HP Operations Orchestration Software Studio

ソフトウェアバージョン: 9.00

Operations Orchestration フロー作成の手引き

ドキュメントリリース: 2010 年 6 月

ソフトウェアリリース: 2010 年 6 月

ご注意

保証

HP 製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載で追加保証を意図するものは一切ありません。ここに含まれる技術的、編集上の誤り、または欠如について、HP はいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

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

著作権

© Copyright 2005-2010 Hewlett-Packard Development Company, L.P.

商標

オープンソースおよび第三者のソフトウェアの承認については、本リリースに関する文書一式内の『オープンソースおよび第三者のソフトウェアの承認 (3rdPartyOpenNotices.pdf)』を参照してください。

インターネット上にある OO のサポートとドキュメント

2 つのサイトで、OO ヘルプシステム、ガイド、チュートリアルの更新を含むサポートおよびドキュメントを入手することができます。

- OO サポートサイト
- BSA Essentials Network

サポート

ヒューレット・パッカード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 Passport 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 Passport のユーザー名とパスワードを入力してログインします。

   または

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

   HP Passport 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。

BSA Essentials Network

パッチ、トラブルシューティング支援、サポート契約の管理、製品マニュアル、その他のサポート情報に関しては、次のサイトを参照してください。 *HTTP://www.hp.com/go/bsaessentialsnetwork*

これは、**BSA Essentials Network** の Web ページです。サインインするには、以下の手順に従ってください。

1. **[Login Now]** をクリックします。
2. **[HP Passport sign-in]** ページで、HP Passport のユーザー ID とパスワードを入力し、**[Sign-in]** をクリックします。
3. HP Passport アカウントをお持ちでない場合は、以下の手順に従ってください。
   a. **[HP Passport sign-in]** ページで、**[New user registration]** をクリックします。
   b. **[HP Passport new user registration]** ページで、必要な情報を入力して **[Continue]** をクリックします。
   c. 確認ページが開きますので、入力した情報を確認し、**[Register]** をクリックします。

d. **[Terms of Service]** ページの利用規約および法的な制限事項を読み、**[Agree]** ボタンを選択し、**[Submit]** をクリックします。

4. **[BSA Essentials Network]** ページで、**[Operations Orchestration Community]** をクリックします。

   [The Operations Orchestration Community] ページには、お知らせ、ディスカッション、ダウンロード、ドキュメント、ヘルプ、サポートへのリンクがあります。

注: この手順にご不明な点がございましたら、OO 窓口にお問い合わせください。

OO 内のヘルプ、PDF、チュートリアル

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。

  HP OO のオペレーション、フローおよび Accelerator Pack 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。

目次

ご注意 ii

保証 ii

権利の制限 ii

著作権 ii

商標 ii

インターネット上にある OO のサポートとドキュメント iii

サポート iii

BSA Essentials Network iii

OO 内のヘルプ、PDF、チュートリアル iv

目次 v

このヘルプシステムおよびガイド 1

Studio 初心者向け： 操作方法フォルダー 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 6

オペレーション: アーキテクチャとデータの移動 7

フロー、ステップ、オペレーションの高度な概念 9
フロー変数とグローバル変数 9
ステップとオペレーション 9
フロー、ステップ、オペレーションの入力 10
出力と結果 10
複数の作成者とバージョン管理 11
HP OO Studio の起動 11
図解による Studio 概要 12
リポジトリペイン 13
[自分の変更 / チェックアウト] パネル 14
作成ペイン 15
作成ペインのツールバー 15
[ブックマーク] パネル 18
[アイコン] パネル 19
[フロー変数] パネル - 概要 21
パネルのサイズ変更および再配置 21
オブジェクトのプロパティシート 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 23
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 /インスペクターのキーボードショートカット 23
[スクリプトレット] タブのキーボードショートカット 23
[ブックマーク] パネル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 31
オブジェクトのバージョンの比較 32

フォルダーの作成 32

フローの作成 33

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

フローまたはオペレーションの検索 37

フローまたはオペレーションの検索方法 38
検索の定義 38
Lucene 構文を使用した検索 41
[検索] パネルからのオペレーションへのアクセス 42
説明：オペレーションを有効に利用するための情報 42
フォルダーの説明の表示 43
フローの説明の表示 43
オペレーションの説明の表示 44

複数のオペレーションやフローの説明の表示 44

Central ユーザーによるドキュメントの生成の有効化 49
カスタム形式のドキュメントの生成 51
.vm テンプレートファイルの説明 52
ドキュメントの生成の出力構成 53
ドキュメントの生成テンプレートの編集 54

特定のオペレーションを使用するフローの検索 55

フローおよびオペレーションのコピー 56

フローまたはオペレーションの名前の変更 57

フローおよびオペレーションのブックマーク 57

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 58
ブックマークのシェルフ： 追加、削除、名前の変更 58

ブックマークのシェルフ： 表示、非表示、移動 59

削除したライブラリオブジェクトの復元 59

ステップの作成 60

フローへのステップの追加 61

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 62

開始ステップの変更 63

ステップのコピー 63

ステップの編集 63

フロー図内のステップの位置変更 64

ステップ実行前に表示するユーザープロンプトの作成 64

トランジション： ステップの接続 65

トランジションの追加 67

トランジションの位置の変更 69

リターンステップ 70

リターンステップの追加 70

リターンステップのレスポンスの変更 71

入力： オペレーションへのデータの提供 72

入力の作成 72

入力の値の取得方法の指定 74

単一の固定入力値の作成 76

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 76

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 77

固定入力値リストの作成 77

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 78

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 79

前のステップの結果を基にした入力値の作成 79

入力値への資格情報の割り当て 80

入力形式の検証 81

入力とフロー実行のスケジュール 81

入力の削除 82

フロー変数： 再利用可能なデータの作成 82
フロー変数の作成と値の割り当て 83
フロー変数の管理 84
[フロー変数] パネル 84
[フロー変数] パネルでのフィルタリング 86
フロー図内のフロー変数の用途を特定する 87
グローバル変数 87
グローバル変数の値の取得または変更 88
同時実行： 複数スレッドの並列実行 88
並列分岐ステップ 89
レーンの順序：開始と終了 90
並列分岐ステップの作成 90
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 92
レーンの追加 93
レーンの複製 93
レーンの削除 93
レーンのサイズ変更 93
レーンの名前変更 93
レーンの開始ステップの変更 93
並列分岐ステップとのデータの移動 94
並列分岐ステップのデバッグ 94
マルチインスタンスステップ 94
フロー設計でのマルチインスタンスステップの使用 95
マルチインスタンスステップの作成 98
マルチインスタンスステップとのデータの移動 99
リスト入力、フロー変数、マルチインスタンスステップ 99
マルチインスタンスステップの制限 100
フロー内のマルチインスタンスステップのデバッグ 100
ステップのノンブロッキング化 100

ステップのレスポンスの単一化 101

チェックポイント： リカバリ用のフロー実行進捗の保存 101

ステップの出力の保存 102

スクリプトレット 103

スクリプトレットの作成 104
スクリプトレットでの結果のフィルタリング 104
オペレーションレスポンス用のスクリプトレットルールの作成 104
スクリプトレットのデバッグ 105
ほかの用途を目的としたスクリプトレットの保存 107
Configuration フォルダーからのスクリプトレットの使用 107

出力、レスポンス、およびステップの結果 107

オペレーション出力 108

オペレーションのプライマリ出力の指定 108
セカンダリ出力の追加 109
出力の削除 109
出力のソースの変更 109

レスポンス 109

ステップの結果 109

ステップの結果の追加 110
ステップの結果の削除 111

結果のソースの変更 111

出力および結果のフィルタリング 111

フィルターの作成 112

フィルターの詳細 113

大文字と小文字の変換 114
数値の抽出 114
形式 114
行カウント 115
正規表現 115
重複行の削除 115
置換 115
数値の丸め 116
スクリプトレット 116

行の選択 117
範囲の選択 117
ソート 118
削除 118
スペースの削除 118
テーブル 119
XML フィルター 120
フィルターの保存と再使用 125

レスポンス： 結果の評価 125
フローへのレスポンスの追加 128
フローからのレスポンスの削除 128

ステップのアイコンの変更 128

システムプロパティ 129
システムプロパティの作成 130
システムプロパティの値の取得 130
システムプロパティの値の変更 131

フロー設計 131
サブフローを使用したフロー設計の単純化 131

サブフローから親フローへのデータの引き渡し 132

ステップのベースとなるオペレーションの変更 133

ステップの説明 134

フローの自動実行 134

フローのデバッグ 135
フローのデバッグ 135
サブフローへのステップイン、またはサブフローからのステップアウト 140
デバッガー内でのフローのリセットおよび再開 141
デバッガーでのフロー変数の値の変更 141
ブレークポイントの使用 142

デバッグ実行でのレスポンスのオーバーライド 143

別のユーザーとしてログオン 144

リポジトリ： フローおよびオブジェクト用のライブラリ 144

リポジトリの追加 145
作成用にリポジトリを開く 146
リポジトリ間のフロー要素の移動 146
ターゲットリポジトリの設定 147
パブリックリポジトリからのパブリッシュと更新 147
リポジトリのパブリッシュ：方法 148
リポジトリの更新：方法 151
パブリッシュまたは更新のロールバック 154
リポジトリのエクスポート 154
リポジトリのインポート 156
リポジトリの検査 158
リポジトリの暗号化 158
リポジトリの暗号化 158
暗号化されたリポジトリを開く 159
リポジトリの復号化 159
再暗号化されたリポジトリのコピーの作成 159
リポジトリのバックアップと復元 160
リポジトリスナップショット 160
スナップショットの作成 161
スナップショットの名前や説明の変更 163
スナップショットの表示 163
スナップショットを閉じる 164
スナップショットのリストのフィルタリング 164
スナップショットの削除 166
スナップショットからの復元 166

Web サービスからのオペレーションの作成 168

Web サービスウィザードのログ記録 171

リモートアクションサービスを使用した Central 外でのオペレーション 171

RAS に依存するオペレーションの Central での実行方法 172
RAS を使用できるかどうかの確認 173

RAS へのアクセス制御 173
既存の RAS から RAS 参照を追加するには 173
RAS 参照の追加 174
既存の RAS 参照の再設定 175
RAS 参照の削除 176
RAS 参照の変更 177
Web サービスにアクセスするオペレーションの作成 178
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 178
Studio の例：任意の複数ドメインで実行するフロー 179
RAS からのオペレーションの作成 179
RAS からのオペレーションの作成 180
デフォルトの RAS にないオペレーションの追加 181
RAS コンテンツのインポート 182

データが正しい形式かどうかの評価 182
エバリュエーターの作成 183
エバリュエーターの編集 184
エバリュエーターの削除 184

レポート作成用の値の記録 184

ダッシュボードグラフ用のドメインターム 185
ドメインタームの追加、名前の変更、または削除 186
ドメインターム値の追加、変更、および削除 187

ユーザーへのメッセージ表示 188

カテゴリ： フローの分類 188

ユーザープロンプトの選択リスト 189

システムアカウント: 安全な資格情報 190
システムアカウントの作成 191
システムアカウントの編集 191
システムアカウントの削除 191

OO オブジェクトへのアクセス管理 192
フォルダーと OO オブジェクトの権限の設定 193
グループのアクセス権限のデフォルト設定 195

新しいオペレーションを作成するには 196
オペレーションタイプ: プロパティの設定 198
Cmd（コマンドライン）オペレーション 198
フローの実行カウンターオペレーション 199
フローの実行概要レポートオペレーション 200
HTTP オペレーション 201
Perl スクリプトオペレーション 202
RAS オペレーション: RAS のための IAction プログラミング 203
スクリプトレットオペレーション 205
セキュアなシェル（ssh）オペレーション 206
Telnet オペレーション 207
シェルオペレーション 208
Shell Wizard： シェルオペレーションを使用するフローの作成 209
シェルオペレーションの使用 214

正規表現での作業 215

フローのテストと配布 217

フロー使用の制限 217
Central でフローの閲覧可能なユーザーを制限 218

オペレーション用の IAction の作成 218
IAction を実装するオペレーションの作成概要 219

作成者へのメッセージの表示 219

トラブルシューティング 219
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。 219
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。 220
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。 220
パブリックリポジトリのロックが失われたというエラーがレポートされます。 220

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。 220

インデックス 222

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 アクセラレーターパック 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： 操作方法フォルダー

必要最小限の知識を習得して実用的なフローを作成し、実行したい場合は、**[操作方法]** 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、**[説明]** 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

フローは、意思決定ロジックによって連結される一連の操作のことで、フローによってヘルスチェックや、トラブルシューティング、またはその他の繰り返し実行する IT サポートタスクなどのタスクが自動化されます。 フローを実行することを実行と言います。

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。

ステップ 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、ステップ 2 を実行します。

ステップ 2: mypage.htm を、サーバー development1 からサーバー production1 にコピーします。

ステップ 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 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の実行、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するソフトウェアです。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操作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、ステップの操作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。 前述の例のステップ 1 では、Web ページに特定のテキストが含まれているかどうかをチェックするというオペレーションを用いています。 ステップ 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: 操作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、「ページが見つからない」、「テキストが見つからない」、「成功」（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、「成功」か「失敗」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。

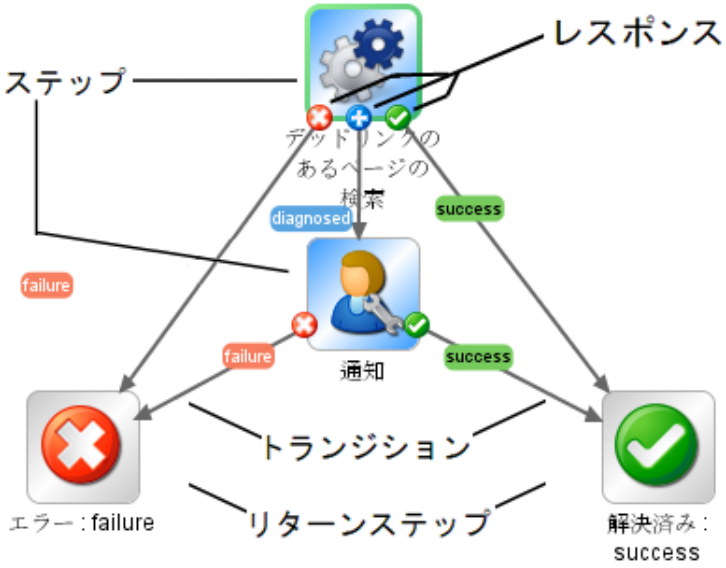


図 1 - フローの構成要素

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ことが必要です。

- 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、ステップに含まれるオペレーションの各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。 ほとんどのオペレーションには複数のレスポンスがあります。したがって、ほとんどのステップには複数のトランジションがあり、1 つ以上の後続ステップに接続しています。 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によって表示されます。

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、次のトピック「*フロー作成の応用*」およびその中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、フローがリアルタイムの状態に適応できるようにする。

  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と割り当て方法については、「*フロー変数の作成と割り当て*」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、後でフロー内で使用するさまざまなデータを戻すものもあります。 Linux のメモリ統計情報を取得するオペレーションでは、1 つのコマンドで、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。

  - 「ページが見つからない」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。
  - 「テキストが見つからない」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。

次の図は、以下の動作を表しています。

- 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。

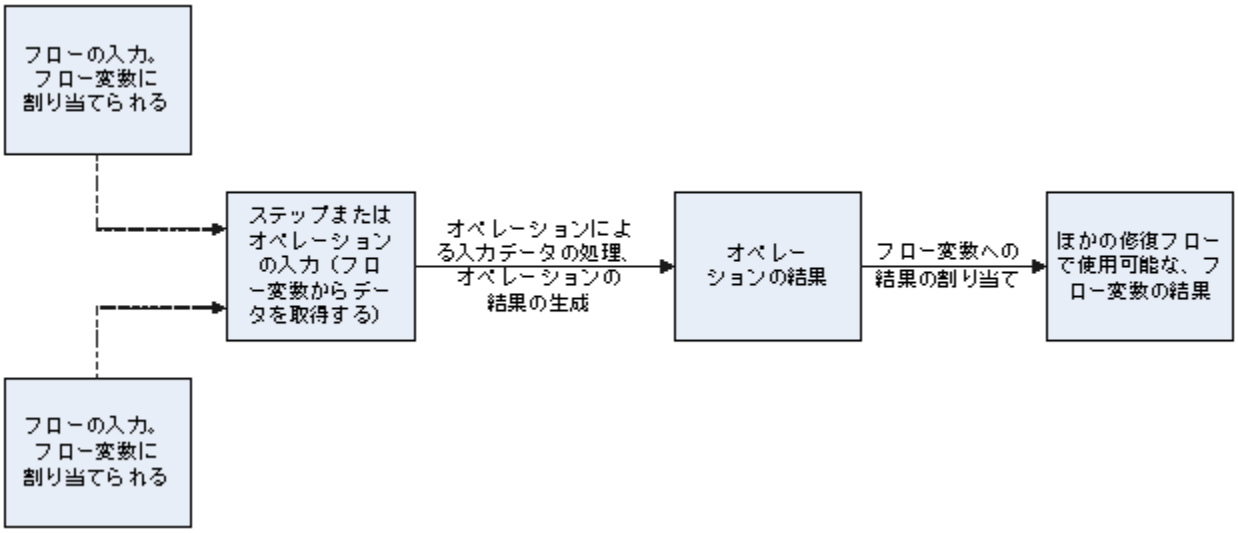


図 3 - オペレーションがデータを取得する仕組み

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。

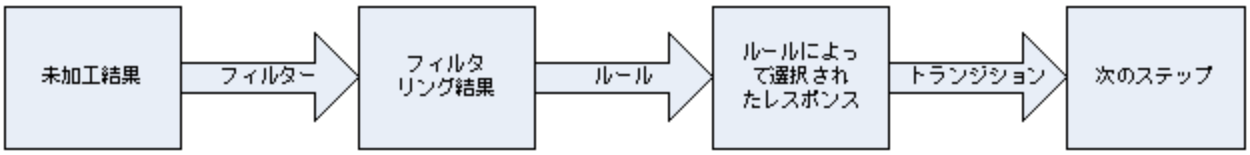


図 4 - オペレーションの未加工結果から次のステップが決まる仕組み

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。

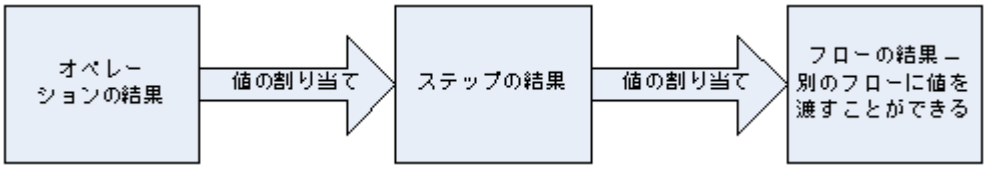


図 5 - オペレーションの結果がフローの結果になる仕組み

高度なフローアーキテクチャおよび概念

「*フローの概念*」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。 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操作の順番
- 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て済むため、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、以下の操作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: アーキテクチャとデータの移動*」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 [前のステップの結果] 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操作です。 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であると言えます。

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ーです（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ます。
- 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間またはグローバルコンテキスト間で値を受け渡すことができますが、特定の状況ではできないことがあります。このような状況については、該当箇所で説明します。

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。

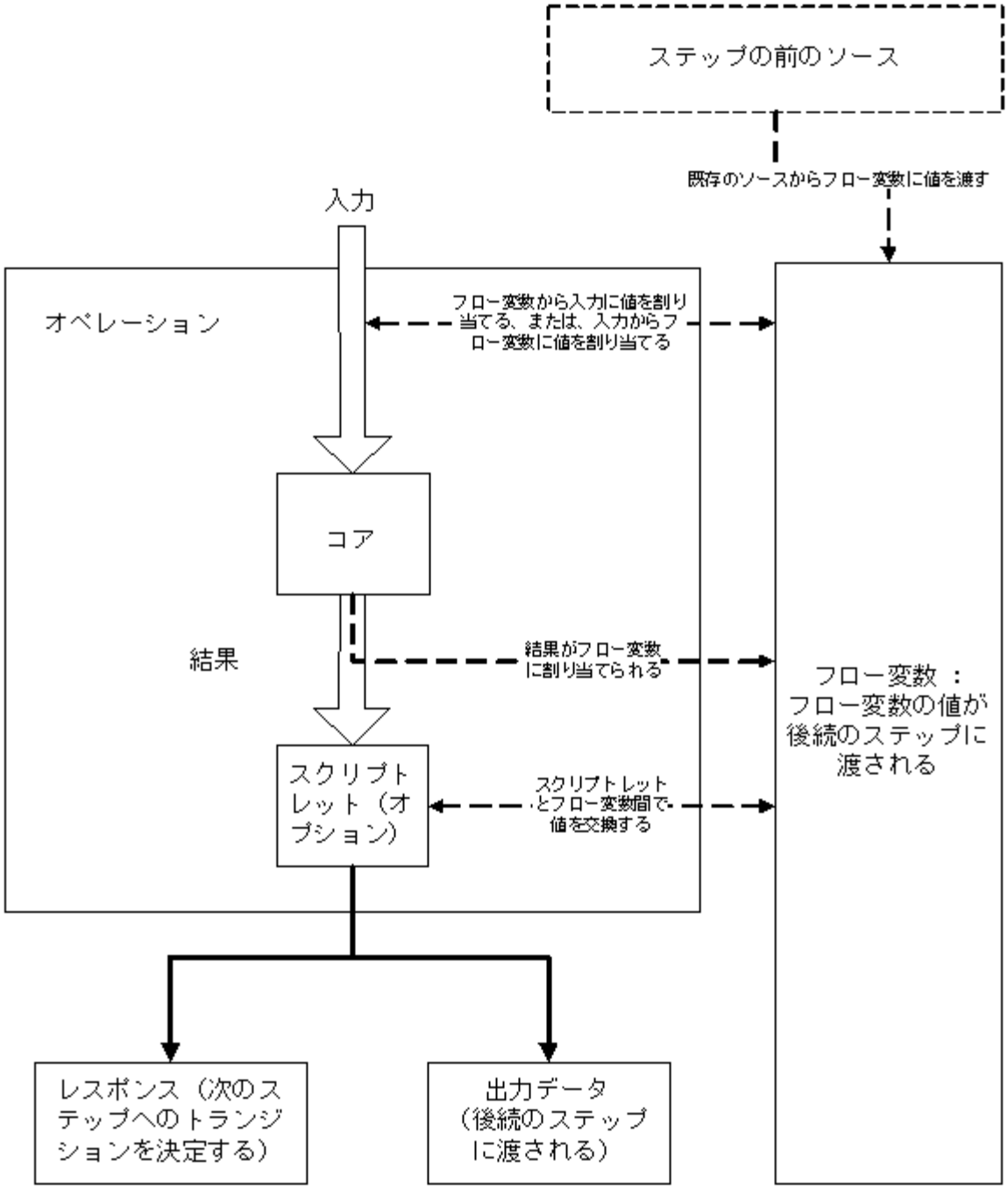


図 6 - オペレーション内でのデータの移動

注: 統合、Operations、ユーティリティ　オペレーション

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。

フロー変数とグローバル変数

フロー変数は範囲においてローカルで、グローバル変数はフローの実行で使用可能なデータを作成します。

- グローバル変数は、システムプロパティと選択リストから取得される名前と値のペアで、フローの開始時にフロー実行に提供されます。
- フローの実行内で、グローバル変数はフローおよび、適切なフロー設計によりサブフローがあればサブフローも利用できます。
- フロー変数は、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。

グローバル変数とフロー変数の詳細については、「*グローバル変数とフロー変数: 再利用可能なデータの作成*」を参照してください。

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、ステップのインスペクターで行います。

  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になります。 つまり、その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壊す可能性があります。
- 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。 以下はその例です。
  - 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めません。
  - 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 [スクリプトレット] 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

各入力は変数にマッピングされます。入力の値は多様なソースから取得されます。

どの要素に入力を追加するかによって、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っては生成されない入力は、フローのプロパティで値を設定することをお勧めします。こうすることで、フローを実行する前に入力の値が利用可能になります。
- 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（ステップのインスペクターで）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。
- Library 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 [プロパティ] 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方法の詳細については、「*オペレーションのプライマリ出力の指定*」を参照してください。

ステップの結果には次の 2 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（「未加工出力」）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は、オペレーションで **[フィールドからプライ**

マリ出力] 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 **[結果]** 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。

- 未加工出力、出力、およびレスポンスはオペレーションに関連しています（フローも含まれます。フローも一種のオペレーションです）。
- ステップに結果を追加できます。

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。

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 フォルダー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。

  または

  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。

  または

  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。この現象は、Central がクラスタ化されたときに発生する可能性もあります。

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。
- *[自分の変更/チェックアウト] パネル*: OO オブジェクトのチェックイン / チェックアウト状態を表示できます。
- *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 [プロパティ] 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。
- *[ブックマーク] パネル*: よく使用するオペレーションやフローのショートカットを保存できます。 **[ブックマーク]** タブをクリックするとこのペインを表示できます。
- *[アイコン] パネル*: 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。 **[アイコン]** タブをクリックするとこのペインを表示できます。
- *フロー変数] パネル*: フロー内で使用されるフロー変数を示し、フロー変数を一覧し、各フロー変数の用途を説明します。 **[フロー変数]** タブをクリックするとこのペインを表示できます。

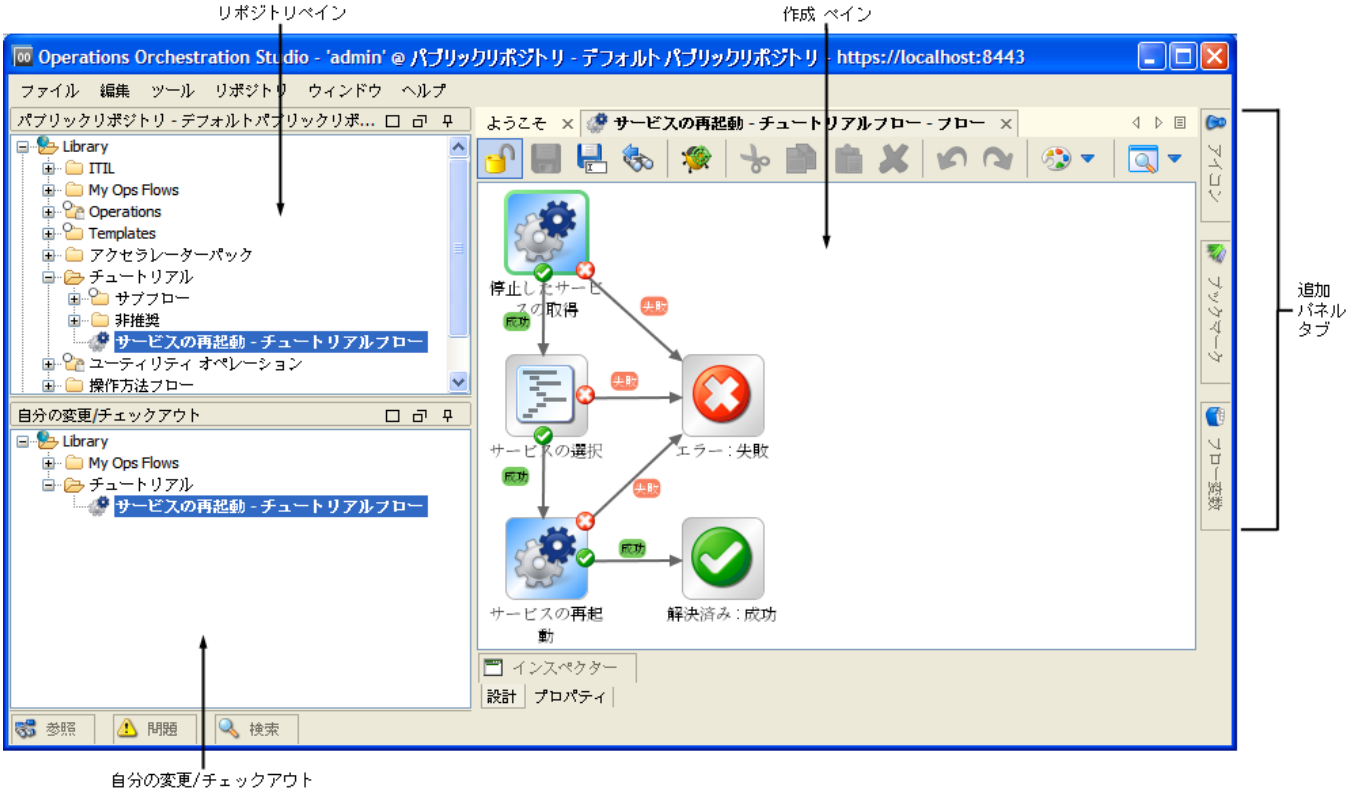


図 7 – Studio の主な構成要素

さらに、メニューバーに注目してください。 メニューとメニューコマンドの使用方法については、本ドキュメントの該当するセクションで説明します。 ただし、以下に注意してください。

- リポジトリに関連するコマンドは、**[リポジトリ]** メニューにあります。ただし、**[新規リポジトリとしてエクスポート]** は例外です。 **[新規リポジトリとしてエクスポート]** コマンドは **[ファイル]** メニューにあります。

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。

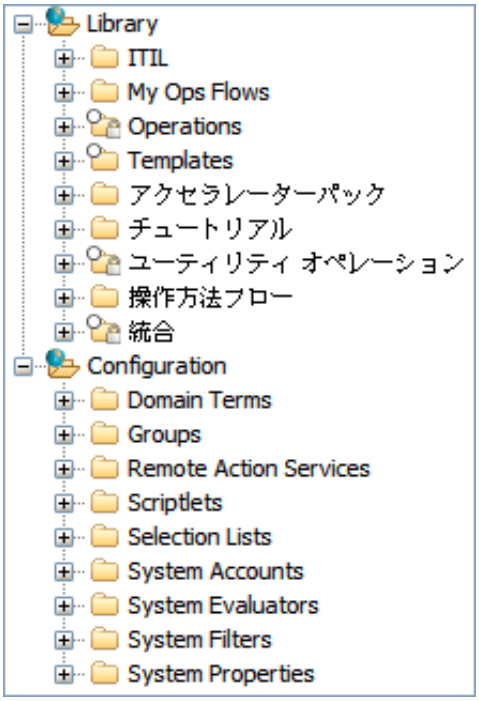


図 8 - リポジトリペインに表示されたライブラリ

[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。 フローの作成に利用可能なフローおよびオペレーションを理解するには、**[アクセラレーターパック]**、**[統合]**、**[ITIL]**、**[Operations]**、および **[ユーティリティ　オペレーション]** フォルダーを開いてください。

デフォルト Studio コンテンツの最上位のフォルダーは以下のとおりです。

- **[アクセラレーターパック]**

  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。

  アクセラレーターパックの最上位にあるフローの多くは、完全なヘルスチェック、問題選別、診断、修復機能を備えています。

- **[統合]**

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、アクセラレーターパック内にあります。
- **[ユーティリティ オペレーション]**
  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。

[自分の変更 / チェックアウト] パネル

[自分の変更 / チェックアウト] パネルは **[リポジトリ]** パネルの下にあり、チェックアウトされた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たオブジェクトは、**[自分の変更 / チェックアウト]** パネルに太字で表示されます（別のユーザーがチェックアウトしたアイテムは、イタリックで表示されます）。

Stueio 内の他のパネルと同様に、Studio アプリケーションウィンドウと関係なく配置できるように **[自分の変更 / チェックアウト]** パネルをフローティング状態にすることができます。

これは、Studio 内でフローグラフまたはデバッガーを操作している際にスクリーンリアルエステートを管理するのに役立ちます。

[自分の変更 / チェックアウト] パネルのサイズ変更および再配置するには、「*パネルのサイズ変更と再配置*」を参照してください。

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。

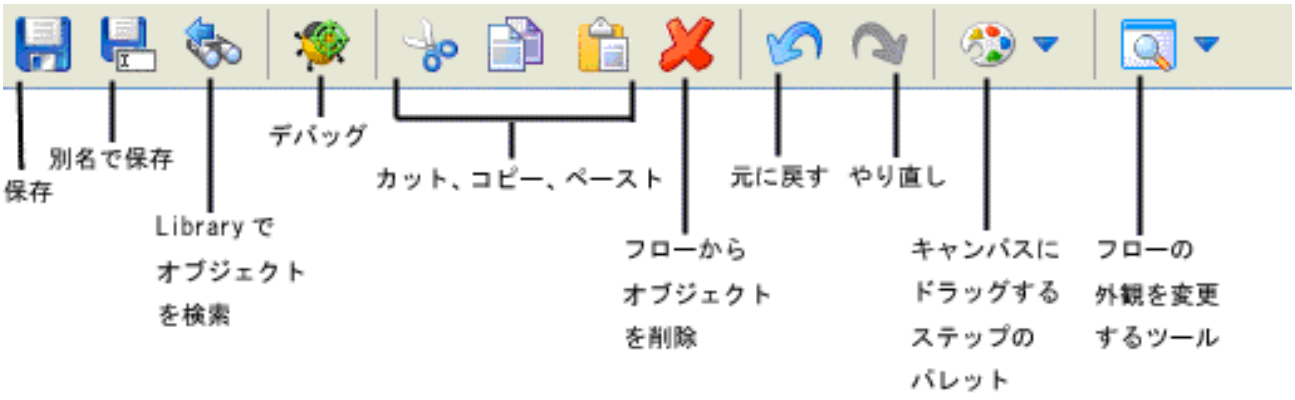


図 9 - フロー作成ツールバー

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（**[保存]** アイコンの隣）に **[チェックイン/チェックアウト]** アイコン（ ）も表示されます。 **[チェックイン/チェックアウト]** 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。

- **[ステップパレット]** アイコン（ ）
- **[表示オプション]** 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。

- リポジトリツリー内で選択 （ ）：使用するフローまたはオペレーションを選択するために Library を展開します。
- フローのデバッグ（ ）：デバッガーを開き、現在のフローの実行を開始します。

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（**成功**、**診断済み**、**操作なし**、**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。

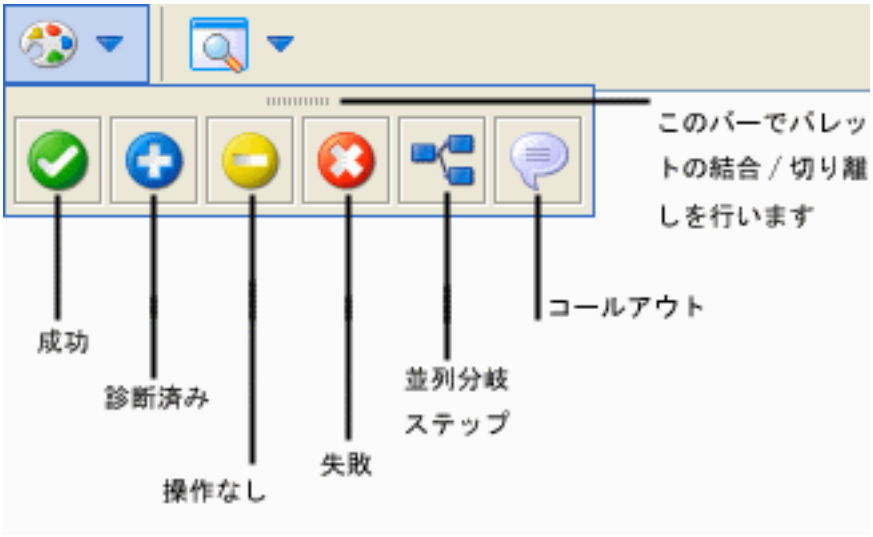


図 10 - ステップパレット

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。

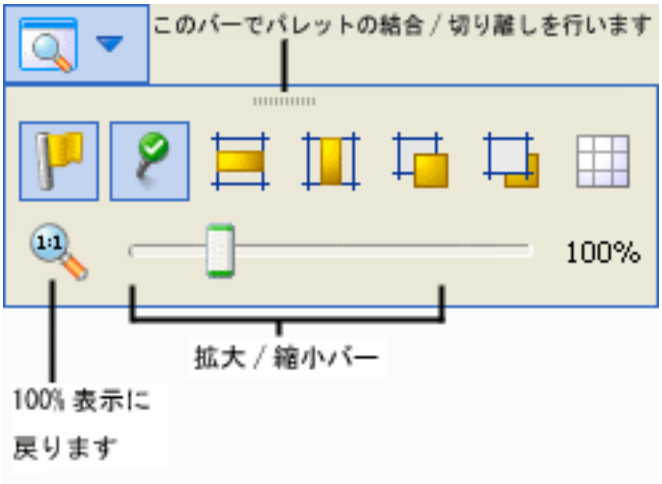


図 11 - オプション表示パレット

このパレットには次のアイコンがあります。

ラベルの表示/非表示：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。

接続されたレスポンスアイコンの表示/非表示: 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。

選択項目を水平方向に整列: 選択したステップを横に並べます。

選択項目を垂直方向に整列: 選択したステップを縦に並べます。

最前面へ移動: 選択したオブジェクトを前面に移動します。

最背面へ移動: 選択したオブジェクトを背面に移動します。

グリッドの表示 / 非表示：ステップを配置するときに利用できる作成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

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。

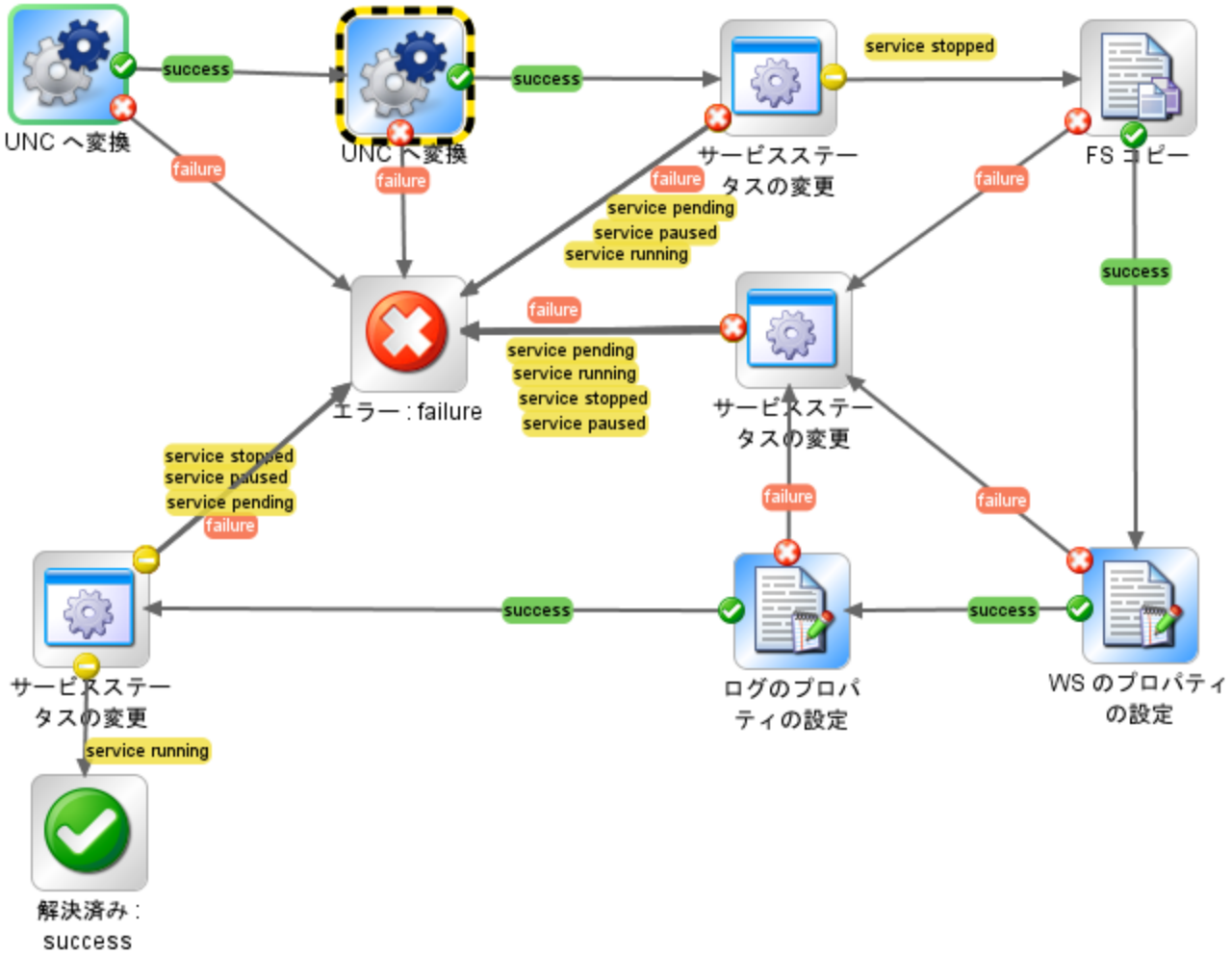


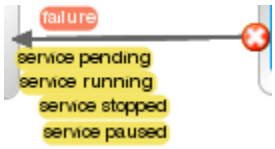
図 12 - フロー図

「*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。

- デバッグ用のブレークポイントが作成されているステップは、下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ー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、またはそのインスペクターを開くことができます。



[ブックマーク] パネル

[ブックマーク] ペイン（ウィンドウの右上にある **[ブックマーク]** タブをクリックすると開きます）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*」を参照してください。

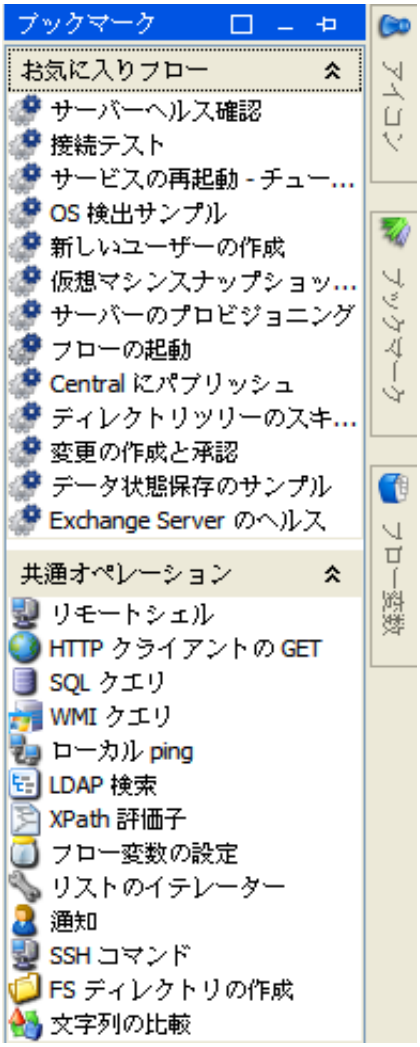


図 13 - [ブックマーク] パネル

Studio 内の **[アイコン]** および **[フロー変数]** パネルと同様に、Studio アプリケーションウィンドウと関係なく **[ブックマーク]** パネルを移動可能にできます。 これは、Studio 内でフローグラフまたはデバッガーを操作している際にスクリーンリアルエステートを管理するのに役立ちます。 パネルのサイズ変更および再配置の詳細については、「*パネルのサイズ変更と再配置*」を参照してください。

[アイコン] パネル

[アイコン] パネル（[Studio] ウィンドウの右上にある **[アイコン]** 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。

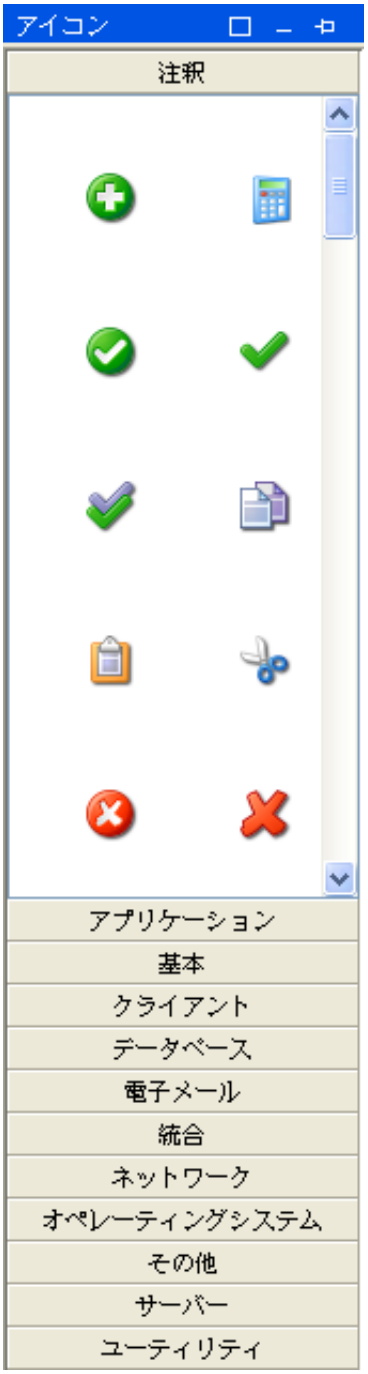


図 14 - フロー内で使用できるアイコン

Studio 内の **[ブックマーク]** および **[フロー変数]** パネルと同様に、Studio アプリケーションウィンドウと関係なく **[アイコン]** パネルをフローティング、または移動可能にすることができます。

これは、Studio 内でフローグラフまたはデバッガーを操作している際にスクリーンリアルエステートを管理するのに役立ちます。

[アイコン] パネルのサイズ変更および再配置の詳細については、「*パネルのサイズ変更と再配置*」を参照してください。

[フロー変数] パネル - 概要

[フロー変数] パネル（ウィンドウの右上にある **[フロー変数]** タブをクリックすると開きます）にはフロー内で使用されるフロー変数が一覧され、フロー変数の各使用方法が説明されます。 このペインでは、フロー変数のスペルのチェックや、その使用場所と使用方法の確認などを効率的に行うことができます。

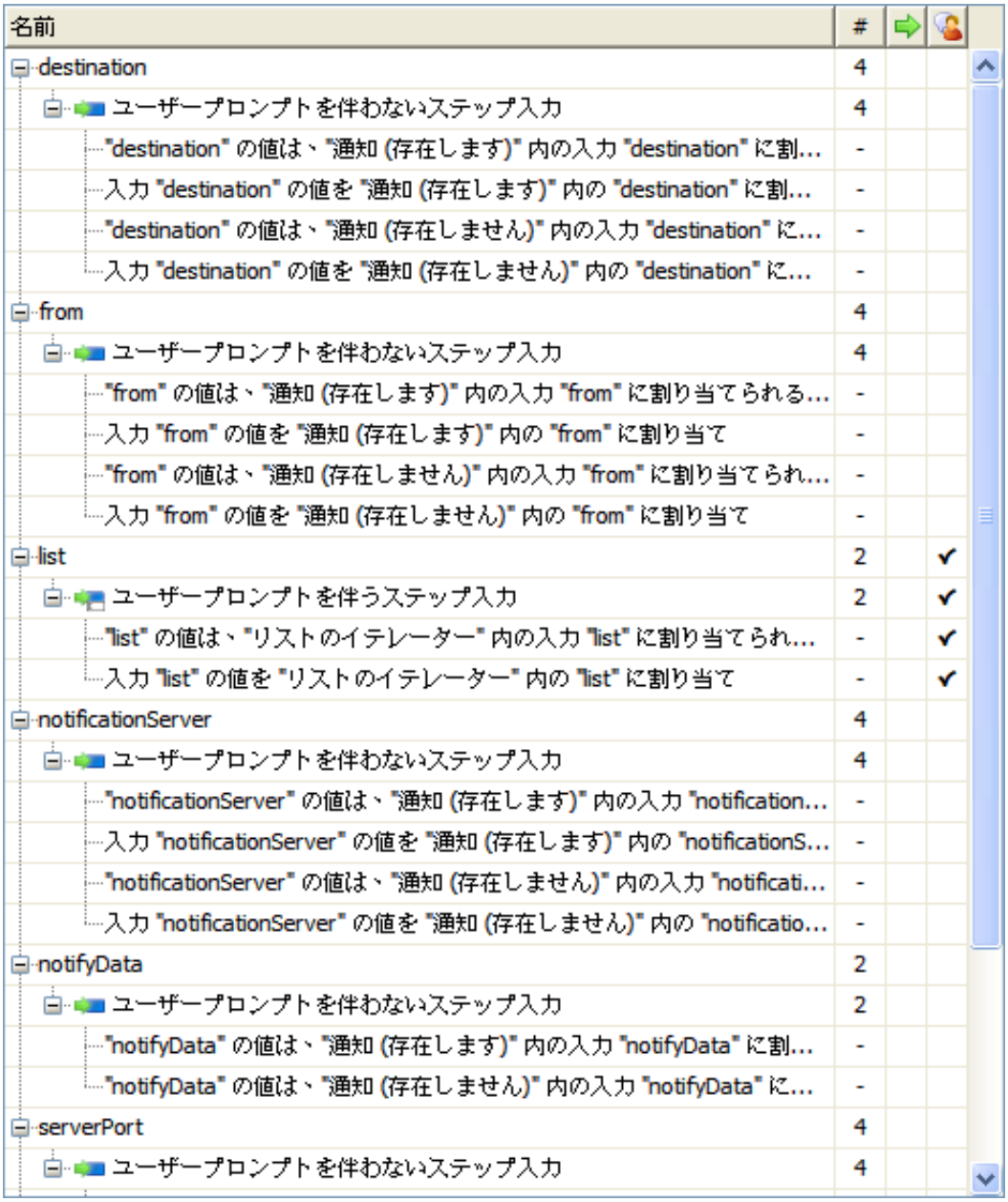


図 15 - [フロー変数] パネル

[フロー変数] パネルの使用の詳細については、「*フロー変数の管理*」を参照してください。

[フロー変数] パネルのサイズ変更および再配置の詳細については、「*パネルのサイズ変更と再配置*」を参照してください。

パネルのサイズ変更および再配置

OO Studio でフローグラフ、およびパネルの 1 つまたはデバッガーを操作している際にスクリーンリアルエステートを容易に管理するには、Studio 内で **[ブックマーク]**、**[アイコン]**、**[フロー変数]**、**[検索]** パネルをフ

ローティングにすることができます。 これにより、Studio アプリケーションウィンドウと関係なくこれらのパネルを配置することができます。

[検索] パネルを開いたままにするには

- **[検索]** ペインの右上隅にある **[ピン]** アイコン（）をクリックします。

[検索] パネルを開いたまま固定しておくと、このパネルをデタッチしてデスクトップ上の適切な場所に移動させることができます（これにより、作成ペイン内にフロー図のためのスペースが広がります）。

パネルをサイズ変更または再配置するには

1. パネルを開いたまま固定（「ピン設定」）した状態で、このパネルをフローティングにするには、**[フロートを切り替え]** アイコン（）をクリックします。
2. このパネルを再度ドッキングするには、**[フロートを切り替え]** アイコン（）の現在のフォームをクリックします。
3. このパネルを最大化表示するには、**[最大化 / 元に戻す]** アイコン（）をクリックします。
4. 最大化表示する前にパネルのサイズを元に戻すには、**[最大化 / 元に戻す]** アイコン（）の現在のフォームをクリックします。

   注: パネルを Studio の左側に配置したまま状態で最小化表示すると、パネルタブはそれ自身を Studio の左の枠にアタッチします。

パネルタブを [Studio] ウィンドウの右側に戻すには

- **[ウィンドウ]** メニューで、**[ウィンドウレイアウトのリセット]** をクリックします。

オブジェクトのプロパティシート

[フロー]、**[オペレーション]**、**[Configuration]** フォルダー内のオブジェクトの **[プロパティ]** シートは、オブジェクトの値を追加、削除、変更するためのエディターです。 Library に格納されている大部分のオブジェクトについては、**[プロパティ]** シートがオブジェクトを操作する際のインターフェースです。 ただし、フローの場合は、**[プロパティ]** シート以外に **設計** ビューも使用できます。

[プロパティ] シートでは編集ができるだけでなく、次の図のように **[プロパティ]** 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。

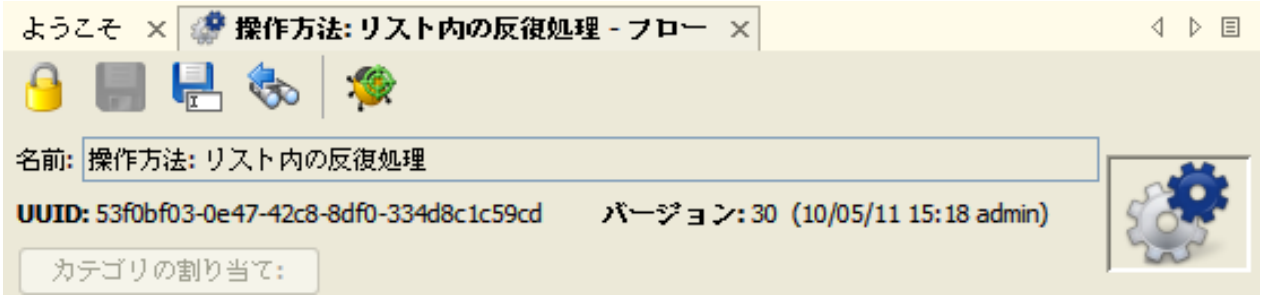


図 16 - [プロパティ] シートのヘッダー

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 / インスペクターのキーボードショートカット

[スクリプトレット] タブのキーボードショートカット

[ブックマーク] パネル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 /インスペクタ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
選択した行の編集	CTRL + <右矢印>

[スクリプトレット] タブ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

[ブックマーク] パネル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
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。
  - オブジェクトの名前の上にカーソルを乗せると、オブジェクトをチェックアウトしたアカウント名が表示される。
- 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 **[プロパティ]** 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 **[プロパティ]** 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 ケイパビリティ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（その作成者が別の管理者である場合でも）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。

重要情報: 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。 この現象は、Central がクラスタ化されたときに発生する可能性もあります。

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、パブリックリポジトリが Studio に追加されていて、その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オブジェクトを編集します。編集したオブジェクトをパブリックリポジトリにチェックインし、オブジェクトの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
  - 変更できる。オブジェクトがフローの場合、白いキャンバスが表示される。
  - **[自分の変更 / チェックアウト]** ペインに表示される。
  - **[リポジトリ]** ペインおよび **[自分の変更 / チェックアウト]** ペインで、太字で表示される。

  さらに、

  - ツールバー上の **[チェックイン / チェックアウト]** アイコンはロック解除アイコン（ ）です。
  - 変更を保存するたびに、オブジェクトの **[プロパティ]** シートに表示されるバージョン番号が 0.1 ずつ増える（1.1、1.2 など）。 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 つずつ増える（2.0、3.0 など）。

    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で、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 **[チェックイン / チェックアウト]** 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ます。
- 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
  - 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 ケイパビリティ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、**[チェックイン/チェックアウト]** 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、**[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、**[リポジトリ]** をポイントしてから **[チェックアウト]** 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、**[リポジトリ]** をポイントしてから **[ツリーのチェックアウト]** 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、**[ツリーのチェックイン]** 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コミットするには、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
- 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、**[チェックイン]** 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、**[リポジトリ]** ペインの**ライブラリ**ツリーで、あるいは **[自分の変更/チェックアウト]** ペインで、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、**[リポジトリ]** をポイントしてから **[チェックイン]** をクリックします。

   [チェックイン] ダイアログボックスが表示されます。

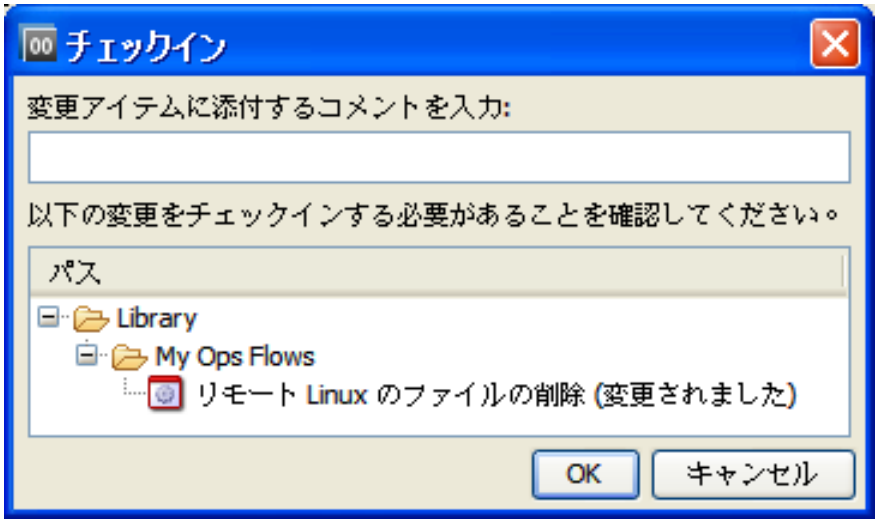


図 17 - [チェックイン] ダイアログボックス

2. **[変更アイテムに添付するコメントを入力]** 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]** ペインの **[Library]** フォルダーまたは **[Configuration]** フォルダーで、あるいは **[自分の変更/チェックアウト]** ペインで、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、**[リポジトリ]** をポイントしてから **[ツリーのチェックイン]** 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 **[自分の変更/チェックアウト]** 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 **[自分の変更/チェックアウト]** 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。

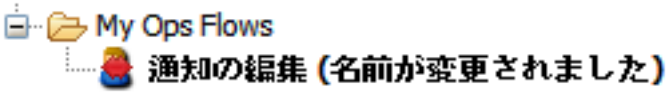


図 18 - 別の作成者のオブジェクトと競合するオブジェクト

このようなときは、両方のバージョンの **[プロパティ]** 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. **[自分の変更/チェックアウト]** パネルで競合が存在するオブジェクトを右クリックし、**[競合の解決]** 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。

