

HP Operations Orchestration

ソフトウェアバージョン: 10.50

Windows オペレーティングシステム

Studio オーサリングガイド

ドキュメントリリース日: 2015 年 9 月 (英語版)

ソフトウェアリリース日: 2015 年 9 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。ここでの記載は、追加保証を提供するものではありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR 12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

© Copyright 2005–2015 Hewlett-Packard Development Company, L.P.

商標について

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

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

UNIX® は、The Open Group の登録商標です。

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

ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

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

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

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの登録は、次のWebサイトから行なうことができます。

<http://h20229.www2.hp.com/passport-registration.html>

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

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

サポート

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

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

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

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

一部のサポートを除き、サポートのご利用には、HP Passportユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDを登録するには、次のWebサイトにアクセスしてください。

<http://h20229.www2.hp.com/passport-registration.html>

アクセスレベルの詳細については、次のWebサイトをご覧ください。

http://h20230.www2.hp.com/new_access_levels.jsp

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

このPDF版オンラインヘルプについて

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

目次

『HP Operations Orchestration Studio オーサリングガイド』へようこそ	8
図解による HP 00 Studio 概要	8
HP 00 Studio の開始 - ワークフローの主なステップ	23
HP 00 Studio ウィンドウの外観の調整	23
作成のベストプラクティス	25
一般的なベストプラクティス	25
共有コンテンツのベストプラクティス	26
命名のベストプラクティス	28
フローのベストプラクティス	30
オペレーションのベストプラクティス	31
ステップのベストプラクティス	33
トランジションのベストプラクティス	35
入力のベストプラクティス	35
デバッグのベストプラクティス	36
Studio の構成のベストプラクティス	36
説明のベストプラクティス	37
ソースコントロール管理のベストプラクティス	44
Subversion	44
Git	45
HP 00 Studio での各言語の使用 - ローカライズ	45
コンテンツパックのローカライズ	45
プロジェクトのローカライズ	46
cp.properties ファイル	46
プロジェクトの使用	48
プロジェクトの管理	48
[プロジェクト]ペインを使用したフォルダーの管理	55
HP 00 Studio のソースコントロールの使用	57
HP 00 Studio のソースコントロールとは	57
参考資料	58
Subversion ソースコントロール管理の使用	60
用語集	60
初期のソースコントロールリポジトリの作成	61
SVN での複数の作成者による作業	65
ロックポリシーの強制	66
Git ソース管理システムの使用	84

Git の用語	84
Studio で Git を使用する前に	86
Git のブランチのサポート	87
Git リポジトリログについて	87
Git での競合の処理	91
認証オプション	92
コンテンツパックの使用	126
コンテンツパックのプロジェクトへのインポート	126
プロジェクトのコンテンツパックと依存関係の管理	128
依存関係	129
構成アイテムの管理	142
構成アイテムの使用	142
構成アイテムの検索	142
構成アイテムフォルダーの使用	143
カテゴリの設定	145
ドメインタームの設定	147
グループエイリアスの設定	150
役割エイリアスの設定	155
スクリプトレットの設定	157
選択リストの設定	162
システムアカウントの設定	165
システム評価子の設定	168
システムフィルターの設定	175
システムプロパティの設定	181
フロー作成 - 基本	185
フローの作成 - 手順	185
新しいフローの作成	188
フロー内でのステップの作成	191
フローの外観の調整	197
フローの変更	200
入力の作成	205
入力ソースの指定	214
入力データの評価	227
トランジションの作成	228
レスポンスの設定	233
出力と結果の作成	242
オペレーションの出力の設定	243
ステップの結果の設定	246
出力および結果のフィルタリング	255

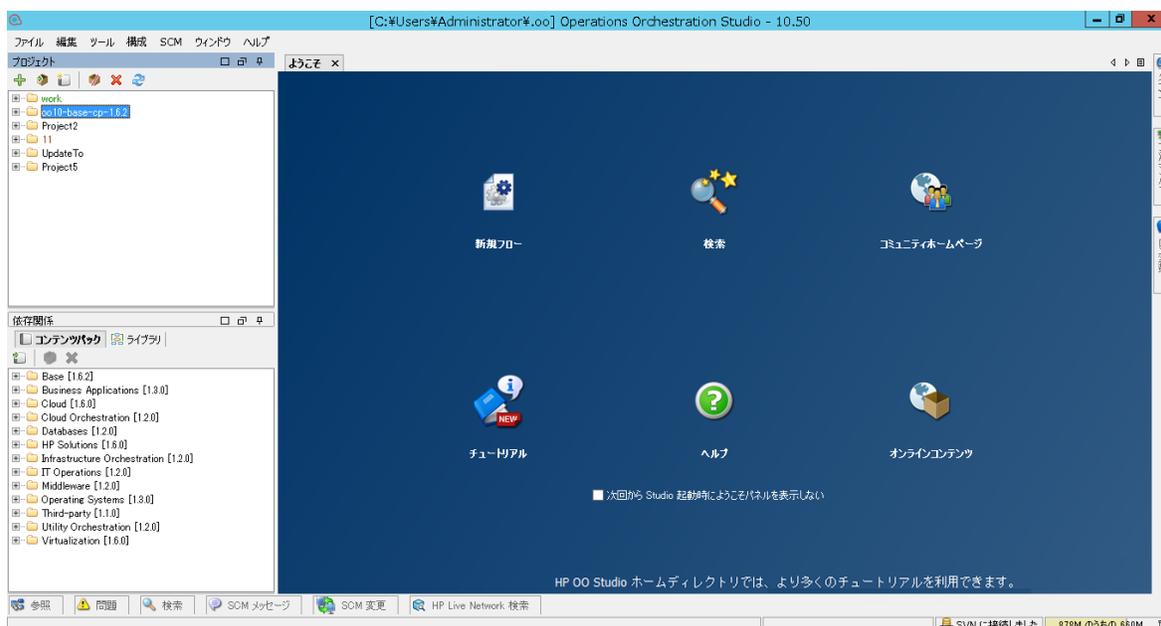
変数の使用	272
リターンステップの作成	278
高度な作成	282
フロー内でのサブフローの作成	282
並列分岐ステップのフローの作成	285
マルチインスタンスステップのフローの作成	289
フロー内でのスクリプトレットの使用	297
フローでの正規表現の使用	302
HP Live Network の内容を Studio から検索	307
検索の種類	307
hpln-index-generator ツール	307
参考資料	310
コンテンツの検査	311
[問題] ペインでのフローの検査	311
フローのテストとデバッグ	313
ユーザー認証が必要なフローのデバッグ	314
複雑なフローのデバッグ	326
リモート Central の Studio でのデバッグ	327
一般的なワークフロー	329
リモートデバッガー接続による証明書の自動インポート	332
リモート Central でのフローのデバッグ	333
コンテンツパックのエクスポート	336
コンテンツパックのバージョン管理のライフサイクル	336
リビジョンコントロールを使用しないコンテンツパックのバージョン管理	337
リビジョンコントロールを使用したコンテンツパックのバージョン管理	338
フローとオペレーションの管理	349
オペレーションの作成	349
フローまたはオペレーションの検索	357
フローおよびオペレーションのコピー	363
ソフトコピーからハードコピーへの切り替え	365
ハードコピー内のプラグインの置き換え	365
フローとオペレーションの使用方法の詳細	366
フローおよびオペレーションに関するドキュメントの生成	368
フローとオペレーションのバージョン履歴管理	373
フローおよびオペレーションのブックマーク	376
Studio プロパティの設定	381
[割り当て元] と [割り当て先] のデフォルト値を定義する	381

