実行するとします。1 台のサーバーでフローを実行するステップを作成するのではなく、フローに関連付けられたステップをマルチインスタンスステップにして、100 台すべてのサーバーで一度にフローを実行できます。

マルチインスタンスステップは、入力にさまざまな値を設定できるステップで、同時に実行する複数のステップのインスタンスで使用されます。このステップを含むフローでは、たとえば、異なるターゲットに対して同一のオペレーションの多くのインスタンスを同時に実行できます。

ただし、100 台のサーバーでフローを同時に実行するとシステムのパフォーマンスが低下する場合、マルチインスタンスステップで同時にフローを開始するサーバーの数を指定して、フローのスロットルレベルを設定できます。



マルチインスタンスステップには常にチェックポイントが設定され、このチェックポイントは削除できません。つまり、マルチインスタンスステップの実行中に実行が中断された場合、その実行を復元すると、マルチインスタンスステップが開始される直前のポイントに復元されます（チェックポイントの詳細については、「[チェックポイント: リカバリ用のフロー実行進捗の保存](#)」を参照してください）。

## フローデザインでのマルチインスタンスステップの使用

マルチインスタンスステップには、追加のレスポンスとして **group done** があります。マルチインスタンスステップの **group done** レスポンス以外の各レスポンスについては、レスポンスに続くパスの最後のステップのトランジションすべてを、マルチインスタンスステップに再び接続する必要があります。これは非常に重要なことです。言い換えれば、**group done** レスポンスのみをフローのリターンステップ、つまりフローの最後につなげるようにします。

次に、フローデザインでマルチインスタンスステップを使用する方法について、最も単純な例を示します。

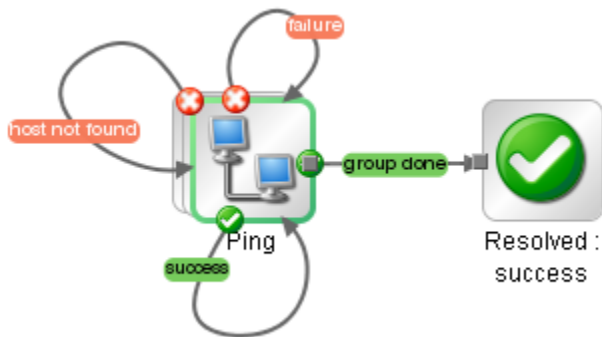


図 54 – フローデザインでのマルチインスタンスステップ使用の要点

さらに現実的に考えると、マルチインスタンスステップのさまざまなレスポンスに続けて、ほかのオペレーションの実行が必要になることがあります。次に示す例でも、必須のパターンに倣っています。

- **Ping** ステップの 4 つのレスポンス (**success**、**failure**、**host not found**、**group done**) のうち、**group done** レスポンスのみがリターンステップにつながります。
- **failure** レスポンスと **host not found** レスポンスはそれぞれ電子メールステップにつながり、両方のレスポンスが (マルチインスタンスの) **Ping** ステップに再び接続します。
- **success** レスポンスはステップのチェーンにつながりますが、いずれ、そのシーケンスのレスポンスはすべて (マルチインスタンスの) **Ping** ステップに再び接続します。

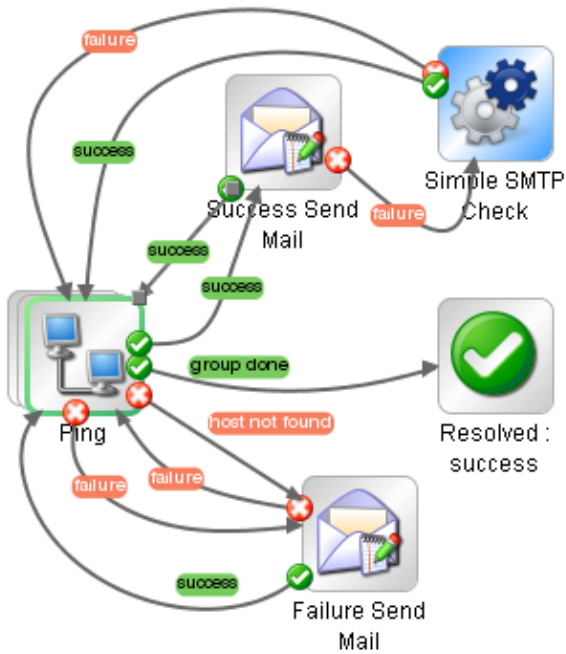


図 55 - 適切なデザイン: マルチインスタンスステップに再び接続されるレスポンス

2つの理由から、マルチインスタンスステップから続くステップのパスのレスポンスはすべて、最終的にマルチインスタンスステップに再び接続する必要があります。

- マルチインスタンスステップでは、各インスタンスの結果は利用可能になったときに評価され、その結果に関するレスポンスは、ほかのインスタンスの結果が評価可能になるのを待たずに選択されます。そのため、マルチインスタンスステップのレスポンスに続くステップが、マルチインスタンスステップに戻らずに、フローの実行を完了するリターンステップに遷移する場合、実行では、ほかに評価すべき結果が存在するかどうかを検出できません。

そのため、次の例では、結果を戻す最初のインスタンスについて、1つだけ電子メールがフローで生成されます。

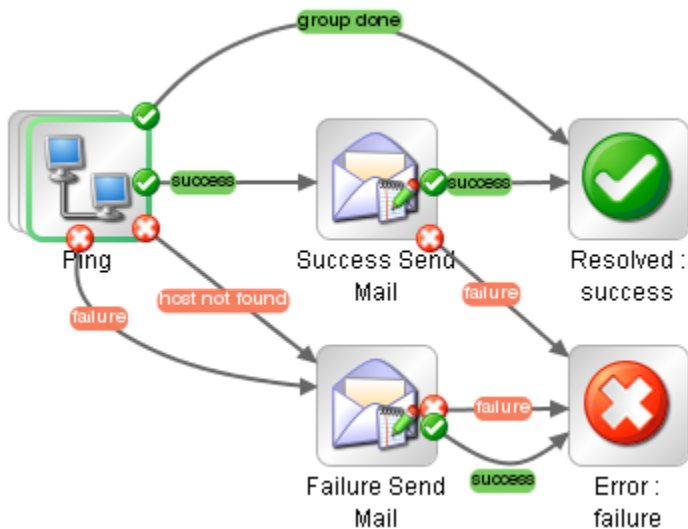


図 56 - 不適切なデザイン: マルチインスタンスステップに再び接続されないレスポンス



- すべてのレスポンスをマルチインスタンスステップに再び接続するもう1つの理由として、サブフローで不適切なデザインを使用すると、**Central**での親の実行は、このサブフローから作成されるステップの後で、渡すことも中断することもできなくなります。マルチインスタンスステップのインスタンスの結果の一部が未処理のままであるためです。

ステップの複数のインスタンスとその結果を処理するこうした方法は、並列分岐ステップとは対照的なマルチインスタンスステップを使用する利点の1つに関係しています。並列分岐ステップでは、フローが次のステップに進む前に、すべてのレーンで処理が完了するのを待ちますが、マルチインスタンスステップでは、ほかのインスタンスの処理中でも、最初に完了したインスタンスについて（選択されたレスポンスに基づき）フローが後続のステップの実行を続行します。そのため、複数のターゲットに同一のオペレーションを実行する場合、マルチインスタンスステップは並列分岐ステップよりも効率的な手段となります。

マルチインスタンスステップの処理の仕組みは次のとおりです。

1. マルチインスタンスステップのインスタンスの1つが処理を完了するとすぐに、評価ルールでレスポンスが選択され、そのインスタンスの後続のステップの実行に進みます。
2. その間、ほかのインスタンスでは処理が続行されています。マルチインスタンスステップ内の各インスタンスの処理が完了すると、以降の処理に備えて結果が保存されます。この時点で、実行では、ステップの複数のインスタンスのシリアル処理が開始されます。
3. 最初のインスタンスと後続のステップの処理が完了し、この最後のステップをマルチインスタンスステップに再び接続してある場合、実行はマルチインスタンスステップに戻り、完了した次のインスタンスの結果についてのシーケンスのステップが繰り返されます（マルチインスタンスステップ内のほかのインスタンスがまだ処理を完了していない場合、次のインスタンスの結果が利用可能になるまで実行は待機します）。

## マルチインスタンスステップの作成

ステップをマルチインスタンスステップにするには、ステップ自体を変更してから、複数の値を含める入力にリストを指定します。入力に複数の値を含めるには、入力の値をリストとして、そのリストを入力に指定します。

マルチインスタンスステップはノンブロッキングにもできます。ほかのノンブロッキングステップと同様に、マルチインスタンスステップはすべてのレスポンスを失い、**done** レスポンスのみを取得します。

### ステップをマルチインスタンスステップにするには

1. ステップを右クリックし、表示されたコンテキストメニューで [Toggle Multi-instance] をクリックします。ステップの表示が次のように変更されます（ステップの複数のインスタンスが層になった枠で囲まれて表示されます）。



図 57 - マルチインスタンスステップ

ステップには現在、追加のレスポンスとして **group done** があります。**group done** レスポンスの使用の詳細については、前述のトピックを参照してください。

2. ステップの Inspector を開いて、複数の値を指定する入力の入力エディターを開きます。
3. 入力の値をリストにするには、[Input Type] ドロップダウンリストで [List of Values] を選択し、[Input Delimiter] テキストボックスに区切り文字（リスト内の要素を区切る文字）を入力します。デフォルトの区切り文字はコンマです。

## 重要:

- 区切り文字は、リスト要素を区切る文字または文字の組み合わせを正しく示すものを指定する必要があります。

たとえば、リスト要素の間にスペースがある場合、指定する区切り文字にはそのスペースも含めます。次のリストを考えてみましょう。

10.51.0.36, 10.51.0.37

区切り文字としてコンマ (,) のみを指定すると、ステップの1つのインスタンスの入力値が「10.51.0.37」となり、エラーが発生することがあります。

- Central ユーザーがマルチインスタンスステップを含むフローのスケジュールを設定する際には、区切り文字を把握しておく必要があるため、必ずユーザーにステップの [Description] タブの情報を知らせます。
  - ステップをマルチインスタンスステップにする場合、ステップのチェックポイントとしての指定を追加または削除することはできません。
4. 指定した区切り文字を使用するリストを入力（または入力が値を取得するフロー変数）に指定します。リストの指定方法は、リストの取得方法によって決まります。たとえば、別のプログラムとの統合によりリストを取得できます。
- フロー内から [Constant Value] テキストボックスでマルチインスタンスステップに直接リストを指定するか（入力の割り当てで [Use Constant] を選択した場合）、またはフロー変数からリストを取得できます。
5. ステップをシングルインスタンスステップに戻すには、ステップを右クリックし、表示されたコンテキストメニューで [Toggle Multi-instance] を再びクリックします。

## マルチインスタンスステップとのデータの移動

値は、ほかのステップの場合と同様にマルチインスタンスステップに渡され、親のグローバルフロー変数が、子の実行（インスタンスの実行）のグローバルフロー変数にマージされます。続いて、マルチインスタンスステップの入力が、子の実行のグローバルフロー変数になります。

マルチ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。インスタンスの実行が完了した後も続行されるのは、（スクリプトレットで作成された）グローバルフロー変数のみです。

ステップ外で使用可能なデータをステップ内に生成するには、ほかのタイプのステップの場合と同じ方法を使用します。スクリプトレットでグローバルフロー変数を作成するか、（ステップがサブフローである場合）フロー出力フィールドを作成します。

## リスト入力、フロー変数、マルチインスタンスステップ

マルチインスタンスステップのリスト入力の値は、そのリスト入力が、入力と同じ名前のフロー変数または異なる名前のフロー変数のいずれに割り当てられているかによって異なります。

- リスト入力が**同じ名前**のフロー変数に割り当てられている場合: マルチインスタンスステップの各評価後、変数の値は、処理される結果に対応するリストの要素です。
- リスト入力が**異なる名前**のフロー変数に割り当てられている場合: リスト入力の値は2つのフロー変数に格納されます。
  - リスト入力の名前を持つフロー変数の値は、処理される結果に対応するリストの要素です。
  - リスト入力とは異なる名前を持つ変数には、リストの値全体が割り当てられます。

そのため、リスト入力のリスト内の各要素を参照する場合は、リスト入力と同じ名前を持つフロー変数を参照する必要があります。

たとえば、次のようなフローがあるとします。

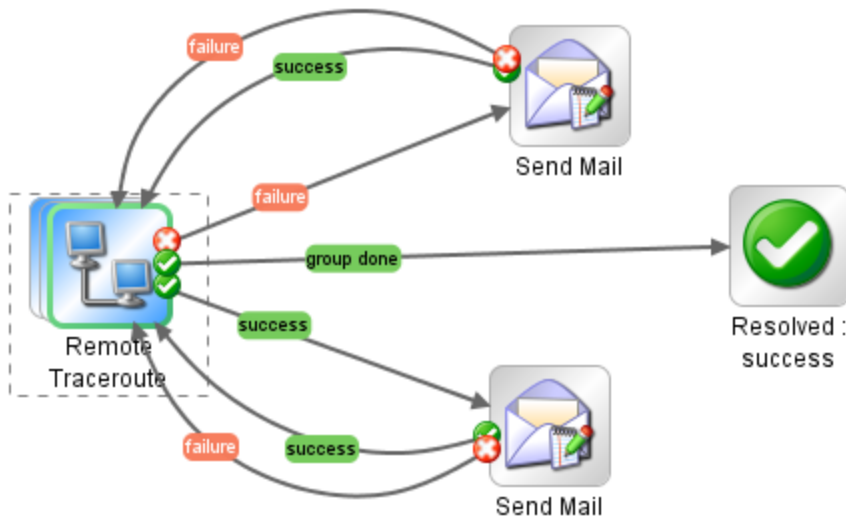


図 58 - フロー内のマルチインスタンスステップ

このフローの内容は次のとおりです。

- **Remote Traceroute** ステップはマルチインスタンスステップです。
- **Remote Traceroute** ステップには、**target** という名前の入力フロー変数があります。この変数はリスト入力で、**machine** という名前のフロー変数に割り当てられています。
- 入力 **target** のリストの値は、**Remote Traceroute** の対象となるマシンの名前です。
- **Send Mail** ステップは、**target** の各対象マシンについて、成功または失敗の電子メールを送信します。

それぞれの結果について **Send Mail** ステップで送信される電子メールの本文に各対象マシンの名前を含めるには、電子メールの本文は、フロー変数 **machine** ではなくフロー変数 **target** を参照する必要があります。電子メール本文が **machine** を参照した場合、**Remote Traceroute** が成功または失敗した特定の対象マシンではなく、リスト全体が印刷されます。

**Send Mail** ステップの電子メール本文には、成功パスまたは失敗パスについて、それぞれ次のような文を指定します。

- Remote Traceroute succeeded for  $\${target}$ .
- Remote Traceroute could not connect to  $\${target}$ .

## マルチインスタンスステップの制限

マルチインスタンスステップで 1000 の並列プロセスを起動すると、システムに大きな負担となることがあります。マルチインスタンスステップの並列処理能力をシステムの処理能力に合わせるために、ステップで一度に処理される入力値の数を制限できます。

### マルチインスタンスステップを制限するには

1. ステップの Inspector を開いて、[Advanced] タブをクリックします。
2. [Execution] で、[Throttle parallel execution...] チェックボックスを選択します。
3. テキストボックスに、一度に実行するステップのインスタンスの最大数を入力します。

## フロー内のマルチインスタンスステップのデバッグ

マルチインスタンスステップが **Central** でフロー実行の一部として実行される方法と、**Studio** デバッガーで実行される方法には、重要な違いがあります。

- **Central** では、マルチインスタンスステップの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。
- デバッガーでは、インスタンスはシリアルで処理されます。

フローのデバッグ方法については、「[フローのデバッグ](#)」を参照してください。

## ステップのノンブロッキング化

**Network Diagnostic** ステップの **failure** レスポンスなどの結果が出た後、何人かに特定の通知を送信するフローが必要になったとします（複数の人に通知する場合、このステップをノンブロッキングにするだけでなく、マルチインスタンスステップに変換することもできます）。

フローの後続のアクションはこのステップの結果に左右されないため、ステップが通知の送信作業を行っている間に、フローでこのステップの実行を続行できれば便利です。

ノンブロッキングステップは、フローでノンブロッキングステップに続くステップの実行を続行している間に完了するステップです。フローが続行しているので、ステップの結果は、フローが続くパスに影響するものであってはなりません。そのため、ノンブロッキングステップのレスポンスは **done** の1つのみです。

ステップをノンブロッキングにすると、自動的にチェックポイントが設定されます。つまり、ノンブロッキングステップの後にチェックポイントがないフローに実行を復元すると、その実行はノンブロッキングステップが開始される直前のポイントに復元されます。チェックポイントの詳細については、「[チェックポイント: リカバリ用のフロー実行進捗の保存](#)」を参照してください。

### ステップをノンブロッキングにするには

1. 作成キャンバスでフローを開いた状態で、ステップを右クリックして、[Toggle Nonblocking] をクリックします。  
ステップの接続を示すアイコンにオレンジ色の稲妻が表示され、ステップは自動的に1つのレスポンス **done** を取得します。



2. **done** レスポンスを次のステップに接続します。  
**注:** 並列処理をステップに追加する場合、ステップのチェックポイントとしての指定を追加または削除することはできません。

## ステップのレスポンスの単一化

ステップのすべてのレスポンスが同一のターゲットステップに遷移し、トランジションがすべて同じプロパティ（同じ説明や ROI 値など）を持つ場合、こうしたレスポンスすべてを1つのレスポンスにまとめることができます（このレスポンスをターゲットステップに接続する）。

### ステップのレスポンスを単一レスポンスに変換するには

- フロー図を開いて、ステップを右クリックし、ドロップダウンメニューで [Toggle Single Response] を選択します。

## 単一レスポンスのステップを複数レスポンスに戻すには

- フロー図を開いて、ステップを右クリックし、ドロップダウンメニューで [Toggle Single Response] を選択します。

## チェックポイント: リカバリ用のフロー実行進捗の保存

チェックポイントは、フロー内のブックマークのようなものです。フローの実行中、各チェックポイントは、フローの状態データを収集して **Central** データベースに保存します。フローの状態データは、フローが中断した場合に実行を完全に再構築するのに必要なすべてのデータを含みます。**Central** のフェイルオーバーまたは実行リカバリクラスタを設定して **Central** サーバーのフロー実行が失敗した場合、クラスタのほかのノードが、失敗した **Central** サーバーで行われていたフロー実行を再開します。フローにチェックポイントが設定されていない場合、実行はフローの開始時から再開されます。フローのステップにチェックポイントが設定されていると、サーバーが失敗したときに到達していた最後のチェックポイントから実行が再開されます。デフォルトでは、ステップの作成時にチェックポイントが作成されます。

**注:** ユーザーが実行を一時停止または中断した場合、中断時の実行の状態がデータベースに保存されるので、中断した時点から実行を再開するためにチェックポイントは必要ありません。チェックポイントは、システム障害で中断された実行を再開するために設定されるものです。

作成したステップにはそれぞれチェックポイントがデフォルトで設定されるため、フロー状態データの情報が膨大になることがあります。こうした大量のデータがデータベースに頻繁に書き込まれると、フローのパフォーマンスに影響を与える可能性があるため、フロー内のチェックポイントの数を削減することが必要になります。たとえば、ループ内の一部のステップからチェックポイントを削除します。ステップからチェックポイントを削除すると、フローの実行は、削除した直前のチェックポイントから再開されます。

**注:**

- チェックポイントを削除しても、フロー実行中にデータベースに送信されるデータの量は変わりませんが、データが送信されるポイントのみが変わります。
- サブフローから作成されたステップのチェックポイントを削除する場合、サブフローのステップ（またはサブフローのステップのサブフロー内のステップ）には、もともとチェックポイントが設定されていたかどうかにかかわらず、チェックポイントは設定されません。
- 並列処理のステップ、つまり並列分岐ステップ、マルチインスタンスステップ、ノンブロッキングステップにはチェックポイントが設定され、こうしたステップのチェックポイントは削除できません。並列処理のステップの実行中に中断された実行を復元する場合、チェックポイントにより、そのステップの直前に実行が復元されます。

チェックポイントと並列処理のステップについては、次のことにも注意してください。

- 並列分岐ステップのレーンステップにはすべてチェックポイントが設定されており、このチェックポイントを削除することはできません。
- マルチインスタンスステップの各インスタンスを処理するステップからは、チェックポイントを削除できます。

## ステップにチェックポイントを作成するには

1. **Design** ビューでフロー図を開き、チェックポイントを追加するステップの **Inspector** を開きます。
2. **[Advanced]** タブの **[Step Persistence]** エリアで、**[This step saves the whole run state]** チェックボックスを選択します。



## ステップからチェックポイントを削除するには

1. **Design** ビューでフロー図を開き、チェックポイントを追加するステップの **Inspector** を開きます。
2. **[Advanced]** タブで、**[This step saves the whole run state]** チェックボックスの選択を解除します。

## スクリプトレット

**Ping**、**Traceroute**、**HTTP Client Get** などのオペレーションからデータを取得した場合、特定の結果について、テスト、フォーマット、加工、分離が必要となることがあります。また、結果を比較するためにフロー変数の値を使用することもあります。スクリプトレットを使用すると、こうしたタスクのほか、JavaScript や Perl で可能な作業を実行できます。

スクリプトレットは、オペレーションのコアアクションに伴うさまざまな副作用を取得したり、オペレーションの機能を拡張するのにも適しています。

スクリプトレットは、オペレーションに含まれる **Sleep** または **JavaScript (Rhino)** スクリプトです。スクリプトレットにより、出力の評価、結果やレスポンスの決定、変数のフローへの追加について、オペレーションの出力フィルターやレスポンスよりも優れた制御力が得られます。

JavaScript スクリプトレットや Perl スクリプトレットは、次のことに使用できます。

- フロー、ステップ、オペレーションの結果のフィルタリング。  
オペレーションの結果のフィルタリングは、オペレーションのレスポンスを決定するルールの一部としても使用されます。
- オペレーション、ステップ、またはフローの本体（それぞれの **[Scriptlet]** タブ）。  
これは、一般的に勧められるスクリプトレットの用途ではありません。スクリプトレットは、結果のフィルタリングに最適です。



**ヒント:** OO で使用する 2 つのスクリプト言語、JavaScript (Rhino) または Sleep のいずれかの使用方法がわかっている場合、これらの言語に関する OO のスクリプトレットテンプレートでは、OO がスクリプト言語で情報交換を行うのに必要な構文とオブジェクトが定義されます。スクリプトレットテンプレートは、ステップの **Inspector** またはオペレーションの **[Properties]** シートの **[Scriptlet]** タブで入手できます。

スクリプトレットの使用にあたっては、次のことに注意してください。

- オペレーションのほかの使用法と同様、繰り返し使用できる一般的な用途の場合は、オペレーションでスクリプトレットを作成します。
- 特定のステップの状態に特別に合わせたスクリプトレットにする場合、そのステップでスクリプトレットを作成します。  
たとえば、ステップにスクリプトレットを追加して、ステップのオペレーションから戻されるデータの評価やフォーマットが可能です。データをフロー変数に格納してフローの結果のいずれかのフィールドに渡し、ほかの結果でもこれを繰り返すと、見つかったデータの証跡を徐々に作成できます。
- 特定のフローを別のフロー内のサブフローとして使用し、オペレーションとして機能するようにする場合、（フローの **[Properties]** タブの **[Scriptlet]** タブで）そのフローにスクリプトレットを追加します。フロー結果のフィールドに渡したデータの一部をサブフローが取得した後、フロー結果フィールドのデータを加工してから親フローに渡すことができます。
- **HP OO** では、各入力から同じ名前の変数を自動的に作成します。そのため、スクリプトレットは、入力と同じ名前を持つ変数を使用して値にアクセスできます。

## スクリプトレットの作成

### スクリプトレットまたはスクリプトレットテンプレートを表示するには

1. オペレーションの [Properties] シートまたはステップの Inspector で、[Scriptlet] タブをクリックします。
2. オペレーションにまだスクリプトレットが含まれていない場合にスクリプトレットテンプレートを表示するには、[Type] ドロップダウンリストで記述するスクリプト言語を選択して [Insert Template] をクリックします。



**ヒント:** 既存のスクリプトレットの例については、OO に同梱されているコンテナツのオペレーション (**Operations** ~~¥~~ **Operating Systems** ~~¥~~ **Linux** ~~¥~~ **Red Hat** フォルダー内のオペレーションなど) を参照してください。

### 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には

1. [Results] タブで、結果の行の右向き矢印を右クリックして、スクリプトレットでフィルタリングする結果のフィルターエディターを開きます。
2. フィルターエディターで、[Add] をクリックして新しいフィルターを作成します。
3. [Select Filter] ダイアログボックスで、[Scriptlet] を選択します。
4. スクリプトレットを作成し、フィルターをテストして、作業結果を保存します。  
共通タスクの実行に役立つ関数の構文を表示するには、[Insert Template] をクリックします。スクリプトレットテンプレートには、一般に使用する関数とその構文が含まれています。

フィルターの作成方法の詳細については、「[出力および結果のフィルタリング](#)」を参照してください。



**ベストプラクティス:** 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 [Responses] 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には

1. オペレーションの [Properties] シートの [Responses] タブで、[Add Response] をクリックして、表示されたボックスにレスポンスの名前を入力します。  
もしくは  
レスポンスの行の右向き矢印を右クリックします。
2. [Rule Type] ドロップダウンリストで、[Scriptlet] を選択します。
3. レスポンスの行の右向き矢印をクリックします。
4. スクリプトレットを作成します。  
共通タスクの実行に役立つ関数の構文を表示するには、[Insert Template] をクリックします。スクリプトレットテンプレートには、一般に使用する関数とその構文が含まれています。
5. 作業結果を保存します。

## スクリプトレットのデバッグ

ローカルリポジトリに存在するフローのスクリプトレットのみデバッグできます。パブリックリポジトリのスクリプトレットをデバッグするには、パブリックリポジトリからローカルリポジトリを更新します。続いて、Studio をローカルリポジトリに接続した状態で、次の手順を使用して、スクリプトレットを含むフローで JavaScript Debugger を実行します。

### スクリプトレットをデバッグするには

1. フローのデバッグを開始する前に、[Tools] メニューから [JavaScript Debugger] をクリックします。JavaScript Debugger が起動します。

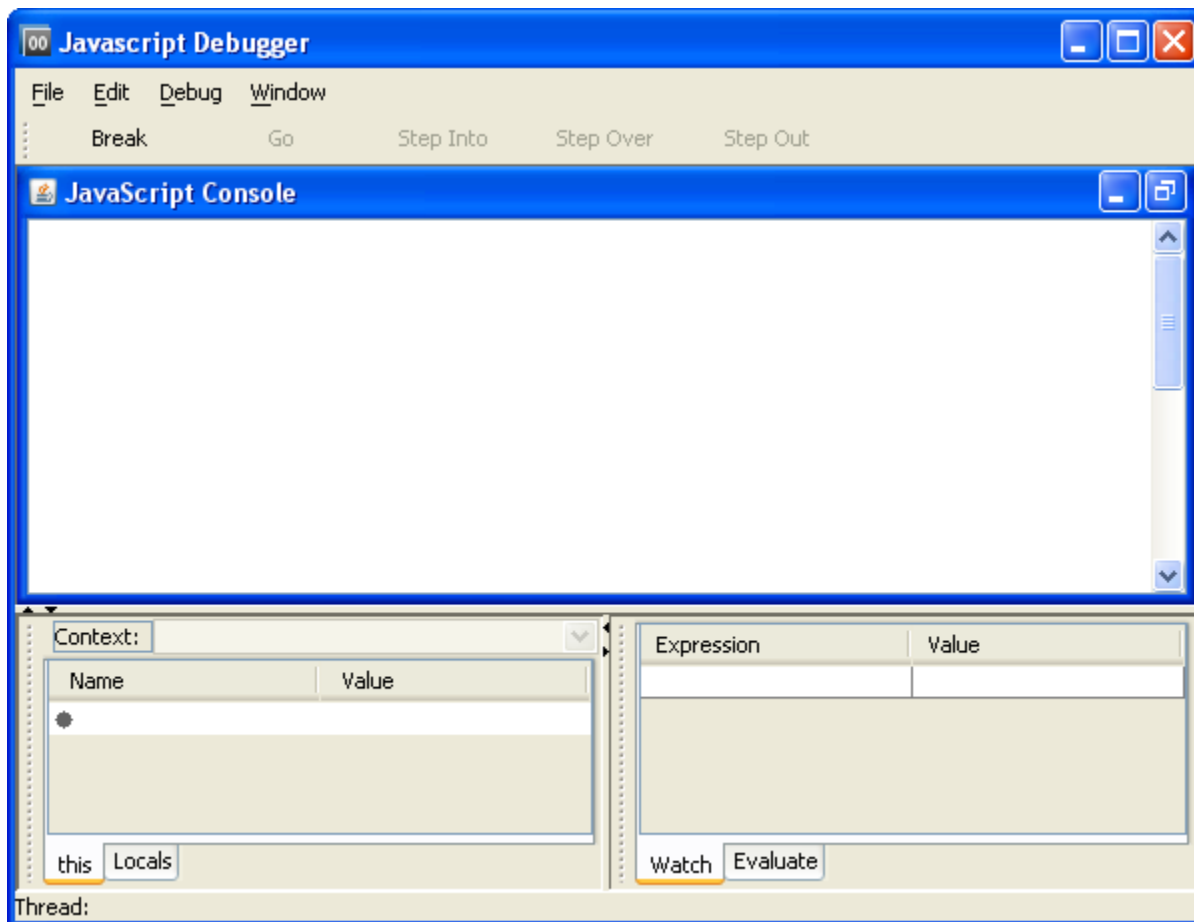


図 59 – Javascript Debugger

2. [Debug] メニューから、[Break on Function Enter] を選択します。  
フロー、各ステップ、各オペレーションに含まれるすべてのスクリプトレットが、デバッガーにロードされます。
3. JavaScript Debugger で、スクリプトレット内の 1 つ以上の行にブレークポイントを設定するには、目的の行をクリックします。  
行の横に赤色の点が表示されます。






図 60 – JavaScript Debugger でのスクリプトレット行のブレークポイント設定

注: スクリプトレット行にブレークポイントを作成した場合、JavaScript Debugger を閉じると、スクリプトはそのブレークポイントで停止しません。

4. Studio で、ステップにブレークポイントを設定するには、ステップを右クリックして [Debugging] をポイントし、[Set Breakpoint] をクリックします。

5. Studio で、フローのデバッグを開始するには、[Debug Flow] ボタン (  ) をクリックします。

デバッガーが起動し、Central (フローがパブリックリポジトリにある場合) または Studio (フローがプライベートリポジトリにある場合) のいずれかでフローをデバッグします。

Studio デバッガーがスクリプトを含むステップに到達すると、JavaScript Debugger に切り替わります。


スクリプトレットが実行されると、コンテキスト内のローカルとグローバルのフロー変数の現在の値が、[Context] ペインの [Value] 列に表示されます。

JavaScript Debugger が設定したブレークポイントに到達すると、監視対象として特定のフロー変数の値を設定できます。

6. 特定のフロー変数を監視するには、スクリプトレットを一時停止して、右下のパネルで、[Expression] ボックスに監視するフロー変数の名前を入力します。

## ほかの用途を目的としたスクリプトレットの保存

### スクリプトレットを Configuration\Scriptlets フォルダーに保存するには

1. スクリプトレットエディターで、スクリプトレットを選択し、[Scriptlet] アイコン (  ) をクリックして **Configuration\Scriptlets** フォルダーにドラッグします。
2. **Configuration\Scriptlets** フォルダーで、スクリプトレットに名前を付けます。
3. 作業結果を保存します。

### Configuration フォルダーからスクリプトレットを使用するには

1. リポジトリペインで、**Configuration\Scriptlets** フォルダーを開きます。
2. 使用するスクリプトレットをフォルダーからスクリプトレットエディターにドラッグします。
3. **Configuration\Scriptlets** スクリプトレットからユーザー作成のスクリプトレットに変更するには、[Switch to a custom scriptlet] をクリックします。

## 出力、レスポンス、およびステップの結果

出力、レスポンス、ステップの結果は、フロー作成で互いに関連しています。

- 出力は、オペレーションのすべての出力またはその一部です。  
フローはオペレーションの一種です。これは、フロー内のステップが別のフローに関連付けられる場合によくわかります。フローにも出力があります。
- レスポンスは、オペレーション出力の評価の結果です。  
結果をテストしてレスポンスを決定する際のルールの定義については、「[レスポンス: 結果の評価](#)」を参照してください。
- ステップの結果は、オペレーション出力に似ており、出力からコンテンツを取得します。  
また、ステップの結果はフロー結果のフィールドとしてフローに渡すことができます。

出力や結果をフィルタリングして、別のオペレーションに渡したりテストをする際に使用可能なデータを微調整できます。フィルターを作成については、「[出力および結果のフィルタリング](#)」を参照してください。

### オペレーション出力

オペレーション出力には2つの種類があります。

- 未加工出力は、すべてのオペレーションのリターンコード、データ出力、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。出力にフィルターを1つ以上作成すると、プライマリ出力やその他の出力の選択の範囲をさらに狭くすることができます。  
出力のフィルタリングの詳細については、「[出力および結果のフィルタリング](#)」を参照してください。

出力は、評価を行うだけでなく、データとして、フロー内のほかのステップや、フロー変数に値を格納してほかのフローに渡すことができます。フィルターを作成すると、結果の抽出や編集が可能になります。

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**  
オペレーションのプライマリ出力。名前付きのフィールドがなく「returns:」が表示された場合、通常これは出力です。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の内部出力。オペレーションが **failure** 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。

## オペレーションのプライマリ出力の指定

オペレーションを作成するには、プライマリ出力を指定できます。出力や結果の詳細については、「[出力と結果](#)」を参照してください。

### オペレーションのプライマリ出力を指定するには

1. 編集用にオペレーションを開くには、Library でオペレーションを右クリックし、[Open] を選択します。
2. オペレーションのエディターで、[Outputs] タブをクリックします。
3. [Extract Primary Output from Field] ドロップダウンリストから、ソースフィールドを選択します。  
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 [Description] タブをクリックします。

必要な特定のデータを抽出するために、出力用のフィルターを作成できます。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](#)」を参照してください。

## レスポンス

レスポンスはオペレーションの結果です。オペレーションから作成されたステップでは、各レスポンスは、トランジションが別のステップに接続される（または同一のステップに戻る）元の開始ポイントとなります。オペレーションの出力フィールドを評価するルールは、使用可能なレスポンスのうち、いずれが現在の実行についてオペレーションの実際のレスポンスとなるかを決定します。そのため、あるステップのオペレーション結果の評価結果により、フロー実行の次のステップが決まります。

レスポンスとそのルールの作成については、「[レスポンス: 結果の評価](#)」を参照してください。

## ステップの結果

ステップの結果は、ステップが作成されたオペレーションの出力に似ています。ステップの結果を作成します。

ステップ結果をデータとして、フロー内のほかのステップやほかのフローに渡すことが可能です。フィルターを作成すると、オペレーションの出力の抽出や編集ができます。

たとえば、特定のサーバーに対する 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。ping オペレーションの未加工結果に含まれるこれらの3つの情報は、未加工結果を3つのフィルタリング結果にフィルタリングすることにより、オペレーションのフィルタリング結果に抽出できます。こうしたフィルタリング済み結果は、次のことに使用可能です。

- フィルタリング済みのデータ結果を1回以上テストしてフローの次のステップを決定するためのレスポンスルール（エバリュエーター）を作成する。
- フィルタリング済みデータを値としてフロー変数に渡し、以降のフロー内でオペレーションやトランジションにアクセスできるようにして、フロー変数を参照するプロンプトによりフローのユーザーがこのデータを使用できるようにする。
- フローが別のフローのステップ（つまり、親フローのサブフロー）である場合に、フィルタリング結果のデータをフロー結果のフィールドに渡して、プロパティが親フローのオペレーション、ステップ、トランジションで使用できるようにする。

出力のフィルタリングの詳細については、「[出力および結果のフィルタリング](#)」を参照してください。

ステップの結果の値は、次のいずれかに渡すことができます。

- ローカルフロー変数。
- フローのフロー出力フィールド。


## 出力の追加と削除

### オペレーションのプライマリ出力をフィールドとして指定するには

1. プライマリ出力のソースを指定するには、オペレーションの [Outputs] タブで、[Extract Primary Output From Field] ドロップダウンリストボックスからオペレーションの出力フィールドのいずれか1つを選択します。
2. 必要に応じて結果を形成するためのフィルターを作成するには、[Edit Filters] をクリックします。  
フィルターエディターが開き、必要なデータを分離するためのフィルターを作成できます。フィルターの作成については、「[出力および結果のフィルタリング](#)」を参照してください。
3. 作業結果を保存します。

注: 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。

### オペレーションにセカンダリ出力を追加するには

1. セカンダリ出力（プライマリ出力に追加する別の出力）を追加するには、[Add Output] をクリックして表示されるダイアログボックスで、出力の名前を入力します。  
注: [Available Outputs] の下にある出力の行の [Name] 列と [Output Field] 列の値を変更して、出力の名前やソースフィールドを変更できます。
2. セカンダリ出力の出力データにフィルターを作成するには、行末の右向き矢印 () をクリックします。  
フィルターエディターが開いてフィルターを作成できます。フィルターの作成については、「[出力および結果のフィルタリング](#)」を参照してください。

必要なフィルターを作成したら、フィルターエディターを閉じます。出力の行の [Filters] 列には、作成したフィルター、または複数のフィルターを作成した場合は作成したフィルターの数が表示されます。

3. 作業結果を保存します。

### オペレーションから出力を削除するには

1. オペレーションの [Outputs] タブで、削除する出力を選択して [Remove Output] をクリックします。
2. 作業結果を保存します。

## 出力のソースの変更


### 出力がデータを取得するフィールドを変更するには

1. プライマリ出力のフィールドを変更するには、[Extract Primary Output From Field] 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。  
もしくは  
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 [Output Field] 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。
2. 出力に必要なフィルターを作成します。必要なフィルターを作成したら、フィルターエディターを閉じます。  
フィルターの作成については、「[出力および結果のフィルタリング](#)」を参照してください。
3. 作業結果を保存します。