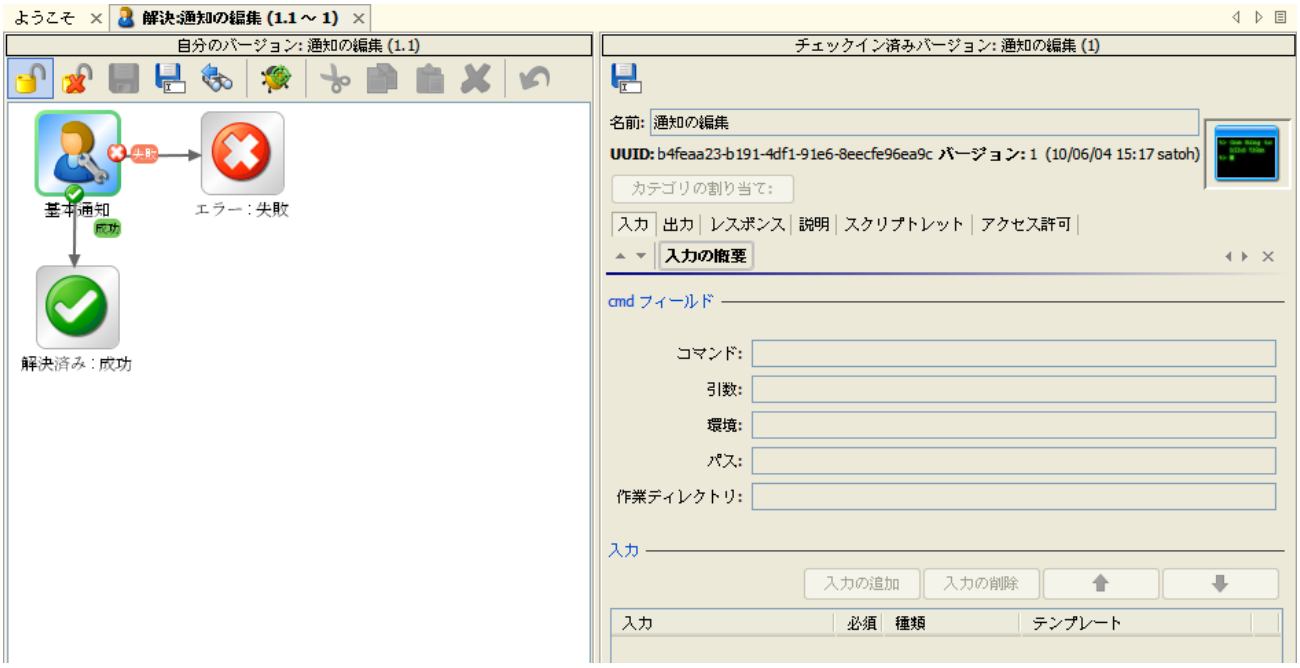


図 19 - フローの競合バージョンの解決

オブジェクトのバージョンのフロー図（または、**[プロパティ]** シート）で、変更を加え保存できます。 ターゲットバージョン（チェックイン済みバージョン）のフロー図（または、**[プロパティ]** シート）では、オブジェクトを別の名前で保存できます（[名前を付けて保存]）。

2. 自分の変更を破棄して競合を解決する場合は、**[変更の破棄]** アイコン（ ）をクリックします。

   または

   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、**[チェックイン]** 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 ケイパビリティ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、**[名前を付けて保存]** 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 ケイパビリティ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、**[リポジトリ]** をポイントして **[強制チェックイン]** 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
オブジェクトのバージョンの比較

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、**[リポジトリ]** をポイントしてから **[変更の破棄]** をクリックします。

   [確認] ダイアログボックスが表示されます。

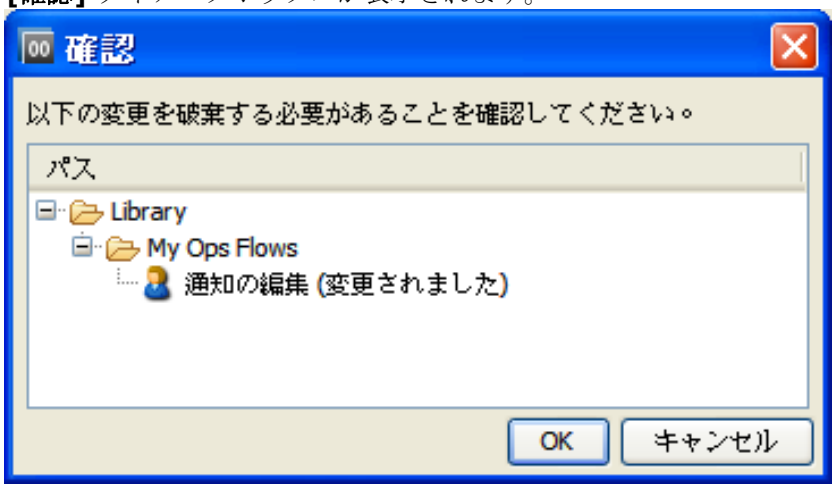


図 20 - [確認] ダイアログボックス

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。

   オブジェクトの変更が元に戻り、オブジェクトはチェックインされます。

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、**[リポジトリ]** をポイントしてから **[履歴の表示]** をクリックします。

   [バージョン履歴] 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、**[開く]** をクリックします。

   開いたバージョンは、作成ペインに開かれます。

   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。

4. 開いたバージョンを保存するには、**[名前を付けて保存]** 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。

5. 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

1. **[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。
2. 表示されたコンテキストメニューで、**[リポジトリ]** をポイントしてから **[履歴の表示]** をクリックします。

   [バージョン履歴] 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

3. バージョンの一覧で、取り消したいバージョンを右クリックし、**[復元]** をクリックします。
   復元するバージョンは、作成ペインに開かれます。
4. オブジェクトを保存してチェックインします。

オブジェクトのバージョンの比較

ライブラリオブジェクトのバージョンを比較するには

1. **[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。
2. 表示されたコンテキストメニューで、**[リポジトリ]** をポイントしてから **[履歴の表示]** をクリックします。
3. **[バージョン履歴]** ダイアログで、比較する 2 つのバージョンを選択します。
4. 選択した 2 つのバージョンのうちのどちらか一方のバージョンの行を右クリックし、**[比較]** をクリックします。

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

1. **[リポジトリ]** ペインで**[Library]** フォルダーを右クリックし、**[新規]** をポイントしてから **[フォルダー]** をクリックします。
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。
   - 作成するフォルダーは自動的にチェックアウトされます。

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。

フローへの入力をユーザープロンプトによる入力から特定の値に変更し、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 **[説明]** タブに次の情報を追加します。（同じテクノロジーと情報をやり取りする複数のフローまたはオペレーションを作成する場合、1 つのフォルダーにまとめて、フォルダーの **[説明]** タブにこの情報を記載します。これはデフォルトの HP OO コンテンツのプラクティスです）。 **[説明]** タブへの情報の追加により、他の作成者や Central ユーザーがドキュメントの生成機能を利用できるようになることにご注意ください。 ドキュメントの生成の詳細については、「*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。
- 任意に追加する実装に関する**注意点**。例えば、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 **サービスの再起動 - チュートリアルフロー**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、**[新規]** をポイントしてから **[フロー]** 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. **[ファイル]** メニューから、**[新規]** をポイントして **[フロー]** 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。

   新しいフローは自動的にチェックアウトされます。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（サービスの再起動）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（ネットワークの確認）もあります。

あるいは、成功および失敗のリターンステップ（フロー最後のステップである場合が多い）のみで開始する必要がある場合には、空白のフロー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、**[ようこそ]** タブをクリックして Studio の **[ようこそ]** ページを開き、**[新規フロー]** 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。

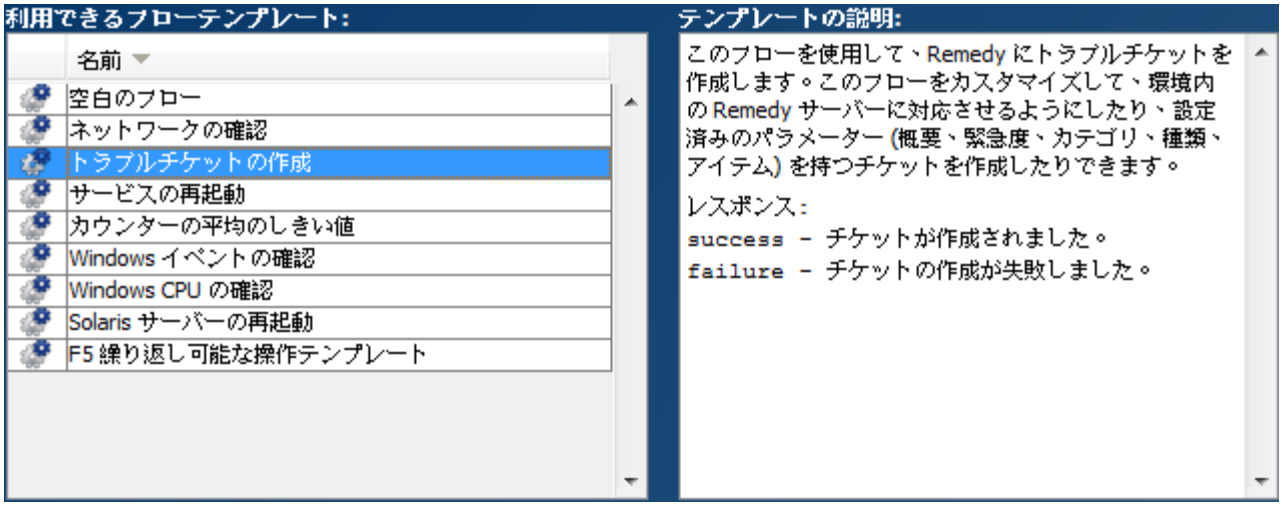


   図 23 - 使用頻度の高いフローのテンプレート

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

   [作成] ペインにフローが表示されます。

また、Studio の **[リポジトリ]** ペインの **[テンプレート]** フォルダーで、目的のテンプレートをダブルクリックするか、作成ペインにドラッグすることで、作成ペインで目的のテンプレートを開くことができます。

次は、オペレーションを作成するか、またはオペレーション（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 **[説明]** タブに次の情報を追加します。（同じテクノロジーと情報をやり取りする複数のフローまたはオペレーションを作成する場合、1 つのフォルダーにまとめて、フォルダーの **[説明]** タブにこの情報を記載します。これはデフォルトの HP OO コンテンツのプラクティスです）。 **[説明]** タブへの情報の追加により、他の作成者や Central ユーザーがドキュメントの生成機能を利用できるようになることにご注意ください。 ドキュメントの生成の詳細については、「*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。
- 任意に追加する実装に関する**注意点**。例えば、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 リポジトリの操作方法フロー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。
   - ステップのパレットで、**[コールアウト]** アイコンをクリックしてフロー図にドラッグする。

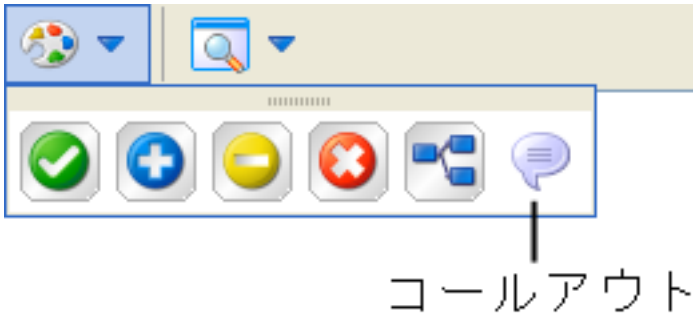


図 24 - ステップのパレット、および [コールアウト] アイコン

   - フロー図内で右クリックし、**[コールアウトの挿入]** 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。

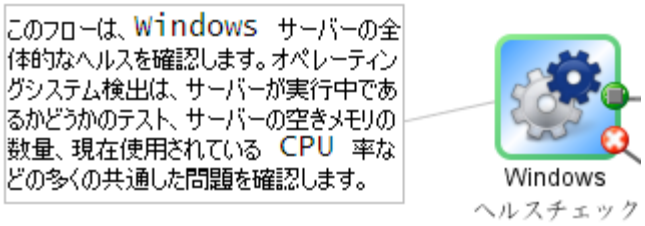


図 26 - テキストが入力されステップに接続されたコールアウト

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]** フォルダーのサブフォルダーに格納されています。 Library の主なフォルダーは次のとおりです。

- [アクセラレーターパック]

  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ーにまとめられています。

- [操作方法フロー]

  必要とされる頻度の高いフローの作成方法を記述するフローです。 このフローにはコールアウトによる注釈が含まれます。コールアウトは、入力、フロー変数、およびステップ結果を介してフロー内でデータが移動する方法に特に焦点を当てたものです。

- [My Ops Flows]

  自分が作成したフローを格納するフォルダー。 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 （テンプレートを基にして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]

  このフォルダーには、アクセラレーターパック 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。 この中のフォルダーの多くには **[サンプル]** というフォルダーがあり、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。

  [アクセラレーターパック]、[統合]、[Operations]、[ユーティリティ　オペレーション] の各フォルダーはシールされています（ロック付きのフォルダーアイコン（ ）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用途が広く、根本的に重要であるため、ユーザーが編集できないようになっています。 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 [ブックマーク] ペインにドラッグします。

- [テンプレート]

  フローテンプレートで、これを使用すると、最もよく必要となる頻度の高いタスクを完了するためのフローの作成がすぐに開始できます。

- [チュートリアル]

  Studio チュートリアルの 1 つで使用されるフローです。

- [ユーティリティ　オペレーション]

  フローを作成する際に最もよく使用されるタスクを実行するオペレーションです。

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 **[検索]** 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。 **[検索]** 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： オペレーションを有効に利用するための情報

フローまたはオペレーションの検索方法

[検索] パネルは、Library 全体に対して全文検索を実行します。

[Studio] ウィンドウの左下にある **[検索]** タブをクリックする（またはキーボードの F3 を押す）と、**[検索]** パネルが開きます。ここで、フローおよび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、フローおよびオペレーションを検索することができます。

図 27 - Studio のタブ

[Studio] ウィンドウとは関係なく、**[検索]** パネルにフォーカスを設定しないで表示させたままにする方法、またはパネルをサイズ変更および再配置する方法については、「*パネルのサイズ変更および再配置*」を参照してください。

次のサブトピックでは、フローおよびオペレーションを検索するためのさまざまな方法を詳細に説明します。

- *検索の定義*
- *Lucene 構文を使用した検索*
- *[検索] パネルからのオペレーションへのアクセス*

検索の定義

OO は、オペレーション（オペレーションの 1 種であるフローなど）を対象に、検索するフィールドと、そのフィールド内で検索する値を指定して検索を実行します。 次のフィールドから、Library のオペレーションの各種プロパティまたはコンポーネントにアクセスできます。

- **名前**
- **UUID**
  オペレーションの UUID の **[プロパティ]** シート上でオペレーションの UUID を見つけることができます。
- **種類**
  一覧内の「フロー」以外の種類は、フローでないオペレーションにのみ適用されます。 オペレーションの種類は、オペレーションの作成者が cmd や、HTTP、Telnet などの新規作成時に選択します。 オペレーションの作成者がオペレーションの種類を選択する手順については、「*新しいオペレーションの作成*」を参照してください。
- **説明**
  [説明] タブに表示される文字列。
- **カテゴリ**
  フローが割り当てられたカテゴリ。 カテゴリは、フローに関連付けできるドメインタームの 1 つです。 カテゴリが割り当てられると、割り当てがフローの **[プロパティ]** シートに表示されます。 フローは作成者の操作のみによってカテゴリに割り当てられます。 カテゴリの詳細については、「*カテゴリ: フローの分類*」を参照してください。

- **入力**

  フローまたはオペレーションへの入力名。 入力の詳細については、「*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。
- **ドメインターム**

  フローまたはオペレーションに割り当てられているドメインターム。 ドメインタームは、手動で作成された選択リストに比べて利点を持つ特殊な選択リストで、作成者によって割り当てられます。 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。

Lucene 検索構文を使用することで、フィールドの組み合わせを検索することもできます。

オペレーションを検索するには

1. **[検索]** パネルを開くには、[Studio] ウィンドウの下部にある**[検索]** タブをクリックします。

2. **[検索]** ドロップダウンリストで、検索するフィールドを選択します。

   または

   検索にすべてのフィールドを含めるには、**[検索]** ドロップダウンリストを ***<すべてのフィールド>*** に設定したままにします。

   Lucene 構文を使用した検索を構築するには、「*Lucene 構文を使用した検索*」を参照してください。
3. **[検索対象]** テキストボックスには、検索する文字列を入力します。

   検索文字列にスペースを含めると、完全一致検索または「あいまい」検索のどちらでも指定できます。 完全一致検索は、文字列に含まれているスペースを含めて文字列全体を 1 つの検索値として扱います。 「あいまい」検索は、スペースで区切られた文字列をすべて検索します。

   注: Lucene 構文は検索のこの側面を制御します。したがって、Lucene 構文を使用して検索を指定する場合 **[完全一致検索]** チェックボックスは使用できません。
4. 1 つの検索文字列の一部としてスペースを扱う検索を指定するには、**[完全一致検索]** チェックボックス選択します。

   または

   他の検索文字列との区切り文字としてスペースを扱う検索を指定するには、**[完全一致検索]** チェックボックスの選択を解除します。
5. 検索アイコン（ ）をクリックします。

   [検索] ドロップダウンリストで **[カテゴリ]** または **[種類]** を選択した場合は、**[検索対象]** ボックスはドロップダウンリストになります。このリストから使用可能なオペレーションの種類または現在定義されているカテゴリを選択できます。

以下に検索例を 2 つ示します。

- **[サービスのインストール]** フローを検索するには、[検索対象] テキストボックスに、「Web サービス」または、単に「サービス」と入力します。

  下記の図では、以下に注意してください。
  - 入力する検索文字列の検索では、大文字と小文字は区別されません。
  - 検索にフィールドが指定されていないので、検索結果には検索文字列がすべてのフィールド内で見つかったオペレーションが含まれます。

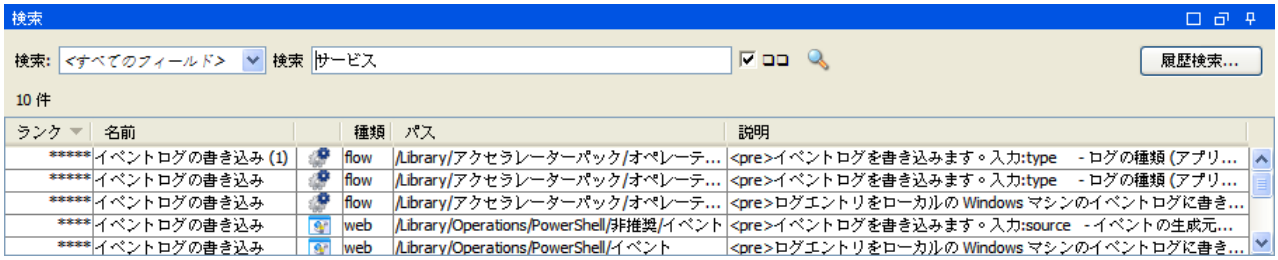


図 28 - [検索] ペインと結果

- 名前に「Web サービスのインストール」（または「web サービスのインストール」）が含まれているオペレーションのみを見つけるように直前の検索を絞り込むには、**[検索]** ドロップダウンリストで **[名前]** を選択します。その場合、検索結果には次のフローのみが含まれます。

図 29 - [名前] フィールド内の検索結果の結果

ヒント: オペレーションの **[説明]** タブから説明が取得され、その説明にはオペレーションの使用を最大限に生かすために欠かせない次のような情報が含まれます。

- オペレーションの入力が必要とする情報。
- 結果に含まれる情報。
- オペレーションの要件と前提。

説明全体を参照するには、オペレーションの**[プロパティ]** シートの **[説明]** タブを参照してください。 オペレーションの **[プロパティ]** シートを開く方法については、次の手順を参照してください。

前に実行した検索コマンドの一覧を表示するには

- **[検索]** パネルで **[検索履歴]** ボタンをクリックします。

[前回入力した検索文字列] ポップアップ ウィンドウが開きます。

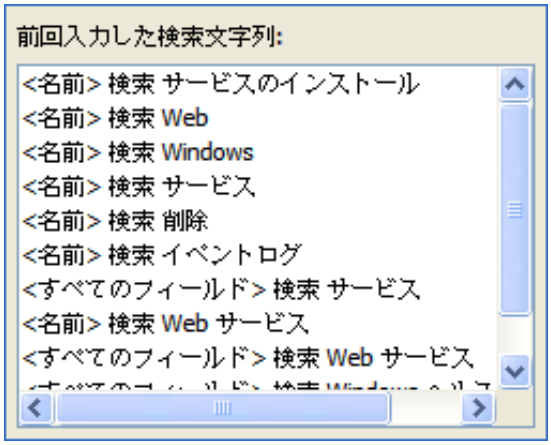


図 30 - [前回入力した検索文字列] ポップアップ ウィンドウ

このウィンドウには、最大 25 個までの前に実行した検索コマンドの一覧が含まれます。

一覧の検索コマンドのフォーマットは、「*検索テキストの<フィールド名>*」です。 一覧内の検索コマンドは、ダブルクリックすると実行できます。 実行した検索コマンドは一覧のトップに追加されます。

検索結果を並べ替えるには

- 列の列見出しをクリックします。

Lucene 構文を使用した検索

より詳細な結果を得るには、Apache Lucene 構文を使用した検索を構築できます。 Lucene 検索構文の詳細については、Apache Software Foundation の Web サイトを参照してください。

Lucene 検索構文を使用して検索するには

1. **[検索]** パネルを開いた状態で、**[検索]** ドロップダウンリストで、***<Lucene クエリを使用>*** を選択します。
2. **[検索対象]** テキストボックスに、Lucene 検索構文を使用したクエリを入力し、検索アイコン（ ）をクリックします。

   最もシンプルな Lucene 検索構文は以下のとおりです。
   <検索可能なフィールド名>:<検索対象の文字列>

   検索のヒント

   - この検索はブール AND を使用します。 AND を用いて検索語を 2 つ入力すると、両方の検索語が含まれるオペレーションまたはフローのみが検索結果として戻されます。 AND を使わずに 2 つの検索語を入力すると、どちらかの検索語を含む検索結果が得られます。
   - **category:database server** の検索のように、スペースを含む文字列のみを検索結果として得たい場合は、 **category:"database server" のように文字列を引用符で囲みます。**

   検索可能なフィールド名は以下のとおりです。 なお、ここで使用している検索文字列は一例です。

   - フローまたはオペレーションの名前

     例:
     `name:Get Temp Dir`
     `name:Clear Temp Dir`
   - オペレーションのタイプ

     例:
     `type:cmd`
   - カテゴリ

     例:
     `category:network`
   - 入力の名前

     例:
     `inputs:server`
   - フローまたはオペレーション UUID

     例:
     `id:1234-3453-3242-32423`
   - フローまたはオペレーションの説明に含まれている文字列

     例:
     `description:clear`
   - 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`

[検索] パネルからのオペレーションへのアクセス

検索結果の一覧から直接オペレーションまたはフローにアクセスし、使用するには

- オペレーションの **[プロパティ]** シートまたはフロー図を開くには、目的のオペレーションの行をダブルクリックします。

  または

  フロー図にオペレーションをドラッグし、オペレーションからステップを作成します。

説明： オペレーションを有効に利用するための情報

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。 **[検索]** タブで検索語「接続」を使って検索を実行した結果、**接続テスト** と **反復接続テスト** 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により判断できます。

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、**[アクセラレーターパック]**、**[統合]**、**[Operations]**、**[ユーティリティ　オペレーション]** 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。
- 任意に追加する実装に関する**注意点**。例えば、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、ステップのインスペクターを開いて、**[説明]** 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 の情報** 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もあります。

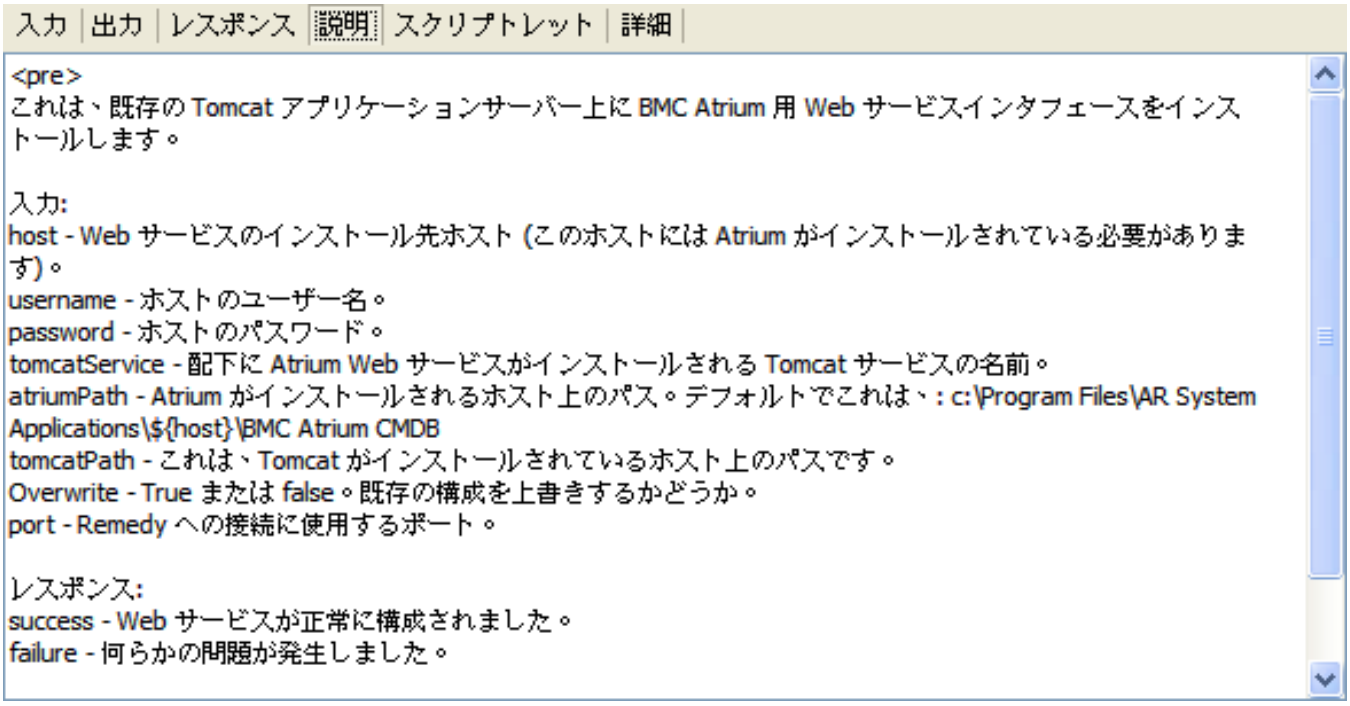


図 31 - オペレーションの説明

さらに、Studio のドキュメントの生成機能では、複数のフローやオペレーションについてのこれらの情報を 1 つにまとめることができます。 ドキュメントの生成の詳細については、「*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、[検索] 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。

フォルダーの説明の表示

フォルダーの説明には、フォルダー内に保存されているフローとオペレーションの概要、および該当する場合は、以下のような情報が記載されます。

- オペレーションを実行するソフトウェアのサポートバージョン。
- オペレーションの使用要件。
- オペレーションが使用するインターフェースまたは API。

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： オペレーションを有効に利用するための情報*」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、**[プロパティ]** をクリックします。
2. フォルダーの **[プロパティ]** シートで、**[説明]** タブをクリックします。

フローの説明の表示

フローはオペレーションの一種ですが、表示される説明はオペレーションの説明とは多少異なります。

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： オペレーションを有効に利用するための情報*」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 **[プロパティ]**（ペインの下部）をクリックし、**[説明]** タブをクリックします。

  または

  [リポジトリ] ペインで、フローを右クリックして **[プロパティ]** 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の表示

オペレーションの説明に含まれている情報の詳細については、「*説明：オペレーションを有効に利用するための情報*」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

1. 目的のオペレーションを **[作成]** ペインで開きます。

   または

   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、**[オペレーションを開く]** をクリックします。
2. オペレーションの **[プロパティ]** シートで、**[説明]** 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

ドキュメントの生成機能は、フロー作成者がフローまたはオペレーションの **[説明]** 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）**[ドキュメントの生成]** 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。 **[ドキュメントの生成]** 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トを生成するには*」を参照してください。
- **ドキュメントの生成** 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。 **ドキュメントの生成** オペレーションの詳細については、「*Central ユーザによるドキュメントの生成の有効化*」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 ドキュメントの生成*」を参照してください。

注: [統合]、[Operations]、[ユーティリティ　オペレーション] フォルダーは Central では非表示です。これらのフォルダーのコンテンツに関するドキュメントを生成するには、**非表示フォルダーを使用したドキュメントの生成**オペレーションを使用するフローを作成する必要があります。

フローのドキュメントを生成するには

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、**[ドキュメントの生成]** をポイントしてから **[標準形式]** をクリックします。

   または

   [ドキュメントの生成] または **[非表示フォルダーを使用したドキュメントの生成]** オペレーションのいずれかを使用するフローを作成します。

   これらのオペレーションの使用の詳細については、「*Central ユーザーによるドキュメント生成の有効化*」を参照してください。

   カスタム形式を使用してドキュメントの生成出力をカスタマイズする方法については、「*カスタム形式のドキュメントの生成*」を参照してください。

3. **[出力ディレクトリの選択]** ダイアログで HTML ファイルの出力先を指定し、**[保存]** 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。

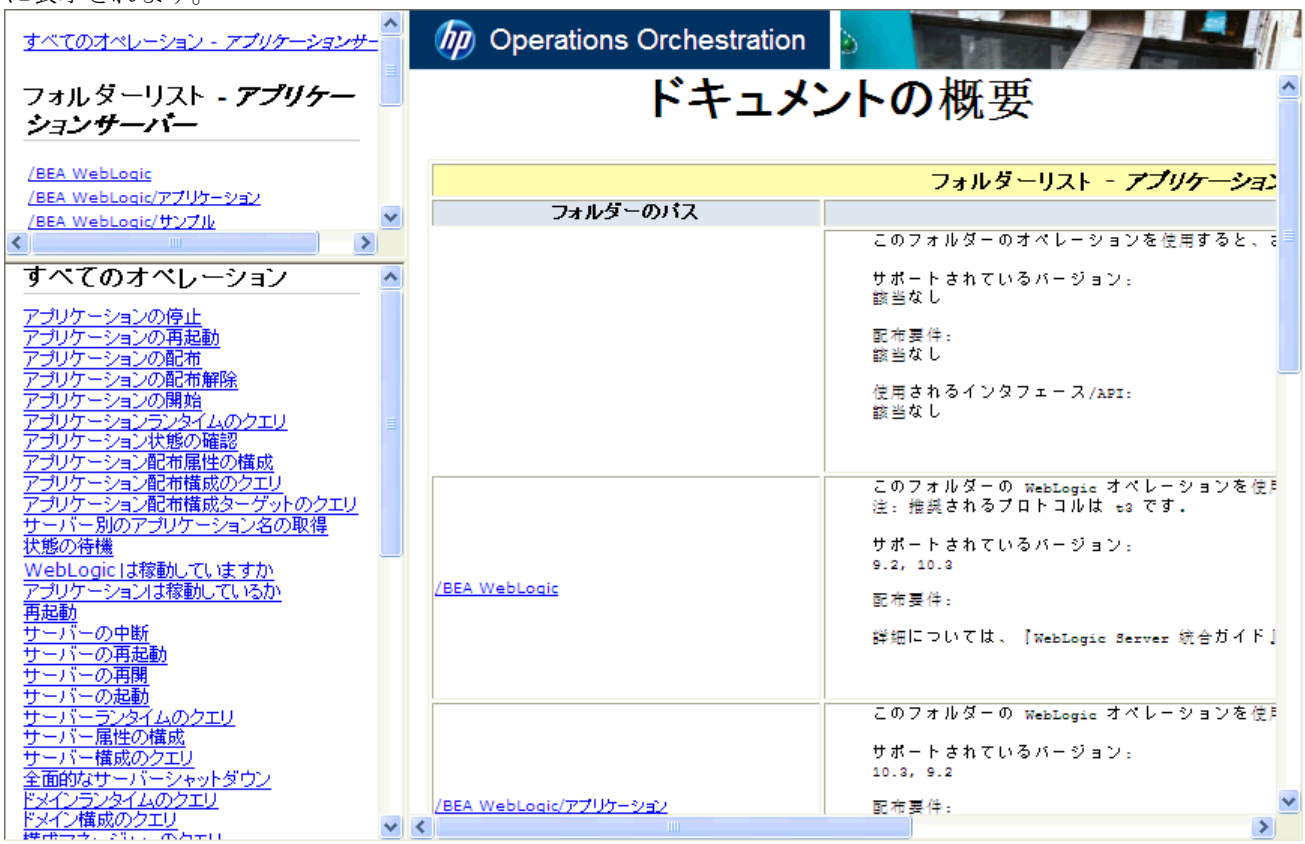


図 32 - アクセラレーターパックのアプリケーション サーバーに関して生成されたドキュメント

4. サブフォルダー内のオペレーションを表示するには、どちらかの **[フォルダーリスト データベース]** 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。

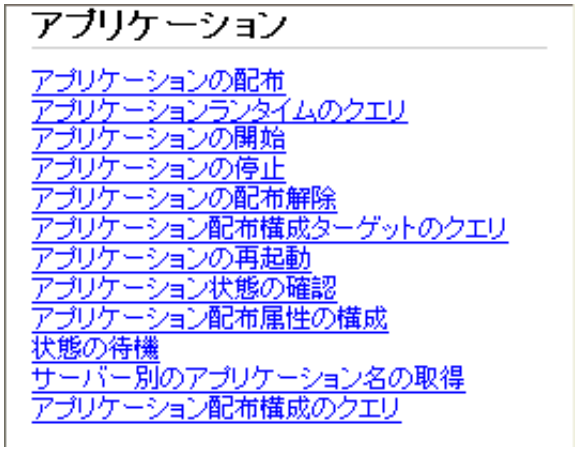


図 33 - アクセラレーターパック / データベース / Microsoft SQL Server / ヘルスチェック / サブフォルダー内のオペレーション

5. 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トを表示するには、次のいずれかの操作を行います。
   - **[すべてのオペレーション - <フォルダー名>]** 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、**[サーバーの再起動]** というフローに関するドキュメントおよびフロー図です。

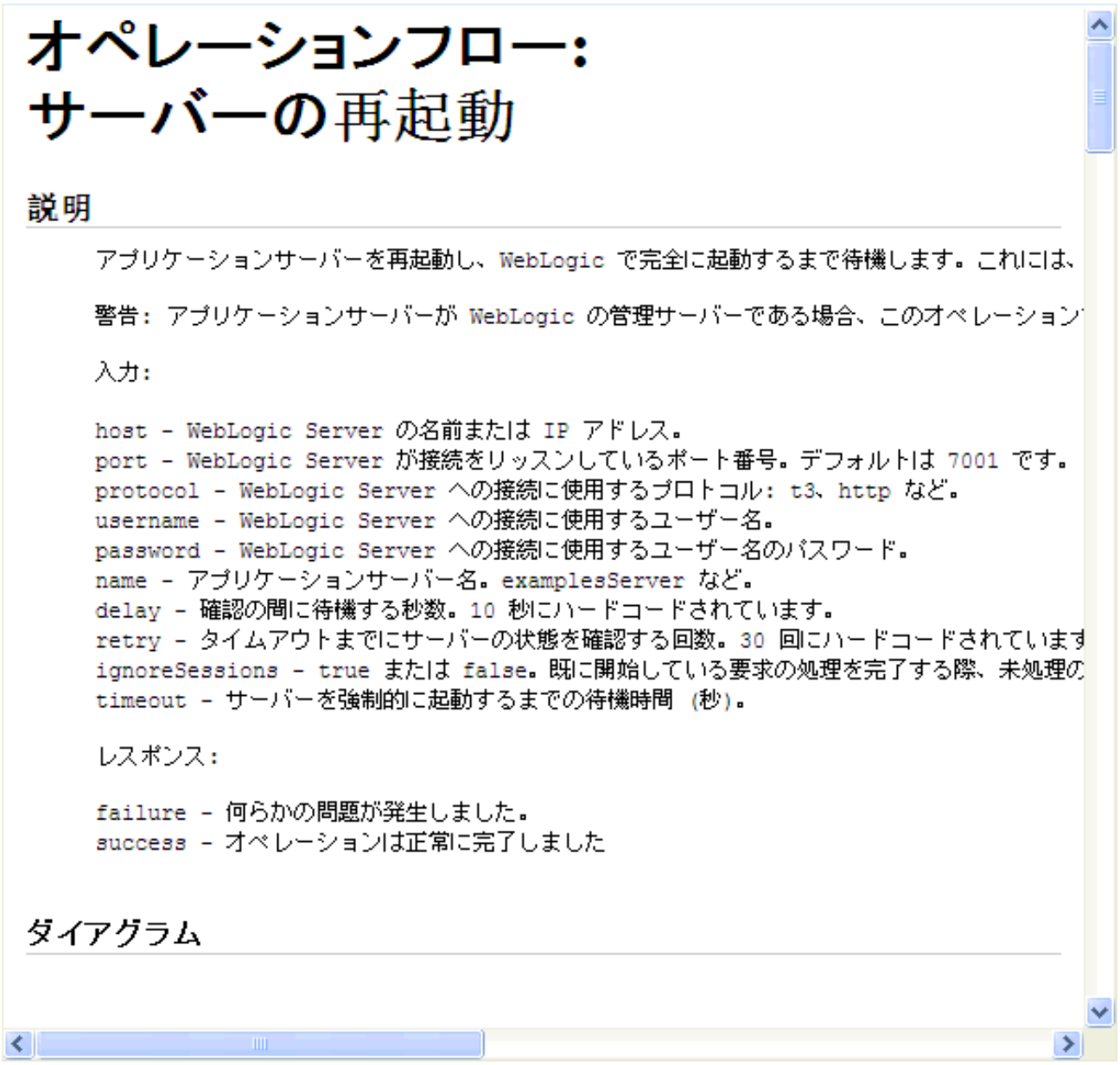


図 34 ― [サーバーの再起動] フロー、およびその入力とレスポンスの説明

下方向にスクロールすると、フロー図があります。

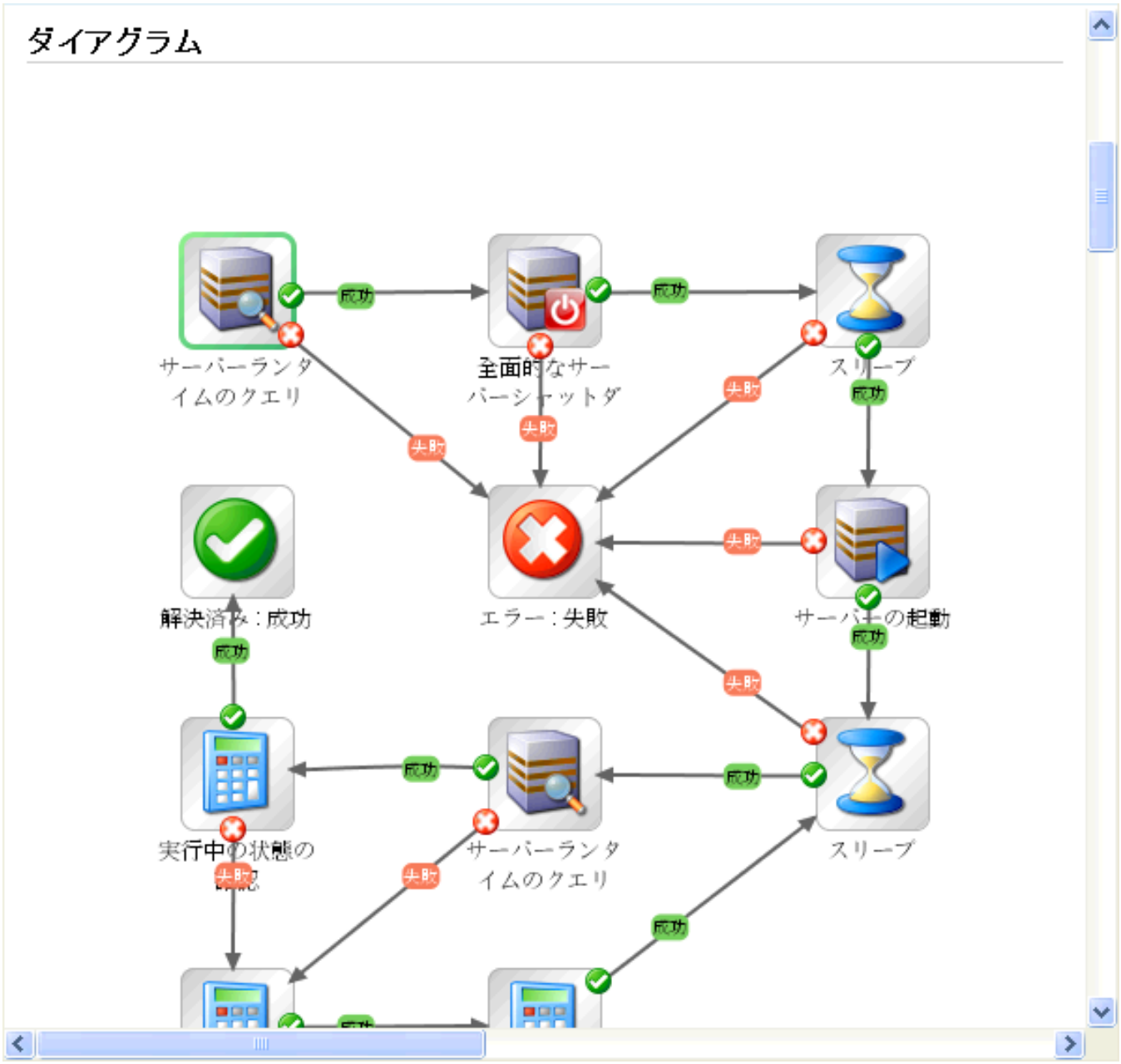


図 35 – [サーバーの再起動]フロー図

フロー図の下には、入力とレスポンスの繰り返し一覧、さらに次のようなフローのステップの説明があります。

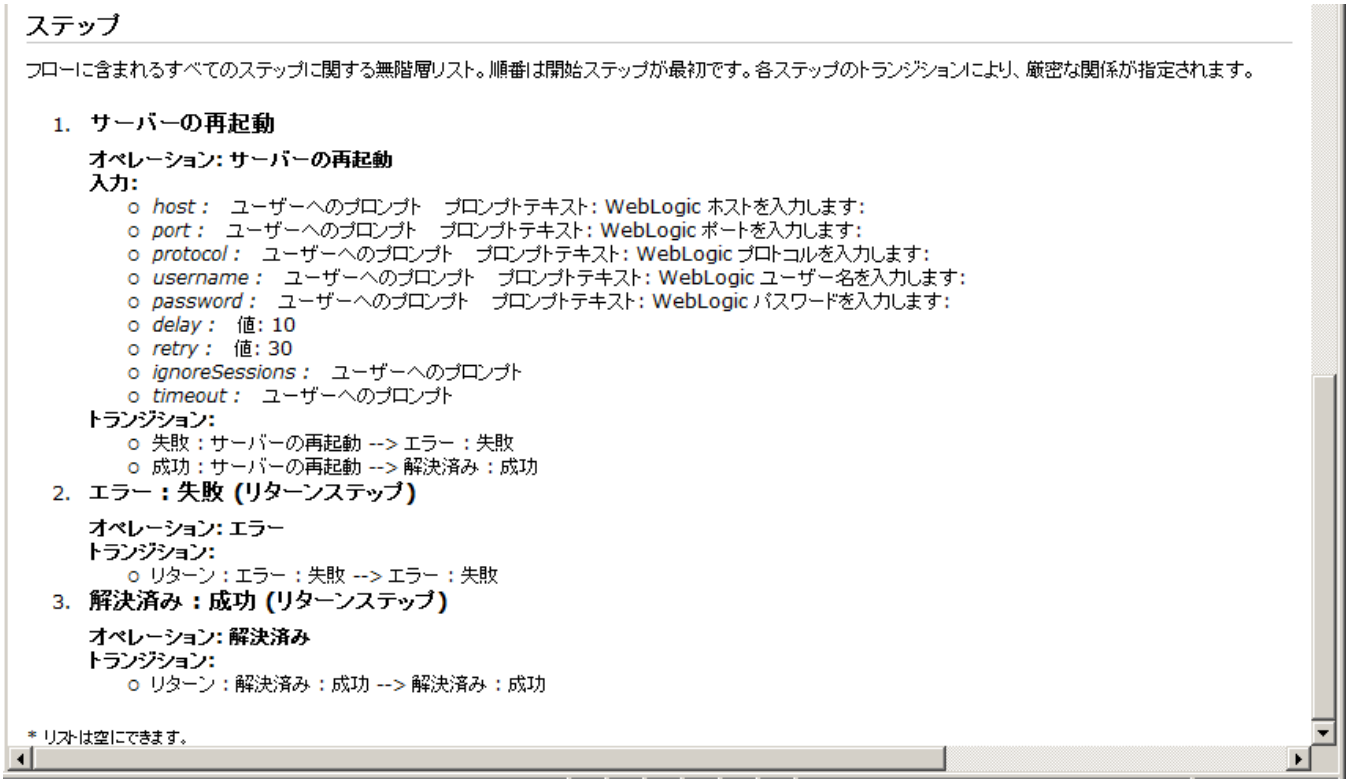


図 36 - フローのステップに関する詳細情報

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トの生成の有効化
カスタム形式のドキュメントの生成

Central ユーザーによるドキュメントの生成の有効化

ドキュメントの生成 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ト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、**[ドキュメントの生成]** 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トを見ることができます。

注: 次の手順では、フローの作成方法、オペレーションを基にしたステップの作成方法、入力用のエディターを開く方法を理解していることを前提としています。 これらの方法を理解していない場合は、以下を参照してください。

- *フローの作成*
- *フローへのステップの追加*
- *入力の値の取得方法の指定*

ドキュメントを生成するフローを作成するには

1. フローを作成し、作成したフローで、**[ドキュメントの生成]** または **[非表示フォルダーを使用したドキュメントの生成]** オペレーション（**統合 / Hewlett-Packard / Operations Orchestration** 内にある）を基にステップを作成します。
2. 新しいステップのインスペクターで、**doc_TargetDir** 入力用にエディターを開きます。

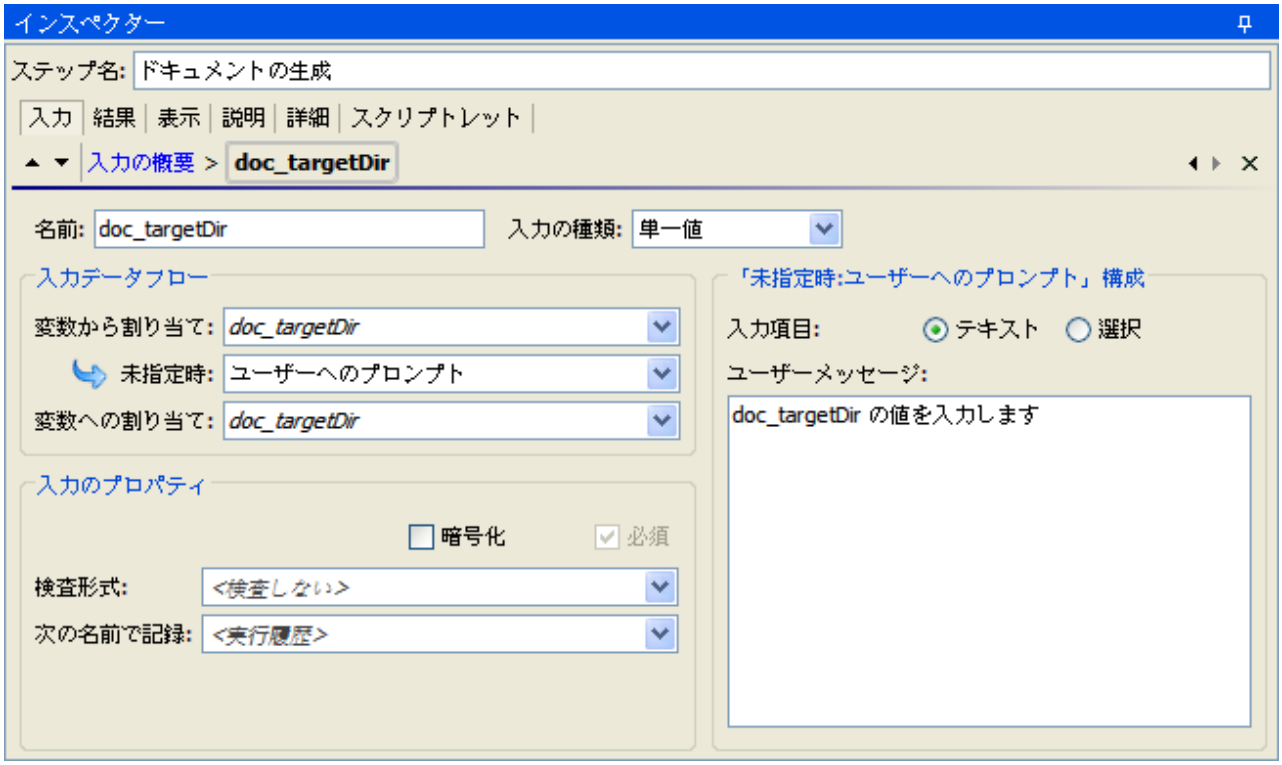


図 37 - doc_targetDir 入力エディター

doc_targetDir 入力には、ドキュメントが生成される場所（ターゲットディレクトリ）を指定します。

3. **[未指定時]** リストボックスで、次のいずれかを選択します。
   - **ユーザーへのプロンプト**

     フローにより、ターゲットディレクトリへの入力を促すメッセージが表示されます。これにより、ユーザーは生成されたドキュメントの出力フォルダーを選択できます。
   - **定数を使用する**
     - **[定数を使用する]** を選択する場合、**[定数値]** ボックスにターゲットディレクトリパスを入力します。

     定数値を指定すると、フローはユーザーの入力なしで実行できます。したがって、フローのスケジュールや Central の外からのフローの開始が可能になります。
4. **[libraryPath]** 入力を開くには、**[入力の概要]** の横にある下向き矢印（▾）をクリックします。

   この入力では、ドキュメントの生成対象となる リポジトリ内のフォルダーを定義します。
5. **[変数から割り当て]** リストボックスで、***[libraryPath]*** を選択します。
6. **[未指定時]** リストボックスで、次のいずれかを選択します。
   - **ユーザーへのプロンプト**

     フローにより、ドキュメントの生成対象となるフォルダーの指定を促すメッセージが表示されます。
   - **定数を使用する**
     - **[定数値]** ボックスに、リポジトリ内のフォルダーへのパスを入力します。

       これは、**[リポジトリ]** ペインの **[Library]** フォルダーの下のフォルダーへのパスです。 このパスは必ず **/Library/**...から始まります。

注：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トは Central ユーザーには表示されません。

7. 作成したステップをフローの開始ステップにします。
8. 必要なその他の機能を追加して、フローを完成させます。

注：**[ドキュメントの生成と送信]** フローは便利な機能を追加します。

カスタム形式のドキュメントの生成

Studio には、生成されたドキュメントの表示を制御する .html 形式の .vm テンプレートのセットが付属しています。 これらのテンプレートは、OO ホームディレクトリの **\Studio\extra\template** サブディレクトリ内に保存されています。

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ト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トの形式をカスタマイズするには、 .vm テンプレートのコピーの HTML ファイルおよびフレームの外観を編集します。 .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（ラベル付きフォルダー）
- **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。

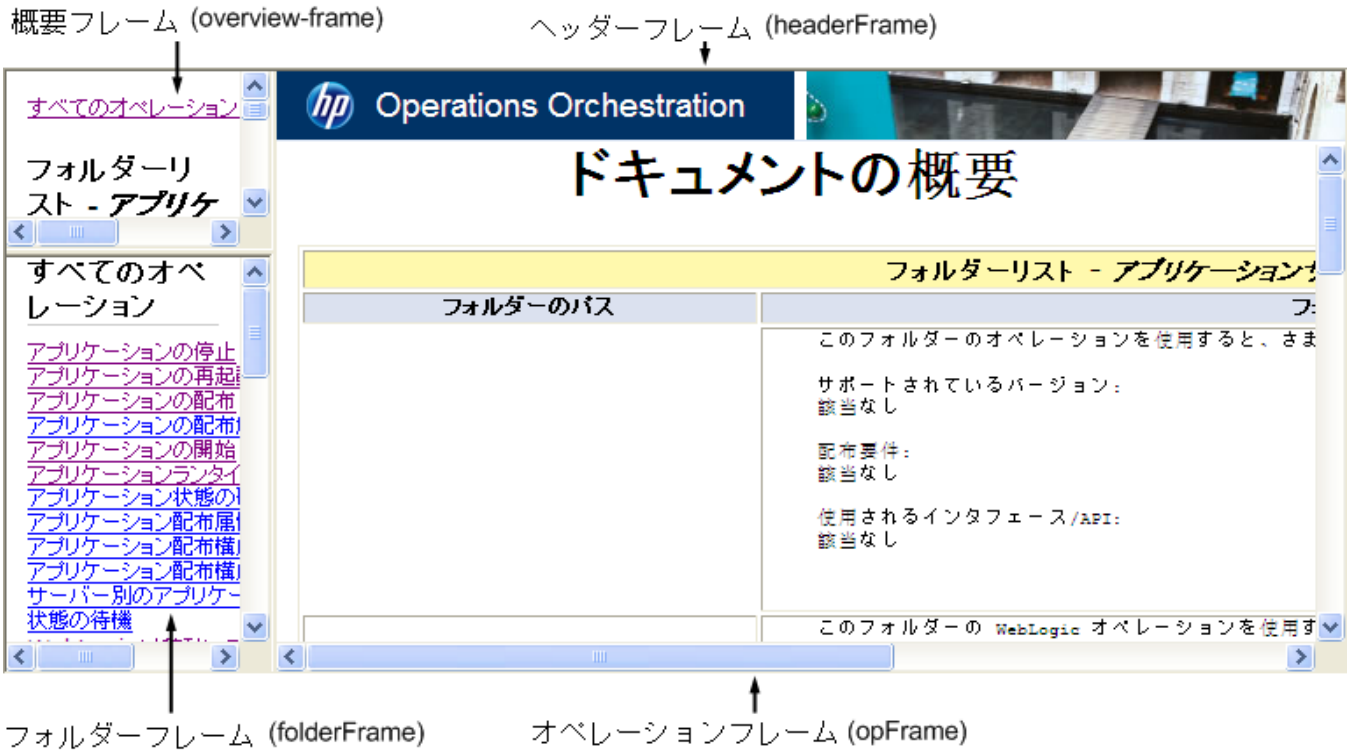


図 38 - ドキュメントの生成のフレームセット

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

ドキュメントの生成 の出力構成
ドキュメントの生成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 **[ドキュメントの生成]**、**[カスタム形式]** の順にクリックしたときに、**[ドキュメント生成テンプレートの選択]** ダイアログボックス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、「*ドキュメントの生成のテンプレートの編集*」を参照してください。

Folder_template.vm
ルートのテンプレート。フレームセットを生成し、以下のテンプレートを呼び出してそのフレームセットに組み込みます。

- All_folders_template.vm
  対象のフォルダーのサブフォルダーのリストを生成し、**overview-frame**（左上）に配置します。
- All_ops_template.vm
  すべてのオペレーションのリストを生成し、**folderFrame**（左下）に配置します。
- Header.html
  ヘッダーを **headerFrame**（右上）に配置します。
- Folder_overview_template.vm
  1 つ以上のオペレーションに関する情報を生成し、**opFrame**（右下）に配置します。

All_folders_template.vm
フォルダーのコンテンツの表を生成します。

- Header.css
  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トの表を生成します。

- 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トを生成します。

- 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トを生成します。

- 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・パッカードのバナー。

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・パッカードのバナー用のスタイルシート。

Hp_steps_307x39.jpg
ヒューレット・パッカードのバナー用の画像。

Logo_hp_smallmasthead.gif
ヒューレット・パッカードのバナー用のロゴ。

ドキュメントの生成の出力構成

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
        - ...