[未指定時]のデフォルト値を[定数を使用する]に変更する	381
コンテキストインスペクターに特殊文字を表示する	382
参考資料	387
トラブルシューティング	393
HP 00 9.x からアップグレードする際のトラブルシューティング	393
Studio のユーザーインターフェイスアイテムの場所	393
HP 00 9.x と 10.x のバージョン比較	394
HPLN のトラブルシューティング	395
GIT のトラブルシューティング	401
ログビューアーでの Studio エラーの表示	402

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

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

図解による HP OO Studio 概要



Studio は主に以下の要素で構成されています。

- **[プロジェクト]** ペイン (左側)。作業中のプロジェクトと、編集可能なフロー、オペレーション、その他のプロジェクトで使用可能な HP OO オブジェクトが表示されます。
- **[依存関係]** ペイン (左側)。インポートされたコンテンツパックが表示されます。このペインでは、コンテンツパックをインポート、削除、閉じることができます。**[依存関係]** ペインには次の2つのタブがあります。
 - **コンテンツパック** - 複数のコンテンツパックごとに複数のツリーが表示されます。このビューから、コンテンツパックのクローズ、削除、またはインポートを実行できます。
 - **ライブラリ** - すべてのコンテンツを汎用の **[Library]** フォルダの下にまとめたツリーが1つ表示されます。このビューから、コンテンツパックをインポートできます。

- **[作成]** ペイン (中央)。フローを作成ペインで開いている場合、次の3つのタブが、作成ペインの下部で使用可能になります。
 - **[設計]** タブ。フロー図で作業できます。
 - **[プロパティ]** タブ。フロー、オペレーション、構成オブジェクトのプロパティを設定できます。
 - **[インスペクター]** タブ。個々のステップとトランジションのプロパティを設定できます (**[設計]** タブを開いているときのみ使用可能)。
 - **[ようこそ]** タブ (中央)。Studio を最初に起動した時に、**[ようこそ]** タブが作成ペインに表示されません。
 - **[アイコン]** ペイン (右側)。オペレーションまたはステップで使用するアイコンが集められています。**[アイコン]** タブをクリックするとこのペインが開きます。
 - **[ブックマーク]** ペイン (右側)。よく使用するオペレーションやフローのショートカットを保存できます。**[ブックマーク]** タブをクリックするとこのペインが開きます。
 - **[フロー変数]** ペイン (右側)。フローで使われているフロー変数と、各フロー変数の使用方法のリストと説明が表示されます。**[フロー変数]** タブをクリックするとこのペインが開きます。
 - **[参照]** ペイン (下部)。既存のフローでフローやオペレーションがどのように使用されているかを表示します。**[参照]** タブをクリックするとこのペインが開きます。
 - **[問題]** ペイン (下部)。選択したフローやオペレーションで発生している問題を表示します。**[問題]** タブをクリックするとこのペインが開きます。
 - **[SCM メッセージ]** ペイン (下部)。ソースコントロール関連のメッセージを表示します。**[SCM メッセージ]** タブをクリックするとこのペインが開きます。詳しくは、[「ソースコントロールの使用」](#)を参照してください。
 - **[SCM 変更]** ペイン (下部)。ソースコントロールの最新の変更を表示します。**[SCM 変更]** タブをクリックするとこのペインが開きます。詳しくは、[「ソースコントロールの使用」](#)を参照してください。
 - **[SCM リポジトリログ]** ペイン (下部)。SCM リポジトリ (Git のみ) に対して加えられた変更のログを表示します。**[SCM リポジトリログ]** タブをクリックするとこのペインが開きます。詳細については、[「Git ソース管理システムの使用」 \(84ページ\)](#)を参照してください。
 - **[ログビューアー]** ペイン (下部)。現在のユーザーセッションで発生したすべてのエラーを表示します。**[ログビューアー]** タブをクリックするとこのペインが開きます。詳細については、[「ログビューアーでの Studio エラーの表示」 \(402ページ\)](#)を参照してください。
- 注:** **[ログビューアー]** タブは、**[Windows]** メニューの **[Studio ログビューアー]** を選択して有効にした後のみ表示されます。
- **[検索]** ペイン (下部)。フロー、オペレーション、または構成アイテムを検索できます。**[検索]** タブをクリックするとこのペインが開きます。

- **HP Live Network Search ペイン** (下部)。HPLN プロファイルへのアクセス許可に基づいて、HP Live network 上の関連情報やコンテンツを Studio から直接検索できます。詳しくは、「[HP Live Network の内容を Studio から検索](#)」を参照してください。
- **ステータスバー** (下部)。現在のプロジェクトが接続されているソース管理システム (Git または SVN) およびその現在の状態を表示します。使用可能な合計ヒープサイズに加えて、現在のヒープサイズの使用率が表示されます。