## ステップの結果の追加と削除

ステップの未加工結果と主結果は、最も基本となるオペレーションの未加工出力とプライマリ出力に由来します。ステップの Inspector で、二次結果を作成、指定できます。

### ステップに結果を追加するには

1. 二次結果を追加するには、ステップの Inspector を開き、[Results] タブで [Add Result] をクリックします。結果のリストに新しく表示される行で、結果の名前は、デフォルトで結果のソースの名前になります。
2. 結果の定義を変更するには、次に示すいずれかのフィールドの行をクリックして、次の手順で行います。
  - [Name] フィールドに、新しい名前を入力します。
  - [From] フィールドで、フィールドをクリックして表示されるドロップダウンリストから、結果のソースを選択します。
  - [Assign To] フィールド：
    - フロー変数の値を保存するには、[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。
  - [Assignment Action] リストから、適切なアクションを選択します。
    - **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。
    - **[ADD]**、**[SUB]**、**[MULTIPLY]**、**[DIVIDE]** の四則演算の割り当てアクションも選択できます。この値を使用して、フロー変数またはフロー出力フィールドの現在の値を演算で変更します。たとえば、ステップの結果が 3.14 で [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。また、タイムアウトフロー変数があるとします。タイムアウトフロー変数の値から 1 を減算する場合、[Assignment Action] に [SUB] を選択して結果（値は 1 と仮定）が「タイムアウト」から引かれるよう指定できます。
3. セカンダリ出力の出力データにフィルターを作成するには、行末の右向き矢印 () をクリックします。フィルターエディターが開いてフィルターを作成できます。フィルターの作成については、「[出力および結果のフィルタリング](#)」を参照してください。
4. 必要なフィルターを作成したら、フィルターエディターを閉じます。出力の行の [Filters] 列には、作成したフィルター、または複数のフィルターを作成した場合は作成したフィルターの数が表示されます。
5. 作業結果を保存します。

### ステップから結果を削除するには

1. オペレーションの [Results] タブで、削除する結果を選択して [Remove Result] をクリックします。
2. 作業結果を保存します。

## 結果のソースの変更

### 結果がデータを取得するフィールドを変更するには

1. 二次結果がデータを取得するフィールドを変更するには、出力の行の [From] 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。
2. 作業結果を保存します。

## 出力および結果のフィルタリング

ネットワーク接続の問題を診断する場合、フローの内外で、最大、最小、平均の往復時間、失われたパケットの割合などのデータをほかの用途に使用するために抽出することが必要となることがあります。

特定のサーバーに対する 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。ping オペレーションの未加工出力を 3 つの出力にフィルタリングすると、これらの情報を分離して抽出できます。こうした出力は、次のことに使用可能です。

- 出力のデータをテストしてフローの次のステップを決定するためのレスポンスルール（エバリュエーター）を作成する。  
注: レスポンスルールにフィルターを作成することも可能です。
- フィルタリング済みデータを値としてフロー変数に渡し、以降のフロー内でオペレーションにアクセスできるようにして、フロー変数を参照するプロンプトとトランジションの説明によりフローのユーザーがこのデータを使用できるようにする。
- フローが別のフローのステップ（つまり、親フローのサブフロー）である場合に、フィルタリング済み出力のデータをフロー結果のフィールドに渡して、プロパティが親フローのオペレーション、ステップ、トランジションで使用できるようにする。

注: ping オペレーションを使用するには、独自の ping オペレーションを作成するか、既存の Ping オペレーションをコピーしてそのコピーを使用します。既存の Ping オペレーションのコピーを使用する場合、使用可能な結果の一部からデフォルトで、フィルターで分離するデータが抽出されます。


## フィルターの作成

必要なデータを正確に抽出するために一連のフィルターを作成することがよくあります。


### オペレーション出力やステップ結果をフィルタリングするには

1. 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 [Properties] シートを [Outputs] タブで開き、[Edit Filters] をクリックします。

もしくは

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、[Available Outputs] で出力の行の右端にある右向き矢印 () をクリックします。

もしくは

ステップの結果にフィルターを作成するには、ステップの Inspector を開き、[Results] タブで、結果の行の右端にある右向き矢印 () をクリックします。

フィルターエディターが表示されます。

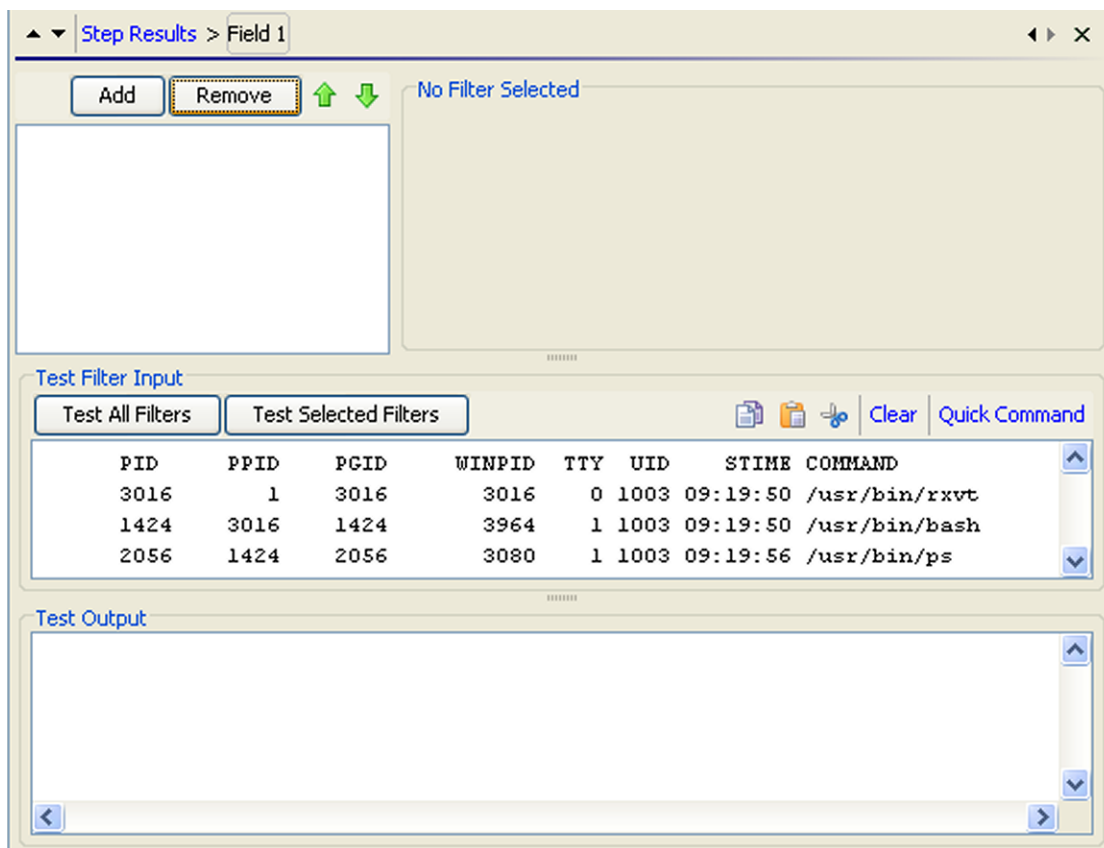


図 61 – フィルターエディター

- フィルターを作成すると、左上のボックスにフィルターのリストが作成されます。
  - フィルターを作成してフィルタータイプを選択すると、（上のスクリーンショットで [No Filter Selected] と表示されている）右上のボックスに、選択したフィルターの種類に応じてフィルター編集のコントロールが含まれます。
2. フィルターを追加するには、[Add] をクリックして、表示された [Select Filter] ボックスのドロップダウンリストからフィルターの種類を選択します（フィルターの種類については、「[フィルターの詳細](#)」を参照してください）。
  3. フィルターエディターの右上にあるエリアの [Details] で、フィルターの詳細を定義します。  
1つの結果に複数のフィルターを作成した場合、リストで表示されます。



**ヒント:** フィルターエディターを開いている間、上向き矢印または下向き矢印（▲ ▼）をクリックして、フィルターを作成する対象の結果を変更できます。

フィルターはオペレーション出力やステップ結果によく適用されますが、出力や結果のデータをフィルタリングする場合、データはフィルターの入力です。そのため、フィルターエディターでは、テストデータを [Test Filter Input] ボックスに入力します。



4. テストの実際のデータを取得するには、[Clear] をクリックして [Test Filter Input] ボックスを空にしてから、次のいずれかを実行します。
  - データがローカルコマンドラインコマンドで生成可能な場合、[Quick Command] をクリックし、表示されたテキストボックスに、目的のデータを生成するコマンドを入力します。
  - 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、次のことが可能です。
    - a. デバッガーでフローを実行する。
    - b. 関連するステップを強調表示する。
    - c. [Step Result Inspector] で、[Raw Result] タブのコンテンツをコピーする。
    - d. フィルターエディターで、コンテンツを [Test Filter Input] ボックスに貼り付ける。
5. フィルターエディターで、[Test All Filters] をクリックします。  
もしくは  
テストするフィルターを選択して [Test Selected Filters] をクリックします。  
フィルターが（上から順番に）[Test Filter Input] ボックスのデータに適用され、フィルタリング結果が [Test Output] ボックスに表示されます。

作成可能なフィルターの種類やフィルターの保存方法については、「[フィルターの詳細](#)」を参照してください。

## フィルターの詳細

各フィルターの詳細の定義については、次のサブトピックを参照してください（フィルターの作成については「[フィルターの作成](#)」を参照してください）。

- [Diff Case](#)
- [Extract Number](#)
- [Format](#)
- [Line Count](#)
- [Regular Expression](#)
- [Remove Duplicate Lines](#)
- [Replace](#)
- [Round Number](#)
- [Scriptlet](#)
- [Select Line](#)
- [Select Range](#)
- [Sort](#)
- [Strip](#)
- [Strip Whitespace](#)
- [Table](#)

次の説明では、フィルターが操作するデータは、オペレーション出力やステップ結果に由来するものであっても、フィルターの入力です。そのため、フィルターエディターでは、テストデータを [Test Filter Input] ボックスに入力します。



## Diff Case

**Diff Case** 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。[To Upper Case] チェックボックスの選択を解除した場合、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。

## Extract Number

**Extract Number** 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 つの数字と見なされます。たとえば、「123Test」や「Test123」という文字列からは、**Extract Number** フィルターにより「123」の数字が抽出されます。

## Format

**Format** フィルターは、結果や出力にテキストを付加したり、元のコンテンツをテキストに置換します。

- [Text] 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。
- [Place Input At] リストで、テキストを先頭に付加するか末尾に付加するかに応じて [Beginning] または [End] を選択します。

もしくは

出力をテキストに置換するには、[Replace] を選択します。

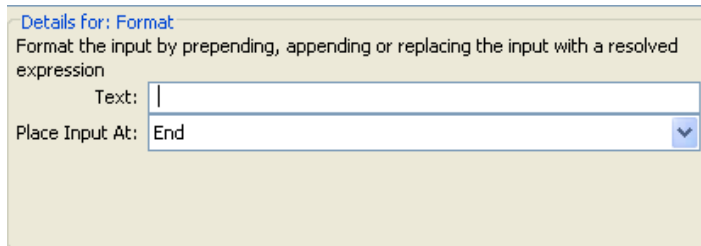


図 62 – Format フィルターの定義

## Line Count

**Line Count** フィルターは、結果に含まれる行の総数を出力します。このフィルターについて編集できる設定はありません。

## Regular Expression

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。正規表現の詳細については、「[正規表現での作業](#)」を参照してください。

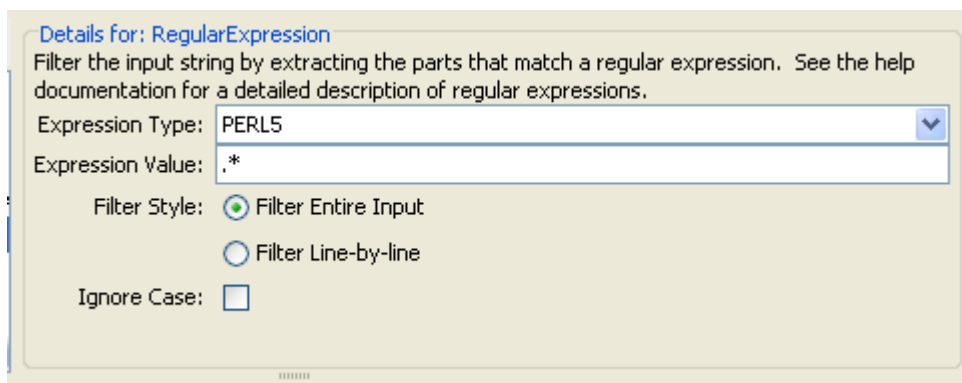


図 63 – Regular Expression フィルターの定義

- [Expression Type] で、適用する regex の種類を選択します。
- [Expression Value] に、regex を入力します。
- このフィルターをどのように未加工結果に適用するかに応じて、[Filter Entire Input] または [Filter Line-by-line] を選択します。
- regex の大文字小文字を区別しない場合は、[Ignore Case] を選択します。

## Remove Duplicate Lines

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、[Consecutive] を選択します。

## Replace

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。

図 64 — Replace フィルターの定義

- [Find] ボックスに、対象の文字列（検索して置換する文字列）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、[Replace] ドロップダウンリストから [First]、[All]、または [Last] を選択します。
- [With] ボックスに、対象文字列と置換する文字列を入力します。
- 検索時に大文字小文字を区別しない場合は、[Ignore case] チェックボックスを選択します。

## Round Number

フィルターエディターでの指定に従って数値を丸めます。

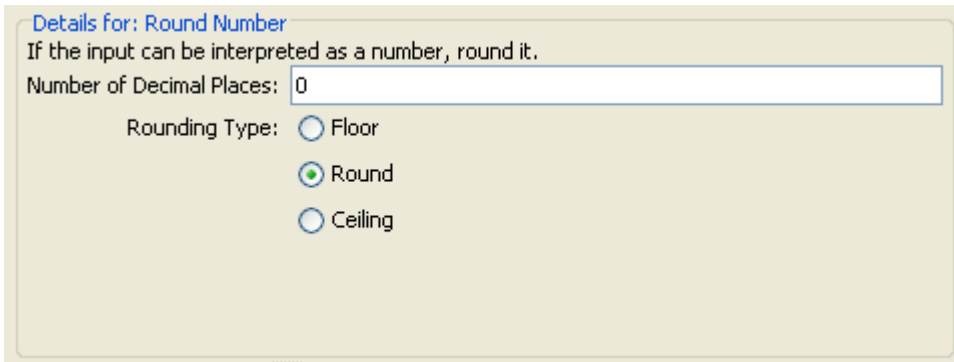


図 65 – Round Number フィルターの定義

- 丸めの精度を指定するには、数値を丸める小数位を [Number of Decimal Places] ボックスに入力します。
- [Floor]、[Round]、[Ceiling] のいずれかを選択して、数値の丸め方を指定します。[Floor] は常に数値を切り捨て、[Ceiling] は常に切り上げます。[Round] 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ます。

## Scriptlet

作成したスクリプトレットでデータをフィルタリングします。スクリプト記述環境は、オペレーションやステップのスクリプトレットを作成する場合と同様です。

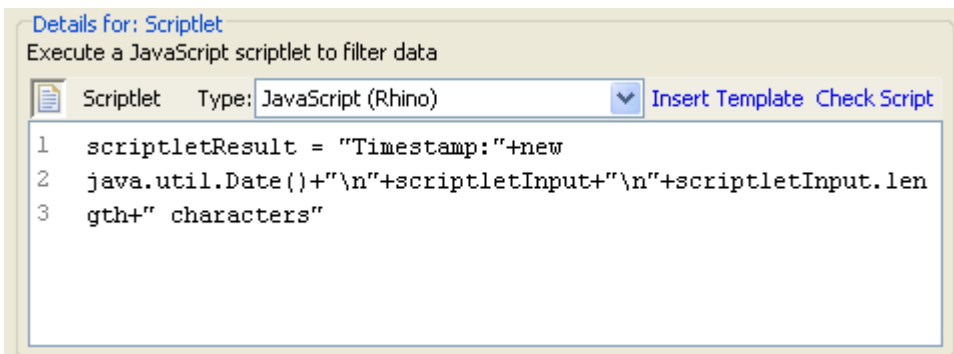


図 66 – Scriptlet フィルターの定義

- [Type] ドロップダウンリストから、スクリプトレット言語として [JavaScript (Rhino)] または [Sleep] を選択します。  
デフォルトは [JavaScript (Rhino)] です。これにより、テキストボックス内にスクリプトレットの最初の行が配置されます。
- スクリプトレットをフィルターとして使用するのに必要となる行で開始するには、[Insert Template] 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（値はコンテキストデータ）、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。
- スクリプトをデバッグするには、[Check Script] をクリックします。

## Select Line

未加工結果から抽出する行を指定します。

Details for: Select Line  
Select a line from the input  
Line Select Method: First Line Starting With  
Parameter:  
Line Ending Type:  Auto  Unix  Windows

図 67 — Select Line フィルターの定義

- [Line Select Method] リストから、対象とする行の基準を選択します。
- [Parameter] テキストボックスに、対象文字列を含む文字列を入力します。
- [Line Ending Type] グループから、フィルタリングするテキストが Unix (LF で行を終了) または Windows (CR/LF で行を終了) のいずれかで生成されたかに応じて行末の種類を選択します。デフォルトで選択される [Auto] は、行末の種類として Unix と Windows の両方を受け入れます。

## Select Range

入力データから抽出する文字列を指定します。文字列の指定基準は2つあり、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。

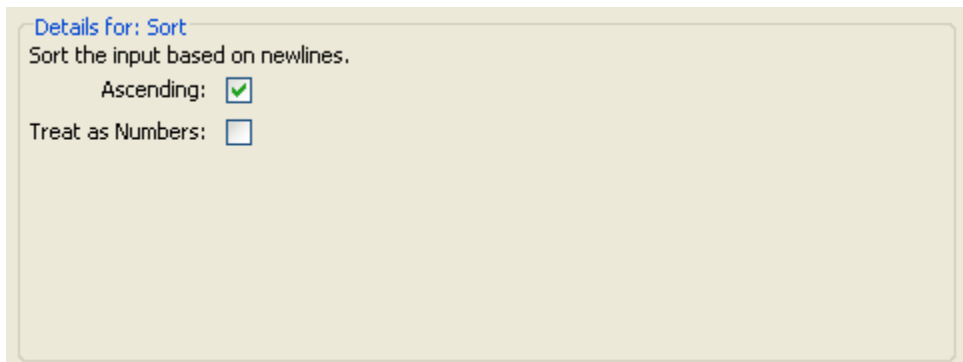
Details for: Select Range  
Specify a range to select from in the input (first character is 0)  
Start: 0  
Length: 0

図 68 — Select Range フィルターの定義

[Start] ボックスと [Length] ボックスに、未加工結果から抽出する文字列について、ゼロベースの開始位置と長さを入力します。

## Sort

入力データを行または列で並べ替えます。



Details for: Sort  
Sort the input based on newlines.  
Ascending:   
Treat as Numbers:

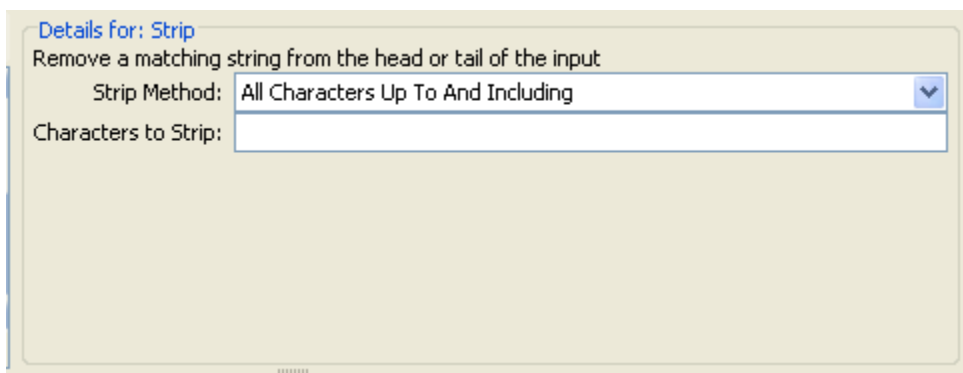
図 69 – Sort フィルターの定義

並べ替えの方向を指定します。昇順の場合は [Ascending] チェックボックスを選択し、降順の場合は [Ascending] チェックボックスの選択を解除します。

[Treat as Numbers] チェックボックスを選択してデータを数値として扱う場合、先頭が数字以外の行は並べ替えで一番下になります。

## Strip

未加工結果から文字を削除します。



Details for: Strip  
Remove a matching string from the head or tail of the input  
Strip Method: All Characters Up To And Including  
Characters to Strip:

図 70 – Strip フィルターの定義

[Strip Method] 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。[Characters to Strip] テキストボックスに文字列を指定し、その文字列以前のすべての文字（該当文字列を含むか含まないかを指定）、またはその文字列以降のすべての文字（該当文字列を含むか含まないかを指定）を削除することを指定できます。

[Characters to Strip] テキストボックスに、検索する文字列を入力します。

## Strip Whitespace

未加工結果の先頭または末尾から空白文字をすべて削除します。このフィルターについて指定できる設定はありません。

## Table

Table 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。

Details for: Table  
Parses the input as a table and sorts it on a specified column

Column Delimiter:	Whitespace	Row Delimiter:	NewLine
First Row is Header:	<input type="checkbox"/>	Strip First Row of Result:	<input type="checkbox"/>
Sort On Column:	-1	Ascending:	<input type="checkbox"/>
Select Row:	0	Select Col:	0
Select Width:	1	Select Height:	1

図 71 – Table フィルターの定義

注: 行の番号設定は 0 ベース、列の番号設定は 1 ベースです。

- [Column Delimiter] リストでは、データを列に分割するための文字を選択します。
- [Row Delimiter] リストでは、データを行に分割するための文字を選択します。

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、列の区切り文字に空白を指定して「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。

- 最初の行を列ヘッダーとするには、[First Row is Header] を選択します。
- 最初の行を削除するには、[Strip First Row of Result] を選択します。
- 列で並べ替えを行うには、[Sort On Column] ボックスに列番号を入力します。列の番号設定は 0 ベースです。

値 **-1** は列で並べ替えが行われないことを意味します。

- 昇順を指定するには、[Ascending] 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。
- フィルターで抽出する行を選択するには、[Select Row] に行番号、[Select Width] に抽出する行に含まれる列の数を入力します。行番号は 0 から始まります。
- フィルターで抽出する列を選択するには、[Select Col] に列番号、[Select Height] に抽出する列に含まれる行の数を入力します。列番号は 1 から始まります。

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。

- [Select Row]: 0
- [Select Height]: 5
- [Select Col]: 2
- [Select Width]: 3

[Select Row]、[Select Col]、[Select Width]、[Select Height] の各ボックスに値 -1 を指定した場合、それぞれ次のようになります。

- [Select Row] に -1 を指定すると、データ内のすべての行が選択されます。
- [Select Height] に -1 を指定すると、[Select Row] 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。
- [Select Col] に -1 を指定すると、データ内のすべての列が選択されます。
- [Select Width] に -1 を指定すると、[Select Col] 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。

## フィルターの保存と再使用

あるオペレーションの結果用に作成したフィルターを保存して、別のオペレーションに再使用することが必要になる場合があります。たとえば、**RAS Ping** のフィルターはほかの ping オペレーションに役立ちます。こうしたフィルターは、システムフィルターとして **Configuration** \ **System Filters** フォルダーに保存できます。

スクリプトレットフィルターをシステムフィルターとして保存すると、フィルター専用スクリプトレットを保存できます。

### フィルターを System Filters フォルダーに保存するには

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、保存するフィルターを選択します。
2. リポジトリペインで、**Configuration** フォルダーを展開します。
3. フィルターを、オペレーションのフィルターエディターから **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、[Rename] をクリックして詳しい名前を付けます。
5. 作業結果を保存します。

### System Filters フォルダーのフィルターを使用するには

1. システムフィルターを使用するエディターを開きます。
2. Library で、**Configuration** \ **System Filters** フォルダーを開きます。
3. 使用するフィルターを、フォルダーから結果エディターのフィルターリストボックスにドラッグします。

## レスポンス: 結果の評価

レスポンスは、オペレーションの結果です。たとえば、データベースに対して SQL クエリを実行するオペレーションの結果は、次のようになります。

- **Failure** レスポンス: データベースが実行されない、または到達できない場合。
- **No rows returned** レスポンス: クエリが実行されたがデータが戻されなかった場合。
- **Rows returned** レスポンス: クエリでデータの取得に成功した場合。




オペレーション結果の特定の状況を表すルールが該当する場合、特定のレスポンスが選択されます。ルールでは、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を比較します。

Response	Default	On-Fail	Type	Rules
port open	<input type="checkbox"/>	<input type="checkbox"/>		1 Rule [Source: returnCode, No Filters, Exact Ma...]
port closed	<input type="checkbox"/>	<input type="checkbox"/>		1 Rule [Source: returnCode, No Filters, Exact Ma...]
host not found	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1 Rule [Source: returnCode, No Filters, Exact Ma...]


図 72 - ステップのレスポンスとレスポンスの種類



上のスクリーンショットでは、ルールの一部を示しています。

- レスポンスの種類には次のようなものがあります。
  - Success (成功) または Resolved (解決済み) : 
  - Diagnosed: 
  - No Action (アクションなし) : 
  - Failure (失敗) : 
- 1つのレスポンス (この例では **port open** レスポンス) に複数のルールを作成する場合、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 [Responses] タブでの表示順で評価されます。ルールが該当すると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。
- レスポンスのルールで該当するものがない場合、指定した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。

### レスポンスを作成するには

- オペレーションの [Responses] タブで、[Add Response] をクリックして新しいレスポンスの名前を入力します。
- オペレーションが実行に失敗した場合に選択されるレスポンスを指定するには、[On-Fail] 列にあるレスポンスのチェックボックスを選択します。
- 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。
- レスポンスルールエディターで、[Add] をクリックします。
- [Apply Rule to Field] 列で、ルールをテストする対象のコンテンツを持つ結果フィールドを選択します。
- [Rule Type] 列で、ルールでテストを行う比較または一致を選択します。
- [Rule Text] 列に、テストで使用するテキストを入力します。



The 'Port listening' response will be selected when all of the following rules are true			Add	Remove
Apply Rule To Field	Rule Type	Rule Text		
code	Match At Least One Word			

図 73 - ルールの定義

ルールエディターを使用すると、前述の変更のほか、次のことが可能です。

- ルール適用前にオペレーション結果をフィルタリングする。
- ルールをテストする。
- システムエバリュエーター (ルール) 、フィルター、またはスクリプトレットをルールにドラッグする。

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が、評価は「false」となります。ただし、**部分一致**エバリュエーターで文字列を比較することにより、この問題に対処できます。

8. ルール詳細エディターを開くには、ルールの行の右端にある右向き矢印 (  ) をクリックします。

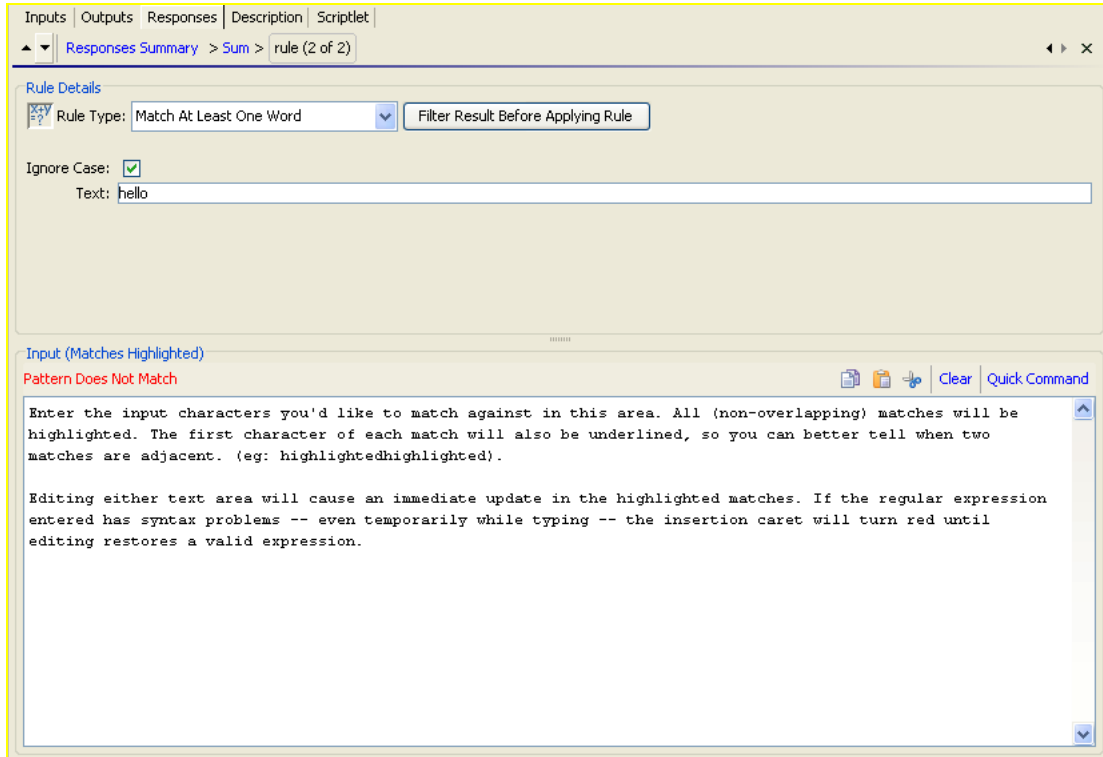


図 74 - ルール詳細エディター

注: ルールタイプとして [Scriptlet] を選択した場合、ほかの場所で使用する際にルール詳細エディターはスクリプトエディターに変わります。スクリプトレットの作成や使用の詳細 (再使用のためのスクリプトレットの保存も含む) については、「[スクリプトレット](#)」を参照してください。スクリプトレットフィルターの作成については、「[出力および結果のフィルタリング](#)」を参照してください。

これで、ルールをテストし、ルール適用前にフィールドの結果をフィルタリングするフィルターを作成できます。

レスポンスルールエディターで選択したルールタイプは、[Rule Type] ドロップダウンリストですでに選択された状態で**ルール詳細**エディターに表示されます。

9. 別のルールタイプを選択するには、[Rule Type] ドロップダウンリストでほかのタイプを選択します。
10. ほとんどのルールタイプについては、[Text] 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、[Ignore Case] チェックボックスを選択します。

もしくは

**正規表現**のルールでは、オペレーション結果の Regular Expression フィルターを作成するときの正規表現とその用途を指定します。**Regular Expression** フィルターの作成については「[出力および結果のフィルタリング](#)」、フィルタータイプの詳細については「[フィルターの詳細](#)」を参照してください。

11. オペレーションのレスポンスについて別のルールを使用するには、[Responses Summary] 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。



**ヒント:** レスポンスルールは、オペレーションの [Responses] タブでの表示順で評価されます。フローを実行すると、該当するものと評価された最初のルールのレスポンスがステップのレスポンスとして選択されます。そのため、レスポンスに指定した順序により、ユーザーがフローから目的の結果を得られるかどうかが決まる場合があります。

## フローへのレスポンスの追加

フローのレスポンスは、フロー実行の結果です。フローでシステムの調整が必要と判断されたかどうか、アクションが成功したかどうかなどを示します。

たとえば、**Find Inactive Domain Administrators** フローには3つのレスポンスがあります。次のレスポンスがどのように取得されるかを確認するには、Studio で Library の **Find Inactive Domain Administrators** フローを見つけてダブルクリックします。

- Failure (失敗) – 何らかの理由で、無効なドメイン管理者のリストを取得できなかった場合。
- Success (成功) – 無効なドメイン管理者が見つからなかった場合。
- Diagnosed (診断済み) – 無効なドメイン管理者が見つかった場合。

### フローにレスポンスを追加するには

1. フローの [Properties] シート (フロー図を開いて、ウィンドウ下部にある [Properties] タブをクリック) で、[Responses] タブをクリックします。
2. [Add Response] 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。



ヒント: フローのレスポンスを削除するには、[Responses] タブで、削除するレスポンスをクリックして [Remove Response] をクリックします。

3. 作業結果を保存します。

## ステップのアイコンの変更

ステップのアイコンは、そのステップの内容をわかりやすく表すものに変更できます。

### ステップのアイコンを変更するには

1. [Icons] ペインを開くには、フローキャンバスの右側にある [Icons] タブをクリックします。

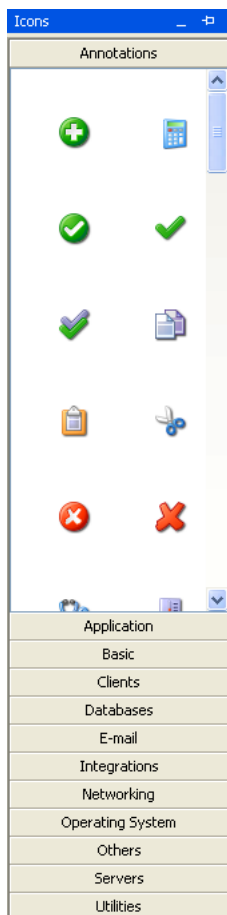


図 75 – [Icons] ペイン

2. [Icons] ペインで、必要なアイコンを含むサブパネル名をクリックして、目的のアイコンを探し、そのアイコンをステップにドラッグします。

## システムプロパティの作成

システムプロパティは、その内容とフローでの働きから、強力であると同時に、危険なこともあります。

システムプロパティは、OO のグローバルコンテキストに格納されるフロー変数です。システムプロパティの範囲は、リポジトリを越えてグローバルであるため、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。これにより、フローの作成や実行について望ましい効果をあげることが可能です。

一方、システムプロパティの範囲がグローバルであり、オペレーション、フロー入力、またはフローステップの入力の作成時や変更時に、システムプロパティの値を変更することはできないため、システムプロパティは慎重に作成する必要があります。また、フローを **Central** で実行する際に、**Central** ユーザーが異なる値をシステムプロパティに割り当てることもできません。入力がフロー変数から値を取得できる場合、その値は、ユーザーレスポンスや特定の値などのソースからは割り当てられません。システムプロパティはグローバルフロー変数であるため、リポジトリ内のいずれのフローの実行にも使用できます。

さらに、システムプロパティとその値は、**Studio Debugger** のみに表示されます。それ以外で、システムプロパティの存在を見つけられるのは、**Studio** で関連ステップを変更してもフローで入力について「誤った」値が使用される場合だけです。たとえば、次の値を指定したシステムプロパティ `#{overrideJRAS}` があるとします。

<https://htudor.north.mycorp.net:9004/JRAS/services/RCAgentService>

作成した入力、`${overrideJRAS}` から入力を取得しようとしたが、別の RAS を指す RAS 参照を指定できるユーザープロンプトから値を取得した場合、そのユーザープロンプトは表示されません。入力がすでにシステムプロパティから値を取得済みであるためです。また、システムプロパティを定義すると、`${overrideJRAS}` も使用するフローが中断、または予期しない結果が生じることがあります。

### システムプロパティを作成するには

1. Studio のリポジトリペインで、**Configuration** フォルダーを開き、**System Properties** フォルダーを右クリックして、[New] をクリックします。
2. 表示されたテキストダイアログボックスに、新しいシステムプロパティの名前を入力し、[OK] をクリックします。

システムプロパティの [Properties] シートが表示されます。

3. [Description] ボックスに、システムプロパティの説明を入力します。
4. [Property Value] ボックスに、システムプロパティの値を入力します。  
たとえば、OO リポジトリ全体に JRAS 上書きを指定するシステムプロパティを作成するには、次の手順で行います。

- システムプロパティに名前を付けます。

**overrideJRAS**

- 次のような値を指定します。

<https://htudor.north.mycorp.net:9004/JRAS/services/RCAgentService>

フローでのシステムプロパティの参照は次のようになります。

`${overrideJRAS}`

## フローデザイン

フローを作成する際には、次のことに注意します。

- 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、一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です。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。
- 「[サブフローを使用してフローデザインを単純化](#)」できます。

### サブフローを使用したフローデザインの単純化

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。

ある共有サーバーから別の共有サーバーにファイルセットをコピーするとします。次の手順で行うことができます。

1. 次のようなリスト入力を指定し（リストの要素の区切り文字としてセミコロン (;) を指定する）、**Iterator** オペレーションからステップを作成します。

```
¥¥server1¥share1¥fileAAA.zip, ¥¥server2¥share2¥fileAAA.zip;
```

```
¥¥server1¥share1¥fileBBB.zip, ¥¥server2¥share2¥fileBBB.zip;
```

最初の繰り返しで、**Iterator** ステップが

```
¥¥server1¥share1¥fileAAA.zip, ¥¥server2¥share2¥fileAAA.zip
```

を抽出します。

2. コピー元のパス名に ¥¥server1¥share1¥fileAAA.zip、コピー先のパス名に ¥¥server2¥share2¥fileAAA.zip を使用するには、2つの **ListItemGrabber** ステップを作成して、一方でコピー元のパス名を抽出し、もう一方でコピー先のパス名を抽出します。

また、次の手順により、フローを単純化し、再使用可能なステップを作成することもできます。

3. 2つの **ListItemGrabber** ステップ、**success** と **failure** のリターンステップのみを含むフローを作成します。

4. この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。

この場合、親フローで、サブフローにより作成または編集されたデータを使用できるようにする必要があります。この例では、親フローのコピーステップには、コピー元のパス名とコピー先のパス名がそれぞれ必要です。サブフローから親フローに情報を渡す方法については、「[サブフローから親フローへのデータの引き渡し](#)」を参照してください。

## サブフローから親フローへのデータの引き渡し

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、ステップの結果をサブフローのフロー出力フィールドに割り当てることにより、フロー外の値を親フローに渡すことは可能です。