ドキュメントの生成テンプレートの編集

ドキュメントの生成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 **[ドキュメントの生成]** コマンドを使用して **[カスタム形式]** を選択したときに、**[ドキュメント生成テンプレートの選択]** ダイアログに表示されます。

   [ドキュメント生成テンプレートの選択] 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 [ドキュメント生成テンプレートの選択] 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、**[ドキュメントの生成]** コマンドを使用したときに以下のテンプレートが **[ドキュメント生成テンプレートの選択]** 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「*ドキュメントの生成の出力構成*」を参照してください。

カスタム形式のドキュメントを生成するには

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、**[ドキュメントの生成]** をポイントしてから **[カスタム形式]** をクリックします。
5. **[ドキュメント生成テンプレートの選択]** ダイアログボックスが表示されたら、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。
6. **[ルートファイル名の指定]** ダイアログボックスで、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. **[ファイル名]** ボックスに開始ページのファイル名を入力して、**[保存]** 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。

ドキュメントが生成されます。

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により、その使用方法や実装方法について理解を深めることができます。

重要： フローまたはオペレーションを変更すると、そのフローまたはオペレーションを他のフローで使用している場合そのフローを中断させる可能性があります。したがって、独自にオペレーションまたは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他のフローで使用されているかを確認するまでは変更しないでください。

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. **リポジトリ**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、**[参照]** をクリックしてから **[これを使用する項目は?]** をクリックします。

   または

   オペレーションまたはフローの参照先を表示するには、**[参照]** をクリックしてから **[これが使用する項目は?]** をクリックします。

[参照] 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: **[参照]** 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、**[参照]** 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

編集が必要な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 の以下のフォルダーのフローおよびオペレーションについてはこのようにしてください。 **[リポジトリ]** ペインでロック付きのフォルダーアイコン（ ）で示されているとおり、これらのフォルダーのコンテンツは変更できません。 これらのフォルダーのコンテンツを編集する場合は、コピーを作成し、そのコピーを編集する必要があります。

- アクセラレーターパック
- 統合
- Operations
- ユーティリティ　オペレーション

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（**[統合]** 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. リポジトリで、コピーする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. **[編集]** をポイントしてから、**[コピー]**、**[ディープコピー]**、**[複製]** のいずれかをポイントします。
   - 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）。
   - **[ディープコピー]** をクリックすると、フローだけでなくそのフローが使用するすべてのオペレーションもコピーされます。
   - 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。
3. **[コピー]** 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（**[統合]** 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. **[名前の変更]** 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、**[ブックマーク]** パネルに追加しておくと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 **[ブックマーク]** パネルに追加すると、フローキャンバスの右クリックメニューにも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。

[ブックマーク] パネルには、フロー用とオペレーション用の 2 つのセクション（*シェルフ*）があります。

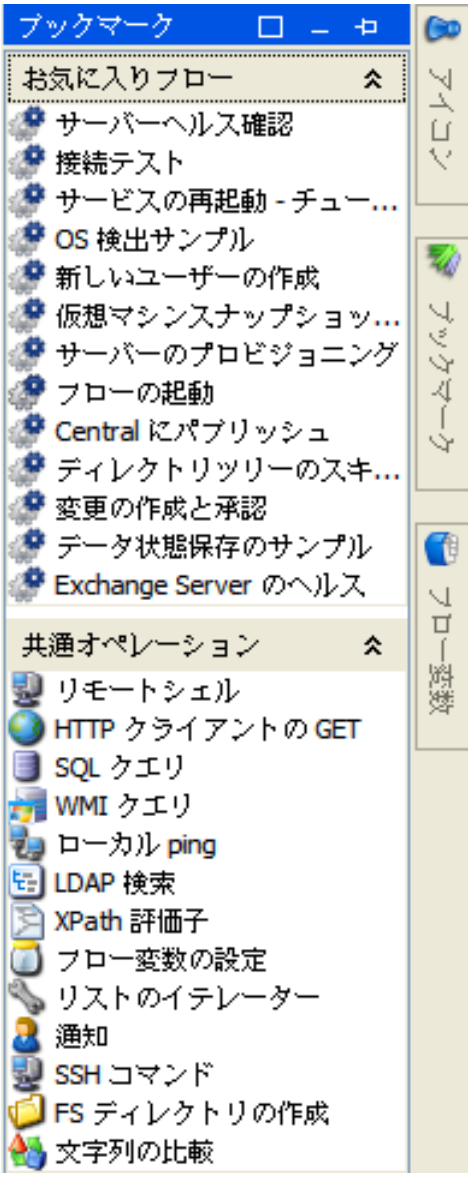


図 39 - [ブックマーク] ペイン — フローセクションとオペレーションセクション

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 [ブックマーク] ペインに追加するには

1. **[ブックマーク]** タブをクリックして **[ブックマーク]** パネルを開いてから、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**のアイコン（⊣）をクリックします。
2. **[Library]** ボックスまたは **[検索]** ボックスの結果から **[ブックマーク]** パネルの適切なシェルフにフローまたはオペレーションをドラッグします。

ブックマークのシェルフ： 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. **[ブックマーク]** パネルのシェルフの 1 つ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。

ブックマークのシェルフ： 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. **[ブックマーク]** パネルのシェルフを表示するには、目的のシェルフのタイトルバーで右クリックし、**[表示]** 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、**[ブックマーク]** パネルの何もない場所で右クリックし、**[すべて表示]** をクリックします。

   または

   [表示] 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、**[上に移動]** または **[下に移動]** 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]** フォルダー内のオブジェクト）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. **[リポジトリ]** メニューで、**[削除履歴の表示]** 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。

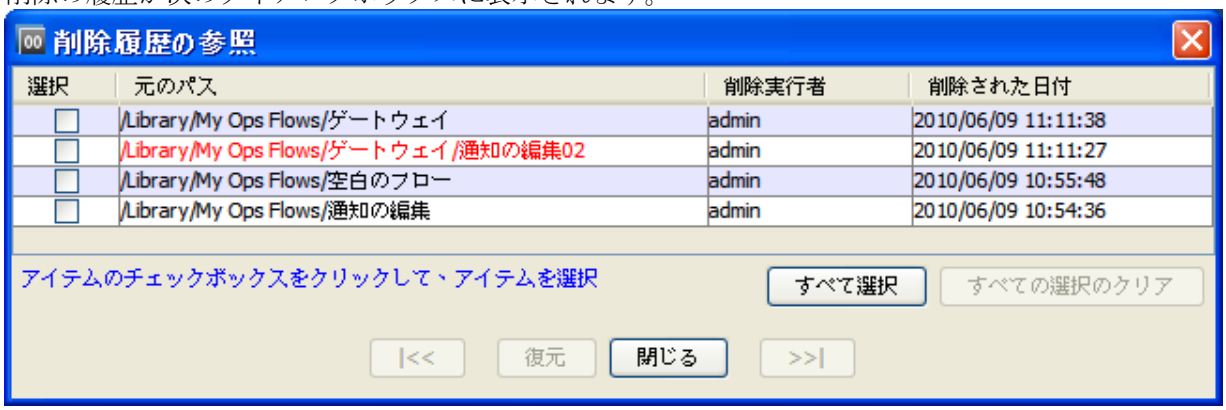


図 40 ― 削除の履歴

2. 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します。
3. 復元するオブジェクトの横にある **[選択]** チェックボックスをオンにして、**[復元]** 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。
   - 作成したオブジェクトを**[すべての選択のクリア]** する。

4. 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。

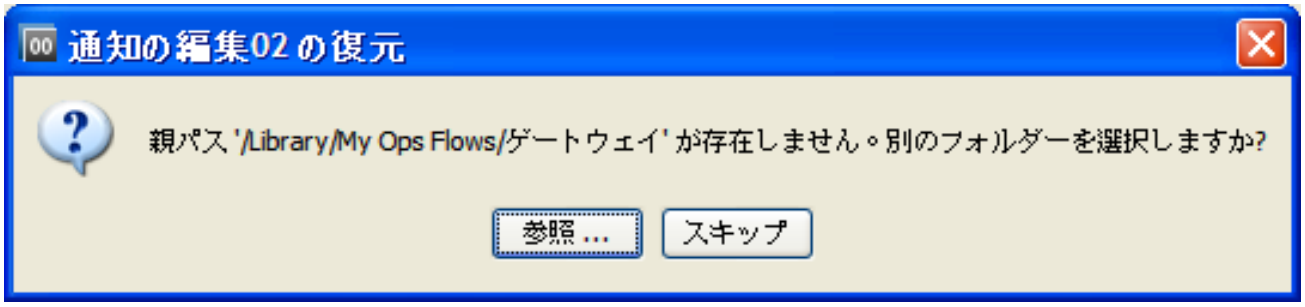


図 41 - 親フォルダーが削除されている場合のエラー

- オブジェクトの復元を中止する場合は、**[スキップ]** をクリックします。

  または

  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には

  i. **[参照...]** をクリックします。
  ii. 新しいフォルダーに移動し、選択します。
  iii. **[OK]** をクリックします。

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、オブジェクトを削除した後に移動されていた場合、オブジェクトは新しい場所にあるフォルダーに復元されます。 ただし、フォルダーは移動後にチェックインしておく必要があります。そのため、オブジェクトを復元する前にフォルダーをチェックインすることを促すメッセージが表示されます。

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もありません。

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。

- **[アクセラレーターパック]**

  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。

  アクセラレーターパック の最上位にあるフローの多くは、完全なヘルスチェック、問題選別、診断、修復機能を備えています。

- **[統合]**

  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。 入力はすべて、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] フォルダーおよびそのサブフォルダー内のフローは、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、**[アクセラレーターパック]** フォルダー内のフローです。

- **[ユーティリティ　オペレーション]**

  このフォルダーには、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
   - **[ブックマーク]** パネル

   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
   a. フロー図で、目的のステップを右クリックします。
   b. **[名前の変更]** 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ます。
   c. 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*.
   - 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。 詳細については、「*ステップのコピー*」を参照してください。

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。

これらのオペレーションまたはフロー一覧を参照するには、次の手順を使用します。 以下を参照してください。

お気に入りフローまたは共通オペレーション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 **[挿入]** をポイントしてから、**[お気に入りフロー]** または **[共通オペレーション]** 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。

   または

- **[ブックマーク]** タブをクリックし、表示された **[ブックマーク]** パネルで、**[お気に入りフロー]** 内のフローまたは **[共通オペレーション]** 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 **[開始ステップの設定]** を選択します。

ステップのコピー

既存のステップをコピーすることで同一のフローまたは別のフロー内に既存のステップのコピーを作成し、このコピーしたステップを編集できます。

このようにすることで、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、**[コピー]** 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、**[貼り付け]** 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 **[設計]** 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 **[設計]** 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、ステップのインスペクターを開きます。

ステップのインスペクターを開くには

1. **[ステップ]** をダブルクリックします。

   または

   [ステップ] を右クリックし、右クリックメニューから **[プロパティ]** をクリックします。

   インスペクター が次のような状態で開きます。

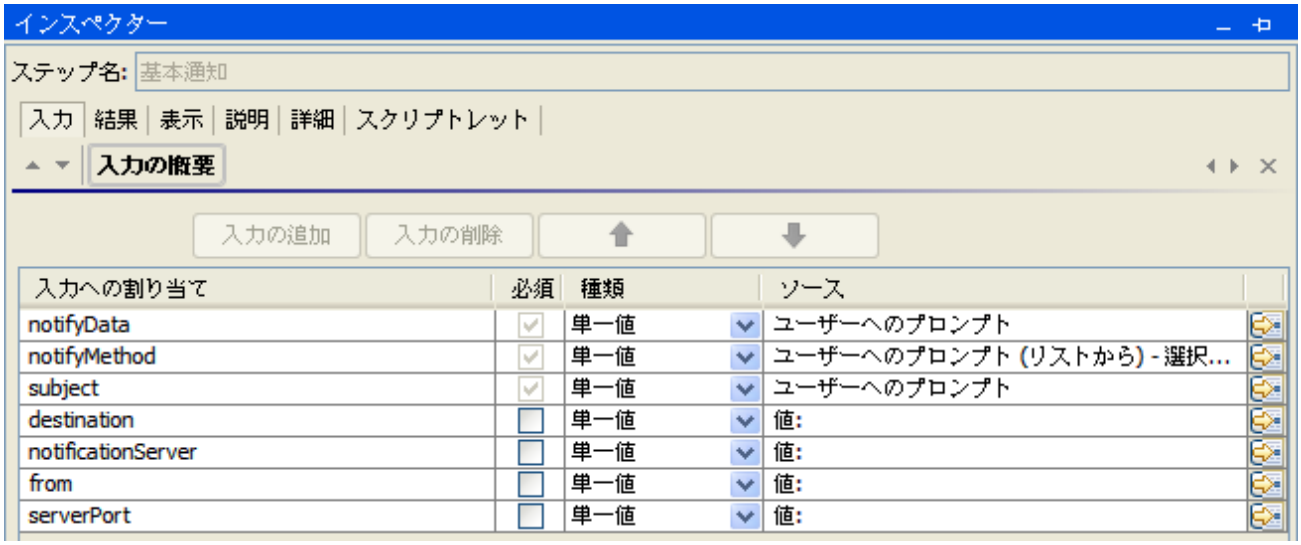


図 42 - ステップのインスペクター

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インスペクターを閉じたり開いたりする必要がないようにインスペクターを開いたままにするには、インスペクター のタイトルバーの右端にあるピンのアイコン（）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

これから説明する「表示」とは、ユーザー入力から値を取得するステップ入力によってユーザーに表示されるテキストとは異なるものです。 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、このステップが必要とする事項を説明するプロンプトを作成することにより、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. ステップのインスペクターを開いて、**[表示]** 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、**[このステップの実行前に常にユーザーに確認する]** チェックボックスをオンにします。
3. 次の各ボックスを使用してプロンプトを作成します。
   - **[プロンプトタイトル]** ボックスでプロンプトのラベルを作成します。
   - **[プロンプトの幅]** ボックスと **[高さ]** 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。
   - **[プロンプトテキスト]** ボックスにユーザーへのメッセージを入力します。
4. **[OK]** をクリックして変更を保存します。
   ステップには、表示プロンプトを示す青色の矢印が付きます。

図 43 - （青色の矢印で示された）表示プロンプトが付いているステップ

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。

- **解決済み**
- **エラー**
- **診断済み**
- **操作なし**

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 **失敗** リターンステップに接続されることがほとんどです。

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、**成功**レスポンスを起点とするトランジションの名前は**成功**になります。

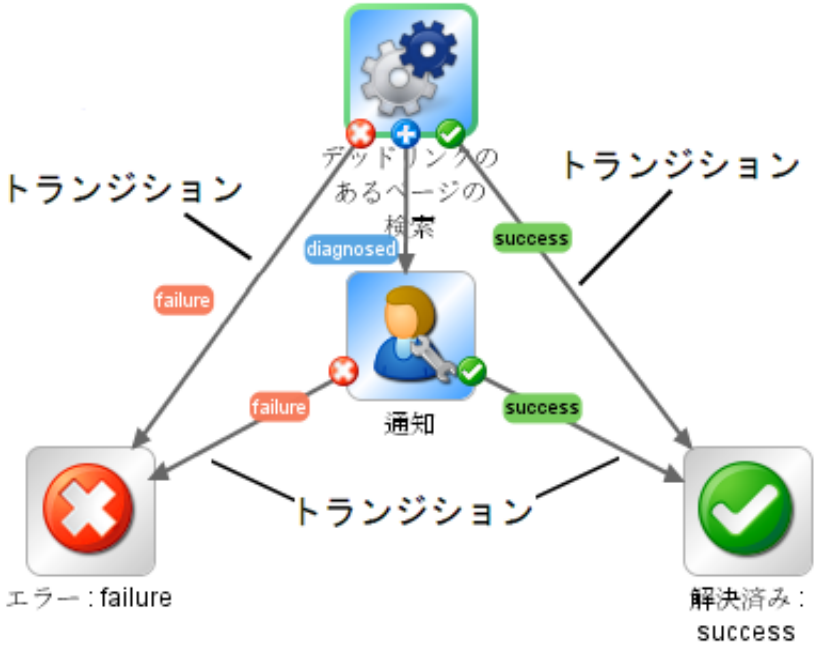


図 44 - フロー内のトランジション

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、また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。

- フロー実行の値を計算するベースとなる。

  トランジションに一定の値を割り当てると、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、このフローの実行の値にトランジションの値が加算されます。 このフローの実行全体の値は、発生したトランジションに割り当てられた値の合計です。

  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 [結果の概要] エリアの [メッセージ] 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 **[結果の概要]** 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。

結果の概要

ステップ	レスポンス	メッセージ
ターゲットシステムに対する ping の実行	✓	localhost に対する ping は成功しました。
システムのアップタイムの取得	✓	localhost の稼働時間は **SystemUpTime 7008** 秒です。
ホスト名の取得	✓	localhost は、自分自身が次の名前であるとしています: **Name Microsoft Windows XP Professional|C:\WINDOWS|\Device\Harddisk0\Partition1**
プロセッサーの数/ステータスの取得	✓	localhost には 1 個のプロセッサーがあります。 すべてオンラインです。 負荷は: **7%** でした。 **60 スレッド**が CPU を使用するために待機していました。
空き物理メモリ	✓	localhost には **FreePhysicalMemory 620964 KB の空きメモリ**があります。
実行中のプロセスリストの取得	✗	プロセスリストを取得できません。
ファイルシステム情報の取得	✓	ファイルシステム情報が取得されました: **FreeSpace,Name,Size 8108445696,C:,80015491072 ,D:,**
正しく動作しないデバイスの確認	✓	次のデバイスが存在する場合、正しく動作していませんでした:
ネットワークインタフェースエラーの取得	✗	ネットワークに対してエラーの問い合わせを実行できませんでした。
システムヘルスレポートの作成	✓	成功
ファイルに書き込み	✓	成功
解決済み : 成功	✓	リターンステップ **-Windows ヘルスのチェックと追加**

図 45 - Central の [結果の概要]

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、**${host}** という構文を使用して値を参照できます。 すると、成功レスポンスの説明はたとえば「${host} に対する ping は成功しました」のように表示されます。 「サーバー1」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「サーバー1 に対する ping は成功しました」と表示されます。

トランジションの追加

トランジションの位置の変更

トランジションの追加

次の手順では、フローを作成し 2 つ以上のステップを追加しているが、トランジションを使用してこれらのステップを接続していないことを前提としています。

トランジションを追加するには

1. Studio の **[作成]** ペイン（つまり、**設計**ビュー）でフローを開きます。
2. 次のステップに接続するステップ上で、レスポンス名、またはレスポンスの 1 つを示すアイコンをクリックして、そのレスポンスのターゲットステップにドラッグします。

   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。
3. トランジションのインスペクターを開くには、**[トランジション]** をダブルクリックします。

   または

   [トランジション] または **[トランジション名]** をクリックして選択して、フローキャンバスの下部にある **[インスペクター]** タブをクリックします。

   以下が成功トランジションのインスペクターです。

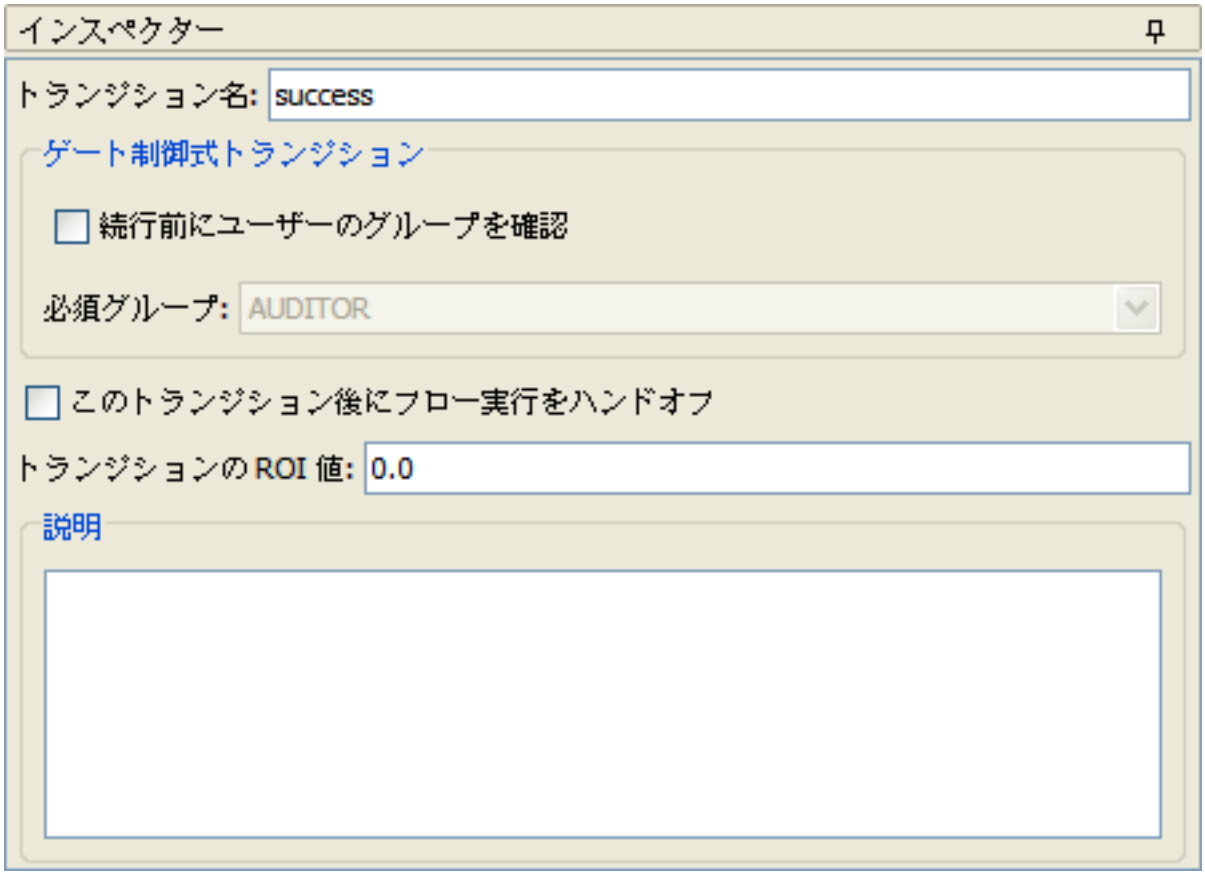


図 46 - トランジションのインスペクター

デフォルトでは、トランジション名はレスポンスの名前を由来としているため、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、**[トランジション名]** ボックスに新しい名前を入力します。
5. 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、**[ゲート制御式トランジション]** エリアで次の操作を行います。
   a. **[続行前にユーザーのグループを確認]** チェックボックスをオンにします。
   b. **[必須グループ]** 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、**[このトランジション後にフロー実行をハンドオフ]** チェックボックスをオンにします。
7. トランジションの完了をフローの値に加算する場合は、**[トランジションの ROI 値]** の横に値を入力します。
8. **[説明]** ボックスに説明を入力します。

   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。 ここで入力した内容は、Central の **[結果の概要]** エリアに表示されます。

   内容が変わる情報を格納するために、説明の中でフロー変数を使用することができます。 たとえば、サーバー名前が **servername** というフロー変数に格納されている場合、このサーバーを識別するために、「サーバー ${servername} に接続可能です」と入力できます。
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 番目のイテレーターステップの 成功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。

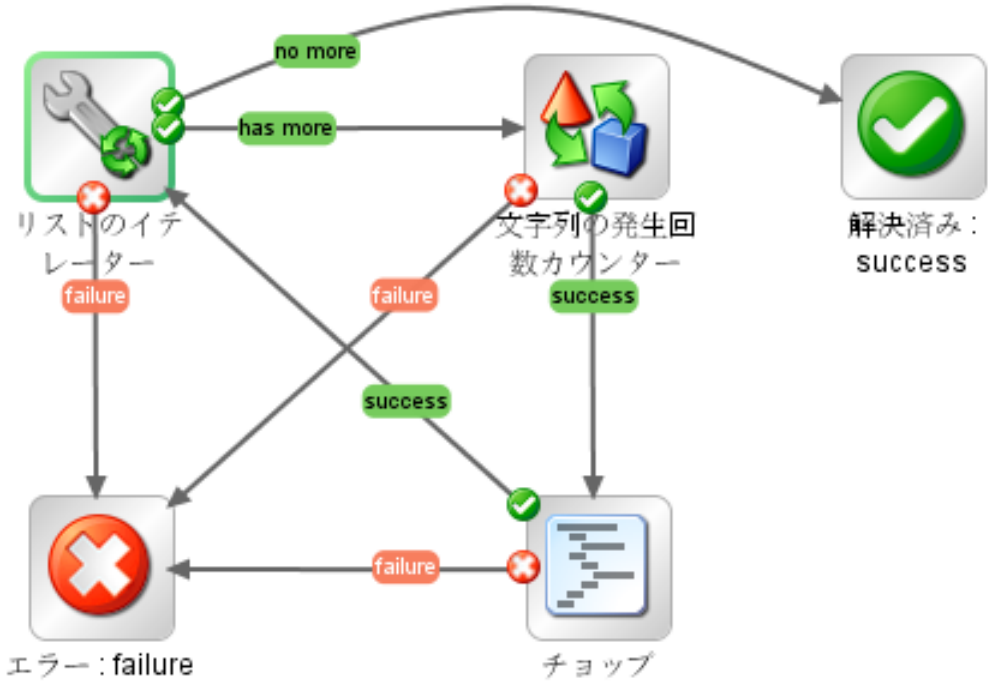


図 47 - SHIFT を押しながらトランジションをドラッグする前

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。

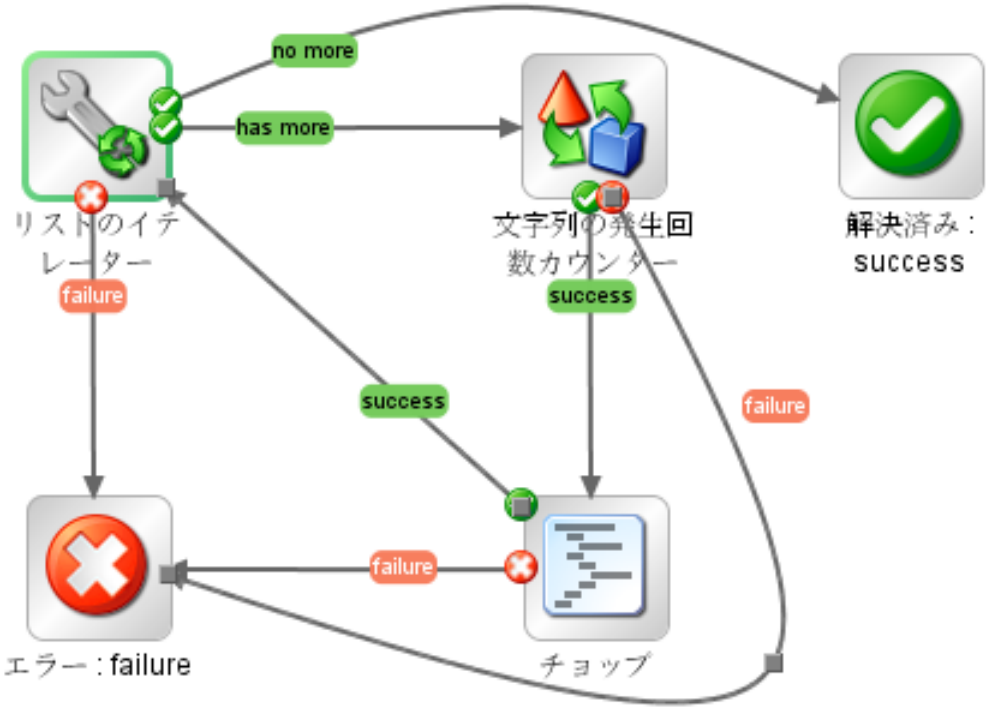


図 48 - SHIFT を押しながらトランジションをドラッグした後

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。

- 解決済み: 成功
- 診断済み: 診断済み
- 操作なし: 不明
- エラー: 失敗

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。

リターンステップの追加

リターンステップをフローに追加するには

1. フロー図ツールバーで、**[ステップパレット]** アイコン（ ）をクリックします。
   ステップパレットが開きます。

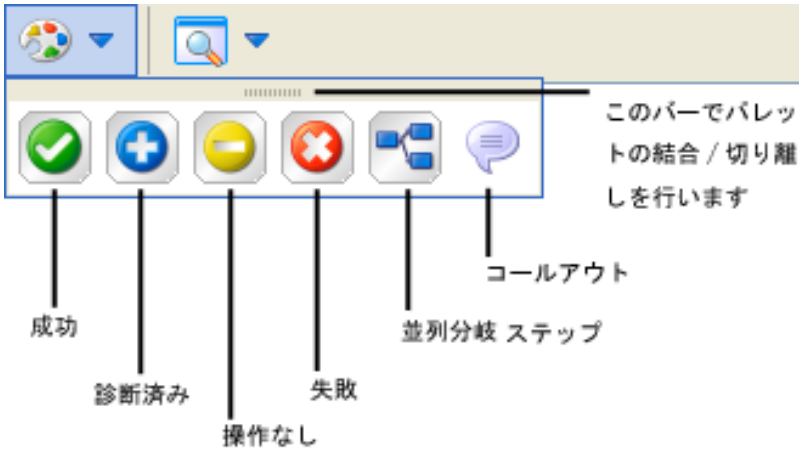


図 49 - ステップパレット

リターンステップアイコンの他にも、ステップパレットには並列分岐ステップとコールアウトを追加するためのアイコンが含まれます。 これらのアイコンはリターンステップの追加には使用しません。

2. ステップパレットから、使用するリターンステップのアイコンをフローキャンバスにドラッグします。

   各リターンステップ名で、コロン（:）の後にあるのがリターンステップのレスポンスです。

図 50 - フローのリターンステップ

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の変更

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、**エラー： 失敗** 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、**エラー： 失敗** 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 **エラー： threshold not met**）を作成することが可能です。

レスポンスの詳細については、「*レスポンス:結果の評価*」を参照してください。

リターンステップのレスポンスを変更するには

- フロー図で、リターンステップを右クリックし、**[レスポンスの選択]** をポイントしてから目的のレスポンスをクリックします。

新しいレスポンスと作成して、リターンステップに追加するには

1. フロー図で、リターンステップを右クリックし、**[レスポンスの選択]** をポイントしてから **[新規レスポンスの追加]** をクリックします。
2. ダイアログで、新しいレスポンスに名前を付けます。

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。

[ネットワークの確認] 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）

  入力の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ます。 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 **[ネットワークの確認]** 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 アドレスをホスト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 ホスト 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

フロー、ステップ、オペレーションのいずれに対して入力を作成する場合も、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プロパティ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 **[プロパティ]** タブをクリックします。

または

ステップのインスペクターを開くには、**[作成]** ペインにフローを開いた状態でステップをダブルクリックします。

2. プロパティ シートまたはインスペクターの **[入力]** タブで、**[入力の追加]** 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。

   ヒント： 既に削除しているデフォルトの入力名を正確に入力すると、入力が復元されます。 デフォルトの入力は、ステップのベースとなるオペレーションの一部として作成されたものです。 既に削除しているデフォルトの入力は、グレーのイタリック体で表示されている入力一覧に表示されます。

   新しい入力は独自の行に表示されます。以下の図ではこの行が選択されています。

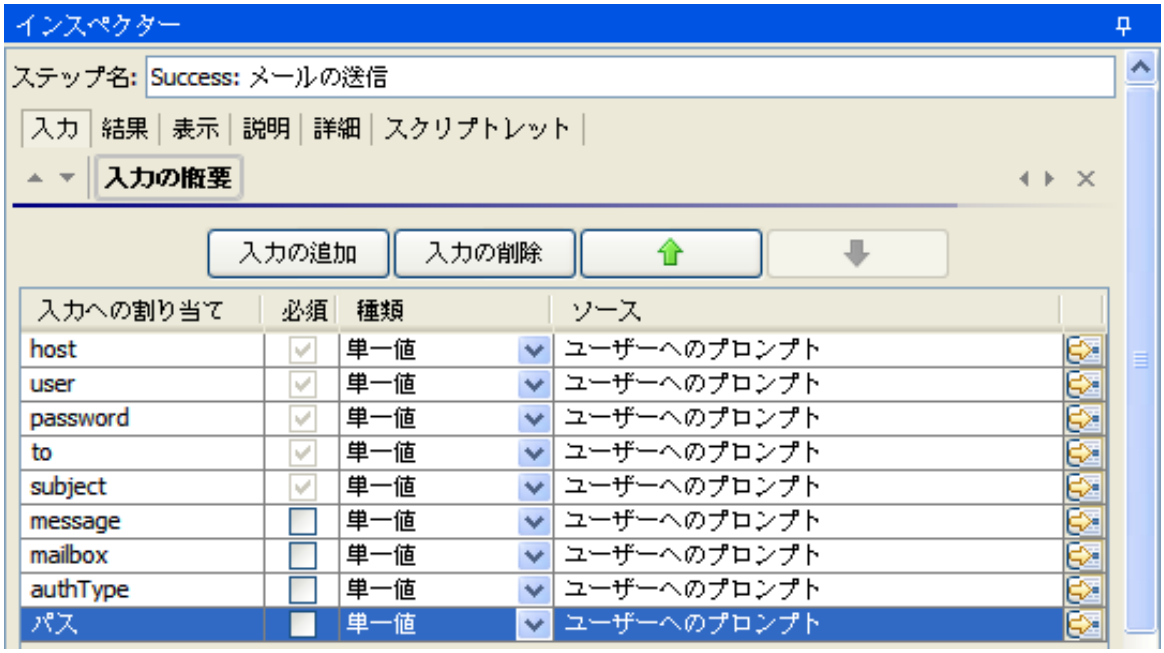


図 51 - フローの [プロパティ] シートと新しい入力

最後の列の列ラベルはステップのインスペクター内とは多少異なります。ステップのインスペクターでは、最初の列には [入力への割り当て] というラベルが付き、最後の列には [割り当て元] というラベルが付きます。 しかし、これらの列は、オペレーションの [プロパティ] シートとステップのインスペクターの両方における入力の同じコンポーネントを表します。 どちらの場合も、最初の列には値を割り当て先となる入力が示され、最後の列にはフロー変数から割り当てできない場合の入力の値のソースが示されます。

3. 入力の名前を変更する場合は、目的の行の **[入力]** 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、**[必須]** の下のチェックボックスをオンにします。

   入力の作成時にまず注意しなければならない事項の 1 つが値の割り当ての**タイプ**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 **[種類]** 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。
   - **単一値**
   - **値のリスト**
   - **資格情報**

- **未割り当て**

  未割り当て 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。

入力エディターで、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します。

6. 入力エディターを開くには、入力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 **[プロパティ]** シート、あるいはステップのインスペクターの **[入力]** 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（割り当てられている場合）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、入力が値を取得するために別の方法を定義する必要がある場合もあります。 次のようなフローの使用および設計では、入力の値の取得先としてユーザープロンプトを**使用しない**ことが要求されます。

- 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、**[入力]** 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。

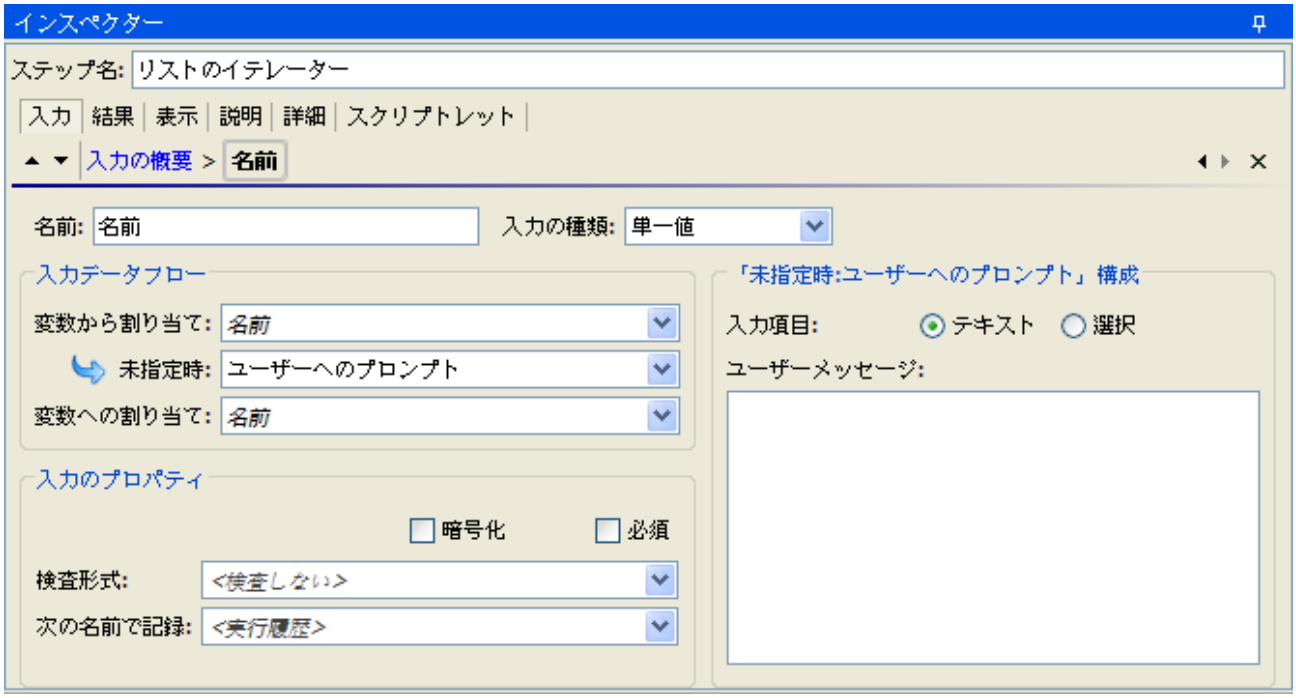


図 52 - 入力エディター

ここでは **[入力]** 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（単一値または値のリストのいずれか）
- 資格情報

2. 入力が値を取得する方法を選択するには、**[入力の種類]** リストから次のいずれかを選択します。
   - **単一値**
   - **値のリスト**

     マルチインスタンスステップまたは イテレーター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。
   - **資格情報**
   - **未割り当て**

   次に、入力の値の取得先を指定します。
3. フロー変数から入力の値を割り当てるには、**[変数から割り当て]** ボックスで、フロー変数の名前を選択または入力するか、あるいは他の使用可能なソースを含むドロップダウンリストからソースを選択します。

   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。

   または

   フロー変数を入力の値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、**[割り当てない]** を選択します。

   注:

   - このボックスには、入力の値の取得先にグローバル変数を指定することもできます。 グローバル変数の値は、フローの実行の開始時にフロー内のどの場所からでも参照できます。
   - 入力の値のソースとしてグローバル変数を指定すると、グローバル変数の名前と値を持つフロー変数が作成され、これが入力の値用の実際のソースになります。 グローバル変数の詳細については、「*グローバル変数*」を参照してください。
   - グローバル変数を指定して（選択リストまたはシステムプロパティ）、グローバル変数と同じフロー変数が存在すると、入力はフロー変数の値を取得します。
   - **[入力の種類]** が **[未割り当て]** のときに、**[変数から割り当て]** ボックスで **[未割り当て]** 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 **[変数から割り当て]** ボックスに入力の名前をそのままにしておいた場合、入力は入力と同じ名前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、（**[スケジューラー]** 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。

   [変数から割り当て] ボックスで **[割り当てない]** 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、**[未指定時]** 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。

   入力の値を暗号化して、Central でフローを実行するときに値がアスタリスクの行で表示されるようにすることができます。
4. 入力の値を暗号化する場合は、**[入力のプロパティ]** の下にある **[暗号化]** チェックボックスをオンにします。

   重要： 入力の割り当てが **[ログインしたユーザーの資格情報]** 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、**[必須]** 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 **[前のステップの結果を使用する]** を指定した場合、各実行の履歴に表示したり、ドメインタームの値として割り当てるために入力の値を記録することができます。 値を記録すると、診断や監査に利用できます。

6. 値を記録するには、[次の名前で記録] ドロップダウンリストボックスで、次のいずれかを選択します。
   - **<実行履歴>**
   - **<記録しない>**
   - 一覧内のドメインタームの 1 つ

   値の記録の詳細については、「*レポート作成用の値の記録*」を参照してください。

   ドメインタームの詳細については、「*o 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。

   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）に合わせることができます。
7. 入力の値の形式を検証する場合は、**[検査形式]** ドロップダウンリストから目的の形式を選択します。

   入力エディターの **[入力データフロー]** セクションと **[構成]** 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*
   - *'s 前のステップの結果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. 入力エディターを閉じずに別の入力を編集するときは、**[入力の概要]** の左側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. 手順「入力のデータソースを定義するには」（*入力の値の取得方法の指定*」セクション）が完了した後、**[未指定時]** リストボックスで **[定数を使用する]** を選択します。
2. **[「未指定時: 定数を使用する」構成]** セクションで、**[定数値]** ボックスに入力の値を入力します。

   または

   次の形式で変数参照を入力します。
   `${variablename}`

ヒント: null 値を指定するには、**[未指定時]** リストボックスで **[定数を使用する]** を選択した後、**[定数値]** 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. 手順「入力のデータソースを定義するには」（*入力の値の取得方法の指定*」セクション）が完了した後、**[入力の種類]** リストボックスで **[単一値]** を選択します。
2. **[未指定時]** リストボックスで **[ユーザーへのプロンプト]** を選択します。

3. **[「未指定時: ユーザーへのプロンプト」構成]** セクションの **[入力項目]** で **[テキスト]** を選択します。
4. **[ユーザーメッセージ]** ボックスに、フローユーザーに対して入力するデータの種類を説明するプロンプトのテキストを入力します。

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. 手順「入力のデータソースを定義するには」（*入力の値の取得方法の指定*」セクション）が完了した後、**[入力の種類]** リストボックスで **[単一値]** を選択します。
2. **[未指定時]** リストボックスで **[ユーザーへのプロンプト]** を選択します。
3. **[「未指定時: ユーザーへのプロンプト」構成]** セクションの **[入力項目]** で **[選択]** を選択します。
4. **[リストソース]** リストボックスで、次のいずれかを選択します。
   - **選択リスト**

     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。
     - **[名前]** 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。
   - **ドメインターム**

     ドメインタームは、手動で作成した選択リストに比べてある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。
     - **[名前]** 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。
   - **フロー変数**

     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。

     i. **[名前]** リストで、リストに含むフロー変数を入力または選択します。

     ii. **[ソース区切り文字]** 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. **[ユーザーメッセージ]** ボックスに、フローユーザーに対して入力するデータの種類を説明するプロンプトのテキストを入力します。

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. 手順「入力のデータソースを定義するには」（*n 入力の値の取得方法の指定*」セクション）が完了した後、**[入力の種類]** リストボックスで **[値のリスト]** を選択します。
2. **[入力のプロパティ]** の下にある **[入力区切り文字]** 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. **[未指定時]** リストボックスで **[定数を使用する]** を選択します。
4. **[「未指定時: 定数を使用する」構成]** セクションの **[定数値]** ボックスで次の操作を行います。

- 入力に使用する値を入力します（**[入力区切り文字]** 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 **[入力区切り文字]** 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。 たとえば、入力区切り文字がコンマ（,）の場合、リスト「赤、青」は受け付け可能なリストですが、「赤、青」は受け付けられません。

  または

  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 **[入力区切り文字]** 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**ホスト**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

  [定数値] 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 **ホスト** 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. 手順「入力のデータソースを定義するには」（「*入力の値の取得方法の指定*」セクション）が完了した後、**[入力の種類]** リストボックスで **[値のリスト]** を選択します。
2. **[入力のプロパティ]** の下にある **[入力区切り文字]** 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. **[未指定時]** リストボックスで **[ユーザーへのプロンプト]** を選択します。
4. **[「未指定時: ユーザーへのプロンプト」構成]** セクションの **[入力項目]** で **[テキスト]** を選択します。
5. **[ユーザーメッセージ]** ボックスに、フローユーザーに対してオペレーションが必要とするデータの種類を説明するプロンプトのテキストを入力します。

   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。 区切り文字のシーケンスにスペースを

指定しないときにリストの要素間にスペースを含めると、オペレーションが失敗する原因にあるので特に注意してください。

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. 手順「入力のデータソースを定義するには」（*入力の値の取得方法の指定*」セクション）が完了した後、**[入力の種類]** リストボックスで **[値のリスト]** を選択します。
2. **[入力のプロパティ]** の下にある **[入力区切り文字]** 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. **[未指定時]** リストボックスで **[ユーザーへのプロンプト]** を選択します。
4. **[「未指定時: ユーザーへのプロンプト」構成]** セクションの **[入力項目]** で **[選択]** を選択します。
5. **[リストソース]** リストボックスで、次のいずれかを選択します。
   - **選択リスト**

     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。

     [名前] 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。
   - **ドメインターム**

     ドメインタームは、手動で作成した選択リストに比べてある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。

     [名前] 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。
   - **フロー変数**

     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。

     i. **[名前]** リストで、リストに含むフロー変数を入力または選択します。

     ii. **[ソース区切り文字]** 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. **[ユーザーメッセージ]** ボックスに、ユーザーがプロンプトに表示する必要がある
   - フローが必要とするデータの種類などの情報を入力します。
   - また、リストプロンプトの場合は、複数の項目が選択可能であるかどうかを説明するテキストを入力します。

前のステップの結果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。

[前のステップの結果] を入力値のソースとして指定するには

1. 手順「入力のデータソースを定義するには」（「*入力の値の取得方法の指定*」セクション）が完了した後、入力に複数の値が必要であるかどうかによって、**[入力の種類]** リストボックスで **[単一値]** または **[値のリスト]** を選択します。
2. **[未指定時]** リストボックスで **[前のステップの結果を使用する]** を選択します。
3. 入力値が複数値の場合は、**[入力のプロパティ]** の下にある **[入力区切り文字]** 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 [前のステップの結果] 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. 手順「入力のデータソースを定義するには」（*入力の値の取得方法の指定*」セクション）が完了した後、**[入力の種類]** リストボックスで **[資格情報]** を選択します。
2. **[未指定時]** リストボックスで、次のいずれかを選択します。
   - **[ユーザーへのプロンプト]**
     - **[ユーザーへのプロンプト]** を選択したときは、**[「未指定時: ユーザーへのプロンプト」 構成]** セクションの **[ユーザー名とパスワードメッセージ]** 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。
   - **[システムアカウント]**

     [システムアカウント] を選択したときは、**[「未指定時: システムアカウント」 構成]** セクションの **[アカウント名]** 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。

     次の入力名要件に従う限り、入力はシステムアカウントの適切な部分から値を取得します。
     - 入力にシステムアカウントのユーザー名部分を割り当てるには、入力に次のいずれかの名前を付ける必要があります（大文字と小文字の要件は厳格なものなので注意してください）。

       username
       Username
       userName
       user
       User
     - 入力にシステムアカウントのパスワード部分を割り当てるには、入力に次のいずれかの名前を付ける必要があります（大文字と小文字の要件は厳格なものなので注意してください）。

       password
       DB Password
       pass
       Pass

- **[ログインしたユーザーの資格情報]**
  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。

入力形式の検証

ステップ、フロー、またはオペレーションの入力に割り当てる値が、値の使用によって要求されるものに確実に一致するように、値を入力に割り当て、フロー変数に保存される前に値をテストするための形式を指定できます。形式を指定するためにシステムエバリュエーターを選択します。

システムエバリュエーターの詳細については、「*データが正しい形式かどうかの評価*」を参照してください。

入力の値をテストするための形式を指定するには

1. オペレーションまたはフローの **[プロパティ]** シート、またはステップのインスペクターで、**[入力]** タブをクリックします。
2. 入力エディターを開くには、入力行の右向き矢印（ ）を右クリックします。

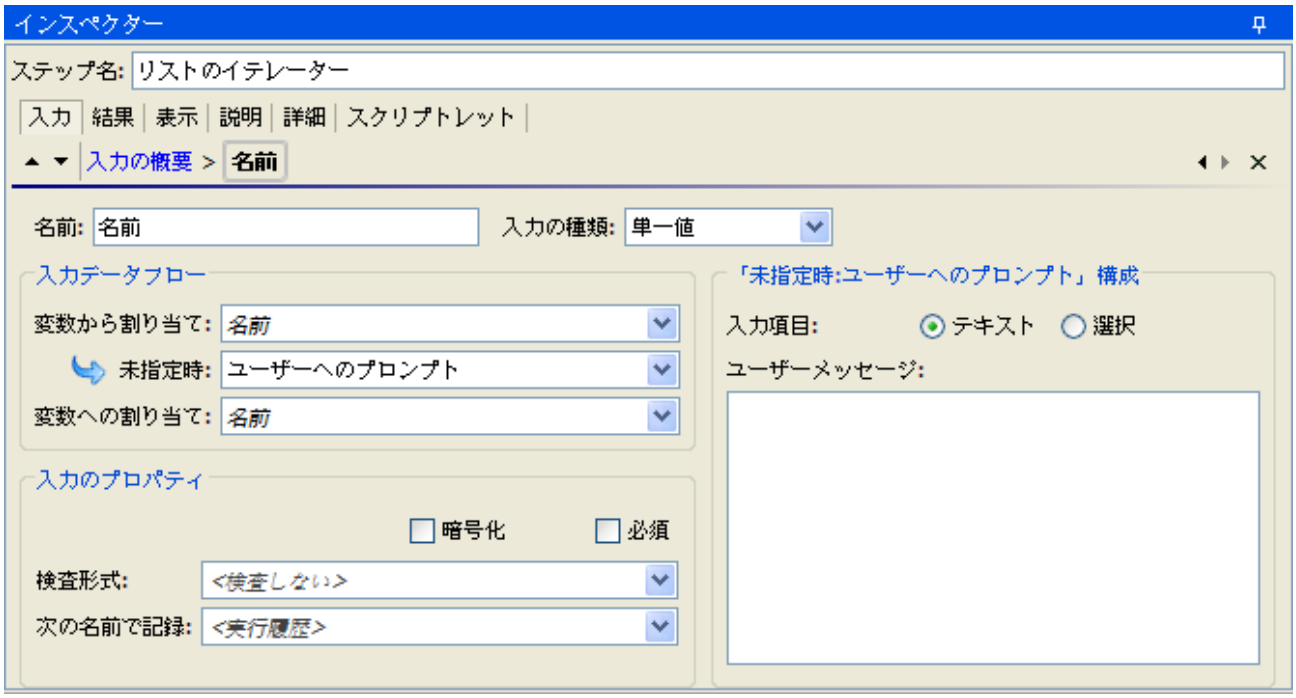


図 53 - 入力エディター

3. **[入力プロパティ]** の下にある **[検査形式]** ドロップダウンリストで、下向き矢印（ ）をクリックして一覧されているシステムエバリュエーターのいずれかを選択します。

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、**ネットワークの確認** 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 ホスト 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、入力のデータソースとして **[定数を使用する]** を選択し、**[定数値]** ボックスに次の構文を入力して変数を参照します。

  `${flowvariablename}`

  データソースを入力に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 インスペクター、あるいはフローまたはオペレーションの **[プロパティ]** シートを開きます。
2. **[入力]** タブの入力のリストで、削除する入力を強調表示して、**[入力の削除]** をクリックします。

   入力を削除すると、その入力はグレーのイタリック体で表示されます。

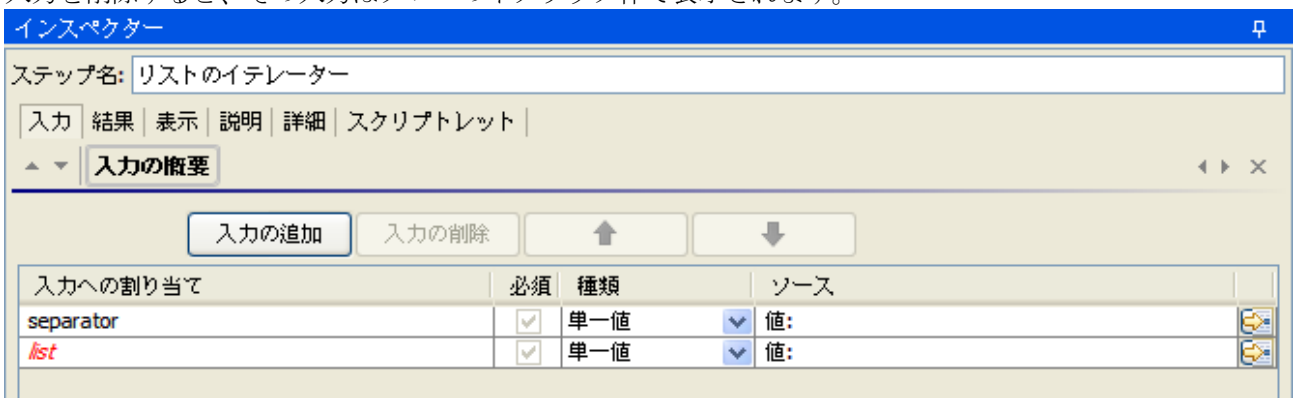


図 54 - デフォルトの入力が削除されている [入力] タブ

3. 削除したデフォルトの入力を再度追加するには
   a. **[入力の追加]** をクリックします。
   b. 表示されたテキストボックスで復元したい入力の正確な名前を入力して、**[OK]** 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 の遅延**オペレーションは、ping の平均応答時間をフィルターにより除外します。 このオペレーションに関連付けられているステップはこの平均応答時間をフロー変数 **latency** として保存し、さらにこのステップに続くトランジションがその値をユーザーにレポートすることが可能です。
- 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、（必要なフロー変数が存在しない場合は）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。

フロー変数の作成と値の割り当て

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。 ユーザープロンプト入力の値をフロー変数に割り当てると、その名前の入力を持つ後続するオペレーションは、その入力に対して別のソースを指定しないかぎり、そのオペレーションは自動的にそのサーバー名を使用します。

オペレーション、ステップ、フローの入力または結果からフロー変数を作成し値を割り当てることができます。どの場合でも作成方法は同じです。 値を割り当てるフロー変数に名前を付けると、フロー変数が作成され、割り当てる値が指定されます。

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。
- オペレーション出力の作成方法については、「*オペレーション出力*」と「*出力および結果のフィルタリング*」を参照してください。
- ステップ結果の作成方法については、「*ステップ結果*」と「*出力および結果のフィルタリング*」を参照してください。

フロー変数に入力の値を割り当てるには

1. オペレーションまたはフローの **[プロパティ]** シート、またはステップの **[インスペクター]** を開きます。
2. **[入力]** タブで、入力を選択するか、または新規作成します。
3. 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 **[変数への割り当て]** 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。

フロー変数に結果を割り当てるには

1. オペレーションの **[プロパティ]** シート、またはステップの **[インスペクター]** を開きます。
2. **[結果]** タブで、目的の結果行の **[割当先]** の下にある **[フロー変数]** を選択します。
3. **[名前]** で、フロー変数の名前を指定します。
4. **[ソース]** で、値のソースを指定します。
5. 正確な結果を取得したい場合は、結果に使用するフィルターを 1 つ以上作成します。

   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。
6. 作業結果を保存します。

スクリプトレットタイプのオペレーションでも、フロー変数を作成して値を割り当てることができます。

スクリプトレットからフロー変数を作成し、値を割り当てるには

- スクリプトレット内で、次の構文を使用してコマンドを含めます。
  `scriptletContext.putLocal("<localflowvariablename>", <value>);`
  `<value>` には、スクリプトレット内で作成されるオブジェクトまたは変数を指定できます。

スクリプトレットの作成の詳細については、「*スクリプトレット*」を参照してください。

フロー変数の管理

フローの作成で最も重要なことの 1 つに、次のようなフロー変数内のデータの保存の追跡が挙げられます。

- データが必要な場所でそのデータを使用可能にするようにフローがどのようにフロー変数を使用するか
- フロー変数そのデータをどこから取得するか

特定のステップでフロー変数が持っている値をフロー変数がどのように取得したかを見つけることは困難な場合があります。 その値を、フロー入力または前のステップの入力がフロー変数に渡したのか? その値を、前のステップの入力がフロー変数に別のフロー変数から渡したのか? 前のステップの結果がこのフロー変数に割り当てられたのか?