注: 前に示した Studio 画面が表示されない場合は、画面の解像度に問題がある可能性があります。Studio を使用するときには、画面解像度を 1280 x 1024 ピクセル以上に設定する必要があります。

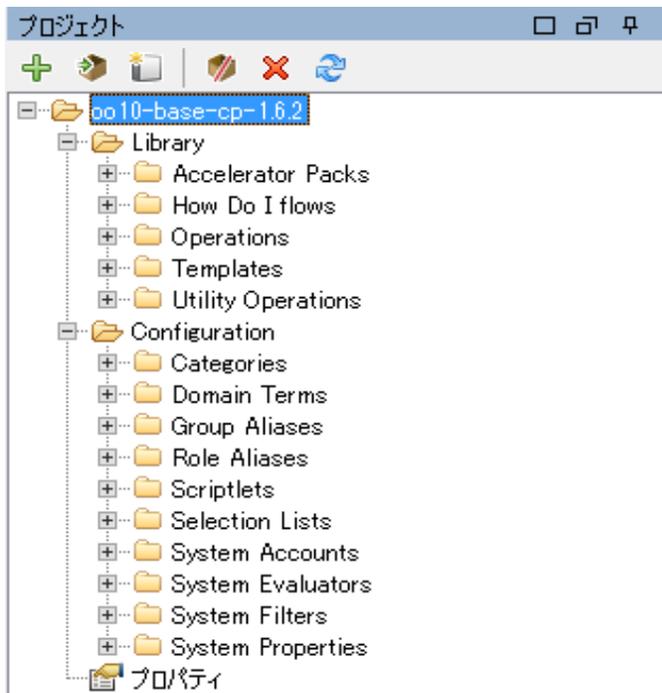
[プロジェクト] ペイン

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

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

注: 既存の構成アイテム構造の下のすべての構成アイテム内にフォルダーを作成できます。

- プロジェクトのプロパティを表示します。



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

プロジェクトの使用の詳細については、「[プロジェクトの使用](#)」(48ページ)を参照してください。

[依存関係] ペイン

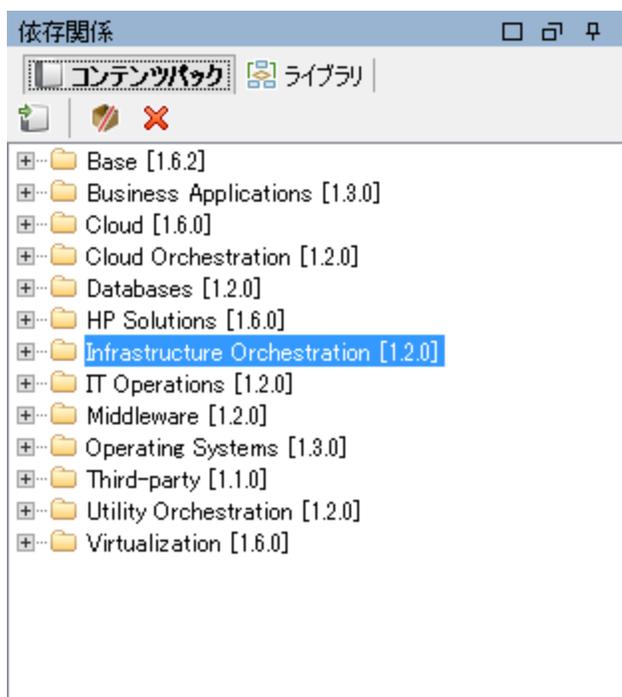
[依存関係] ペインには、使用可能なコンテンツパックが、オペレーションおよびフローが格納されたフォルダーとともに表示されます。

[依存関係] ペインには次の2つの表示オプションがあります。[コンテンツパック] ツリービューと、インポートしたすべてのコンテンツパックの集計ビューである[ライブラリ] です。

ビューを[ライブラリ] ペインに切り替えると、すべてのアイテムが同じツリーの下にマージされます。さらに、すべての構成アイテムが一般的な **Configuration** フォルダーの下にマージされます。ツリー内の共通のパスを持つすべてのアイテムが同じフォルダーにマージされます。

Studio を次に起動した時には、最後に選択したビューが自動的に表示されます。

- **コンテンツパック:** 複数のツリーが表示されます (複数のコンテンツパックの場合)。このビューから、コンテンツパックのクローズ、削除、またはインポートを実行できます。アイテムを右クリックすると、ドロップダウンメニューが表示され、このビューの利用可能なオプションが表示されます。
- **ライブラリ:** すべてのライブラリと構成アイテム (すべてのフォルダーとサブフォルダーを含む) が表示されます。このビューから、コンテンツパックをインポートできます。[コンテンツパック] ビューから削除されたコンテンツパックによって、[ライブラリ] ビューが自動的に更新されます。アイテムを右クリックすると、ドロップダウンメニューが表示され、このビューの利用可能なオプションが表示されます。

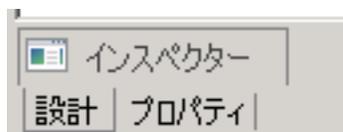


【作成】ペイン

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

フローが作成ペインで開いている場合、次の3つのタブが、使用可能になります。

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



【作成】ペインのツールバー

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

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



ボタン	機能
ステップパレット 	ステップパレットが開きます。ステップオブジェクトをキャンバスにドラッグします。
表示オプション 	表示オプションパレットの表示
ツリーで選択 	使用するフローまたはオペレーションを選択するためにライブラリツリーを展開します。
フローのデバッグ 	Central デバッガーの接続を追加または編集します。
ステップに移動 	フロー内の特定のステップに移動できます。移動先のステップの名前を入力するか、ステップの最初の何文字かを入力してリストから選択します。

作成ペインの使用の詳細については、「[フロー作成 - 基本](#)」(185ページ)を参照してください。

[ようこそ] タブ

Studio を最初に起動した時に、[ようこそ] タブが作成ペインに表示されます。タブを閉じたら、[ヘルプ] > [ようこそパネルの表示] を選択して再び開きます。