### サブフローのステップのデータを親フローのステップで使用できるようにするには

1. サブフローの作成キャンバスで、親フローでデータを使用可能にするステップの **Inspector** を開きます。ステップの **Inspector** の開き方については、「[ステップの編集](#)」を参照してください。
2. **[Results]** タブをクリックして結果を追加し、結果の設定用に表示された行で、次の手順を実行します。
  - **[Assign To]** で、**[Flow Output Field]** を選択します。  
サブフローの出力フィールドに値が格納され、サブフロー外でデータが使用可能になります。
  - **[Name]** で、フロー出力フィールドの名前を入力します。
  - **[From]** で、**[Result Field:Result]** を選択します。
3. 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 **Inspector** を開きます。
4. 「[ステップの結果](#)」の手順に従って、**[Results]** タブをクリックしてステップの結果を作成します。
5. 新しいステップ結果の行で、**[Assign To]** にある **[Flow Output Field]** を選択します。

## ステップのベースとなるオペレーションの変更

フロー内のステップに別のオペレーションが必要なことが判明したが、そのステップの既存のトランジション（ステップからのとステップへのトランジションを含む）を保持したいとします。このような場合、ステップの Inspector が役に立ちます。

### ステップのベースとなるオペレーションを切り替えるには

1. フローの Design ビューで、ステップの Inspector を開いて [Advanced] タブをクリックします。

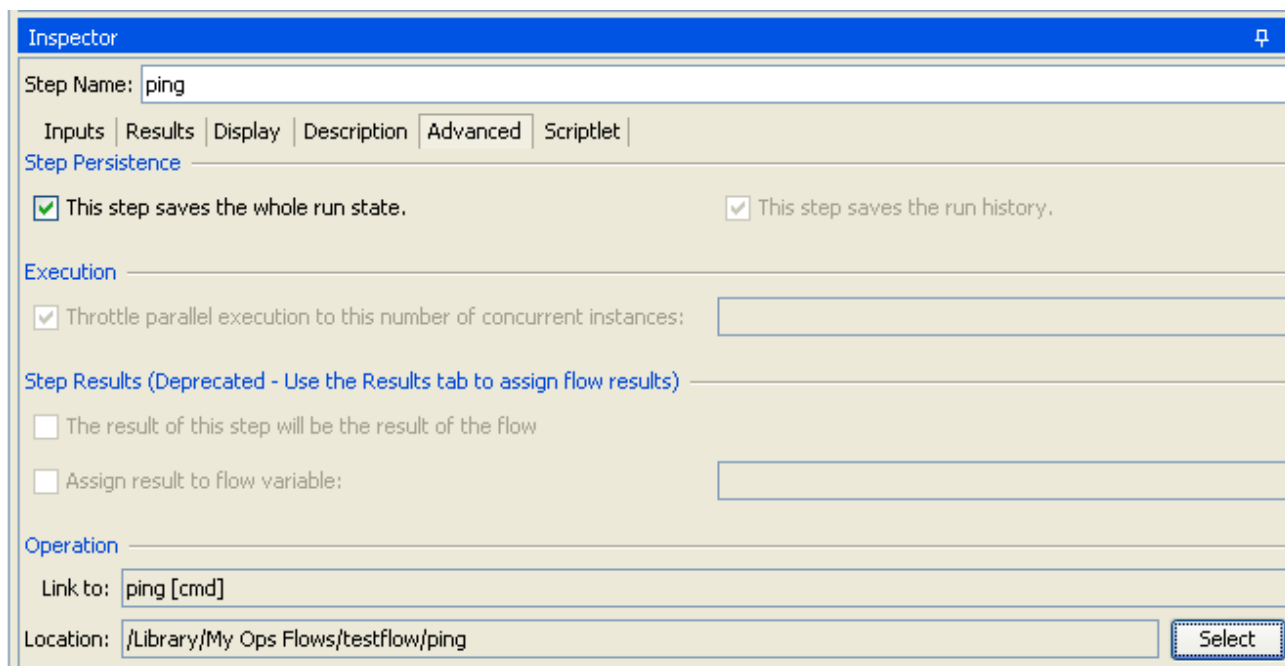


図 76 – ステップの Inspector の [Advanced] タブ

2. [Advanced] タブで、[Operation] にある [Select] をクリックします。  
[Select Operation] ダイアログボックスが表示されます。



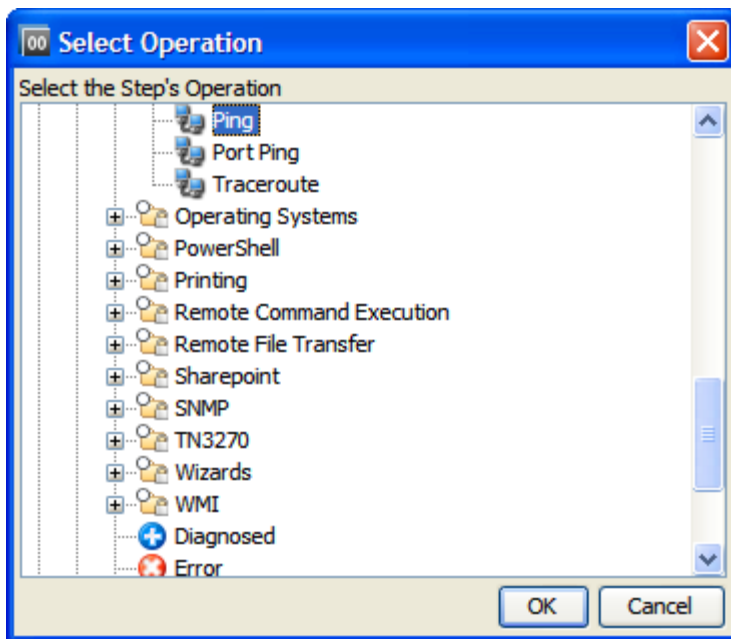


図 77 – [Select Operation] ボックス

3. ステップのベースとす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。  
注: ステップの名前は、最も基本となるオペレーションを変更しても影響を受けないため、ステップに以前のオペレーションの名前が付けられている場合、ステップは変更されていないように見えます。ただし、[Advanced] タブの [Operation] では、変更が反映されています。
4. 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。
5. 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。
6. 作業結果を保存します。

## ステップの説明

適切な検索語を含む説明は、フローを検索する際に役立ちます。たとえば、「Performs a ping and a traceroute to the server」という説明を持つオペレーションを探すとします。[Search] タブに、次のような検索条件を指定します。

`description:ping traceroute`

この検索の結果には、「ping」と「traceroute」のいずれかまたは両方をフローまたはステップの説明に持つフローとオペレーションが含まれます。

### ステップまたはフローの説明を作成するには

1. フローの [Properties] シートまたはステップの Inspector で、[Description] タブをクリックしてステップの説明を入力します。
2. [OK] をクリックして変更を保存します。

## フローの自動実行

フローが自動的に実行可能な場合、Central ユーザーは、フロー実行のスケジュールを設定し、URL からフローを開始できます。自動的に実行すると、フローの開始や完了に人の介入が不要になるため、ユーザープロンプトからデータを取得する入力をフローに含めることができません。また、フローの入力がフロー変数からデータを取得する場合、実行のスケジュールを設定する Central ユーザーは、各実行の入力に割り当てるデータを制御できます。

フローを自動的に実行できるようにするには、次の手順で行います。

- ユーザープロンプトからデータを取得する入力について、データソースを [Use Constant] に変更します。
- Central ユーザーが実行ごとに異なるデータを入力に割り当てられるようにするには、フロー変数を作成し、入力の [Use Constant] データソース割り当ての値についてそのフロー変数を参照します。  
入力のデータソースの変更については、「[入力: オペレーションへのデータの提供](#)」を参照してください。

## フローのデバッグ

Studio のデバッガーは、テストした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。デバッガーでは、次の情報が表示されます。

- 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

また、デバッガーのブレイ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。

この機能はすべて、中断したフローの調整と、説明対象のデータの種類の重点を置いています。パラレル処理を使用するステップを持つフローが Central でどのように動作するかについて詳しく知るには、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の Central でフローを実行する以外の方法はありません。

Studio デバッガーは、フローがパブリックリポジトリにある場合は Central で、プライベートリポジトリにある場合は Studio で、それぞれフローをデバッグします。これは、フローにパラレル処理が含まれる場合には特に重要です。Studio でのフローの動作が Central とは異なります。



**鍵となる情報:** フローのパラレル処理（1 つ以上のノンブロッキングステップ、マルチインスタンスステップ、または並列分岐ステップで実行される処理）は、Central ではパラレルで、Studio デバッガーではシリアルで実行されます。

- 並列分岐ステップでは、レーンは常に連続で実行されます。Central では、並列分岐ステップはすべて同時に開始し、終了の順序は、Studio で予測不能な変数ファクター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、どのレーンを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。一方、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。

- マルチインスタンスステップでは、インスタンスは連続で実行されます。つまり、実際の状況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。
- ノンブロッキングステップは、デバッガーではノンブロッキングステップとして動作しません。デバッガーでは、ノンブロッキングステップに続くステップは、ノンブロッキングステップが完了するまで実行されません。

## フローのデバッグ

Studio デバッガーを使用してフローをデバッグする場合、次のようになります。

- パブリックリポジトリに接続している場合、フローは **Central** でデバッグされます。
- プライベートリポジトリに接続している場合、フローは **Studio** でデバッグされます。



**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。

フローをデバッグするには、キーボードショートカットが便利です。デバッガーのキーボードショートカットについては、「[デバッガーのキーボードショートカット](#)」を参照してください。

次に示す手順は、**Central** でのフローのデバッグ（パブリックリポジトリに接続している場合）と **Studio** でのフローのデバッグ（プライベートリポジトリに接続している場合）の両方に使用できます。

### フローをデバッグするには

1. **Library** で、フローを右クリックして **[Debug]** をクリックします。  
もしくは

デバッグするフローの図を開き、**作成**ペインで、**[Debug Flow]** ボタン (🔍) をクリックします。

**Studio** デバッガーが開きます。デバッガーのツールバーには、フローの実行に使用するコントロールが含まれており、完了するかステップごとに実行するかを指定します。開始から終了までフローを実行する場合、実行は、設定したブレークポイントでのみ中断されます。

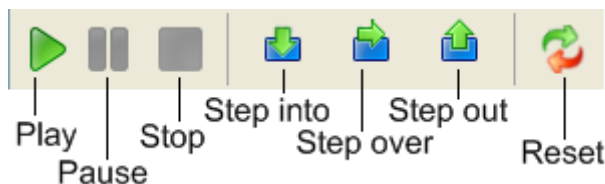


図 78 – Studio デバッガーのツールバー

2. フローを終了まで実行するには、**[Play]** アイコン (▶) をクリックするか、**F11** を押します。  
もしくは

フローをステップごとに実行するには、各ステップについて **[Step Over]** アイコン (📄) をクリックするか、**F6** を押します。

**Windows Health Check** フローとデバッガーについて見てみましょう。

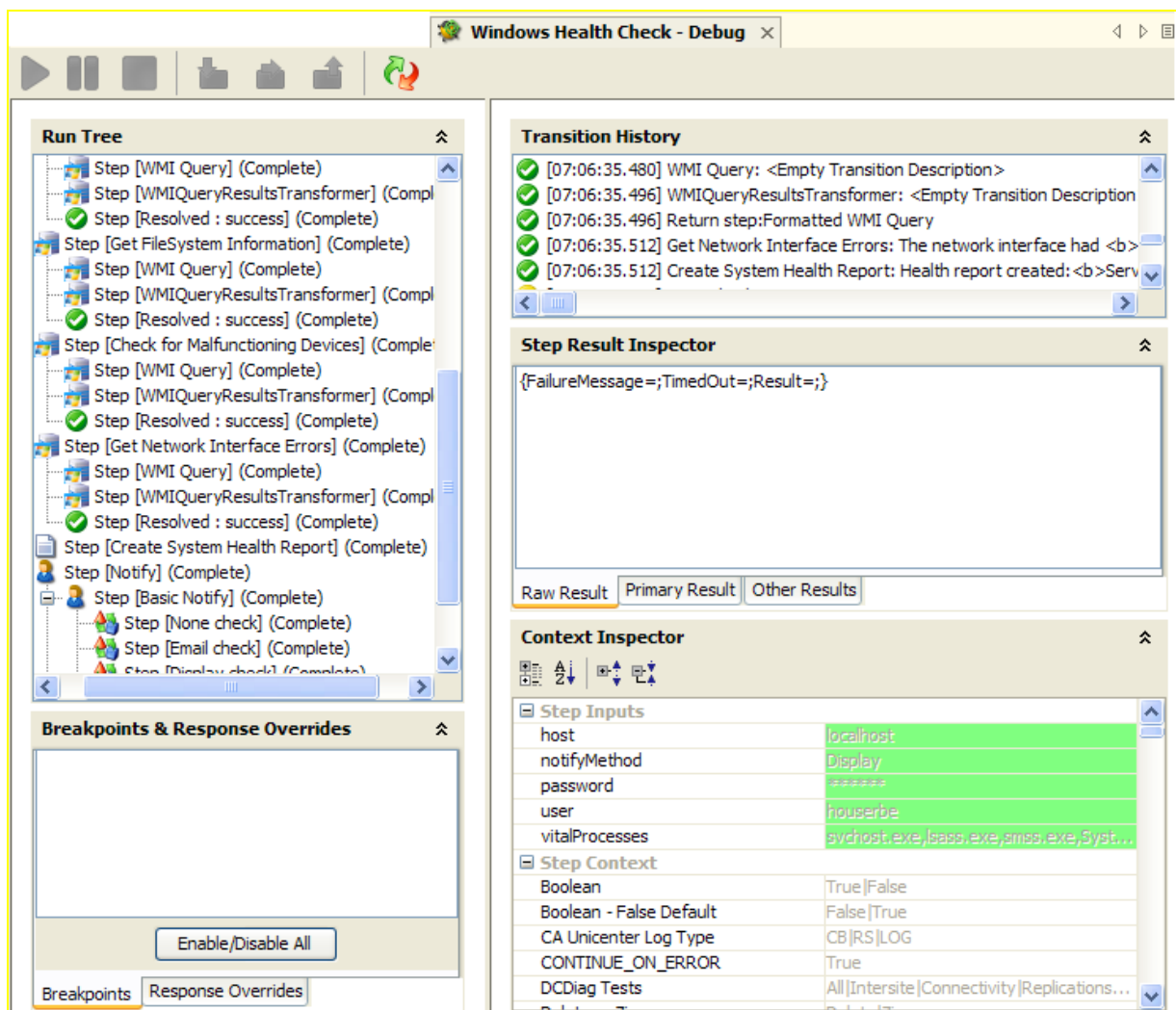
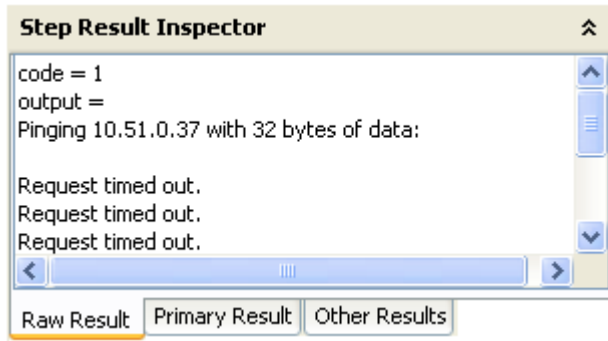


図 79 - フローデバッガーの一部

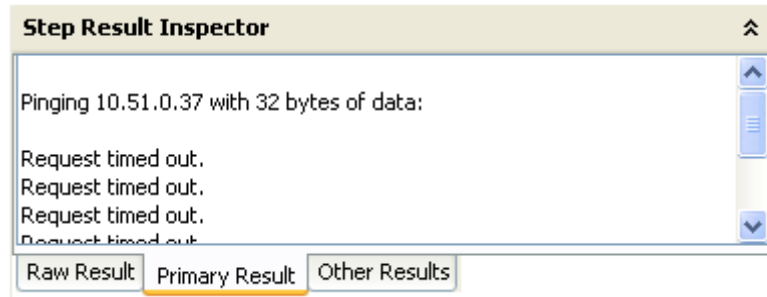
- [Run Tree] には、フローのサブフロー内のステップも含めて、実行する各ステップが表示されます。Central で同時に実行するステップは、デバッガーで連続して実行されます。
  - [Breakpoints & Response Overrides]: ブレークポイントは、特定のステップで実行を自動的に一時停止し、停止ポイントでのフロー変数の値、実行のパス、または結果を調べるためのフラグです。レスポンス上書きは、該当するオペレーションの結果の評価によって選択されたレスポンスとは関係なく、選択したレスポンスを強制的に指定します。このペインでは、これらの要素が一覧表示され、該当の実行について各要素の削除、有効化、または無効化が可能です。
  - [Transition History] には、実行内のトランジションとその説明が一覧表示されます（各トランジションについて説明を指定することをお勧めします）。
  - [Step Result Inspector] には、未加工結果（ステップのオペレーションの結果）と、ステップのフィルタリング済み結果が表示されます。
  - [Context Inspector] 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。
3. 完了したステップについて上記の各ペインの情報を確認するには、[Run Tree] で目的のステップをクリックします。

4. [Step Result Inspector] の説明は次のとおりです。

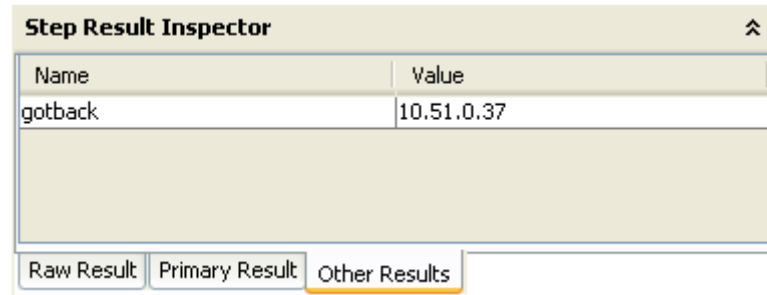
- ステップの未加工結果を表示するには、[Raw Result] をクリックします。



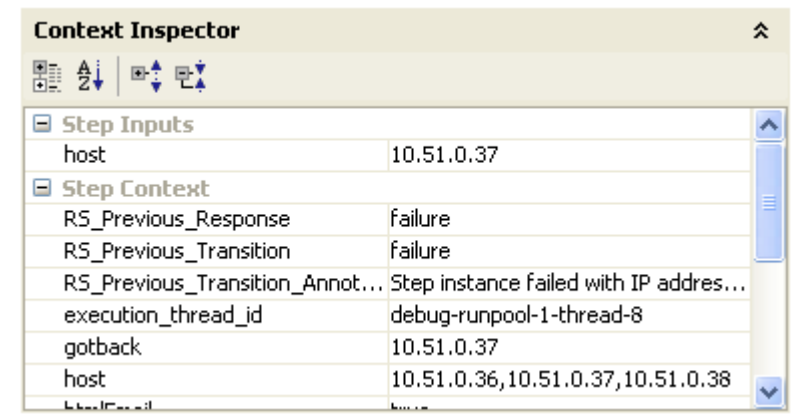
- 主結果を表示するには、[Primary Result] をクリックします。



- 作成した結果がほかにもあり、そうした結果を表示するには、[Other Results] をクリックします。



5. フロー変数、ステップの入力の値、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、[Context Inspector] の該当セクションに移動します。



[Context Inspector] の説明は次のとおりです。

- ステップ入力の値は、ステップ開始前に入力に割り当てられた値です。
- [Step Context] 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）。

この例のステップ（マルチインスタンスの **Ping** ステップのインスタンスの1つ）の場合、[Step Inputs] にはステップの **host** フロー変数、ステップのコンテキストには **gotback** フロー変数と、マルチインスタンスステップのこのインスタンスのコンテキストの **host** フロー変数が含まれます。

フロー変数の値を含むテキストボックスは、次の例に示すように色分けされます。

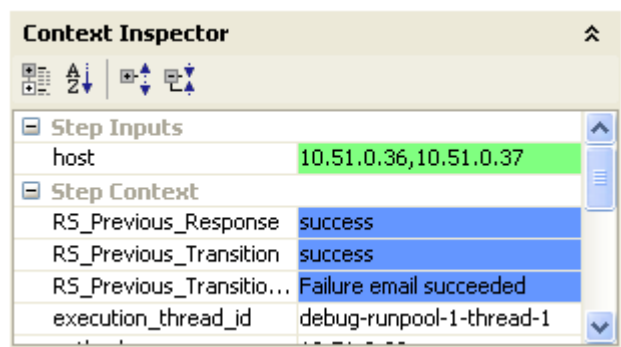





図 80 – フロー変数の色分け

- 青色は、実行で変更された値を示します。
- 緑色は、新しい変数を強調表示します。

### サブフローにステップイン、またはサブフローからステップアウトするには

1. ステップのサブフローにステップインするには、[Step-Into] アイコン () をクリックするか、F5 を押します。
2. サブフローからステップアウトするには、[Step-Out] アイコン () をクリックするか、F7 を押します。

### デバッガーでフローをリセットして再開するには



- ツールバーで、[Reset] アイコン () をクリックするか、F12 を押します。  
フローが再開される前に、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。

## 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。

### デバッガーでフロー実行中にフロー変数を変更するには

1. デバッガーでフローを開きます。
2. フロー変数を変更するステップに移動するには、次のいずれかを実行します。

- 目的のステップが保留中になるまで、[Step-Over] ( または F6) をクリックします。
- ステップのブレークポイントを設定済みである場合、[Play] () をクリックします。  
ブレークポイントを設定したステップで、実行が一時停止します。ブレークポイントの設定については、「[ブレークポイントの使用](#)」を参照してください。

[Context Inspector] に、ステップが保留中になっているポイントの [Step Inputs] および [Step Context] の現在の値が表示されます。

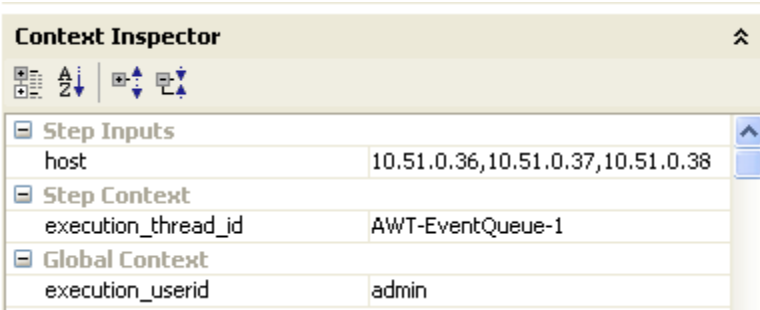


図 81 – デバッガーの [Context Inspector]

ここで、次のことを実行できます。

- このステップの実行で使用される値を変更する。
  - 後続のステップで使用される値を変更する。
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、フロー変数はステップ入力の値を指定する必要があります。


そのため、フロー変数が [Step Inputs] に一覧表示されるので、ここでフロー変数の値を変更します。ここで示した例では、ステップはマルチインスタンスステップです。ホストフロー変数のリストに別の IP アドレスを追加できます。

もしくは

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、[Step Context] に表示される該当リストの値を変更します。

4. フローの実行を続行します。

もしくは

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、[Reset] アイコン () をクリックします。

## ブレークポイントの使用

ブレークポイントは、Studio 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。ブレークポイントを設定したステップで、次に示すような作業を行うときに便利です。

- 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。



## ブレークポイントを設定するには

- 作成ペインでフローを開いて、ブレークポイントを設定するステップを右クリックし、[Debugging] をポイントして [Set Breakpoint] をクリックします。

注: ブレークポイントの有効化や無効化を行うのはデバッガーですが、デバッガー内からブレークポイントを作成することはできません。

## ブレークポイントを有効または無効にするには

- デバッガーでフローを開くには、ツールバーの [Debug Flow] アイコン (🌐) をクリックします。
- デバッガーの [Breakpoints & Response Overrides] ペインの [Breakpoints] タブに、既存のブレークポイントが表示されます。

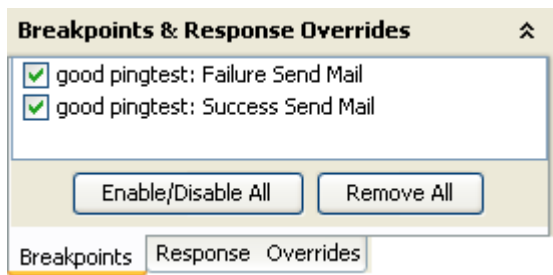


図 82 – ブレークポイントの有効化と無効化

- ブレークポイントのチェックボックスを選択または選択解除します。

## デバッグ実行でのレスポンスの上書き

レスポンスを上書きすると、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。次に挙げるのがその例です。

- あるフローのステップに、必要な情報がないとします。このステップが失敗していることとは関係なく、フローの残りのステップについてテストが必要です。介入なしに発生した **failure** レスポンスではなく、指定したレスポンスとトランジションに強制的に実行を続けることができます。
- また、フローの失敗パスで実行のテストが必要な場合には、**success** レスポンスを上書きすることも可能です。

### 単一ステップにレスポンス上書きを設定するには

- 作成ペインでフローを開き、レスポンスを上書きするステップを右クリックします。
- ドロップダウンメニューで、[Debugging] に続いて [Override Response] をポイントし、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。

選択できるレスポンスのオプションは、**None**、**success**、**failure**、および **Prompt** です。

レスポンス上書きを作成した後、デバッガーで上書き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。

### レスポンス上書きを有効化または無効化、別のレスポンスを選択するには

- デバッガーでフローを開くには、ツールバーの [Debug Flow] アイコン (🌐) をクリックします。
- デバッガーの [Breakpoints & Response Overrides] ペインの [Response Overrides] タブに、既存のレスポンス上書きが表示されます。



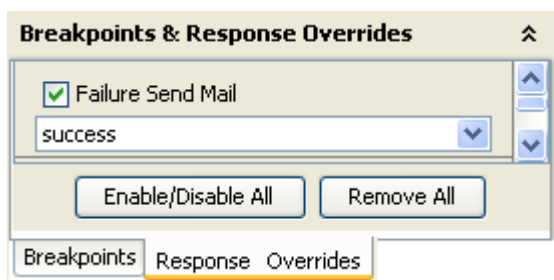


図 83 - レスポンス上書きの使用

3. 目的のレスポンス上書きにスクロールします。
4. レスポンス上書きを有効または無効にするには、チェックボックスを選択または選択解除します。
5. 上書きの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。

## 別のユーザーとしてログオン

Studio を停止して再起動しなくても、別のユーザーとして Studio にログオンできます。これは、Central のパブリックリポジトリにパブリッシュできるユーザーアカウントが 1 つしかなく、別のユーザーとしてローカルの Studio リポジトリで作業する場合に必要になります。

### 別のユーザーとしてログオンするには

- [File] メニューで、[Switch User] をクリックします。  
これで、別のユーザーとしてログオンできます。

## リポジトリ: フローおよびオブジェクト用のライブラリ

フロー、オペレーション、それらが参照するすべての OO オブジェクト（ドメインターム、リモートアクションサービス、スクリプトレット、選択リストなど）は、XML ファイルの構造的セットである *リポジトリ* に格納されます。

Studio をインストールすると、デフォルトで Central のパブリックリポジトリに接続されます。ただし、作業は、ローカルリポジトリ、つまり、Studio に対してローカルで Central サーバーの外部であるプライベートリポジトリで行うことができます。そのため、フローの作成には 2 つの方法があります。

- パブリックリポジトリで作業する。  
この場合、1 つの Central サーバーで IT インフラに影響を与えずにフローを作成し、本稼働環境にある別の Central サーバーでフローを実際に実行することを前提としています。

この方法で作業を行い、フローの作成者が複数いる場合、フロー、オペレーション、その他のリポジトリオブジェクト（システムアカウント、システムフィルター、ドメインタームなど）をチェックアウトし、チェックインしてパブリックリポジトリに対する変更を確定します。パブリックリポジトリでの作業に関するチェックアウトやチェックインの要件により、変更内容の競合が回避されます。



**鍵となる情報:** どのユーザーアカウントも、Studio の 2 つの異なるインスタンスから Central の同一インストールのパブリックリポジトリに接続していないことを確認する必要があります。その場合、リポジトリ内にあるユーザーの作業スペースが壊れることがあります。また、Central がクラスタ化されたときに起こることもあります。

開発サーバーで完了したフローを実際に実行する準備ができれば、そのフローを開発用の Central サーバーから本稼働環境の Central サーバーにパブリッシュします。

- ローカルリポジトリで、オフラインでフローを作成してから、Central サーバーのパブリックリポジトリにパブリッシュする。

オフラインで作業する（つまり、Central のパブリックリポジトリでは作業しない）場合、作業対象のオブジェクトをチェックアウトする必要はありません。ただし、Central リポジトリにパブリッシュする際には、影響を受けるオブジェクトを Central リポジトリにチェックインする必要があります。

オフラインの場合、リポジトリの選択した部分をエクスポートおよびインポートすることも可能です。

ローカルリポジトリにアクセスできるのは、そのリポジトリの作成者のみです。作成者以外が、ローカルリポジトリを更新したり、ローカルリポジトリにパブリッシュすることはできません。

## 作成用にリポジトリを追加して開く

次のような場合、デフォルトのパブリックリポジトリに加えて、別のリポジトリで作業することが必要になることがあります。

- 複数の作成者がいる環境で作業する。
- ほかのフローとは別個に、特定のバージョンのフローで作業を続行する。

複数のリポジトリの使用が必要になる状況については、「[リポジトリ: フローおよびオブジェクト用のライブラリ](#)」を参照してください。

別のリポジトリで作業するには、そのリポジトリを追加して開きます。

### リポジトリを追加するには

- [Repository] メニューで、[Add Repository] をクリックします。  
次のダイアログボックスが表示されます。

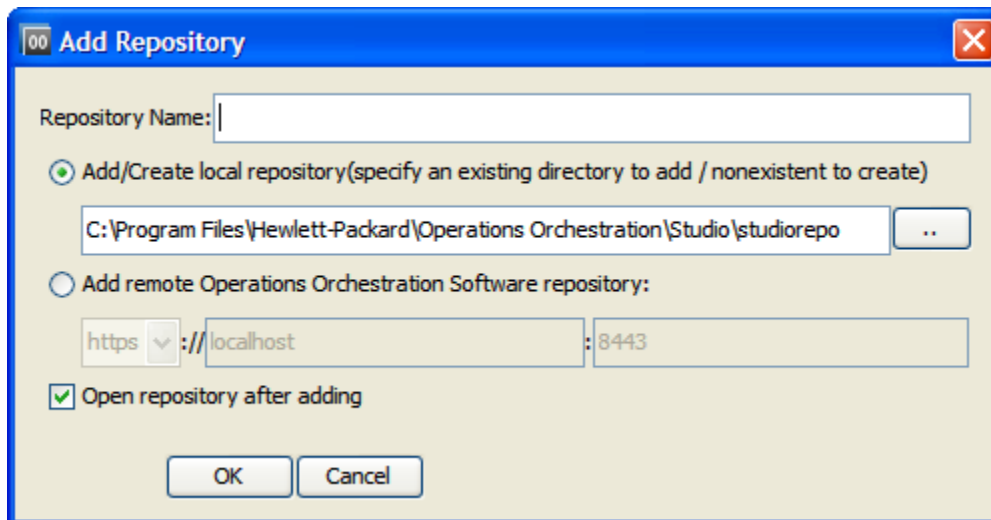


図 84 – リポジトリの追加

- [Repository Name] テキストボックスに、リポジトリの名前を入力します。  
任意の名前を付けることができます。
- ローカルリポジトリを追加するには、[Add/Create local repository] を選択して、リポジトリフォルダーの場所を指定します。

もしくは

Central サーバーにリポジトリを追加するには、[Add remote Operations Orchestration Software repository] を選択して、次のことを行います。

- a. 通信プロトコルのドロップダウンリストで、Central が通信に使用するプロトコルを選択します。  
デフォルトでは、Central は **https** プロトコルを使用します。
- b. リストの右にあるテキストボックスに、Central サーバーの名前と、HP OO 通信に使用するポートを入力します。  
デフォルトでは、Central が https プロトコルを使用する場合のポートは **8443** です。

4. リポジトリを作成した後に開くには、[Open repository after adding] をクリックします。

5. [OK] をクリックします。

リポジトリが、使用可能なリポジトリのリストに追加されます。

### 作成用にリポジトリを開くには

1. [Repository] メニューで、[Open Repository] をクリックします。
2. 使用可能なリポジトリのリストから、作業するリポジトリを選択します。

## リポジトリ間のフロー要素の移動

フロー、オペレーション、フローで使用するその他のオブジェクトを交換するには、2つの方法があります。

- リポジトリ全体または一部のエクスポートやインポートの目的は、社外の作成者とフローおよび関連オブジェクトをやりとりすることです。  
リポジトリのエクスポートでは、スタンドアロンリポジトリが作成されます。作成者がエクスポートされたフローをインポートする際には、リポジトリで生じる競合を手動で解決する必要があります。  
リポジトリをインポートする際には、インポートするオブジェクトを選択できます。
- リポジトリ全体または一部の更新やパブリッシュの目的は、同一のパブリック (Central) リポジトリへのアクセスを共有しているほかの作成者が、フローおよび関連オブジェクトを使用できるようにすることです。  
フォルダーをパブリッシュまたは更新する際には、フォルダー内のすべての項目、フォルダー内のフローで使用または参照されるすべてのオブジェクトが対象となります。つまり、パブリッシュまたは更新するオブジェクトを選択することはできません。

インポート、エクスポート、更新、パブリッシュの手順については、このセクションの次のトピックを参照してください。

- [ターゲットリポジトリの設定](#)
- [リポジトリのパブリッシュ: 方法](#)
- [リポジトリからの更新: 方法](#)
- [パブリッシュまたは更新のロールバック](#)
- [リポジトリのエクスポート](#)
- [リポジトリのインポート](#)

リポジトリ間でフロー、オペレーション、その他の OO オブジェクトのパブリッシュや更新を行うには、最初にターゲットリポジトリを設定する必要があります。ターゲットリポジトリではないリポジトリは、ソースリポジトリになります。

注: 開いているリポジトリをターゲットリポジトリに設定することはできません。

リポジトリのストレージは Studio 外にあるため、エクスポートやインポートを行う場合はターゲットリポジトリを設定する必要があります。

- パブリッシュする場合、ソースからターゲットにパブリッシュを行います。
- 更新する場合、ターゲットからソースに更新を行います。

## ターゲットリポジトリの設定

### ターゲットリポジトリを設定するには

1. 開いているリポジトリがターゲットとして設定するリポジトリである場合は、別のリポジトリを開きます。
2. [Repository] メニューで、[Set Target Repository] をポイントし、ターゲットにするリポジトリをクリックします。

もしくは

ターゲットリポジトリを設定しない場合は、[None] をクリックします。

## パブリックリポジトリからのパブリッシュとパブリックリポジトリへの更新

パブリッシュと更新は、直接作業していないターゲットリポジトリ、つまり各自の Studio で開いていないリポジトリと、フローを交換する方法です（一方、Studio が Central のパブリックリポジトリに接続している場合、パブリックリポジトリの作業スペースで作業しているので、作業結果をそのリポジトリで作業している他の作成者にも使用可能にするには、チェックイン/チェックアウトを使用します）。

他の作成者が作業しているパブリックリポジトリにパブリッシュする場合は、パブリッシュの過程で [Apply] アイコンをクリックすると、OO によって、リポジトリにパブリッシュしようとしている変更の対象となるオブジェクトのチェックアウトが試行されます。パブリッシュしようとしているオブジェクトを他の作成者がチェックアウトしていた場合、パブリッシュは失敗します。チェックイン/チェックアウト機能の詳細については、「[ライブラリオブジェクトのバージョン管理](#)」を参照してください。

ターゲットの Central サーバーがステージング環境または本稼働環境のいずれにあっても、Central のリポジトリとの間でパブリッシュまたは更新を行うことによってフローを交換します。他のフロー作成者と共同で作業する場合に推奨される方法は、ステージングサーバーに Central をインストールしておき、それぞれの作成者は各自のコンピューターにインストールした Studio 上で作業するというものです。この手法により、次のことが可能になります。

- 作成者は、（ステージングサーバー上の）Central のパブリックリポジトリとの間でパブリッシュや更新を行うことにより、互いに相手の作業結果を使用できます。作成者 A が Central にパブリッシュを行い、作成者 B はパブリックリポジトリから自分の（B の）ローカルリポジトリを更新することによって A の作業結果を取得できます。
- ステージングサーバーでフローのテストが正常に終了したら、作成者のどちらかがステージングサーバーのリポジトリをロックし、Central の本稼働インストールで使用されているリポジトリにフローをパブリッシュします。

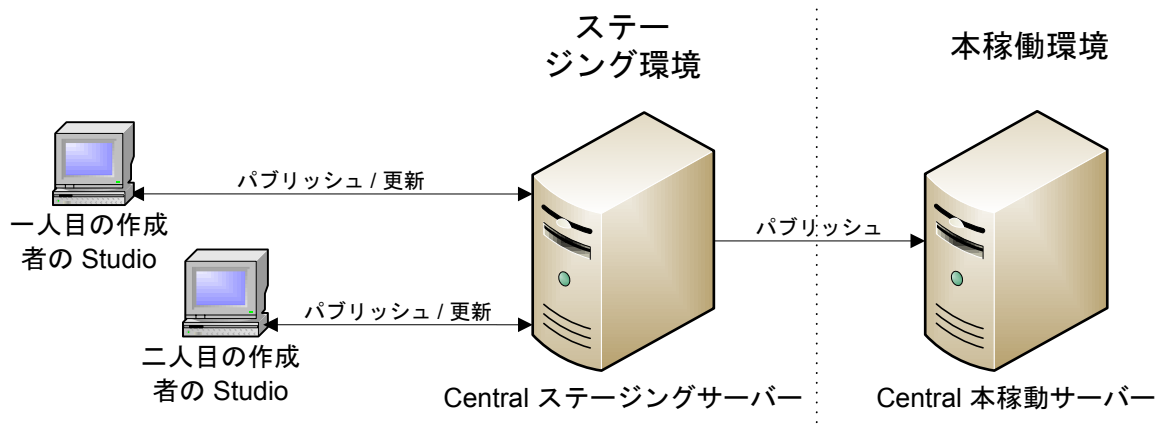


図 85 - 二人以上の作成者がフローを作成する場合の推奨手法

パブリッシュまたは更新を行うとき、パブリッシュするオブジェクトのグループを選択できます。ただし、フローをパブリッシュする場合、従属オブジェクト（つまり、そのフローで使用するオペレーションやシステムオブジェクト）すべてがフローとともにパブリッシュまたは更新されるという制約があります。