[フロー変数] パネルでは、ツリー構造内にこのような疑問に対する回答が用意されています。 **[フロー変数]** パネルには、現在のフローに使用されているすべてのフロー変数が示され、各フロー変数の作成日と用途が一覧されます。 フロー内のフロー変数に対して変更を加えると、その変更内容が自動的に **[フロー変数]** パネルに反映されます。

以下を行う場合、**[フロー変数]** パネルを使用すると便利です。

- フロー変数名の不適切な管理を防止する場合。 使用したくない場所でフロー変数名を間違えて再利用したり、使用したくない別のフローステップ内で間違って名前を指定するといったことを防止します。
- フロー変数の値が変更される可能性があるポイントを識別する場合。
- どのフロー変数がステップ結果またはユーザー入力から取得されたかを識別する場合。
- フロー変数が確実に正しく使用されるように、フロー変数のすべての用途を迅速にレビューする場合。
- フロー図で、フロー変数の特定の用途を探する場合。
- フロー変数を使用する入力、結果、またはスクリプトレットのエディターを開きます。

[フロー変数] パネルをさらに詳しく調べるために、「操作方法:リストの反復処理」フローの例を挙げます。 このフローはフォルダー一覧内の各フォルダー名をフロー変数に割り当てます。 参照用に Studio 内でこのフローを開くことができます。 フローは、**[操作方法フロー]** フォルダーに保存されています。

詳細については、以下のセクションのトピックを参照してください。

- *[フロー変数] パネル*
- *[フロー変数] パネルでのフィルタリング*
- *フロー図でフロー変数の使用 を特定する*

[フロー変数] パネル

作成ペインでフローを開いた場合、**[フロー変数]** パネルには各フロー変数がアルファベット順に一覧され、フローの各用途、つまり、フロー変数を使用する可能性のあるフロー内の各位置の説明が示されます。

注:

- パネル内の一覧には、現在表示されている図があるフローのステップでの用途が示されており、このフローのサブフローのステップでの用途は示されません。
- 用途の一覧には **${flowVariableName}** 参照は含まれません。 つまり、フロー変数を参照するためにフォーム **${flowVariableName}** を使用するトランジション説明や、フィルター、スクリプトレットなどのテキストは、**[フロー変数]** パネル内の用途一覧には含まれません。

したがって、[操作方法: リスト内の反復処理)] フローの場合、**[フロー変数]** パネルには、次のようにフローのフロー変数および変数の用途が一覧されます。

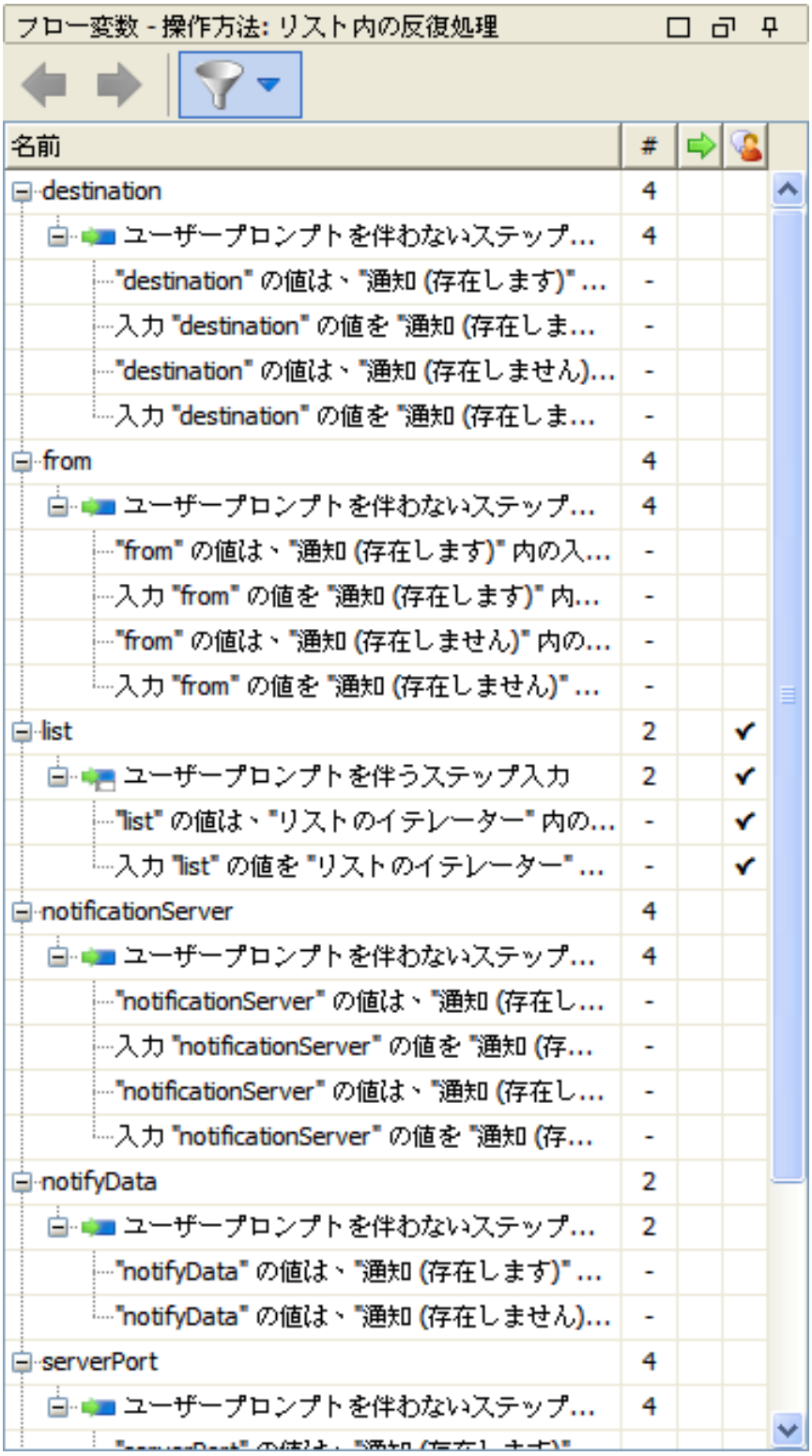


図 55 - [フロー変数] パネルの詳細。フロー変数とその用途の一覧

右の列には以下が示されます。

列	情報の内容
#	フロー内でのフロー変数の使用回数。

	フロー変数の特定の用途がフロー入力内で発生しているかどうかの有無。
	フロー変数の特定の用途がフロー変数の値をユーザー入力から取得するかどうかの有無。

[フロー変数] パネルには、各フロー変数の用途（この場合では、フロー変数 "list" の用途）について、以下に示すような情報が表示されます。

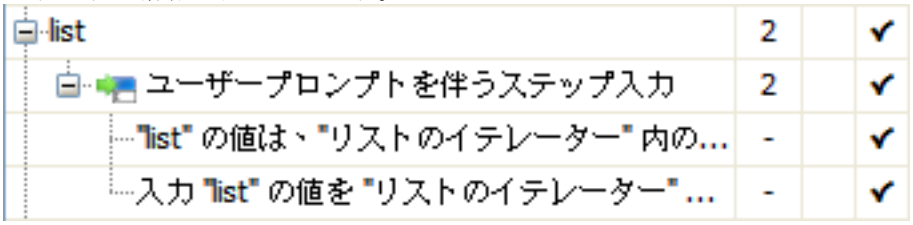


図 56 - [フロー変数] パネル: あるフロー変数の説明

フロー変数 "list" のこのような用途では、以下のようになります。

- 値が割り当てられている既存のフロー変数 "list " が存在しない場合、ユーザーに値の入力を要求するステップ入力にあります。
- フロー変数 "list" の値を "リストのイテレーター" ステップ上のステップ入力 "list" に割り当てます。
- "リストのイテレーター" ステップで、入力 "list" の値をフロー変数 "list" に割り当てます。

最後の 2 つは、（ステップ）入力へのフロー変数の値の割り当てと、フロー変数への入力の値の割り当てを区別しています。 この場合、“リストのイテレーター” ステップで、フロー変数 "list" の値がステップ入力 "list" に供給され、ステップ入力 "list" の値が再びフロー変数 "list" に割り当てられます。

[フロー変数] パネルをフローティングする方法と再度ドッキングする方法については、「*パネルのサイズ変更と再配置*」を参照してください。

[フロー変数] パネルでのフィルタリング

最も関心があるフロー変数の用途に焦点を合わせるために、パネル内に表示するフロー変数の用途を選択することができます。 フロー変数がそのデータの取得先を次のいずれにするかによってフィルターが定義されます。

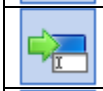
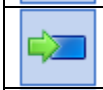
- フロー入力
- ユーザープロンプトステップ入力
- ユーザープロンプトを設定しないステップ入力
- [Result]
- スクリプトレット

フロー変数の用途をフィルタリングするには

1. **[フロー変数]** ペインの最上部で、[じょうご] アイコンまたはその横の下向き矢印（ ）をクリックします。
2. 表示されたフィルタータイプのアイコンの行で、必要に応じてアイコンをクリックして各タイプのオン / オフを切り替えます。

   [フロー変数] パネルの一覧に表示するために用途のアイコンのタイプに合わせてアイコンを選択すると、アイコンは青色の背景で表示されます。

フィルタータイプアイコン	フィルタータイプ

	フロー入力。
	ユーザープロンプトを設定したステップ入力。
	ユーザープロンプトを設定しないステップ入力
	結果 - フロー変数はステップ結果が関連付けられる。
	スクリプトレット - フロー変数はスクリプトレット内で参照される。

データソースの各タイプを切り替えると、そのタイプが表示されたり、表示から削除されます。

フロー図内のフロー変数の用途を特定する

フロー変数一覧が参照する入力を見つけるには

- 特定の用途を表示して、その用途を定義するエディターを開くには、用途インスタンスを（**[フロー変数]** パネル内で）選択します。
  - 用途がフロー入力用の場合、フローの **[プロパティ]** シートを **[入力]** タブで開きます。このとき、該当する入力のエディターが開きます。
  - 用途がステップ入力または結果用の場合、フロー図が開き、ステップが選択されます。 フロー図の下には、入力または結果のエディターが開きます。

用途を選択すると、[前へ]（ ）および[次へ]（ ）ツールバーボタンを使用して用途リスト内を上下に移動できます。

グローバル変数

グローバル変数は、グローバルコンテキストの一部であるキー名と値のペアで構成されます。したがって、すべてのフロー実行での使用または参照に常に利用できます （反対に、フロー変数はローカルコンテキストの一部です。したがって、フロー変数を作成したフローの実行でのみ使用できます）。

同じ名前を持つフロー変数とグローバル変数がある場合、その変数名への参照は、変数への値の割り当てまたはその変数の値の取得のどちらの場合も、その名前の（ローカルな）フロー変数にアクセスし、グローバル変数にはアクセスしません。

たとえば、Host Type という名前のグローバル変数があるとします。 フロー内で Host Type という名前のフロー変数を作成し、Host Type に値 “local host” を割り当てるとします。 さらに、フローの後半の Notify ステップで、ユーザーに対して Host Type の値を表示するとします。 表示される値は、“local host” になります。

入力の値の取得先をグローバル変数に指定すると、作成されるフロー変数はグローバル変数の値を持ち、その値はフロー変数から入力に提供されます。

デフォルトで、グローバル変数は **Selection Lists** フォルダーのコンテンツです。 システムプロパティ（**[System Properties]** フォルダー内に保存されている）を作成すると、システムプロパティはグローバル変数として保存されます。 システムプロパティの作成方法については、「*システムプロパティ*」を参照してください。

ヒント: すべてのグローバル変数を表示するには、フローをデバッグしてください。 グローバル変数（およびフロー変数）とその現在の値は、**[コンテキストインスペクター]** に一覧されます （**コンテキストインスペクター** の詳細については、「*フローのデバッグ*」を参照してください）。

グローバル変数の値の取得または変更

重要：グローバル変数の値を変更する際は、グローバル変数はすべてのフローの実行で利用可能であることに注意してください。したがって、グローバル変数の値を変更したことによって、そのグローバル変数の値を使用しているフローまたはオペレーションが中断することがあります。

グローバル変数とその値の取得または変更には次のようないくつかの方法があります。

- プロパティ、ステップ、フロー入力の場合

  入力の値をグローバル変数から取得したり、グローバル変数に割り当てることの影響については、「*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。
- 選択リストとシステムプロパティであるグローバル変数の値にはアクセスできます （詳細については、「*ユーザープロンプトの選択リスト*」および「*システムプロパティ*」を参照してください）。
- フローをデバッグすると、フロー内の各ステップにあるグローバル変数の値を確認できます。
- スクリプトレットを使用すると、グローバル変数の値を変更することもできます。

  （scriptletcontext.putGlobal() を使用する構文については、オペレーションまたはステップの **[スクリプトレット]** タブで、JavaScript テンプレートを挿入します）。 スクリプトレットの使用方法については、「*スクリプトレット*」とそのサブトピックを参照してください。

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 **[スケジュール]** 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各インスタンスが単一入力の複数の変種に対して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。

  a. レーン A で、ステップ A1 が「42」を threshold に渡す。

  b. レーン B で、ステップ B2 が「34」を threshold に渡す。

  c. レーン B が完了する。

  d. 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。

このセクションでは以下のトピックについて説明します。

- *レーンの順序: 開始と終了*
- *並行分岐ステップの作成*
- *並行分岐ステップとそのコンポーネントの移動またはコピー*
- *n レーンの追加*
- *レーンの複製*
- *レーンの削除*
- *レーンのサイズ変更*
- *レーンの名前変更*
- *レーンの開始ステップの変更*
- *変更ステップとのデータの移動*
- *並行分岐ステップのデバッグ*

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 **成功**しかありません。つまり、すべてのレーンが完了したことを意味します。
- 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. **[ステップパレット]** アイコン（ ）をクリックし 、**[並行分岐ステップ]** アイコン（ ）をクリックしてフロー設計にドラッグします。

   デフォルトでステップには 2 つのレーンがあります。

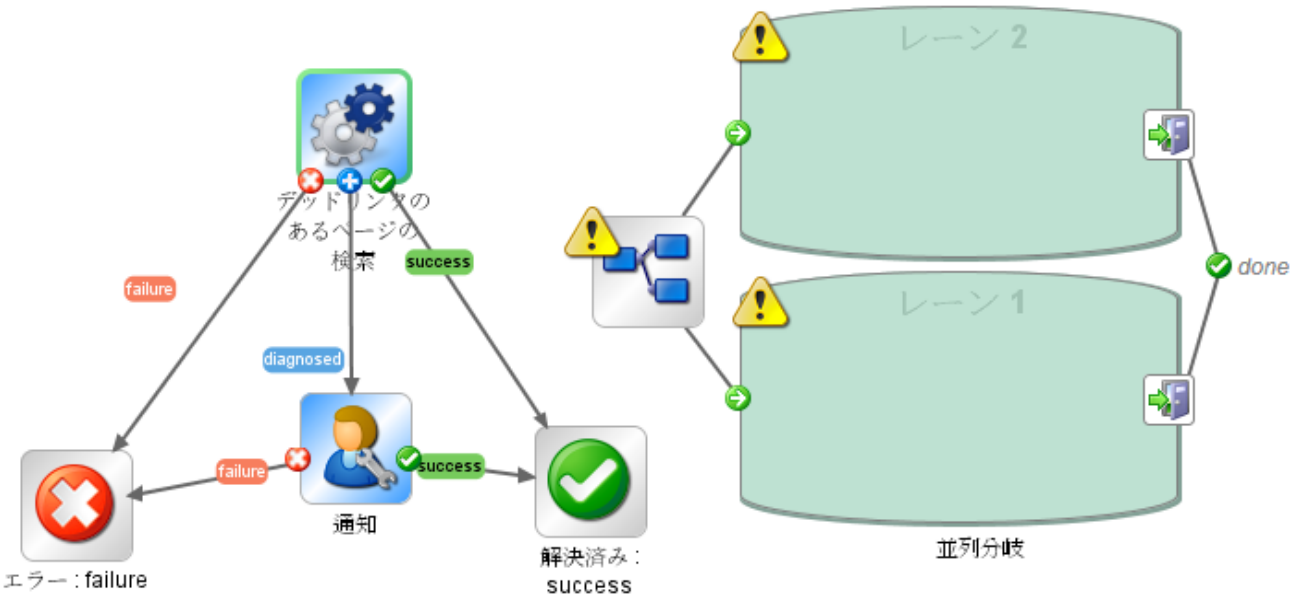


図 57 - 新しい並列分岐ステップ

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。 成功や失敗などのレスポンス、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 **[レーン-エンド]** 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 **[並列分岐ステップ]** アイコンに接続します。
   - 並列分岐ステップの完了レスポンスをフローの次のステップに接続します。

   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。

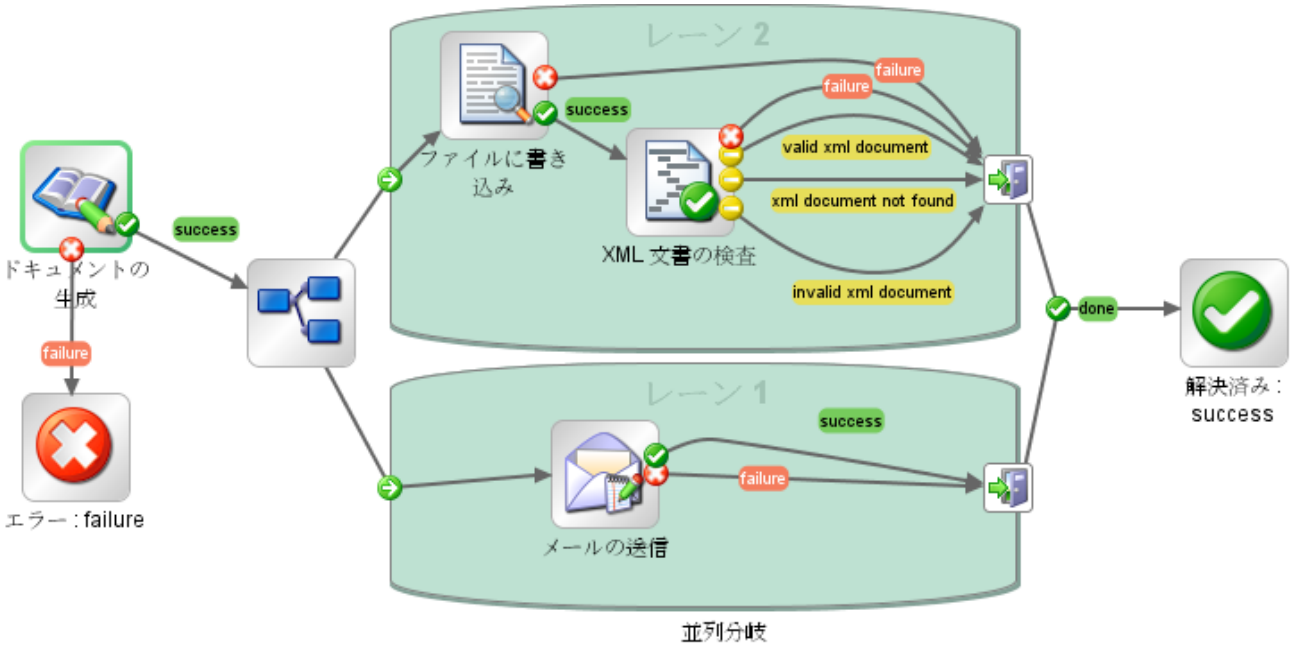


図 58 - レーンステップとトランジションを追加した並行分岐ステップ

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

- **設計** 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、フロー図で、ステップ全体を表す**[並列分岐ステップ]** 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。

フロー図内のレーンの視覚的な順序も変更できます（次の手順の説明を参照してください）。 ただし、フローを実行する際、レーンは同時に開始し、 フロー図での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 **[レーンを上に移動]** または **[レーンを下に移動]** をクリックします。

[編集] メニューコマンド、右クリックメニュー、キーボードの組み合わせ（CTRL+C、CTRL+V）のいずれを使用するか、あるいは **[作成]** ペインのツールバー上の **[コピー]** アイコン（ ）と **[貼り付け]** アイコン（ ）を使用して、特定の並行分岐ステップ、またはそのコンポーネントの 1 つまたは複数をコピーできます。 **[編集]** メニューの使用方法を次の手順で説明します。

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

1. **[設計]** ビューでフローを開いて、コピーする要素を選択します。

   並行分岐ステップ全体をコピーするには、**[並行分岐ステップ]** アイコン（ ）を右クリックします。
2. **[編集]** メニューで、**[コピー]** をクリックしてから **[貼り付け]** をクリックします。

レーンをコピーする場合、**[貼り付け]** コマンドを実行する際にカーソルをレーンの外側に置かないようにしてください。

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 **[レーンの追加]** 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 **[レーンの複製]** 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

[編集] メニューコマンド、右クリックメニュー、キーボードの組み合わせ（CTRL+X）、**作成**ペインのツールバー上の **[削除]** アイコン（ ）と **[切り取り]** アイコン（ ）のいずれかを使用して、特定の並行分岐レーンを削除できます。 **[編集]** メニューの使用方法を次の手順で説明します。

レーンを削除するには

- レーンを選択して、**[編集]** メニューで、**[切り取り]** 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て、ドロップダウンメニューで、**[名前の変更]** 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

ただし、レーンの開始ステップにはフローの開始ステップにあるような緑色の枠がないので注意してください。

レーンの開始ステップを変更するには

- 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。

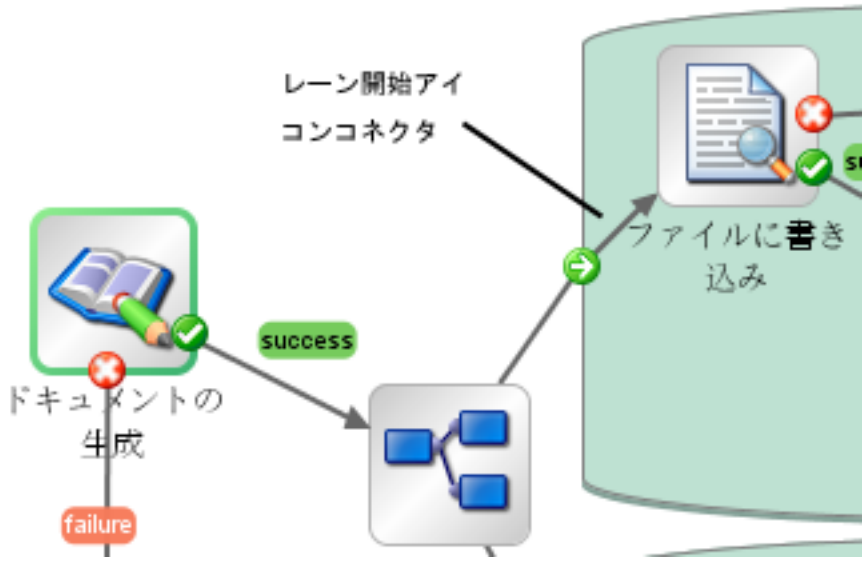


図 59 - レーン開始アイコンコネクタ

並列分岐ステップとのデータの移動

並行分岐ステップを開始すると、並行分岐ステップの各レーンは、グローバルコンテキストのフロー変数、ローカルコンテキスト変数、並行分岐ステップ自身の入力を取得します。 各レーンはこれらの変数を使用でき、また他のレーンとは関係なく、通常のフロールールに従ってこれらの変数の作成、変更、削除することができます。

レーンの実行が終了すると、各コンテスト内のフロー変数は呼び出し側のフロー（並行分岐ステップの一部に含まれるフロー）のコンテキスト内に再度マージされます。 マージ順序はレーンが終了した順序です。 そのため、2 つのレーンが同じフロー変数に書き込んだ場合、最後に終了したレーンが変数の最終値を提供します。

重要：

- 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。 （scriptletcontext.putGlobal() を使用する構文については、オペレーションまたはステップの **[スクリプトレット]** タブで、**JavaScript** テンプレートを挿入します）。

並列分岐ステップのデバッグ

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、デバッガーでは実稼働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 診断]** 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、同時に実行する複数のステップのインスタンスを開始するステップで、各インスタンスはステップの入力の 1 つに対して指定した複数の値のいずれか 1 つを使用します。 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。

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ような場合は、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。

ステップの作成元となるオペレーションは、フローまたは単純なオペレーションのいずれでも構いません。 マルチインスタンスステップをフローから作成された場合（したがって、サブフローになります）、マルチインスタンスステップのサブフローの各実行内のフロー変数や入力に保存されているデータには、親フローのローカルコンテキストからアクセスすることはできないので注意してください。 マルチインスタンスステップの子の実行から親フローにデータを渡すには、「*サブフローから親フローへのデータの引き渡し*」で説明する方法を使用する必要があります。

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。

フロー設計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グループとして完了があります。 マルチインスタンスステップの グループとして完了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、 グループとして完了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、フロー設計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 60 - フロー設計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（成功、失敗、ホストが見つからない、グループとして完了）のうち、グループとして完了レスポンスのみがリターンステップにつながります。
- 失敗レスポンスとホストが見つからない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。
- 成功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。

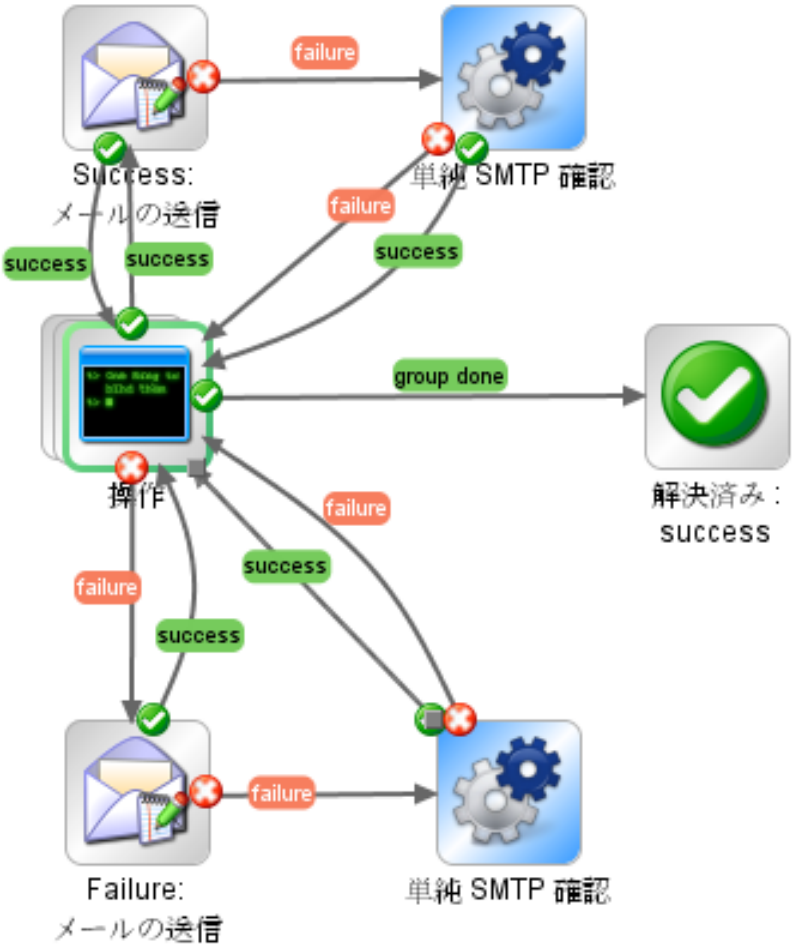


図 61 - 適切なデザイン： マルチインスタンスステップに再び接続されるレスポンス

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。

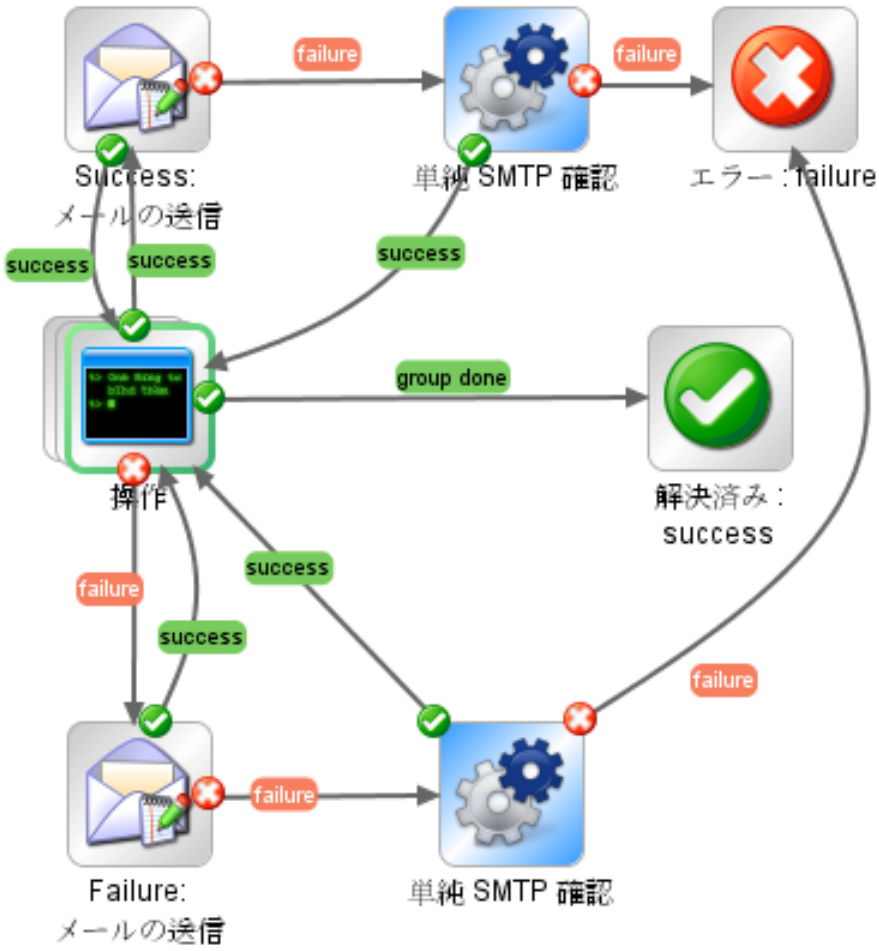


図 62 - 不適切なデザイン： マルチインスタンスステップに再び接続されないレスポンス

- 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、完了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 **[複数インスタンスの切り替え]** 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 63 - 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には、追加のレスポンスとして **[グループとして完了]** があります。 **[グループとして完了]** レスポンスの使用の詳細については、「*フロー設計でのマルチインスタンスステップの使用*」を参照してください。

2. ステップのインスペクターを開いて、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、**[入力の種類]** ドロップダウンリストで **[値のリスト]** を選択し、**[入力区切り文字]** 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 **[説明]** 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 **[定数値]** 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 **[定数を使用する]** 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 **[複数インスタンスの切り替え]** 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フロー変数には、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 64 - 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。

- **[トレースルート]** ステップはマルチインスタンスステップです。
- **[トレースルート]** 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、**トレースルート**の対象となるマシンの名前です。

- **[メールの送信]** 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 **[メールの送信]** 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、**トレースルート**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。

[メールの送信] 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。

- ${target} に対するトレースルートが成功しました。
- トレースルートが ${target} に接続できませんでした。

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. ステップのインスペクターを開いて、**[詳細]** タブをクリックします。
2. **[実行]** で、**[並列実行を、この同時インスタンス数にまで絞ります]** 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

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完了の 1 つのみです。

[ネットワーク診断] ステップの失敗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）。

フローの後続の操作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、フローが続行できれば便利です。

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、**[非ブロックの切り替え]** 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 つのレスポンス完了を取得します。 ただし、次の図では、ステップが現在到着不可能で別のステップへのトランジションがないため、黄色の三角形の警告が表示されています。

図 65 - ノンブロッキング化されたステップ

2. **[完了]** 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 **[単一レスポンスの切り替え]** 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 **[単一レスポンスの切り替え]** 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または実行履歴のみを収集して 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いて 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量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ので、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。

チェックポイントがフローの実行状態全体ではなく実行履歴のみを保存するように指定することで、チェックポイントで保存されるデータ量を減らすこともできます。

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. **[設計]** 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インスペクターを開きます。
2. **[詳細]** タブの **[ステップの永続性]** エリアで、**[このステップは、実行状態全体を保存します]** チェックボックスを選択します。

   または

   実行履歴のみを保存するには、**[このステップは実行履歴を保存します]** チェックボックスを選択します。

   注: 実行状態全体には実行履歴のすべてのデータが含まれるため、**[このステップは、実行状態全体を保存します]** チェックボックスを選択すると、**[このステップは実行履歴を保存します]** チェックボックスは使用できなくなります。

チェックポイントのコンテンツをステップから削除または削減するには

- ステップインスペクターで、**[詳細]** タブの **[ステップの永続性]** エリアで、必要に応じて **[このステップは、実行状態全体を保存します]** または **[このステップは実行履歴を保存します]** チェックボックスのいずれかの選択を解除します。

ステップの出力の保存

データベースにステップの未加工結果を保存するか、主結果を保存するか、または何も保存しないかを指定できます。 ステップの結果がフローにとって重要でないことがわかった場合は、ステップの結果を保存しないように設定することもできます。 フローの多くが別のステップからの結果を使用します。したがって、結果を保存しないステップはどのステップかに留意する必要があります。 結果を保存しないことの利点は、作成者、ユーザー、分析者がフィルタリングするデータを削減できるばかりかパフォーマンスも向上します。

ステップの出力を保存するには

1. **[設計]** 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インスペクターを開きます。
2. **[詳細]** タブの **[ステップの永続性]** エリアで、**[このステップは、その未加工出力を保存します]** チェックボックスを選択します。

   または

   プライマリ出力のみを保存するには、**[このステップは、そのプライマリ出力を保存します]** チェックボックスを選択します。

ステップの出力の保存を削除または削減するには

- ステップインスペクターで、**[詳細]** タブの **[ステップの永続性]** エリアで、必要に応じて **[このステップは、実行状態全体を保存します]** または **[このステップは実行履歴を保存します]** チェックボックスのいずれかの選択を解除します。

スクリプトレット

ping、トレースルート、**HTTP クライアントの 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、オペレーションの主要な操作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 **[スクリプトレット]** 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、ステップのインスペクターまたはオペレーションの **[プロパティ]** シートの **[スクリプトレット]** 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方法と同様、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を評価したり、フォーマットすることが可能です。 データをフロー変数に格納してフローの結果のフィールドに渡し、ほかの結果でもこれを繰り返すと、データの変化を追跡できるようになります。
- 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 **[プロパティ]** タブの **[スクリプトレット]** 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

スクリプトレットを記述するのに役立つ 3 つのソースを以下に示します。

- スクリプトレットテンプレート
- **[Configuration\Scriptlets]** フォルダー内のデフォルトのスクリプトレット
- デフォルトのコンテンツ内の既存のスクリプトレット

スクリプトレットを作成するには

1. オペレーションまたはフローの **[プロパティ]** シートまたはステップのインスペクターで、**[スクリプトレット]** タブをクリックします。
2. 次のいずれかを実行します。
   - スクリプトレットテンプレートを表示するには、**[種類]** ドロップダウンリストでスクリプトを記述するのに使用する言語を選択し、**[テンプレートの挿入]** をクリックします。
   - **[Configuration\Scriptlets]** フォルダー内のいずれかのスクリプトを開き、フロー、オペレーション、またはステップの **[スクリプトレット]** タブにスクリプトをコピーしてから、目的に合わせて編集します。
   - スクリプトレットが含まれるデフォルトのコンテンツ内のいずれかのオペレーション（**Operations\Operating Systems\Linux\Red Hat** フォルダー内のオペレーションなど）から、フロー、オペレーション、またはステップの **[スクリプトレット]** タブにスクリプトをコピーしてから、目的に合わせて編集します。

スクリプトレットでの結果のフィルタリング

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. **[結果]** タブで、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、結果の行の右向き矢印（ ）を右クリックします。
2. フィルターエディターで、**[追加]** をクリックして新しいフィルターを作成します。
3. **[フィルターの選択]** ダイアログボックスで、**[スクリプトレット]** を選択します。

   スクリプトレット行がデフォルトで表示されます。

   その他の役立つ関数の構文を表示するには、**[テンプレートの挿入]** 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。
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。

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**成功** を指定します。 続いて、オペレーションの **[レスポンス]** タブで、デフォルトのレスポンスとして **失敗**を選択します。

オペレーションレスポンス用のスクリプトレットルールの作成

この手順では、作業するレスポンスが既に作成されていることを前提とします。 レスポンスの作成については、「*レスポンス: 結果の評価*」を参照してください。

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. レスポンスの行の右端で、右矢印（ ）を右クリックします。

   レスポンスルールエディターが表示されます。

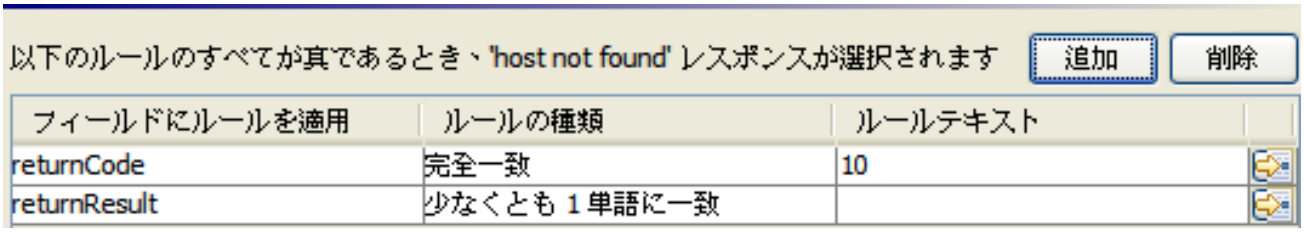


図 66 - レスポンスを決定するためのルールの作成

2. **[ルールの種類]** ドロップダウンリストで、**[スクリプトレット]** 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、**[テンプレートの挿入]** 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内に保存されている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 デバッガーを実行します。

スクリプトレットデバッガーは各スクリプトにアクセスするたびにスクリプトごとに 1 つのウィンドウを表示する JavaScript デバッガーです。 したがって、スクリプトを含むフローをデバッグする間にスクリプトレットデバッガーを実行すると、デバッグするスクリプトレットのウィンドウが複数表示されることがあります。

スクリプトレットをデバッグするには

1. JavaScript デバッガーを開くには、**[ツール]** メニューで **[JavaScript デバッガー]** をクリックします。

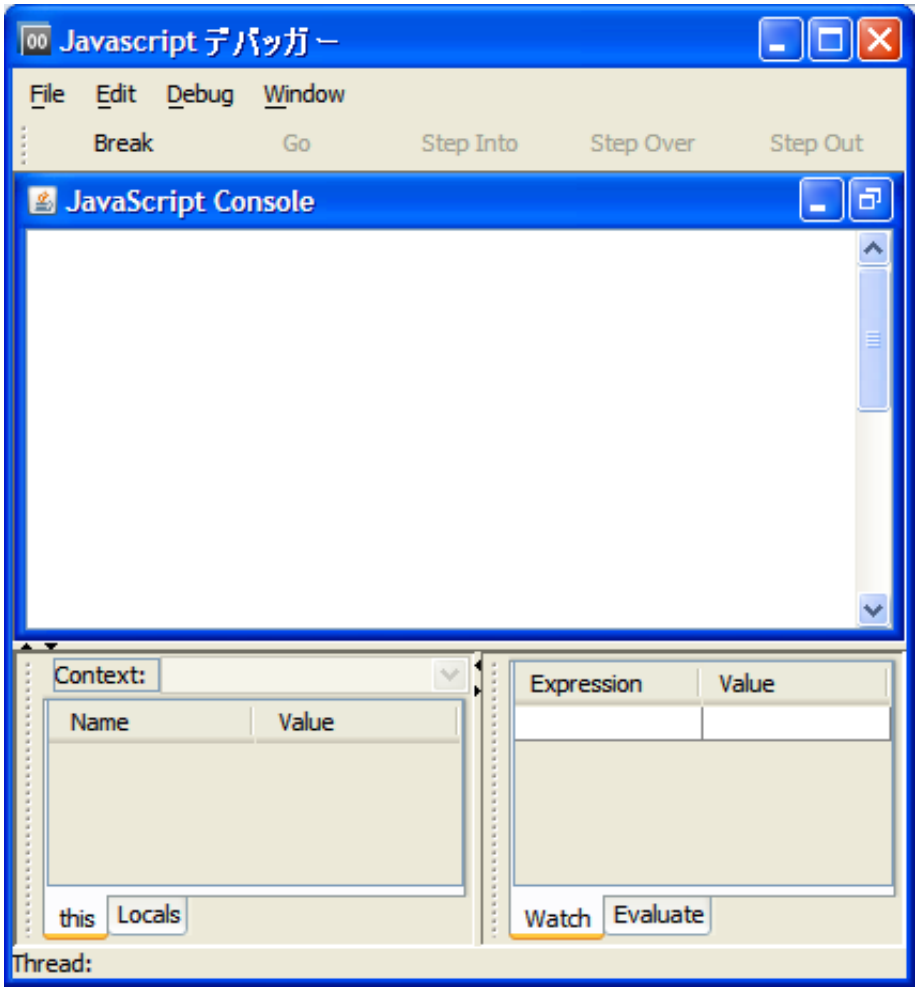


図 67 - JavaScript デバッガー

2. **[Debug]** メニューで、**[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 デバッガーで、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。

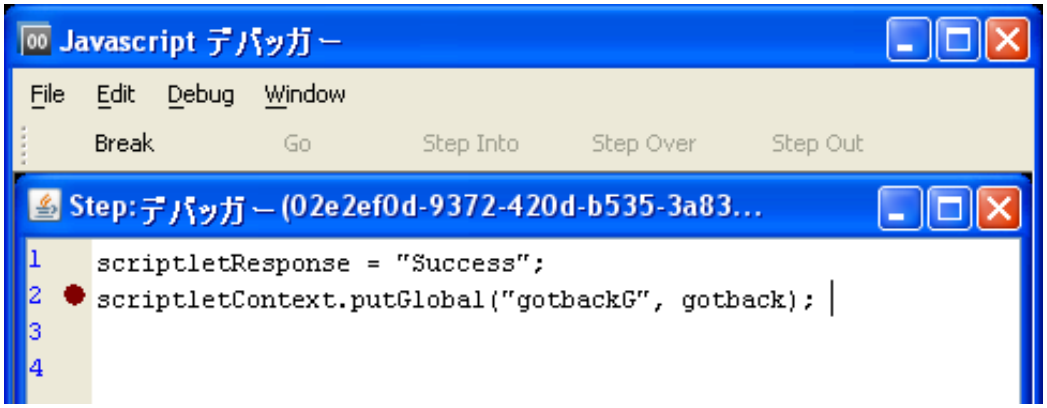


図 68 - JavaScript デバッガーでのスクリプトレット行のブレークポイント設定

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 デバッガーを閉じると、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 [デバッグ] をポイントし、**[ブレークポイントの設定]** 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、**[フローのデバッグ]** 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 デバッガーに切り替わります。

   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、**[コンテキスト]** ペインの [値] 列に表示されます。

   JavaScript デバッガー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. スクリプトレットを含むステップのインスペクターで、スクリプトレットエディターを開きます。
2. スクリプトレットを選択し、**[スクリプトレット]** 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。
3. **[Configuration\Scriptlets]** フォルダーで、スクリプトレットに名前を付けます。
4. 作業結果を保存します。

Configuration フォルダーからのスクリプトレットの使用

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、**[カスタムスクリプトレットに切り替え]** 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。

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 アクセラレーターパック、統合、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、通常これが出力です。 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際には、プライマリ出力を指定できます。 プライマリ出力を作成すると、そのソースを変更できますが、プライマリ出力なしの状態に戻すことはできません。

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。

次の手順では、オペレーションが編集可能であることを前提としています。 オペレーションがロックまたはシールされている場合は、オペレーションをのコピーを作成し、そのコピーに変更を加えるようにしてください。

オペレーションのプライマリ出力を指定するには

1. 編集のためにオペレーションの **[プロパティ]** シートを開くには、**[リポジトリ]** ペインで、目的のオペレーションをダブルクリックします。
2. **[プロパティ]** シートで、**[出力]** タブをクリックします。
3. **[フィールドからプライマリ出力を抽出]** 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 **[説明]** タブをクリックします。

必要な特定のデータを出力から抽出するために、1 つのフィルターまたは一連のフィルターを作成することができます。

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。

セカンダリ出力の追加

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、[出力の追加] 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: [利用できる出力] の下にある出力の行の [名前] 列と [出力フィールド] 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 [フィルター] 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。

出力の削除

オペレーションから出力を削除するには

- オペレーションの **[プロパティ]** シートの **[出力]** タブで、削除する出力を選択して **[出力の削除]** をクリックします。

出力のソースの変更

このトピックでは、オペレーションまたはフローの出力のソースの変更について説明します。 ステップの結果のソースの変更については、「*結果のソースの変更*」を参照してください。

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、**[フィールドからプライマリ出力を抽出]** ボックスの右にある下向き矢印をクリックし、リストから目的のフィールドを選択します。

   または

   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 **[出力フィールド]** 列をクリックして、ドロップダウン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。

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。

レスポンスおよび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。

ステップの結果の追加

ステップの未加工結果と主結果は、基底の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。 ステップのインスペクターで、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、ステップのインスペクターを開き、**[結果]** タブで **[結果の追加]** 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。

   または

   既存の結果がある場合は、新しい結果は既存の結果のコピーになります。
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、必要な変更を行います。
   - **[名前]** フィールドに、新しい名前を入力します。
   - **[送信元]** 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。
   - **[割り当て先]** フィールド：
     - フロー変数の値を保存するには、**[フロー変数]** 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、**[フロー出力フィールド]** を選択します。
   - **[割り当て操作]** リストで、適切な操作を選択します。
     - **OVERWRITE**

       フロー変数またはフロー出力フィールドの現在の値をこの値に置き換えます。
     - **APPEND**

       フロー変数またはフロー出力フィールドの現在の値の末尾にこの値を追加します。
     - **PREPEND**

       フロー変数またはフロー出力フィールドの現在の値の先頭にこの値を追加します。
     - 四則演算 **ADD**、**SUB**、**MULTIPLY**、**DIVIDE** の割り当て操作も選択できます。

       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。

または、「タイムアウト」という名前のフロー変数があり、ステップの結果の値が 1 であるとします。**[割当先]**で**[フロー変数]** を選択しておき、**[割り当て操作]** に **[SUB]** を選択することで、1 を「タイムアウト」値から減算することができます。

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 [フィルター] 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。

ステップの結果の削除

ステップから結果を削除するには

- ステップのインスペクターの **[結果]** タブで、削除する結果を選択して **[結果の削除]** をクリックします。

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

- 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 **[送信元]** 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。

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、最大、最小、平均の往復時間や、喪失したパケットの割合などのデータの一部の抽出が必要になることがあります。 また、このデータをフロー内またはフロー外で使用する必要があることもあります。

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: または、未加工データを出力に割り当て、レスポンスルールでフィルターを作成できます。
- フィルタリングされたデータを値としてフロー変数に割り当て、そのデータを以降のフローでオペレーションにアクセスできるようする。 さらに、
  - 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の中で、フローのユーザーにこのデータが使用できるようにする。
  - フィルタリングされたデータを電子メールで送信したりファイルに書き込むためにフロー変数を参照するステップで、組織内の他の人間がそのデータを使用できるようにできる。
- 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

フィルターは、単一のものであろうと一連のものであろうと、オペレーション出力とステップ結果から必要なデータを抽出します。 単一フィルターを作成できるだけではく、出力と結果から余分なデータを除外して必要なデータのみにするために一連のフィルターを作成することができます。

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をフィルタリングするには、オペレーションの **[プロパティ]** シートを **[出力]** タブで開き、**[フィルターの編集]** ボタンをクリックします。

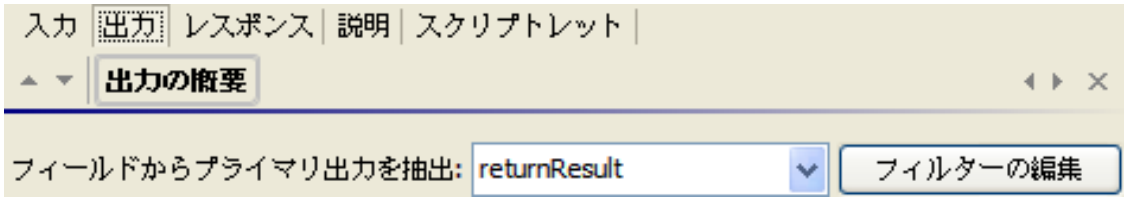


図 69 - [フィルターの編集] ボタンのあるプライマリ出力

または

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、**[利用できる出力]** で出力の行の右端にある右向き矢印（）をクリックします。

または

ステップの結果にフィルターを作成するには、ステップのインスペクターを開き、**[結果]** 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。

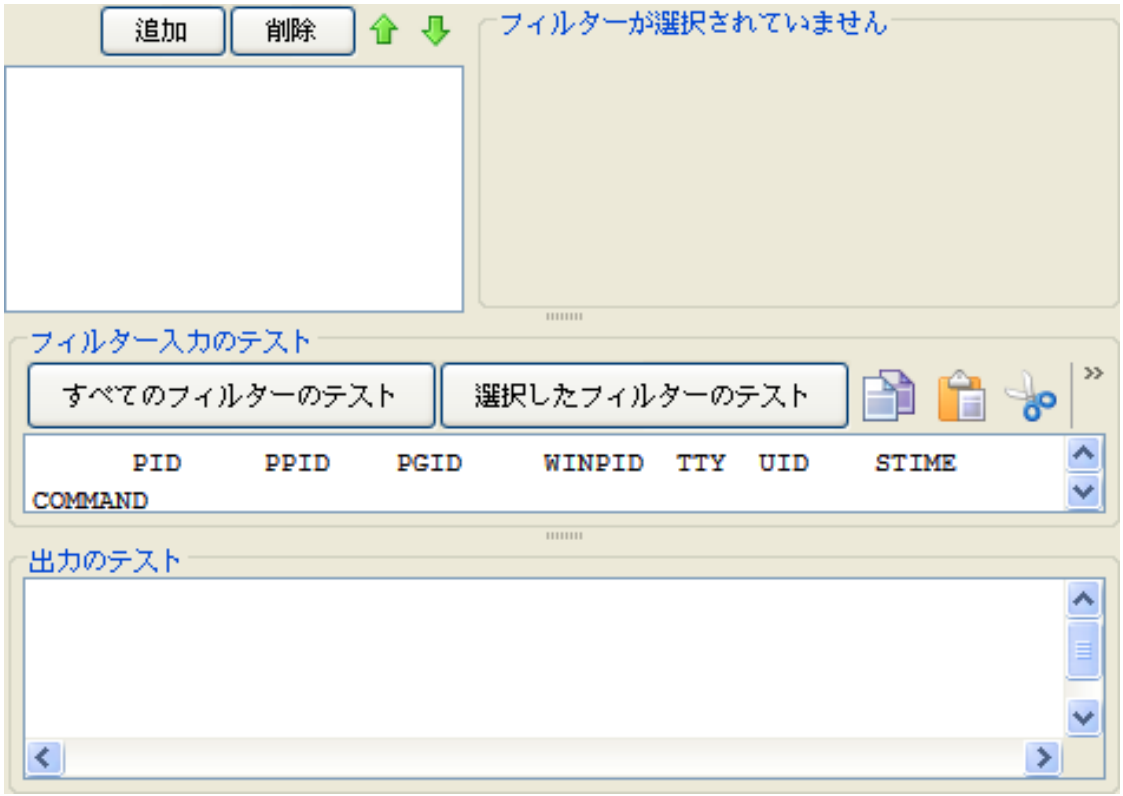


図 70 - フィルターエディター

- 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 [フィルターが選択されていません] と表示されている）右上のボックスが変わり、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表示されます。

2. フィルターを追加するには、**[追加]** をクリックし、表示された **[フィルターの選択]** 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。
3. フィルターエディターの右上にあるエリアの **[詳細]** で、フィルターの詳細を定義します。

   フィルターの各種種類の構成については、「*フィルターの詳細*」を参照してください。

   ヒント：フィルターエディターを開いている間、**[出力の概要]** の横の上向き矢印または下向き矢印（ ）をクリックして、別の結果のフィルターを作成できます。

   出力または結果のデータをフィルタリングする場合、データはフォルターの入力になります。 したがって、フィルターエディターでは、**[フィルター入力のテスト]** ボックスにテストデータを入力します。 実際のデータを使用してフィルターをテストするには、出力または結果のデータを **[フィルター入力のテスト]** ボックスに入力します。
4. **[フィルター入力のテスト]** ボックスをクリアするには、**[クリア]** をクリックします。
5. **[フィルター入力のテスト]** ボックスに選択したデータを指定するには、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、**[クイックコマンド]** 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、**[OK]** をクリックします。

     コマンドの出力は **[フィルター入力のテスト]** ボックスに表示されます。
   - 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. **[ステップ結果インスペクター]** で、**[未加工結果]** 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 **[フィルター入力のテスト]** ボックスに貼り付ける。
6. フィルターエディターで、**[すべてのフィルターのテスト]** をクリックします。

   または

   テストするフィルターを選択して **[選択したフィルターのテスト]** をクリックします。

   フィルターが（上から順番に）**[フィルター入力のテスト]** ボックスのデータに適用され、フィルタリング結果が **[出力のテスト]** ボックスに表示されます。

作成可能なフィルターの種類、および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

各種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。

- *大文字と小文字の変換*
- *数値の抽出*
- *形式*
- *行カウント*
- *正規表現*

- *重複行の削除*
- *置換*
- *数値の丸め*
- *スクリプトレット*
- *行の選択*
- *範囲の選択*
- *ソート*
- *削除*
- *スペースの削除*
- *テーブル*
- *XML フィルター*
  - *XML 属性の取得*
  - *XML 要素の取得*
  - *XML 要素値の取得*
  - *XPath クエリ*

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 [フィルター入力のテスト] ボックスに入力します。

大文字と小文字の変換

[大文字と小文字の変換] フィルターは、文字列に含まれるすべての文字を大文字または小文字に変更します。 **[大文字に変換]** チェックボックス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。

数値の抽出

[数値の抽出] フィルターは、結果内の最初の数字を抽出します。 連続した整数は 1 つの数字と見なされます。 たとえば、「123Test」や「Test123」という文字列からは、**[数値の抽出]** フィルターにより「123」の数字が抽出されます。

形式

[形式] フィルターは、結果や出力にテキストを付加したり、結果や出力の元のコンテンツを指定したテキストに置換します。

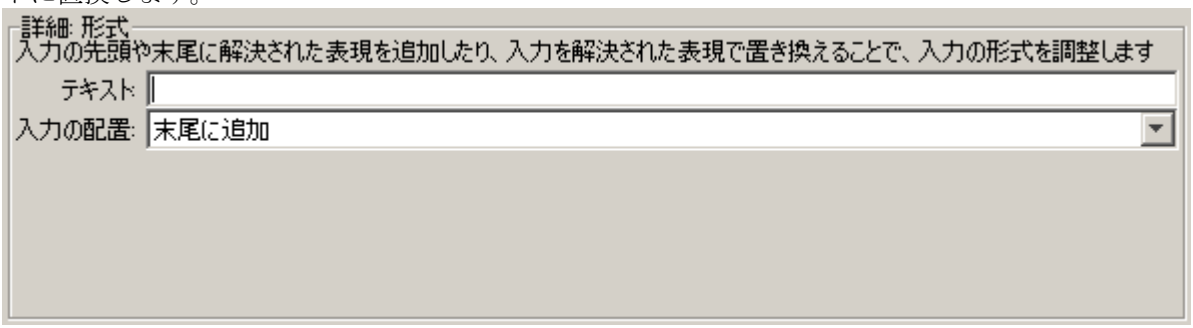


図 71 – 形式フィルターの定義

- **[テキスト]** ボックスに、結果に付加するテキスト、または結果と置換するテキストを入力します。
- **[入力の配置]** リストで、以下を選択します。

- 既存のテキストの先頭にテキストを追加するには、**[先頭に追加]** を選択します。
- 既存のテキストの末尾にテキストを追加するには、**[末尾に追加]** を選択します。
- 出力をテキストに置換するには、**[置換]** を選択します。

行カウント

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

正規表現

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。

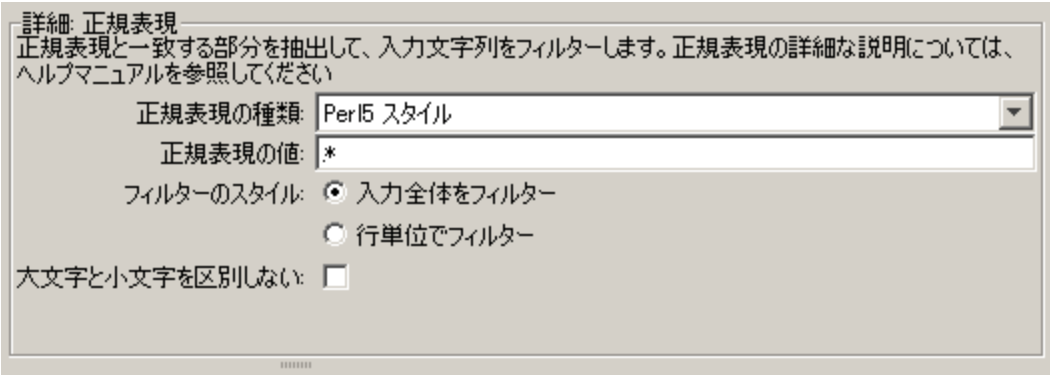


図 72 - 正規表現フィルターの定義

- **[正規表現の種類]** に、データーをフィルタリングする正規表現の種類を選択します。
- **[正規表現の値]** に、正規表現を入力します。
- 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、**[入力全体をフィルター]** または **[行単位でフィルター]** を選択します。
- 正規表現の大文字小文字を区別しない場合は、**[大文字と小文字を区別しない]** を選択します。

重複行の削除

同一の行を検出し、そのうち 1 つを残してすべて削除します。

- 連続している重複行のみにこのフィルターを適用するには、**[連続]** を選択します。

置換

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。

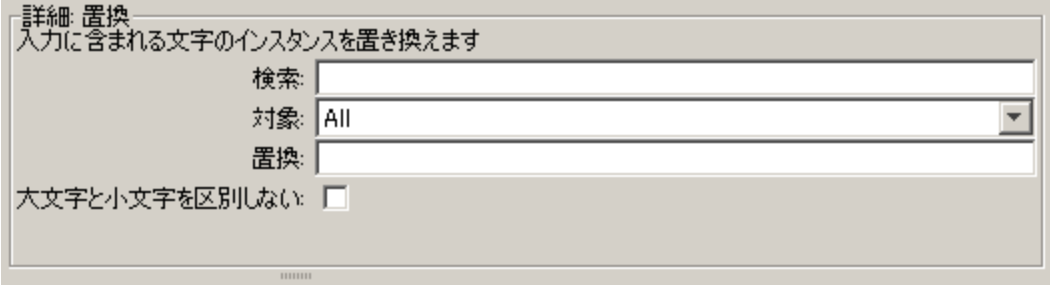


図 73 - 置換フィルターの定義

- **[検索]** ボックスに、検索および置換対象の文字列を入力します。

- 置換する対象文字列のインスタンスに応じて、**[置換]** ドロップダウンリストから **[First]**、**[All]**、または **[Last]** を選択します。
- **[置換]** ボックスに、対象文字列と置換する文字列を入力します。
- 検索時に大文字小文字を区別しない場合は、**[大文字と小文字を区別しない]** チェックボックスを選択します。

数値の丸め

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。

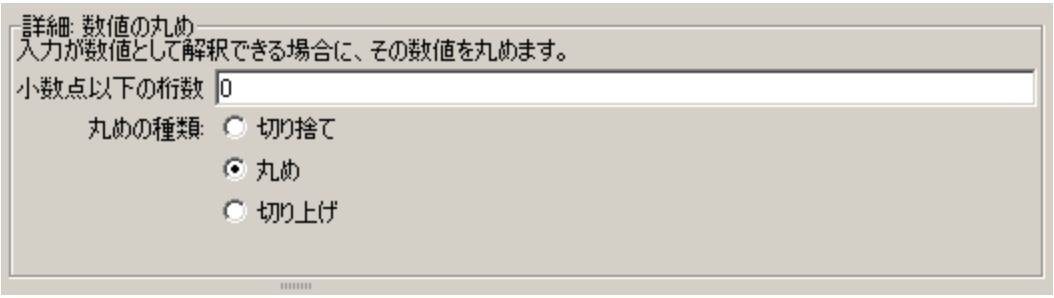


図 74 - 数値の丸めフィルターの定義

- 丸めの精度を指定するには、**[小数点以下の桁数]** ボックスに、数値を丸める小数位を入力します。
- **[丸めの種類]** で、丸める数値を指定します。
  - **[切り捨て]** は常に数値を切り捨てます。
  - **[切り上げ]** は常に数値を切り上げます。
  - **[丸め]** は、末尾の数字が 5 以上の場合に切り上げ、4 以下の場合は切り捨てます。

スクリプトレット

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 75 - スクリプトレットフィルターの定義

- **[種類]** 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、**[テンプレートの挿入]** をクリックします。

挿入されたテンプレートは、選択した言語に固有のもので、フロー変数、グローバル変数の値、オペレーション結果、および入力へのアクセスや、フロー変数の値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、**[スクリプトの確認]** をクリックします。

行の選択

未加工結果から抽出する行を指定します。

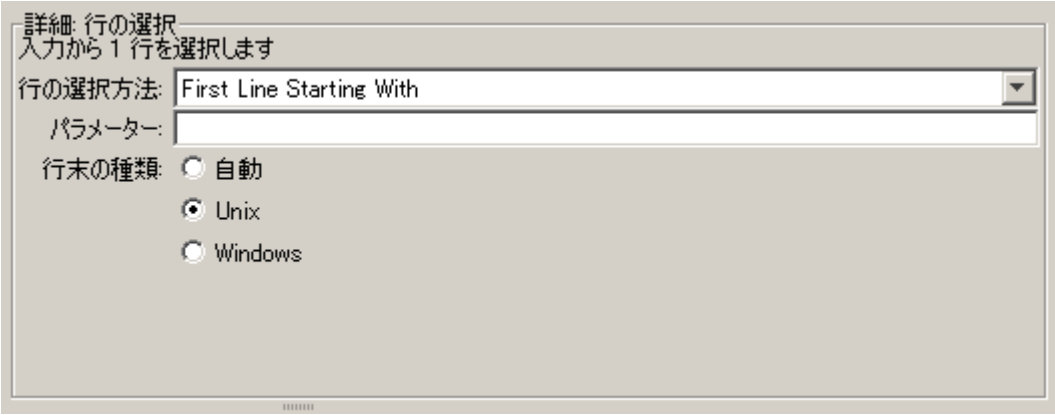


図 76 - 行の選択フィルターの定義

- **[行の選択方法]** リストから、対象とする行の基準を選択します。
- **[パラメーター]** 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。
- **[行末の種類]** 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）システムのいずれ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 **[自動]** は、行末の種類として Unix と Windows の両方を受け入れます。

範囲の選択

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。

図 77 - 範囲の選択フィルターの定義

[開始] ボックスと **[長さ]** 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。

改行は、フィルタリングするデータを取得するオペレーティングシステムによって 1 つの文字または 2 つの文字としてカウントされるので注意してください。 各種オペレーティングシステムでの行末については、「*行の選択*」を参照してください。

ソート

入力データ行を、各行の最初の文字によって順序付けます。

図 78 - ソートフィルターの定義

- 並べ替えの方向を指定します。
  - 昇順の場合は、**[昇順]** チェックボックスを選択したままにします。
  - 降順の場合は、**[降順]** チェックボックスの選択を解除します。
- データを ASCII 順に順序付けるには、**[数値として扱う]** チェックボックスを選択します。
  ただし、ASCII 昇順は、英語の文字の場合次のようにおおまかな順序になります。
  - スペース
  - 記号
  - 数値
  - アルファベット文字

削除

未加工の結果の先頭または末尾から文字を削除します。

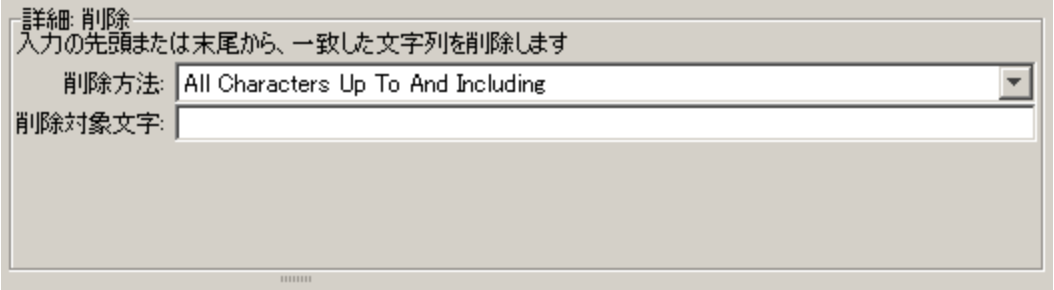


図 60 - 削除フィルターの定義

- **[削除方法]** リストから、フィルターで未加工結果から文字を削除する方法を選択します。 **[削除対象文字]** テキストボックスに指定した文字列を削除する場合、次のオプションを指定できます。
  - **対象文字より前のすべての文字**
  - **対象文字より前 (対象文字も含む) のすべての文字**
  - **対象文字より後のすべての文字**
  - **対象文字より後 (対象文字も含む) のすべての文字**
- **[削除対象文字]** テキストボックスに、検索する文字列を入力します。