ボタン	説明
新規フロー	クリックして定義済みのテンプレートから新しいフローを作成します。
検索	クリックすると、[検索]ペインが開くので、コンテンツリポジトリのフローを検索できます。
コミュニティホームページ	クリックすると、HPLN コミュニティホームページに移動します。
チュートリアル	クリックして HP 00 のチュートリアルを表示します。
ヘルプ	クリックして HP 00 のヘルプを表示します。
オンラインコンテンツ	クリックすると、HP 00 コンテンツカタログのダウンロードページに移動します。
次回から Studio 起動時によろこパネルを表示しない	このチェックボックスをオンにすると、Studio を次に起動したときによろこ画面が表示されなくなります。

注: HP 00 のヘルプ、HPLN コミュニティホームページ、オンラインコンテンツへのリンクも、[ヘルプ]メニューから使用できます。

定義済みテンプレートからの新しフローの作成の詳細については、[「新しいフローの作成」\(188ページ\)](#)を参照してください。

フローの検索の詳細については、「[フローまたはオペレーションの検索](#)」(357ページ)を参照してください。

ステップパレット

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

テップパレット] ボタン  をクリックして、ステップパレットを表示します。



ボタン	説明
成功 	[解決済み] リターンステップをフローにドラッグできます。
診断済み 	[診断済み] リターンステップをフローにドラッグできます。
操作なし 	[操作なし] リターンステップをフローにドラッグできます。
失敗 	[エラー] リターンステップをフローにドラッグできます。
並列分岐ステップ 	[並列分岐ステップ] をフローにドラッグできます。
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] をフローにドラッグできます。
コールアウト 	[コールアウト] をフローにドラッグして、ユーザーに情報を提供できます。
ドッキングバー 	クリックして、パレットのドッキングとドッキング解除を行います。

リターンステップの使用の詳細については、「[リターンステップの作成](#)」(278ページ)を参照してください。

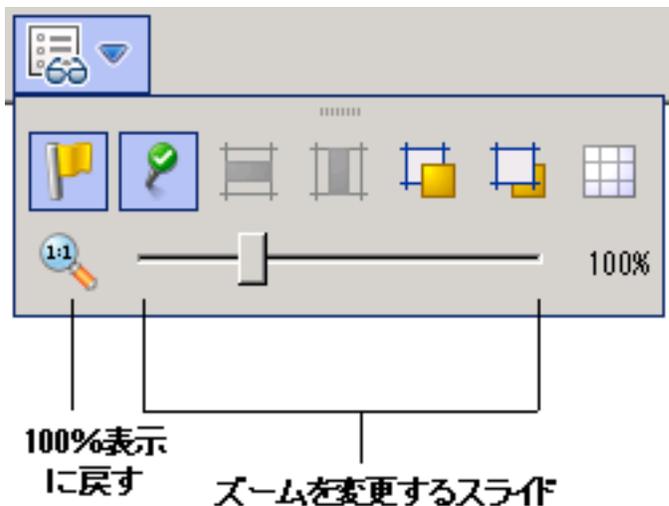
並列分岐ステップとマルチインスタンスステップの詳細については、「[高度な作成](#)」(282ページ)を参照してください。

表示オプションパレット

表示オプションパレットには、作成ペイン上のフローの外観を変更するためのアイコンがあります。作成ペインのツールバーから表示オプションボタン



をクリックして、**表示オプションパレット**を表示します。ズームバーを使用すれば、フローを 300% まで拡大できます。



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

表示オプションの使用の詳細については、「[フローの外観の調整](#)」(197ページ)を参照してください。

オブジェクトのプロパティシート

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

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

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

入力	結果	表示	説明	詳細	スクリプトレット		
xml	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	種類	割り当て元	未指定時	割り当て先
xpathQuery	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	xml	xpathQuery	ユーザーへのプロンプト(テキストから)	xpathQuery
delimiter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	delimiter	ユーザーへのプロンプト(テキストから)	delimiter	delimiter

[プロパティ]シートの使用の詳細については、「[入力の作成](#)」(205ページ)および「[オペレーションの出力の設定](#)」(243ページ)を参照してください。

ステップのインスペクター

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

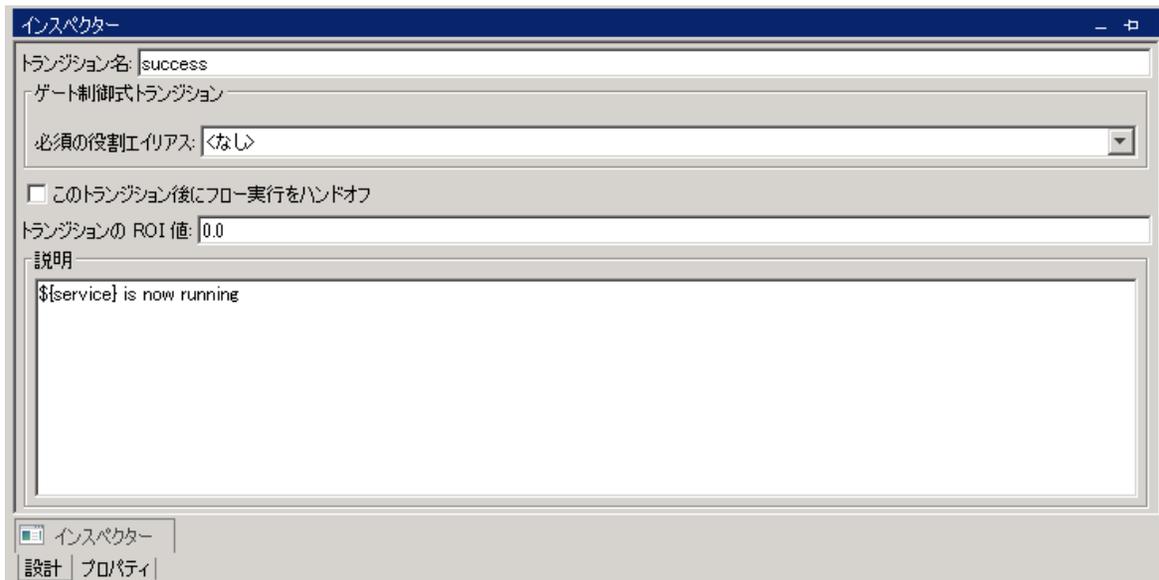
入力	結果	表示	説明	詳細	スクリプトレット		
targetHost	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	種類	割り当て元	未指定時	割り当て先
packetCount	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<未割り当て>	<未割り当て>	ユーザーへのプロンプト(テキストから)	<未割り当て>
packetSize	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<未割り当て>	<未割り当て>	定数の使用: 3	<未割り当て>
protocol	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<未割り当て>	<未割り当て>	定数の使用: 64	<未割り当て>
hostname	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<未割り当て>	<未割り当て>	定数の使用: ssh	<未割り当て>
username	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<未割り当て>	<未割り当て>	ユーザーへのプロンプト(テキストから)	<未割り当て>
password	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<未割り当て>	<未割り当て>	ユーザーへのプロンプト(テキストから)	<未割り当て>
keyFile	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<未割り当て>	<未割り当て>	ユーザーへのプロンプト(テキストから)	<未割り当て>
				<未割り当て>	<未割り当て>	定数の使用:	<未割り当て>