フロー、オペレーション、システムオブジェクトのパブリッシュと更新の詳細については、「[リポジトリ間のフロー要素の移動](#)」を参照してください。

## リポジトリのパブリッシュ: 方法

パブリッシュは、新規または変更されたオブジェクトを、ソースリポジトリからターゲットリポジトリにコピーすることです。ソースリポジトリは、パブリッシュまたは更新を開始するリポジトリです。ソースはローカルリポジトリである必要があり、ターゲットは Studio が接続されている Central のパブリックリポジトリである必要があります。

**注:** パブリッシュを行う場合、以前にターゲットにパブリッシュしたけれどもソースには存在しない変更は、ローカル（ソース）のリポジトリの状態によって上書きされません。たとえば、2つのローカル（ソース）リポジトリ A と B があり、パブリック（ターゲット）リポジトリに次の順でパブリッシュします。

1. ローカル（ソース）リポジトリ A と B にある Testflow1 のバージョンは、同期されていて同一です。
2. 作成者 A が Testflow1 にステップを追加し、パブリック（ターゲット）リポジトリにパブリッシュします。
3. パブリックリポジトリには、現在、新しいステップが追加されています。
4. 二人の作成者が、新しいステップの追加は誤りだったとして意見が一致します。
5. パブリックリポジトリの誤りを修正するために、作成者 B が元の Testflow1 のバージョンをパブリックリポジトリにパブリッシュしようとしています。

作成者 B がパブリッシュを行うと、変更はありませんというメッセージが表示され、パブリックリポジトリの Testflow1 には新しいステップが残ったままになります。

パブリックリポジトリの Testflow1 のバージョンから不要なステップを取り除くには、更新またはパブリッシュのプレビューステージを使用します。[Update/Publish Preview] ペインで、パブリッシュまたは更新するオブジェクトを選択します。

**注:** ターゲットリポジトリのバックアップをインポートすることで、実質的にパブリッシュを元に戻すことができます。方法については、「[パブリッシュまたは更新のロールバック](#)」を参照してください。

このセクションでは、パブリッシュとパブリッシュプレビューの両方の手順を説明します。

## 留意点:

- リポジトリをパブリッシュまたは更新する前に、ターゲットリポジトリが設定済みであり、ソースリポジトリが開かれていることを確認してください。ターゲットリポジトリの設定の詳細については、「[ターゲットリポジトリの設定](#)」を参照してください。
- パブリックリポジトリからローカルリポジトリを更新する場合は、ローカルリポジトリに取得するフローをチェックイン済みであることを確認してください。
- パブリッシュする場合、ソースからターゲットにパブリッシュを行います。更新する場合、ターゲットからソースに更新を行います。
- フォルダー内のすべてのアイテムに対する書き込み権を持っていないかぎり、パブリッシュの操作でフォルダーを削除することはできません。

## リポジトリにパブリッシュするには

- ローカルリポジトリを開いた状態で、パブリックリポジトリをターゲットリポジトリとして設定します。
- [Studio] ウィンドウの下部にある [Publish/Update] タブをクリックし、[Pin] アイコン (📌) をクリックして [Publish/Update] ペインを開いたままにします。



**ヒント:** パブリッシュしているときにターゲットリポジトリから更新する必要があることがわかった場合は、[Publish/Update] ペインのツールバーの左端で、[Update Preview] アイコン (🔄) または [Publish & Update Preview] アイコン (🔄) のいずれかをクリックします。

ソースとターゲットにある2つのリポジトリのオブジェクトのバージョン間に競合がある場合は、[Publish/Update] ペインにその内容が一覧表示されます。

- ペイン内の各フォルダーを完全に展開するには、プラス記号のアイコン (+) をクリックします。

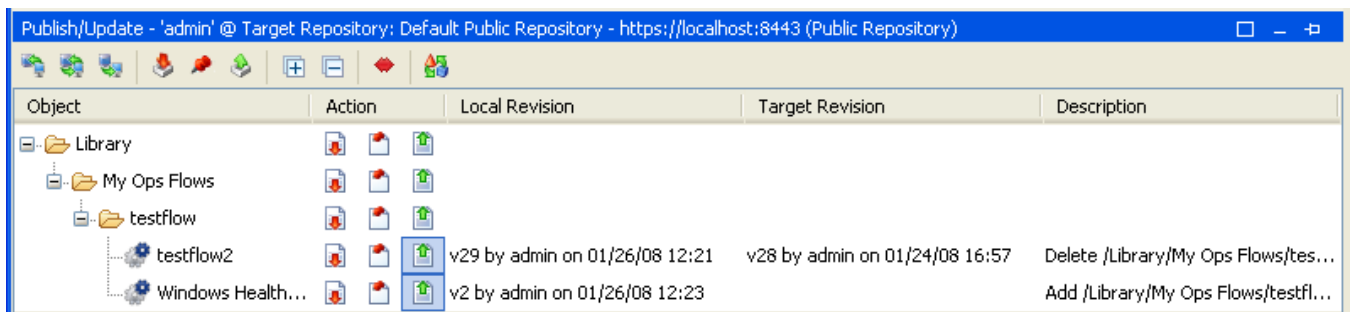


図 86 – [Publish/Update] ペイン

ペイン内の次の情報に注意してください。




- ローカルとターゲットの両方のバージョンに関するバージョン情報と最終更新日時。
- 変更を適用することによって行われる操作 ([Delete] または [Add]) と、変更が行われるリポジトリのパス名。
- 内容が異なっているけれども前回リポジトリをパブリッシュしてから変更していないフォルダーのコンテンツが一覧表示されますが、これらに対しては [No Change] (📄) の操作が選択されています。
- オブジェクトのバージョン間に競合がある場合は、オブジェクトの2つのバージョンを、個別に、または横に並べて調査できます。



4. いずれかのバージョンを個別に表示する場合は、そのオブジェクト（例では Testflow1）の行で右クリックし、表示されたメニューで、調査する対象に応じて [Open Local] または [Open Target] をクリックします。  
[Publish/Update] ペインの上に、Studio の作成ペインで、選択したバージョンのフロー図が開きます。  
もしくは  
両方のバージョンを横に並べて表示するには、そのオブジェクト（Testflow1）の行で右クリックし、[Compare] をクリックします。
5. オブジェクトを比較して必要な変更を行った後、各オブジェクトまたはフォルダーに対する操作を選択します。




ヒント: [Publish & Update Preview] では、一部のオブジェクトをパブリッシュし、その他を更新するように選択できます。

- ターゲットリポジトリのオブジェクトを変更するには、上向き矢印 () をクリックします。
- ソースからオブジェクトまたはフォルダーを削除するには、下向き矢印 () をクリックします。
- 何もしない場合は、[No Change] アイコン () をクリックします。

もしくは

オブジェクトを右クリックし、対応するコマンドを選択します。

- ターゲットからオブジェクトを更新するには、[Modify in local] を選択します。
  - ターゲットにオブジェクトをパブリッシュするには、[Modify in target] を選択します。
  - 何もしない場合は、[No Change] を選択します。
6. 指定した変更を適用するには、[Apply] アイコン () をクリックします。
  7. [Enter Publish Comment] ダイアログボックスが開くので、パブリッシュする変更についてコメントを入力し、[OK] をクリックします。
  8. すべての変更が正常に適用されたというメッセージが表示されたら、[OK] をクリックします。
  9. 作業結果を保存します。

## リポジトリからの更新: 方法

リポジトリからの更新は、リポジトリへのパブリッシュの逆です。しかし、ソースリポジトリがパブリッシュまたは更新を開始するリポジトリであることは変わりません。ソースはローカルリポジトリである必要があり、ターゲットは Studio が接続されている Central のパブリックリポジトリである必要があります。よって、更新する場合、ターゲットからソースに更新を行います。

更新を開始する場合、更新の中止や、ターゲットのパブリックリポジトリからローカルのソースリポジトリにコピーするオブジェクトの選択はできません。ただし、更新のプレビューは可能です。また、事後に更新を実質的に元に戻すことができます。

- 更新をプレビューする場合は、ローカルリポジトリに更新する OO オブジェクトを選択して更新することができます。
- 更新を元に戻すには、ソースリポジトリのバックアップをインポートする必要があります。方法については、「[パブリッシュまたは更新のロールバック](#)」を参照してください。



ヒント: パブリックリポジトリから更新する前に、対象のフローがターゲットリポジトリにチェックイン済みであることを確認してください。



## リポジトリから更新するには

- ローカル（ソース）リポジトリを開いた状態で、**更新元**のリポジトリをターゲットリポジトリとして設定します。  
ターゲットリポジトリの設定の詳細については、「[ターゲットリポジトリの設定](#)」を参照してください。
- [Repository] メニューから、[Update Source from Target – Preview] を選択します。  
もしくは  
[Studio] ウィンドウの下部にある [Publish/Update] タブをクリックします。
- [Pin] アイコン (📌) をクリックし、[Publish/Update] ペインを開いたままにします。  
ソースとターゲットにある2つのリポジトリのオブジェクトのバージョン間に競合がある場合は、[Publish/Update] ペインにその内容が一覧表示されます。更新する場合、ターゲットとソースのリポジトリで同じ場所にあり同じ名前を持つけれども ID が異なるオブジェクトについて、競合がレポートされます。  
たとえば、次の場合を考えます。
  - ローカルリポジトリとパブリックリポジトリは同期されていて、ともに Testflow1 を含んでいます。
  - パブリックリポジトリで、Testflow1 を削除し、Testflow1 の別のバージョンまたはインスタンスを追加します。現在、パブリックリポジトリの Testflow1 の ID は、ローカルリポジトリの Testflow1 の ID とは異なっています。パブリックリポジトリからローカルリポジトリへの更新をプレビューすると、Testflow1 について競合がレポートされます。
- ダイアログボックス内の各フォルダーを完全に展開するには、プラス記号 (⊕) をクリックします。  
**重要:** オブジェクトのバージョン間に競合がある場合は、オブジェクトの2つのバージョンを、個別に、または横に並べて調査できます。

この図では、ローカルとターゲットのリポジトリに競合している Testflow のバージョンがあり、別のローカルリポジトリからパブリックリポジトリにパブリッシュされた新しいフロー、Testflow2 が存在します。

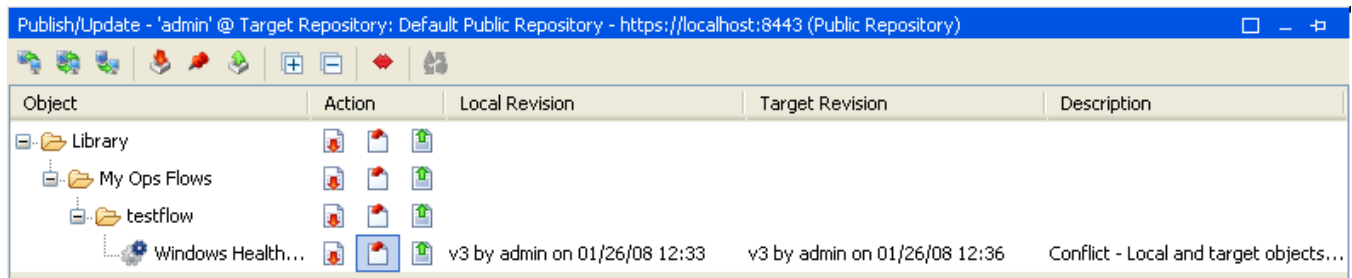


図 87 – 競合が存在する [Publish/Update] ペイン

- いずれかのバージョンを個別に表示する場合は、そのオブジェクト (Testflow1) の行で右クリックし、表示されたメニューで、調査する対象に応じて [Open Local] または [Open Target] をクリックします。  
[Publish/Update] ペインの上に、Studio の**作成**ペインで、選択したバージョンのフロー図が開きます。  
もしくは

両方のバージョンを横に並べて表示するには、そのオブジェクト (Testflow1) の行で右クリックし、[Compare] をクリックします (または、その行をダブルクリックします)。

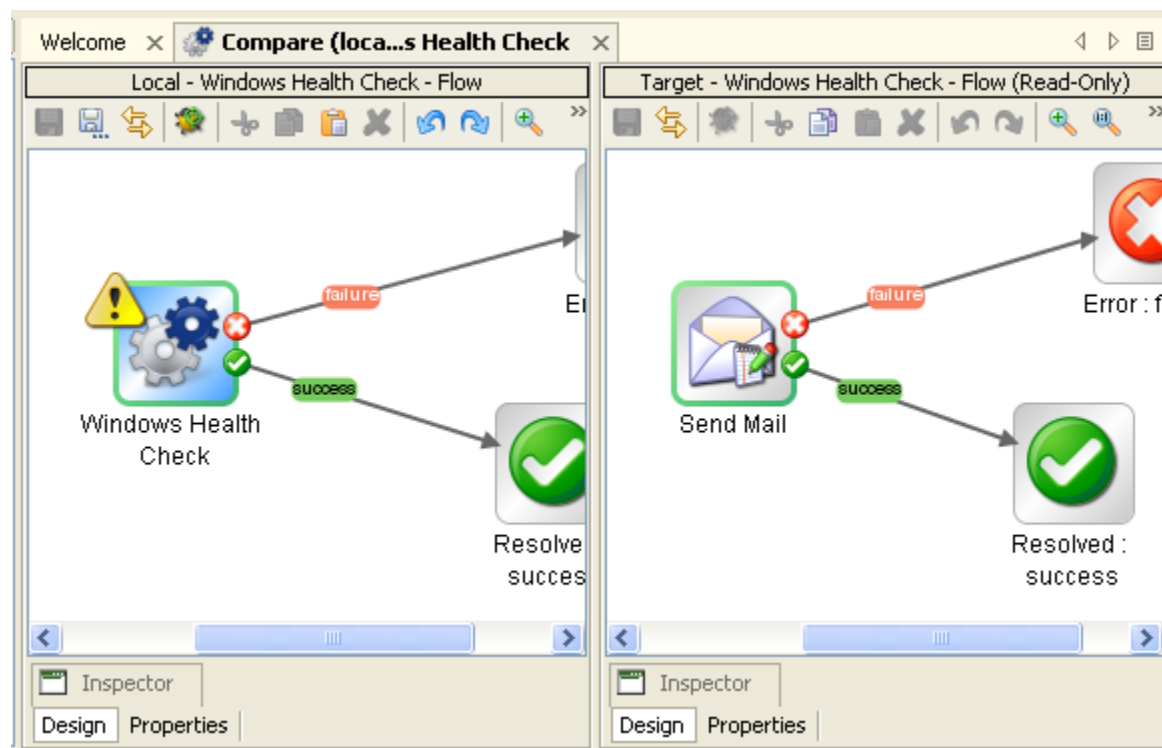





図 88 - フローの 2 つのバージョンの比較

ローカルリポジトリのフロー図で、変更を行い、保存できます (ターゲット (パブリック) リポジトリのフロー図は読み取り専用です)。

6. オブジェクトを比較した後、各オブジェクトまたはフォルダーに対する操作を選択します。



**ヒント:** [Publish & Update Preview] では、一部のオブジェクトをパブリッシュし、その他を更新するように選択できます。

- 更新している場合、 をクリックすると、ローカルリポジトリをパブリックリポジトリのオブジェクトで更新できます。
- 更新している場合、 をクリックすると、ローカルバージョンのオブジェクトをパブリックリポジトリにコピーできます。
- 何もしない場合は、[No Change] アイコン () をクリックします。


もしくは

オブジェクトの行を右クリックし、[Select Incoming] (ローカルリポジトリの競合バージョンをパブリックリポジトリから更新する) または [Select Outgoing] (ローカルリポジトリの競合バージョンをパブリックリポジトリにパブリッシュする) のいずれかをクリックします。

もしくは

競合の受信バージョンまたは送信バージョンを選択する意味の詳細を表示するには、そのオブジェクトの行を強調表示して右クリックし、対応するコマンドを選択します。

- ターゲットからオブジェクトを更新するには、[Modify in local] を選択します。
- ターゲットにオブジェクトをパブリッシュするには、[Modify in target] を選択します。
- 何もしない場合は、[No Change] を選択します。

7. 指定した変更を適用するには、[Apply] アイコン () をクリックします。
8. 作業結果を保存します。

## パブリッシュまたは更新のロールバック

リポジトリをパブリッシュまたは更新し、後からリポジトリに対するそれらの変更が不要だったことがわかった場合は、リポジトリを開いてバックアップをインポートすることにより、実質的に元に戻す、つまりロールバックすることができます。

リポジトリは、変更が行われてから 10 分後に、自動的に .jar ファイルにバックアップされます。したがって、次の手順を実行します。

### パブリッシュまたは更新の結果を元に戻すには

- .jar をインポートし、作業結果を保存します。  
詳細については、「[リポジトリのインポート](#)」を参照してください。

## リポジトリのエクスポート

フローや OO オブジェクトを、パブリックリポジトリを共有していない作成者にも利用できるようにするために、フローとその従属オブジェクトをリポジトリとしてエクスポートできます。これを、他の Studio の作成者がインポートできます。

リポジトリを安全なバックアップ場所にエクスポートすることは、フローやその従属オブジェクトのバックアップを作成する方法でもあります。

### リポジトリをエクスポートするには

1. Library で、エクスポートする OO オブジェクトを含むフォルダーを右クリックし、[Repository] をポイントして、[Export as New Repository] をクリックします。
2. [Select Repository Directory] ダイアログボックスで、新しいフォルダーを配置する場所に移動します。
3. [File name] ボックスに、フォルダー名を入力します。  
既存のフォルダーは指定できません。
4. [Save] をクリックします。  
[Export Options] ダイアログボックスが表示されます。



図 89 – エクスポートのオプション

5. [Export Options] ダイアログボックスで、エクスポートされるリポジトリに含めないアイテムを選択します。エクスポートしたリポジトリを、リポジトリを使用する Studio のインストールにインポートする場合は、HP OO コンテンツを含める必要はありません。いずれかのフローで HP OO コンテンツが使用されていて、エクスポートしたリポジトリをインポートせずに開く場合は、エクスポートに必ず HP OO コンテンツを含めてください（つまり、エクスポートから除外しないでください）。

星印のついたアイテム（RAS、選択リスト、システムアカウント、HP OO コンテンツ）は、エクスポートするフローやオペレーションで参照されていない場合、（除外するためにマークしなくても）エクスポートから除外されます。

エクスポートするリポジトリに、Windows または Linux のシェルコマンドを実行するオペレーションを使用するフローを Shell Wizard を使用して追加する場合は、Library のサブフォルダーではなく、Studio の Library 全体をエクスポートすることをお勧めします。Shell Wizard の使用方法については、「[Shell Wizard: シェルオペレーションを使用するフローの作成](#)」を参照してください。

Library のサブフォルダーをエクスポートし、結果として得られるエクスポートで Shell Wizard または Web Service Wizard（Web Service Wizard の詳細については、「[Web サービスからのオペレーションの作成](#)」を参照）を指定する場合、エクスポートから OO コンテンツを除外しないでください。それらを指定する場合は、リポジトリに SSH Shell または Telnet Shell のいずれかのオペレーションが含まれている必要があります。SSH Shell または Telnet Shell オペレーションは、デフォルトの OO コンテンツの一部であるため、結果として得られるリポジトリに Shell Wizard を使用してシェルオペレーションを追加する場合、エクスポートに OO コンテンツを含める必要があります。

リポジトリのエクスポートから除外できるその他のアイテムとして、ある環境に固有のアイテムであって、別の環境で使用するためにリポジトリをエクスポートする場合には、次のものを除外できます。

- リモートアクションサービス（RAS）

RAS は、フローが HP OO 外やリモートコンピューター上でコマンドを実行できるようにしたり、他のアプリケーションプログラミングインターフェースと統合できるようにしたりする RAS への参照です。

- 選択リスト  
ユーザーに表示される、保存されたリストです。
- システムプロパティ  
固定の値を持つ、多くのフローで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。
- システムアカウント  
参照に使用されるシステムアカウント名の背後に隠されているユーザー資格情報です。

**重要:** 選択するアイテムは、エクスポートから除外するアイテムであることに注意してください。つまり、エクスポートしないアイテムです。

6. 各自が作成した、または各自が変更の権限を持つ、エクスポート対象のフロー、オペレーション、HP OO オブジェクト（選択リストなど）について、全員に読み取り、書き込み、実行、リンクの権限を与えるには、[Give EVERYBODY group full access to exported items for which you have write access] を選択します。
7. [OK] をクリックします。

## リポジトリのインポート

追加で既存のフローやオペレーション（他者が作成したものや、新しい Accelerator Pack に含まれているもの）を取得するには、リポジトリフォルダー、Java アーカイブ（.jar）ファイル、または .zip ファイルのいずれかの形式で保存されているリポジトリであれば、それをインポートできます。

フローをインポートする場合、フローやそのオペレーションで使用されるシステムフィルターとして保存されている、ドメインタームやフィルターなどの、そのフローの設定可能な OO オブジェクトもインポートされます。

**留意点:** フローやオペレーションをインポートする場合、オブジェクトそのもの（オブジェクトのコピーやインスタンスではなく）がインポートされます。さらに、オブジェクトはリポジトリ内の 1 箇所には存在できません。オペレーションは、オペレーションを使用するフローに関連するライブラリ内の任意の場所に配置できます。

### リポジトリをインポートするには

1. Library で、フォルダーを選択するか、または作成します。
2. [Repository] メニューで、[Import Repository] を選択します。  
作成ペインでフローまたはオペレーションを開いている場合は、メッセージが表示され、プログラムですべてのエディターを閉じることを許可するよう求められます。
3. メッセージを確認したら、[OK] をクリックします。
4. [Select Repository Directory（またはリポジトリを含む .jar / .zip）] ダイアログボックスが開くので、インポートするリポジトリ（Accelerator Pack の場合もある）、.jar ファイル、または .zip ファイルを含むディレクトリに移動し、[Open] をクリックします。

Studio によりソースとターゲットのリポジトリの相違が確認された後、[Importing from...] ダイアログボックスが開きます。

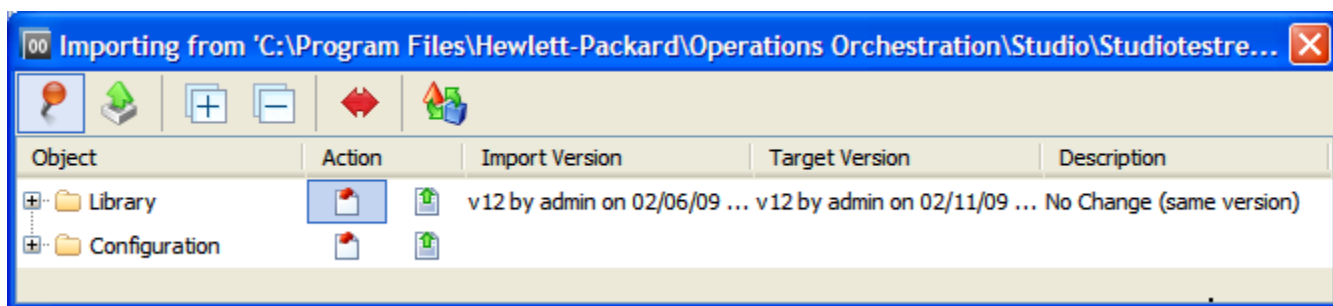


図 90 – ダイアログボックスからのインポート

このダイアログボックスでは、インポートするコンテンツのフォルダーが一覧表示されます。

- **Library** フォルダーには、インポートするフローやオペレーションが含まれます。
- **Configuration** フォルダーには、ドメインタームと OO オブジェクトが含まれます。

フォルダーを展開すると、ダイアログボックスには、そのフォルダー内のインポートされる各オブジェクトについて、この情報が表示されます。フォルダーに対して表示されている操作が [No Action] (📄) であっても、そのフォルダー内に調査すべき変更が存在する場合があります。

5. ダイアログボックス内の各フォルダーを完全に展開するには、ツールバーのプラスのボタン (+) をクリックします。  
フォルダーを展開すると、ダイアログボックスには、各 OO オブジェクトについて同様の情報が表示されます。

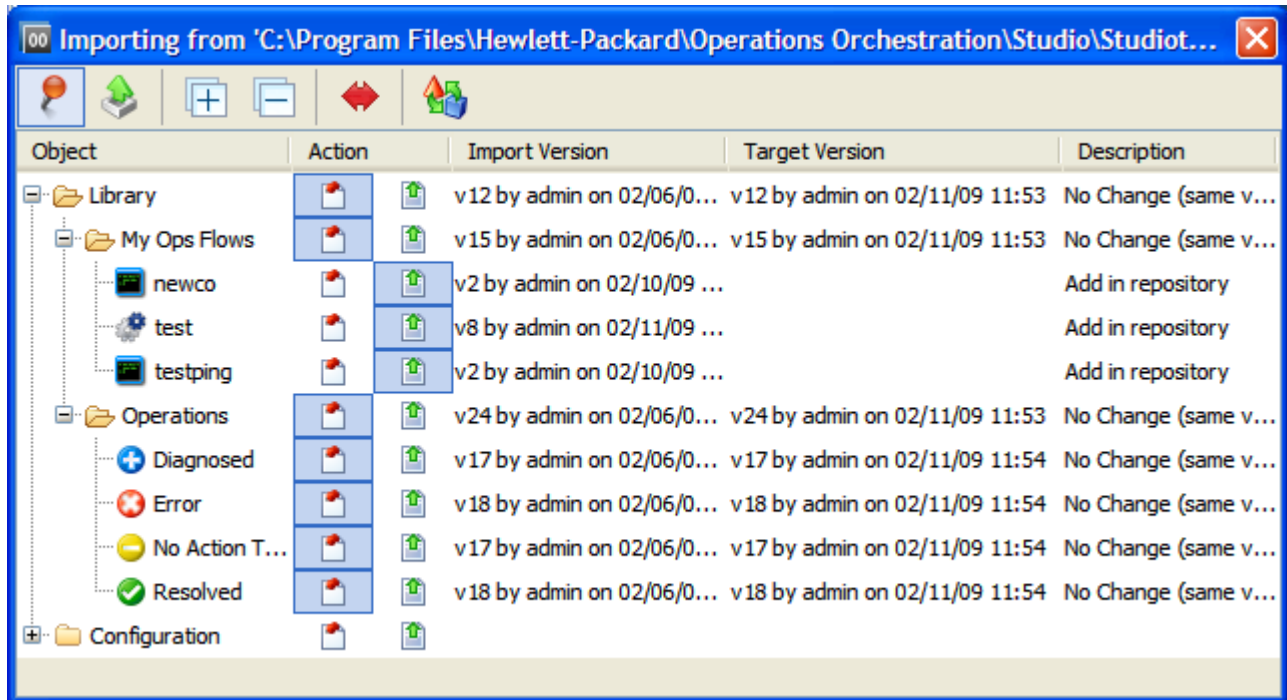


図 91 – インポートするオブジェクトの選択


ターゲットとソースの両方に存在するアイテムについて、ターゲットリポジトリのそのアイテムの次の情報に特に注意してください。

- そのアイテムのターゲットとソースのリビジョン（バージョン）番号。
- アイテムの作成や変更をだれがいつ行ったか。
- 変更内容の説明、または日付スタンプ以外に 2 つのアイテムの間に相違がないことの注釈。
- ところで、この例では、フォルダーとそのコンテンツはエクスポートされてから変更されていません。エクスポートしてから変更していないフォルダーや OO オブジェクトをインポートする場合、フォルダーまたはオブジェクトが一覧表示されますが、その [No Change] アイコン (📄) が強調表示されています。[Apply] ボタン (📁) をクリックしても、そのアイテムには何の操作も行われません。

注: [View All Changes] ボタン (🔍) をクリックすると、競合だけを表示できます。

6. フォルダーとそのコンテンツ、または OO オブジェクトをターゲットリポジトリにインポートするには、[Action] の下で、適切な行の上向き矢印 (📁) をクリックします。  
もしくは



ソースのオブジェクトを現在のままにしておくには、[No Change] アイコン (  ) をクリックします。



ヒント: 親フォルダーの上向き矢印または [No Change] アイコンをクリックすると、フォルダー内のすべてのオブジェクトに同じ操作を実行できます。

7. ダイアログボックスのツールバーで、[Apply] ボタン (  ) をクリックします。

リポジトリがエクスポートされる場合、Studio の外に配置されるので、そのインポート時にターゲットリポジトリを設定する必要はありません。

## リポジトリの検査

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。個々のフローまたはオペレーションの問題の有無は、Library のフローやオペレーションを強調表示し、[Problems] タブをクリックすることによって確認できます (「[有効なフローの要件](#)」を参照してください)。

または、存在する可能性のあるすべての問題を一度に発見して修正するために、リポジトリ全体を検査できます。この場合、リポジトリのすべてのフロー、オペレーション、システムアカウントが検査されます。

### リポジトリを検査するには

- リポジトリを開いた状態で、[Tools] メニューから、[Validate Flows and Operations] をクリックします。  
[Problems] タブを使用したときと同様に、問題がその場所、説明とともに一覧表示され、問題を修復する方法が示されます。

## リポジトリの暗号化

ローカルリポジトリの暗号化されたコピーを作成することにより、作業のスナップショットを改ざんや盗難から保護できます。ローカルリポジトリを暗号化する場合、リポジトリのコピーを作成してそのコピーを暗号化します。暗号化されたリポジトリについてパブリッシュ、更新、インポート、エクスポートを行うなど、何らかの方法で暗号化されたリポジトリを変更するには、正しいパスワードを入力する必要があります。

注:

- 暗号化できるのはローカルリポジトリのみです。Studio でパブリックリポジトリを開いている場合、リポジトリの暗号化に関するコマンドは使用できません。
- リポジトリ内のどのフォルダーを選択していても、リポジトリ全体が暗号化されたコピーにエクスポートされます。
- リポジトリを暗号化する場合、元の暗号化されていないリポジトリは Studio で開かれたままになります。
- ある場所に暗号化されたリポジトリのコピーを作成したら、その同じ場所に対してリポジトリの再暗号化や別のリポジトリの暗号化はできません。

### リポジトリの暗号化

リポジトリを暗号化する場合、リポジトリをエクスポートして、エクスポートされたリポジトリを暗号化します。

### リポジトリを暗号化するには

- 暗号化するリポジトリを開いた状態で、[Repository] メニューで [Encrypt Repository] をクリックします。  
次のダイアログボックスが表示されます。



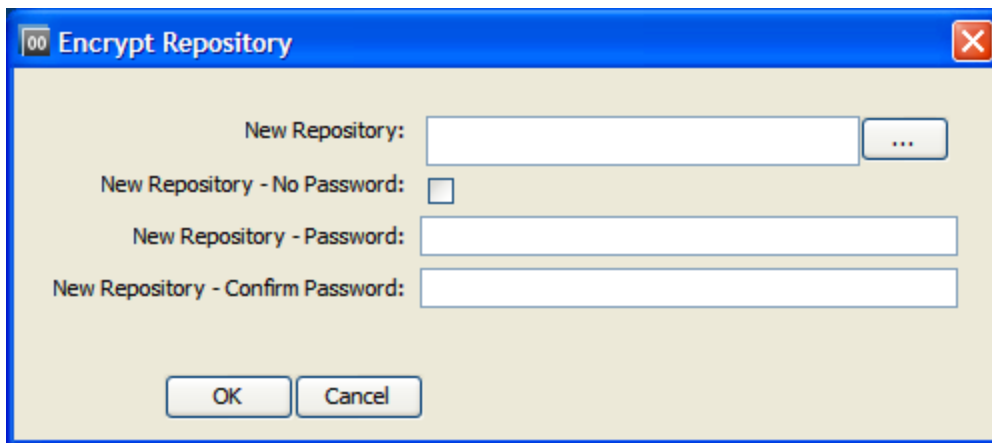


図 92 – リポジトリの暗号化

2. [New Repository] で、新しいリポジトリのコピーのパスと名前を入力します。  
もしくは  
[...] ボタンをクリックして目的の場所またはリポジトリに移動し、暗号化されたコピーの名前を付けます。
3. 暗号化されたコピーのパスワードの入力と確認を行い、[OK] をクリックします。

## 暗号化されたリポジトリを開く

### 暗号化されたリポジトリを開くには

1. 暗号化されたリポジトリのコピーを開くには、[Repository] メニューから、[Open Repository] をクリックして該当の暗号化されたリポジトリを選択します。  
もしくは  
暗号化されたコピーを初めて開く場合、リポジトリを追加します。[Repository] メニューで、[Add Repository] をクリックし、暗号化されたコピーに移動します。
2. パスワードを求められたら、リポジトリを暗号化したときに指定したパスワードを入力します。

## リポジトリの復号化

### リポジトリを復号化するには

1. 暗号化されたリポジトリを開きます。
2. [Repository] メニューから、[Decrypt Repository] をクリックします。  
異なるパスワードを持つ、2つ目の暗号化されたリポジトリのコピーを作成するとします。これを行うには、リポジトリを再暗号化します。

## 2つ目の暗号化されたリポジトリのコピーの作成

### 2つ目の暗号化されたリポジトリのコピーを作成するには

1. 暗号化されたリポジトリを、追加して（必要な場合）、開きます。
2. [Repository] メニューから、[Re-encrypt Repository] をクリックします。

[Re-encrypt Repository] ダイアログボックスが表示されます。

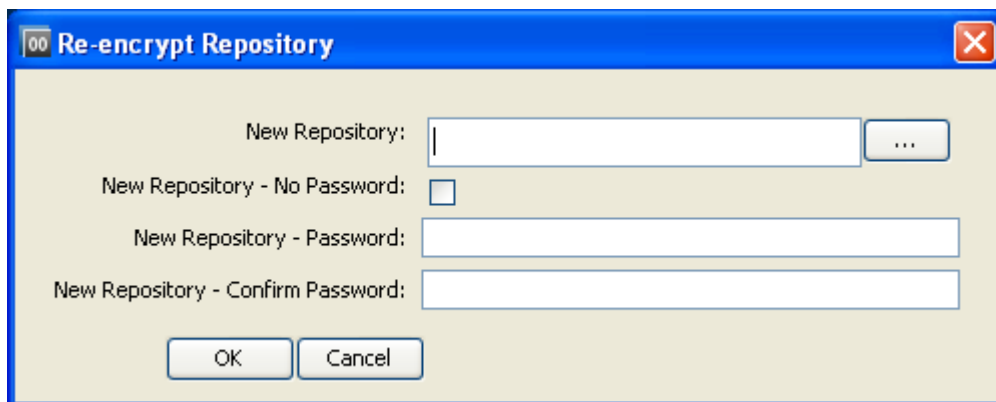


図 93 - リポジトリの再暗号化

3. [New Repository] で、新しいリポジトリのコピーのパスと名前を入力します。  
もしくは  
[...] ボタンをクリックして目的の場所またはリポジトリに移動し、暗号化されたコピーの名前を付けます。
4. 暗号化されたコピーのパスワードの入力と確認を行い、[OK] をクリックします。

## リポジトリのバックアップと復元

パブリックリポジトリ（Central が使用するリポジトリ）と Studio のデフォルトのローカルリポジトリはそれぞれ、Studio の起動時とオブジェクトの保存時に、自動的に .jar ファイルにバックアップされます。デフォルトでは、10 個のバックアップ .jar ファイルが保存されます。11 個目が保存されると、最も古いバックアップファイルが削除されます。ただし、プライベートリポジトリを手動でバックアップすることもできます。

### リポジトリをバックアップするには

1. バックアップするリポジトリを開きます。
2. [Repository] メニューから、[Create Backup] を選択します。

### リポジトリを復元するには

1. [Repository] メニューで、[Import Repository] を選択します。
2. [Select file for backup] ダイアログボックスで、リポジトリを定義する .jar ファイルを含むフォルダーに移動します。  
jar ファイルを含むフォルダーには、2 つのサブフォルダー（**backups** と **data**）があります。
3. [Open] をクリックします。  
Central のパブリックリポジトリについて、インポートするオブジェクトとのバージョンの競合を確認した後、OO がリポジトリをインポートします。
4. 競合がある場合は、リポジトリをインポートするときと同様に、競合を解決します。  
リポジトリのインポートの詳細については、「[リポジトリのインポート](#)」を参照してください。

## Web サービスからのオペレーションの作成

Web Services Wizard は、ウィザードで指定した Web サービスの Web サービス定義言語 (WSDL) の API に基づいて、OO オペレーションを作成します (Web サービスは、インターネット上に存在するビジネスロジックの 1 つです)。

Web Services Wizard を実行する場合、特定の Web サービスに対する WSDL を指定します。ポインターとして指定する WSDL 文字列は、ファイルの場所と名前、または URL になります。

たとえば、Web サービスと API を通じてチケットを作成する AlertAlert と呼ばれるアプリケーションがあり、AlertAlert でチケットを作成するとします。Web Services Wizard は、そのアプリケーションで実行できる操作 (チケットの作成や変更など) について、アプリケーションの API を Web サービスの WSDL から抽出します。WSDL は、Web サービスのメソッド、各メソッドが必要とする入力、各入力に必要な形式を定義します。

ウィザードに WSDL (例では AlertAlert) を指定して実行すると、Web サービスで実行できるオペレーションが生成されます。Studio の Library に、必要な入力で作成された状態でオペレーションが表示されます。このオペレーションはフローで使用できます。これらのオペレーションの 1 つを使用するフローを実行するには、Studio で、Web サービスを指定するリモートアクションサービス (RAS) 参照を作成する必要があります。