スペースの削除

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。

テーブル

テーブルフィルターでは、未加工結果の表への変換は行いませんが、列の並べ替えや、列、行、ブロックの選択など、未加工結果を表として操作できます。

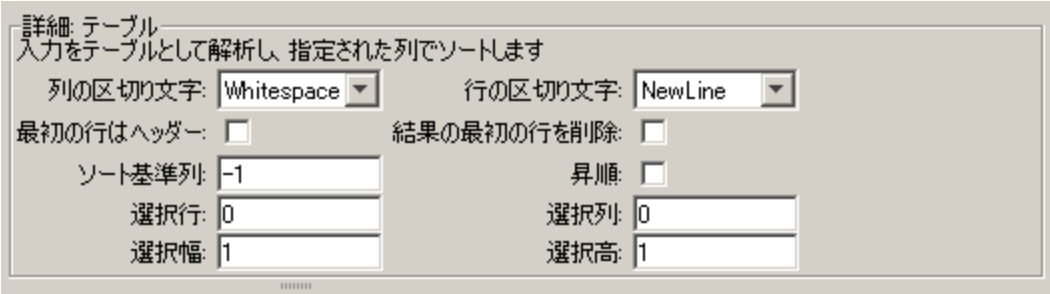


図 79 - テーブルフィルターの定義

注: 行の番号設定は 0 ベース（0 [0] で開始）、列の番号設定は 1 ベースです。

- **[列の区切り文字]** 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。
- **[行の区切り文字]** 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、**[最初の行はヘッダー]** を選択します。
- 最初の行を削除するには、**[結果の最初の行を削除]** を選択します。
- 列で並べ替えを行うには、**[ソート基準列]** ボックスに（1 ベースの）列番号を入力します。
  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、**[昇順]** ボックスを選択します。
  デフォルトの並べ替え順序は、降順です。
- フィルターで抽出する行を選択するには、**[選択行]** に（0 ベースの）行番号、**[選択幅]** に抽出する行に含まれる列の数を入力します。
- フィルターで抽出する列を選択するには
  - **[選択列]** に、列番号を入力します。
  - **[選択高]** に、抽出する列に含まれる行数を入力します。

たとえば、2 ～ 4 列目の最初の 5 行を抽出するには、次のように指定します。 次に示す設定のうち、最初の 2 つでは選択する行を指定し、残りの 2 つでは選択する列を指定します。

- [選択行]: 0
- [選択高]: 5
- [選択列]: 2
- [選択幅]: 3

[選択行]、**[選択列]**、**[選択幅]**、**[選択高]** 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。

- **[選択行]** 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。
- **[選択高]** に **-1** を指定すると、**[選択行]** に指定した行の下にある、データ内の残りの行がすべて選択されます。
- **[選択列]** 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。
- **[選択幅]** に **-1** を指定すると、**[選択列]** に指定した列の右にある、データ内の残りの列がすべて選択されます。

XML フィルター

XML フィルターを使用すると、ステップ内の XML （この XML はステップの入力または結果から取得されるもので、フローを作成する必要はない）を解析し、その XML を OO デフォルトコンテンツ内の XML 処理オペレーションの 1 つに渡すことができます。

オペレーションの XML フィルターの使用方法とデフォルトコンテンツ内の XML 処理オペレーションの使用方法では、いくつかの点で異なります。 まず、オペレーション内のタスクの完了と、タスクを完了するためのフローのインフラストラクチャの使用において違いがあります。 さらに、オペレーション内のフィルターには、XML 処理オペレーションにはない制限がいくつかあります。 これらの制限については、以下に示す各フィルターのセクションで説明します。 フィルターまたはオペレーションによる入力 XML のフィルタリングを設定するかどうかは、XML の取得方法によって決まります。

XML フィルターの説明では、次の XML 例を参照として例示します。

```
<?xml version="1.0" encoding="utf-8"?>
<tickets>
  <ticket id="1448" severity="3">
    <customer firstName="John" lastName="Doe">
      <volume>30000</volume>
      <company>myOrg</company>
      <position>CIO</position>
      <contactInfo>
        <email>jdoe@myorg.com</email>
        <email>johnsSecondEmail@myorg.com</email>
        <mobile>12065551212</mobile>
        <description internal="1">Private contact info</description>
        <description>Partial contact info</description>
      </contactInfo>
      <description>Our best customer</description>
    </customer>
    <details>
      <description>A simple Test xml</description>
      <comment user="john"> Initially raising ticket</comment>
      <comment user="frank"> Problem diagnosed, not a real issue</comment>
      <comment user="albert">ok, I'm going to close it.</comment>
      <state>Closed</state>
    </details>
  </ticket>
  <ticket id="1886" severity="5">
    <customer firstName="Elaine" lastName="Benson">
      <volume>50000</volume>
      <company>herCompany</company>
      <position>CEO</position>
      <contactInfo>
        <email>ebenson@herco.com</email>
        <mobile>011445551212</mobile>
        <description internal="1">Private contact info</description>
        <description>Partial contact info</description>
      </contactInfo>
      <description>Our other best customer</description>
```

```
      </customer>
      <details>
        <description>datastream bug</description>
        <comment user="jack">Customer found bug.</comment>
        <comment user="elsbeth">It is a third-party supplier bug.</comment>
        <state>Closed</state>
      </details>
    </ticket>
  </tickets>
```

XML 属性の取得

XML 属性の取得フィルターは、指定した属性の 1 つまたは複数のインスタンスの各値を抽出します。 フィルターエディターでは、属性に要素パスを指定することで、フィルターを適用する属性のインスタンスを制御できます。

テーブル内に返される属性の単一または複数のインスタンスの値を取得できます。 このようなテーブルでは、列はカンマで区切られ、行は改行で区切られます。

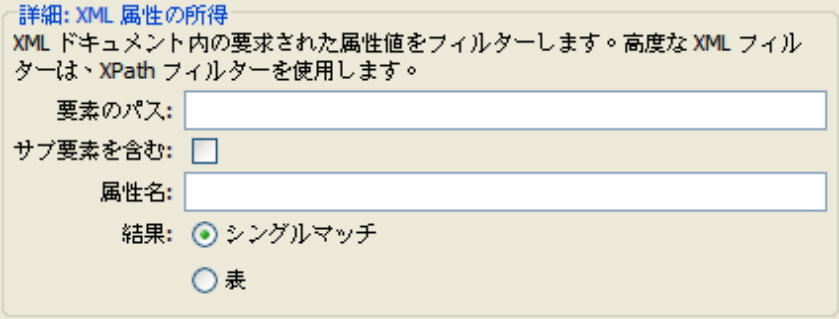


図 80 - XML 属性の取得フィルターの定義

- **[要素のパス]** ボックスで、抽出する値を持つ属性を含む要素のパスを指定します。 要素へのパスの各部分をスラッシュで区切ります。

  フィルターが属性の値を要素のどのインスタンスから取得するかを制御するには、`[2]` や `[3]` などの指定を追加してください。 要素の番号付けは 1 ベース（`[1]`) で開始など）です。 したがって、要素の 2 番目のインスタンスを指定するには、`[2]` を使用します。
- 指定した要素の子の要素を検索するには、**[サブ要素を含む]** チェックボックスを選択します。
- **[属性名]** ボックスに、目的の値を持つ属性名を入力します。
- 抽出される結果を、属性の単一インスタンスの値に制限するには、**[シングルマッチ]** を選択します。

  または

  名前を指定した属性のすべてのインスタンス値を抽出するには、**[表]** を選択します。

たとえば、XML 例で、以下の操作を行います。

- コメントの 1 つのユーザー名を探すには
  - **[要素のパス]** ボックスに、次のように入力します。
    `/ticket/details/comment`

    または

    特定のコメント（たとえば、2 番目のコメント）のユーザー名を取得するには、**[要素のパス]** に次のように入力します。
    `/ticket/details/comment[2]`
  - **[属性名]** ボックスに、次のように入力します。
    `user`

- **[結果]** の横の **[シングルマッチ]** を選択します。

出力は「john」となります。

- 各コメントのユーザー名を探すには
  - **[要素のパス]** ボックスに、次のように入力します。
    `/ticket/details/comment`
  - **[属性名]** ボックスに、次のように入力します。
    `user`
  - **[結果]** の横の **[表]** を選択します。

出力は次のようになります。

```
Path,user
/ticket/details/comment[1],john
/ticket/details/comment[2],frank
/ticket/details/comment[3],albert
```

XML 要素の取得

XML 要素の取得フィルターを使用すると、次のいずれかで要素を記述することで、子の要素、値、属性などの要素全体から 1 つの要素を抽出できます。

- 相対パスまたは絶対パスで。
- 抽出する要素の子の要素で。 子の要素の特定の値で検索することもできます。
- 抽出する要素の属性で。 属性の特定の値で検索することもできます。

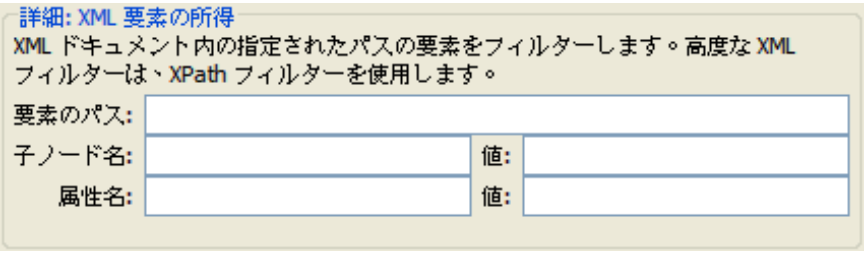


図 81 - XML 要素の取得フィルターの定義

次のいずれか、または次の組み合わせてを使用して、抽出する要素を指定できます。

- **[要素のパス]** ボックスに、要素への絶対パスを入力します。

  パス内の相対パスインジケーターは、相対パスインジケーターに先行する要素からの相対位置を示します。
  - ../ は、最後に指定された要素の親を指定します。
  - ./ は、最後に指定された要素を指定します。

  たとえば、XML 例では、`<volume>` と `<company>` は兄弟要素で、共に `<customer>` 要素の子です。`<company>` 要素は次の相対パスで指定できます。
  `/tickets/ticket/customer/volume/../company`

  識別された要素の複数のインスタンスが存在する場合、単にパスを指定するだけで、前述の例のように要素のすべてのインスタンスが返されます。

  整数を角かっこで囲んだ整数を指定して、パス内の任意の要素の特定のインスタンスを指定できます。 次に挙げるのがその例です。

  `/tickets/ticket/details/comment` は、すべてのチケットの詳細に含まれるすべてのコメントを指定します。

  `/tickets/ticket/details/comment[2]` は、各チケットの 2 番目のコメントを指定します。

  `/tickets/ticket[2]/details/comment` は、2 番目のチケットのすべてのコメントを指定します。

- **[子ノード名]** ボックスに、抽出する要素（複数可）の子となる要素の名前を入力します。 子の要素に値がある場合、**[値]** ボックスに値を入力することで結果を絞り込むことができます。

  注:

  - **[子ノード名]** ボックスは、子の要素の 1 つのレベルのみに機能します。 したがって、このフィルターにより、指定した子の要素の直接の親のみが返されます。
  - **[値]** ボックスは、短い値を指定するためのものです。ここに入力する値は、抽出する要素の子の要素値と完全に一致する必要があります。
- **[属性名]** ボックスに、抽出する要素に対して一意の属性名を入力します。 結果をさらに絞り込むために、**[値]** ボックスに属性の値を入力できます。

たとえば、XML 例では、以下の操作を行います。

- **customer** 要素とそのコンテンツすべてを抽出するには、次のいくつかの方法があります。
  - **[要素のパス]** ボックスに、次のように入力します。

    `/ticket/customer`
  - **[子ノード名]** ボックスに、顧客の次のいずれかの子の要素を入力します。

    `company`

    `position`

    `contactInfo`

    [子ノード名] ボックスに 「company」と入力した場合は、付随する **[値]** ボックスにも次のように入力できます。

    `myOrg`
  - **[属性名]** ボックスに、次のいずれかを入力します。

    `firstname`

    `lastname`

    付随する **[値]** ボックスに、これらの属性の次の各値を入力できます。

    `John`

    `Doe`

  これらのフィルターのそれぞれでは、出力は次のような顧客要素になります。

```
<customer firstName="John" lastName="Doe">
  <company>myOrg</company>
  <position>CIO</position>
  <contactInfo>
    <email>jdoe@myorg.com</email>
    <email>johnsSecondEmail@myorg.com</email>
    <mobile>12065551212</mobile>
    <description internal="1">Private contact info</description>
    <description>Partial contact info</description>
  </contactInfo>
  <description>Our best customer</description>
</customer>
```

XML 要素値の取得

XML 要素値の取得フィルターを使用すると、特定の要素の値を取得できます。 その他のフィルターと同様、要素のインスタンスが複数ある場合、異なるインスタンスを指定しない限りフィルターにより要素の最初のインスタンスが返されます。

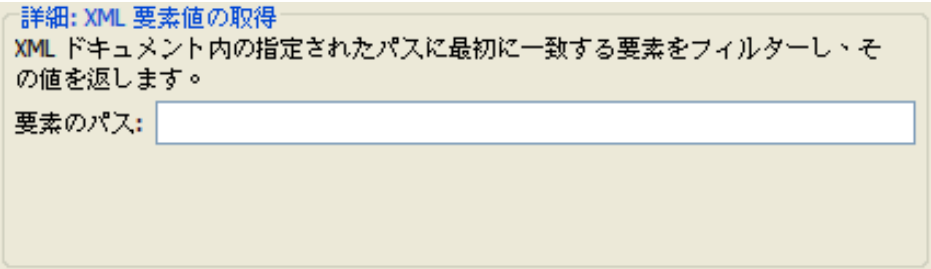


図 82 - XML 要素値の取得フィルターの定義

たとえば、XML 例を使用すると、電子メール要素の値を取得するには、**[要素のパス]** ボックスに次のように入力します。

```
/tickets/ticket/customer/contactInfo/email
```

出力は、指定した 2 つの電子メールのうちの 1 つとなります。

```
jdoe@myorg.com
johnsSecondEmail@myorg.com
```

ただし、次のように入力する場合、

```
/ticket/customer/contactInfo/email[2]
```

出力は次のようになります。

```
johnsSecondEmail@myorg.com
```

XPath クエリ

XPath クエリフィルターを使用すると、**[XPath クエリ]** テキストボックスに入力する標準 XPath 構文を使用したクエリによる結果から、データを抽出できます。

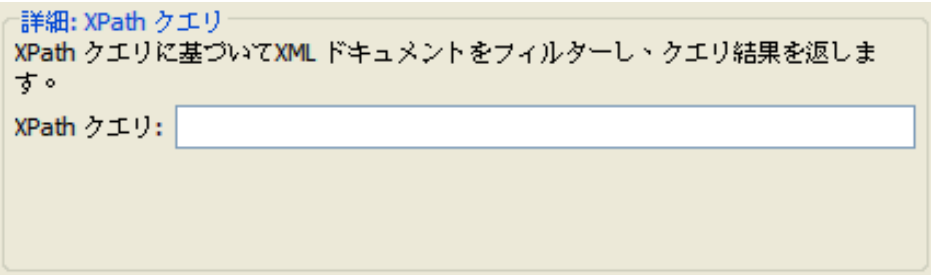


図 83 - XPath クエリフィルターの定義

XPath 構文では、以下のようになります。

- 角かっこを先頭に指定するパスは、結果を絞り込むためのクエリの範囲を識別します。
- 角かっこにはクエリのフィルタリング部分が含まれます。 1 つのクエリで複数セットのフィルターを指定できます。

たとえば、上記の XML 例を使用すると、次のいずれかのクエリで、いくつかのユニットの 40,000 を超えるボリュームを持つ顧客を抽出できます。

- この XPath クエリは、顧客のボリュームが 40,000 を超える会社すべてを見つけます。

  ```
  /tickets/ticket/customer/company[../volume>40000]
  ```

  `<volume>` 要素は、`<company>` タグの兄弟です。したがって、要素 `<volume>` を見つけるには、次のシーケンスを角かっこで囲んで、`<company>` への相対パスを明確に指定します。

  ```
  ../
  ```

- この XPath クエリは 40,000 を超えるボリュームを持つ顧客すべてを見つけます。

  ```
  /tickets/ticket/customer[volume>40000]
  ```

  `<volume>` は `<company>` の子であるため、その相対パスを指定する必要はありません。

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。 フィルターを作成したオペレーションとは別にフィルターを保存するには、**Configuration\System Filters** フォルダーにシステムフィルターとして保存します。

ただし、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、そのフィルター専用にスクリプトレットを保存することになります。

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 **[Configuration\System Filters]** フォルダーにドラッグします。
4. 新しいシステムフィルターの名前を変更するには、右クリックし、**[名前の変更]** をクリックして名前を変更します。

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出力または結果の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。

- 失敗レスポンス: データベースが実行されない、または到達できない場合。
- 行が返されなかったレスポンス: クエリが実行されたがデータが戻されなかった場合。
- 行が返されたレスポンス: 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。

図 84 - 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。
  - 解決済み:
  - 診断済み:
  - 操作なし:
  - エラー:
- 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 **[レスポンス]** 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、デフォルトのレスポンスとして識別されるレスポンス t 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

1. オペレーションの **[レスポンス]** タブで、**[レスポンスの追加]** をクリックして新しいレスポンスの名前を入力します。
2. 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、**[失敗時]** 列にあるレスポンスのチェックボックスを選択します。
3. レスポンスをデフォルトのレスポンスとして識別するには、**[デフォルト]** 列でレスポンスのチェックボックスを選択します。
4. **種類** 列で、レスポンスの種類を選択します。
   - 解決済み
   - 診断済み
   - 操作なし
   - エラー
5. レスポンスのルールを作成するには、レスポンスの行の右端にある右向き矢印（ ）をクリックします。
6. レスポンスルールエディターで、**[追加]** をクリックします。

   新しいルールが表示されます。

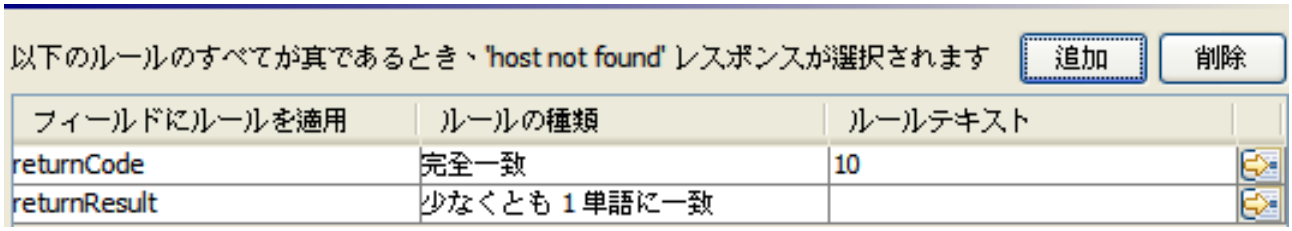


図 85 - ルールの定義

7. **[フィールドにルールを適用]** 列で、ルールをテストする対象の値を持つ結果フィールドをドロップダウンリストから選択します。

   テスト可能な結果フィールドには、結果の終了コード、出力文字列、エラー文字列、失敗メッセージ、タイムアウトの true または false があります。
   - これらの結果フィールドの詳細については、**[説明]** タブを参照してください。
   - これらのフィールドの値の内容を確認するには、Studio デバッガーを使用してフロー内のオペレーションをテストしてください。 デバッガーでフローを実行すると、**[ステップ結果インスペクター]** で任意のステップの結果を見つけることができます。 デバッガーの詳細については、「*フローのデバッグ*」を参照してください。
8. **[ルールの種類]** 列で、フィールド値をテストするのに使用する比較または一致を、ドロップダウンリストから選択します。
9. **[ルールテキスト]** 列に、テストで使用するテキストを入力します。

   [ルール詳細] エディターでは、以下を行うことができます。
   - システムエバリュエーター（ルール）、フィルター、スクリプトレットの使用など、より詳細なルールを指定します。
   - ルールを作成しながらテストします。

注: ルールでは、数字で始まる文字列の評価に比較演算子（=、!=、<、> など）を使用すると、比較されるのは文字列の数字部分のみです。 たとえば、!=（等しくない）を使用して「123」と「123Test」を比較する場合、「123」は明らかに「123Test」と同じではありませんが、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。

10. **[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。

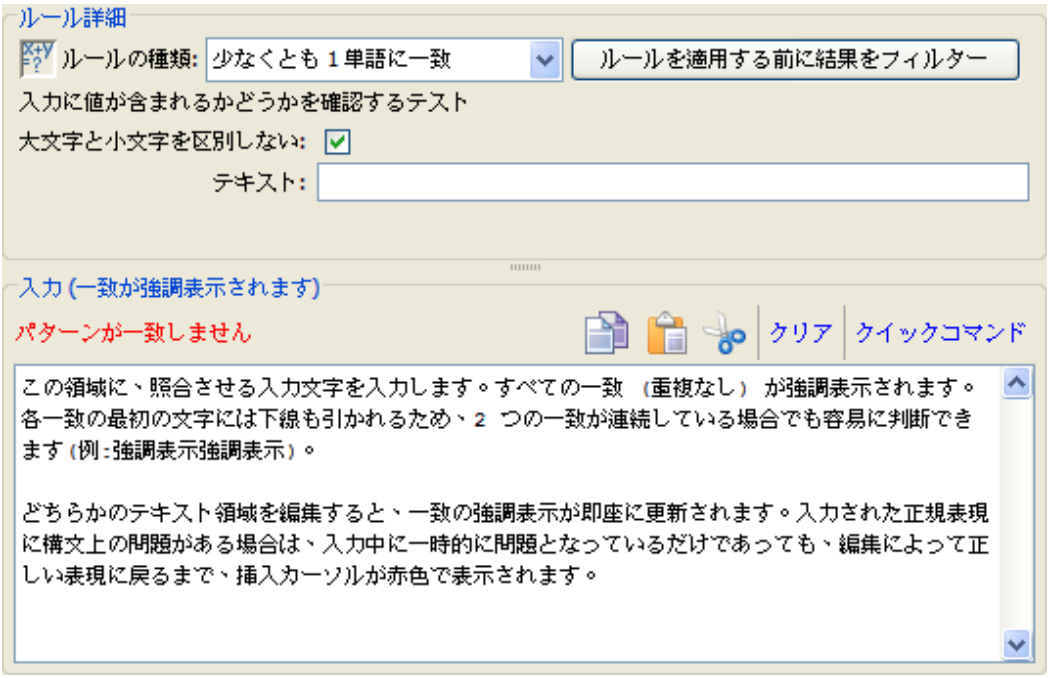


図 86 - ルール詳細エディター

注: ルールの種類に **[スクリプトレット]** を選択すると、**[ルールの詳細]** エディターは、スクリプトレットまたはスクリプトレットフィルターを作成したときと同様スクリプトレットエディターに変わります。 スクリプトレットおよびスクリプトレットフィルター用のエディターの詳細については、「*スクリプトレット*」および「*出力と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、**[ルールの種類]** ドロップダウンリストで既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。

11. 別のルールタイプを選択するには、**[ルールの種類]** ドロップダウンリストでほかのタイプを選択します。
12. ルールを適用する前に結果をフィルタリングするには、**[ルールを適用する前に結果をフィルター]** をクリックし、表示されたフィルターエディターでフィルターを作成します。

    フィルターエディターを使用した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*」、フィルターの種類の詳細については、「*フィルターの詳細*」を参照してください。
13. ほとんどのルールタイプについては、**[テキスト]** ボックスに、比較をテストするテキストを入力します。大文字小文字を区別しない場合は、**[大文字と小文字を区別しない]** チェックボックスを選択します。

    または

    [正規表現] ルールでは、オペレーション結果の正規表現フィルターを作成するときの正規表現とその用途を指定します。**[正規表現]** 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。
14. 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、**[レスポンスの概要]** の横にある上向き矢印または下向き矢印をクリックします。

ヒント: レスポンスルールは、オペレーションの **[レスポンス]** 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ことがあります。

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、操作が成功したかどうかなどを示します。

たとえば、**[非アクティブなドメイン管理者の検索]** 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**[非アクティブなドメイン管理者の検索]** フローを見つけてダブルクリックします。

- 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。
- 成功:

  無効なドメイン管理者が見つからなかった場合。
- 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 **[プロパティ]** シート（フロー図を開いて、ウィンドウ下部にある **[プロパティ]** タブをクリック）で、**[レスポンス]** タブをクリックします。
2. **[レスポンスの追加]** をクリックし、表示されたテキストボックスに、レスポンスの名前を入力します。

フローからのレスポンスの削除

フローからレスポンスを削除するには

- フローのレスポンスを削除するには、**[レスポンス]** タブで、レスポンスをクリックして **[レスポンスの削除]** をクリックします。

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. **[アイコン]** パネルを開くには、フローキャンバスの右側にある **[アイコン]** タブをクリックします。

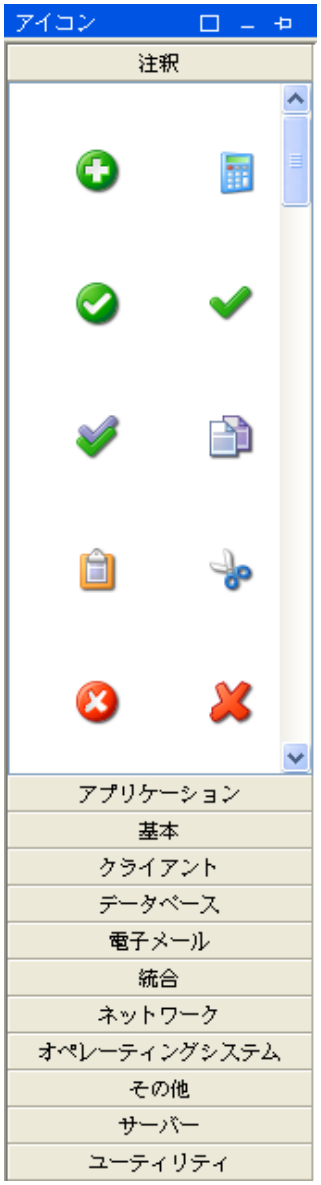


図 87 - [アイコン] ペイン

2. **[アイコン]** パネルで、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

システムプロパティはグローバル変数で、OO グローバルコンテキストに保存され、フローの開始時にすべてのフロー実行に提供される名前と値のペアで構成されます。 システムプロパティへの参照はすべて、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。

ただし、以下の理由から、システムプロパティの作成やシステムプロパティの値の変更は慎重に行う必要があります。

- システムプロパティはグローバルな範囲を持ち、フローの実行を開始するとシステムプロパティはフロー実行のコンテキストの一部となるため。

  その結果、システムプロパティの値を変更すると、既存のオペレーションとフローが中断することがあります。

- フローの実行を開始するとシステムプロパティはフロー実行のコンテキストの一部となるため。
- システムプロパティはすぐには表示できないため。

  システムプロパティは、Studio デバッガー（**コンテキストインスペクター**の **[システムプロパティ]** の下）、および**[リポジトリ]** ペインの **[Configuration]** フォルダーの下の **[System Properties]** フォルダーに表示されます。

  また、フローの作成時に入力を作成すると、フローの実行時に入力と同じ名前を持つフロー変数が自動的に作成されます。 したがって、意図しなくてもシステムプロパティと同じ名前を持つ空のフロー変数が作成され、これにより予期しない動作が発生することがあります。
- システムプロパティは同じ名前を持つフロー変数によって置き換えられるため。

  入力がその値をフロー変数から取得できる場合、すなわちフロー変数が存在しそのフロー変数に値が割り当てられている場合、そのフロー変数は入力の値のソースとしてシステムプロパティより優先されます。

その一方、入力値または結果をシステムプロパティに割り当てることでは、システムプロパティの値を変更することはできません。 これらの 2 つのソースのいずれかの値をシステムプロパティに割り当てると、システムプロパティと同じ名前を持つフロー変数を実際に作成し、そのフロー変数に値を割り当てることになります。

システムプロパティの作成

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、**[新規]** 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 **[プロパティ]** シートが表示されます。
3. **[説明]** ボックスに、システムプロパティの説明を入力します。
4. **[プロパティ値]** ボックスに、システムプロパティの値を入力します。

   たとえば、OO リポジトリ全体に RAS 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。

     overrideRAS
   - 値として、有効な RAS 参照の名前（**RAS_Operator_Path** など）を割り当てます。

   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。

   ${overrideRAS}

システムプロパティの値の取得

フローでシステムプロパティの値を使用するには

- システムプロパティの値を、フローまたはステップ入力のデータソースとして指定します。

  システムプロパティ
- スクリプトレットで、適切なコマンドを使用してシステムプロパティの値を取得します。

  必要なコマンドおよびその構文の詳細については、オペレーションの [スクリプトレット] タブで、**[テンプレートの挿入]** をクリックします。 テンプレートには、グローバルコンテキストで作業するのに必要なコマンドが用意されています。

システムプロパティの値の変更

システムプロパティの値を変更するには次のようないくつかの方法があります。

- スクリプトレット内での変更。

  これはスクリプトを実行するポイントからの値の変更に影響します。 必要なコマンドおよびその構文の詳細については、オペレーションの **[スクリプトレット]** タブで、**[テンプレートの挿入]** をクリックします。 テンプレートには、グローバルコンテキストで作業するのに必要なコマンドが用意されています。
- システムプロパティの値を設定するオペレーションを作成することによる変更。
- **[Configurations\System Properties]** フォルダー内でシステムプロパティを探し、それを開いてその値を変更することによる変更。

フロー設計

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設計を単純化する*ことが可能です。

サブフローを使用したフロー設計の単純化

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）、**List イテレーター**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、**List 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、成功と失敗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、このステップの作成の基になったサブフローのフロー出力フィールドの値をステップの結果のために取得することにより、フロー外の値を親フローに渡すことができます。

これを可能にする割り当ての変更を完了すると、データの移動はおおよそ次のように行われます。

1. サブフロー内のステップで、結果のデータがフロー出力フィールドに割り当てられます。
2. 親フローの、サブフローを基に作成されたステップで、ステップ結果はその値をステップの作成の基になったオペレーション（サブフロー）のフロー出力フィールドから取得します。
3. さらに親フローで、親フローで以降から使用するためにステップ結果がフロー変数に渡されます。

このプロセスを詳細に説明する例を以下に示します。 以下のステップでは、フローを有効にするために必要なその他の作業が完了していること、および結果の作成方法と入力への値の割り当て方法を熟知していることを前提としています。

1. コマンドオペレーションを作成し、このオペレーションに「dir C:\」を実行させます。 このオペレーションを「dir」と呼びます。
2. 「flowdir」という名前のフローを作成します。
3. 「flowdir」フローで、オペレーション「dir」を使用してステップを作成します。
4. 「dir」ステップで、オペレーションの出力文字列からの結果を追加します。
5. 結果をフロー出力フィールドに割り当て、結果「foo」という名前を付けます。

   これで、フローは「foo」という名前のフロー出力フィールドも持つことになります。
6. もうひとつフローを作成します。 このフローを「parentflow」と呼びます。
7. 「parentflow」で、「flowdir」を基にしたステップを作成します。
8. 作成したばかりの「flowdir」ステップに結果を追加します。
9. デフォルトで、新しい結果には「foo」という名前が付けられ、新しい結果は結果フィールド: "foo" からその値を取得します。この値は同じ「foo」という名前が付いたフロー変数に割り当てられます。

   これで、サブフローのステップ「dir」の結果「foo」は、親フロー内の「flowdir」ステップの後にくるトランジションとステップで利用できるようになりました。
10. これをテストするには、Basic Notify オペレーションから作成されたステップを「flowdir」ステップの後に追加します。 この新しいステップでは、以下を行います。
    - notifyData 入力を、定数値を使用する単一の値として定義し、定数値を ${foo} と指定します。
    - notifyMethod 入力を、定数値を使用する単一の値として定義し、定数値を Display と指定します。
    - ユーザープロンプトを回避するために、サブジェクト入力を、定数値を使用する単一の値として定義し、定数値を “If this worked, the flow output field says: "contents of outputString, aka foo" となるように指定します。
    - フローをデバッグします。

詳細な手順を以下に示します。

サブフローのステップのデータを親フロー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インスペクターを開きます。

   ステップのインスペクター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. **[結果]** タブをクリックし、「*ステップの結果*」で説明する手順に従って結果を追加します。
3. 結果の設定用に表示された行で、次の手順を実行します。
   - **[割り当て先]** 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。

- **[名前]** で、フロー出力フィールドの名前を入力します。
- **[送信元]** で、**[結果フィールド:Result]** を選択します。
- 結果をフィルタリングするには、フィルターを作成します（「*出力および結果のフィルタリング*」を参照してください）。

4. 親フローの作成キャンバスで、サブフローから作成したステップのインスペクターを開きます。
5. **[結果]** タブをクリックし、ステップの結果を作成します。

   デフォルトで、新しい結果は以下のようになります。
   - サブフローのフロー出力フィールドの名前を持つ結果フィールドからその値を取得します。
   - サブフローのフロー出力フィールドの名前と同じ名前を持ちます。
   - フロー変数に割り当てられます。デフォルトで、フロー変数には結果と同じ名前が指定され、このステップの後に続くトランジションやステップで使用可能になります。

ステップのベースとなるオペレーションの変更

既存のフローステップを別のオペレーションに関連付ける必要があるが、そのフローステップとの間にある既存のトランジションはそのまま維持しなければならないという場合があるとします。 このような場合、ステップのインスペクターが役に立ちます。

ステップのベースとなるオペレーションを変更するには

1. フローの **設計** ビューで、ステップのインスペクターを開いて **[詳細]** タブをクリックします。

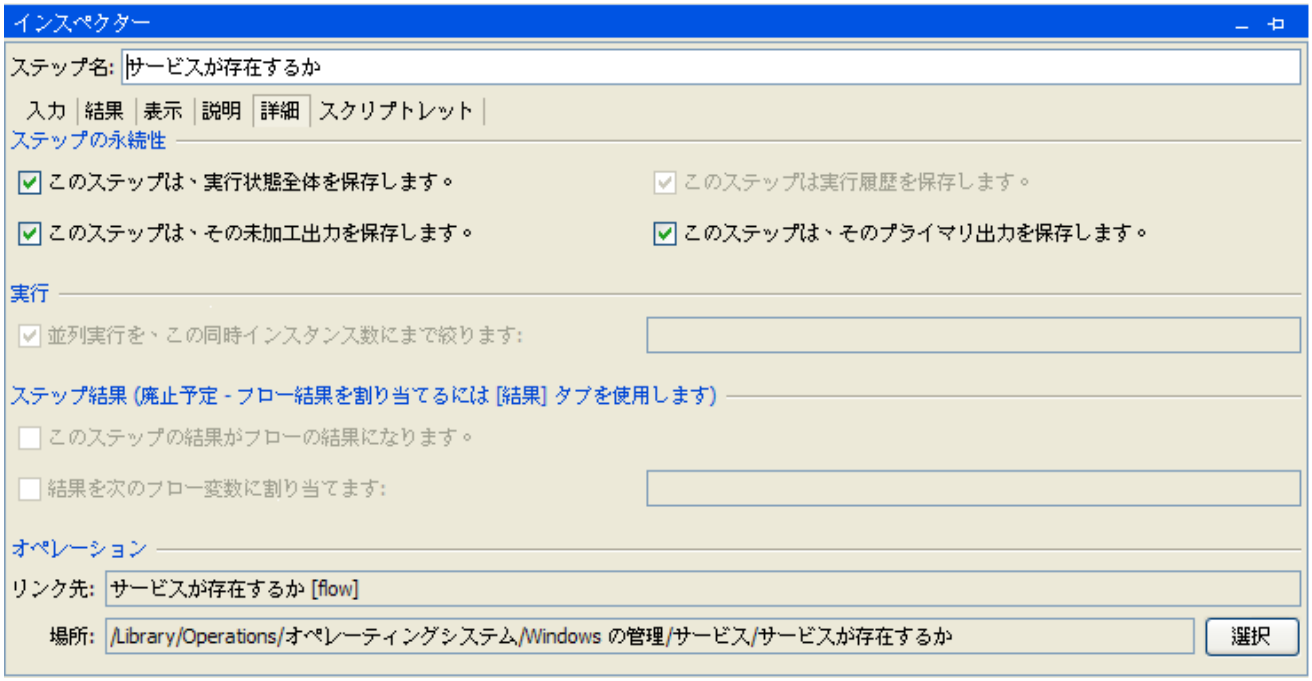


図 88 - ステップのインスペクターの [詳細] タブ

2. **[オペレーション]** で、**[選択]** ボタンをクリックします。

   [オペレーションの選択] ダイアログボックスが表示され、Studio 内に表示されるときのリポジトリ Library を表示します。

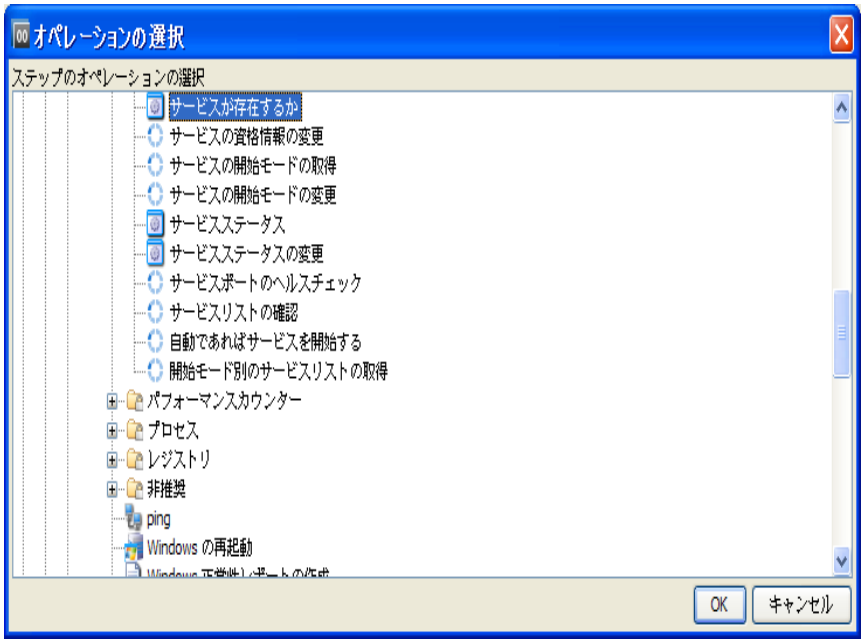


図 89 - [オペレーションの選択] ダイアログボックス

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、基底の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、**[オペレーション]** の**[リンク先]** ボックスと **[場所]** ボックスでは共に、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。

ステップの説明

説明の中に検索されそうな単語が入っていると、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。 [検索] 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 **[プロパティ]** シートまたはステップの**インスペクター**で、**[説明]** 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 [定数を使用する] 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 [定数を使用する] 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。

  入力用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、デバッガーのブレークポイントを設定し、テスト対象の動作に集中するためにレスポンスを強制的に選択することもできます。

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に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。

重要情報: 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際には、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 **[デバッグ]** をクリックします。

   または

   デバッグするフローの図を開き、**[作成]**ペインで、**[フローのデバッグ]** 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。

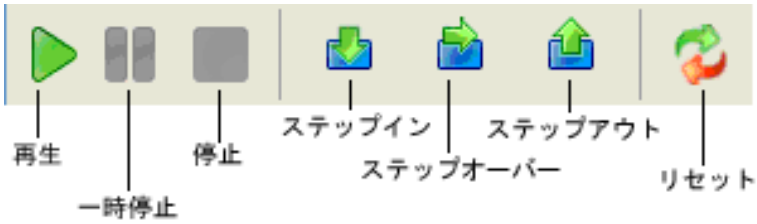


   図 90 - Studio デバッガーのツールバー

2. フローを終了まで実行するには、**[再生]** アイコン（ ）をクリックするか、F11 を押します。

   または

   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 **[ステップオーバー]** アイコン（ ）をクリックするか、F6 を押します。

   [Y2K およびそれ以前のファイルシステム診断] フローとデバッガーについて見てみましょう。

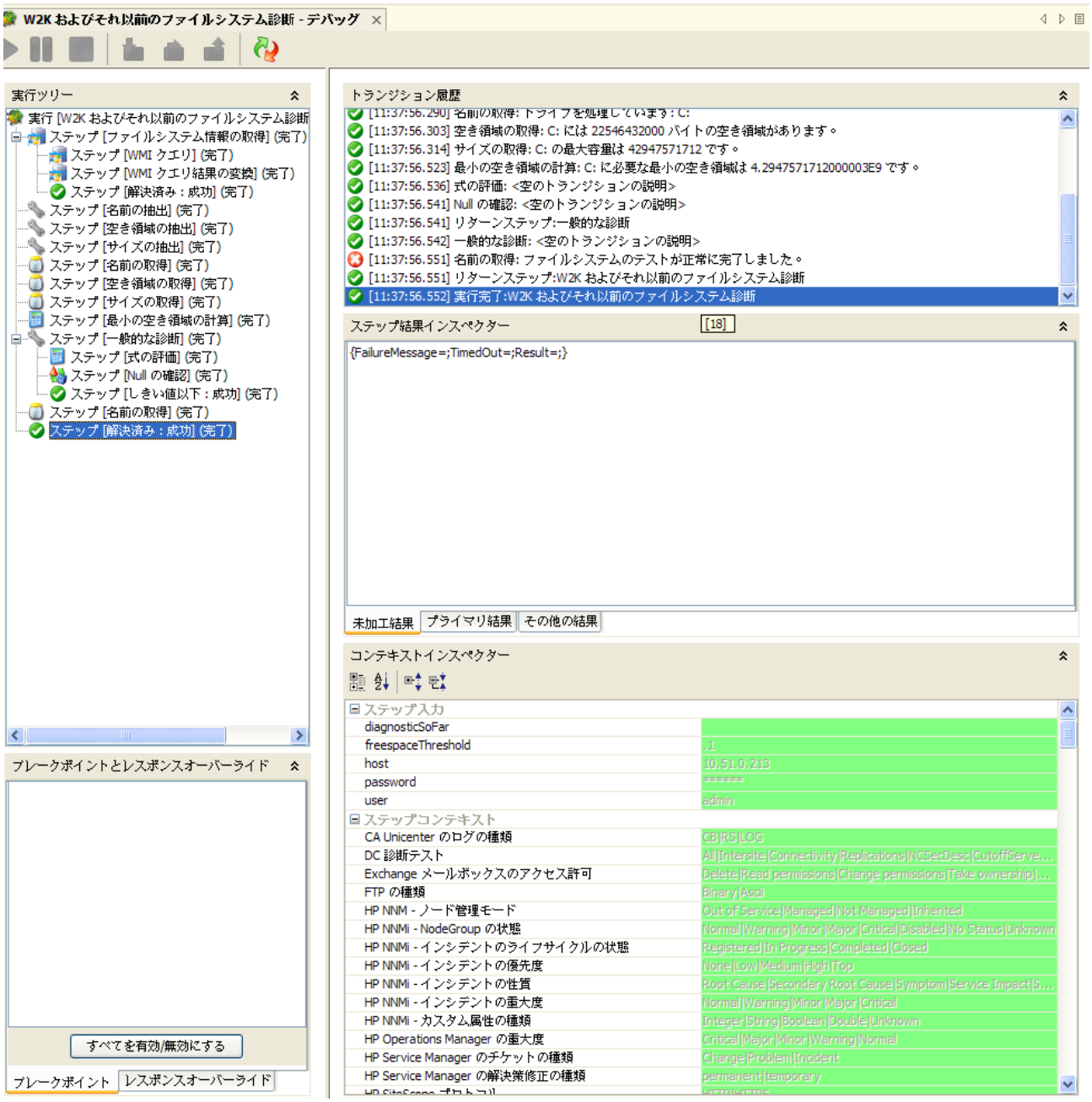


図 91 - フローデバッガーの一部

デバッガーは次のパネルに分類されます。

- **実行ツリー**

  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。

- **ブレークポイントとレスポンスオーバーライド**

  ブレークポイントは、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。

- **トランジション履歴**

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）。

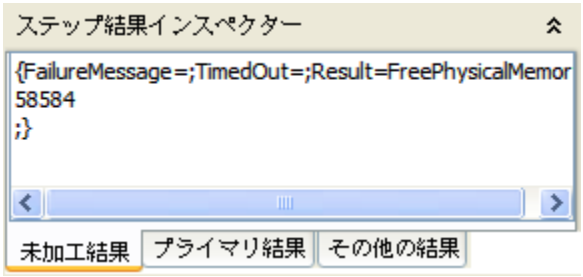
- **ステップ結果インスペクター**

  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。

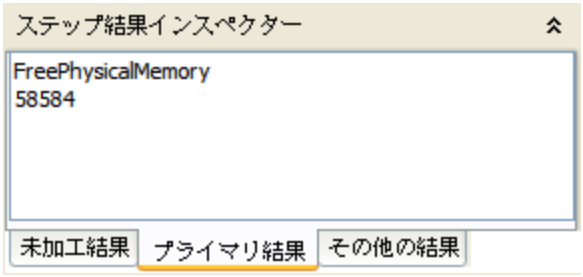
- **コンテキストインスペクター**

  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。

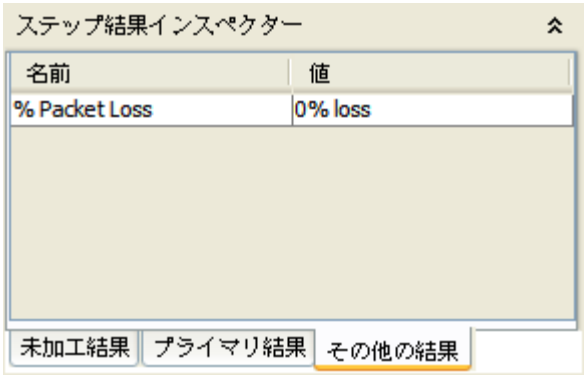
3. これらのパネルを（別のパネルを表示するためのスペースを確保するために）折りたたむ、または折りたたんだパネルを元に戻すには、パネルの右上にある二重矢印（≪または≫）をクリックします。
4. 完了したステップについて上記の各ペインの情報を確認するには、**[実行ツリー]** で目的のステップをクリックします。
5. **[ステップ結果インスペクター]** の説明は次のとおりです。
   - ステップの未加工結果を表示するには、**[未加工結果]** をクリックします。

   - 主結果を表示するには、**[プライマリ結果]** 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、**[その他の結果]** をクリックします。

6. グローバル変数、フロー変数、ステップの入力のこれらの変数値、ステップおよびグローバルコンテキストを確認するには、**[コンテキストインスペクター]** の該当セクションに移動します。

[コンテキストインスペクター] 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。
- **[ステップコンテキスト]** セクションの値は、ステップ開始後に更新された値です。

  ステップのコンテキストは、ステップのフローと親フローのローカルコンテキストに含まれるフロー変数と値の割り当ての集合です （フローが別のフローのステップである場合、2 つのフローの関係はサブフローと親フローになります）。
- **グローバルコンテキスト**は、システムプロパティと、作成済みのグローバル変数を比較します。

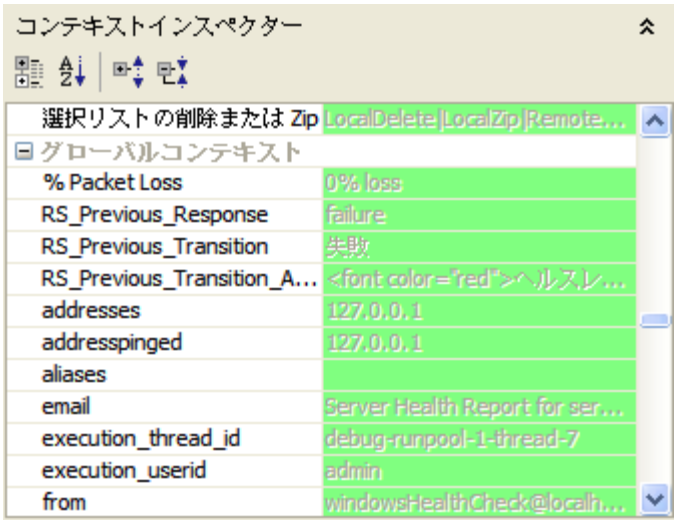


図 92 - グローバルコンテキストを表示するコンテキストインスペクター

- グローバルコンテキストには選択リストも含まれますが、これらは **[選択リスト]** セクションに表示されます。

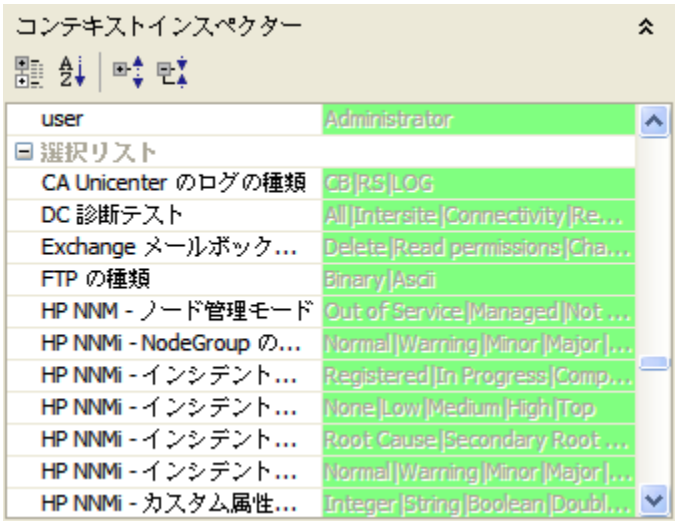


図 93 - 選択リストを表示するコンテキストインスペクター

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。

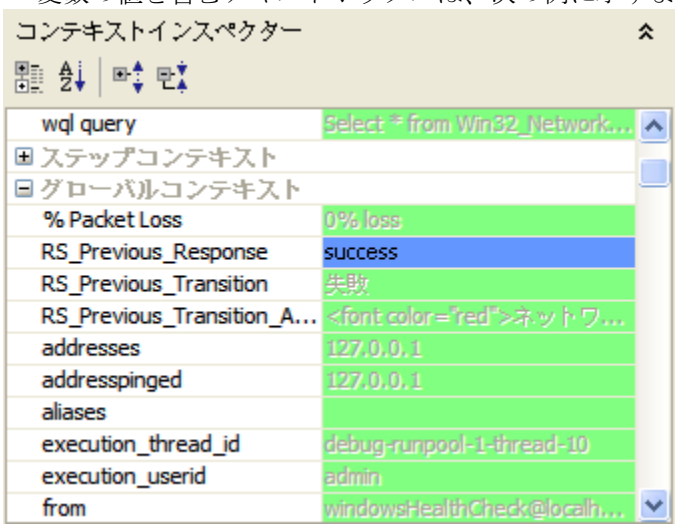


図 94 - フロー変数の色分け

- 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。

サブフローへのステップイン、またはサブフローからのステップアウト

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、**[ステップイン]** 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、**[ステップアウト]** アイコン（ ）をクリックするか、F7 を押します。

デバッガー内でのフローのリセットおよび再開

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。

デバッガーでフローをリセットして再開するには

1. ツールバーで、**[リセット]** アイコン（ ）をクリックするか、F12 を押します。
2. **[再生]** アイコン（ ）をクリックします。

デバッガーでのフロー変数の値の変更

フロー変数の値を変更したときにフローがどのように動作するかを確認する場合、ステップを実行する前にフロー変数の値を変更できます。

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、**[ステップオーバー]**（ 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、**[再生]**（ ）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。

   [コンテキストインスペクター] に、ステップが保留中になっているポイントの **[ステップ入力]** および **[ステップコンテキスト]** の現在の値が表示されます。

図 95 - コンテキストインスペクター

3. このステップで使用するフロー変数の値を変更するには、**[ステップ入力]** で、フロー変数の一覧を探し、その値を強調表示して、置き換えるための新しい値を入力します（リストの入力の場合は、新しい値がリストの末尾に追加されます）。

   上記の例では、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。

   または

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、**[ステップコンテキスト]** の下のフロー変数一覧の値を変更します。

4. フローの実行を続行します。
5. 変更したフロー変数の値を、前回のフロー保存時に設定した値にリセットするには、[リセット] 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。 これは、たとえば次のような処理を行うのに便利です。

- 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、**[デバッグ]** をポイントして **[ブレークポイントの設定]** をクリックします。

  フロー図内でブレークポイントは、ステップを囲む黄色と黒の枠で示されます。

図 96 - ブレークポイントを設定したステップ

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

1. デバッガーでフローを開くには、ツールバーの **[フローのデバッグ]** アイコン（ ）をクリックします。
2. デバッガーの **[ブレークポイントとレスポンスオーバーライド]** ペインの **[ブレークポイント]** タブに、既存のブレークポイントが表示されます。

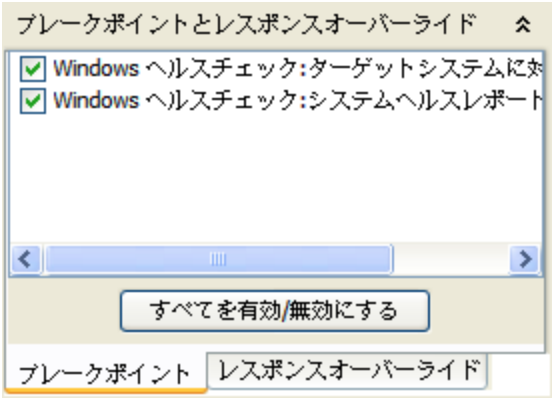


図 97 - ブレークポイントの有効化と無効化

3. 次のいずれかを実行します。
   - 1 つのブレークポイントを有効にするには、ブレークポイントのチェックボックスをオンにします。

- 1 つのブレークポイントを無効にするには、ブレークポイントのチェックボックスをオフにします。
- すべてのブレークポイントを有効または無効にするには、**[すべてを有効 / 無効にする]** をクリックします。
- すべてのブレークポイントをクリアするには、**[ツール]** メニューから **[すべてのブレークポイントの削除]** を選択します。

デバッグ実行でのレスポンスのオーバーライド

レスポンスをオーバーライドすると、デバッガーを終了して入力値を変更せずに、フローの特定のパスをテストできます。 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。 ユーザーが介入しない場合に発生する **失敗** レスポンスではなく、目的のレスポンスとトランジションをたどるように強制的に実行を続けることができます。
- 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、**成功** レスポンスをオーバーライドすることも可能です。

単一ステップにレスポンスのオーバーライドを設定するには

1. **[作成]** ペインでフローを開き、レスポンスをオーバーライドするステップを右クリックします。
2. ドロップダウンメニューで、**[デバッグ]** に続いて **[レスポンスのオーバーライド]** 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、**なし**、**成功**、**失敗**、および **プロンプト** です。

レスポンスのオーバーライドを作成した後、デバッガーでオーバーライドを有効または無効にしたり、別のレスポンスを選択したりできます。

レスポンスのオーバーライドを有効または無効にする、または別のレスポンスを選択するには

1. デバッガーでフローを開くには、ツールバーの [フローのデバッグ] アイコン（ ）をクリックします。
2. デバッガーの **[ブレークポイントとレスポンスオーバーライド]** ペインの **[レスポンスオーバーライド]** タブに、既存のレスポンスのオーバーライドが表示されます。

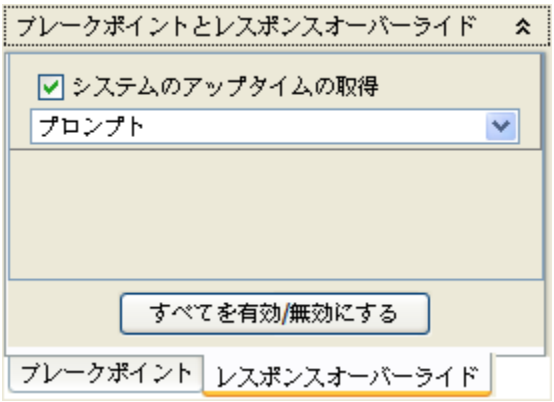


図 98 - レスポンスオーバーライドの使用

3. 目的のレスポンスオーバーライドにスクロールします。
4. 次のいずれかを実行します。
   - オーバーライドに別のレスポンスを選択するには、下向き矢印をクリックしてレスポンスを選択します。
   - 1 つのレスポンスオーバーライドを有効にするには、チェックボックスをオンにします。
   - 1 つのレスポンスオーバーライドを無効にするには、チェックボックスをオフにします。

- すべてのレスポンスオーバーライドを有効または無効にするには、**[すべてを有効/無効にする]** をクリックします。
- すべてのレスポンスオーバーライドをクリアするには、**[ツール]** メニューから **[すべてのレスポンスオーバーライドの削除]** を選択します。
- すべてのステップでレスポンスをオーバーライドするには、**[すべてのレスポンスのオーバーライド]** チェックボックスをオンにします。

  すべてのレスポンスをオーバーライドした後にデバッガーでフローを実行すると、各ステップで、ステップのレスポンスを手動で選択することを促すメッセージが表示されます。

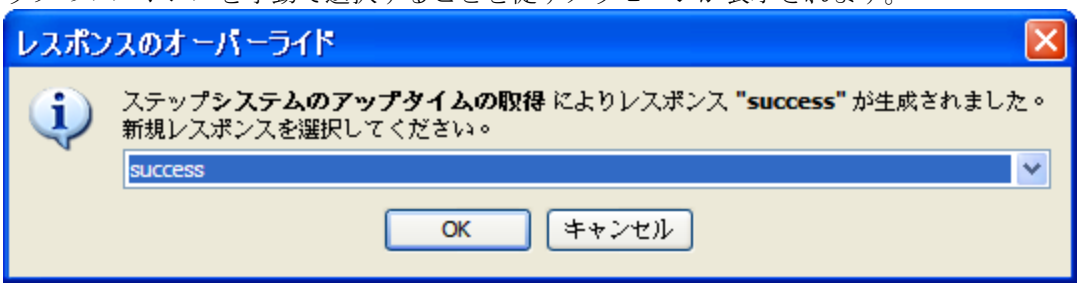


図 99 - レスポンスの手動での選択

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

- **[ファイル]** メニューで、**[ユーザーの切り替え]** をクリックし、別のユーザーとしてログオンします。

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： 1 つのユーザーアカウントで、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ないでください。 そのように接続した場合、リポジトリ内のユーザーの作業スペースが壊れることがあります。 この現象は、Central がクラスタ化されたときに