ステップのインスペクターの使用の詳細については、「[入力の作成](#)」(205ページ)を参照してください。

トランジションのインスペクター

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

を選択します。



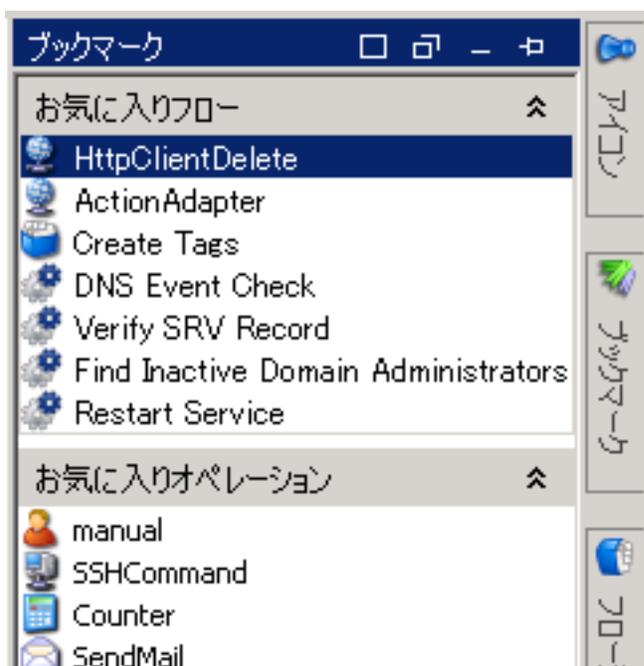
トランジションのインスペクターの使用の詳細については、[「トランジションの作成」\(228ページ\)](#)を参照してください。

[ブックマーク] ペイン

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

よく使用するフローやオペレーションをライブラリからドラッグして [ブックマーク] ペインに追加することが可能です。[ブックマーク] ペインから [プロジェクト] ペインにフローおよびオペレーションをドラッグして、プロジェクトにコピーすることもできます。

ブックマークの詳細については、[「フローおよびオペレーションのブックマーク」\(376ページ\)](#)を参照してください。



【アイコン】ペイン

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



[アイコン]ペインの使用の詳細については、[「フローの変更」\(200ページ\)](#)を参照してください。

[フロー変数] ペイン

[フロー変数]ペインはStudioウィンドウの右上にある[フロー変数]タブをクリックすると開き、フローで使用されている変数のリストと、どのように使用されているかが説明されています。

名前	#	✓	✓
altpass	8	✓	✓
フロー入力	3	✓	✓
フロー入力 "altpass" として宣言	-	✓	
"altpass" の値は入力 "altpass" に割り当てることができます	-	✓	✓
入力 "altpass" の値を "altpass" に割り当て	-	✓	✓
ユーザープロンプトを使用しないステップ入力	5		
"altpass" の値は "Change Service Status" の入力 "password" に割り当ててこ...	-		
"altpass" の値は "Does Service Exist" の入力 "altpass" に割り当てることがで...	-		
入力 "altpass" の値を "Does Service Exist" の "altpass" に割り当て	-		
"altpass" の値は "ServiceStatus" の入力 "password" に割り当てることができま...	-		
入力 "password" の値を "ServiceStatus" の "altpass" に割り当て	-		
altuser	8	✓	✓
フロー入力	3	✓	✓

[フロー変数] ペインの使用の詳細については、[「変数の使用」\(272ページ\)](#)を参照してください。

[参照] ペイン

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

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

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

[参照] ペインの使用の詳細については、[「フローとオペレーションの使用方法の詳細」\(366ページ\)](#)を参照してください。

[問題] ペイン

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

シヨンの問題が、場所と説明とともに表示されます。

種類	ソースの種類	名前	説明	場所
エラー	フロー	Flow	フローにリターンステップがありません	/Project 1/Library/Floder/Flow
エラー	フロー	Flow	フローに開始ステップがありません	/Project 1/Library/Floder/Flow

[問題] ペインの使用の詳細については、[「\[問題\] ペインでのフローの検査」 \(311ページ\)](#)を参照してください。

[検索] ペイン

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

ランク	名前	種類	パス	説明
****	SQL Script	operation	/Base [1.0.121]/Library/Operations/Databas...	データベースに対して SQL 複数行コマンドを実行...
****	SQLScript	operation	/MyProject 1/Library/My Ops Flows/oo-dotne...	<pre>Runs the requested SQL commands se...
****	SQLScript	operation	/MyProject 1/Library/My Ops Flows/oo-dotne...	<pre>Runs the requested SQL commands se...
****	SQLScript	operation	/MyProject 1/Library/My Ops Flows/oo-dotne...	<pre>Runs the requested SQL commands se...
****	SQL Server Soft Reset	flow	/Databases [1.0.115]/Library/Accelerator Pa...	MS SQL サービスを再起動し、MS SQL Server ...
****	Is An SQL Server	flow	/Databases [1.0.115]/Library/Operations/Dat...	このサンプルは、指定されたホストが Microsoft S...