### Web Services Wizard で Web サービス用のオペレーションを作成するには

1. OO のホームディレクトリの **¥Studio¥tools¥** フォルダで、**wswizard.exe** をダブルクリックします。Web Services Wizard が起動します。

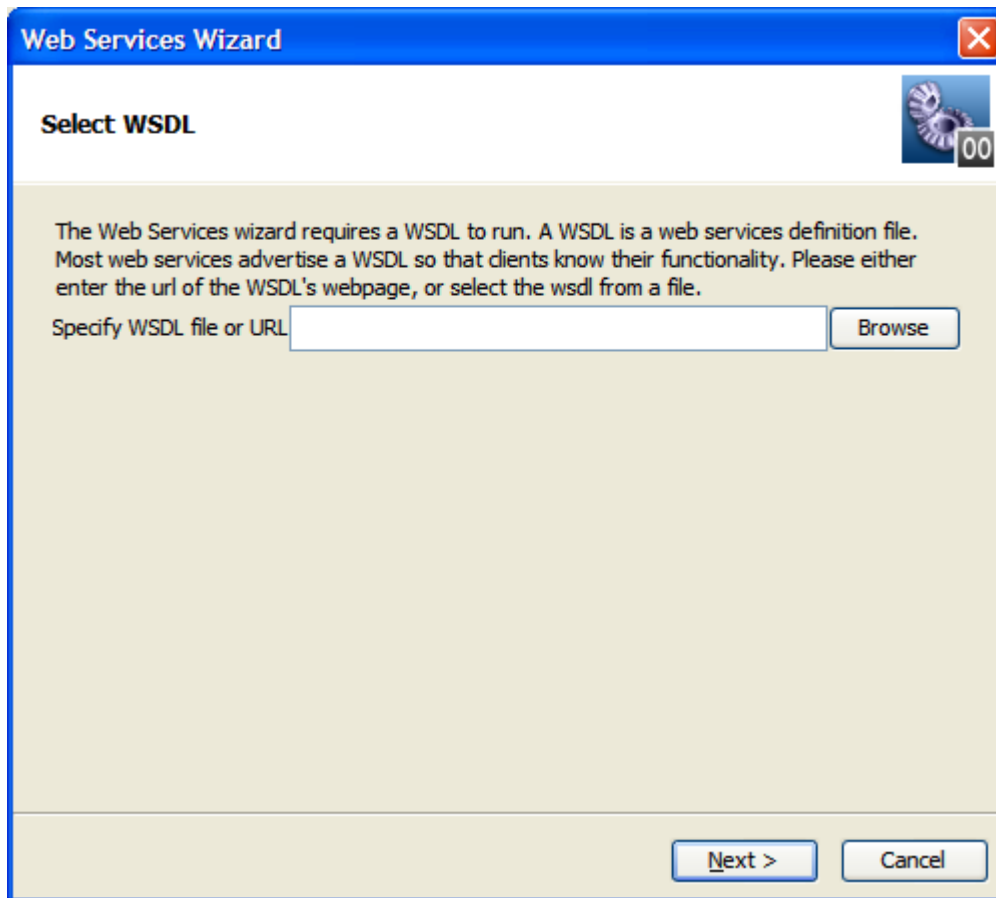


図 94 — Web Services Wizard

- [Specify WSDL file or URL] ボックスで、WSDL の URL またはファイルの場所を入力します。  
もしくは  
[Browse] をクリックし、ファイルを保存する場所に移動します。  
WSDL の場所は、Studio で作成した RAS 参照の URL にできます。これは、ポインターの参照する RAS が Web サービスであるためです。
- [Next] をクリックします。  
ウィザードが WSDL をロードする間、進捗状況ボックスが表示されます。  
[Choose Operations to generate] ペインが表示されます。

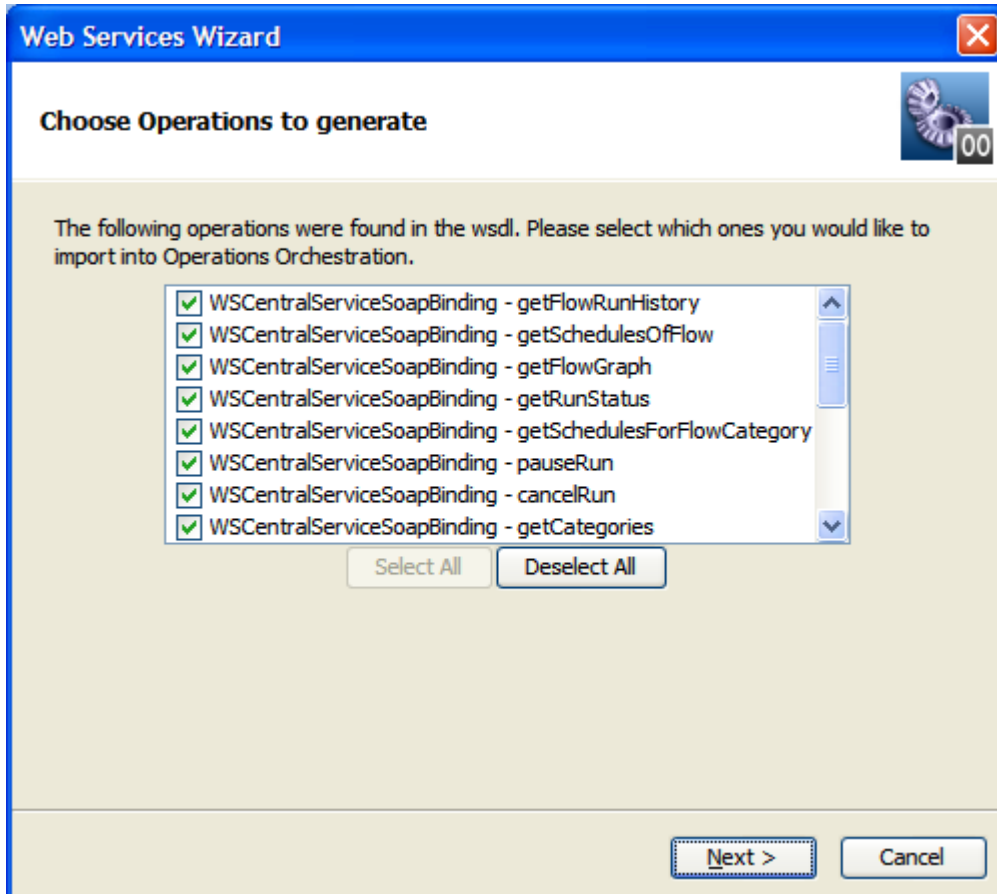


図 95 - 作成するオペレーションの選択

WSDL のメソッドがすべて表示されます。これらはオペレーションに変換できます。

- オペレーションに変換しないメソッドを選択解除し（チェックボックスからチェックを外す）、[Next] をクリックします。  
[Select Repository] ページが開くので、新しいオペレーションを格納するリポジトリの作成先フォルダーを指定します。現在は空であるフォルダーを指定することをお勧めします。
- [Enter the repository to open] で、ウィザードで作成するオペレーションを格納するリポジトリの作成先のパスとフォルダーを入力し、[Next] をクリックします。  
もしくは  
[Browse] をクリックし、リポジトリを作成する場所に移動して、[Next] をクリックします。  
ウィザードによりオペレーションが作成され、指定したリポジトリに格納されます。

- さらにオペレーションを作成するには、[Next] をクリックし、この手順のここまでの操作を繰り返します。  
もしくは  
このメソッドによる新しいオペレーションの作成が終了したら、[Finish] をクリックします。
- Studio を開き、作成したリポジトリをインポートします。

## リモートアクションサービスを使用した Central 外での運用

リモートアクションサービス (RAS、ラスと発音) は、Java、.NET の両方と相互作用して、コマンドを使用してフローをマシン上で実行したり、本来 OO のフローにアクセスできない API に対してフローを実行したりできるようにするサービスのインスタンスです。

たとえば、RAS を使用すると次のことが可能になります。

- Exchange API との相互作用。
- Central がインストールされているものとは別のドメインでのサーバーの再起動。
- Central が Linux サーバーにインストールされている場合の、Windows サーバーでのオペレーションの実行。

RAS の使用目的により、Central のインストール時にデフォルトで含まれている RAS を使用できることもあれば、スタンドアロンの RAS をリモートでインストールする必要がある場合もあります。

RAS を使用して、Central が別のドメインにアクセスできるようにする場合、必要なのは RAS をそのドメインにインストールすることだけです。

RAS を使用して、Central が別の API にアクセスできるようにする場合、RAS は、その API へのアクセスを提供する Web サービスにアクセスするか、または、その API を使用するオペレーションを RAS に追加することによって、これを実現します。

RAS を使用して、Central が本来アクセスできないコマンドやマシンにアクセスできるようにする場合、次のことが必要です。

- RAS は、特定のコードまたはコードへのアクセスを持ち、さらにそのコードで OO のフローを実行できるようにする必要があります。

RAS は、次の 2 つのいずれかの方法で、コードの入手またはコードへのアクセスが可能になります。

- RAS が Web サービスと相互作用する場合: RAS はすでに、任意の Web サービスと相互作用するために必要なコード (IAction) を持っています。  
RAS は Web サービスと相互作用できますが、Central では、フロー作成者がフローを作成、使用できるようにするためには、Web サービスで必要となるコマンドや入力が必要です。そのため、OO では Web Service Wizard を用意し、作成者が目的の Web サービスからオペレーションを作成できるようにしています。Web Service Wizard の使用方法の詳細については、「[Web サービスからのオペレーションの作成](#)」を参照してください。
- RAS が Web サービスの仲介なしに API と相互作用する場合、オペレーションの作成者は API のコマンドとの相互作用に必要なコードを含む IAction を作成する必要があります。  
IAction を作成するには、Java または .NET でサポートされるプログラミング言語を使用します。このレベルの作成には、Windows Scripting Host または Perl でのスクリプト作成だけでなく、.NET でのプログラミング能力も必要です。

IAction を作成した後、開発者は IAction を Java アーカイブ (.jar) ファイルまたは動的リンクライブラリ (.dll) にコンパイルして、RAS の Repository フォルダに追加します。IAction を作成して RAS に追加する方法の詳細については、『*HP OO Software Development Kit Guide*』 (SDKGuide.pdf) を参照してください。

Web サービスの場合と同様に、Central では、IAction を使用するフローを作成するには、IAction のコマンドと入力が必要です。必要なコマンドを Central に提供するために、OO では、作成者が RAS からオペレーションをインポートできます。RAS からオペレーションをインポートする方法については、「[RAS からのオペレーションの作成](#)」を参照してください。

- Central は RAS に方向付けられている必要があります。

Central を RAS に方向付けるには、Studio で RAS 参照を作成し、作成した RAS 参照を RAS に対応したオペレーションにドラッグします。RAS 参照には、名前と RAS にアクセスするための URL が含まれています。Studio のデフォルトの RAS 参照は、**RAS\_Operator\_Path** です。

注: すべてのオペレーションに RAS 参照を追加できるわけではありません。RAS に対応したオペレーション ([Inputs] タブに RAS 参照を追加するためのフィールドを持つ) には、次のものが含まれます。

- RAS または Web サービスから作成されたオペレーション
  - Studio で作成した、RAS オペレーションタイプを持つ新しいオペレーション
  - RAS 上で実行する必要がある、デフォルトの OO コンテンツのオペレーション
- デフォルトの OO コンテンツのオペレーションの多くには、RAS に対応した同等のオペレーションがあります (Studio のリポジトリペインの **Operations**  $\forall$  **Remote Command Execution** フォルダ内のオペレーションなど)。

## このセクションのトピック

[Central での RAS に依存するオペレーションの実行方法](#)

[RAS を使用できるかどうかの確認](#)

[既存の RAS の追加](#)

[RAS 参照の追加](#)

[既存の RAS 参照の再設定](#)

[RAS 参照の変更](#)

[Web サービスにアクセスするオペレーションの作成](#)

[RAS に依存するオペレーションの実行のトラブルシューティング](#)

[RAS からのオペレーションの作成](#)

[RAS 参照の削除](#)

[RAS コンテンツのインポート](#)

## Central での RAS に依存するオペレーションの実行方法

RAS は、動的リンクライブラリ（DLL または .dll）と、オペレーションのクラスが定義されている Java アーカイブ（.jar）から構成されています。RAS を呼び出すと、必要なオペレーションのクラスのインスタンスが実行されます。

RAS に依存するオペレーションには、オペレーションを実行する必要がある対象の RAS を特定するために Central が必要とする情報が含まれています。

- **Action Class** は、アーカイブまたは DLL 内にあるオペレーションの名前で、実行するオペレーションです。
- **Archive** は、オペレーションを含む .dll または .jar（Dotwebactions.dll など）です。
- **RAS** は、オペレーションが使用する RAS 参照です。

これらの情報は、オペレーションの [Properties] シートにある [RAS Operation] フィールドで指定します。Library の **Integrations**、**Operations**、**Utility Operations** の各フォルダーにあるオペレーションでは、すでにこの情報が指定されています。RAS の URL を変更する場合は、Studio の RAS 参照の URL も変更する必要があります。RAS に依存するオペレーションの新規作成には、次の作業が含まれます。

- オペレーションの定義を持つ .dll または .jar を作成する。
- オペレーションを実行する RAS サービスのインストールで、その .dll または .jar を指定する。
- オペレーションの [RAS Operation] フィールドで、オペレーションのクラス、.dll または .jar、RAS ポインターの名前を指定する。

RAS オペレーションの [Properties] シートにある [RAS Operation] フィールドで値を指定する方法については、「[RAS オペレーション: RAS のための IAction プログラミング](#)」を参照してください。

## RAS を使用できるかどうかの確認

リモートアクションサービスの概要については、「[リモートアクションサービスを使用した Central 外での運用](#)」を参照してください。

既存の RAS を使用できるかどうかを確認するには

1. **Library** で、**Configuration**  $\gg$  **Remote Action Services** フォルダーを右クリックして [Properties] をクリックします。
2. [Remote Action Services] シートで、使用できるかどうかを確認する RAS を強調表示し、[Check Availability] をクリックします。

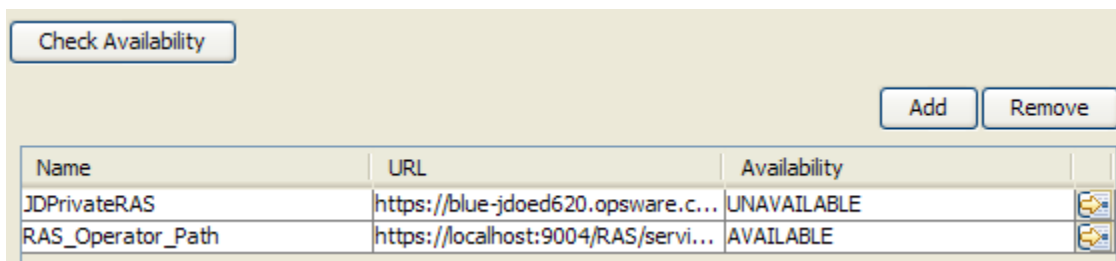


図 96 — [Remote Action Services] シート

RAS を使用できるかどうか [Availability] 列にレポートされます。

3. RAS を使用できない場合、RAS ポインターの行の末尾にある右向き矢印をクリックして、[URL] ボックスで RAS の場所を修正します。



## 既存の RAS の追加

[Configuration ¥ Remote Action Services] シートを開くと、ネットワークにインストールされている RAS が [Discovered RAS] に一覧表示されます。OO 外での実行のために RAS を必要とするオペレーションでこれらの RAS を使用できるようにするには、次の手順を使用します。

リモートアクションサービスの概要については、「[リモートアクションサービスを使用した Central 外での運用](#)」を参照してください。

### 既存の RAS を追加するには

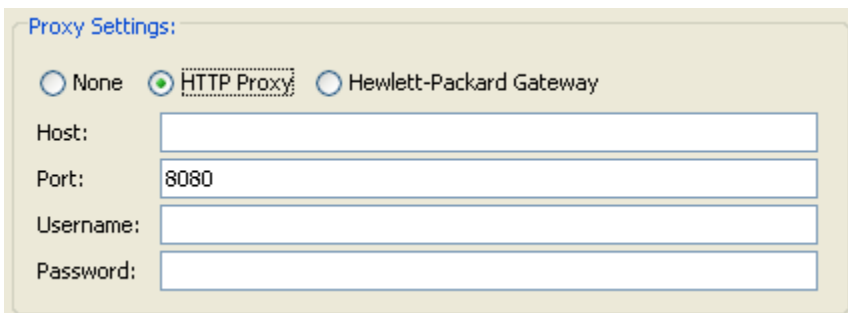
1. [Configuration ¥ Remote Action Services] シートを開きます。
2. [Discovered RAS] で、オペレーションで使用できるようにする RAS を選択し、[Discovered RAS] の下にある [Add] ボタンをクリックします。
3. 表示されたダイアログボックスで、新しい RAS に名前を付けます。  
新しい RAS が、[Configuration ¥ Remote Action Services] シートの RAS のリストに表示されます。

## RAS 参照の追加

リモートアクションサービスの概要については、「[リモートアクションサービスを使用した Central 外での運用](#)」を参照してください。

### 新しい RAS 参照を追加するには

1. **Library** で、**Configuration ¥ Remote Action Services** フォルダを右クリックして [Properties] をクリックします。  
作成ペインで [Remote Action Services] シートを開き、上部にある [Add] ボタンをクリックします。  
もしくは  
[Remote Action Services] シートがすでに開いている場合は、上部にある [Add] ボタンをクリックします。
2. 表示されたボックスで、新しい RAS 参照の名前を入力して [OK] をクリックします。  
新しい RAS の RAS エディターが開きます。
3. [URL] テキストボックスで、次の構文を使用して URL を入力します。  
`http://<ホスト名>:<ポート>/RAS/services/RCAgentService`  
RAS インストールプログラムでデフォルトのポート番号を受け入れた場合は、ポート番号は 4085 です。  
Central から見てファイアウォールの反対側にある RAS に接続するには、RAS と通信するためにプロキシを指定する必要がある場合があります。
4. プロキシを選択するには、[Proxy Settings] で、[HTTP Proxy] または [Hewlett-Packard Gateway] のいずれかを選択します。
  - [HTTP Proxy] を選択した場合は、次のテキストボックスに入力します。



Proxy Settings:

None  HTTP Proxy  Hewlett-Packard Gateway

Host:

Port:

Username:

Password:

図 97 — HTTP Proxy の設定

- [Host] テキストボックスで、プロキシが配置されているサーバーのマシン名または IP アドレスを入力します。
- [Port] テキストボックスで、プロキシが使用するポートを入力します。
- [Username] と [Password] テキストボックスで、プロキシを使用するために必要な資格情報を入力します。
- [Hewlett-Packard Gateway] を選択した場合は、次のテキストボックスに入力します。

Proxy Settings:

None
  HTTP Proxy
  Hewlett-Packard Gateway

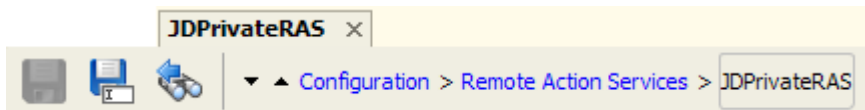
Host:

Port:

Realm:

図 98 – HP Gateway のプロキシ設定

- [Host] テキストボックスで、プロキシが配置されているサーバーのマシン名または IP アドレスを入力します。
  - [Port] テキストボックスで、プロキシが使用するポートを入力します。
  - [Realm] テキストボックスで、ゲートウェイがメンバーになっている Hewlett-Packard Gateway のレルムの名前を入力します。
5. 新しい RAS を保存し、[Configuration] の隣の [Remote Action Services] をクリックします。



6. [Remote Action Services] シートで、[Check Availability] をクリックします。  
これによりサービスの URL を使用できるかどうかチェックされ、[Availability] 列に使用可否が表示されます。

## 既存の RAS 参照の再設定

リモートアクションサービスの概要については、「[リモートアクションサービスを使用した Central 外での運用](#)」を参照してください。

### 既存の RAS 参照を再設定するには

1. **Library** で、**Configuration** > **Remote Action Services** フォルダを開き、設定する RAS をダブルクリックするか、または対象の RAS を右クリックして [Open] をクリックします。

作成ペインで、選択した RAS の RAS エディターが開きます。

もしくは

[Remote Action Services] シートが開いた状態で、再設定する RAS 参照を選択し、表のそのポインタの行の末尾にある右向き矢印 (→) をクリックします。

2. [URL] テキストボックスでは、指定する URL に次の構文を使用する必要があります。

`http://<ホスト名>:<ポート>/RAS/services/RCAgentService`

RAS インストールプログラムでデフォルトのポート番号を受け入れた場合は、ポート番号は 4085 です。

Central から見てファイアウォールの反対側にある RAS に接続するには、RAS と通信するためにプロキシを指定する必要がある場合があります。

3. プロキシを選択するには、[Proxy Settings] で、[HTTP Proxy] または [Hewlett-Packard Gateway] のいずれかを選択します。

- [HTTP Proxy] を選択した場合は、次のテキストボックスに入力します。

Proxy Settings:

None
  HTTP Proxy
  Hewlett-Packard Gateway

Host:

Port:

Username:

Password:

図 99 – HTTP Proxy の設定

- [Host] テキストボックスで、プロキシが配置されているサーバーのマシン名または IP アドレスを入力します。
- [Port] テキストボックスで、プロキシが使用するポートを入力します。
- [Username] と [Password] テキストボックスで、プロキシを使用するために必要な資格情報を入力します。
- [Hewlett-Packard Gateway] を選択した場合は、次のテキストボックスに入力します。

Proxy Settings:

None
  HTTP Proxy
  Hewlett-Packard Gateway

Host:

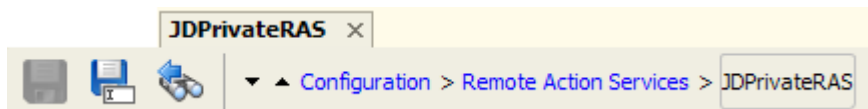
Port:

Realm:

図 100 – HP Gateway のプロキシ設定

- [Host] テキストボックスで、プロキシが配置されているサーバーのマシン名または IP アドレスを入力します。
- [Port] テキストボックスで、プロキシが使用するポートを入力します。
- [Realm] テキストボックスで、ゲートウェイがメンバーになっている Hewlett-Packard Gateway のレルムの名前を入力します。

4. RAS を保存し、[Configuration] の隣の [Remote Action Services] をクリックします。



5. [Remote Action Services] シートで、[Check Availability] をクリックします。  
これによりサービスの URL を使用できるかどうかチェックされ、[Availability] 列に使用可否が表示されます。

## RAS 参照の変更

IAction から作成されたオペレーションは、RAS 参照を持っています。RAS 参照の作成は IAction の作成の一部であるためです。オペレーションの [Properties] シートの RAS 参照を指定する部分は、次のようになっています。


RAS:

Override RAS:

図 101 – RAS 参照のフィールド

詳細については、「[RAS オペレーション: RAS のための IAction プログラミング](#)」を参照してください。

OO コンテンツフォルダー (**Integrations**、**Operations**、**Utility Operations** の各フォルダー) で提供されているすべての IAction ベースのオペレーションには、[Override RAS] フィールドに、**overrideNRAS** または **overrideJRAS** フロー変数への参照が含まれています。RAS オペレーションを作成する場合、または IAction からオペレーションを作成する場合、[Override RAS] フィールドに、フロー変数参照 (**overrideJRAS**) (Java IAction の場合) または **overrideNRAS** (.NET IAction の場合) を含めることを強くお勧めします。

- RAS オペレーションを作成する場合、RAS 参照を変更するには、該当のオペレーションの [Properties] シートで、[Inputs] タブの [RAS] フィールドにある [Globe] アイコン () に、新しい RAS 参照を **Configuration Remote Action Services** フォルダーからドラッグします。オペレーションそのものの RAS 参照は、この方法でのみ変更できます。
- 既存の RAS オペレーションを含むフローで作業している場合は、RAS 参照を上書きできます。上書きの方法は、RAS 参照を変更する範囲によって異なります。
  - OO の特定のタイプの RAS オペレーション (つまり、NRAS 上書き、または JRAS 上書きを使用している) のすべてが同じ RAS を使用している場合、その RAS を参照するシステムプロパティを作成できます。




**鍵となる情報:** ただし、システムプロパティはグローバルコンテキストに追加されるフロー変数であることに注意してください。このため、システムプロパティから値を割り当てられる任意の入力について値を変更することはできません。**overrideJRAS** や **overrideNRAS** をシステムプロパティとして設定するということは、いずれかのシステムプロパティに割り当てられた RAS 参照が、そのグローバルフロー変数を使用するすべてのフローのすべての実行時に使用され、割り当て値を変更する機会がないことを意味します。システムプロパティは、フロー実行時に状況を修正する方法がないため、実行時にフローの中断や予期しない結果の作成を引き起こす可能性があります。

システムプロパティを作成する方法については、「[システムプロパティの作成](#)」を参照してください。

- 単独のフロー実行において、特定のタイプのすべての RAS オペレーションの RAS 参照を変更する場合は、RAS 上書きをフロー変数として設定し、フローまたはステップのいずれかの入力から値を割り当てます。
- 単独のフロー実行において、実行中の特定のポイント以降で、特定のタイプのすべての RAS オペレーション (つまり、NRAS 上書き、または JRAS 上書きを使用している) の RAS 参照を変更する場合は、RAS 上書きを次の値を持つフロー変数として設定します。
  - **Configuration Remote Action Services** フォルダーで作成した RAS 参照の名前
  - ステップ入力として割り当て
- 単独のフロー実行において、特定のステップのみでベースとなる RAS オペレーションの RAS 参照を変更する場合は、RAS 上書きをステップ入力に割り当て、フロー変数には割り当てないようにします。

### RAS オペレーションの RAS 参照を変更するには

- RAS オペレーションの [Properties] シートの [Inputs] タブで、RAS 参照を **Configuration Remote Action Services** フォルダーから [RAS] フィールドのラベルの隣にある [Globe] アイコン () にドラッグします。

### デフォルトの RAS 参照を別の RAS 参照で上書きするには

1. フローでフロー入力を作成するか、または、RAS 参照を含むオペレーションから作成されたステップでステップ入力を作成します。
2. その入力の値を、関連するオペレーションの [Properties] シートの [Inputs] タブの、[Override RAS] フィールドで参照されるフロー変数に割り当てるように指定します。

OO コンテンツで提供されるオペレーションでは、[Override RAS] フィールドで参照されるフロー変数は、**overrideJRAS** または **overrideNRAS** です。

3. [Override RAS] フィールドの参照が、前の手順で作成したフロー入力から値を取得するようにします。

## Web サービスにアクセスするオペレーションの作成

Web サービスとは、特定の一連の機能を実行する Web ベースのプログラミングロジックの一種です。フローを Web サービスと統合したり、フローを監視したり、フローの実行を検証したりする場合には、Web サービスの WSDL (Web サービス定義言語) ファイルに定義されているメソッドを介して Web サービスにアクセスできます。WSDL には、Web サービスのインターフェースが定義されています。Web サービスにアクセスするには、SOAP (簡易オブジェクトアクセスプロトコル) メッセージを作成して (XML で記述)、これを Web サービスに送信します。Web サービスが SOAP メッセージに応答し、XML を返します。

フローでの使用を目的として、WSDL には、フローのオペレーションが Web サービスと相互作用するときに行えるメソッドが定義されています。Web Service Wizard は、ウィザードで指定した Web サービスのインターフェースにあるメソッドのリストを表示するツールです。ウィザード内で、使用するメソッドを選択してクリックすると、選択した各メソッドについて、そのメソッドを実行できる OO オペレーションが作成されます。

または、Google 検索エンジンを使用して Web 検索を実行するオペレーションを作成する場合を考えます。この場合、Web Service Wizard で **GoogleSearch** Web サービスの WSDL を検索し、該当の WSDL に移動します。ウィザードに表示されたメソッドのリストから、**doGoogleSearch** を選択して [Generate XML] をクリックし、Google 検索を実行するオペレーションを定義する XML を作成します。

注: RAS に依存するオペレーションを作成するには、OO でも拡張機能を設定する必要があります。設定方法については、「*Administering Operations Orchestration Software*」 (AdminGuide.pdf) を参照してください。

Web Service Wizard の使用方法については、「[Web サービスからのオペレーションの作成](#)」を参照してください。

## RAS に依存するオペレーションの実行のトラブルシューティング

RAS に依存するオペレーションを実行しようとして Central が失敗する場合の理由は 3 つあります。

- RAS 参照が正しい RAS サービス (実行するオペレーションが定義されている .dll または .jar を含む RAS、またはターゲットマシンのドメインにインストールされている RAS) を指定していない。  
RAS オペレーション用の RAS 参照を追加、設定、置換する方法については、「[リモートアクションサービスを使用した Central 外での運用](#)」の関連するトピックを参照してください。
- RAS サービスがターゲットマシンにインストールされていない。
- RAS サービスにライブラリが存在しない。

## Studio の例: 任意の複数ドメインで実行するフロー

フロー変数を使用することにより、オペレーションをリモートで実行するための正しい RAS を選択するフローを作成できます。基本的には、目的の RAS 参照の名前をフロー変数に格納し、フローのロジックでそれがどの RAS 参照なのかを決定するようなフローを作成します。

たとえば、DomainA の ServerA または DomainB の ServerB のいずれかを再起動する場合を考えます。フローに次のようなステップを作成することにより、Central ユーザーは、フローを実行するときに、どのサーバーでフローを実行するかを決定できます。

1. ユーザーにドメイン名 / サーバー名を求める入力を持つステップを作成し、ドメイン名を取得します。



**ヒント: Display** オペレーション (Utility Operations フォルダーにある) は、ユーザープロンプトから取得したデータの一部をフィルタリングするだけのステップが必要な場合に便利です。

2. Display ステップで、ステップの結果を作成し、先ほど作成した入力からデータを与えます。



- ステップの結果として、ドメイン名を抽出するフィルターを作成し、抽出したドメイン名をフロー変数（たとえば `domainName`）に格納します。  
注: サーバー名も後で使用するためにフロー変数に割り当てる必要がありますが、ここではフローを実行するドメインに基づいて正しい RAS サービスを選択することに焦点を当てています。
- `domainName` の値をそのドメインで RAS サービスを介してフローを実行する RAS 参照の名前にマップする構造を作成し、マップされた RAS 参照の名前をフロー変数（たとえば、`rasName`）に格納します。  
これを行う場合に最も簡単なのは、List Map サブフローを使用する方法です。このサブフローでは、2つのリストを入力とします。片方のリストでは、使用する可能性のあるドメイン名を指定し、もう片方のリストでは、ユーザーが指定したドメイン名に従って選択する RAS 参照の名前を、最初のリストと同じ順序で指定します。
- フロー変数 `rasname`（およびサーバー名を格納したフロー変数）を、サーバーを再起動する RAS オペレーションの入力として与えます。

## RAS からのオペレーションの作成

RAS を使用するオペレーションを作成するには、次の方法があります。

- 独自の RAS とオペレーションを作成する。  
Web サービスとオペレーションをインポートした後、オペレーションを当初インポートされた場所から Library に移動します。オペレーションを元の場所に置いたままにして、その後、そのオペレーションを使用するフローをエクスポートすると、オペレーションはフローとともにエクスポートされません。エクスポートされたフローは、他のマシンでは動作しなくなります。この事態を避けるために、オペレーションを Library 内の別のフォルダーに移動します。
- リモートアクションサービス（OO によりデフォルトで提供されている「`RAS_Operator_Path`」という名前の RAS 参照により指定される）を使用する OO コンテンツをインポートして使用する。

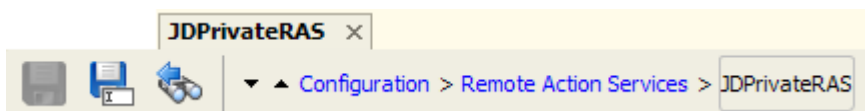
カスタムの Web サービスコンテンツをインポートするには、RAS 参照を作成し、作成した RAS 参照に目的の Web サービスコンテンツを含む RAS の URL を割り当てます。RAS を URL ではなく主として名前で指定すると、Studio で URL を変更することにより、その RAS を複数の場所で使用できます。

**Configuration** フォルダーには **Remote Action Services** フォルダーがあり、既存の RAS ポインターが格納されています。

これで、RAS から Web サービスコンテンツ（オペレーション、フロー）をインポートする準備ができました。

### 既存の検索済みの RAS を使用して RAS 参照を設定するには

- Library** で、**Configuration** > **Remote Action Services** フォルダーを右クリックして [Properties] をクリックします。  
作成ペインで [Remote Action Services] シートが開きます。
- [Discovered RAS] で、参照を作成する検索済みの RAS を選択し、下部にある [Add] ボタンをクリックします。
- 表示されたテキストボックスで、RAS の名前を入力します。  
新しい RAS の RAS エディターが開きます。
- [Configuration] の隣の [Remote Action Services] をクリックします。



- [Remote Action Services] シートで、[Check Availability] をクリックします。  
これによりサービスの URL を使用できるかどうかチェックされ、[Availability] 列に使用可否が表示されます。

## このセクションのトピック

[RAS からのオペレーションの作成](#)

[デフォルトの RAS がないオペレーションの追加](#)

## RAS からのオペレーションの作成

リモートアクションサービスの概要については、「[リモートアクションサービスを使用した Central 外での運用](#)」を参照してください。

### RAS からオペレーションを作成するには

1. Studio で、Web サービスからオペレーションを作成するフォルダーを作成するか、または強調表示します。
2. [File] メニューから、[Create Operations from RAS] をポイントします。  
何らかのエディターまたは [Properties] シートが開いている場合は、それらを閉じるように求められます。  
それらを閉じた後、[RAS Import] ダイアログボックスが表示されます。
3. ドロップダウンリストボックスで、目的のオペレーションを含む RAS を選択し、[OK] をクリックします。  
進捗状況ボックスがオペレーションの作成を監視します。すべてのオペレーションが作成されると、RAS オペレーションを含むフォルダーが、手順 2 で選択したフォルダーに表示されます。  
このフォルダーにそれらのオペレーションがすでに作成されている場合は、すでにフォルダーに存在するオペレーションを上書きするかどうかを確認するよう求められます。

## デフォルトの RAS がないオペレーションの追加

Java アーカイブ (.jar) または動的リンクライブラリ (.dll) ファイルに含まれている IAction を RAS に追加することにより、RAS に依存するオペレーションのベースを作成できます。その後、OO のフローで使用できる OO オペレーションを作成するために、その RAS からオペレーションを作成します。

リモートアクションサービスの概要、およびリモートアクションサービスと Central リポジトリに含まれていない、または Central リポジトリからアクセスできないアクションとの関係の概要については、「[リモートアクションサービスを使用した Central 外での運用](#)」を参照してください。

### デフォルトの RAS に含まれていないオペレーションを追加するには

1. オペレーションで実行可能にするタスクを実行する IAction を作成し、作成した IAction を動的リンクライブラリ (.dll) または .jar ファイルに格納します。  
もしくは  
目的のオペレーションを含む、サードパーティ製の .dll または .jar を入手します。
2. .dll または .jar を RAS に追加するには、OO のホームディレクトリにある次のいずれかの適切な場所にコピーします。
  - RAS に追加する .dll (独自またはサードパーティ製にかかわらず、次の場所へ)、または .jar の場合：  
`¥RAS¥Java¥Default¥repository¥`
  - RAS に追加するサードパーティ製 .jar の場合：  
`¥RAS¥Java¥Default¥webapp¥WEB_INF¥lib¥`

注: RAS アクションはすべて IAction インターフェースを実装する必要があるため、サードパーティ製のライブラリを動作させるには、そのライブラリを含む IAction クラスを作成しなければならない場合があります。



3. リモートまたは拡張 RAS オペレーションが正しく機能することを確認するには、次の手順で行います。
  - 必要な RAS を Studio でインストール、設定した後、その RAS が Studio で使用できることを必ず確認します。  
RAS を使用できるかどうかを確認する方法については、「[RAS を使用できるかどうかの確認](#)」を参照してください。
  - 使用する RAS でオペレーションがサポートされていることを確認する必要がある場合もあります。確認を行うには、RAS のオペレーションをインポートして、オペレーションを検証します。「[RAS からのオペレーションの作成](#)」を参照してください。

注: デフォルトでは、IAction から作成するオペレーションのレスポンスは、IAction の戻りコードのみに基づいています。ただし、オペレーションを作成した後で、IAction のその他の出力をテストするレスポンスルールを作成できます。

IAction の作成方法については、『[SDK Guide](#)』（[SDKGuide.pdf](#)）の IAction の作成の項を参照してください。

## RAS 参照の削除

リモートアクションサービスの概要については、「[リモートアクションサービスを使用した Central 外での運用](#)」を参照してください。

### RAS 参照を削除するには

- リモートアクションサービスのエディターで、削除する RAS 参照を選択し、[Remove] をクリックします。

## RAS コンテンツのインポート

リモートアクションサービスの概要については、「[リモートアクションサービスを使用した Central 外での運用](#)」を参照してください。

### RAS コンテンツをインポートするには

1. 対象のコンテンツを含む RAS をインストールします。  
RAS を単独でインストールする方法については、「[Installing Operations Orchestration Software](#)」（[InstallGuide.pdf](#)）を参照してください。
2. Studio を開きます。
3. RAS コンテンツをインポートするには、コンテンツをインポートするフォルダーを選択し、[File] メニューから [Create Operations from RAS] を選択して、このセクションで前述した「フローのインポート」で説明されているように、インポートプロセスを完了します。
4. 使用する RAS でオペレーションがサポートされていることを確認するには、検証が必要になります。  
注: HP OO とともにインストールされる RAS (RAS\_Operator\_Path) には、Accelerator Pack やデフォルトのライブラリコンテンツで使用されるすべてのオペレーションが含まれています。

# データが正しい形式かどうかの評価

エバリュエーターは、システムアカウント以外のデータソースの入力を検証するために使用する文字列形式です。エバリュエーターは、次のようなデータ割り当てから入力に取得したデータを検証する場合に特に有効です。

- ユーザープロンプト
- 定数

定数値にフロー変数（ $\{\text{domain}\}$  など）が含まれている場合、エバリュエーターは、定数値に格納される変数の値（変数参照ではなく）を使用して定数値に適用されます。