発生する可能性もあります。

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たら、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。

リポジトリ関連タスクの詳細については、以下を参照してください。

- *リポジトリの追加*
- *作成用にリポジトリを開く*
- *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*
- *リポジトリの検査*
- *リポジトリの暗号化*
- *リポジトリの復号化*
- *再暗号化されたリポジトリのコピーの作成*
- *リポジトリのバックアップと復元*
- *リポジトリのスナップショット*

リポジトリの追加

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。

複数のリポジトリを使用するシナリオについての説明は、「*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

1. **[リポジトリ]** メニューで、**[リポジトリの追加]** 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。

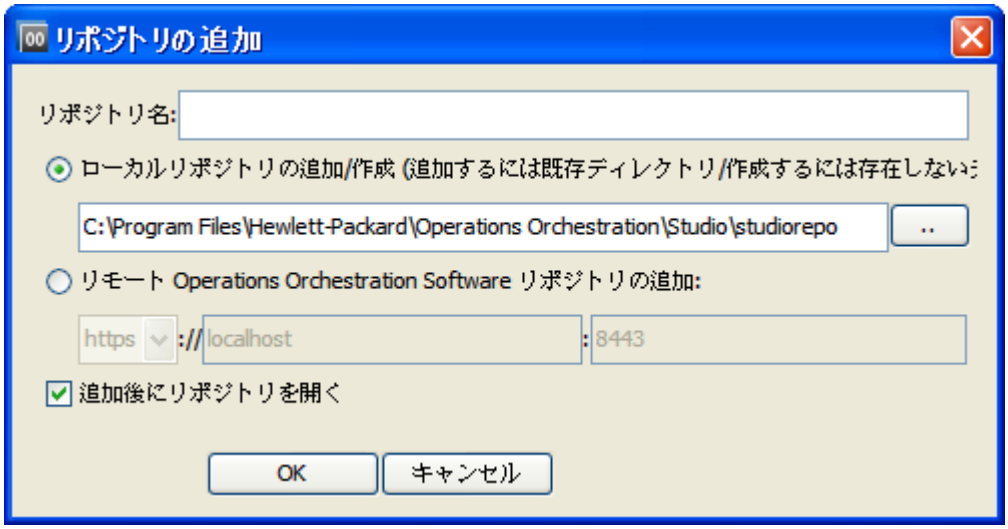


図 100 - リポジトリの追加

2. **[リポジトリ名]** テキストボックスに、リポジトリの名前を入力します。
3. ローカルリポジトリを追加するには、**[ローカルリポジトリの追加/作成]** を選択して、リポジトリフォルダーの場所を指定します。

   または

   Central サーバーにリポジトリを追加するには、**[リモート Operations Orchestration Software リポジトリの追加]** 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、**[追加後にリポジトリを開く]** 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

作成用にリポジトリを開くには

1. **[リポジトリ]** メニューで、**[リポジトリを開く]** 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せん。

- 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として設定できるのはパブリックリポジトリのみです。

ターゲットリポジトリを設定するには

1. 開いているパブリック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. **[リポジトリ]** メニューで、**[ターゲットリポジトリの設定]** 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。

   または

   ターゲットリポジトリを設定しない場合は、**[なし]** 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 [適用] 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。

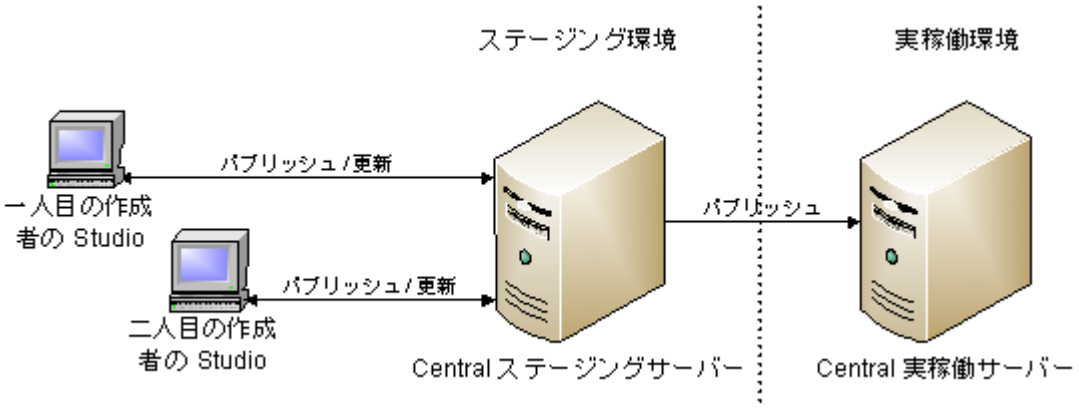


図 101 - 二人以上の作成者がフローを作成する場合の推奨手法

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）リポジトリに次のような経緯でパブリッシュするとします。

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。 [更新/パブリッシュのプレビュー] 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

1. 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。
2. **[Studio]** ウィンドウの下部にある **[パブリッシュ / 更新]** タブをクリックし、ピンのアイコン（ ）をクリックして **[パブリッシュ / 更新]** 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、**[パブリッシュ / 更新]** ペインのツールバーの左端で、**[更新のプレビュー]** アイコン（ ）または **[パブリッシュと更新のプレビュー]** 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、**[パブリッシュ / 更新]** ペインにその内容が一覧表示されます。
3. 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。

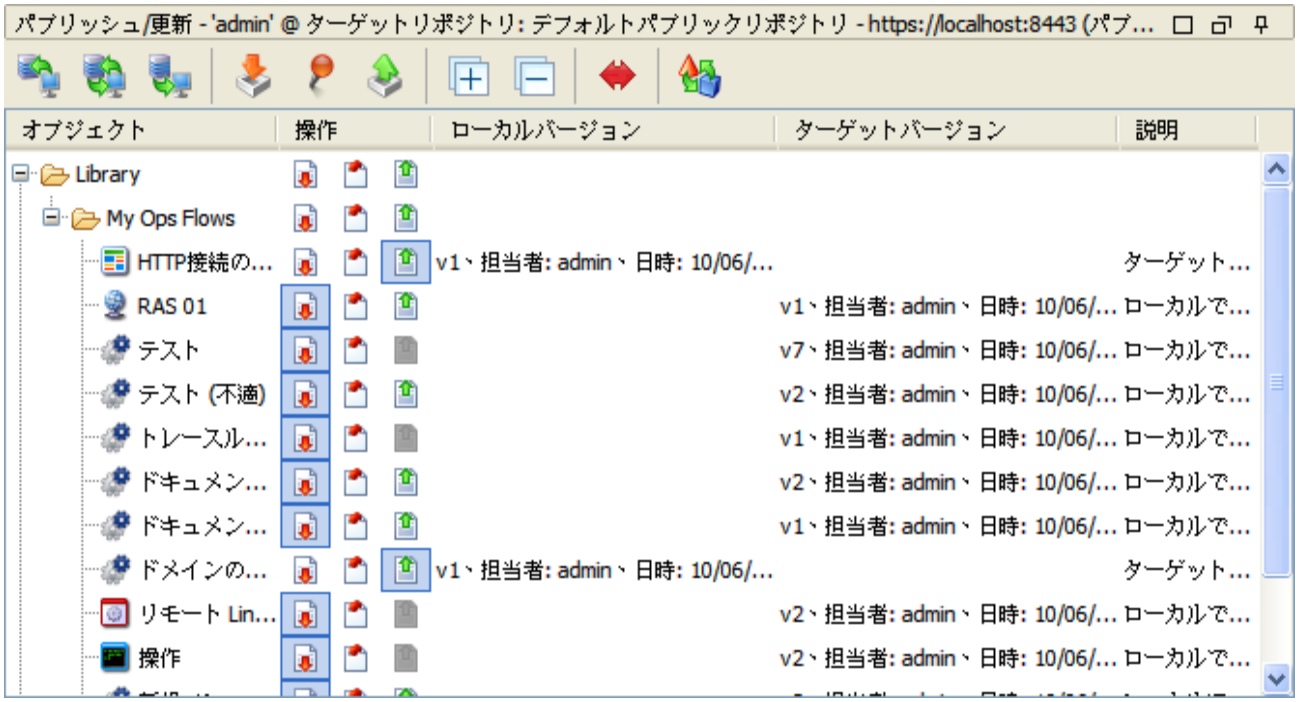


図 102 - [パブリッシュ / 更新] ペイン

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影響を受ける操作（[削除] または [追加]）と、変更が行われるリポジトリのパス名。
- 内容が異なるが、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 [変更なし] （）の操作が選択されています。

  注: スナップショットをパブリッシュする場合、スナップショットとターゲットリポジトリの間で異なるすべてのアイテムに対して [変更なし] 列()が選択されます。 相違をターゲットリポジトリに適用するには、適用する変更について()を選択します。
- オブジェクトのバージョン間に競合がある場合は、**[説明]** の下に表示されます。

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 **Testflow2**）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 **[ローカルで開く]** または **[ターゲットを開く]** をクリックします。 **[パブリッシュ / 更新]** 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。

   または

   両方のバージョンを横に並べて表示するには、そのオブジェクトの行で右クリックし、**[比較]** 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 :**[パブリッシュと更新のプレビュー]** 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、**[変更なし]** アイコン （） をクリックします。

   または

a. オブジェクトの行を右クリックします。
b. 次のようなコマンドが、コンテキストメニューに表示されます。

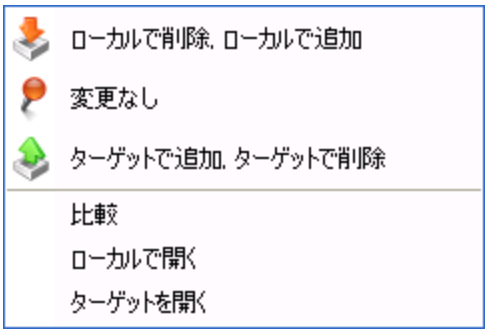


図 103 - [パブリッシュ / 更新のプレビュー] のコンテキストメニューコマンド

c. 適切なコマンドをクリックします。

6. 指定した変更を適用するには、**[適用]** アイコン（ ）をクリックします。
7. **[パブリッシュコメントの入力]** 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。

リポジトリの更新： 方法

リポジトリの更新は、リポジトリ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。 したがって、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。

リポジトリを更新するには

9. 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。

1. **[リポジトリ]** メニューから、**[ターゲットからソースの更新 - プレビュー]** を選択します。

   または

   [Studio] ウィンドウの下部にある **[パブリッシュ / 更新]** タブをクリックします。
2. **ピン**アイコン（ ）をクリックし、**[パブリッシュ / 更新]** 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、**[パブリッシュ / 更新]** 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していて、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、別のバージョンまたは Testflow1 のインスタンスを追加します。

現在、パブリックリポジトリ内の Testflow1 の ID は、ローカルリポジトリ内の 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。

3. 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。
4. 存在する可能性のある競合を検出するには、オブジェクトの行の **[説明]** 列を調査します。

   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。

   この図では、いくつかある競合の中の一つとして、ローカルとターゲットのリポジトリ間で統合の通知のバージョンが競合しています。

図 104 - 競合が存在する [パブリッシュ / 更新] ペイン

5. 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（この場合、統合の通知）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 **[ローカルで開く]** または **[ターゲットを開く]** をクリックします。 **[パブリッシュ / 更新]** 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。

   または

   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（統合の通知）の行で右クリックし、**[比較]** 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）。

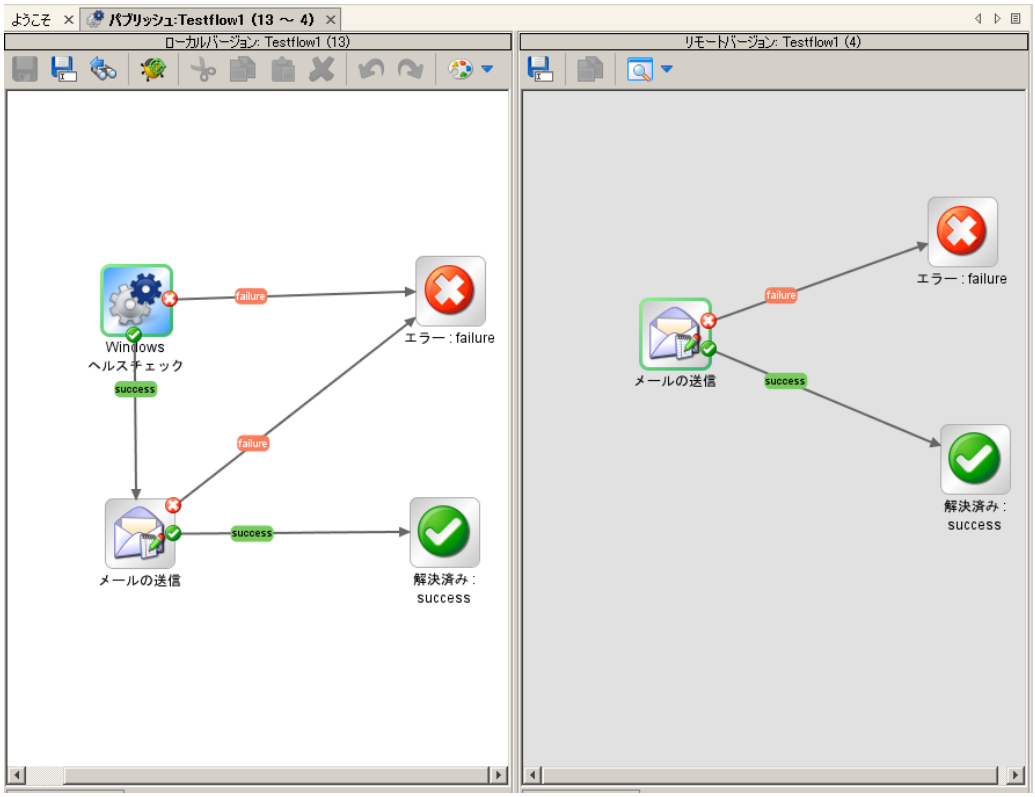


図 105 - フローの 2 つのバージョンの比較

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 :**[パブリッシュと更新のプレビュー]** 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、下向き矢印 ( ) 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、**[変更なし]** アイコン ( ) をクリックします。

または

オブジェクトの行を右クリックします。

次のようなコマンドが、コンテキストメニューに表示されます。

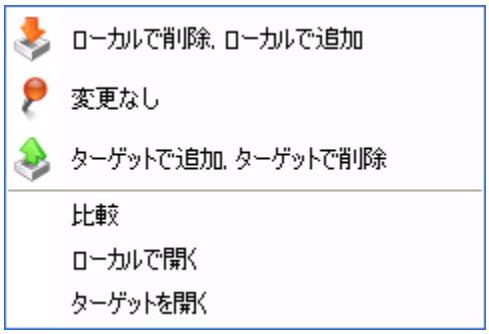


図 106 - [パブリッシュ / 更新のプレビュー] のコンテキストメニューコマンド

適切なコマンドをクリックします。

7. 指定した変更を適用するには、[適用] アイコン（ ）をクリックします。

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た後は、リポジトリを開いてバックアップからインポートすることで、不要な変更を効率よく元に戻すまたはロールバックできます。

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。

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をバックアップ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、**[リポジトリ]** をポイントして、**[新規リポジトリとしてエクスポート]** をクリックします。
2. **[リポジトリディレクトリの選択]** 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. **[ファイル名]** ボックスに、フォルダー名を入力します。
   重要： 既存のフォルダーは指定できません。
4. 新しいフォルダーを開き、次に **[保存]** をクリックします。
   [エクスポートオプション] ダイアログボックスが表示されます。

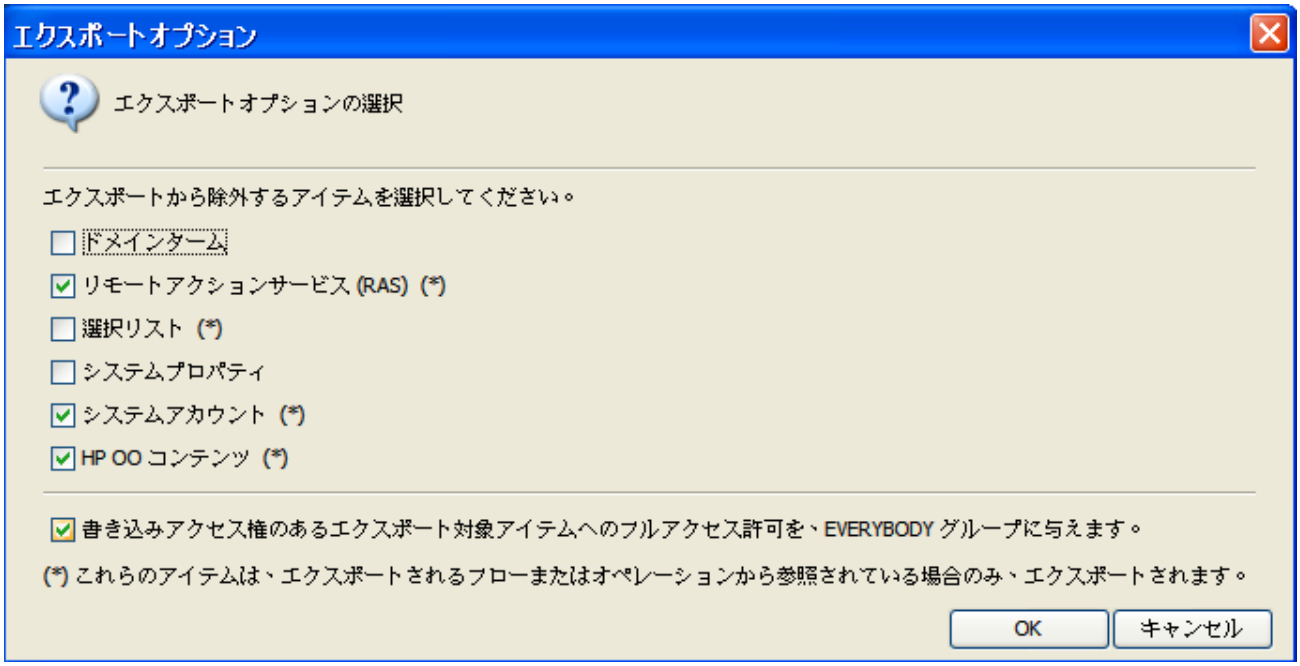


図 107 - エクスポートのオプション

5. **[エクスポートオプション]** 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。

   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。

   エクスポートするもの、しないもの、およびその理由。

   - 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 サービスウィザード（Web サービスウィザードの詳細については、「*Web サービスからのオペレーションの作成*」を参照）をポイントする場合、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。 それらをポイントする場合は、リポジトリに SSH シェルまたは Telnet シェル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 シェルまたは Telnet シェルオペレーションは、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。

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、**[書き込みアクセス権のあるエクスポート対象アイテムへのフルアクセス許可を、EVERYBODY グループに与えます。]** 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 アクセラレーターパック 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。

次のことに留意します。

- 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 箇所にしか存在できません。 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 Library 内の任意の場所に配置できます。
- エクスポートされたリポジトリは Studio の外部に存在するので、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. **[リポジトリ]** メニューで、**[リポジトリのインポート]** を選択します。
3. **[作成]** ペインでフローまたはオペレーションを開いていて、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た場合は、**[OK]** をクリックします。

   [リポジトリディレクトリの選択] (またはリポジトリを含む .jar / .zip) ダイアログボックスが開きます。
4. ダイアログで、インポートするリポジトリ、.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、次に **[開く]** 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、**[...からインポートしています]** ダイアログボックスが開きます。

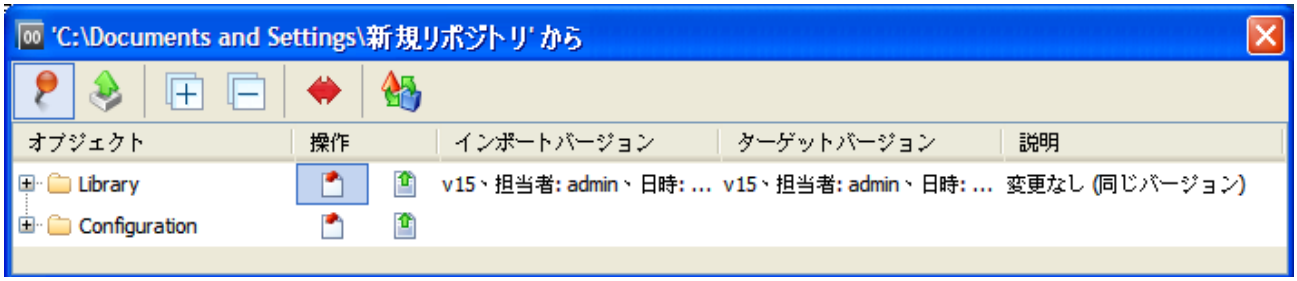


図 108 - ダイアログボックスからのインポート

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 **[操作なし]** （）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。

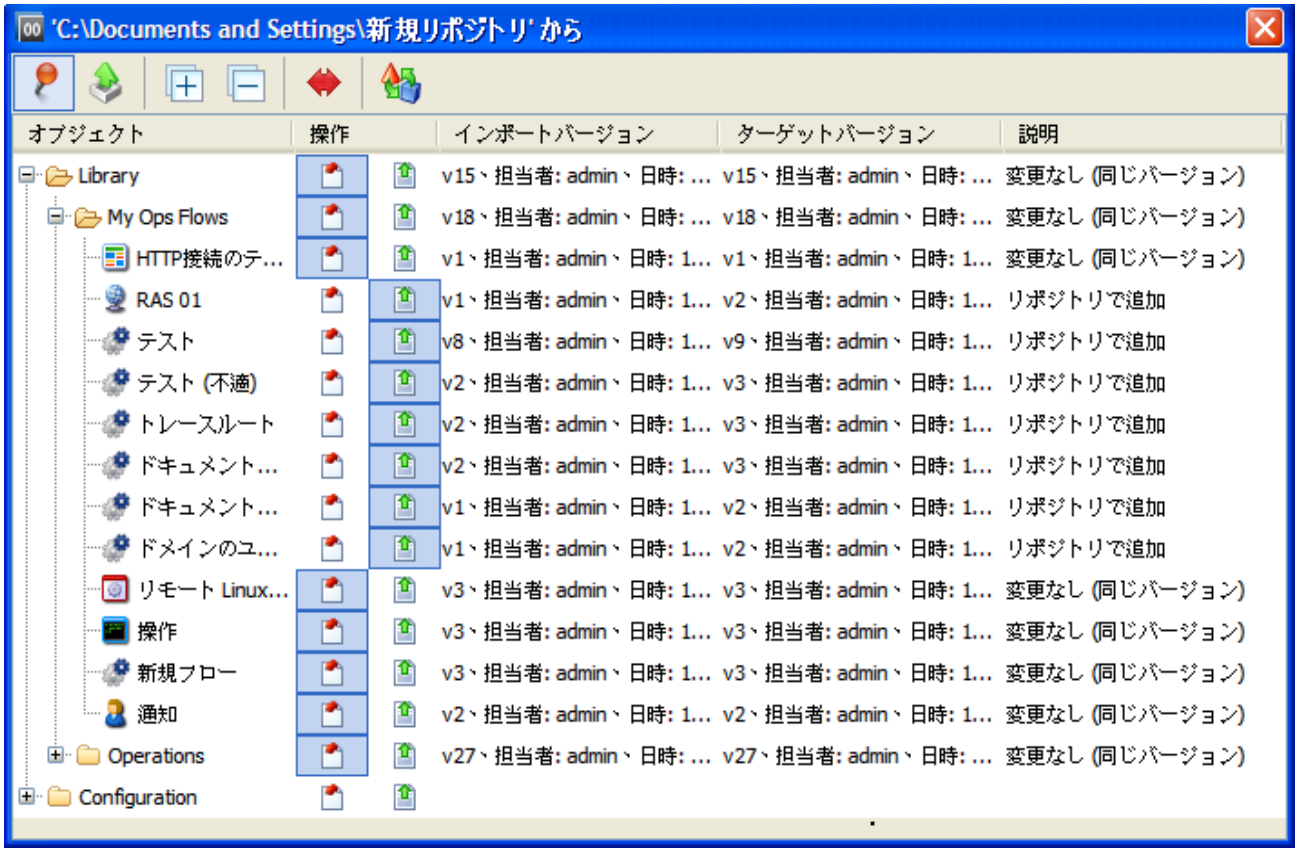


図 109 - インポートするオブジェクトの選択

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 **[変更なし]** アイコン（）が強調表示されています。

  [適用] 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: **[競合だけを表示]** 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、**[操作]** の下で、適切な行の上向き矢印アイコン（）をクリックします。

   または

   ソースのオブジェクトを現在のままにしておくには、**[変更なし]** 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 **[変更なし]** 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、**[適用]** ボタン（ ）をクリックします。

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、**[問題]** 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。

ヒント: 特定のタスクについてバックグランドで続行可能なリポジトリを検査します。

リポジトリを検査するには

- リポジトリを開いた状態で、**[ツール]** メニューで、**[フローとオペレーションの検査]** をクリックします。

  [問題] 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

1. 暗号化するリポジトリを開いた状態で、**[リポジトリ]** メニューで **[リポジトリの暗号化]** 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。

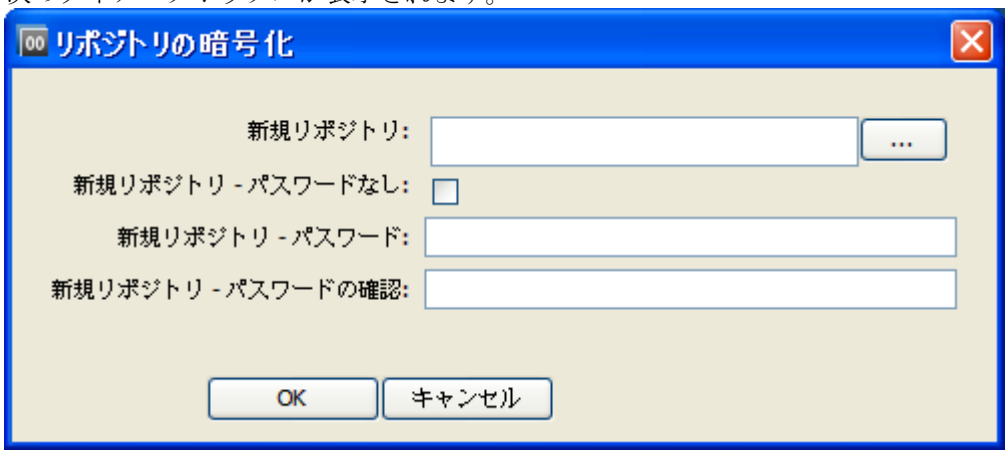


図 80 - リポジトリの暗号化

2. **[新規リポジトリ]** 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。
   または
   [...] 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

リポジトリの暗号化されたコピーを初めて開く場合、開く前にリポジトリを追加する必要があります。

暗号化されたリポジトリを初めて開くには

1. **[リポジトリ]** メニューで、**[リポジトリの追加]** をクリックします。
2. 暗号化されたコピーに移動し、次に **[開く]** をクリックします。
3. 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。

既に追加済みの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、**[リポジトリ]** メニューから、**[リポジトリを開く]** 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。
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. **[リポジトリ]** メニューから、**[リポジトリの復号化]** をクリックします。

異なるパスワードで、2 回目の暗号化を行い、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。

再暗号化されたリポジトリのコピーの作成

再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. **[リポジトリ]** メニューで、**[リポジトリの再暗号化]** をクリックします。

[リポジトリの再暗号化] ダイアログボックスが表示されます。

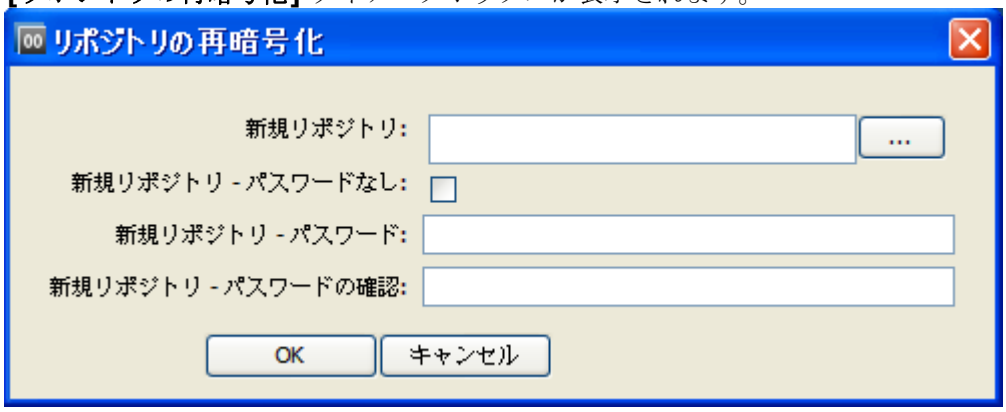


図 110 - リポジトリの再暗号化

3. **[新規リポジトリ]** 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。
   または
   [...] 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. **[リポジトリ]** メニューで、**[バックアップの作成]** を選択します。
3. **[バックアップファイルを選択]** ダイアログボックスで、バックアップを保存するフォルダーに移動します。
   フォルダーは空である必要があります。
4. **[保存]** をクリックします。

リポジトリを復元するには

1. **[リポジトリ]** メニューで、**[リポジトリのインポート]** を選択します。
2. **[リポジトリディレクトリの選択(または .jar / .zip...]** ダイアログボックスで、復元するバックアップに移動し、次に **[開く]** をクリックします。
   リポジトリを含むフォルダーには 3 つのサブフォルダーがあります。 **.idx**、**tmp**、および **data** です。
   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。
3. 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。

リポジトリスナップショット

リポジトリは絶えず流動的で、フロー作成チームによりレポジトリに対して頻繁に追加および切り捨てられ、フローおよびオペレーションを改善または適応しています。 スナップショット機能により、任意の時点の状態を取り込んで後から表示することで、パブリックリポジトリの初期の状態を見ることができます。 スナップショット

を現在のリポジトリにパブリッシュすることで、現在のパブリックリポジトリをスナップショットに戻すことができます。 するとスナップショットは現在の状態になり、その中で選択する任意の作業を行うことができます。

スナップショットのパブリッシュ先のパブリックリポジトリは、スナップショットで取得したものか、または別のもののいずれかです。 たとえば、開発環境およびテスト環境の両方に OO インストールがある場合、開発環境のパブリックリポジトリのスナップショットを次のいずれかにパブリッシュできます。

- 自分自身、それにより開発のパブリックリポジトリはスナップショットを取得した時点の状態に戻ります。

  または

- テスト環境のパブリックリポジトリに対して。

  リポジトリがテストのために移行され、そのフローのいくつかが失敗する場合、リポジトリの前回正常実行状態のスナップショットがあると、その開発環境のパブリックリポジトリにおける前回正常実行状態のバージョンのスナップショットを、テスト環境のパブリックリポジトリにパブリッシュできます。

重要：

- スナップショットで作業を始める前に、リポジトリとリポジトリのパブリッシュについて十分な理解をする必要があります。
- スナップショットはパブリックリポジトリからのみ取得でき、パブリックリポジトリに対してのみパブリッシュできます。
- スナップショットは、取得された時点のリポジトリの状態を保持します。 所定の時点におけるリポジトリの状態には、作成されたすべてのオブジェクト、およびチェックインされたオブジェクトへのすべての変更が含まれます。 作成されたがチェックインされていないオブジェクトもスナップショットの一部になります。なぜなら、オブジェクトを作成した時点でリポジトリに作成されるからです。 このように、リポジトリの現在の状態の一部は、スナップショットによって保存されます。 ただし、オブジェクトに変更を行う場合、チェックインされるまでは変更はリポジトリの現在の状態の一部ではありません。
- スナップショットをパブリックリポジトリにパブリッシュする場合、スナップショットの取得時点でチェックインされていないすべてのオブジェクトは、パブリックリポジトリに適用できません。
- スナップショットは、単一ユーザーの作業スペースを取得するのではなく、リポジトリ全体を取得します。したがって、スナップショットを表示するとパブリックリポジトリ全体が表示されます。その状態の中では、制作者の作業で直近のチェックインが続きます。 スナップショットをパブリッシュする場合、スナップショットとターゲットリポジトリの間で異なるすべてのアイテムに対して **[変更なし]** 列( )が選択されます。

  相違をターゲットリポジトリに適用するには、適用する変更について( )を選択します。次に **[変更を適用]** アイコン( )をクリックします。 パブリッシュの詳細については、「*リポジトリのパブリッシュ：方法*」を参照してください。
- スナップショットの作成、名前または説明の編集、スナップショットをパブリッシュまたは削除するには、OO Central ADMINISTRATOR または PROMOTOR グループのメンバーである必要があります。

スナップショットの作成

リポジトリのスナップショットを取得するには

1. **[リポジトリ]** メニューで、**[スナップショットの管理]** をクリックします。

   [スナップショットの管理] ダイアログボックスが開きます。

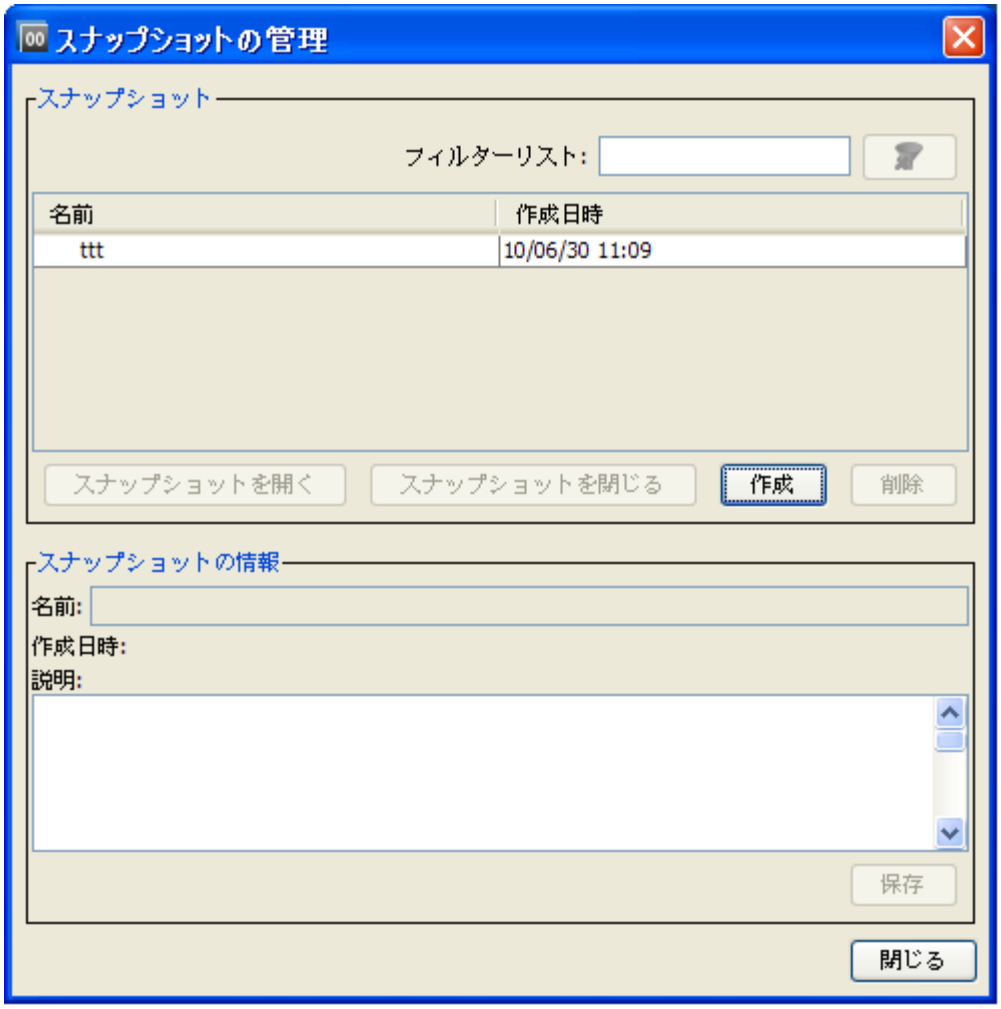


図 111 - リポジトリの状態を取り込む

2. **[作成]** をクリックします。
   [新規スナップショットの作成] ダイアログボックスが開きます。

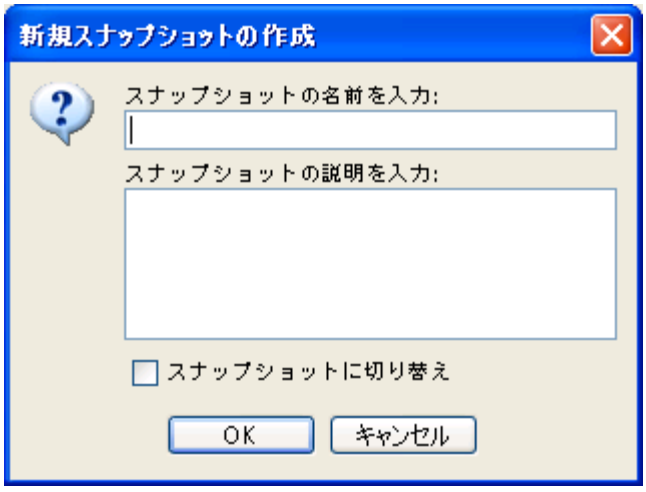


図 112 - スナップショットの名前と説明の作成

3. スナップショットの名前と説明を、それぞれ適切なボックスに入力します。

ベストプラクティス： OO インストールの開発方法およびトポロジーによって、スナップショットが保持している状態や、もしあれば同期している他のスナップショットといった、スナップショットの説明が一目で確認できる情報を表すことは非常に重要な場合があります。

4. スナップショットを作成した後に、それが **[スナップショットの管理]** スナップショットのリストで選択された、Studio に表示されるよう指定するには、**[スナップショットに切り替え]** を選択します。
5. **[OK]** をクリックし、次に **[スナップショットの管理]** ダイアログで、**[閉じる]** をクリックします。

スナップショットの名前や説明の変更

スナップショットの名前や説明を変更するには

1. **[リポジトリ]** メニューで、**[スナップショットの管理]** をクリックします。
2. **[スナップショットの管理]** ダイアログボックスの、**[スナップショット]** の下のリストで、情報を編集するスナップショットを選択します。
3. **[名前]** およびまたは **[説明]** ボックスで目的の変更をして、次に **[保存]** をクリックします。

スナップショットの表示

スナップショットは、リポジトリの読み取り専用のビューです。 スナップショットを表示することで、パブリックリポジトリの現在の状態になることはありません。

(パブリックリポジトリの状態をスナップショットの状態に変更して、それに変更を加えるには、スナップショットをリポジトリにパブリッシュします。 スナップショットのパブリッシュの詳細については、「*スナップショットからの復元* を参照してください。)

スナップショットを表示するには

1. **[リポジトリ]** メニューで、**[スナップショットの管理]** をクリックします。
   [スナップショットの管理] ダイアログボックスが開きます。

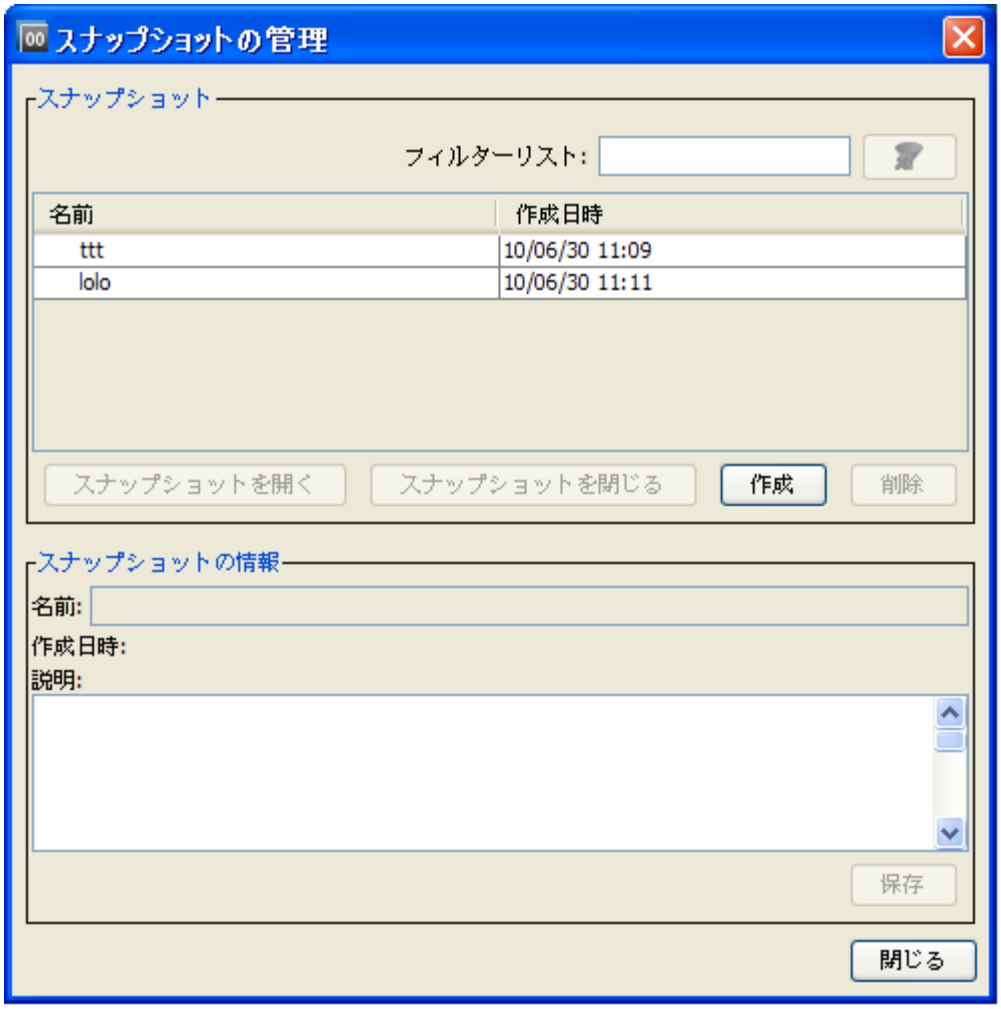


図 113 - 表示するスナップショットの選択

2. **[名前]** で、表示するスナップショットを選択します。
3. **[スナップショットを開く]** をクリックして、次に **[閉じる]** をクリックします。

スナップショットを閉じる

現在のリポジトリの状態に戻すには

1. **[スナップショットの管理]** ダイアログボックスを開きます。
2. **[スナップショットを閉じる]** をクリックして、次に **[閉じる]** をクリックします。

スナップショットのリストのフィルタリング

スナップショットのリストを、指定する文字列でフィルターできます。

注: フィルターする文字列は、**[名前]**、**[作成日時]**、および **[説明]** ボックスに対して試行され、それらのボックスのいずれかに文字列がヒットしたかに関係なくフィルターされたリストがスナップショットに含まれるようにします。

スナップショットのリストをフィルターするには

1. **[スナップショットの管理]** ダイアログの **[フィルターリスト]** テキストボックスに、リストに表示されたスナップショットに含まれるようにする文字列を入力します。

2. テキストボックスに入力すると、**[名前]**、**[作成日時]**、**[説明]** ボックスに該当する文字列があるかどうかが試行され、リストが文字列でフィルターされます。

   たとえば、次のスナップショットがあると仮定します。

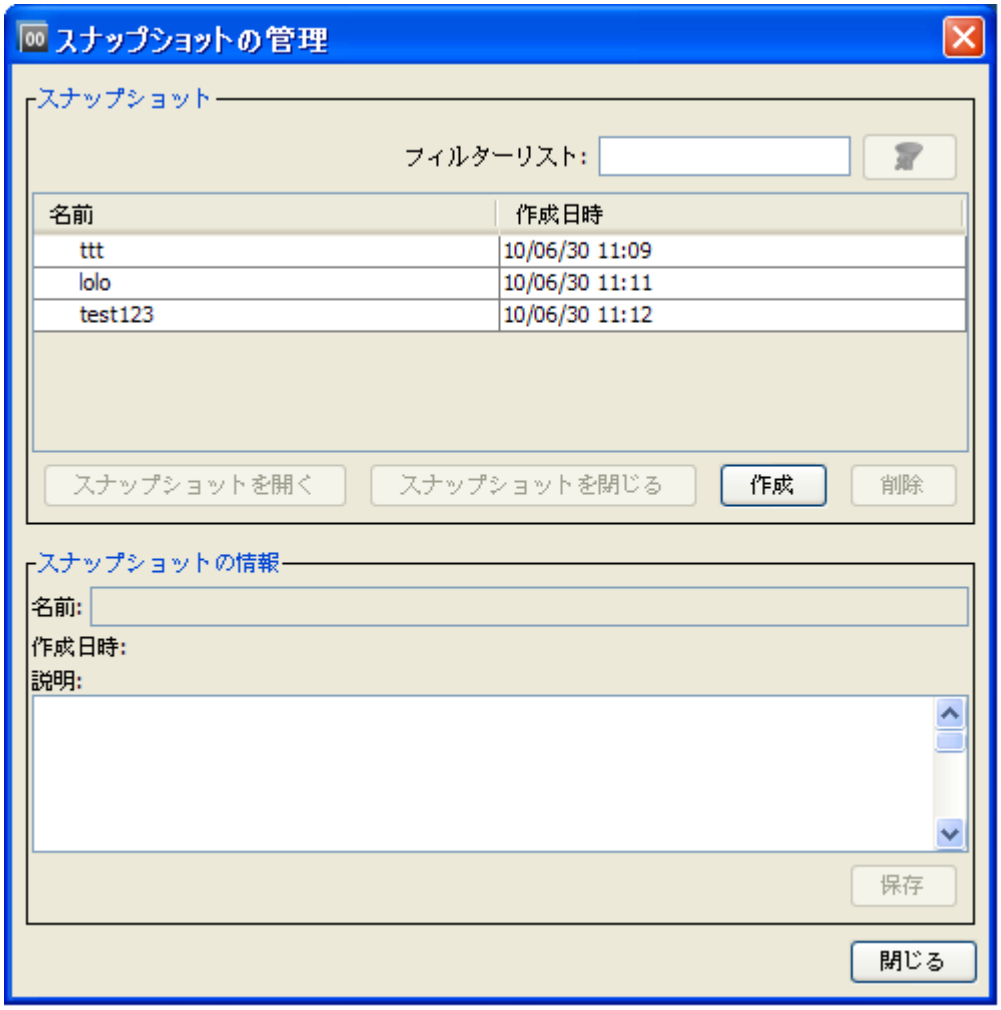


図 114 - フィルター前のスナップショットのリスト

3 番目以外をすべてフィルターするには、テキストボックスに「123」と入力します。

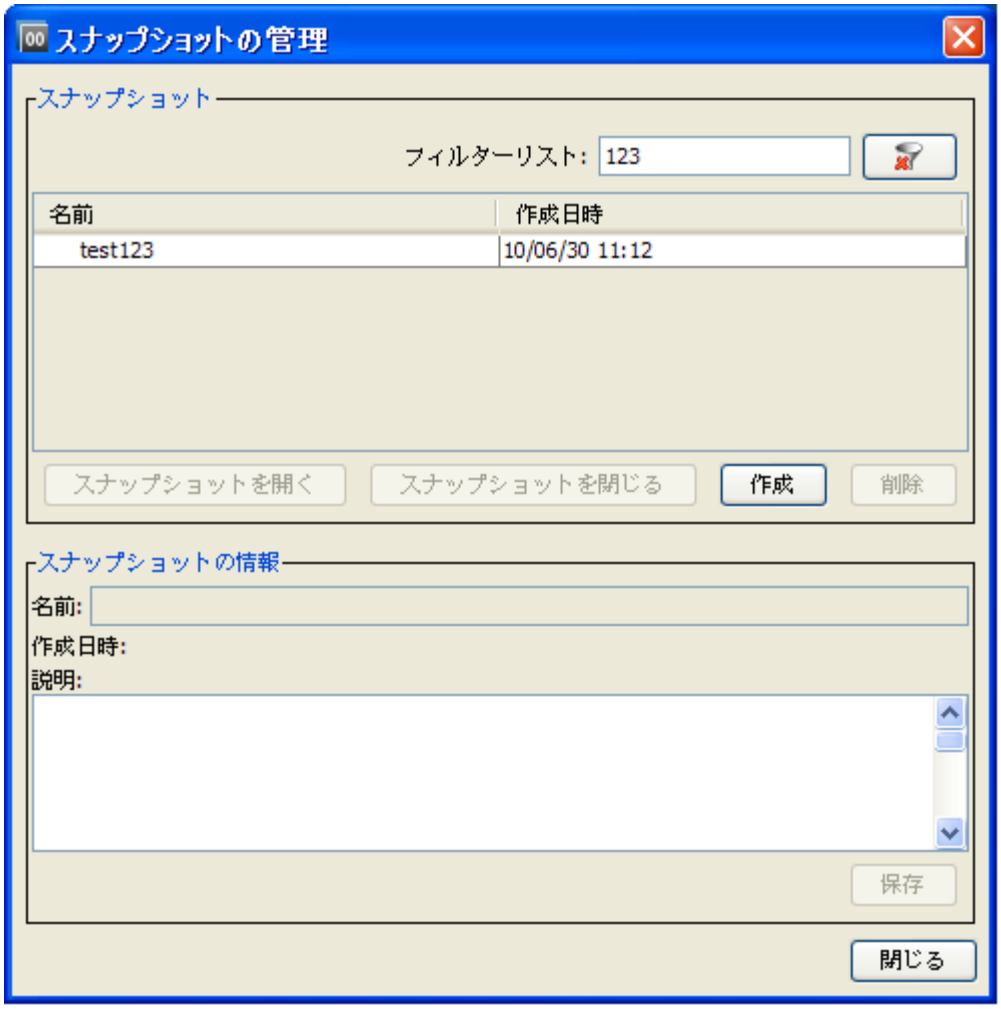


図 115 - 文字列「123」によるスナップショットのリストのフィルタリング

3. フィルターを削除するには、テキストボックスが空になるまでバックスペースを押します。
   または
   [フィルターの削除] ボタン( )をクリックします。

スナップショットの削除

スナップショットのリストが増えるにつれて、削除を行いたいと思うようになるかもしれません。

スナップショットを削除するには

1. **[スナップショットの管理]** ダイアログボックス(**[リポジトリ]** メニューの **[スナップショットの管理]** コマンド)で、削除するスナップショットを選択します。
2. **[削除]** をクリックします。
3. 削除の確認が表示されたら、**[はい]** をクリックします。

スナップショットからの復元

スナップショットをパブリックリポジトリにパブリッシュすることは、リポジトリの状態をスナップショットの状態に変更する手段です。 スナップショットに切り替えるだけでは、スナップショットはリポジトリの現在の状態になりません。

スナップショットがリポジトリの現在の状態になると、リポジトリのフロー、オペレーション、および他のオブジェクトをチェックアウトして通常と同じように作業を行うことができます。 スナップショットがリポジトリのみに対して適用された場合、その読み取り専用のビューが取得され、そのオブジェクトに対して作業することはできません。

注:

- パブリックリポジトリのスナップショットに対してのみ作業を行えることを覚えておいてください。
- パブリックリポジトリに対してパブリッシュするには、パブリッシュ先のパブリックレジストリを所有している OO インストールの ADMINISTRATOR または PROMOTER グループのメンバーである必要があります。
- スナップショットは、現在作業中のパブリックリポジトリまたは別のパブリックリポジトリのいずれかにパブリッシュできます。

スナップショットから復元するには

1. **[スナップショットの管理]** ダイアログボックス(**[リポジトリ]** メニューの **[スナップショットの管理]** コマンド)で、目的のスナップショットを選択して、次に **[スナップショットを開く]** をクリックします。
2. パブリックリポジトリになるターゲットリポジトリ(そのスナップショットは現在の状態になります)を設定します(**[リポジトリ]** メニューで、**[ターゲットリポジトリの設定]** コマンド)。
   [パブリッシュ / 更新 – ターゲットリポジトリ] パネルが開きます。
3. ターゲットリポジトリにスナップショットをパブリッシュするには、**[パブリッシュのプレビュー]** アイコン ( ) をクリックします。
   プレビューの、**[操作]** では、すべての相違に対して **[操作なし]** ( ) がデフォルトで選択されています。

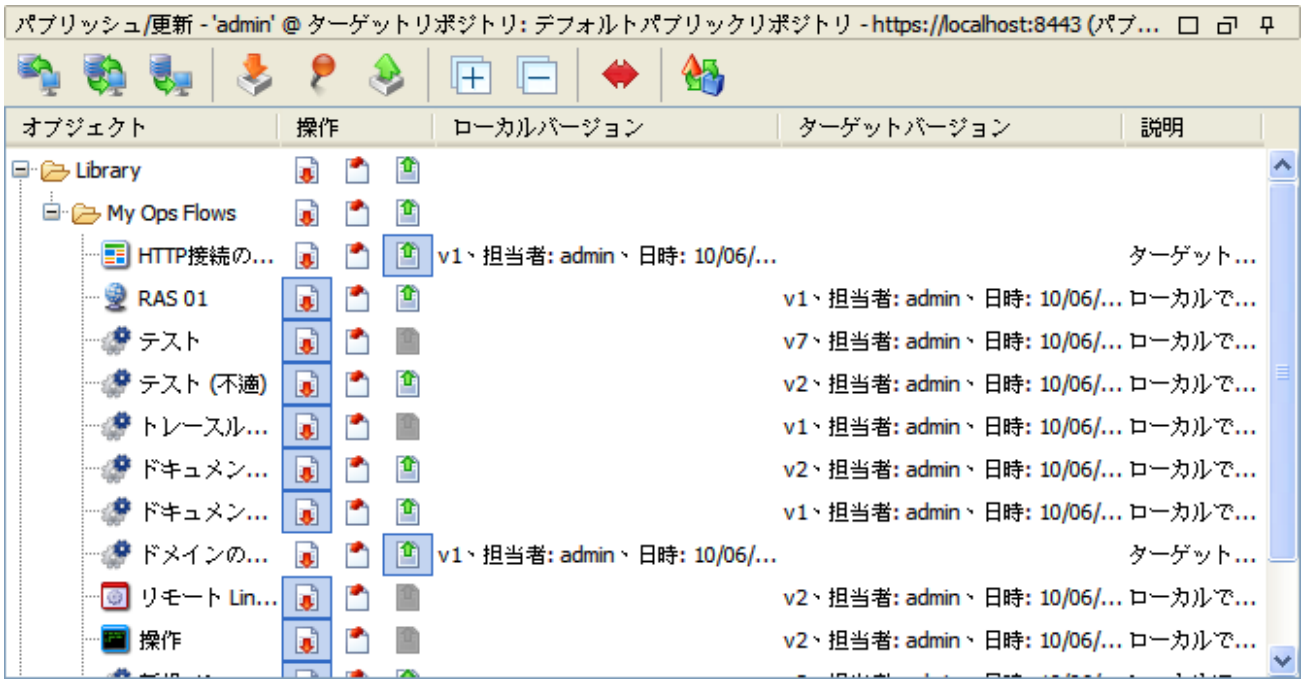


図 116 - スナップショットのパブリッシュのプレビュー

4. ターゲットリポジトリにおいて現在の状態の一部にするアイテムそれぞれについて **[送信]** アイコン( )をクリックし、次に **[適用]** アイコン( )をクリックします。
   パブリッシュしたばかりのスナップショットの状態は、ターゲットリポジトリの現在の状態です。
5. パブリックリポジトリの新しい現在の状態を表示するには

a. **[スナップショットの管理]** の、スナップショットのリストで、パブリッシュしたばかりのスナップショットを選択します。
b. **[スナップショットを閉じる]** をクリックします。
c. **[スナップショットの管理]** ダイアログボックスを閉じるには、**[閉じる]** をクリックします。

Web サービスからのオペレーションの作成

Web サービスウィザードは、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）で記述された 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）。

WSDL とは、Web サービスへの要求の説明と Web サービスからの応答です。 簡易オブジェクトアクセスプロトコル (SOAP) に基づいており、かつ Web Services Interoperability (WS-I) の標準に適合しているすべての Web サービスには、WSDL があります。

Web サービスウィザードを実行する場合、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 MyAlert と呼ばれるアプリケーションがあり、MyAlert でチケットを作成するとします。 Web サービスウィザードは、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 MyAlert）を指定して実行すると、Web サービスで実行できるフローおよびオペレーションが生成されます。 Studio の Library に、必要な入力が作成された状態でフローおよびオペレーションが表示されます。 フローの実行には、Web サービスをポイントするリモートアクションサービス (RAS) 参照が必要になります。 RAS 参照の作成と設定については、「*リモートアクションサービスを使用した Central 外でのオペレーション*」および関連するサブセクションを参照してください。

注:

- Web サービスウィザードは SOAP UI ユーティリティを使用して WSDL を XML に解析します。
- 解析では、入力のほか、オペレーション出力（ステップの結果として表現される）として取り込める結果も取得されます。 XML の配列は単一の XML 結果としてのみ抽出され、フロー作成者はそこから絞り込んで抽出できます。
- Web サービスウィザードが生成するフロー内では、Web サービスに対応するフローの入力はオプションです。 Web サービスの定義によって必要であるとされている入力が、実際には必要ではないという場合がよくあります。このような設定をフローに適用すると、フローの実行時に、ユーザーに使用されない値を入力させることになります。 したがって、Web サービスウィザードはすべての入力をオプションに設定しています。 Web サービスが フィールドがオプションであると示している場合、フィールドの前に "<!--Optional:-->" または "<!--zero or more repetitions-->" というコメントがつきます。 必要とされる入力の詳細については、オペレーションを作成する Web サービスのドキュメントを参照してください。
- アクセスする WSDL の Web サービスが Studio マシンから見てファイアウォールの反対側にある場合、その Web サービスに到達するために使用する HTTP プロキシを指定する必要があります。 その方法については、次の手順で説明します。