[検索] ペインの使用の詳細については、[「フローまたはオペレーションの検索」 \(357ページ\)](#)を参照してください。

HP OO Studio の開始 – ワークフローの主なステップ

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



1. **新しいプロジェクトの作成** - 業務で使用するフロー、オペレーション、フォルダー、構成アイテムを含むプロジェクトを作成します。
[「プロジェクトの管理」\(48ページ\)](#)を参照してください。
2. **コンテンツパックのインポート** - 必要なコンテンツパックをインポートし、関連コンテンツをプロジェクトにコピーできるようにします。

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

- [「コンテンツパックのプロジェクトへのインポート」\(126ページ\)](#)を参照してください。
3. **フローの作成** - オペレーション、入力、トランジション、レスポンス、リターンステップを配置してフローを作成します。
[「フローの作成 – 手順」\(185ページ\)](#)および [「高度な作成」\(282ページ\)](#)を参照してください。
4. **フローの実行とデバッグ** - 作成したフローをデバッガーで検証します。
[「フローのテストとデバッグ」\(313ページ\)](#)を参照してください。
5. **コンテンツのリリースとコンテンツパックへのパッケージ** - プロジェクトを、フロー、オペレーション、操作、構成アイテムを含めコンテンツパックにパッケージし、HP OO Central へ昇格します。
[「コンテンツパックのエクスポート」\(336ページ\)](#)を参照してください。

HP OO Studio ウィンドウの外観の調整

HP OO Studio のペインを以下の状態に設定できます。

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

実行する操作

ペインを移動可能にする

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

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

ペインをドッキングする

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

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

ペインを最大化する

ペインを最大化して HP 00 ウィンドウ全体のサイズに拡大するには、**[最大化]**  ボタンをクリックします。

ペインを元のサイズに復元する

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

Studio ウィンドウの脇にペインを固定する

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

ペインの固定を解除する

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

ペインのサイズを調整する

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

Studio ウィンドウをデフォルトのレイアウトにリセットする

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

注: Studio ウィンドウをリセットする場合、新規プロジェクトの作成またはインポート後になります。

作成のベストプラクティス

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

• 一般的なベストプラクティス	25
• 共有コンテンツのベストプラクティス	26
• 命名のベストプラクティス	28
• フローのベストプラクティス	30
• オペレーションのベストプラクティス	31
• ステップのベストプラクティス	33
• トランジションのベストプラクティス	35
• 入力のベストプラクティス	35
• デバッグのベストプラクティス	36
• Studio の構成のベストプラクティス	36
• 説明のベストプラクティス	37
• ソースコントロール管理のベストプラクティス	44

一般的なベストプラクティス

フォルダー構造

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

Studio 内での名前の変更

プロジェクト、フロー、またはほかの HP OO エンティティの名前を変更する場合は、Studio で行います。ファイルブラウザでエンティティの名前を変更しないでください。

ベストプラクティスのドキュメント

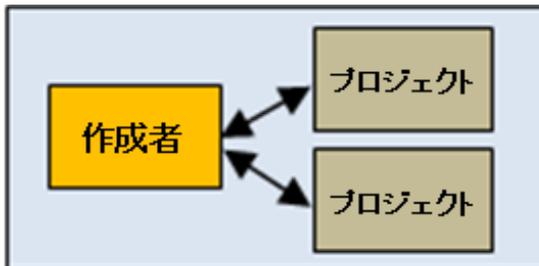
命名規則、フォルダー構造、フローの作成者が従うべきその他のガイドラインを記述するドキュメントを作成します。例については、「[命名のベストプラクティス](#)」(28ページ)を参照してください。