- **[Previous Step's Result]**

Studio では、次の値を検証するシステムエバリュエーターをデフォルトで用意しています。

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

デフォルトの電話番号用データエバリュエーターは、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-nnn-nnnn）のみをサポートします。その他の地域の電話番号形式を検証するには、別のエバリュエーターが必要です。新しいシステムエバリュエーターを作成する方法については、「[エバリュエーターの作成](#)」を参照してください。

- UUID

エバリュエーターでは、=、!=、Begins with、Contains、Match All Words、Match At Least One Word などの標準のエバリュエーターを使用することも、次のような評価ツールを使用することもできます。


- 正規表現  
正規表現の作成方法については、「[正規表現での作業](#)」を参照してください。
- スクリプトレット
- 共有エバリュエーターへの参照

## エバリュエーターの作成

注: デフォルトの **Telephone Number** システムエバリュエーターは、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-nnn-nnnn）を検証します。地域の電話番号形式を検証するエバリュエーターを作成するには、正規表現を使用する必要があります。正規表現の詳細については、「[正規表現での作業](#)」を参照してください。

### エバリュエーターを作成するには

1. Library で、**Configuration**  $\forall$  **System Evaluators** フォルダーを右クリックし、[New] をクリックします。  
もしくは  
[Properties] をクリックし、表示された [System Evaluators] シートで [Add] をクリックします。
2. 表示された [Select Evaluator] ドロップダウンリストで、エバリュエーターまたは評価ツールを選択して [OK] をクリックします。
3. 次のボックスで、エバリュエーターに名前を付け、[OK] をクリックします。

- [New] をクリックした場合は、新しいエバリュエーター用にシステムエバリュエーターのエディターが開きます。
- [Properties] をクリックした場合は、[System Evaluators] シートのエバリュエーターのリストに新しいエバリュエーターが表示されます。該当の行の右端にある右向き矢印 () をクリックして、システムエバリュエーターのエディターを開きます。

システムエバリュエーターのエディターの外観は、選択したエバリュエーターの種類によって異なります。

4. エバリュエーターが入力のテスト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。
5. [Test Filter Input] ボックスの内容をクリアするには、[Clear] をクリックします。
6. エバリュエーターのテストを、次のいずれかの方法で行います。
  - 検証するテキストの例を [Test Filter Input] ボックスに入力する。
  - [Quick Command] をクリックして、表示された [Input] ボックスで検証またはテストする出力を取得するためのコマンドと引数を入力し、[OK] をクリックする。
7. 作業結果を保存します。


たとえば、URL 形式の入力を検証するエバリュエーターを作成するには、エバリュエーター [Begins With] を選択して、次に [Test] テキストボックスに [http://www] と入力します。または、URL の最初の部分を正規表現で記述することもできます。

## エバリュエーターの編集

### エバリュエーターを変更するには

1. Library で、**Configuration**  $\forall$  **System Evaluators** フォルダを開きます。
2. 変更するエバリュエーターを右クリックし、[Open] をクリックします。
3. エバリュエーターを作成する場合と同様の手順で変更を行います。
4. 作業結果を保存します。

### エディターを閉じずに別のエバリュエーターを編集するには

- [Configuration] の隣にある上向き矢印または下向き矢印 () をクリックします。

## エバリュエーターの削除

### エバリュエーターを削除するには

1. [System Evaluators] シートを開きます。
2. エバリュエーターのリストで、削除するエバリュエーターを選択して [Remove] をクリックします。
3. 作業結果を保存します。

## ダッシュボードグラフでのレポート作成用の値の記録

Central のダッシュボードグラフで使用するためのフローからのデータの記録は、フロー内の単一ステップの入力で行います。そのステップに、記録するデータを格納しているフロー変数からデータを取得する入力を作成します。1つのステップですべての値を記録しなければならないため、このステップはレポート作成を行う値の1つが生成された最後のステップの後ろに置く必要があります。通常は、**success** リターンステップです。

入力の値を記録するにはドメインタームを指定します。Central のダッシュボードの棒グラフでは、これにより、このフローの入力値を記録するために必要な情報を取得します。ドメインタームは棒グラフの縦軸または横軸のいずれかになり、入力値がこの軸にグラフ化されます。たとえば、横軸にフローの名前を、縦軸にフローを実行したサーバーを表すグラフが考えられます。

レポート作成のために関連させるデータはすべて、単一ステップの入力値である必要があります。したがって、そのステップは、レポート作成用に記録して関連させるデータを取得する最後のステップの後ろに置く必要があります。つまり、多くの場合、ダッシュボードでのレポート作成に必要な入力すべてが定義され、記録されるのは、リターンステップや問題の修正方法を含むステップになります。値の記録のみを行うそれらのステップで、入力を作成できます。

例として、IT マネージャーが Central のダッシュボードグラフで、**Restart Windows Server** フローが実行されたサーバーとそのアクションの内容を表示する場合を考えます。**Ping** または **TraceRoute** のいずれかのステップで、Central のダッシュボードグラフの1つで使用するドメインタームとして、ホスト入力を記録できます。そうしたドメインタームは**構成アイテム**になり、アプリケーションやハードウェアコンポーネント（サーバーなど）の名前や IP アドレスを取得します。IT マネージャーが **Alerts Per Configuration Item** グラフを選択すると、フローが実行されたサーバーの IP アドレスやドメイン名を表示できます。ダッシュボードに **Flows Per Configuration Item** グラフを追加すると、それらのサーバーでどのフローが実行されたかをすぐに表示できます。さらに、特定のサーバーで実行された **Restart Windows Server** フローを表す棒グラフをダブルクリックすると、フローの結果を表示できます。

レポート作成はほかのさまざまな用途にも使うことができます。一例として、サーバーの再起動などターゲットサーバーを操作するステップがある場合、割り当てた入力を**アクション**として記録できます。Central で特定のサーバーの再起動回数を確認するには、その後に **All CI's Organized by Action** グラフを表示します。

次に、この情報をレポート作成用に記録する方法を説明します。**構成アイテム**をドメインタームに指定して入力を記録してみましょう。

Central では、次にグラフを追加して、このドメインタームをグラフ化できます。ダッシュボードグラフの表示と作成については、Central のヘルプを参照してください。

### ダッシュボードのレポート作成用に入力を記録するには

- 記録する入力の [Inputs] タブを開き、[Record Under] を選択して、ドロップダウンリストから記録する入力のドメインタームを選びます。

このフローの入力値は Central にレポートされます。どのダッシュボードグラフでも、選択したドメインターム用の記録データが軸に表示される場合は、この入力値がその軸に表示されます。

## ダッシュボードグラフ用のドメインターム

ドメインタームは、フローと入力に割り当てることができる属性です。通常、フローで取得する特定のデータと関連付けられます。そのようなデータの例として、フローが実行される IT コンポーネントの種類および特定の IT コンポーネント、フローの実行を要求するトラブルチケット、フローが実行する操作などがあげられます。一度こういった特定のデータとドメインタームを関連付けると、IT インフラの診断ツールとしての Central の機能が強化されます。

ある IP アドレスをフローが実行されるサーバーの IP アドレスとして認識するフローを実行して、ドメインタームである**構成アイテム**をその入力に割り当てるとします。Central でフローが実行された後、Central ユーザーは、**構成アイテム**に対するほかの要因をレポートするグラフを確認できます。表示されるグラフには、**Flows per Configuration Item for the Last 7 Days** などがあります。フローが 192.118.55.109 に対して実行された場合、グラフに「192.118.55.109」とラベル付けされた行ができます。また、このサーバーで実行されたフローが中断されたことを示す障害も表示されます。

Central の [Dashboard] タブではグラフが表示されます。グラフは、ドメインタームと入力値のさまざまな組み合わせを表示したものです。これらのデータと関連付けて表面分析を行い、IT インフラの次のような傾向を提示します。

- フローの全般的な使用と結果
- 発生したアラートまたはイベントの種類と頻度
- 影響を受けた構成アイテム（サーバーやアプリケーションなど）

ドメインタームにはデフォルトで値が入っているものもあれば、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もあります。

ダッシュボードグラフへのデータレポート作成に使えるドメインタームは次のとおりです（ドメインタームの追加方法や、ドメインタームへの特定値の追加方法については、「[ドメインタームまたはドメインターム値の追加](#)」を参照してください）。

- **アクション**  
フローが実行した操作です。これらの値をリストに追加し、このドメインタームの関連入力値として入力します。
- **アラート**  
フローと関連付けられたアラートを識別するものです。インシデントの関連値はフロー入力で取得できます。
- **カテゴリ**  
フローを分類します。多数のカテゴリがデフォルトで用意されていますが、カテゴリリストへ追加もできます。
- **CI マイナータイプ**  
これらのドメインタームは CI タイプの区分に入ることがあります（次のドメインタームを参照してください）。「ネットワークハードウェア」を例として挙げた場合、これは CI タイプのタームにあたり、CI マイナータイプのタームには、「ルーター」、「サーバー」、「ゲートウェイ」があてはまります。これらの値をリストに追加し、このドメインタームの関連入力値として入力します。
- **CI タイプ**  
これらのドメインタームは、「ネットワークハードウェア」など IT インフラ要素のメジャーな区分にあたります。これらの値をリストに追加し、このドメインタームの関連入力値として入力します。
- **構成アイテム**  
特定のサーバー、ルーター、スイッチャー、アプリケーションなど、IT を構築する要素です。構成アイテムの関連値はフロー入力で取得できます。

- **インシデント**  
フロー実行のトリガーとなるインシデントです。インシデントの関連値はフロー入力で取得できます。
- **問題**  
フローが正しく実行されているかを示すものです。これらの値をリストに追加してから、このドメインタームの関連入力値として入力します。
- **重大度**  
フローをトリガーする問題やインシデント、またはアラートの重大度を示すものです。


## ドメインタームまたはドメインターム値の追加

ドメインタームの一覧は、**Configuration** > **Domain Terms** フォルダにある Studio のリポジトリパネルに表示されています。ドメインターム用の値の作成、またはエディターでドメインタームの変更を行います。

### ドメインタームを追加するには

1. リポジトリペインで **Configuration** フォルダを展開し、**Domain Terms** フォルダを右クリックして、[New] をクリックします。
2. [Name] テキストボックスに新しいドメインタームの名前を入力します。
3. [Description] テキストボックスには説明文を入力します。
4. 作業結果を保存します。
5. 共有の Central リポジトリで作業している場合は、新規作成したドメインタームをチェックインします。

### ドメインターム値を追加または削除するには

1. リポジトリペインで、**Configuration** フォルダと **Domain Terms** フォルダを展開します。
2. 共有の Central リポジトリで作業し、編集用のドメインタームのチェックアウトが済んでいない場合は、ドメインタームを選択し、[Checkout] アイコン () をクリックします。  
詳細については、「[ライブライオブジェクトのチェックインおよびチェックアウト](#)」を参照してください。
3. 値を追加するドメインタームをダブルクリックします。
4. 作成ペインで表示されるドメインタームのエディターで、[Add] をクリックし、[Name] と [Description] の列に値と説明文をそれぞれ入力します。
5. ドメインターム値を削除するには、対象となる値を強調表示してから [Remove] をクリックします。

## ユーザーへのメッセージ表示

トランジションと同様にステップでもフローユーザーにメッセージを表示し、情報を伝えることができます（トランジションでユーザーへステップを提供する方法については、「[トランジション: ステップの接続](#)」を参照してください）。

### ステップでメッセージをフローユーザーへ表示するには

1. ステップの Inspector の [Display] タブで、[Always prompt user before executing this step] を選択します。

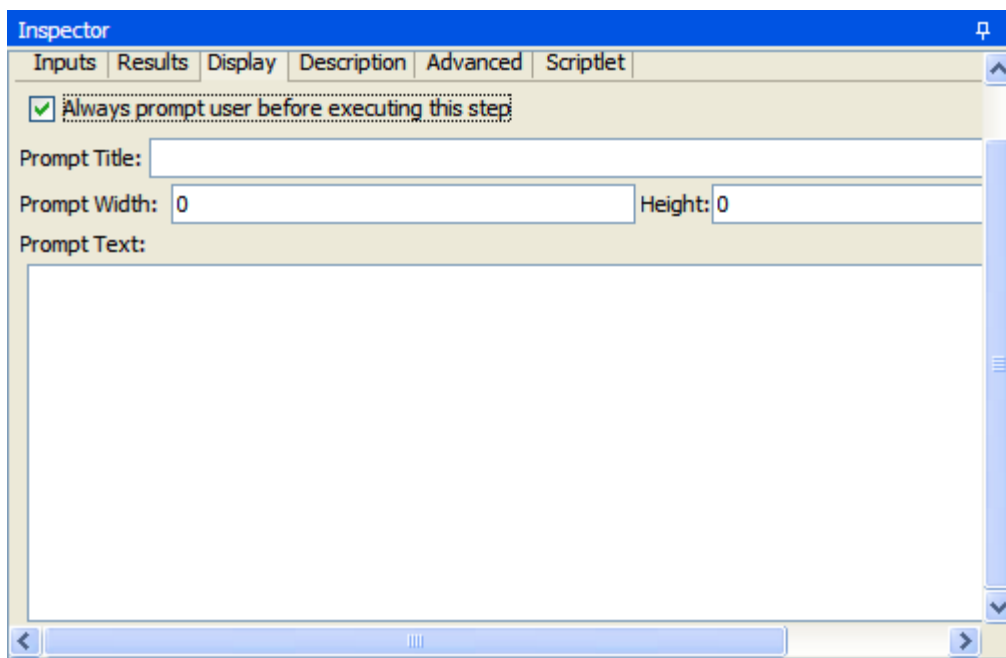


図 102 – ユーザーへの表示の指定

2. メッセージにタイトルをつけるには、[Prompt Title] ボックスにタイトルを入力します。
3. 表示の幅と高さを指定するには、メッセージまたはプロンプトの幅と高さを該当するテキストボックスにピクセル値で入力します。
4. [Prompt Text] ボックスに、表示するテキストを入力します。
5. 変更を保存します。

## カテゴリ: フローの分類

カテゴリはドメインタームの一種で、入力よりもむしろフローに割り当てるものです。フローにカテゴリを割り当てると、Central ユーザーがダッシュボードグラフ作成で利用できる要素が新たに追加されます。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。

Central 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。一例として、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。

### フローにカテゴリを割り当てるには

1. フローの [Properties] シートを開くには、フロー図の下部にある [Properties] タブをクリックします。
2. [Assign Categories] ボタンをクリックし、次のダイアログボックスから 1 カテゴリ、または複数のカテゴリ (CTRL + クリックで選択) を選択した後、[OK] をクリックします。



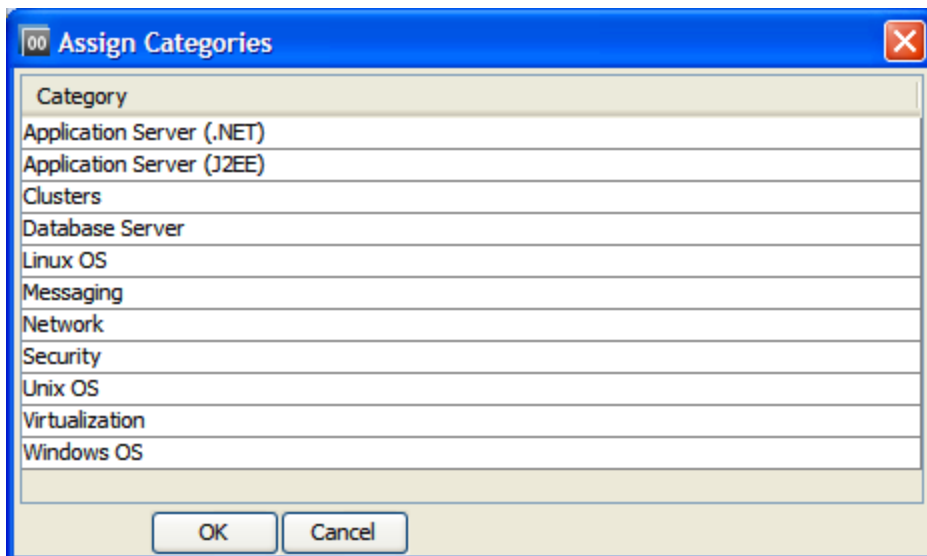


図 103 - カテゴリへのフローの割り当て

ほかのドメインタームと同じく、カテゴリの新規作成もできます。新しいドメインタームの追加については、「[ダッシュボードグラフ用のドメインターム](#)」を参照してください。

## ユーザープロンプトの選択リスト

選択リストは、フローユーザープロンプトで用いるアイテムを一覧にしたものです。多種多様な選択リストがデフォルトで用意されていますが、その多くはさまざまなテクノロジーを使用する作業に特化したものです。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 **Service Status** を指定できます（この選択リストは、**Running**、**Stopped**、**Paused** で構成されています）。また、SQL Server 認証タイプの取得には、入力用データソースの選択リストに **SqlAuthentication (Windows, Sql)** を指定することもあります。

選択リストは、Library の **Configuration** \ **Selection Lists** フォルダに格納されています。

### 選択リストを作成するには

1. リポジトリペインで、**Configuration** \ **Selection Lists** フォルダを右クリックし、[New] をクリックします。
2. 表示されたボックスで、新しい選択リストに名前を付けます。

選択リストのエディターが起動します。

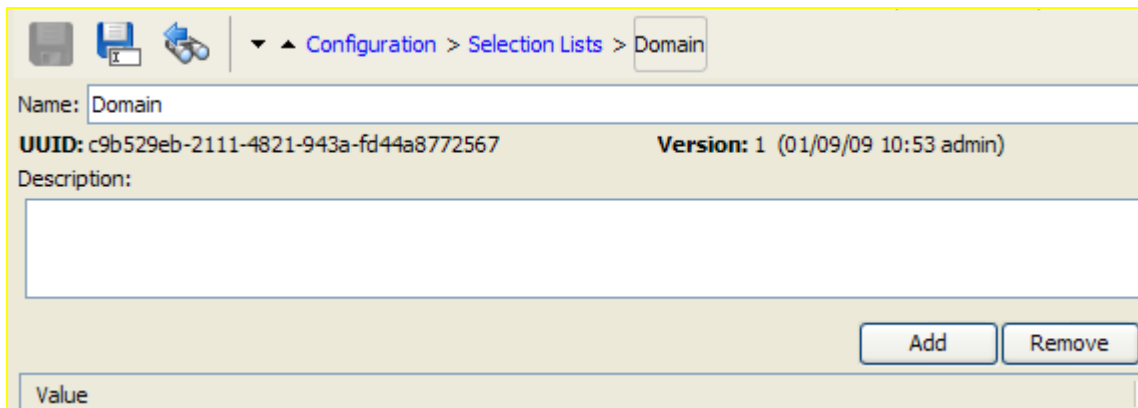



図 104 - 選択リストエディター

3. 必要に応じて、選択リストの説明文を入力します。
4. リストにアイテムを追加するには、[Add] をクリックし、アイテムに名前を付けます。
5. アイテムを削除するには、[Remove] をクリックします。
6. 選択リストエディターを閉じるには、選択リストのタブにある [X] をクリックします。


#### 選択リストを編集するには

1. リポジトリペインで、**Configuration**  **Selection Lists** フォルダを右クリックし、[Properties] をクリックします。  
選択リストの一覧が表示されます。

ここに戻る

2. リストを編集するには、そのリストの行の右端にある右向き矢印 () をクリックします。
3. リストにアイテムを追加するには、[Add] をクリックし、アイテムに名前を付けます。
4. アイテムを削除するには、[Remove] をクリックします。
5. 選択リストエディターを閉じるには、[Selection Lists] タブにある [X] をクリックします。

#### 選択リストを削除するには

1. リポジトリペインでプラス記号をクリックして、**Configuration**  **Selection Lists** フォルダを展開します。
2. 削除する選択リストを右クリックし、[Delete] をクリックします。
3. メッセージボックスが表示され、[OK] をクリックして作業を保存します。

## システムアカウント: 安全な資格情報

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。

フローを作成する際、フロー作成者は入力にシステムアカウント名を付けます。つまり、システムアカウント名はユーザーに一度も表示されることなく、フローにはリモートマシンへのアクセスに必要なユーザーアカウント資格情報が送られます。また、ユーザーはユーザープロンプトに対してもシステムアカウント名を入力できません。したがって、資格情報は暗号読解から保護され、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、Library の **Configuration**  **System Accounts** フォルダに格納されます。

### システムアカウントの作成

#### システムアカウントを作成するには



1. リポジトリペインで、**Configuration**  **System Accounts** フォルダを右クリックし、[New] をクリックします。
2. ボックスが表示されるので、新しいシステムアカウントに名前を付けます。  
システムアカウントのエディターが表示されます。

図 105 - システムアカウントエディター

3. 必要に応じて、システムアカウントの説明文を入力します。
4. 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。  
<domain>\<username>
5. [Assign Password] をクリックし、表示されたボックスにパスワードを入力します。再入力を要求された場合は再度パスワードを入力します。
6. [Save] アイコンをクリックします。システムアカウントエディターを閉じるには、選択リストのタブにある [X] をクリックします。

## システムアカウントの編集

### システムアカウントを編集するには

1. リポジトリペインで、**Configuration** ¥ **System Accounts** フォルダを開きます。編集するシステムアカウントを強調表示させ、アカウント行の右端にある右向き矢印 () をクリックします。
2. 必要な変更を終えたら、作業結果を保存します。

## システムアカウントの削除

### システムアカウントを削除するには

1. リポジトリペインで、**Configuration** ¥ **System Accounts** フォルダを開き、削除するシステムアカウントを右クリックして [Remove] をクリックします。
2. 作業結果を保存します。

## OO オブジェクトへのアクセス管理

権限とは、個々のフォルダーやフロー、オペレーション、システムアカウントなどの個別のオブジェクトへのアクセス権を指します。

個別にというよりは、グループ (ADMINISTRATOR、AUDITOR、EVERYBODY、PROMOTER、Service Desk、LEVEL\_ONE、LEVEL\_TWO、LEVEL\_THREE) 別に権限を割り当てます。つまり、リポジトリはパブリッシュするが、別の作成者がリポジトリを更新するときにフローを操作できないようにする場合には、その作成者のグループがフローへの書き込み権限を持たないように指定できます。HP OO のロールの定義や、ここで説明した HP OO グループへの外部ロールまたは外部グループのマッピングについては、HP OO の『*Administration Guide*』および **Central** のヘルプを参照してください。

また、デフォルトのアクセス権限をグループに設定することで、作成する新規オブジェクトすべてに指定した権限を付与できます。あるグループにデバッグを許可するがフロー作成を禁止する場合には、メンバーに**読み取り**、**実行**および**リンク**の権限を与えますが、**書き込み**の権限は除外します。

グループ構成と機能管理の手順は、**Central** のヘルプを参照してください。

**注:** ADMINISTRATOR ロールの権限は指定できません。ADMINISTRATOR は HP OO オブジェクトすべての全権限を有しており、これらの権限を無効にできないためです。

オブジェクトに適用される権限は次のとおりです。

### 読み取り

この権限でのみ、作成者は **Studio** のオブジェクトを、ユーザーは **Central** のフローを参照できます。

### 書き込み

作成者がオブジェクトを修正または削除するのに必要な権限です。

### 実行

この権限により、ユーザーは **Studio** または **Central** のどちらでもフローの実行を開始できます。再帰的な要件ではありません。つまり、**Central** ユーザーがフローを実行する場合や、作成者がフローのデバッグを行う場合は、フロー関連の各オブジェクトに対する**実行**の権限は必要ありません。この場合のオブジェクトとは、システムアカウントなどの設定可能なアイテムやオペレーションなどを指します。ただし、フローを実行するには、ユーザーや作成者はフローが参照するオブジェクトに対して**読み取り**と**書き込み**の権限が必要です。

### リンク

フローまたはオペレーションからのステップ作成など、フローオブジェクトの使用が作成者に許可されます。

したがって、**Central** でフローを検出したり実行したりするには、ユーザーはフローの**読み取り**と**実行**の権限が必要です。

**Studio** では：

- 作成者がフローをデバッグするには、そのフローの**実行権**が必要です。
- フロー作成者は、フロー、またはフローのステップ作成元であるオペレーションへの**リンク**の権限が必要です。
- 作成者がシステムアカウントを変更するには、システムアカウントの**読み取り**と**書き込み**の権限が必要です。

次の表は、ライブラリオブジェクト、各オブジェクトの実行可能な操作、操作に必要な権限の対応表です。

### HP OO オブジェクトとオブジェクト操作に必要な権限

オブジェクト	アクション	必要な権限
フォルダー		
	コンテンツの表示	読み取り
	コンテンツへの追加	読み取り、書き込み (子フォルダーすべてにも必要)
	移動	読み取り、書き込み
	名前の変更	読み取り、書き込み
フローまたはオペレーション		
	表示 / 開く	読み取り
	編集	読み取り、書き込み
	名前の変更	読み取り、書き込み
	実行	読み取り、実行
	ステップまたはサブフローとしての使用	読み取り、リンク
システムアカウントや選択リストなど、Configuration フォルダーのオブジェクト		
	アカウント名の表示	読み取り
	アカウントパスワードの変更	読み取り、書き込み
	アカウント名の変更	読み取り、書き込み
	フローまたはオペレーションでの使用	読み取り、リンク
	実行時に使用	読み取り、実行

## フォルダーと HP OO オブジェクトの権限の設定

権限を直接設定できるものは次のとおりです。

- フォルダー
- フローおよびオペレーション
- システムアカウント

権限変更の適用方法は、権限を変更する対象がオブジェクトかフォルダーかによって少し異なります。

- フォルダーの場合、フォルダーの全コンテンツにその権限を再帰的に適用するかどうかを指定できます。
- フローまたはオペレーションの場合、権限の定義対象にあたるフローまたはオペレーションの参照先、もしくはその参照元となるフロー、オペレーション、システムオブジェクト（ドメインタームや **Configuration** フォルダーのアイテム）すべてにその権限を適用するかどうかを指定できます。参照は間接的な場合も直接的な場合もあります。

注: システムアカウントへ権限を直接設定できるほか、これ以外の **Configuration** フォルダーの全システムオブジェクトには、暗黙的にのみ権限を設定できます（フローまたはオペレーションへ設定した権限をその参照先の全オブジェクトにも適用するように指定する必要があります）。

### フォルダーまたは HP OO オブジェクトへ権限を設定するには

1. Library で、オペレーション、フローまたはシステムアカウントを右クリックし、メニューから [Permissions] をクリックします。

次のようなダイアログボックスが表示されます。

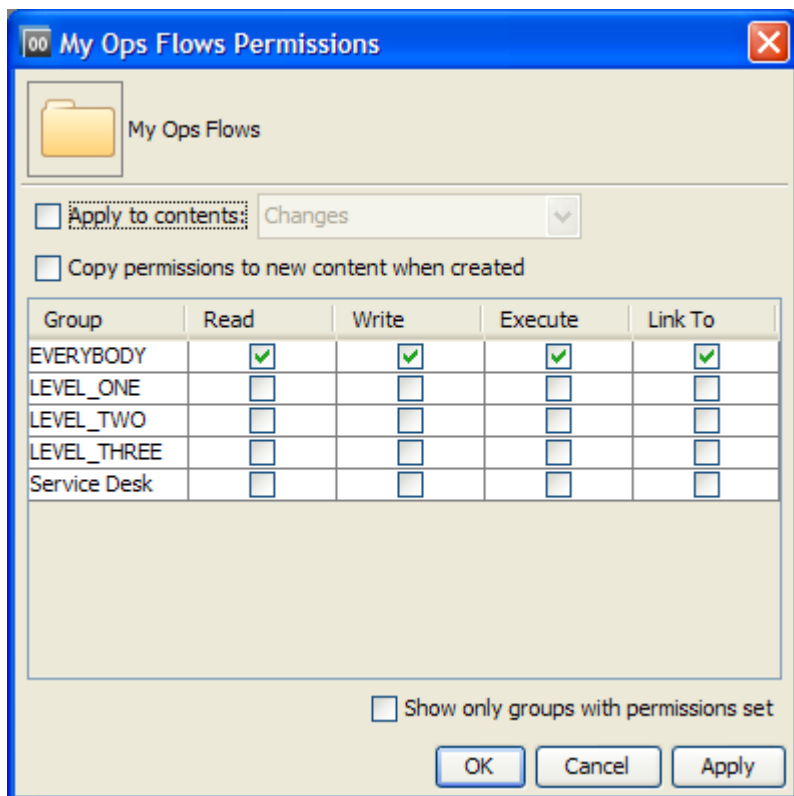


図 106 — 権限の設定



ヒント: [Show only groups with permissions set] のチェックボックスにチェックをつけると、1つの権限も設定されていないグループは非表示にできます。

2. アクセス許可の必要性に応じて、ユーザーの各グループの権限を選択します。

注:

- グループの権限を変更すると、その権限のボックスが黒枠で囲まれます。
- フローの各親フォルダーには、フローへ設定した権限と同じものを指定します。

[Apply to] ボックスには、[Apply to contents] または [Apply to referenced objects] のどちらかが、アクセス権限を指定するオブジェクトの種類に応じて表示されます。

- フォルダーの場合は、[Apply to contents] となります。
- フロー、オペレーション、システムアカウントの場合は、[Apply to referenced objects] となります。  
このダイアログボックスに表示される [Referenced objects] とは、オブジェクトが参照するフローやオペレーション、システムオブジェクトだけでなく、オブジェクトを参照するフローとオペレーションも対象となります。

3. フォルダーに権限を設定する場合、フォルダーのアクセス権限をフロー、オペレーション、サブフォルダー（コンテンツを含む）すべてに適用するには、[Apply to contents] を選択します。

もしくは

フロー、オペレーション、またはシステムアカウントに権限を設定する場合、オブジェクトの参照先あるいは参照元であるフロー、オペレーション、その他のシステムアカウントすべてにオブジェクトのアクセス権限を適用するには、[Apply to referenced objects] を選択します。

次に、設定済みだが未適用の権限（グリッド表中の太字のもの）、もしくは現在表示されている権限すべてに対して、変更点のみ適用する方法を説明します。

4. フォルダーまたはオブジェクトにまだ適用されていない変更を適用するには、[Apply to...] ボックス横のドロップダウンリストから [Changes] を選択します。

もしくは

指定したばかりの設定も含め、現在指定しているグリッド表中の権限すべてに変更を適用するには、[Apply to...] ボックス横のドロップダウンリストから [All] を選択します。

次の手順は、フォルダーに権限を指定していることを想定したものです。

5. フォルダー内に後で追加した任意のフロー、オペレーションまたはシステムアカウントに権限を適用するには、[Copy permissions to new content when created] を選択します。

このチェックボックスを選択すると、フォルダーに設定していた権限が、グループにデフォルトで設定されていた権限に上書きされます。次のような例が考えられます。

- **LEVEL\_THREE** グループにデフォルトのグループマスクの**書き込み権**を設定する場合
- フォルダー A への権限について、**LEVEL\_THREE** グループに**読み取り権**のみを設定する場合
- フォルダー A の権限を今後、同フォルダーで作成されるものすべてに適用するように設定する場合

設定後にフォルダー A にフローを作成した場合、**LEVEL\_THREE** グループは新しくできたフローの**読み取り権**はありますが、**書き込み権**はありません（新規フローへの **LEVEL\_THREE** グループの権限は設定後も変更できません）。

グループの権限のデフォルト設定については、「[グループのアクセス権限のデフォルト設定](#)」を参照してください。

6. [OK] をクリックし、作業結果を保存します。



## グループのアクセス権限のデフォルト設定

新規作成したオブジェクトに対するアクセス権限をグループごとにデフォルトで設定できます。アクセス権限のデフォルト設定をセットにしたものを、「グループマスク」といいます。ただし、フォルダーに何らかのオブジェクトを作成し、そのオブジェクトのアクセス権限とグループマスクが対立した場合は、そのフォルダーに設定したアクセス権限が優先されます。

たとえば、次のような場合があります。

- グループマスクがデフォルトで設定された **LEVEL\_THREE** グループに、**書き込み権**を設定する場合
- フォルダー A への権限について、**LEVEL\_THREE** グループに**読み取り権**のみを設定する場合
- フォルダー A の権限を今後、同フォルダーで作成されるものすべてに適用するように設定する場合

設定後にフォルダー A にフローを作成した場合、**LEVEL\_THREE** グループは新しくできたフローの**読み取り権**はありますが、**書き込み権**はありません（新規フローへの **LEVEL\_THREE** グループの権限は設定後も変更できません）。

### デフォルトのグループアクセス権限を設定するには

1. [Tools] メニューから [Set Group Mask] をクリックします。
2. 次のダイアログボックスで、Studio および Central で各グループが持つデフォルトの権限をそれぞれ選択し、[OK] をクリックします。ここで設定した権限は HP OO の新規オブジェクトすべてに適用されます。

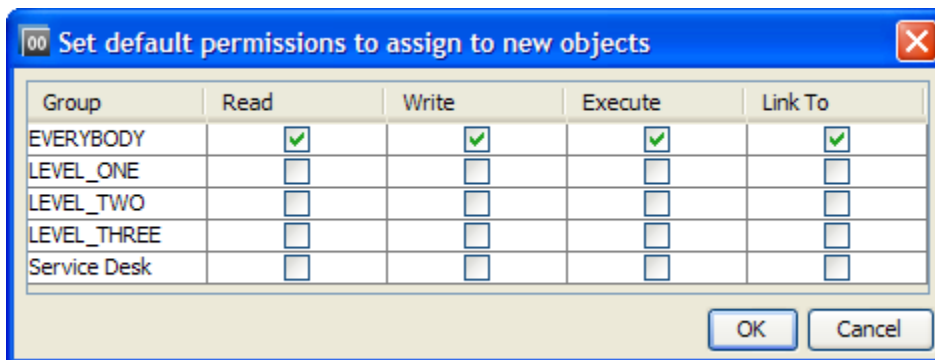


図 107 - 新規オブジェクトのデフォルト権限の設定

## 新しいオペレーションを作成するには

どの種類のオペレーションが必要かによって、オペレーション作成には次の処理が一部必要になってきます。本ヘルプシステムのほかの項でも説明しています。

- ステップおよびオペレーションの「**入力**」および「**出力**」の定義
- オペレーションを実行するリモートアクションサービス (RAS) の参照 (Central 外での運用の場合。オペレーションのコマンドが Central リポジトリで定義されていない場合、もしくは Central サーバーに対してリモート環境にあるマシンでオペレーションを実行する場合を指す)
- 「**スクリプトレット**」の記述
- トランジションを決定する「**レスポンス**」の指定、およびレスポンスへの結果のマッピング
- OO ロールへのオペレーション「**アクセス権限**」の必要に応じた割り当て
- 「**フロー変数**」での出力データの保存

オペレーションの有効化には次の処理が必要です。

- 各入力用データソースの定義を用いた 1 つ以上の入力
- オペレーション結果を表す有効式へのレスポンスのマッピング
- オペレーションタイプに応じたプロパティの指定

オペレーション作成後は、オペレーションの [Properties] シートを完了させます。オペレーションの全タイプに共通の [Properties] シートのアイテムは次の手順で確認できます。この手順を終了後、作成したオペレーションタイプに特定のプロパティを指定します。オペレーション特有のプロパティの説明一覧については、この手順の後に記載している「[オペレーションタイプ: プロパティの設定](#)」を参照してください。

### オペレーションを作成するには

1. オペレーションを作成するフォルダーを右クリックし、[New]、[Operation] の順にポイントして、表示されたリストからオペレーションタイプを選択します。
2. 表示されたダイアログボックスで、テキストボックスに新しいオペレーションの名前を入力し、[OK] をクリックします。

#### 注:

- Studio では、名前の大文字と小文字は区別されません。同じ名前を 2 つのオペレーションにつけることはできません。
  - 名前の長さは最大 128 文字までです。
  - 新しいオペレーションの [Properties] シートが**作成**ペインに表示されます。
3. 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、[Assign Categories] をクリックし、リストからカテゴリを選択します。
  4. オペレーションで入力が必要とされたら、[Inputs] タブをクリックし、[Add Input] をクリックします。フロー変数から入力値が取得できます。
  5. 必要な入力を追加し、データソースを割り当てます。
  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。  
入力の追加については、「[入力: オペレーションへのデータの提供](#)」を参照してください。データソースを入力に割り当てるときのフロー変数の使用については、「[フロー変数: 再利用可能なデータの作成](#)」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（結果）の追加と運用については、「[出力、レスポンス、およびステップの結果](#)」を参照してください。
  8. 必要なレスポンスを作成します。  
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<sup>1</sup>の定義については、「[レスポンス: 結果の評価](#)」を参照してください。  
オペレーション各タイプに特定したレスポンス定義については、この手順の後に記載している「[オペレーションタイプ: プロパティの設定](#)」を参照してください。
  9. Central ユーザー対象のオペレーションを記録するには、[Description] タブをクリックし、テキストボックスに説明文を入力します。
  10. プロパティの指定を完了するには、次のセクション（「[オペレーションタイプ: プロパティの設定](#)」）で説明している手順に従い、最後に [OK] をクリックします。

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**リポジット**ペインに**赤**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（オペレーション有効化の要件を確認するには、この手順の前に記載されているリストを参照してください）。

## オペレーションタイプ: プロパティの設定

作成可能なオペレーションのクラス一覧を次に示します。

- Central および Studio でのオペレーション使用上のガイドライン
- 各オペレーションタイプの [Properties] シートのスクリーンショット、およびオペレーションで設定が必要なプロパティの注意点

cmd、HTTP、Secure シェル、Telnet オペレーションでは、**op-timeout** 入力をオプションで追加できます。入力的时间値はミリ秒単位で指定できます。デフォルト値は2分（120000 ミリ秒）です。ほかにも **op-timeout** 入力追加できます。入力の追加手順は、「オペレーションを作成するには」を参照してください。

**注:** RAS オペレーションは、IAction のプロキシとしてしか動作しないため、**op-timeout** 入力は追加できません。任意のオペレーションタイムアウトを IAction でプログラムする必要があります。

一部の RAS オペレーションタイプは、オペレーションの作成方法が多岐にわたります。オペレーションタイプの使用推奨メソッドについては、関連項目で説明しています。

次のスクリーンショットでは、オペレーションの [Properties] シートにオペレーションの汎用一意識別子 (UUID) と最後に編集した日付が入っています。