Web サービスウィザードで Web サービス用のオペレーションを作成するには

1. プロキシを指定する必要がある場合:
   a. OO ホームディレクトリの **\Studio\tools\conf** フォルダーにある **wsw.properties** ファイルをテキストエディターで開きます。
   b. 次の行を探し、非コメント化（先頭のポンドまたはハッシュ [#] 記号を削除）します。

```
# http.proxyHost=myproxyhost.com
# http.proxyPort=8080
```

c. 各行で、等号（=）記号の右側の文字列を適切な値に置き換えます。 それぞれ、HTTP プロキシとプロキシポートです。

2. OO のホームディレクトリの **\Studio\tools** フォルダーで、**wswizard.exe** をダブルクリックします。
Web サービスウィザードが起動します。

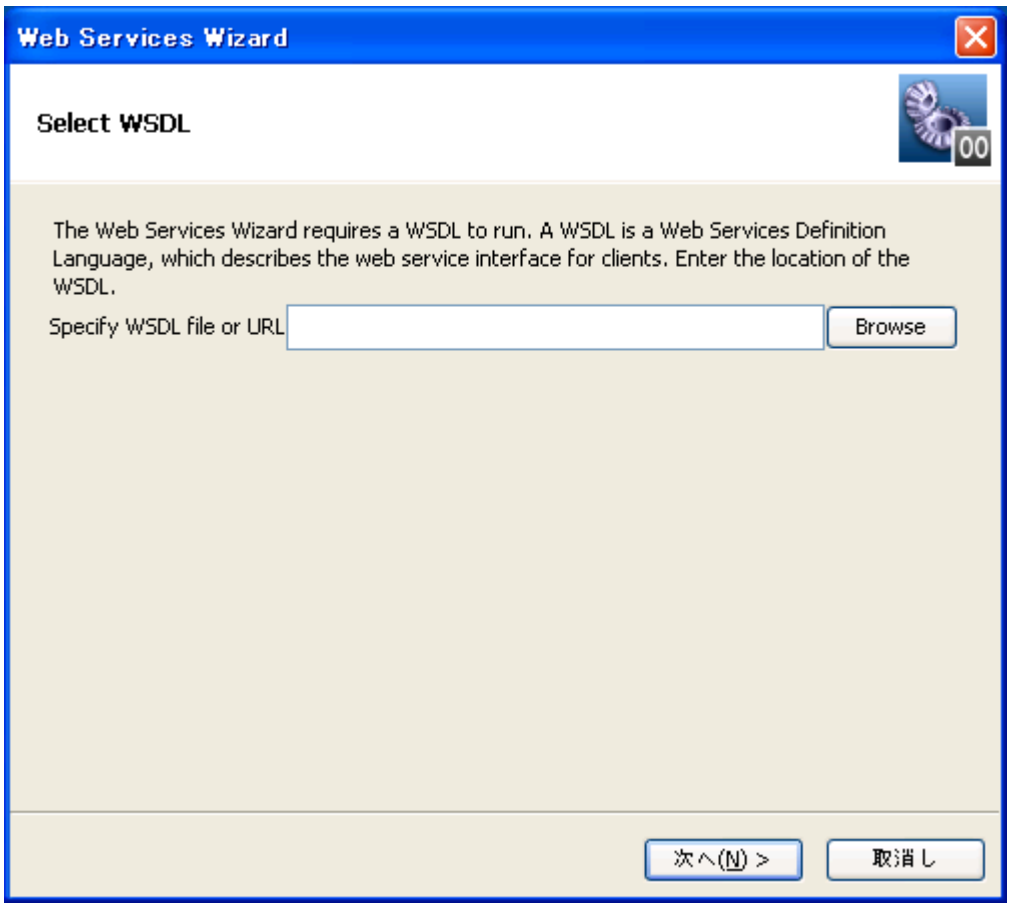


図 117 - Web サービスウィザード

3. **[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。
または
[Browse] をクリックし、WSDL の場所に移動します。
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。
4. **[次へ]** をクリックします。
WSDL にアクセスするため、Web サービスによってユーザー名とパスワードが要求されることがあります。
5. 資格情報を求められたら、入力します。
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。
注: Web サービスウィザードが応答を停止した場合は、Web 上の場所からハードディスクに保存された WSDL を示していることによる可能性があります。 WSDL が別の Web 上の場所の XSD ファイルを参照している場合、Web サービスウィザードはディスク上に置かれたファイルから参照を追従できない可能性があります。
[Choose Operations to generate] ペインが表示されます。

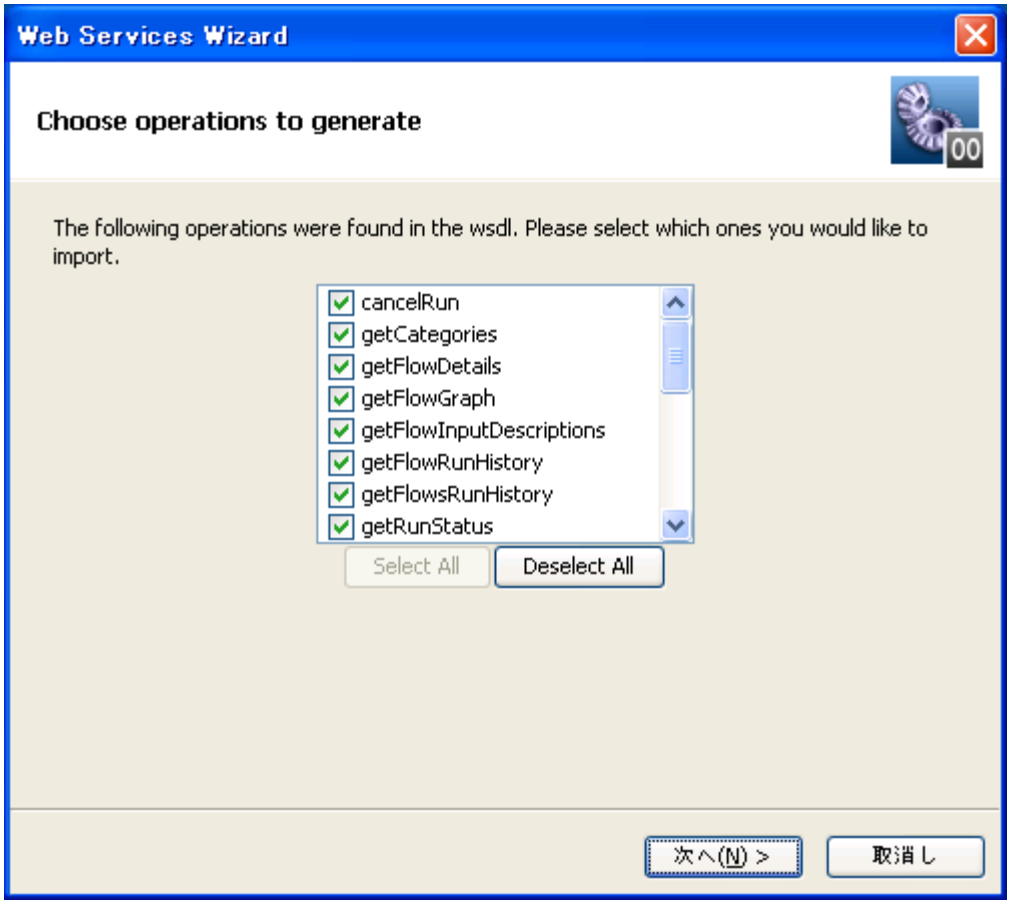


図 118 - 作成するオペレーションの選択

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。 デフォルトでは、すべてのメソッドが確認されます。

6. オペレーションに変換しないメソッドを選択解除し（選択から外す）、**[次へ]** をクリックします。

   [Select All] または **[Deselect All]** ボタンのいずれかを使用して、目的のメソッドをより効率的に選択できます。

   [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。

7. **[Enter the repository to open]** で、目的のパスとフォルダーを入力してから、**[次へ]** をクリックします。

   または

   [Browse] をクリックし、リポジトリを作成する場所に移動して、**[次へ]** をクリックします。

   ウィザードは指定した場所にオペレーションを作成します。

8. ウィザードが完了したら、**[完了]** をクリックします。
9. さらにオペレーションを作成するには、ウィザードを再実行します。
10. Studio を開き、作成したリポジトリをインポートします。
11. Web サービスウィザードが作成したオペレーションのレビュー時に、何らかの入力が必要とされる場合があります。

    このセクションの導入教材で説明した理由により、Web サービスウィザードは WSDL によって生成されたオペレーションに対するすべての入力を、省略可能として扱います。

    重要： 必要とされる入力の詳細については、この一連のオペレーションを作成した Web サービスのドキュメントを参照してください。

Web サービスウィザードのログ記録

Web サービスウィザードによってログファイル（WebServiceWizard.log）が生成され、OO ホームディレクトリの **\Studio\tools** サブディレクトリ内に保存されます。 このログには、情報、警告、エラーの各ログエントリを含む、Web サービスウィザードのすべてのアクティビティが記録されています。 ログ記録の構成は **\Studio\tools\conf\wsw.log4j.properties** ファイルで行うことができます。

リモートアクションサービスを使用した Central 外でのオペレーション

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 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 サービスウィザードを用意し、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 サービスウィザード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 のリポジトリフォルダーに追加します。 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 オペレータパス（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（**[入力]** 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 上で実行する必要のある、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\リモートコマンド実行** フォルダー内のオペレーションなど）。

注: 作成者またはフローユーザーが RAS を使用するオペレーションまたはフローを使用するには、RAS に対する読み取り権限と実行権限が必要です。 作成者が RAS 参照を変更するには、RAS に対する読み取り権限と書き込み権限が必要です。 詳細については、「*RAS へのアクセス制御*」を参照してください。

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
essRAS へのアクセス制御
既存の RAS から RAS 参照を追加するには
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

RAS に依存するオペレーションの Central での実行方法

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 [プロパティ] シートにある [RAS Operation フィールド] で指定します。 Library の [統合]、[Operations]、[ユーティリティ　オペレーション] 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」オペレーション] 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 **[プロパティ]** 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 外でのオペレーション*」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. **[リポジトリ]** で、**[Configuration\Remote Action Services]** フォルダーを右クリックして、**[プロパティ]** 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、**[可用性の確認]** をクリックします。

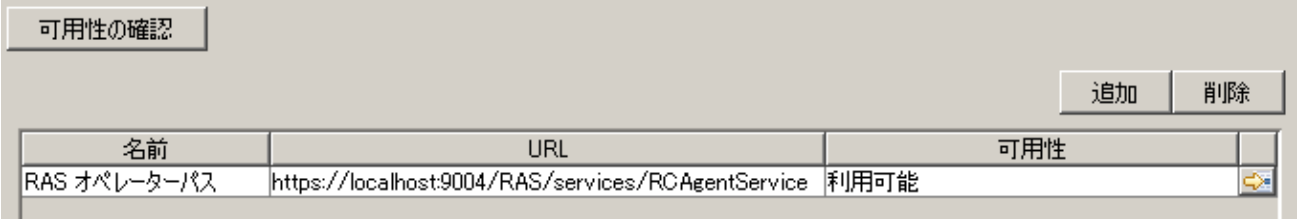


図 84 - [Remote Action Services] シート

RAS を使用できるかどうかが **[可用性]** 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。

RAS へのアクセス制御

ユーザーが RAS を使用するフローまたはオペレーションを使用するには、RAS 参照に対する実行権限が必要です。 ほかの OO オブジェクトと同様に、RAS 参照を変更するには、読み取り権限と書き込み権限が必要です。

OO オブジェクトの読み取り、書き込み、実行、その他の権限の詳細と設定については、*OO オブジェクトへのアクセス制限*」および「*フォルダーおよび OO オブジェクトへのアクセス制御*」を参照してください。 グループのデフォルトの権限の設定については、「*グループのデフォルトのアクセス権限の設定*」を参照してください。

既存の RAS から RAS 参照を追加するには

[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 [検出された 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 外でのオペレーション*」を参照してください。

既存の RAS から RAS 参照を追加するには

1. **[Remote Action Services]** シートを開くには、**[リポジトリ]** で、**[Configuration\Remote Action Services]** フォルダーを右クリックして、**[プロパティ]** をクリックします。
2. **[検出された RAS]** で、オペレーションで使用できるようにする RAS インストールを選択し、**[検出された RAS]** の下にある **[追加]** 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、RAS に名前を付けます。

   新しい RAS が、**[Remote Action Services]** シートの RAS 参照のリストに表示されます。
4. 新しく追加された RAS 参照の RAS の可用性を確認します。

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 外でのオペレーション*」を参照してください。

新しい RAS 参照を追加するには

1. **[Remote Action Services]** シートを開くには、**[リポジトリ]** で、**[Configuration\Remote Action Services]** フォルダーを右クリックして、**[プロパティ]** をクリックします。
2. **[Remote Action Services]** シートで、上部にある **[追加]** ボタンをクリックします。
3. 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。
4. **[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。
5. プロキシを選択するには、**[プロキシ設定]** で、**[HTTP プロキシ]** または **[Hewlett-Packard ゲートウェイ]** のいずれかを選択します。
   - **[HTTP プロキシ]** を選択した場合は、次のテキストボックスに入力します。

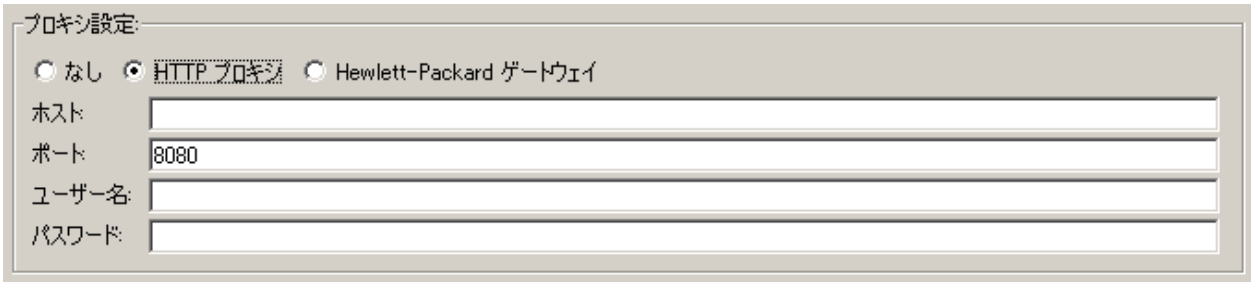


図 119 - HTTP プロキシの設定

   - **[ホスト]** 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。
   - **[ポート]** 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。
   - **[ユーザー名]** と **[パスワード]** 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 ゲートウェイ]** を選択した場合は、次のテキストボックスに入力します。

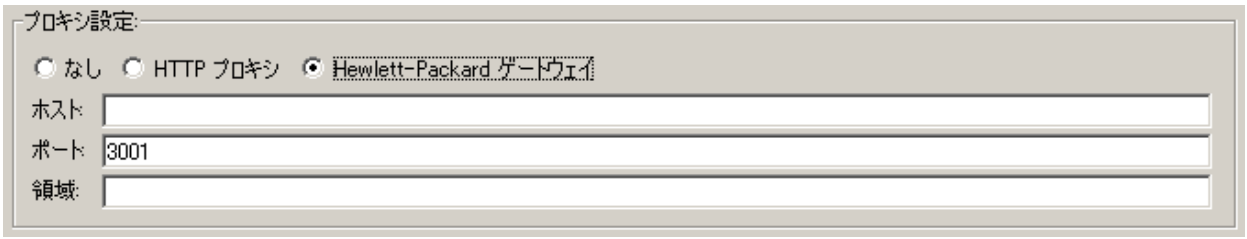


図 120 - HP ゲートウェイのプロキシ設定

  - **[ホスト]** 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。
  - **[ポート]** 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。
  - **[領域]** 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 ゲートウェイのレルムの名前を入力します。

6. 新しい RAS を保存し、**[Configuration]** の隣の **[Remote Action Services]** をクリックします(以下の図を参照)。

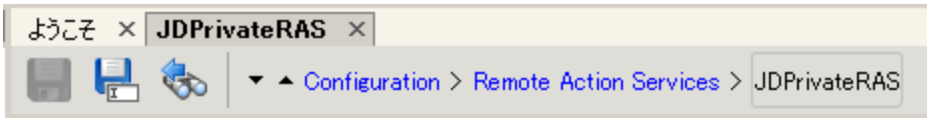


図 121 - [Remote Action Services] シートへのリンクの場所

7. **[Remote Action Services]** シートで、作成したばかりの RAS 参照を選択します。
   これによりサービスの URL を使用できるかどうかがチェックされ、**[可用性]** 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 外でのオペレーション*」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]** フォルダーを開き、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 **[開く]** 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。
   または
   [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、表のそのポインター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、**[プロキシ設定]** で、**[HTTP プロキシ]** または **[Hewlett-Packard ゲートウェイ]** のいずれかを選択します。

- **[HTTP プロキシ]** を選択した場合は、次のテキストボックスに入力します。

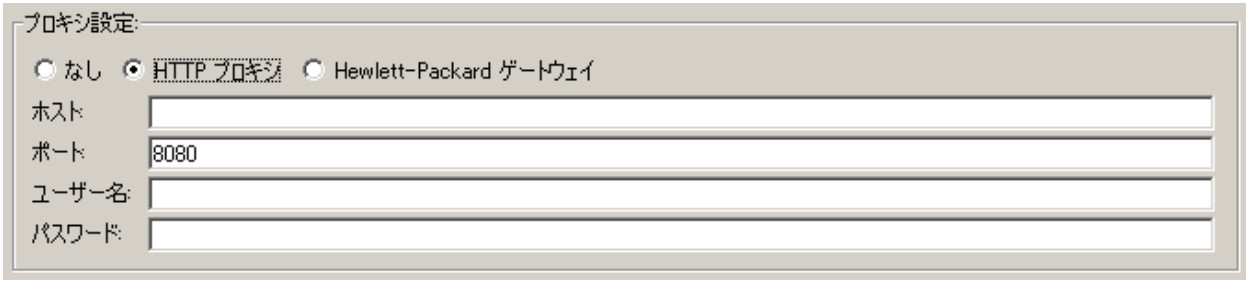


図 122 - HTTP プロキシの設定

  - **[ホスト]** 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。
  - **[ポート]** 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。
  - **[ユーザー名]** と **[パスワード]** 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 ゲートウェイ]** を選択した場合は、次のテキストボックスに入力します。

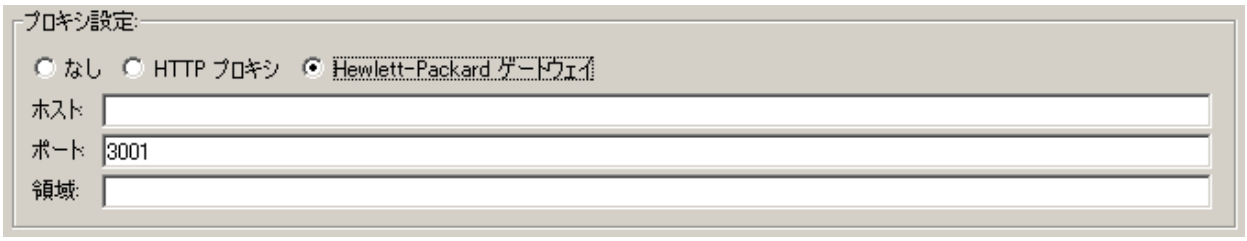
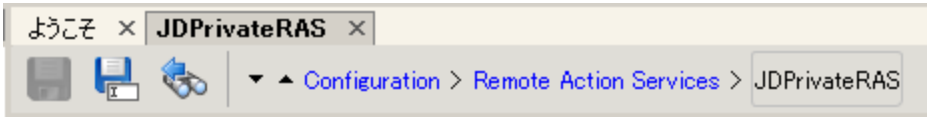


図 123 - HP ゲートウェイのプロキシ設定

  - **[ホスト]** 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。
  - **[ポート]** 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。
  - **[領域]** 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 ゲートウェイのレルムの名前を入力します。

4. RAS を保存し、**[Configuration]** の隣の **[リモートアクションサービス]** をクリックします。



5. **[Remote Action Services]** シートで、**[可用性の確認]** 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がチェックされ、**[可用性]** 列に使用可否が表示されます。

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 外でのオペレーション*」を参照してください。

RAS 参照を削除するには

- **[Remote Action Services]** シートで、削除する RAS 参照を選択し、**[削除]** をクリックします。

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 **[プロパティ]** 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。

図 124 - 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、**[RAS のオーバーライド]** フィールドに、**overrideRAS** 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、**[RAS のオーバーライド]** フィールドに、フロー変数参照（**${overrideRAS}** を含めることを強くお勧めします。

- RAS オペレーションの作成時に、RAS 参照を変更することができます。RAS 参照を変更するには、**[Configuration\Remote Action Services]** フォルダーから、オペレーションの **[プロパティ]** シートの **[入力]** タブの **[RAS]** フィールド内にある **[地球]** アイコン（）に新しい RAS 参照をドラッグします。 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。**${overrideRAS}** をシステムプロパティとして設定するということは、こ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ることを意味します。

    警告 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*」を参照してください。
  - 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。注: これによって、グローバル変数と同じ名前を持つフロー変数（範囲においてローカルである、つまり範囲に子フローが含まれず、フローの現在の実行に限定されるもの）の値が変更されます。 グローバル変数 overrideRAS の値は変更されません。

    したがって、RAS 上書きをステップ入力から割り当てられたフロー変数として設定すると、フロー実行のそのステップ以降の RAS 参照のみが変更されます。
  - 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 **[プロパティ]** シートの **[入力]** タブで、**[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 **[地球]**アイコン（）に、RAS 参照をドラッグします。

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、次のいずれかを作成します。
   - フロー入力。
   - 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 **[プロパティ]** シートの **[入力]** タブの、**[RAS のオーバーライド]** 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、**[RAS のオーバーライド]** 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. **[RAS のオーバーライド]** 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、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使用できるメソッドが定義されています。 Web サービスウィザードは、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 サービスウィザードで **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、『管理者ガイド』（AdminGuide.pdf）を参照してください。

Web サービスウィザード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）をポイントしていない。

  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 外でのオペレーション*」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： **表示** オペレーション（**[ユーティリティ　オペレーション]** 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. **[表示]** 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。
3. 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。
4. 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フローを設計し、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、マップの一覧表示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。
5. 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 参照によりポイントされる）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。

RAS 参照の構成の追加または変更の詳細は、次のいずれかを参照してください。

- *既存の RAS から RAS 参照を追加するには*
- *RAS 参照の追加*
- *既存の RAS 参照の再設定*

目的の RAS を指し示す RAS 参照を取得したら、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。

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 に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 外でのオペレーション*」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、RAS からオペレーションを作成するフォルダーを強調表示します。

   フォルダーの作成が必要な場合は作成します。
2. 開いているエディターまたは **[プロパティ]** シートをすべて閉じます。
3. **[ファイル]** メニューで、**[RAS からオペレーションを作成]** をポイントします。

   それらを閉じた後、**[RAS インポート]** ダイアログボックスが表示されます。
4. 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。

   [RAS からオペレーションを作成] ダイアログボックスが表示され、RAS から利用可能な .jar ファイルがリスト表示されます。

   オペレーションの作成元から、それぞれの .jar ファイルを展開して IActions を表示できます。
5. 利用可能な .jar / IActions のツリーで、オペレーションの作成元とするものを選択して、**[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、設計したフォルダーに表示されます。

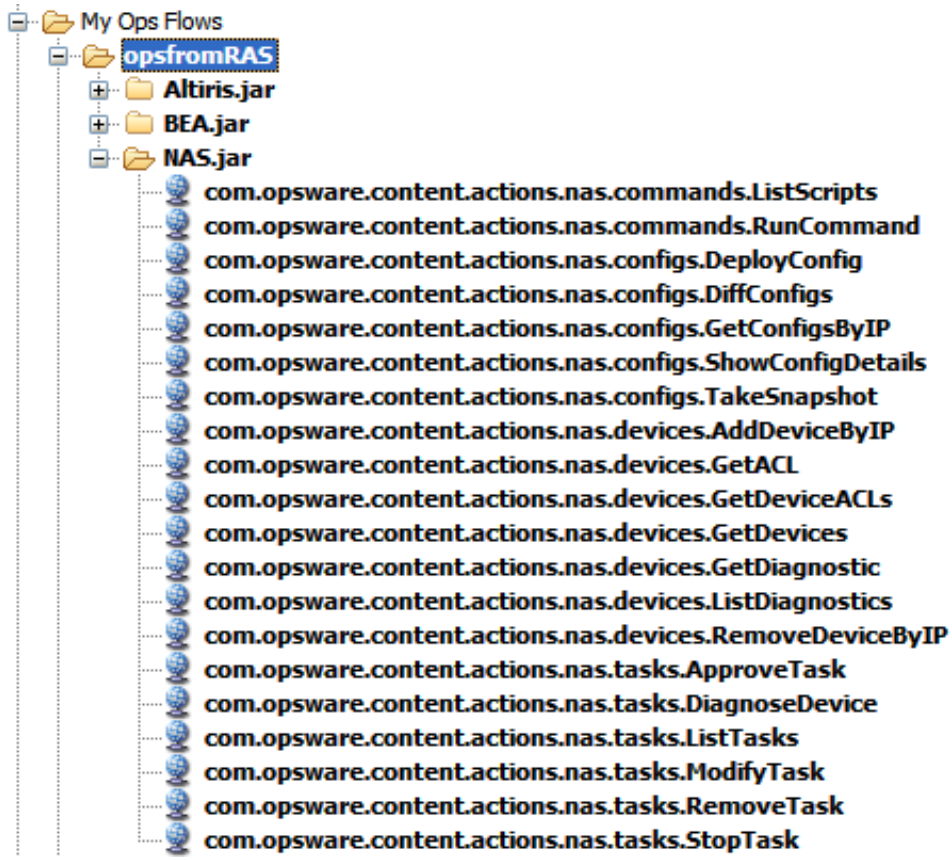


図 125 - RAS からの作成されたオペレーション

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 に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 外でのオペレーション*」を参照してください。

利用可能な 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。

   または

   目的のメソッドまたは関数を含む、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 に追加された .dll（独自またはサードパーティ製にかかわらず）を次の場所にコピーします。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 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 外でのオペレーション*」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、**[ファイル]** メニューから **[RAS からオペレーションを作成]** 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、アクセラレーターパック 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（${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。
- 前のステップの結果

Studio には、次の値を検証する標準のシステムエバリュエーターがリスト表示されています。

- 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 **電話番号** 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. **[リポジトリ]** の、**[Configuration]** フォルダーで、**[System Evaluators]** フォルダーを右クリックして、**[新規]** をクリックします。

   または

   [System Evaluators] フォルダーを右クリックして、**[プロパティ]** をクリックして、表示される **[System Evaluators]** シートで **[追加]** を右クリックします。
2. **[評価子の選択]** ダイアログボックスに表示された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. 表示された次のダイアログボックスに、エバリュエーターの名前を入力し、**[OK]** をクリックします。
   - **[新規]** 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。
   - **[プロパティ]** 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. エバリュエーターをテストするために独自のデータを提供するには
   a. **[フィルター入力のテスト]** ボックスの内容をクリアするには、**[クリア]** をクリックします。
   b. 独自のデータに貼り付けます。

      または

検証するテキストの例を **[フィルター入力のテスト]** ボックスに入力します。

または

クイックコマンドで：

- **[クイックコマンド]** をクリックします。
- 表示された **[入力]** 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ます。
- **[OK]** をクリックします。

指定したパターンにテスト入力が一致または一致する部分が含まれている場合、**[フィルター入力のテスト]** のレポートは次のようになります。

図 126 - パターンの一致レポート

指定したパターンにテスト入力が一致しない場合、**[フィルター入力のテスト]** のレポートは次のようになります。

図 127 - 一致なしのレポート

6. 作業結果を保存します。

エバリュエーターは、**[Configuration\System Filters]** フォルダーに保存されます。

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. リポジトリの、**[Configuration\System Evaluators]** フォルダーを開きます。
2. 変更するエバリュエーターを右クリックし、**[開く]** 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。

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 **[削除]** をクリックします。

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、**成功** 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相関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相関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、**Windows Server の再起動** フローが実行されたサーバーとその操作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 マネージャーが構成アイテムごとのアラート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。 ダッシュボードに構成アイテムごとのフロー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 **Windows Server の再起動** 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**操作**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 [操作によって整理されたすべての CI] 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

1. フローの **[プロパティ]** シートまたはステップのインスペクターで、記録する入力用に入力エディターを開きます。
2. **[入力のプロパティ]** で、**[次の名前で記録]** 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、構成アイテムごとのフロー(過去 7 日間)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 [ダッシュボード] タブではグラフが表示されます。ドメインタームおよびドメインタームに関連する入力値のさまざまな組み合わせを分析することで、IT インフラの次のような傾向を顕在化することができます。

- 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、「*ドメインタームの追加、名前の変更、または削除*」および「*ドメインターム値の追加、変更、および削除*」を参照してください）。

- **操作**

  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（Category）**

  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 のマイナー種類**

  これらのドメインタームは CI タイプの区分に入る場合があります（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 の種類**

  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。

ドメインタームの追加、名前の変更、または削除

ドメインタームの一覧は、**Configuration\Domain Term** フォルダーにある 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]** フォルダーを展開します。
2. **[Domain Terms]** フォルダーを右クリックして、**[新規]** をクリックします。
3. 表示された名前を入力するダイアログボックスで、**[名前]** 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、**[OK]** をクリックします。

   新しいドメインタームのエディターが表示されます。

4. **[説明]** テキストボックスには説明文を入力します。
5. 作業を保存し、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。

ドメインタームの名前を変更するには

1. **リポジトリ**ペインで、**[Configuration\Domain Terms]** フォルダーを開き、名前を変更するドメインタームを右クリックします。
2. **[名前の変更]** をクリックし、新しい名前を入力して ENTER を押します。

ドメインタームを削除するには

1. **[リポジトリ]** ペインで、**[Configuration\Domain Terms]** フォルダーを開き、削除するドメインタームを右クリックします。
2. **[削除]** をクリックします。

ドメインターム値の追加、変更、および削除

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\Domain Terms]** フォルダーを開き、ドメインタームをダブルクリックして、そのエディターで開きます。
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、**[チェックアウト]** アイコン（ ）をクリックします。
   詳細については、「*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*」を参照してください。
3. ドメインタームのエディターで、**[追加]** をクリックし、**[名前]** と **[説明]** の列に値と説明文をそれぞれ入力します。
4. 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 **[削除]** をクリックします。

ドメインターム値を変更するには

- ドメインタームのエディターを開き、ドメインターム値のリストの **[名前]** 列で、変更する値をダブルクリックして新しい値を入力します。

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. ステップのインスペクターの **[表示]** タブで、**[このステップの実行前に常にユーザーに確認する]** を選択します。

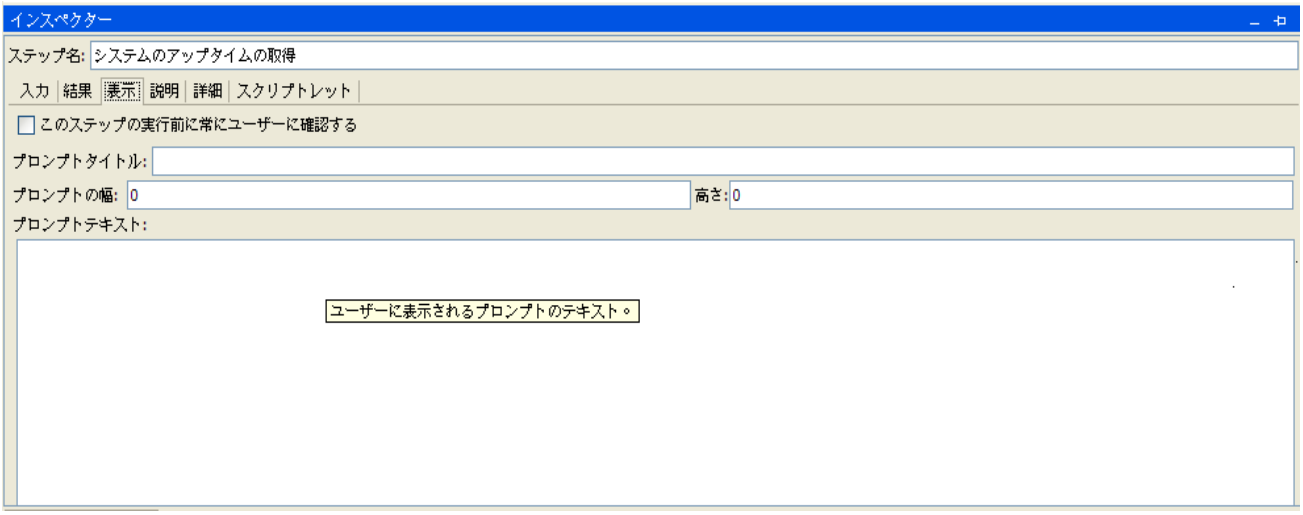


図 90 - ユーザーへの表示の指定

2. メッセージにタイトルをつけるには、**[プロンプトタイトル]** 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. **[プロンプトテキスト]** 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使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 [プロパティ] シートを開くには、フロー図の下部にある [プロパティ] タブをクリックします。
2. [カテゴリの割り当て] 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。

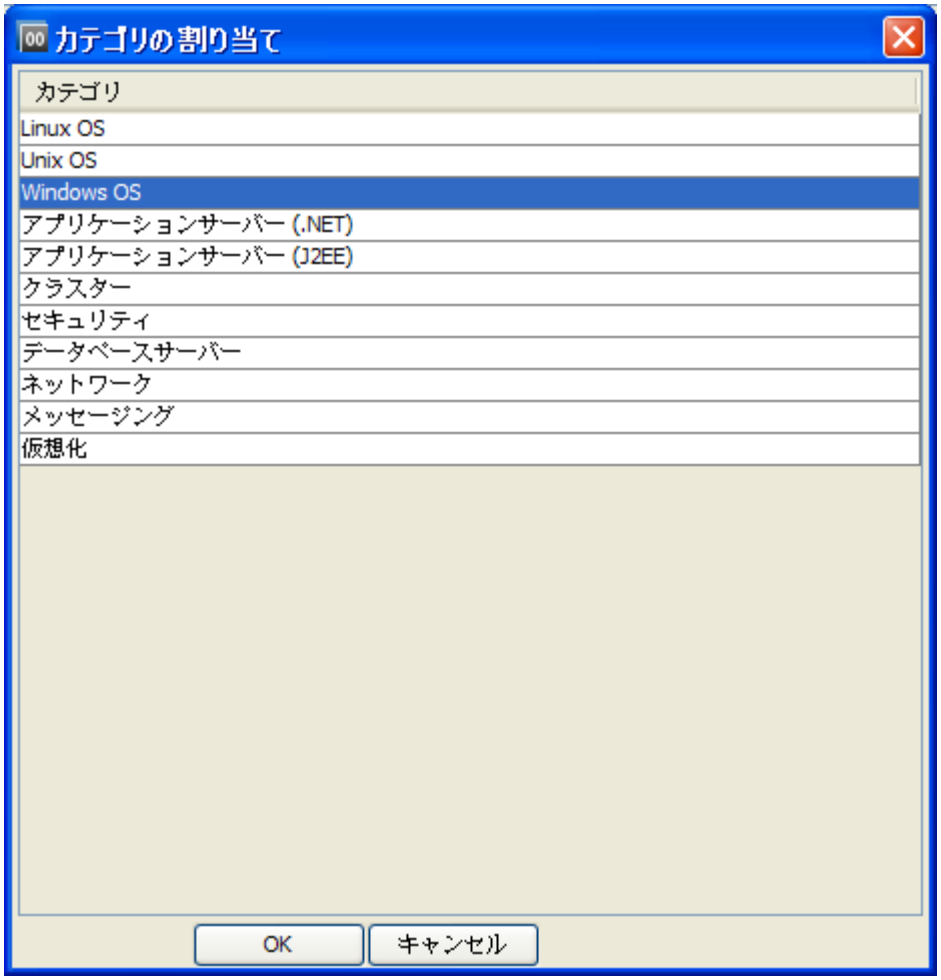


図 128 - カテゴリのフローへの割り当て

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 **[サービスステータス]** 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 フォルダー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]** フォルダーを右クリックし、**[新規]** 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 129 - 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、**[追加]** 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、**[削除]** 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]** フォルダーを右クリックし、**[プロパティ]** をクリックします。

   選択リストの一覧が表示されます。
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、**[追加]** 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、**[削除]** 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、**[Configuration\Selection Lists]** フォルダーを展開します。
2. 削除する選択リストを右クリックし、**[削除]** をクリックします。
3. 表示されたメッセージボックスで、**[OK]** をクリックします。

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 フォルダーに格納されます。

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]** フォルダーを右クリックし、**[新規]** をクリックします。
2. 表示された名前の入力ダイアログボックスで、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 130 - 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. **[ユーザー名]** ダイアログボックスに、次の構文を使用して、このシステムアカウントを表すアカウントのユーザー名を入力します。
   `<domain>\<username>`
5. **[パスワードの割り当て]** をクリックします。
   [パスワードの入力] ダイアログボックスが表示されます。
6. [パスワード] ボックスにパスワードを入力し、[パスワードの確認] ボックスにもう一度入力します。
7. **[保存]** 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]** フォルダーを展開し、編集するシステムアカウントを右クリックして **[開く]** 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

- **[リポジトリ]** ペインで、**[Configuration\System Accounts]** フォルダーを展開し、削除するシステムアカウントを右クリックして **[削除]** をクリックします。

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。

グループのメンバーシップ管理およびケイパビリティ管理の手順は、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 のどちらででもフローの実行を開始できます。 この権限は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	操作	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

オブジェクト	操作	必要な権限
	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
RAS 参照		
	表示	読み取り
	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での使用	読み取り、リンク
	実行時に使用	読み取り、実行
システムアカウント		
	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 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
- 選択リスト
- RAS 参照とシステムアカウント（Configuration\RAS フォルダー）

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、また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 RAS 参照など）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 OO オブジェクトへ権限を設定するには

1. **[リポジトリ]** ペインで、フォルダー、フロー、オペレーション、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 **[アクセス許可]** 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。

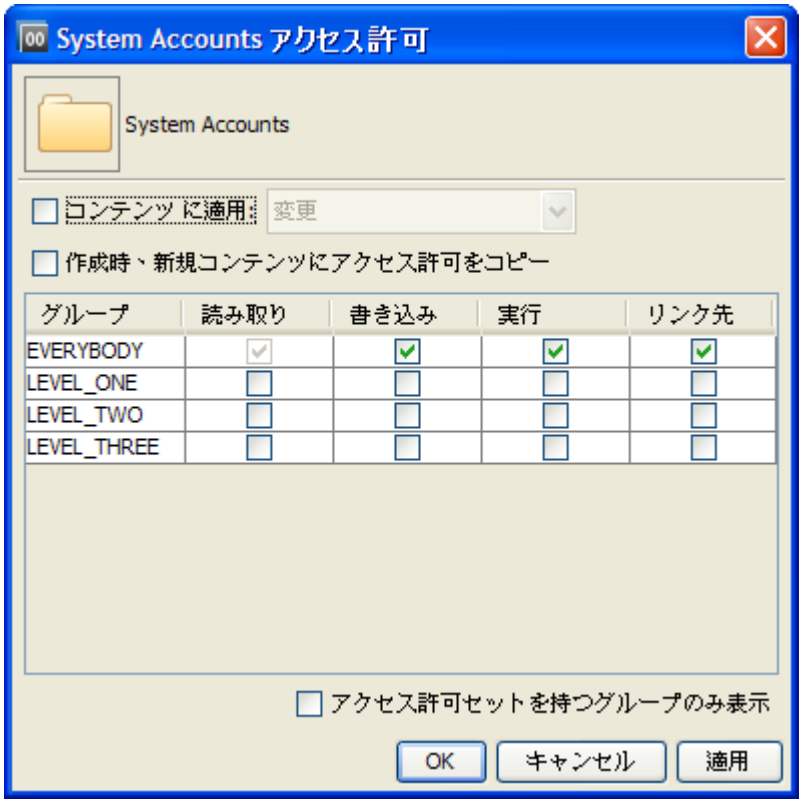


図 131 - 権限の設定

ヒント：**[アクセス許可セットを持つグループのみ表示]** 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。

   [適用] ボックスには、[コンテンツに適用] または [参照されたオブジェクトに適用] 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、[コンテンツに適用] 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、[参照されたオブジェクトに適用] となります。

     このダイアログボックスに表示される **[参照されたオブジェクト]** 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、**[コンテンツに適用]** を選択します。

   または

   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にするには、**[参照されたオブジェクトに適用]** 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）、また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、**[...に適用]** ボックス横のドロップダウンリストから **[変更]** を選択します。

または

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、**[...に適用]** ボックス横のドロップダウンリストから **[すべて]** 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、**[作成時、新規コンテンツにアクセス許可をコピー]** 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す）。

   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す）。

デフォルトのグループアクセス権限を設定するには

1. **[ツール]** メニューから **[グループマスクの設定]** 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。

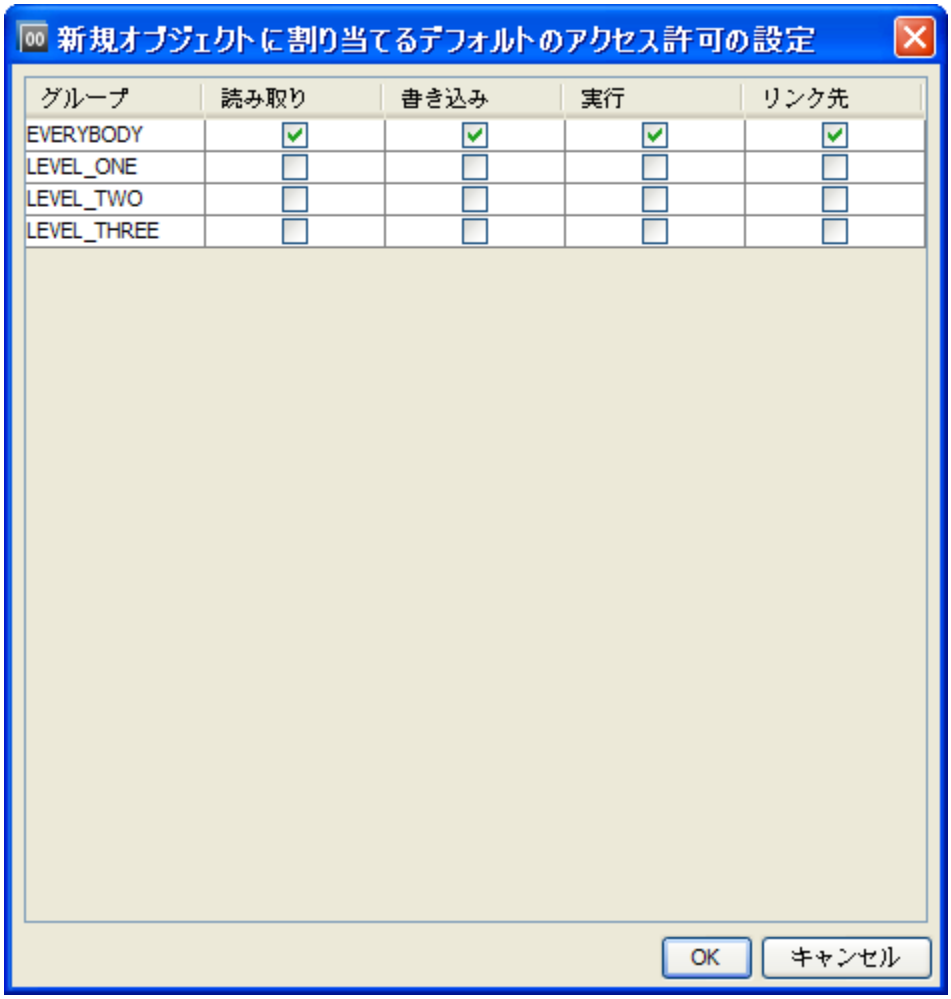


図 132 - 新規オブジェクトのデフォルト権限の設定

新しいオペレーションを作成するには

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

次のステップでは、オペレーションの作成方法を要約しています。

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、**[新規]**、**[オペレーション]** 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 **[プロパティ]** 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、**[カテゴリの割り当て]** をクリックし、リストからカテゴリを選択します。
4. 入力を作成するには、**[入力]** タブをクリックし、**[入力の追加]** をクリックします。
5. 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。
   - 入力の定義および使用方法については、「*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。
   - 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。

   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。 たとえば、データを入力に指定することに加えて、それをフロー編集に保存することも必要です。
6. 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。 出力データで行う重要なことの一つは、フローで後から使用する時のためにフロー変数に保存することです。
7. 必要なレスポンスをすべて作成し、結果をレスポンスにマッピングします。

   オペレーションにどのレスポンスが選択されるのかを定めるルールの定義については、「*レスポンス: 結果の評価*」を参照してください。
8. すべてのタイプのオペレーションで一般的な、次のプロパティを指定します。

   次のセクションで説明するように、作成したオペレーションのタイプに固有なプロパティをすべて指定します(*オペレーションのタイプ: プロパティの設定*):
   - *Cmd (コマンドライン)オペレーション*
   - *フローの実行カウンターオペレーション*
   - *フローの実行概要レポートオペレーション*
   - *HTTP オペレーション*
   - *Perl スクリプトオペレーション*
   - *RAS オペレーション: RAS のための IAction プログラミング*
   - *スクリプトレットオペレーション*
   - *セキュアな shell (ssh)オペレーション*
   - *Telnet オペレーション*
9. 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*
10. Central ユーザー対象のオペレーションを記録するには、**[説明]** 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。

    あなたのオペレーションを使用するかもしれない他の作成者のために、**[説明]** タブに含める必要がある情報については、作成するオペレーションに表示される **[説明]** タブのデフォルトの状態を参照してください。

    それぞれの入力について、次のことを含めることは他の作成者にとって非常に役立ちます。
    - 入力が受け取るデータの必要な文法(必要とされる場合)は次のとおりです。
    - サンプルデータ。
    - 実行可能な状況を制限する、オペレーションのすべての環境制限。
11. **[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**[リポジトリ]** 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

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。

次のスクリーンショットでは、オペレーションの **[プロパティ]** シートにオペレーションを一意に特定する識別子（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 からのオペレーション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には制限があります。

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。

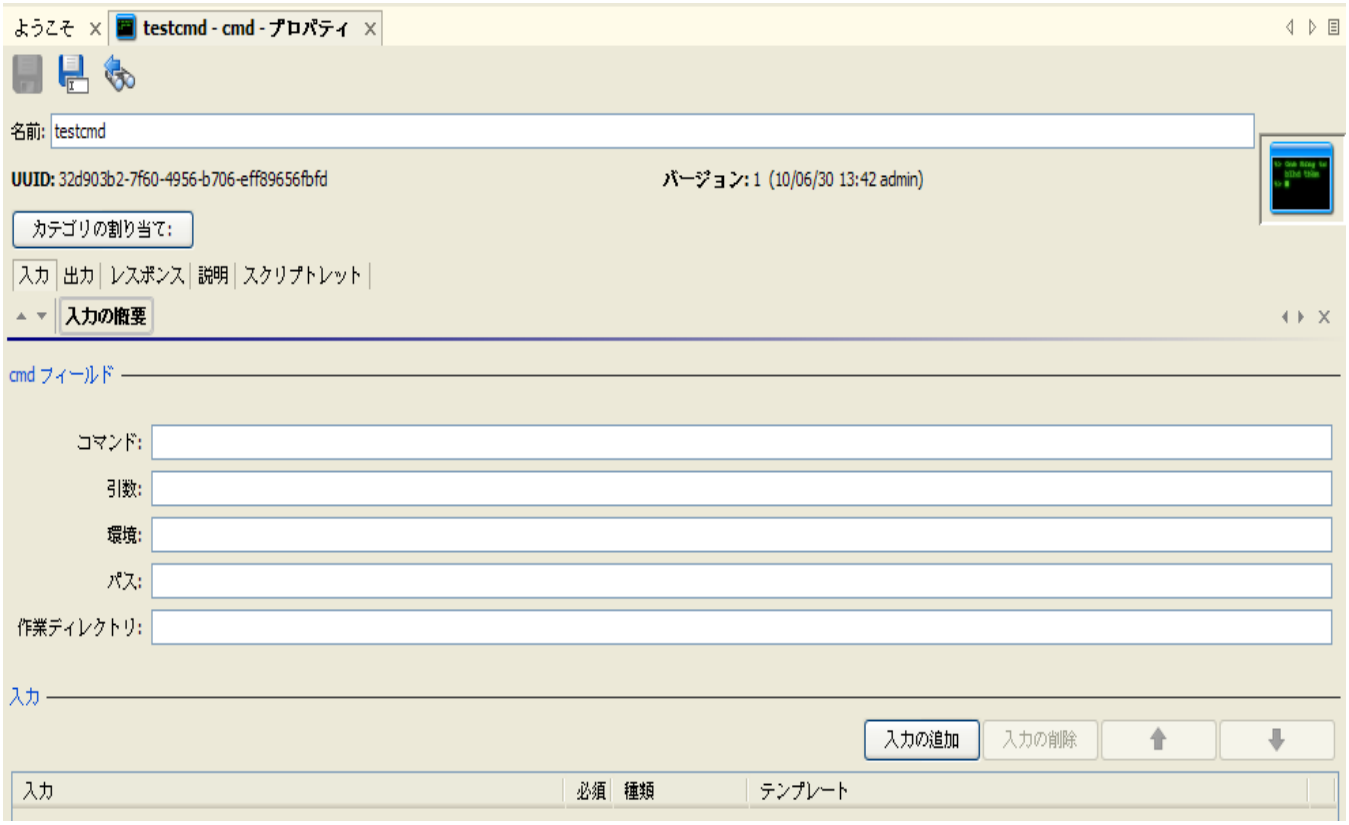


図 133 - Cmd オペレーションのプロパティ

入力に必要な値については、オペレーションの **[説明]** タブを参照してください。

[レスポンス] 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。詳細については、「*レスポンス: 結果の評価*)で、**[フィールドにルールを適用]** 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 **[説明]** タブを参照してください。**[結果フィールド]** の下部に詳細が表示されます。

- [Code]

- [Output String]
- [Error String]
- [FailureMessage]
- [TimedOut]

フローの実行カウンターオペレーション

使用回数は、オペレーションに対する次の入力のうち 1 つ以上を使用して指定した、1 つ以上のフローの完了した実行数を数えます。

- **時間ウィンドウ**

  現在と比較した時間(分)で指定する、特定の時間ウィンドウの内部

- **入力の制約**

  inputs name=value の組で指定します。 複数の input name=value の組を、セミコロン (;) でそれぞれ区切ることで指定できます。

  入力の制約 入力の文法によって、指定にある程度の柔軟性ができます。 オペレーションのプロパティシート上の **[説明]** タブには、文法、例、および許容できる文法の含意の議論が表示されます。

- **フロー入力のみ**

  これを **[True]** に設定すると、**入力の制約** 入力で定義した指定はフロー入力にのみ適用されます。 それ以外の場合は、すべてのステップ入力に適用されます。

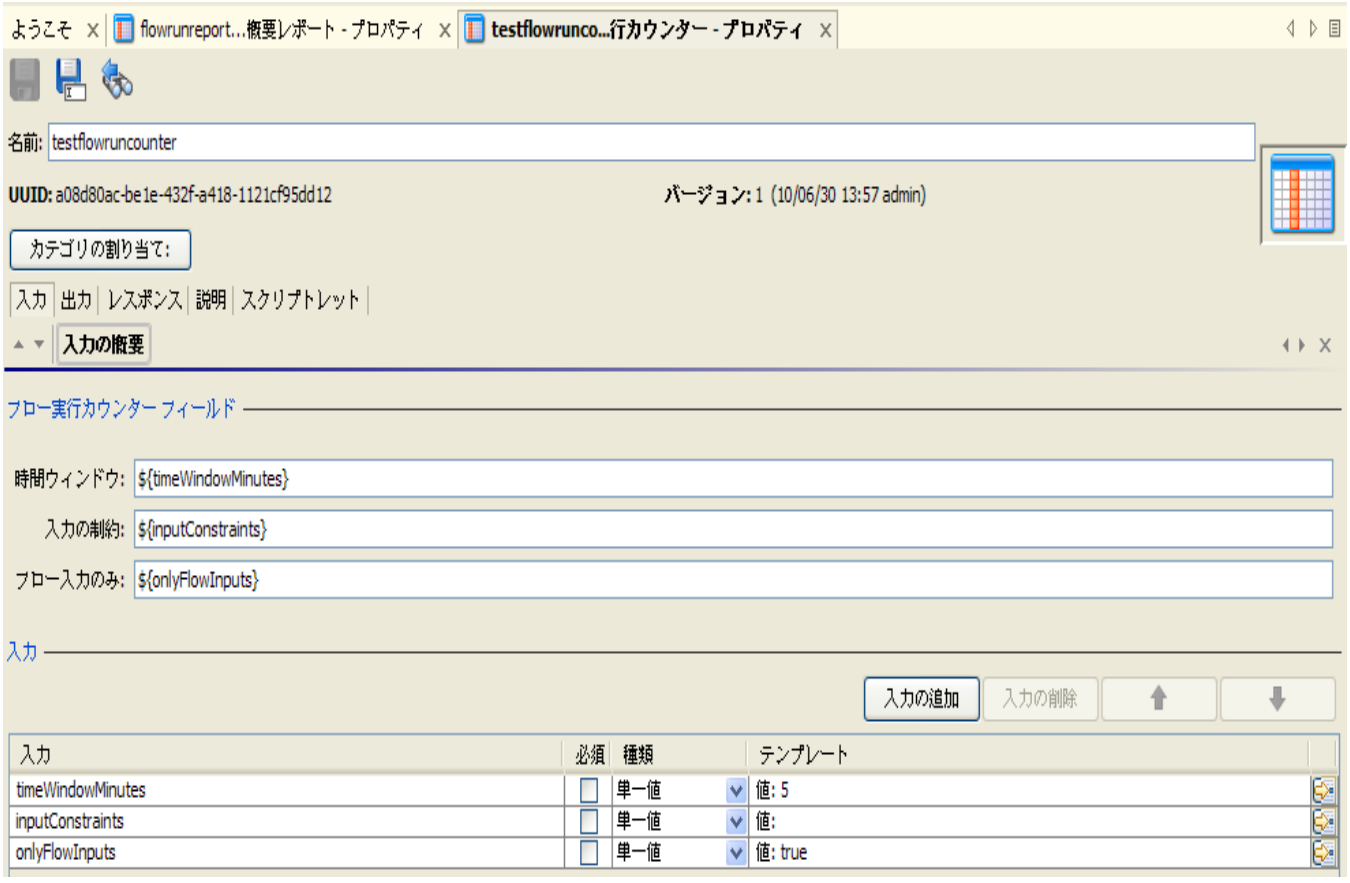


図 134 - フロー実行カウンターオペレーションのプロパティシート

オペレーションの主結果は、オペレーション入力で指定する制約に対応して開始された実行の回数です。

フローの実行概要レポートオペレーション

[使用回数] はフロー実行で実行されたステップの出力を要約するレポートを作成します。 これを使用して、異なるフロー実行のステップのレポートを作るか、またはオペレーションのステップを含むフローの実行で完了したステップのレポートを作ることができます。

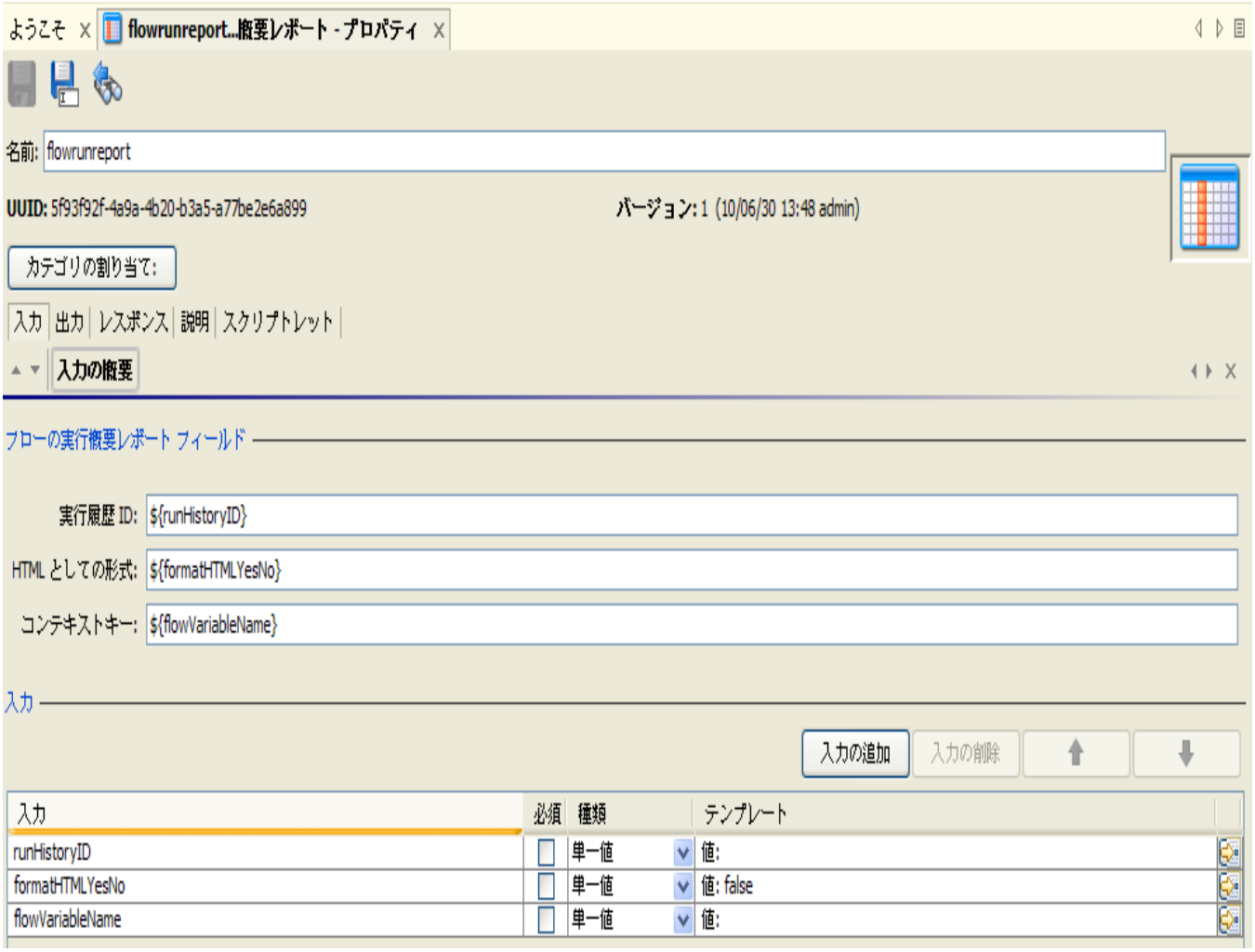


図 135 - フロー実行サマリレポートオペレーションのプロパティ

スクリプトレット内部からステップの配列にアクセス可能な JavaScript コードといった、オペレーションタイプの詳細については、オペレーションの **[説明]** タブを参照してください。

以下はフロー実行概要レポートオペレーションへの入力です。次の値が必要です。

- **runHistoryID**
  OO データベースの実行用の履歴 ID です。 空白の場合は、現在の実行履歴 ID が使用されます。 実行履歴 ID の検索の詳細については、Central のヘルプを参照してください。
- **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 つのインデックスされた列には次のものが格納されます。
    - Index [0]

フローのステップへのパス

- Index [1]
  ステップ名
- Index [2]
  レスポンス
- Index [3]
  説明

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。 これらのようなタスクでは、標準 HTTP オペレーションはおそらく IAction プログラミングと同等に効率的です。

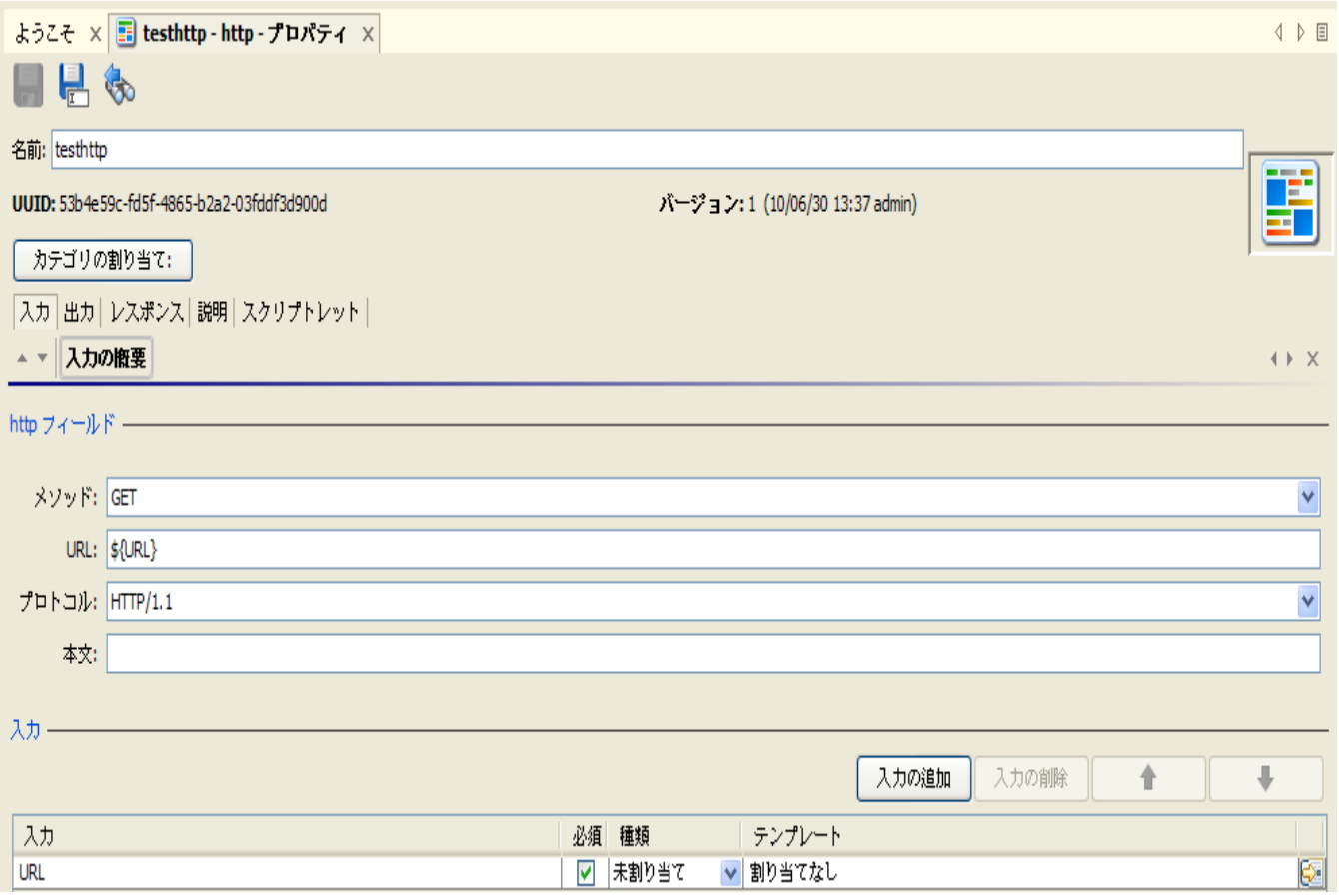


図 136 - HTTP オペレーションのプロパティ

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。

[メソッド] - 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。

[プロトコル] - 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。

[本文] - 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。

[レスポンス] 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、[フィールドにルールを適用] 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。

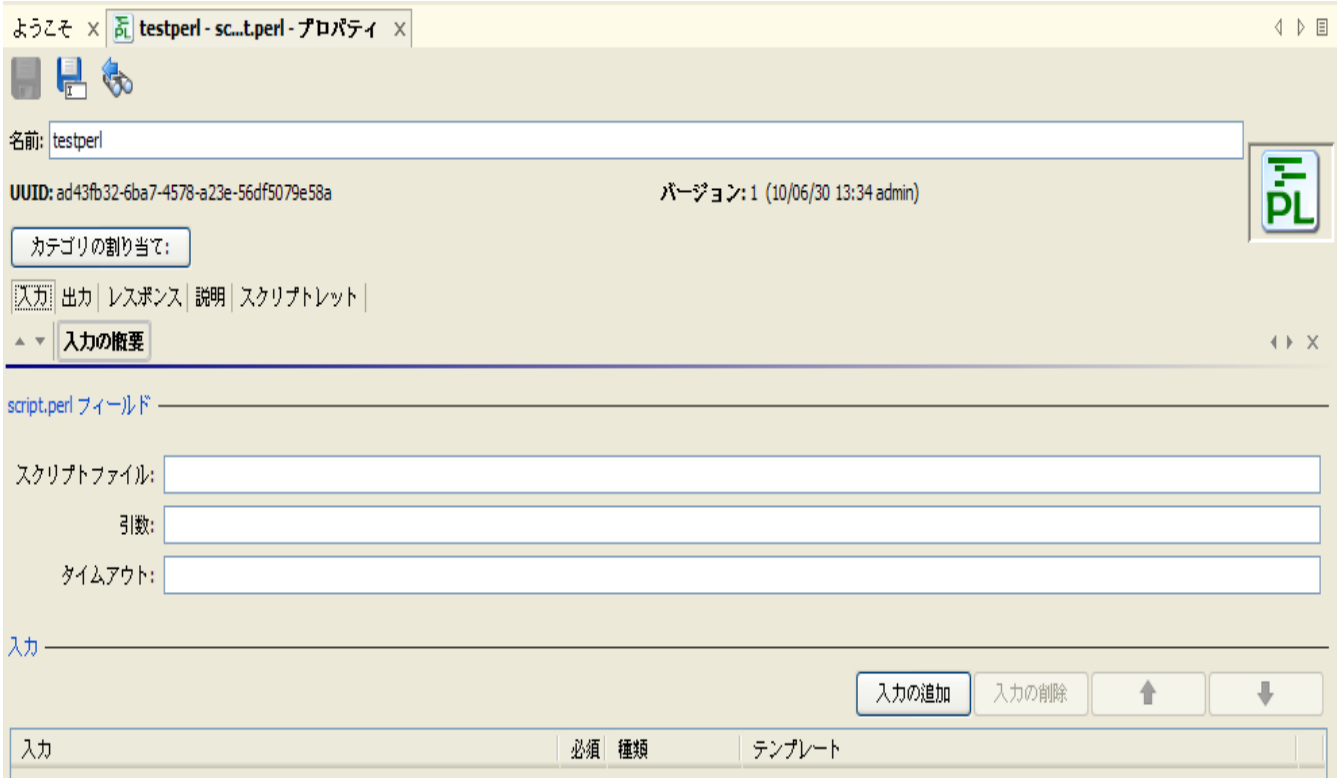


図 137 - Perl スクリプトオペレーションのプロパティ

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。

[スクリプトファイル] - 実行する Perl スクリプトの名前

[引数] - スクリプトに渡す引数

[タイムアウト] - スクリプトの最大実行時間

[レスポンス] 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、**[フィールドにルールを適用]** 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。

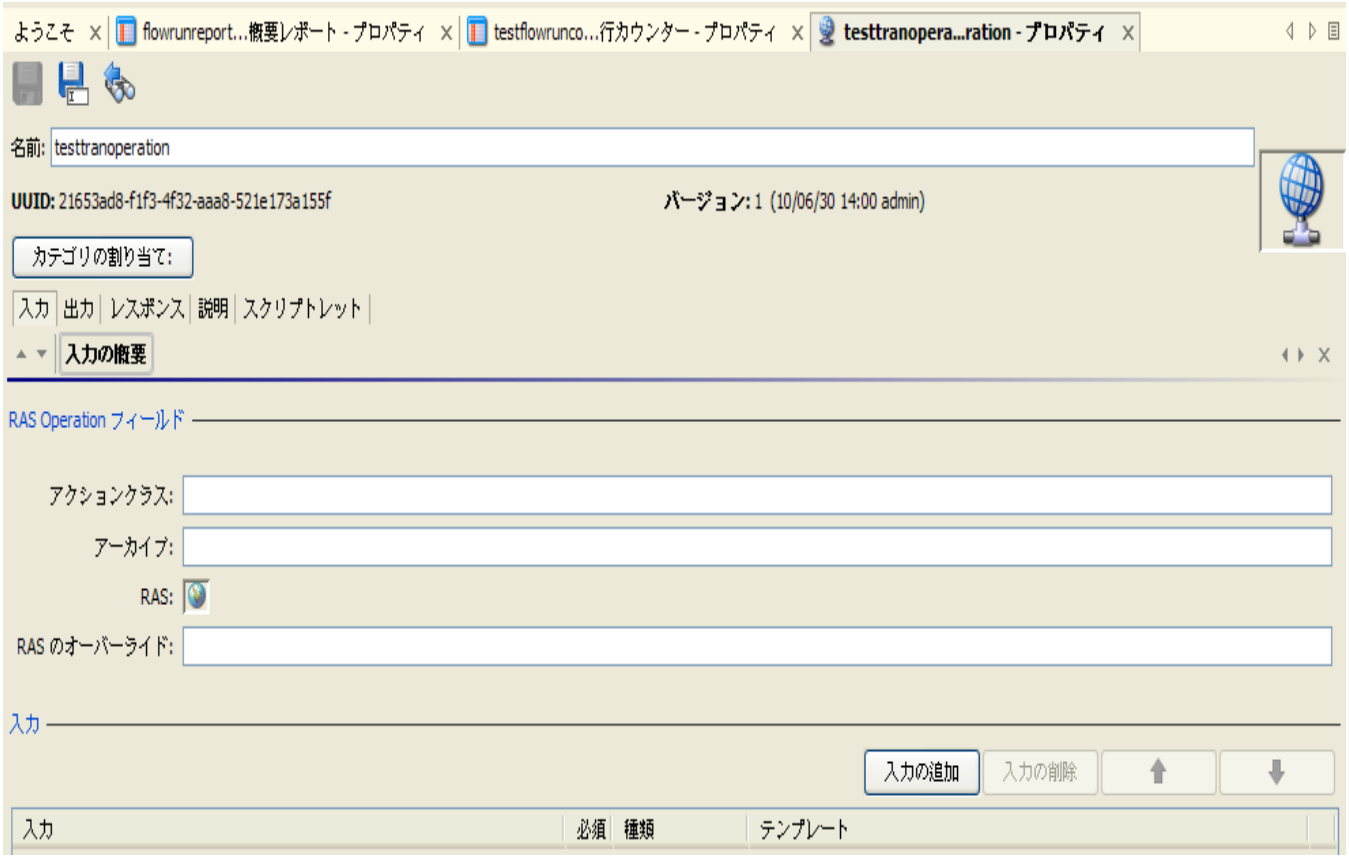


図 138 - RAS オペレーションのプロパティ

次に RAS オペレーションのフィールドについて説明します。次の値が必要です。

[アクションクラス] - 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。

[アーカイブ] - 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]** フォルダーから **[RAS]** フィールドのラベルの隣にある **地球**アイコン（）にドラッグします。

[RAS のオーバーライド] - 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。

[レスポンス] 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、**[フィールドにルールを適用]** 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 [説明] タブを参照してください。[追加の結果] 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

- **[Result]**
  オペレーションの主結果。

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 外でのオペレーション*」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。

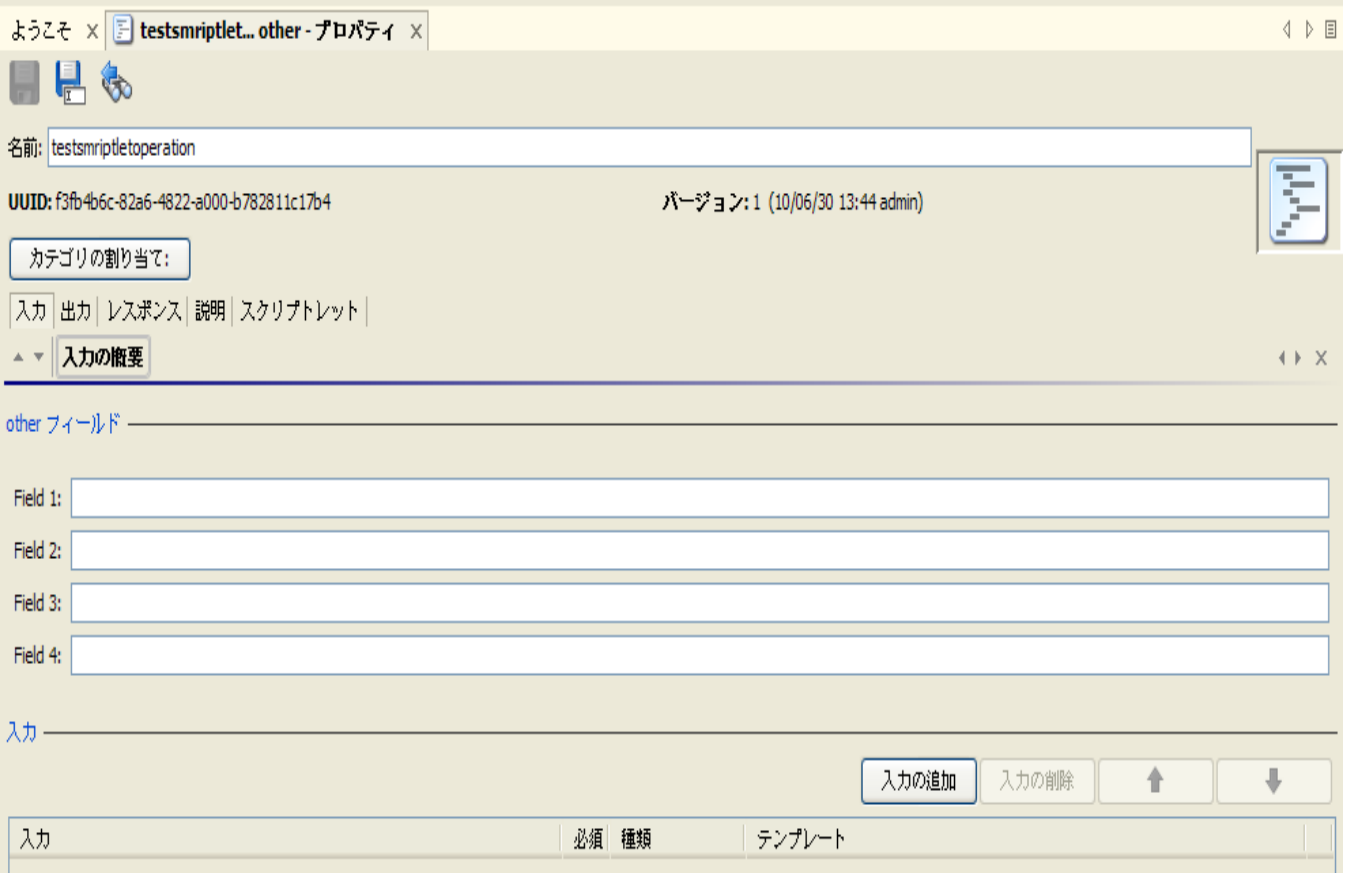


図 139 - スクリプトレットオペレーションのプロパティ

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。

[レスポンス] 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、[フィールドにルールを適用] 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

セキュアなシェル（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、**サービスの開始** フローを使用できます（フローは Library の、**アクセラレーターパック\オペレーティングシステム\Red Hat\状態の変更** にあります）。 **サービスの開始** フローでは、SSH オペレーションである **サービスの実行** オペレーション（Library の、**Operations\オペレーティングシステム\Linux\SUSE Linux\プロセスオペレーション** にあります）を使用します。 この用途での **サービスの実行** 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。

図 140 - セキュアな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 **[説明]** タブを参照してください。

注:

- **ホスト** 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>

- **[疑似端末]** 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。

[レスポンス] 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、**[フィールドにルールを適用]** 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 **[説明]** タブを参照してください。**[追加の結果]** 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。

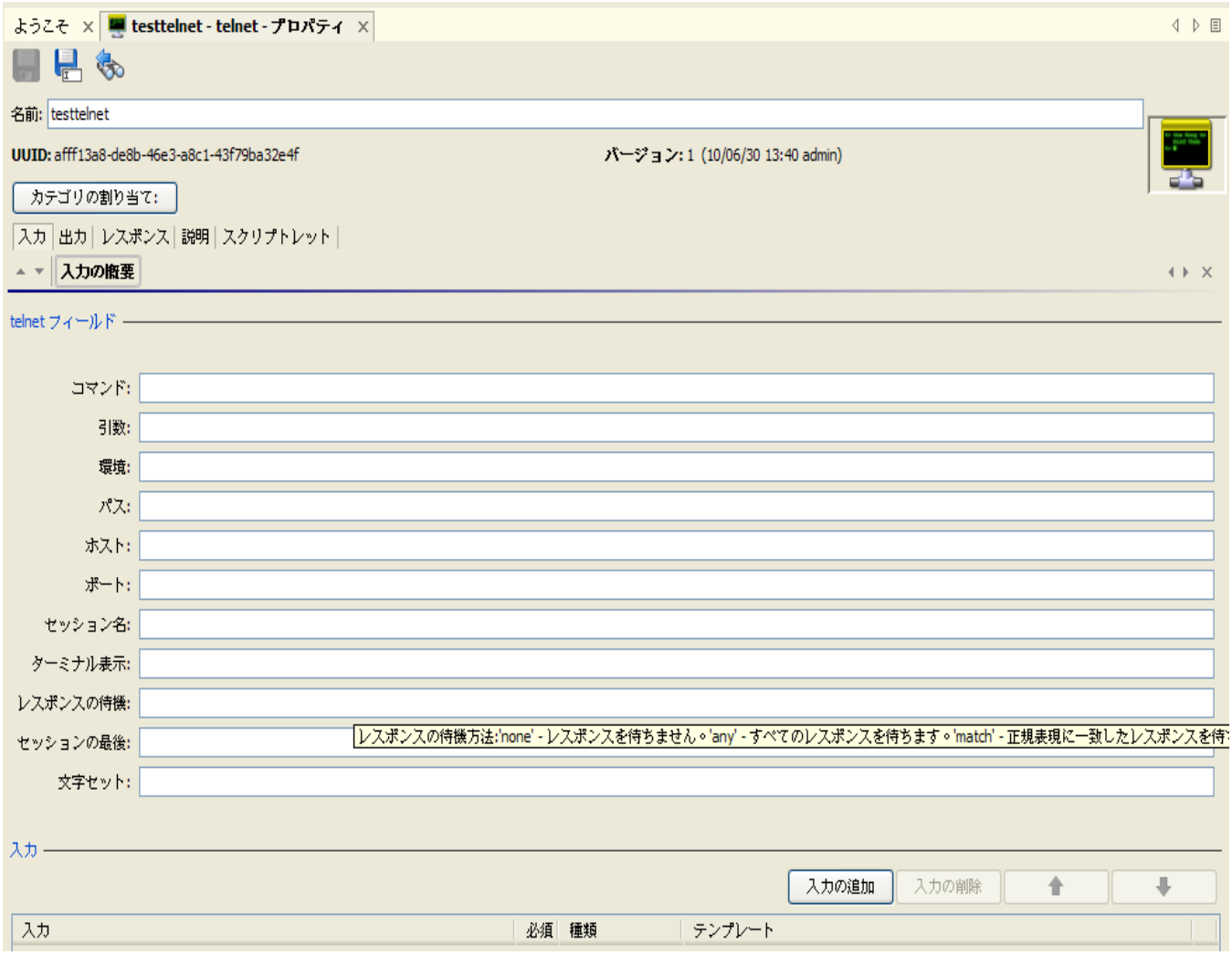


図 141 - Telnet オペレーションのプロパティ

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。

[コマンド] - オペレーションが実行するコマンド

[引数] - コマンドライン引数（スイッチ）

[環境] - 既存の環境変数への追加

[パス] - 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>

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ることに注意してください。

[ホスト] - コマンドのターゲットコンピューター

[ポート] - 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

[セッション名] - コンテキストに格納するセッションの名前

[ターミナル表示] - 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。

[レスポンスの待機] – タイムアウト時間。

[セッションの最終] – TK

[文字セット] – TK

[レスポンス] 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、**[フィールドにルールを適用]** 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交信できません。

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\]** フォルダーで、**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。

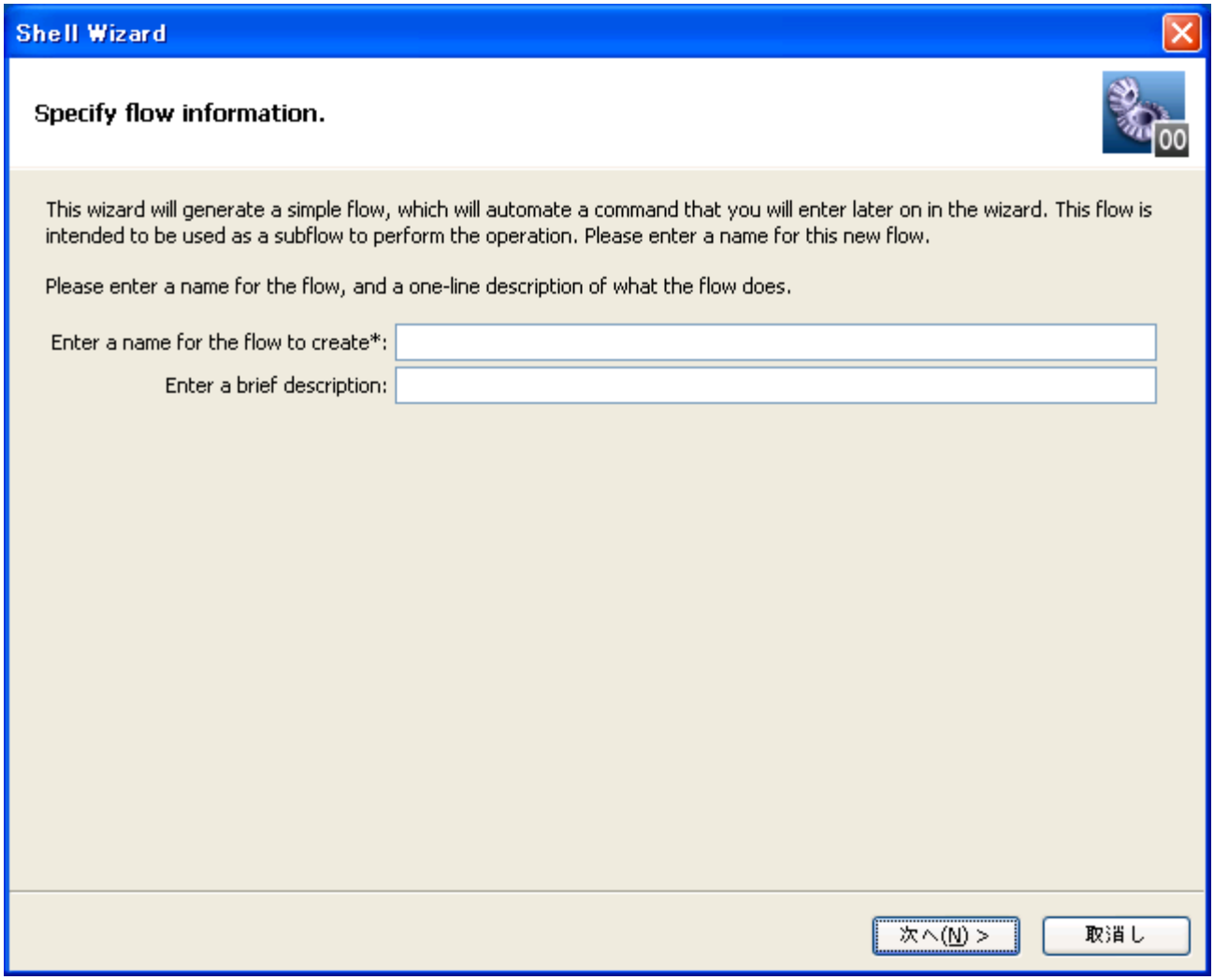
   または

   [Browse] をクリックし、リポジトリの場所に移動して、**[開く]** をクリックします。

   重要：

   - 空または存在しないフォルダーにポイントした場合、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ーを作成し、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。
   - 既存のリポジトリにポイントする場合は、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 のサブフォルダー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ー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 を空のフォルダーにポイントするか（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にポイントすることをお勧めします。
3. ターゲットリポジトリを作成済みの場合は、**[次へ]** をクリックします。

   [Specify flow information] ページが表示されます。



4. **[Enter a name for the flow to create]** 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、**[次へ]** 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。
   [Please specify connection settings] ページが表示されます。

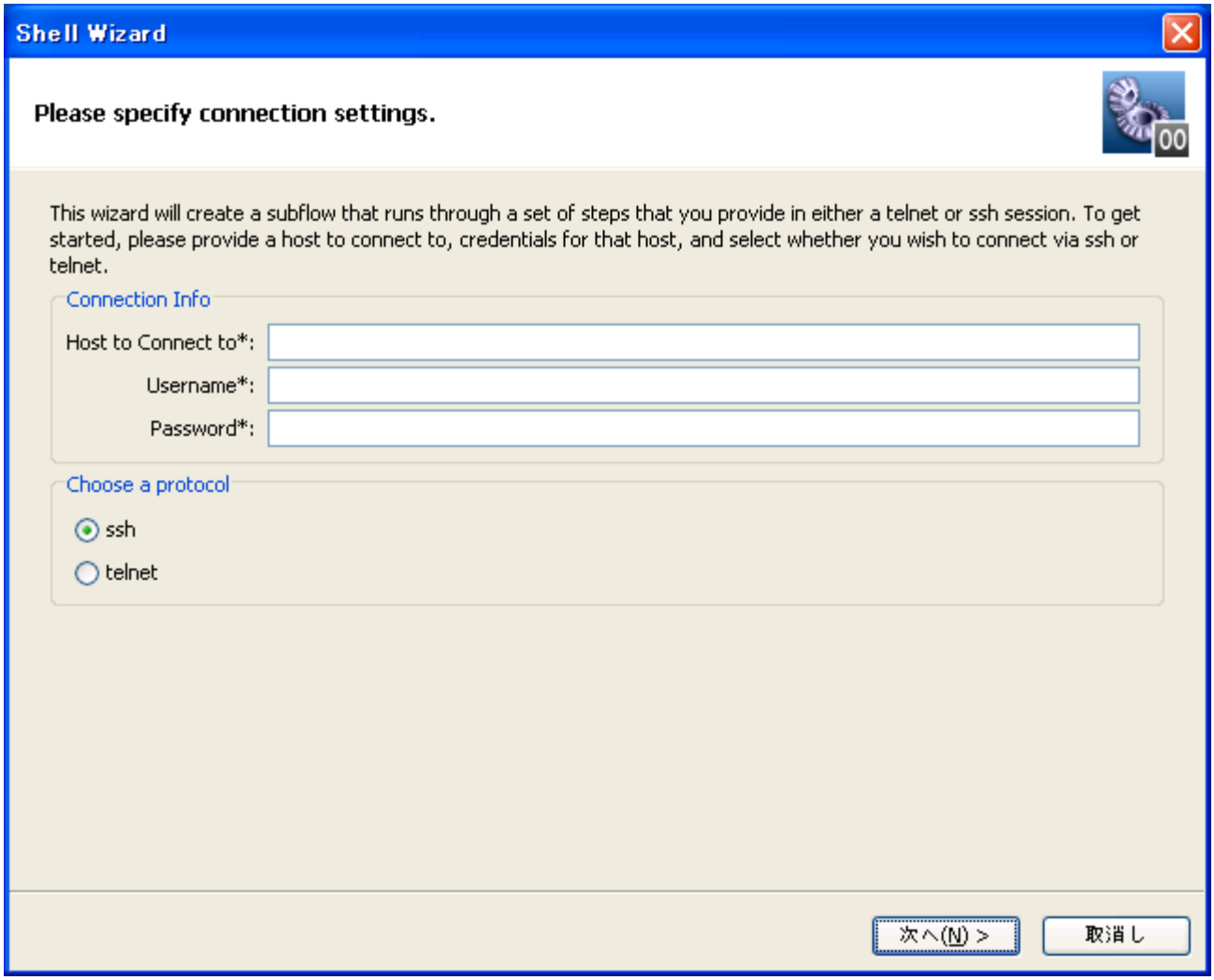


図 142 - Shell Wizard、接続の設定

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または IP アドレスを入力します。
7. **[Usernam]** 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、**[次へ]** 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。

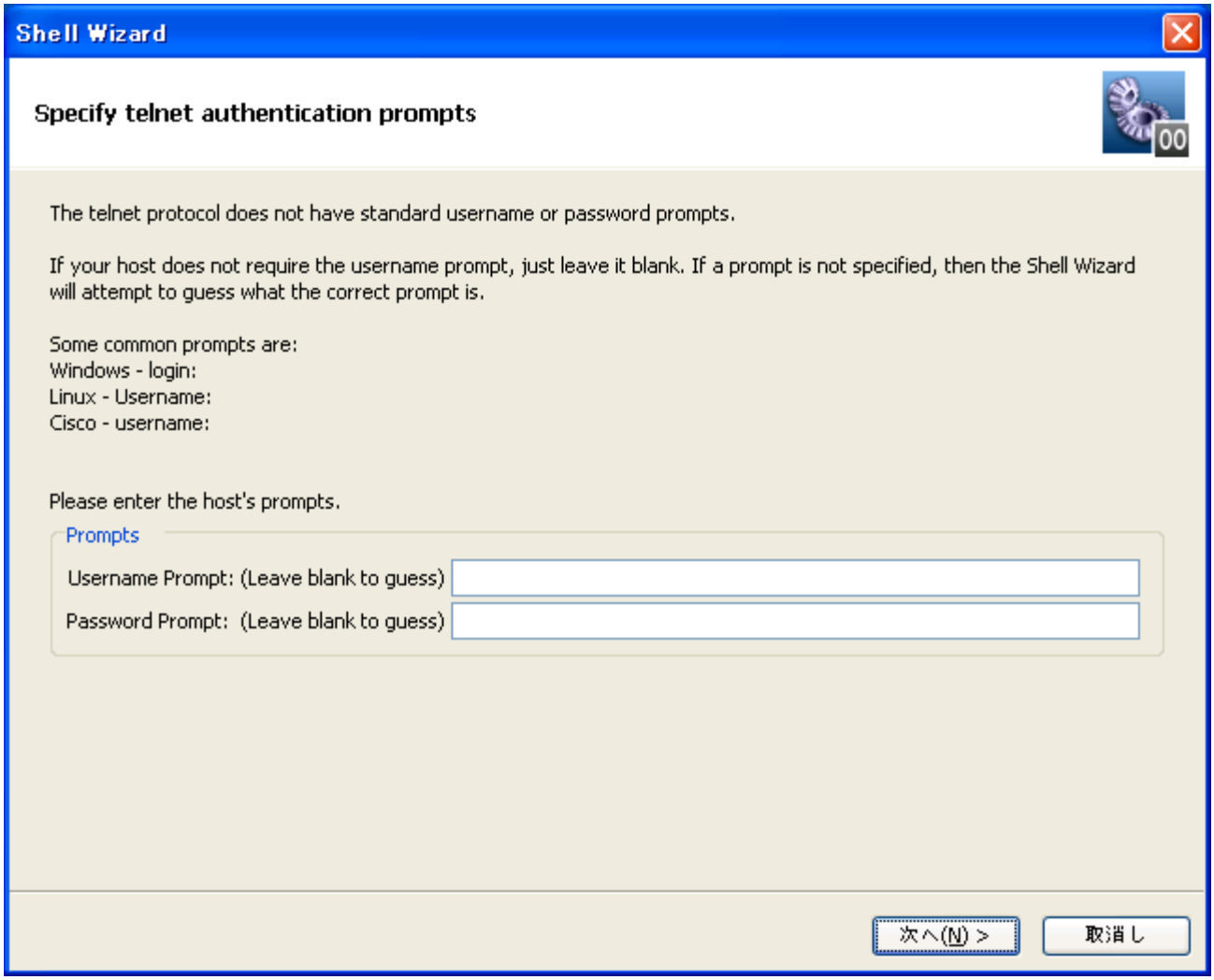


図 143 - 資格情報にホストプロンプトを指定

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。

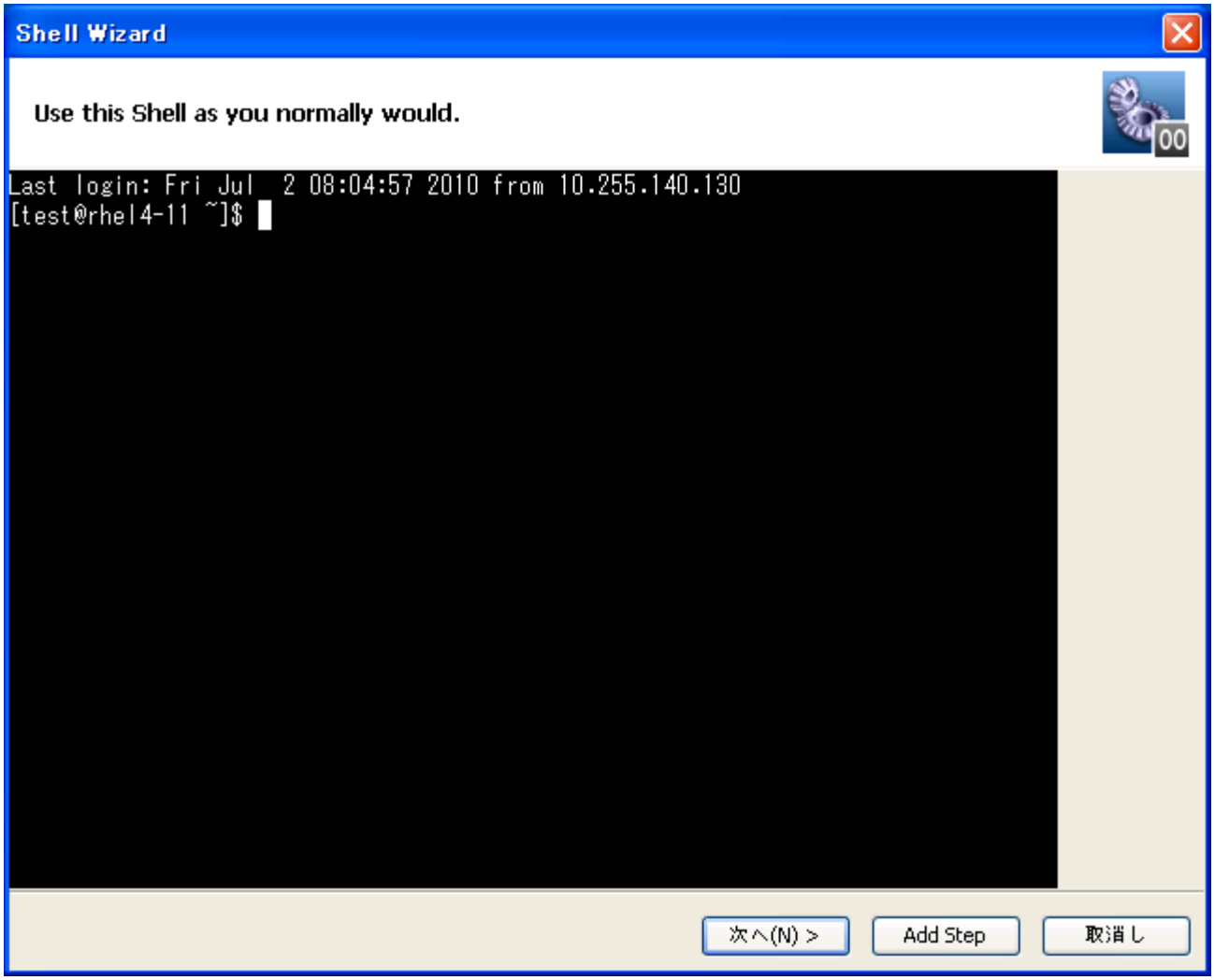


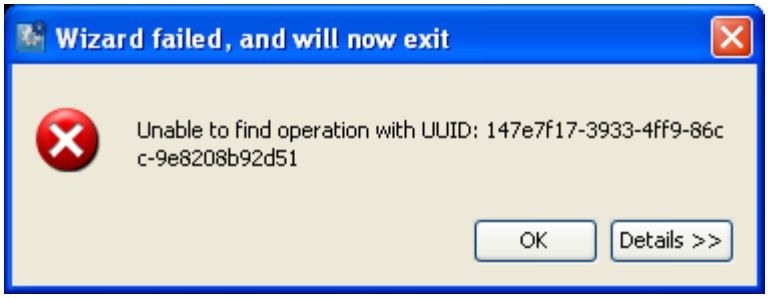
図 144 - ホストコンピューターのコマンドライン

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、**[次へ]** 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]** フォルダー、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ポイントする。

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 **[説明]** タブに次の情報を追加します（同じテクノロジーと情報をやり取りする複数のフローまたはオペレーションを作成する場合、1 つのフォルダーにまとめて、フォルダーの **[説明]** タブにこの情報を記載します。これはデフォルトの HP OO コンテンツのプラクティスです）。 **[説明]** タブへの情報の追加により、他の作成者や Central ユーザーがドキュメントの生成機能を利用できるようになることにご注意ください。 ドキュメントの生成の詳細については、「*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。
- 任意に追加する実装に関する**注意点**には、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しているコマンドがメニュー駆動型である、視覚的に書式化され（表形式など）ている、印字不能文字が含まれる、または画面に収まりきらない場合、結果を資格化して取得すると、結果の読み取りやフィルタリングに役立ちます。

視覚化された結果は、XML フォーマットで結果を表示したものです。XML フォーマットは表などの形式を維持し、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含まれ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インスペクターを開きます。
2. **[結果]** タブで新しい結果を追加し、**[ソース]** 列で **[Field:visualized]** を選択します。

正規表現での作業

正規表現（regex とも呼ばれる）という強力なツールを使用すれば、以下のようなことが可能になります。

- 次のような目的で、結果からデータの一部を抽出するフィルターを作成する。
  - 後のオペレーションで使用するためにデータを変数に保存する。
  - ステップのレスポンスを決定するためにデータをテストする。
- 入力形式を検査する。

たとえば、入力変数が、パケット損失に関する情報を **ping** オペレーションから必要としていると仮定します。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、“Lost = ”と、その後に続く任意の数字を検索します。 ここで正規表現が役に立ちます。

正規表現を使うと、一致するテキストだけでなく、文字の種類も検索することができます。たとえば、ワイルドカードの \d を使用すると任意の数字を検索できます。

したがって、前述の例で正規表現 `Lost = \d` を使用して検索を実行すると、文字列 Lost = 0 を検索結果として得ることができます。

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
^	文字列の先頭に一致
$	文字列の末尾に一致
.	改行以外の任意の文字
\b	単語境界
\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

ワイルドカード	用途
\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

修飾子	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
\	特殊文字をエスケープする (例えば、\. の場合は を意味し、どの文字にも一致しない）。

これらのワイルドカードや修飾子を使用して、次の正規表現で **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、複数の正規表現を組み合わせて、必要な値を抜き出すことができます。 例として、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、Unix の **ps** オペレーションを開きます。
2. **結果フィールドのフィルター**ダイアログボックスを開きます。

   結果フィールドのフィルター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、2 つの正規表現が必要です。1 つ目で出力を ps の行まで絞り込んで、2 つ目で時間を抽出します。
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、**[行単位でフィルター]** ボックスにチェックをつけます。
5. **[選択したフィルターのテスト]** をクリックします。

   [出力のテスト] ボックスで出力されるのは、「ps」の文字が入った行だけとなります。
6. 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. **[選択したフィルターのテスト]** 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。

変数への値の格納については、*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。
   - **[管理]** タブをクリックします。

     現在実行中のフローが、**[実行の管理]** 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実稼働のサーバーにパブリッシュします。
7. 実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 での実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仮定します。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）。
- **ロックの取得** 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。

  ロックの取得 および **ロックの解放** 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 **[説明]** 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 ケイパビリティにかかわらず、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. フローの **[プロパティ]** シートを開くには、**[リポジトリ]** ペインでフローに移動して右クリックし、次に **[プロパティ]** をクリックします。
2. **[プロパティ]** シートの **[詳細]** タブで、**[Central では非表示]** 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. **リポジトリ** ペインで、フローを右クリックして **[アクセス許可]** をクリックします。
2. **[アクセス許可]** 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 と通信します。 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を経由したシステムとの通信
- IAction を介して RAS が相互作用可能なエンティティと OO との統合

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 の作成方法については、OO 『*SDK Guide*』（SDKGuide.pdf）の IAction の作成の項を参照してください。

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。

作成者へのメッセージの表示

Studio は現在ログインしている作成者にメッセージを表示できます。

メッセージを表示するには

- [Studio] の下部にある **[メッセージ]** タブをクリックします。

図 145 - [メッセージ] タブ(その他を含む)

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。

重要情報: 1 つのユーザーアカウントで、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ないでください。 そのように接続した場合、リポジトリ内のユーザーの作業スペースが壊れることがあります。 この現象は、Central がクラスタ化されたときに発生する可能性もあります。

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. HP OO ホームディレクトリの **\Central\conf** サブディレクトリにある Central.properties ファイルの中から、次の行を検索します。
   ```
   dharma.app.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。
   ```
   dharma.app.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が失われ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でステップに加えた変更は保存される。

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ない。

この後の保存は正常に機能します。

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, 52

[自分の変更/チェックアウト] パネル
  概要, 14

「フロー」.「オペレーション」を参照

Central
  RAS に依存するオペレーション, 172

Cmd オペレーション, 198

Configuration フォルダー
  スクリプトレット、使用, 107

Configuration フォルダーのオブジェクト
  削除したオブジェクトの復元, 59

DescribeFlows.vm
  テンプレートファイルの構成, 53

HP OO オブジェクト
  権限、設定, 193

Http オペレーション
  プロパティ、レスポンス定義, 201

IAction
  オペレーション用に作成, 218

Lucene 構文、検索ペインで使用, 41

OO オブジェクト
  アクセス管理, 192
  権限, 192

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, 202

RAS
  ACL, 173
  アクセス制御リスト, 173
  オペレーション, 179
  オペレーションの作成, 180
  オペレーションの追加, 180, 181
  使用, 171
  使用、要件, 173
  使用できるかどうか、, 173
  既存, 173
  権限, 173
  要件, 171

RAS オペレーション
  RAS 参照の変更, 177
  プロパティ、レスポンス定義, 203

RAS に依存するオペレーション
  トラブルシューティング, 178

RAS 参照
  RAS オペレーションの変更, 177
  上書き, 177
  再設定, 175
  削除, 176
  変更, 177
  既存の RAS からの設定, 179
  権限の設定, 193
  追加, 174

Regex.「正規表現」を参照

Shell Wizard
  フローの作成に使用, 209

Studio.「Studio」を参照
  キーボードショートカット, 22
  ライブラリ, 13
  初心者向け機能, 1
  図解, 12
  起動, 11

Telnet オペレーション
  プロパティ、レスポンス定義, 207

Web サービス
  オペレーションの作成, 168

Web サービスウィザード
  ログ記録, 171

Web サービスコンテンツ
  使用, 182

XML
  解析, 120

XML フィルター, 120

XML 属性の取得フィルター, 121

XML 要素の取得フィルター, 122

XML 要素値の取得フィルター, 123

XPath クエリフィルター, 124

アイコンパネル
  ドッキング, 19, 21
  フローティング, 19, 21
  概要, 19

アクセス権限
  グループのデフォルト設定, 195
  グループマスクの作成, 195

アクセス管理
  オーサリングオブジェクト, 192

アクセラレーターパック

インポート, 156
インスペクター
キーボードショートカット, 23
開く, 63
エディター、オブジェクト
概要, 22
エバリュエーター
作成, 183
削除, 184
編集, 184
オーサリングオブジェクト
アクセス管理, 192
オブジェクトのプロパティシート、概要, 22
オペレーション
IAction で作成、概要, 219
IAction を作成, 218
Lucene 構文を使用した検索, 41
RAS からの作成, 179, 180
Web サービス, 178
Web サービスからの作成, 168
アーキテクチャ, 7
オプション入力, 198
クイックビュー, 4
コア機能, 7
コピー, 56
シール, 37
シール、定義, 9
スクリプトレット, 8
スクリプトレット、作成, 103
ステップとの比較, 9
チェックアウト, 24, 25
チェックイン, 24, 25
ブックマーク, **57**
プライマリ出力の指定, 108
プロパティ, 198
プロパティシート, 197
レスポンス, 8, 107, 109
レスポンス定義, 198
レスポンス用のスクリプトルール, 104
作成, 196
入力, 10
出力, 107, 108
出力、フィルタリング, 111
出力、削除, 109
出力、追加, 109
出力データ, 8
出力の削除, 109
削除したオペレーションの復元, 59
参照元, 55
参照先, 55
名前変更, 57
妥当性の要件, 196
定義, 7
情報フロー, 7
未加工結果, 7
検索, 37, 38
検索ペインからのアクセス, 42
検索対象, 38
権限の設定, 193
特定のオペレーションを使用するフローの検索, 55
複数の表示, 44
詳細情報の表示, 55
説明, 42
高度な作成, 6
高度な概念, 9
オペレーションタイプ, 198
Cmd、またはコマンドライン, 198
シェル, 208
フローの実行カウンター, 199
フローの実行概要レポート, 200
オペレーションの参照
検索, 55
オペレーションの説明
表示, 43, 44
オペレーションレスポンス
ルール, 125
オペレーション出力
フィルタリング, 111
カテゴリ, 188
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
インスペクター, 23
スクリプトレットタブ, 23
デバッガー, 23
ブックマークペイン, 23
プロパティエディター, 23
リポジトリペイン, 23
作成キャンバス, 23
作成ペイン, 23
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, 184
グループ
アクセス権限のデフォルト設定, 195
グループの権限, 192
グループマスク
作成, 195
グローバルコンテキスト, 7
グローバル変数, 87
システムプロパティ, 129

値の取得, 88
定義, 129
コールアウト
パレットから追加, 15
フローへの追加, 36
コマンドラインオペレーション, 198
コンテキスト
グローバル、定義, 7
ローカル、定義, 7
定義, 7
ご注意, ii
保証, ii
商標, ii
権利の制限, ii
著作権, ii
サブフロー
デバッグ、ステップアウト, 140
デバッグ、ステップイン, 140
使用, 131
サンプルフロー
操作方法フォルダー, 1
シール
定義, 37
シェルオペレーション, 208
Shell Wizard, 209
使用, 214
未加工結果の視覚化, 214
システムアカウント, 190
作成, 191
削除, 191
編集, 191
システムエバリュエーター, 182
システムオブジェクト
権限、設定, 193
システムプロパティ, 129
作成, 130
値の取得, 130
値の変更, 131
定義, 129
権限の設定, 193
ショートカット
キーボード, 22
スクリプトレット, 103
Configuration フォルダー、使用, 107
デバッグ, 105
並列分岐ステップ, 94
作成, 103, 104
再使用のための保存, 107
定義, 8
結果のフィルタリング, 104
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, 205
スクリプトレットタブ
キーボードショートカット, 23
スクリプトレットフィルター, 116
スクリプトレットフォルダー, 107
スケジュール
入力, 81
ステップ
アイコンの変更, 128
オペレーションとの比較, 9
オペレーションの変更, 133
コピー, 63
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, 100
ノンブロッキング、作成, 100
フローへの追加, 62
フロー間でのコピー, 63
ユーザープロンプト, 64
作成, 60, 61, 62
保存, 102
入力, 10
単一レスポンス, 101
実行, 6
接続, 65
移動, 64
結果, 107, 109
結果、削除, 111
結果、追加, 110
編集, 63
説明, 134
追加, 61
高度な概念, 9
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
ステップアイコン
変更, 128
ステップ出力
保存, 102
スナップショット
および現在のリポジトリの状態, 166
パブリッシュ, 166
リストのフィルタリング, 164
作成, 160, 161
削除, 166
復元元, 166
概要, 160
編集, 163
表示, 163
閉じる, 164

スペースの削除フィルター, 118
セカンダリ出力
追加, 109
セキュアなシェルオペレーション
プロパティ、レスポンス定義, 206
セルフドキュメンテーション
フロー, 44, 49
ソートフィルター, 118
ターゲットリポジトリ
設定, 147
ダッシュボードでのレポート作成
クイックビュー, 184
チェックアウト, 11
取り消し, 30
チェックイン, 11, 27
変更の破棄, 30
強制, 29
競合の解決, 28
チェックポイント, 101
ツールバー
オプション表示パレット, 16
ステップおよびコールアウトのパレット, 15
パレット, 15
概要, 15
データ
親フローへの引き渡し, 132
データの移動
追跡, 84
データ移動, 132
テーブルフィルター, 119
デバッガー
キーボードショートカット, 23
フローのリセット, 135
フローの再開, 135
フロー変数の値の変更, 141
概要, 135
デバッグ
サブフロー, 140
テンプレート
フローの作成, 35
テンプレートファイル, 53
構成, 53
ドキュメントの生成
オペレーション, 44, 49
カスタム, 51
カスタムテンプレートの編集, 54
コマンド, 44, 51
ドメインターム
作成, 185
値、削除, 187
値、変更, 187
値、追加, 187
定義, 185
編集, 187
追加, 186
トランジション, 65
作成, 65
移動, 69
追加, 67
ノンブロッキングステップ, 88, 100
作成, 100
バージョン
前のバージョンへの復元, 31
前のバージョンを開く, 31
比較, 32
管理, 30
バージョン管理, 11, 24
パネル
ドッキング, 21
フローティング, 21
パネルタブ
移動, 21
パラレル処理. 「並列処理」を参照
パレット
オプション表示, 16
ステップおよびコールアウト, 15
フィルター, 111
XML, 120
XML 属性の取得, 121
XML 要素の取得, 122
XML 要素値の取得, 123
XPath クエリ, 124
スクリプトレット, 116
スペースの削除, 118
ソート, 118
テーブル, 119
フォーマット, 114
作成, 112
保存, 113, 125
再使用, 125
削除, 118
大文字と小文字の変換, 114
数値の丸め, 116
数値の抽出, 114
正規表現, 115
範囲の選択, 117
置換, 115
行カウント, 115

行の選択, 117
重複行の削除, 115

フォルダー
作成, 32
権限の設定, 193

ブックマーク, **57**
シェルフ、名前の変更, 58
シェルフ、移動, 59
シェルフ、表示, 59
シェルフ、追加と削除, 58
シェルフ、非表示, 59
フローおよびオペレーション、追加と削除, 58

ブックマークパネル, 57
ドッキング, 18
ドッキング, 21
フローティング, 18, 21
概要, 18

ブックマークペイン
キーボードショートカット, 23

プライマリ出力
指定, 108

ブレークポイント
有効化, 142
無効化, 142

フロー, 33
Central で非表示, 217
Central によるアクセスの制限, 218
Lucene 構文を使用した検索, 41
インポート, 156
カテゴリ, 188
ゲートトランジション, 217
コールアウトの追加, 36
コピー, 56
サブフロー, 132
シール, 37
シール、定義, 9
ステップ, 61, 「ステップ」を参照
ステップの作成, 61
ステップの追加, 61
セルフドキュメンテーション, 44, 49
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, 101
データの提供, 72
テスト, 217
デバッグ, 135
デバッグ、リセット, 141
デバッグ、再開, 141
テンプレートから作成, 35
ブックマーク, **57**
リポジトリ間の移動, 146
レスポンス、追加, 128
レスポンスの削除, 128
作成, 4, 33, 34
使用制限, 217
入力, 10, 72
出力, 107
削除したフローの復元, 59
名前変更, 57
変更の配布, 217
検索, 38
検索, 37
概要, 1
権限の設定, 193
自動実行, 134
複数ドメイン、実行, 179
要件, 3
説明, 42
高度な概念, 9

フロー、, 2

フロー、作成
概要, 33

フローの, 6

フローの 出力 フィールド, 132

フローの実行カウンターオペレーション, 199

フローの実行概要レポートオペレーション, 200

フローの説明
表示, 43

フロー図, 17

フロー変数, 82
グローバル, 9
スクリプトレット, 83
デバッガーでの値の変更, 141
フロー変数パネルでのフィルタリング, 86
マルチインスタンスステップ, 99
リストの入力, 99
ローカル, 9
作成, 82
作成, 83
値の割り当て, 83
管理, 84
範囲, 9
追跡, 84

フロー変数パネル, 84
ドッキング, 21
フィルタリング, 86
フローティング, 21
フロー変数の管理, 87

フロー変数ペイン、概要, 21

フロー状態データ
定義, 101

フロー設計

単純化, 131
注意事項, 131

プロパティエディター
キーボードショートカット, 23

マルチインスタンスステップ, 88, 94
サブフロー, 132
データの移動, 99
デバッグ, 100
フロー変数, 99
フロー設計, 95
リスト入力, 99
作成, 98
制限, 100

メッセージ
ユーザーへの表示, 188
作成者へ、表示, 219

ユーザー
切り替え, 144

ユーザープロンプト
選択リストの対象, 189

ユーザープロンプトの入力
リストの作成, 78, 79

ユーザーメッセージ
表示, 188

ライブラリ、概要, 13

ライブラリオブジェクト
エディター、概要, 22
チェックアウト, 26
チェックイン, 27
バージョンの比較, 32
前のバージョンへの復元, 31
前のバージョンを開く, 31
復元, 59

リスト
ユーザープロンプト用, 78, 79

リストの入力
フロー変数, 99

リスト入力
マルチインスタンスステップ, 99

リターン, 70

リターンステップ
パレットから追加, 15
レスポンスの変更, 70, 71
定義, 70
追加, 70

リポジトリ, 144
インポート, 156
エクスポート, 154
スナップショット、概要, 160
スナップショットの作成, 160
ソース, 146
ターゲット, 146
ターゲットリポジトリの設定, 147
バックアップ, 160
バックアップ、インポート, 154
パブリック, 144
パブリッシュ, 148
パブリッシュ、元に戻す, 154
パブリッシュのプレビュー, 148
プライベート, 144
復元, 160
復号化, 159
暗号化, 158
暗号化されたコピー, 159
更新, 151
更新、元に戻す, 154
更新のプレビュー, 151
検査, 158
状態、保存, 160
追加, 145
開く, 146

リポジトリ、パブリック
パブリッシュ, 147
更新, 147

リポジトリ、暗号化
開く, 159

リポジトリ、現在
表示, 164

リポジトリスナップショット. 「スナップショット」を参照, 「スナップショット」を参照

リポジトリペイン
キーボードショートカット, 23

リモートアクションサービス. 「RAS」を*参照*, 「RAS」を参照

レーン
コピー, 92
サイズ変更, 93
削除, 93
名前変更, 93
移動, 92
複製, 93
追加, 93
開始ステップの変更, 93

レーンステップ
コピー, 92
移動, 92

レスポンス
デバッガーでのオーバーライド, 143
フローからの削除, 128
定義, 8, 109

ローカルコンテキスト, 7
ログイン
　切り替え, 144
ログファイル
　WebServiceWizard.log, 171
並列処理
　マルチインスタンスステップ, 94
並列分岐ステップ, 88
　コピー, 92
　スクリプトレット, 94
　データの移動, 94
　デバッグ, 94
　パレットから追加, 15
　レーンのサイズ変更, 93
　レーンの削除, 93
　レーンの名前変更, 93
　レーンの複製, 93
　レーンの追加, 93
　レーンの順序, 90
　作成, 90
　移動, 92
　開始ステップの変更, 93
並列分岐ステップのレーンステップ
　移動, 92
並列分岐ステップレーン
　移動, 92
並列実行, 88
作成
　リポジトリ、開く, 146
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
　フロー図, 17
作成ペイン、概要
　サクセイペインガイヨウ, 15
作成者、複数.「複数の作成者」を参照
保証, ii
入力, 10, 72
　グローバル変数, 87
　ダッシュボードでのレポート作成用の記録、クイックビュー, 184
　データソースの割り当て, 74
　フロー実行のスケジュール, 81
　ユーザープロンプト、リストの作成, 78, 79
　作成, 72
　削除, 82
　形式の検証, 81
　暗号化, 74
　消去, 82
入力値、作成
　フローユーザー入力から, 76
　一覧からのユーザーの選択から, 77
　前のステップの結果から, 79
　定数値, 76
　指定可能な値のリスト, 77
入力値、割り当て
　資格情報から, 80
出力, 108
　セカンダリ, 108
　ソースの変更, 109
　プライマリ, 108
　削除, 109
　説明, 10
　追加, 109
出力データ, 8
初心者向け機能, 1
削除
　復元, 59
削除フィルター, 118
単一レスポンスのステップ
　作成, 101
同時実行, 88
商標, ii
変数.「フロー変数」を参照
変更
　破棄, 30
大文字と小文字の変換, 114
形式フィルター, 114
操作方法フォルダー, 1
数値の丸めフィルター, 116
数値の抽出フィルター, 114
文字列形式.「エバリュエーター」*を参照*
未加工出力
　定義, 108
未加工結果
　定義, 7
検索
　あいまい一致, 38
　完全一致, 38
　定義、検索パネル, 38
検索パネル, 38
　検索の定義, 38
検索ペイン
　Lucene 構文の使用, 41

オペレーションへのアクセス, 42

概念
高度, 9

構成アイテム
定義, 185

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

権利の制限, ii

権限
HP OO オブジェクトへの設定, 193
システムオブジェクトへの設定, 193
設定, 192

正規表現, 215

正規表現フィルター, 115

競合
解決, 28

管理者
強制チェックイン, 29

範囲の選択フィルター, 117

結果
ソースの変更, 111
フィルタリング, 113
フィルタリング用スクリプトレット, 104
削除, 111
説明, 10
追加, 110

結果、ステップ
定義, 109

置換フィルター, 115

自分の変更/チェックアウトパネル
ドッキング, 21
フローティング, 21

著作権, ii

行カウントフィルター, 115

行の選択フィルター, 117

表示
現在のリポジトリの状態に戻す, 164

複数ドメイン
フローの実行, 179

複数の作成者
バージョン管理, 11

説明, 134

説明、オペレーション
表示, 43, 44

説明、フロー
表示, 43

選択リスト, 189
権限の設定, 193

重複行の削除フィルター, 115

開始ステップ
変更, 63