### Cmd (コマンドライン) オペレーション

既存コマンド、スクリプト、バッチ、または実行ファイルの実行に使用します。フロー以外ではコマンドラインから実行するものが対象です。cmd オペレーションで実行するプログラミングもまた、IAction の作成またはスクリプトレットの使用により実行できる場合もあります。ただし、スクリプトレットの使用は推奨していません。

cmd オペレーションでは1つのコマンド（スクリプト、バッチ、実行ファイルなど）を実行できます。一連のコマンドを実行するには、シェルオペレーションを使用する必要があります。シェルオペレーションについては、「[シェルオペレーション](#)」を参照してください。

- スクリプトレットの使用については、「[スクリプトレットオペレーション](#)」を参照してください。
- IAction プログラミングの使用については、「[オペレーション用の IAction の作成](#)」および OO の『*SDK Guide*』（[SDKGuide.pdf](#)）を参照してください。

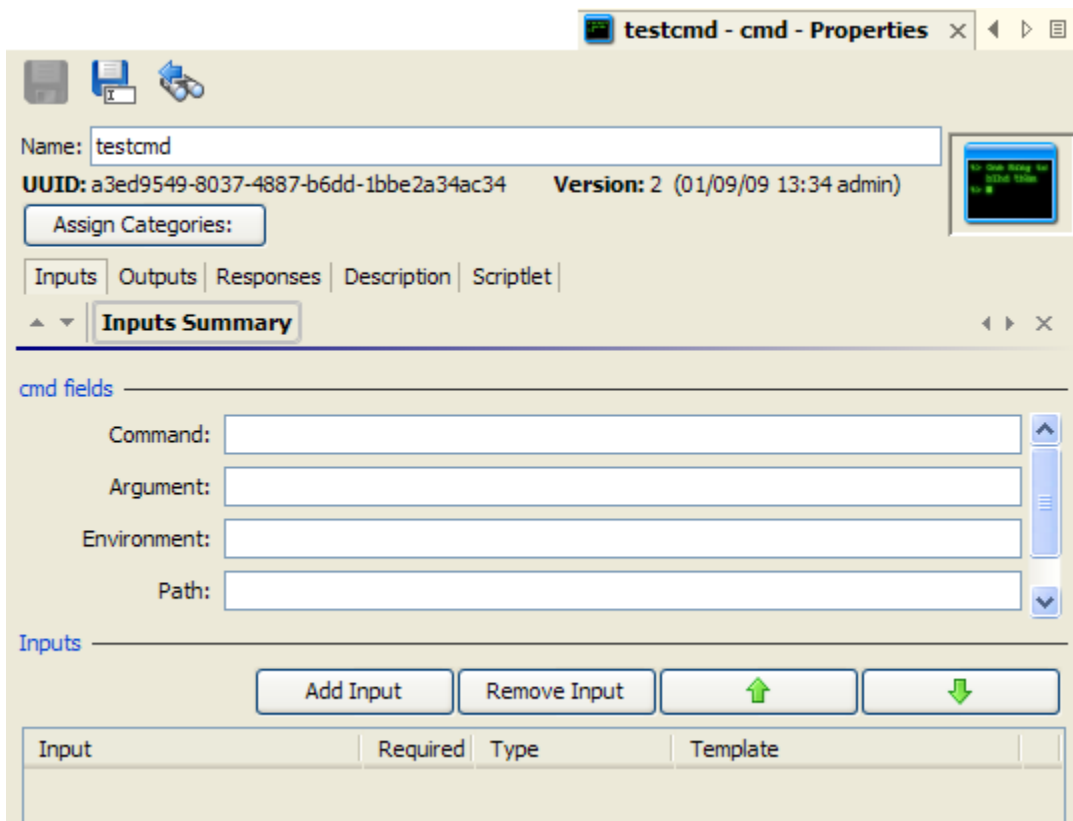


図 108 - Cmd オペレーションのプロパティ

入力に必要な値については、オペレーションの [Description] タブを参照してください。

[Responses] タブ上のレスポンスエディター（レスポンス行の右向き矢印をクリックして開きます）で、[Apply Rule to Field] 列の次のタームを使用して、レスポンスの結合先となる命令文を生成できます。これらのタームについては、オペレーションの [Description] タブを参照してください。[Result Fields] の下部に詳細が表示されます。

- [Code]
- [Output String]
- [Error String]
- [FailureMessage]
- [TimedOut]

## フロー実行サマリレポートオペレーション

実行したステップのサマリレポートを生成します。ただし、このコマンドを含むステップは除外されます。

Input	Required	Type	Template
runHistoryID	<input type="checkbox"/>	Single Value	Value:
formatHTMLYesNo	<input type="checkbox"/>	Single Value	Value: false
flowVariableName	<input type="checkbox"/>	Single Value	Value:

図 109 - フロー実行サマリレポートオペレーションのプロパティ

次に Http オペレーションの入力を説明します。次の値が必要です。

[runHistoryID] - OO データベースの実行履歴用の数値 ID です。空白の場合は、現在の実行履歴 ID が使用されます。

[formatHTMLYesNo] - 定数値に **true** を指定した場合、オペレーション結果は HTML 表の形式で生成されます。デフォルトは **false** です。結果はテキスト形式で生成されます。

[flowVariableName] - 定義した場合、この入力では 2 次元配列 [N][4] が次のフロー変数に変数名とともに代入されます。

- [N] はレポートのステップ数です。
- 4 列には次のものが格納されます。インデックス [0] はフローのステップへのパス、インデックス [1] はステップ名、インデックス [2] はレスポンス、インデックス [3] は説明文です。

## HTTP オペレーション

オペレーションの単純な発動または取得に使用します。イントラネットまたはエクストラネットからログファイルなどを取得します。

IAction プログラミングは、これらのタスクをまったく異なる方法で実行するわけでも、通常の HTTP オペレーションに比べて利点があるわけでもないため、どちらか適当なほうを選択してください。

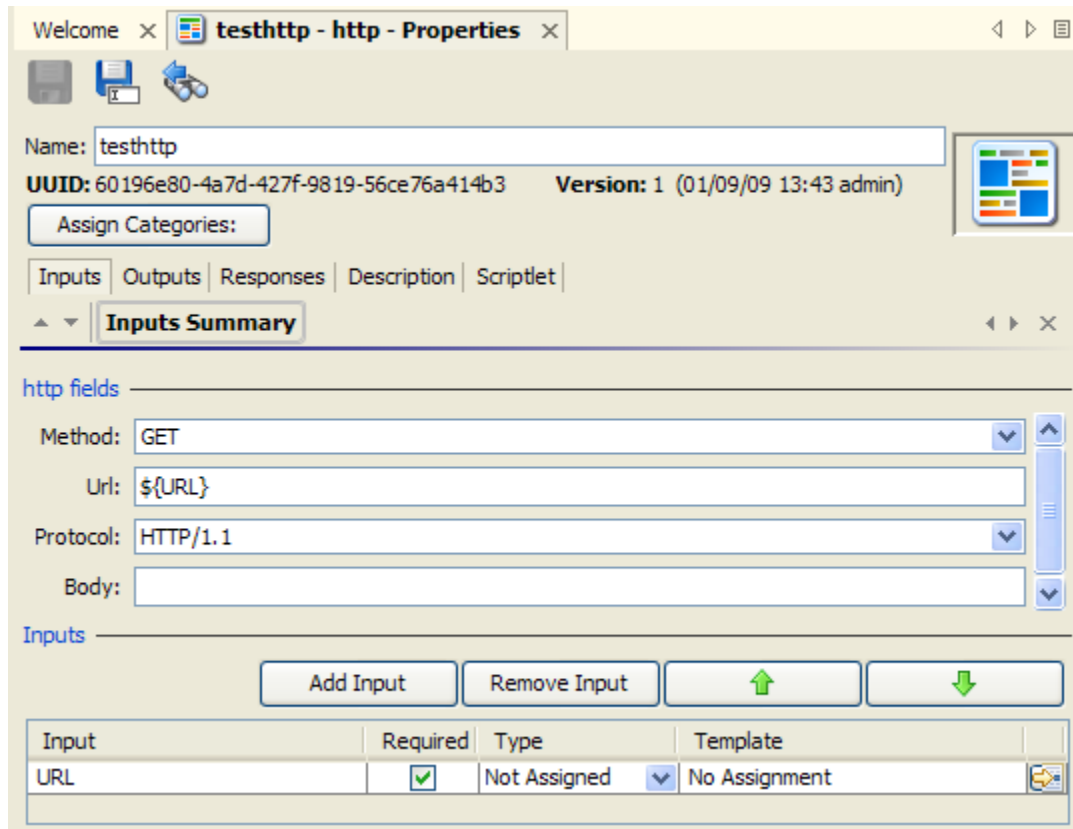


図 110 – HTTP オペレーションのプロパティ

次に Http オペレーションの入力を説明します。次の値が必要です。

[Method] – HTTP プロトコルメソッドです。ドロップダウンリストから 1 つ選択します。

[Url] – ターゲットとする Web ページの URL です。

[Protocol] – HTTP プロトコルのバージョンです。ターゲットサーバーがサポートするバージョンを選択します。

[Body] – リクエストにボディ部がある場合は、RFC 2616 の規定に従い、ボディのデータを入力します。

[Responses] タブ上のレスポンスエディター（レスポンス行の右向き矢印をクリックして開きます）で、[Apply Rule to Field] 列の次のタームを使用して、レスポンスの結合先となる命令文を生成できます。

- [code]  
RFC 2616 で規定された HTTP 戻りコード
- [reason]  
RFC 2616 で規定された HTTP 戻りコードの文字列値
- [headers]  
RFC 2616 で規定されたレスポンスヘッダー

- [document]  
サーバーから実際に戻された記録
- [FailureMessage]  
失敗時に受信するメッセージ
- [TimedOut]  
[TimedOut] 値と「true」（かぎかっこは含まない）との比較
- [Result]  
オペレーション結果

## Perl スクリプトオペレーション

Perl スクリプトレットを実行する Perl スクリプトの呼び出しに使用します。Perl スクリプトレットによってフロー内で変数の受け渡しおよび変数値の編集ができ、編集した変数値を Perl スクリプトに戻すこともできます。

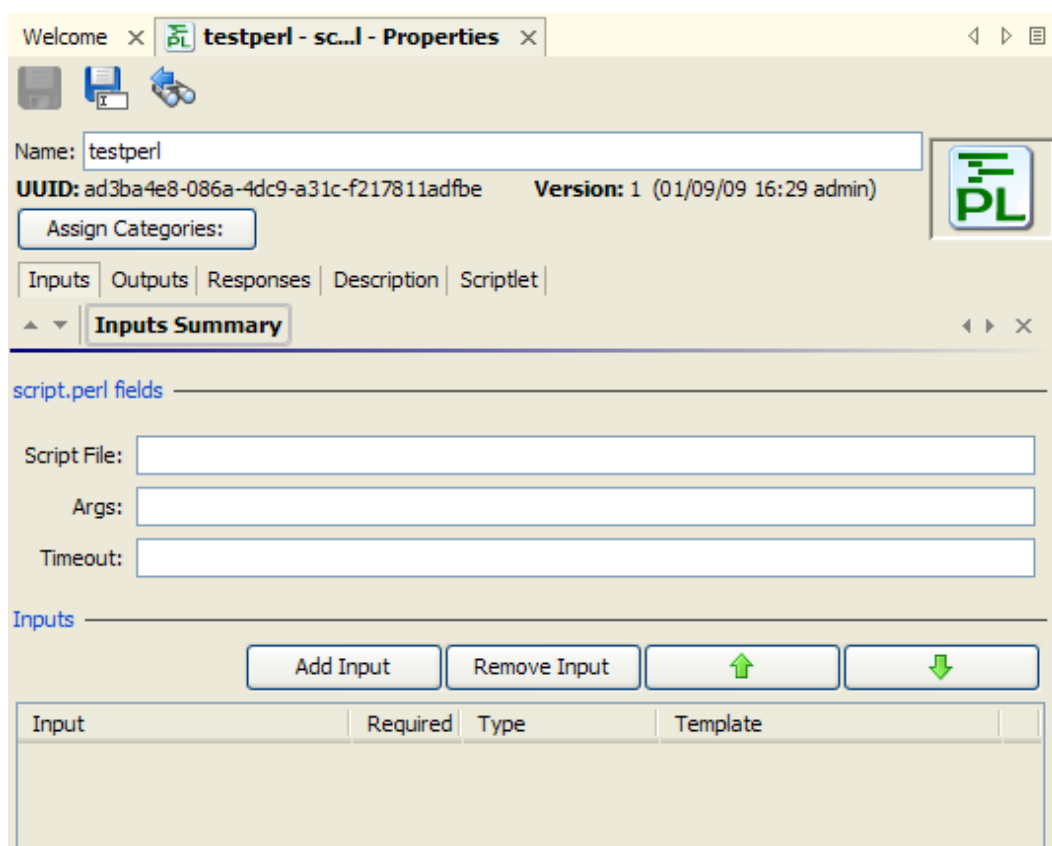


図 111 – Perl スクリプトオペレーションのプロパティ

次に Perl スクリプトオペレーションへの入力を説明します。次の値が必要です。

[Script File] – 実行する Perl スクリプトの名前

[Args] – スクリプトに渡す引数

[Timeout] – スクリプトの最大実行時間



[Responses] タブ上のレスポンスエディター（レスポンス行の右向き矢印をクリックして開きます）で、[Apply Rule to Field] 列の次のタームを使用して、レスポンスの結合先となる命令文を生成できます。

- [Code]  
オペレーションの数字の戻りコードです。オペレーションが成功した場合、値は通常、0 になります。
- [Output String]  
オペレーションの標準出力（オペレーションでの `stdout` への書き込み）
- [Error String]  
オペレーションのエラー出力（オペレーションでの `stderr` への書き込み）
- [Script Response]  
スクリプトオペレーションにおいて、スクリプトで戻される文字列
- [Script Result]  
スクリプトオペレーションにおいて、スクリプトで戻される出力
- [FailureMessage]  
失敗時に受信するメッセージ
- [TimedOut]  
[TimedOut] 値と「true」（かぎかっこは含まない）との比較
- [Result]  
オペレーション結果

## RAS オペレーション: RAS のための IAction プログラミング

フローの作業の大部分に使用します。ユーザーの所属する組織が IAction オブジェクトのプログラミングリソースを有していることを想定しています。

IAction プログラミングの利点は次のとおりです。

- 大容量データの受け渡しの処理能力の強化
- 高速化
- プロセス開始時のオーバーヘッドの削減
- パフォーマンスの向上
- 優れたスケーラビリティ
- 確かな安定性
- 再利用の容易さ

スクリプトレットオペレーションで行うプログラミングもまた、コマンドまたは IAction プログラミングオペレーションで実行可能です。コマンドまたは IAction プログラミングオペレーションが指示されるタイミングについては、このリストの各解説を参照してください。

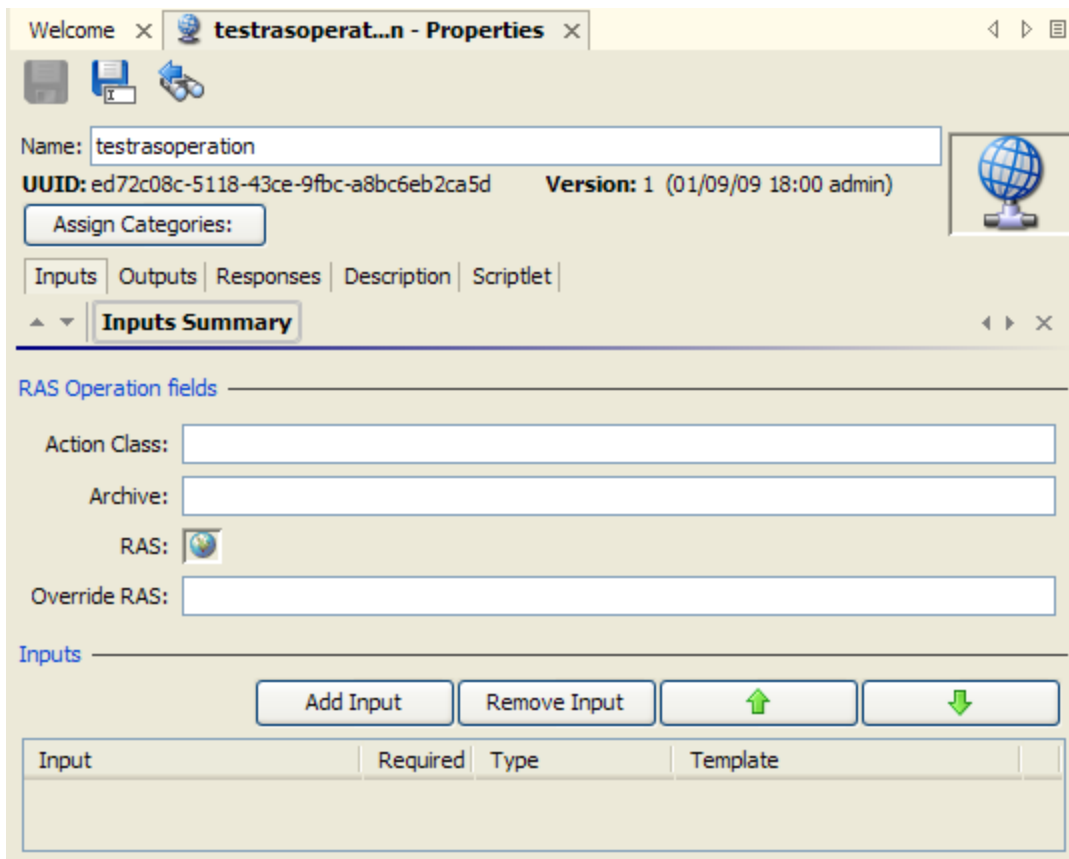


図 112 – RAS オペレーションのプロパティ

次に RAS オペレーションの入力について説明します。次の値が必要です。

[Action Class] – アクションクラス名。このアクションクラスからオペレーションを作成します。

[Archive] – アクションクラスが格納される Java アーカイブ (.jar) または動的リンクライブラリ (.dll) ファイルの名前

[RAS] – Studio で設定した RAS 参照名。参照先は、インストールしたデフォルトまたはカスタムの RAS です。どちらかでオペレーションがアクションクラスを実行します。RAS 参照を変更するには、RAS 参照を **Configuration** \ **Remote Action Services** フォルダから [RAS] フィールドのラベルの隣にある [Globe] アイコン (🌐) にドラッグします。

[Override RAS] - オペレーション (前述の [RAS] フィールドで指定したオペレーション) と関連付けられている RAS の代わりに使用する RAS の名前



ヒント: `{rasOverride}` などの作成したフロー変数を使用して、上書きが指定する RAS を格納できます。

オペレーションでの RAS 参照変更、またはステップでの RAS 参照の上書きについては、「[RAS 参照の変更](#)」を参照してください。

[Responses] タブ上のレスポンスエディター (レスポンス行の右向き矢印をクリックして開きます) で、[Apply Rule to Field] 列の次のタームを使用して、レスポンスの結合先となる命令文を生成できます。これらのタームについては、オペレーションの [Description] タブを参照してください。[Extra Results] の下部に詳細が表示されます。

- [FailureMessage]  
失敗時に受信するメッセージ
- [TimedOut]  
[TimedOut] 値と「true」（かぎかっこは含まない）との比較

RAS、.jar、.dll の概要と、フロー機能を拡張する上での使用法については、「[リモートアクションサービスを使用した Central 外での運用](#)」および「[オペレーション用の IAction の作成](#)」を参照してください。

## スクリプトレットオペレーション

複数のフローで再利用可能なスクリプトレットの作成に使用します。

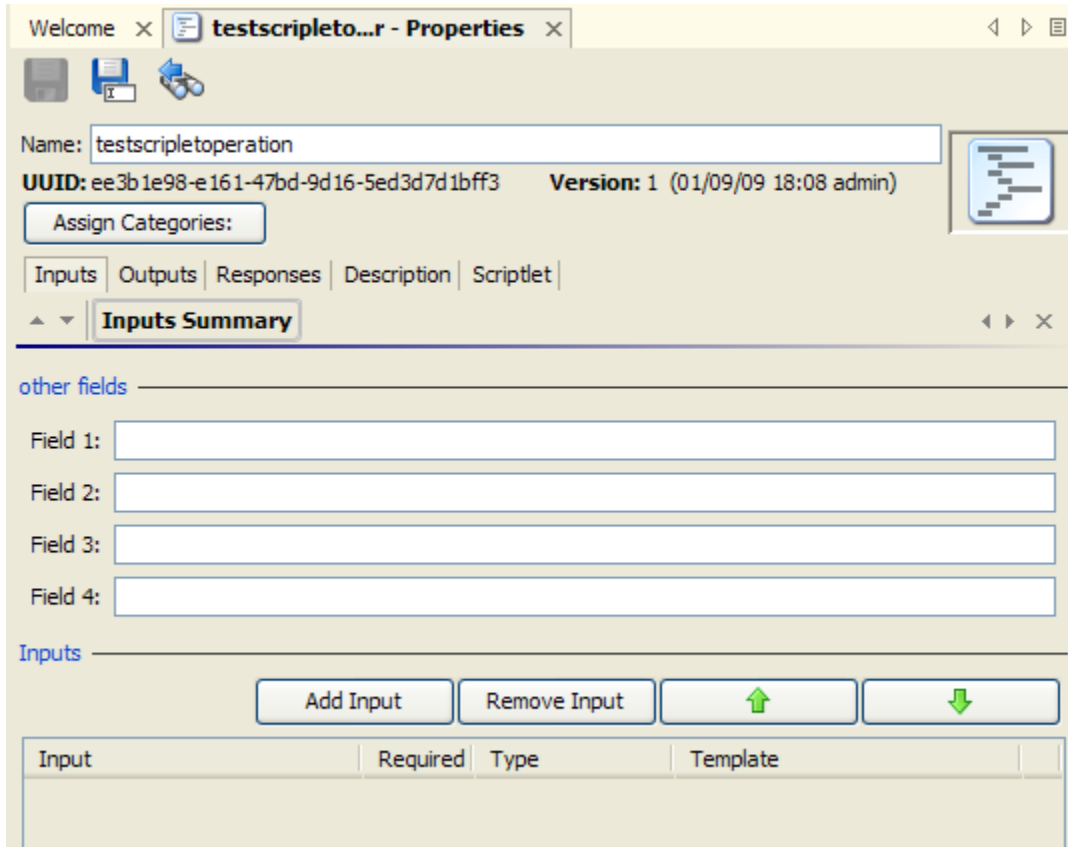


図 113 – スクリプトレットオペレーションのプロパティ

最大 4 つのデータを収集してフロー変数に受け渡し、フローの別用途に利用する場合に使用します。

スクリプトレットオペレーションの用途の 1 つに、ユーザープロンプトを指定して、フロー全体を自動的に実行することがあります。このとき、ステップがフロー変数からプロンプトの入力値を別途取得できるようにしておく必要があります。

次にスクリプトレットオペレーションの入力について説明します。次の値が必要です。

[Field 1] から [Field 4] – 4 つのフィールドには、このオペレーションに定義した入力を使用して投入できます。

[Responses] タブ上のレスポンスエディター（レスポンス行の右向き矢印をクリックして開きます）で、[Apply Rule to Field] 列の次のタームを使用して、レスポンスの結合先となる命令文を生成できます。

- [Field 1] から [Field 4]  
4つのフィールドのコンテンツは、入力からのデータを格納したものです。
- [FailureMessage]  
失敗時に受信するメッセージ
- [TimedOut]  
[TimedOut] 値と「true」（かぎかっこは含まない）との比較
- [Result]  
スクリプトで戻される出力

## Secure シェル (ssh) オペレーション

セキュアチャンネル上のリモートコマンドに使用します。

たとえば、オペレーティングシステムが Linux の Red Hat バージョンのマシンでサービスを開始したい場合は、**Start Service** フローを使用できます（フローは Library の、**Accelerator Packs** **Operating Systems** **Red Hat** **State change** にあります）。**Start Service** フローでは、SSH オペレーションである **Run Service** オペレーション（Library の、**Operations** **Operating Systems** **Linux** **SUSE Linux** **Process Operations** にあります）を使用します。この用途での **Run Service** オペレーションへの入力、出力、レスポンスの設定方法は、オペレーションを開くと表示されます。

The screenshot shows a configuration window for 'testsecureshellproperties'. It includes fields for Name, UUID, and Version. Below these are tabs for Inputs, Outputs, Responses, Description, and Scriptlet. The 'Inputs Summary' section is expanded, showing 'ssh fields' with input fields for Command, Argument, Environment, Host, User, Password, and Private Key File. There is also a checkbox for 'Pty'. At the bottom, there are buttons for 'Add Input', 'Remove Input', and arrows for moving items. A table lists the current inputs:

Input	Required	Type	Template
host	✓	Not Assigned	No Assignment
identity	✓	Credentials	Prompt user for credentials

図 114 — Secure シェルオペレーションのプロパティ

入力に必要な値については、オペレーションの [Description] タブを参照してください。

注:

- **Host** 入力で、ホストに特定のポートを指定するには、次の構文を使用してホストとポートを指定します。  
<ホスト名>:<ポート名>
- [Pty] チェックボックスにチェックをつけると、擬似端末が作成されます。これにより、端末が必要な Unix オペレーションの実行が、作成したセキュアシェルオペレーションから可能となります。  
警告: 擬似端末は標準出力（出力文字列）と標準エラー（エラー文字列）を区別しないため、両方とも受け取ります。そのため、エラー文字列は常に空です。

[Responses] タブ上のレスポンスエディター（レスポンス行の右向き矢印をクリックして開きます）で、[Apply Rule to Field] 列の次のタームを使用して、レスポンスの結合先となる命令文を生成できます。これらのタームについては、オペレーションの [Description] タブを参照してください。[Extra Results] の下部に詳細が表示されます。

- [Code]
- [Output String]
- [Error String]
- [FailureMessage]
- [TimedOut]
- [Result]

## Telnet オペレーション

Telnet プロトコルを使用してメッセージまたはコマンドをサーバーに送るときに、使用します。

testtelnet - ...t - Properties

Name: testtelnet

UUID: 1dc62329-0123-4433-ab36-9467c1547cfb Version: 1 (01/09/09 18:43 admin)

Assign Categories:

Inputs Outputs Responses Description Scriptlet

Inputs Summary

telnet fields

Command:

Argument:

Environment:

Path:

Host:

Port:

Session Name:

Terminal Expression:

Inputs

Add Input Remove Input ↑ ↓

Input	Required	Type	Template
-------	----------	------	----------

図 115 – Telnet オペレーションのプロパティ

次に Telnet オペレーションの入力について説明します。次の値が必要です。

[Command] – オペレーションが実行するコマンド

[Argument] – コマンドライン引数 (スイッチ)

[Environment] – 既存の環境変数への追加

[Path] – コマンド実行パス



**ヒント:** ホストに特定のポートを指定するには、次の構文を使用してホストとポートを指定します。

<ホスト名> <ポート名>

次のことに注意してください。Telnet オペレーションでは、<ホスト名> と <ポート名> の間は1つのスペースで区切られています。

[Host] – コマンドのターゲットコンピューター

[Port] – オペレーションをターゲットコンピューターに接続するポート

[Session Name] – コンテキストに格納するセッションの名前

[Terminal Expression] – オペレーションの完了通知シーケンスをオペレーションレスポンスから検出するために使用できる正規表現です。コマンドによって異なるため、このフィールドを用いるには Telnet コマンドの戻り値に関する知識が必要です。このフィールドを使用しない場合、Telnet オペレーションでは Telnet タイムアウト値に達するまでコマンドを実行できます。

[Responses] タブ上のレスポンスエディター（レスポンス行の右向き矢印をクリックして開きます）で、[Apply Rule to Field] 列の次のタームを使用して、レスポンスの結合先となる命令文を生成できます。

- [Code]  
オペレーションの数字の戻りコードです。オペレーションが成功した場合、値は通常、0 になります。
- [Output String]  
オペレーションの標準出力（オペレーションでの stdout への書き込み）
- [Error String]  
オペレーションのエラー出力（オペレーションでの stderr への書き込み）
- [FailureMessage]  
失敗時に受信するメッセージ
- [TimedOut]  
[TimedOut] 値と「true」（かぎかっこは含まない）との比較
- [Result]  
オペレーション結果

## シェルオペレーション

コマンドラインオペレーションが実行できるコマンドは 1 つだけですが、これとは対照的にシェルオペレーションは一連のコマンドの実行が可能です。一連のコマンドは、シェルオペレーションから作成したステップへの入力で指定します。シェルオペレーションの使用方法は次の 2 つです。

- OO コンテンツ（Studio のインストールと同時にインストールされているフローおよびオペレーションのコレクション）にデフォルトで用意されている 3 つのシェルオペレーションの 1 つを使用する。3 つのシェルオペレーションのうち、1 つは SSH プロトコルを、もう 1 つは Telnet プロトコルを使用し、最後の 1 つはどちらかのプロトコルを使用する。
- Shell Wizard でシェルオペレーションを使用するフローを作成する。  
Shell Wizard の使用方法については、次のトピック「[Shell Wizard: シェルオペレーションを使用するフローの作成](#)」を参照してください。  
シェルオペレーションの使用については、「[シェルオペレーションの使用](#)」を参照してください。

### Shell Wizard: シェルオペレーションを使用するフローの作成

Shell Wizard のガイドにより、Secure Sockets (SSH) または Telnet プロトコルのどちらかを基にしたシェルオペレーションを含むフローを作成します。ウィザードには、シェルウィンドウで実行するコマンドからフローのステップを作成するレコーダー形式のテクノロジーを導入しています。

ウィザードの実行には、ホスト（指定コマンドを実行するマシン）の次の接続情報が必要になります。



- ホスト名
- ホストでコマンドを実行するときのユーザーアカウント名とパスワード

注: Shell Wizard は、SSH サーバーまたは Telnet サーバーのいずれもインストールされていないサーバーとは通信できません。

### シェルオペレーションを使用するフローを作成するには

1. HP OO のホームディレクトリの **¥Studio¥tools¥** フォルダで、**shellwizard.exe** をクリックまたはダブルクリックします。

Shell Wizard が起動します。

2. [Select repository] ページで、コマンドライン (シェル) オペレーションを作成するリポジトリの場所を入力します。

もしくは

[Browse] をクリックし、リポジトリの場所に移動して、[Open] をクリックします。

#### 重要:

- 空または存在しないフォルダを指定した場合、ウィザードは必要に応じてその場所に入力した名前のフォルダを作成し、新しいリポジトリを作成します。新しいリポジトリに格納されるのは、作成するオペレーションの実行に必要なものだけです。そのうちの SSH Shell または Telnet Shell オペレーションは、このウィザードで作成するオペレーションの実行を有効にします。
  - 既存のリポジトリを指定する場合は、リポジトリに SSH Shell オペレーション (SSH プロトコル使用の場合)、または Telnet Shell オペレーション (Telnet プロトコル使用の場合) が含まれている必要があります。適切な Shell オペレーションが含まれていない場合は、ウィザードを完了できません。SSH Shell または Telnet Shell オペレーションは、Studio のデフォルトコンテンツ (オペレーション、フロー、および HP OO オブジェクト) に入っています。ただし、Studio の Library のサブフォルダをリポジトリとしてエクスポートする場合は、エクスポートにデフォルトコンテンツが含まれている必要があります。エクスポートにデフォルトコンテンツが含まれていない場合は、エクスポートするフォルダのフローまたはオペレーションが使用するデフォルトコンテンツのみ、エクスポートに含めます。
  - Shell Wizard を空のフォルダを指定するか (Studio リポジトリへは後でインポート可能)、Studio の Library 全体のエクスポートを指定することをお勧めします。
3. リポジトリを入力するかナビゲートによってリポジトリを特定し、[Next] をクリックします。  
[Specify flow information] ページが表示されます。
  4. ウィザードの [Specify flow information] ページで、[Enter a name...] ボックスにシェルオペレーション用にウィザードで作成するフローの名前を入力します。
  5. [Enter a brief description] ボックスに簡単な説明を入力し、[Next] をクリックします。

Central ユーザーは入力した説明を基にフローを選択することもありますので、ご注意ください。

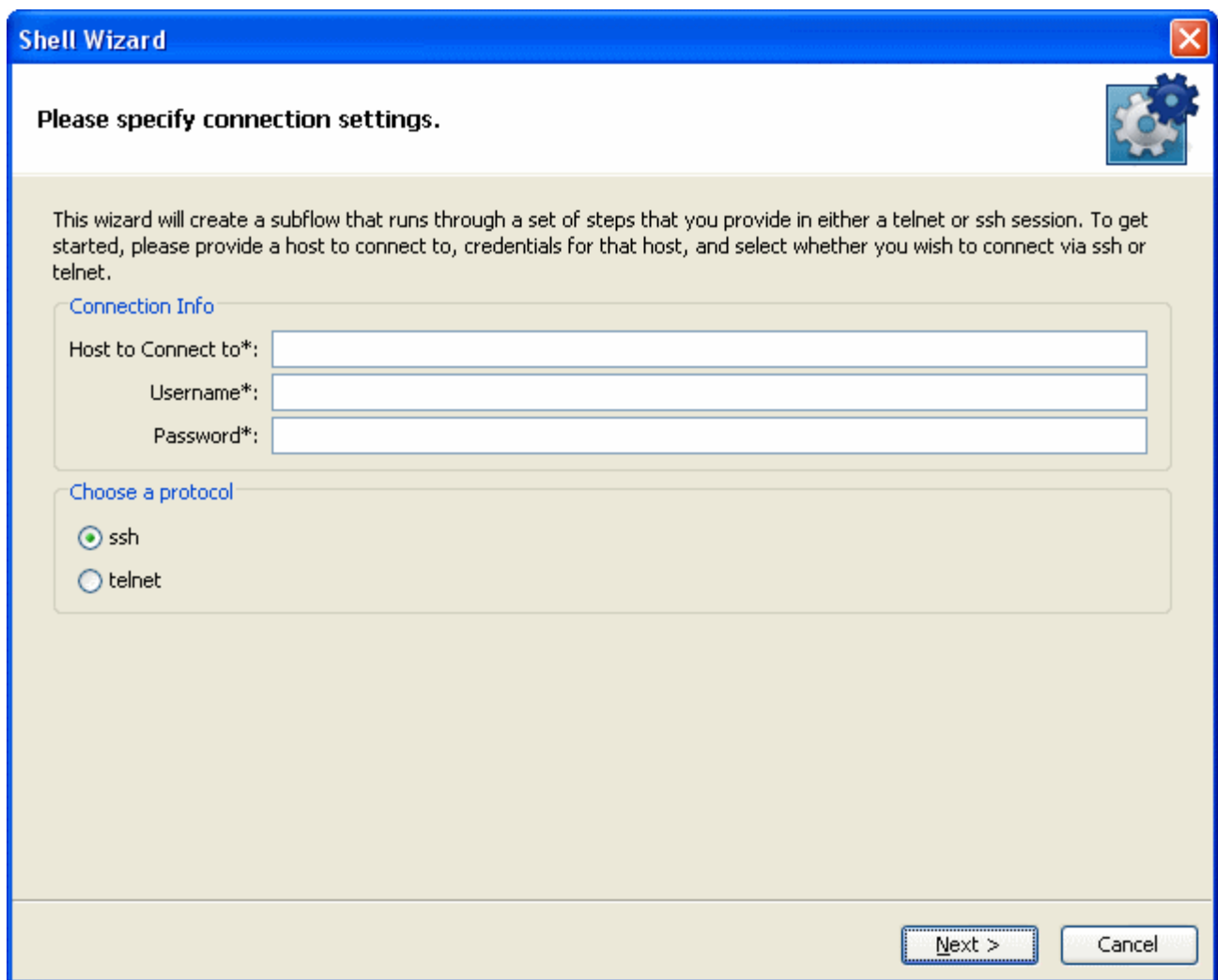


図 116 — Shell Wizard、接続の設定

6. [Connection Info] の下部にある [Host to Connect to] ボックスに、コマンドを実行するコンピューターの名前もしくは IP アドレスを入力します。
7. [Username] と [Password] ボックスに、アカウント名の資格情報を入力します。このアカウント名のもとコマンドを指定したホストで実行します。
8. [Choose a protocol] の下部で、コマンドを送るプロトコルに応じて [ssh] または [telnet] のどちらかを選択し、[Next] をクリックします。  
[ssh] の選択を推奨します。

コマンド送信に telnet プロトコルを指定する場合は次のものが表示されます。

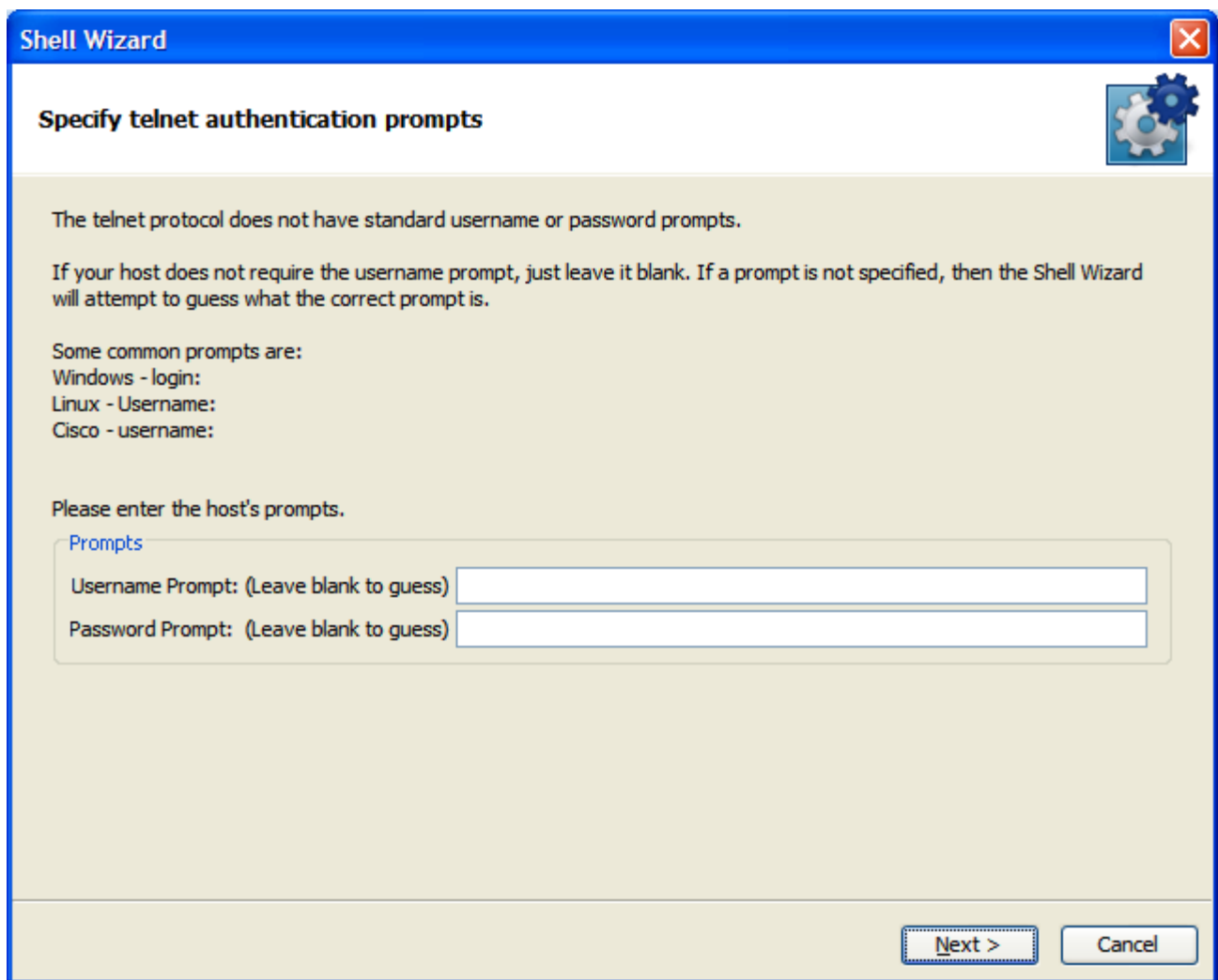


図 117 - 資格情報にホストプロンプトを指定

9. ホストマシンのユーザー名とパスワードのプロンプトがわかっている場合は、入力します。ウィザードの次のページで、ホストマシンのコマンドウィンドウが表示されます。

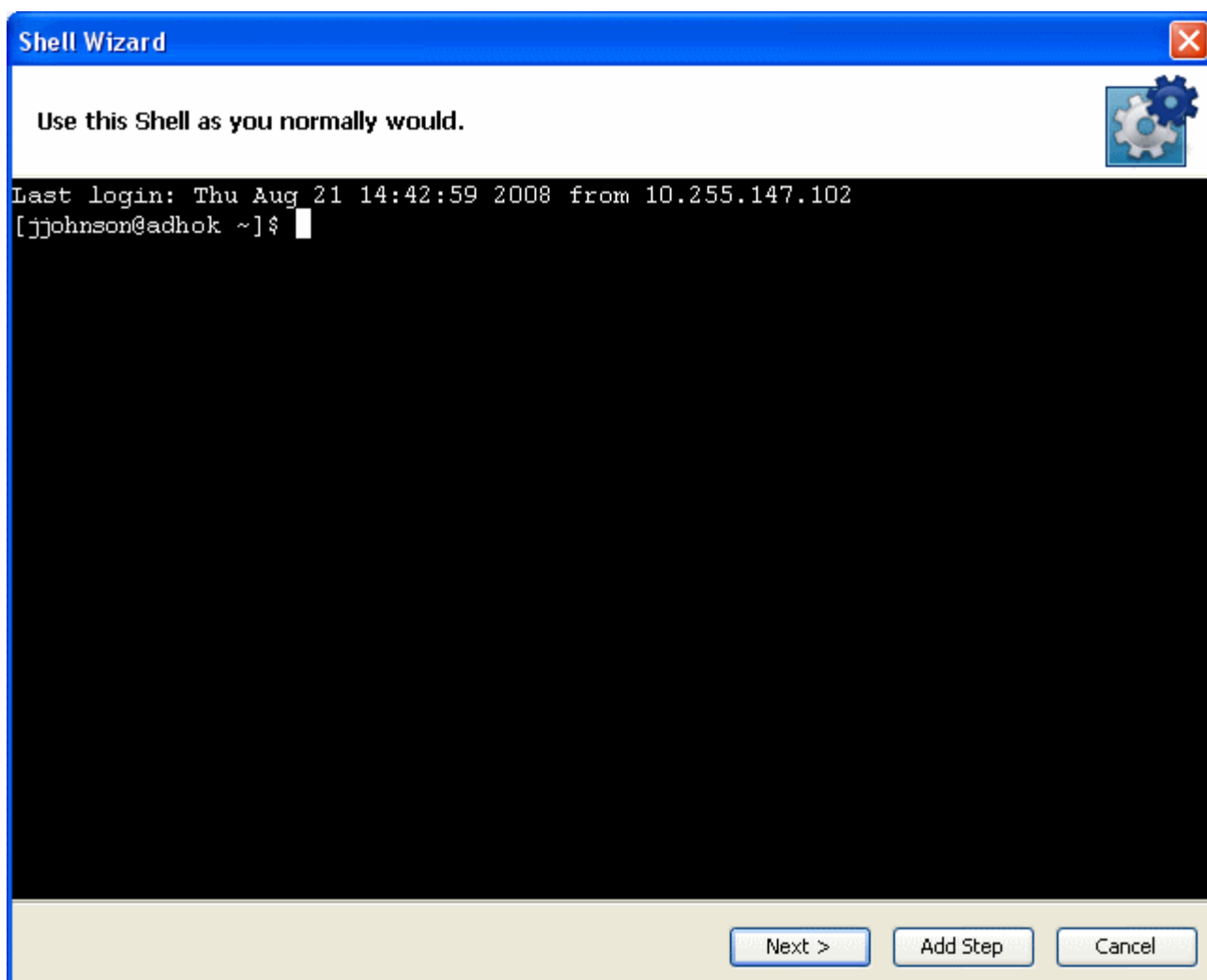
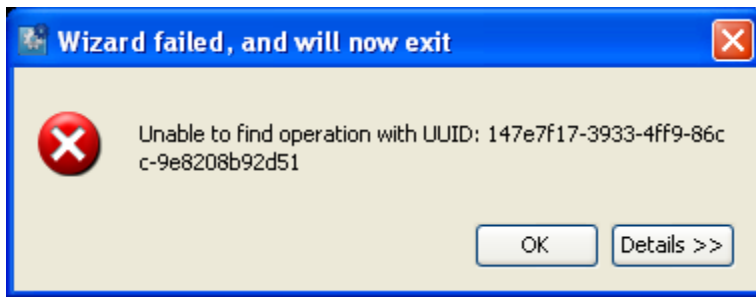


図 118 – ホストコンピューターのコマンドライン

10. 実行するコマンドまたはコマンドシーケンスを入力します。
11. オペレーション（およびオペレーションのインスタンスとなるステップ）を追加して、入力したコマンドシーケンスを実行するには、[Add Step] をクリックします。
12. Shell Wizard で作成するフローで別のステップを作成するには、別のコマンドまたはコマンドシーケンスを入力して、[Add Step] をクリックします。  
必要な数だけステップを追加できます。
13. ステップの追加が完了したら、[Next] をクリックします。  
ウィザードでオペレーションとステップを指定したフローに追加します。

次のエラーが表示された場合、その原因としては、オペレーションを既存のリポジトリに追加しようとしていることと、そのリポジトリに SSH Shell オペレーション（SSH プロトコル使用の場合）または Telnet Shell オペレーション（Telnet プロトコル使用の場合）が含まれていないことが挙げられます。



[OK] をクリックして、次のいずれかの操作を行います。

- デフォルトのパブリックリポジトリ (Studio の Library 全体、Library のサブフォルダーは不可) をこのウィザードで作成したリポジトリにエクスポートしてから、ウィザードを再起動する。
- ウィザードを再起動し、空のフォルダーまたは存在していないフォルダーを指定する。

14. [Wizard is Finished] のページで、[Finish] をクリックします。



**ベストプラクティス:** オペレーションを使用してフローを作成する他の作成者にわかりやすくするには、オペレーションの [Description] タブに次の情報を追加します (同じテクノロジーを用いて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 [Description] タブにはこの情報を記載します。HP OO コンテンツにデフォルトで設定されたベストプラクティスです)。[Description] タブへの情報の追加により、他の作成者や Central ユーザーが Generate Documentation 機能を利用できるようになることにご注意ください。Generate Documentation の詳細については、「[複数のオペレーションやフローの説明の表示](#)」を参照してください。

- 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。
- 実装を追加するときの**注意点**として、次のようなものがあります。
  - 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。フローを実行するために RAS が必要な場合は特に重要 (RAS オペレーションではフローの作成者やユーザーにこの情報が表示されない可能性があるため)。
  - その他の環境要件または使用要件

## シェルオペレーションの使用

シェルオペレーションを使用するには、次のどちらかを行います。

- OO デフォルトコンテンツのシェルオペレーションの 1 つからフローでステップを作成する。
- Shell Wizard で作成したフローを使う。

シェルオペレーションへの主結果は、SSH または Telnet のセッションの未加工出力 (文字ストリーム) です。シェルオペレーションで実行中のコマンドがメニュー駆動型の場合、視覚的に書式化されるか (表形式など)、印字不能文字が含まれるか、スクリーン以外で表示されます。視覚化した結果の取得は、結果の読み取りやフィルタリングに役立ちます。

視覚化した結果は、XML 形式で表示されます。表などの形式を保存し、その形式で画面に結果を表示します。次の手順は、作成したフローにシェルオペレーションを基にしたステップが格納されている場合を想定しています。

### シェルオペレーションの結果を視覚化するには

1. フローまたはサブフローで、シェルオペレーションで作成したステップの Inspector を開きます。
2. [Results] タブで新しい結果を追加し、[From] 列で [Field:visualized] を選択します。

## 正規表現での作業

正規表現 (regex と呼ばれる) という強力なツールを使用して作成できるものは次のとおりです。

- 結果のフィルター。結果データを絞り込みます。
  - 変数の保存。後のオペレーションで使用します。
  - ステップのレスポンスを判断するテスト
- 入力形式の検査

一例として、**Ping** オペレーションから入力変数が必要とする情報がパケット損失になる場合を説明します。IP アドレス 111.111.111.111 の ping 出力は次のように表示されています。

```
Pinging 111.111.111.111 with 32 bytes of data:
```

```
Reply from 111.111.111.111:bytes=32 time=27ms TTL=246
Reply from 111.111.111.111:bytes=32 time=26ms TTL=246
Reply from 111.111.111.111:bytes=32 time=29ms TTL=246
Reply from 111.111.111.111:bytes=32 time=28ms TTL=246
```

```
Ping statistics for 111.111.111.111:
```

```
    Packets:Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 26ms, Maximum = 29ms, Average = 27ms
```

パケット損失の数を抽出するには、OO で、“Lost = ”とこの後に続く任意の数の文字列を検索します。正規表現が役に立ちます。

正規表現を使うと、正確なテキストだけでなく、文字のクラスも検出します。ワイルドカードの `¥d` を使用して任意の桁数を検出することもできます。

したがって、前述の例で正規表現 `Lost = ¥d` を使用して検出を実行すると、文字列 **Lost = 0** が戻ってきます。

正規表現の重要なワイルドカードは次のとおりです。

ワイルドカード	用途
<code>^</code>	文字列の先頭に一致
<code>\$</code>	文字列の末尾に一致
<code>.</code>	改行以外の任意の文字
<code>\b</code>	単語境界

ワイルドカード	用途
\B	単語境界以外
\d	任意の数字 0～9
\D	数字以外
\n	改行
\r	キャリッジリターン
\s	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

修飾語句	結果
*	ゼロ (0) 以上の文字数に一致
+	1 以上の文字数に一致
?	ゼロ (0) または 1 つの文字数に一致
{n}	n 個の出現数に完全一致
{n,}	n 個以上の出現数に一致
{n,m}	n ～ m 個の出現数に一致
[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
a b	a または b に一致
\	特殊文字のエスケープ (たとえば \. の場合は、「.」。一致なし)

これらのワイルドカードや修飾語句を使用して、次の regex で Ping 出力から IP アドレスを抽出します。

```
\d{1-3}\.\d{1-3}\.\d{1-3}\.\d{1-3}
```



Studio で結果フィルターを使用する場合は、複数の regex を組み合わせて、必要な値を検出します。例を挙げて、Unix の ps コマンドの出力を考えてみます。

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	512	21604	21603	0	75	0	-	1096	wait	pts/1	00:00:00	Bash
0	R	512	2659	21604	0	76	0	-	1110	-	pts/1	00:00:00	Ps

次のフィルターを作成して、ps コマンドの時間を抽出できます。

1. Studio では、Unix の ps オペレーションを開きます。
2. [Output String Field Filters] ダイアログボックスを開きます。  
[Output String Field Filters] ダイアログボックスを開き、次のタスクを実行する手順は、「[フィルターの詳細](#)」を参照してください。  
ps の時間を抽出するには、regex が 2 つ必要です。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。
3. 正規表現のフィルターを新しく追加します。
4. 文字列に、`.*ps` を入力します（「ps」で終わる任意の文字列を検出する。必ず先頭にピリオド「`.`」をつけること）。次に、[Filter line by line] ボックスにチェックをつけます。
5. [Test Selected Filters] をクリックします。  
[Test Output] 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. この行から時間を抽出するには、正規表現のフィルターを新たに追加します。
7. 文字列に、`%d*:%d*:%d*`（コロンで区切られた 3 セットの数字）を入力します。
8. [Test Selected Filters] をクリックします。

テスト出力に ps 行の時間だけが表示されました。これで、この値を変数に割り当てることができます。

変数値の保存については、「[入力: オペレーションへのデータの提供](#)」または「[フロー変数: 再利用可能なデータの作成](#)」を参照してください。

## フローのテストと配布

フロー変更を安全に作成し配布するには

1. Central をインストールした開発用サーバーのローカルリポジトリまたはパブリックリポジトリで作業します。
2. ステージングサーバーで作業をテストします。
3. 非稼働時間中にフローの変更したバージョンを配布します。

### 既存のフローへの変更をテストサーバーから配布するには

1. 開発用サーバーで、変更内容を作成し、フローをテストします。
2. 変更内容を保存後、リポジトリとしてフローをエクスポートします。  
注: フローまたはオペレーションで保存されていない変更内容はエクスポートには含まれません。
3. フローをステージングサーバーにパブリッシュし、そこでテストします。

4. 現在実行中のフローがあるかどうか確認するには
  - Central の Web サイトを開き、HP OO の管理者の権限のあるアカウントでログインします。
  - [Administration] タブをクリックします。  
現在実行中のフローが、[Run Administration] の下に一覧表示されます。
5. 変更する予定のフローが現在実行中の場合は実行をキャンセルし、Central ユーザーがフローを新しく開始しないように指示します。
6. ステージングサーバーから本稼働のサーバーにパブリッシュします。
7. 本稼働のサーバーでフローをテストするには、自分以外のユーザーがフロー実行のアクセス権限を持たないようにプロパティを設定した後、テストを開始します。
8. フローテストが成功した後は、アクセス権限を適切なエンドユーザーに付与します。

## フロー使用の制限

次の方法で、フローを使用できるユーザーを制限できます。

- アクセス権限の制限。フローの閲覧および実行開始を許可するユーザーグループを指定します。  
OO オブジェクトのアクセス権限については、「[OO オブジェクトへのアクセス管理](#)」を参照してください。  
OO のロールの定義や、ここで説明した OO グループへの外部ロールまたは外部グループのマッピングについては、『[HP OO Administration Guide](#)』（AdminGuide.pdf）および Central のヘルプを参照してください。
- Central ユーザーにフローを非表示にする  
所属している組織がフローテスト用のステージングサーバーを使用していない場合に、Central での本稼働準備ができていないフローをパブリッシュするときは、Central ユーザーに対してフローを非表示にすることができます。  
詳細については、次のトピック「[Central でフローの閲覧可能なユーザーを制限](#)」を参照してください。
- 特定のトランジションを越えたフローアクセスを制限。トランジションのプロパティを設定して、トランジションを越えてフロー実行を続行できる OO ロールメンバーを指定します。  
フローの一部を **LEVEL\_ONE** の OO ロールは実行できるが、フローの後続部分には **LEVEL\_ONE** ユーザーには与えられていないユーザー資格情報が求められるため、段階的に実行する必要があるとします。フローの作成時にゲートトランジションを生成し、トランジションのステップで資格情報を求めることができます。ゲートトランジションでは、ユーザーがフロー実行を続けるには特定ロールのメンバーシップが要求されるようになります。  
ゲートトランジションの生成については、「[トランジション: ステップの接続](#)」を参照してください。
- **Acquire Lock** ステップをフローに追加。ロックのキーを保持するユーザー以外はそのポイントでフローの実行を続けられなくなります。  
**Acquire Lock** および **Release Lock** オペレーションは OO コンテンツにデフォルトで用意されています。使用については、各 [Description] タブを参照してください。

## Central でフローの閲覧可能なユーザーを制限

Central ユーザーからフローを隠す方法は 2 つあります。

- フローを非表示にマークして、ユーザーのアクセス権限やグループ機能にかかわらず、Central ユーザー全員にフローを表示しないようにします。
- フローをだれが見られて、だれが見られないかを指定するには、フローの読み取り権を与えるグループを指定します。

### フローを非表示にマークするには

1. フローを開いてから、フローの [Properties] シートを開き、作成ペインの下部にある [Properties] タブをクリックします。
2. [Advanced] タブで、[Hide in Central] を選択し、作業結果を保存します。

### 読み取り権限を持つグループを選択するには

1. Library で、フローを右クリックして [Permissions] をクリックします。
2. [Permissions] ダイアログボックスで、グループに読み取り権限を選択します。選択したグループは Central と Studio フローの閲覧ができるようになります。次に、[OK] をクリックします。

## オペレーション用の IAction の作成

高度なオペレーションの作成では、リモートアクションサービス (RAS) を用いて HP OO 外のオペレーションのコアを実行します。

RAS は、リモートアクションを管理するサービスです。RAS インスタンスをインストールすると、IAction インターフェースが提供され、RAS を実装します。RAS は、SOAP/HTTP(S) プロトコルを使用して、Central の Web アプリケーションおよび Studio と交信します。RAS は、IAction 実装のデータベースリポジトリを管理します。IAction は、Java 環境で使用する Java アーカイブ (.jar ファイル) と、.NET 環境で使用する動的リンクライブラリ (.dll ファイル) に格納されています。

IAction インターフェースでは、execute() メソッドと、必要な操作を実行する Web アプリケーション、スタンドアロンアプリケーション、プラットフォームまたは拡張サービスに固有のメソッドを使用します。これらのメソッドは、OO SDK の操作をターゲットシステムまたはアプリケーションの SDK の操作にマップします。IAction インターフェースは前述のように OO と OO の外部システムとを調整します。

標準オペレーションは、OO インフラストラクチャ (データベースおよび Internet Information Services) と相互作用し、OO インフラストラクチャ内で使用される操作は限定されています。IAction インターフェースをオペレーションでプログラムするときに、HP の外部のエンティティと (拡張サービスを介して) 相互作用できます。したがって、次のオペレーションが作成できます。

- ネットワークまたはインターネット上でのシステムの相互作用
- IAction を介して RAS が相互作用可能なエンティティと OO を統合

Web 拡張機能を使用してオペレーションをプログラミングするには、.NET または Java のプログラミングスキルが必要です。

## IAction を実装するオペレーションの作成概要

全般的に、作成のプロセスでは、1 個以上の IAction 実装クラスの作成と Studio への移動が組み込まれます。

1. カスタム IAction 実装クラスを開発環境で作成し、動的リンクライブラリ (.dll) または Java アーカイブ (.jar) ファイルにコンパイルします。
2. .dll または .jar を自分の Web サービス (Web サーバーの ¥bin¥Actions ディレクトリ) にコピーします。
3. Web サービスを Studio にインポートします。

IAction の作成と実装方法については、OO の『*SDK Guide*』 (SDKGuide.pdf) の IAction の作成の項を参照してください。

## トラブルシューティング

**同じパブリックリポジトリに接続した Studio の 2 つのインスタンスにロギンしています。Studio の一方のインスタンスの作業スペースで変更した内容がもう一方の Studio に反映されません。**



**鍵となる情報:** どのユーザーアカウントも、Studio の 2 つの異なるインスタンスから Central の同一インストールのパブリックリポジトリに接続していないことを確認する必要があります。その場合、リポジトリ内にあるユーザーの作業スペースが壊れることがあります。また、Central がクラスタ化されたときに起こることもあります。

**存在していることがわかっているフロー、オペレーション、またはシステムオブジェクトが、自分の Library に表示されません。**

それらのアイテムの読み取り権が与えられていない場合があります。オブジェクトの読み取り権がない場合、リポジトリペインでは表示されません。

**オペレーションのスクリプトレットが正常に実行されません。**

デフォルトでは、Central は HP OO のホームディレクトリ (¥Central¥scripts¥サブディレクトリ) 内のスクリプトを探すように設定されています。このデフォルトの場所は Central.properties ファイルで指定されています。

オペレーションが使用するスクリプトが、¥Central¥scripts¥サブディレクトリに配置されていない場合は、次のいずれかを実行します。

1. スクリプトを ¥scripts¥サブディレクトリに移動させます。
2. Central.properties ファイルで、次の行を検索します。  
`dharmapp.script.repository=${iconclude.home}/Central/scripts`
3. セミコロンの区切り文字と使用する任意のスクリプトの場所をこの行の末尾に付加します。  
たとえば、**c:¥MyScripts** ディレクトリに配置されているスクリプトを実行するには、行を次のように編集します。

```
dharmapp.script.repository=C:\Program Files\Hewlett-Packard\Operations  
Orchestration\Central\scripts;c:\MyScripts
```

## パブリックリポジトリのロックを損失したというエラーがレポートされます。

リポジトリをパブリックリポジトリにインポートしようとしている場合にこのエラーが表示されることがあります。考えられる原因の1つに、Studioの起動中にCentralを再起動した可能性があります。たとえば、コンピューターに省電力スキームが備わっていて、しばらく休止時間が続いて休止状態になると、このエラーが起きることがあります。

パブリックリポジトリのロックを再取得するには、Studioで、プライベートリポジトリを開いてからパブリックリポジトリを開きます。

## ステップを削除した後に削除を元に戻しましたが、変更内容が保存されていません。

ステップを複製するには

1. フロー図を開き、ステップを削除します。
2. 作業結果を保存します。
3. 削除を元に戻します。
4. 削除したステップを変更します。
5. 作業結果を保存します。

**予想していた結果:** 削除を元に戻した後ステップに加えた変更は保存されます。

**実際の結果:** 削除を元に戻した後にステップに加えた変更は保存されません。

この後の作業は正常に保存されます。

**回避策:** 削除、編集、保存を元に戻した後に、ステップを再移動して保存します。この後の変更は正常に保存されます。

# インデックス

- .vm テンプレートファイル, 48
- Accelerator Pack
  - インポート, 140
- [Bookmarks] ペイン
  - キーボードショートカット, 23
- [Bookmarks] ペイン、概要, 20
- Central
  - RAS に依存するオペレーション, 149
- Cmd オペレーション, 173
- Configuration フォルダのオブジェクト
  - 削除したオブジェクトの復元, 57
- DescribeFlows.vm
  - テンプレートファイルの構成, 50
- Generate documentation
  - カスタムテンプレートの編集, 51
- Generate Documentation
  - オペレーション, 41, 45
  - カスタム, 47
  - コマンド, 41, 47
- How do I... フォルダ, 1
- HP OO オブジェクト
  - 権限、設定, 169
- Http オペレーション
  - プロパティ、レスポンス定義, 176
- IAction
  - オペレーション用に作成, 194
- [Icons] ペイン、概要, 21
- Inspector
  - キーボードショートカット, 23
  - 開く, 63
- [My Changes/Checkouts] ペイン
  - 概要, 16
- OO オブジェクト
  - アクセス管理, 167
  - 権限, 167
- Ops フロー。「フロー」を参照
- Perl スクリプトオペレーション
  - プロパティ、レスポンス定義, 177
- RAS
  - オペレーションの作成, 155
  - オペレーションの作成, 156
  - オペレーションの追加, 156
  - 使用, 147
  - 使用できるかどうか、確認, 149
  - 既存の追加, 150
  - 要件, 147
- RAS オペレーション
  - プロパティ、レスポンス定義, 178
- RAS に依存するオペレーション
  - トラブルシューティング, 154
- RAS 参照
  - 上書き, 152
  - 既存の RAS からの設定, 155
  - 再設定, 151
  - 削除, 157
  - 追加, 150
  - 変更, 152
- Regex。「正規表現」を参照
- Secure シェルオペレーション
  - プロパティ、レスポンス定義, 181
- Shell Wizard
  - フローの作成に使用, 184
- Studio。「Studio」を参照
  - キーボードショートカット, 22
  - 起動, 12
  - 初心者向け機能, 1
  - 図解, 13
  - ライブラリ, 14
- Telnet オペレーション
  - プロパティ、レスポンス定義, 183
- Web サービス
  - オペレーションの作成, 145
- Web サービスコンテンツ
  - 使用, 157
- アクセス権限
  - グループのデフォルト設定, 171
  - グループマスクの作成, 171
- アクセス管理
  - オーサリングオブジェクト, 167
- エバリュエーター
  - 削除, 159
  - 作成, 158
  - 編集, 159
- オーサリングオブジェクト
  - アクセス管理, 167
- オペレーション
  - [Properties] シート, 172
  - IAction で作成、概要, 195

- IAction を作成, 194
- RAS からの作成, 155, 156
- Web サービス, 154
- Web サービスからの作成, 145
- オプション入力, 173
- クイックビュー, 4
- 検索, 37
- コア機能, 8
- 構成, 8
- 高度な作成, 6
- 高度な概念, 10
- コピー, 54
- 削除したオペレーションの復元, 57
- 作成, 171
- 参照先, 53
- 参照元, 53
- シール, 37
- シール、定義, 9
- 出力, 100
- 出力、フィルタリング, 104
- 出力、削除, 102
- 出力、追加, 102
- 出力データ, 8
- 詳細情報の表示, 53
- 情報フロー, 8
- スクリプトレット, 8
- スクリプトレット、作成, 96
- ステップとの比較, 10
- 説明, 39
- チェックアウト, 24, 25
- チェックイン, 24, 25
- 定義, 8
- 特定のオペレーションを使用するフローの検索, 53
- 入力, 11
- 複数の表示, 41
- ブックマーク, 55
- プロパティ, 173
- 未加工結果, 8
- レスポンス, 8, 100, 101
- レスポンス定義, 173
- オペレーションタイプ, 173
  - Cmd、またはコマンドライン, 173
  - シエル, 184
  - フロー実行サマリレポート, 175
- オペレーションの参照
  - 検索, 53
- オペレーションの説明
  - 表示, 41
- オペレーションレスポンス
  - ルール, 113
- オペレーション出力
  - フィルタリング, 104
- 開始ステップ
  - 変更, 62
- カテゴリ, 163
- 管理者
  - 強制チェックイン, 29
- 概念
  - 高度, 10
- キーボード
  - ショートカット, 22
- キーボードショートカット
  - [Bookmarks] ペイン, 23
  - Inspector, 23
  - 作成キャンバス, 23
  - 作成ペイン, 23
  - スクリプトレットペイン, 23
  - デバッガー, 23
  - プロパティエディター, 23
  - リポジトリペイン, 22
- 競合
  - 解決, 28
- クイックビュー
  - オペレーション, 4
  - 入力、ダッシュボードでのレポート作成用の記録, 160
- グループ
  - アクセス権限のデフォルト設定, 171
- グループの権限, 167
- グループマスク
  - 作成, 171
- グローバルコンテキスト, 8
- 結果
  - 削除, 103
  - 説明, 11
  - ソースの変更, 104
  - 追加, 103
  - フィルタリング, 106
- 結果、ステップ
  - 定義, 101
- 権限
  - HP OO オブジェクトへの設定, 169
  - システムオブジェクトへの設定, 169
  - 設定, 167
- 構成アイテム
  - 定義, 161
- 構成オブジェクト
  - チェックアウト, 24, 25
  - チェックイン, 24, 25



- コールアウト
  - フローへの追加, 36
  - パレットから追加, 17
- コマンドラインオペレーション, 173
- コンテキスト
  - グローバル、定義, 8
  - 定義, 8
  - ローカル、定義, 8
- ご注意, ii
  - 商標, ii
  - 制限付き権利, ii
  - 著作権, ii
  - 保証, ii
- 削除
  - 復元, 57
- 作成キャンバス
  - キーボードショートカット, 23
- 作成ペイン
  - キーボードショートカット, 23
- 作成ペイン、概要, 16
- 作成者、複数. 「複数の作成者」を参照
- サブフロー
  - 使用, 118
- サンプルフロー
  - How do I... フォルダー, 1
- シール
  - 定義, 37
- シェルオペレーション, 184
  - Shell Wizard, 184
  - 使用, 189
  - 未加工結果の視覚化, 189
- システムアカウント, 165
  - 作成, 165
  - 削除, 166
  - 編集, 166
- システムエバリュエーター, 158
- システムオブジェクト
  - 権限、設定, 169
- システムプロパティ
  - 作成, 117
  - 使用, 117
- 出力
  - ソースの変更, 102
  - 削除, 102
  - 説明, 11
  - 追加, 102
- 出力データ, 8
- ショートカット
  - キーボード, 22
- 商標, ii
- 初心者向け機能, 1
- スクリプトレット, 96
  - 再使用のための保存, 99
  - 作成, 96, 97
  - 定義, 8
  - デバッグ, 98
  - 並列分岐ステップ, 85
- スクリプトレットオペレーション
  - プロパティ、レスポンス定義, 180
- スクリプトレットペイン
  - キーボードショートカット, 23
- スケジュール
  - 入力, 77
- ステップ
  - アイコンの変更, 116
  - 移動, 63
  - オペレーションとの比較, 10
  - オペレーションの変更, 120
  - 結果, 100, 101
  - 結果、削除, 103
  - 結果、追加, 103
  - 高度な概念, 10
  - コピー, 62
  - 作成, 59, 61
  - 実行, 7
  - 接続, 64
  - 説明, 121
  - 単一レスポンス, 94
  - 追加, 59
  - トランジション. 「トランジション」を参照
  - ノンブロッキング, 94
  - ノンブロッキング、作成, 94
  - 入力, 11
  - フローへの追加, 61
  - フロー間でのコピー, 62
  - 編集, 63
  - ユーザープロンプト, 63
- ステップアイコン
- ステップ、並列分岐. 「並列分岐ステップ」を参照
- ステップ、マルチインスタンス. 「マルチインスタンスステップ」を参照
- 変更, 116
- 正規表現, 190
- 制限付き権利, ii
- 説明, 121
- 説明、オペレーション
  - 表示, 41

- 説明、フロー
  - 表示, 41
- セルフドキュメンテーション
- 選択リスト, 164
  - フロー, 41, 45
- ターゲットリポジトリ
  - 設定, 132
- 単一レスポンスのステップ
  - 作成, 94
- ダッシュボードでのレポート作成
  - クイックビュー, 160
- チェックアウト, 12
  - 取り消し, 30
- チェックイン, 12, 27
  - 競合の解決, 28
  - 強制, 29
  - 変更の破棄, 30
- チェックポイント, 95
- 著作権, ii
- ツールバー
  - オプション表示パレット, 18
  - 概要, 16
  - ステップおよびコールアウトのパレット, 17
  - パレット, 16
- テンプレート
  - フローの作成, 35
- テンプレートファイル, 50
  - 構成, 50
- データ
  - 引き渡し, 119
- データ移動, 119
- デバッガー
  - 概要, 123
  - キーボードショートカット, 23
  - フローのリセット, 123
  - フローの再開, 123
  - フロー変数の値の変更, 126
- トランジション, 64
  - 移動, 67
  - 作成, 64
  - 追加, 66
- ドメインターム
  - 作成, 161
  - 値、追加, 162
  - 追加, 162
  - 定義, 161
  - 編集, 162
- 同時実行, 83
- ノンブロッキングステップ, 83, 94
  - 作成, 94
- バージョン
  - 管理, 30
- バージョン、前
  - 開く, 31
  - 復元, 32
- バージョン管理, 12, 24
- パラレル処理. 「並列処理」を参照
- パレット
  - オプション表示, 18
  - ステップおよびコールアウト, 17
- フィルター, 104
  - 再使用, 113
  - 作成, 104
  - 保存, 106, 113
- フォルダー
  - 作成, 32
- 複数ドメイン
  - フローの実行, 154
- 複数の作成者
  - バージョン管理, 12
- 「フロー」. 「オペレーション」を参照
- フロー. を参照 「フロー」
  - Central で非表示, 193
  - インポート, 140
  - カテゴリ, 163
  - 概要, 1
  - 検索, 37
  - ゲートトランジション, 193
  - コールアウトの追加, 36
  - 高度な概念, 10
  - コピー, 54
  - 削除したフローの復元, 57
  - 作成, 4, 33, 34
  - サブフロー, 119
  - シール, 37
  - シール、定義, 9
  - 使用制限, 193
  - 出力, 100
  - 自動実行, 122
  - ステップ. 「ステップ」を参照
  - ステップ 対比 オペレーション, 60
  - ステップ 対比 サブフロー, 60
  - ステップの作成, 59
  - ステップの追加, 59
  - 説明, 39
  - セルフドキュメンテーション, 41, 45
  - チェックアウト, 24, 25
  - チェックイン, 24, 25
  - チェックポイント, 95

- データの提供, 68
- テスト, 192
- デバッグ, 122
- テンプレートから作成, 35
- トランジション. 「トランジション」を参照
- 入力, 11, 68
- 複数ドメイン、実行, 154
- ブックマーク, 55
- 変更の配布, 192
- 要件, 3
- リターン ステップ、追加, 68
- リポジトリ間の移動, 131
- レスポンス、追加, 116
- フロー、構成要素, 2
- フロー、作成
  - 概要, 33
- フローデザイン
  - 単純化, 118
  - 注意事項, 118
- フローの オペレーション
  - 高度 作成, 6
- フローの 出力 フィールド, 119
- フローの説明
  - 表示, 41
- フロー変数, 78
  - グローバル, 10
  - 作成, 78
  - デバッガーでの値の変更, 126
  - 範囲, 10
  - 予約済み, 79
  - ローカル, 10
  - ローカル、作成, 79
- フロー実行サマリレポートオペレーション, 175
- フロー状態データ
  - 定義, 95
- ブックマーク, 55
  - シェルフ、移動, 57
  - シェルフ、追加と削除, 56
  - シェルフ、名前の変更, 56
  - シェルフ、表示, 57
  - シェルフ、非表示, 57
  - フローおよびオペレーション、追加と削除, 56
- ブレークポイント
  - 無効化, 127
  - 有効化, 127
- プロパティエディター
  - キーボードショートカット, 23
- 並列処理
  - マルチインスタンスステップ, 88
- 並列実行, 83
- 並列分岐ステップ, 83, 84
  - 移動, 86
  - 開始ステップの変更, 88
  - コピー, 86
  - 作成, 85
  - スクリプトレット, 85
  - データの移動, 85
  - デバッグ, 88
  - パレットから追加, 17
  - レーンのサイズ変更, 87
  - レーンの削除, 87
  - レーンの順序, 85
  - レーンの追加, 87
  - レーンの名前変更, 88
  - レーンの複製, 87
- 並列分岐ステップのレーンステップ
  - 移動, 86
- 並列分岐ステップレーン
  - 移動, 86
- 変数. 「フロー変数」を参照
- 変更
  - 破棄, 30
- 保証, ii
- マルチインスタンスステップ, 83, 88
  - 作成, 91
  - 制限, 93
  - データの移動, 92
  - デバッグ, 94
  - フローデザイン, 89
- 未加工結果
  - 定義, 8
- メッセージ
  - ユーザーへの表示, 162
- 文字列形式. 「エバリュエーター」を参照
- ユーザー
  - 切り替え, 129
- ユーザープロンプト
  - 選択リストの対象, 164
- ユーザープロンプトの入力
  - リストの作成, 75
- ユーザーメッセージ
  - 表示, 162
- ライブラリ、概要, 14
- ライブラリオブジェクト
  - チェックアウト, 26
  - チェックイン, 27
  - 復元, 57

- 前のバージョン、復元, 32
- 前のバージョン、開く, 31
- リスト
  - ユーザープロンプト用, 75
- リターンステップ
  - 追加, 68
  - 定義, 68
  - パレットから追加, 17
  - レスポンスの変更, 68
- リポジトリ, 129
  - 暗号化, 142
  - 暗号化されたコピー, 143
  - インポート, 140
  - エクスポート, 138
  - 検査, 142
  - 更新, 135
  - 更新、元に戻す, 138
  - 更新のプレビュー, 135
  - ソース, 131
  - ターゲット, 131
  - ターゲットリポジトリの設定, 132
  - 追加, 130
  - バックアップ, 144
  - バックアップ、インポート, 138
  - パブリック, 129
  - パブリッシュ, 133
  - パブリッシュ、元に戻す, 138
  - パブリッシュのプレビュー, 133
  - 開く, 130
  - 復元, 144
  - 復号化, 143
  - プライベート, 129
- リポジトリ、暗号化
  - 開く, 143
- リポジトリ、パブリック
  - 更新, 132
  - パブリッシュ, 132
- リポジトリペイン
  - キーボードショートカット, 22
- リモートアクションサービス。「RAS」を参照、「RAS」を参照
- レーン
  - 移動, 86
  - 開始ステップの変更, 88
  - コピー, 86
  - サイズ変更, 87
  - 削除, 87
  - 追加, 87
  - 名前変更, 88
  - 複製, 87
- レーンステップ
  - 移動, 86
  - コピー, 86
- レスポンス
  - 定義, 8, 101
  - デバッガーでの上書き, 128
- ローカルコンテキスト, 8
- ローカルフロー変数
  - 作成, 79
- ロゲイン
  - 切り替え, 129