共有コンテンツのベストプラクティス

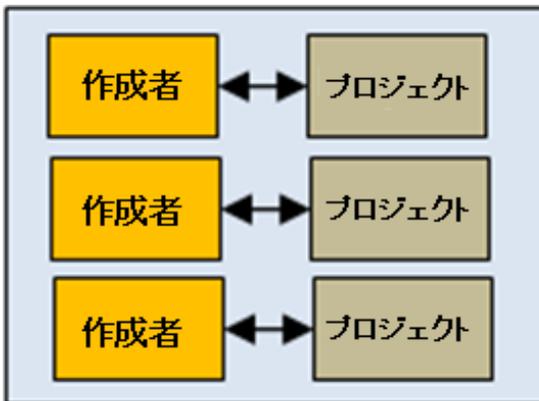
環境の識別

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

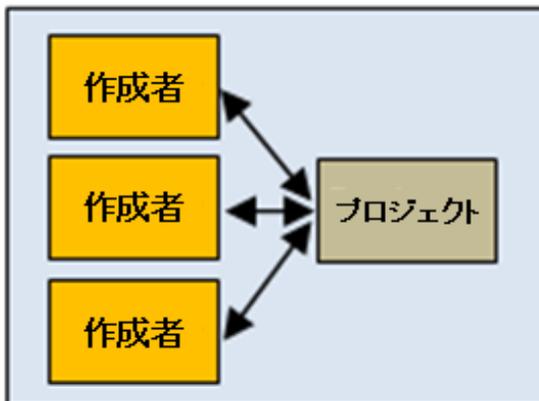
- 1人の作成者が1つまたは複数のプロジェクトで作業する



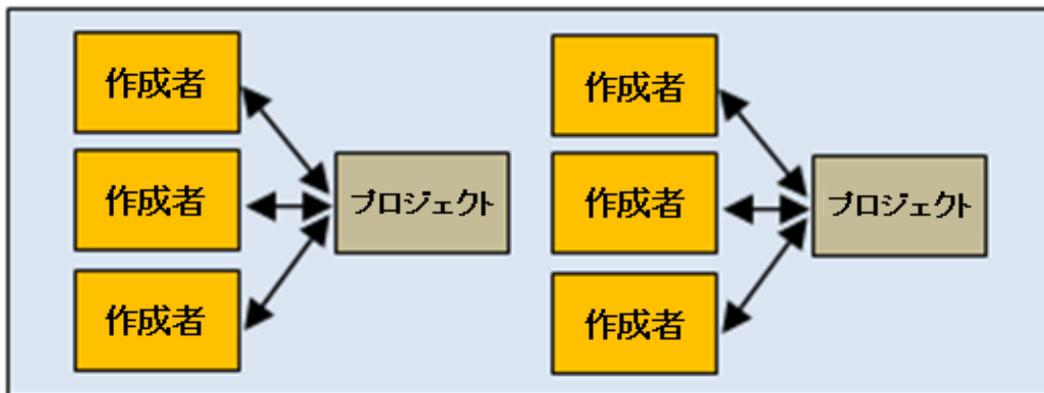
- 複数のプロジェクトが存在し、各作成者が1つのプロジェクトで作業する



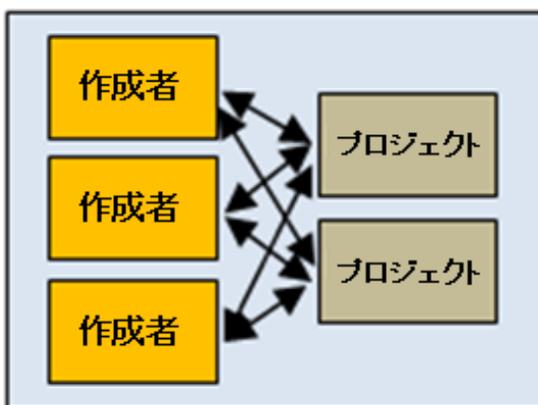
- 1つのプロジェクトで複数の作成者が作業する



- 複数のチームが存在し、各チームが1つのプロジェクトで作業する



- 複数の作成者が複数のプロジェクトを同時に進める



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

コミュニティのやり取り

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

作成時のコミュニティのやり取りには次のようなものが考えられます。

1. コンテンツをコミュニティからダウンロードする

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

詳細については、「[コンテンツパックのプロジェクトへのインポート](#)」(126ページ)を参照してください。

2. コミュニティで問題について話し合う

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

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

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

3. コンテンツをコミュニティに提供する

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

命名のベストプラクティス

フローの種類

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

命名規則: 大文字と小文字

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

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

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

フローの種類別の命名規則: 接頭辞

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

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

変数別の命名規則: 接頭辞

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

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

直感的な名前

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

単語の使用

一部の入力名は、多くのオペレーションやステップで共通で使用します。次の入力名は、HP OO コンテンツで使用されています。

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

その他の共通の入力名には、次のようなものがあります。

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

フローのベストプラクティス

フローの設計

作成するフローの構造を、構築を始める前に設計します。

フローはシンプルに、1 画面に格納

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

フローの再利用

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

変更の前にフロー/オペレーションが使用されている箇所のチェック

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

変更の前にフローをコピー

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

一貫性を持たせる

異なるフローを、同じ開始および終了位置を使用して設計します。

オペレーションのベストプラクティス

オペレーションを変更するときの注意

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

オペレーションの変更が必要になりそうな場合はディープコピー

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

ディープコピー用のフォルダーの作成

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

カスタマイズしない場合は元のオペレーションを使用

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

統合のために元の名前とメッセージを保持

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

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

封印されているオペレーションではなくステップをコピー

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

値をフロー変数に割り当てる

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

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

複数のオペレーションで同じコマンドを実行するのを避ける

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

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

スクリプトレットではなく結果フィルターを使用

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

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

必要なレスポンスのみを使用

ほとんどのオペレーションには、成功と失敗という、2つのレスポンスのみがあります。レスポンスの数を少なくすると、フローの作成が簡単になり、分かりやすくなります。異なるタイプの失敗を基

にした複数のレスポンスは、明らかに異なるパスに進む場合や、結果がその状況のために失敗にしかならない場合 (HTTP Get のリダイレクトレスポンスなど) でのみ使用してください。

ただし、この原理は、意味がない場合には当てはまりません。たとえば、データを取得してしきい値をチェックするオペレーションでは、失敗、しきい値越え、しきい値未満の3つのレスポンスが必要な場合があります (どれも成功ではないレスポンス)。

デフォルトレスポンスは失敗

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

ステップのベストプラクティス

説明なし

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

コールアウト

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

開始ステップ

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

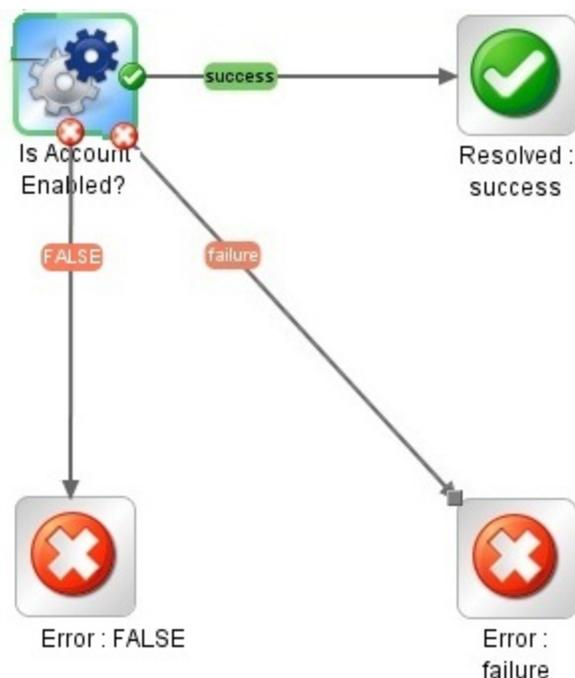
- 開始ステップに複数のレスポンスがあり、それぞれが別のステップに進む場合。
- 開始ステップを左上隅に配置すると、トランジションが交差するなどして表示が複雑になりすぎる場合。

リターンステップの名前の変更

フロー内の同じタイプのリターンステップが複数ある場合、リターンステップの名前を変更して原因を含めます。たとえば、「エラー: 失敗」や「エラー: しきい値に到達していない」などになります。

失敗と否定の結果とを区別する

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



操作なしと情報の収集の成功とを区別する

次の状態を混同しないようにしてください。

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

結果を使用して値をフロー変数に割り当てる

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

フロー変数への注意

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

不要な結果を追加しない

ステップのベースとなるオペレーションまたはフローでは、複数の出力を提供することができます。ただし、ステップの結果を追加するときに、フローに必要な出力のみを使用していることを確認しま

す。結果が多すぎると、不要なデータによりフロー速度が低下し、パフォーマンスに影響を与えません。

エラーコードを取得する結果の作成

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

トランジションのベストプラクティス

トランジションの整理と簡潔さの保持

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

グリッドの使用

グリッドを有効にして、ステップの整列を保持します。

入力のベストプラクティス

不要な入力の削除

ステップ内のオプションの入力が必須でない場合、これらの入力を削除します。

順序ルールに従った入力の追加

順序ルールに従って、一貫した順序で入力を追加します。例:

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

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

```
input_a="The first input"
input_b=${input_a}+
```

input_a を **input_b** の前に定義してください。

データをフロー入りに割り当てる

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

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

ローカライズされたデータ

(フランス語、日本語、中国語、ドイツ語、スペイン語などの言語の特殊文字を含む) 入力として送信されるローカライズされたファイルの場合は、Studio で UTF-8 ファイルエンコードが使用されていることに注意してください。その他のファイルエンコードは認識されない可能性があります。このようなファイルには、UTF-8 エンコードを使用することをお勧めします。

デバッグのベストプラクティス

サブフローのデバッグ

親フローをデバッグする前に、サブフローをデバッグすることをお勧めします。

Studio の構成のベストプラクティス

システムプロパティ

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

構成アイテム

構成アイテム(システムフィルター、スクリプトレット、選択リストなど)を削除する前に、**[これを
使用する項目は?]**機能を使用して、ほかのアイテムが依存していないかをチェックすることをお勧めします。詳細については、「[フローとオペレーションの使用法の詳細](#)」(366ページ)を参照してください。

構成アイテムの名前の重複はエラーとして検出され、**[問題]**ペインのそのアイテムの上にツールチップとして表示されます。重複が発生した場合には、1つのアイテムの名前を変更するか、削除する必要があります。

同じUUIDを持つ構成アイテムがある場合もあります。たとえば、複数のプロジェクトをリファクタリングして、最後に既存のプロジェクトと同じアイテムを含むコンテンツパックをワークスペースにインポートする場合です。もう1つの例は、00 9.xから00 10.00または10.01(バージョン10.02より前)にリポジトリを移行し、結果をプロジェクトまたはコンテンツパックとしてインポートした後で、00 Base コンテンツパックをインポートした場合です。

UUIDが重複すると、Studioの動作が一貫しなくなる可能性があります。重複するアイテムの1つを削除し、結果として壊れた可能性のあるフローをすべて修正することによって、この問題を修正することを強くお勧めします。

プロジェクトとコンテンツパックを含め、ワークスペース全体の名前とUUIDの両方の重複が検出されます。

注: 重複名には、大文字と小文字の区別がないものと見なされます。たとえば、MySystemAccount は mysystemaccount と重複していることとなります。

説明のベストプラクティス

すべての説明

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

フォルダーの説明

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

フローの説明

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

このフローは以下の場合にのみ動作します。

- Windows 2003 またはそれ以降
- Windows Telnet Service が有効である場合

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

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

オペレーションの説明

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

オペレーションの内容の説明。

入力:

入力 1 - この入力の詳細

入力 2 - この入力の詳細

入力 3 - この入力の詳細

レスポンス:

レスポンス 1 - このレスポンスの詳細

レスポンス 2 - このレスポンスの詳細

結果:

フロー/オペレーションのプライマリ結果

その他の結果:

結果 1 - 最初の追加の結果

結果 2 - 2 番目の追加の結果

入力の説明

- 必要な入力に対する意味がある説明を含めます。
- 入力に必要なサンプルデータを正しい形式で含めます。
- オプション入力と必須入力を区別します。
- 説明で列記する入力の順番が、[入力] タブに表示される順番と同じであることを確認してください。

- 必要な場合、入力にとるデータの構文を記述します。
- 説明中のセクション見出しの構文は同じものを使用します(たとえば、“Example” や “ex:” ではなく “Examples” を使用するなど。英語の場合)。
 - セクション見出しの形式は、「値の形式」、「例」、「デフォルト値」、および「有効な値」です。これらは、該当する情報があり、使用可能な場合はいつでも必要です。
 - 「値の形式」、「例」、「デフォルト値」、および「有効な値」からの値で、句読点のみがふくまれる値は、アポストロフィで囲む(、')必要があります。
- 各入力名の前に4つスペースを置きます。
- 入力の説明のベースとして、次のテンプレートを使用してください。

入力:

```
inputName - 最初の入力の名前
```

```
Value format: domain\user
```

```
Valid values: us.east.1a, us.east.1b
```

```
Default value: us.east.1b, ','
```

```
Examples: valueString, 31241423, one, two, three
```

```
secondInputName - 2番目の入力の名前
```

```
Value format: text
```

```
Valid values: one, two, three
```

```
Default value: one
```

例:

トランジションの説明

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

出力の説明

- 説明で列記する出力の順番が、[出力]タブに表示される順番と同じであることを確認してください。
- 各出力名の前に4つスペースを置きます。

- オペレーションが実行可能な状況を制限するオペレーションの環境的な制限がある場合は含めません。

結果の説明

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

結果:

```
returnResult - これはプライマリ出力です。
```

```
firstResult - 次の列を持つテーブル内の EC2 インスタンス:(以下にテーブル形式の結果の例を表示します。)
```

```
Instance Id
```

```
AMI ID Machine type
```

```
Status - 可能な値: 「パブリックDNS」、「キーの組の名前」、「RAM ディスクID」
```

レスポンスの説明

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

レスポンス:

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

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

```
no more values -
```

注の説明

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

前提条件：

1. 最初の前提条件

最初の前提条件の説明

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

2.2 番目の前提条件

2 番目の前提条件の説明

その他の設定：

1. abc
2. def

トラブルシューティング:

1. abc
2. def

その他:

1. abc
2. def

データベースの可用性の取得グループを完全に説明する例

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

入力:

- host - 接続先の Exchange 2010 サーバーホスト。
- username - サーバーへの接続時に使用するユーザー名。
- password - サーバーへの接続時に使用するパスワード。
- authType - ユーザーの資格情報の認証に使用されるメカニズムを指定します。

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

デフォルト値: Default。

- dagName - 停止する DAG の名前。

例: oodag。

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

デフォルト値: ','。

結果:

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

区切り文字で区切られます。

- servers - DAG のメンバーであるサーバーのリスト。
- operationalServers - DAG からの動作中サーバーのリスト。

primaryActiveManager - クラスターコアリソースグループを所有するノードを表します。
distinguishedName - Active Directory からの DAG の DN。
isValid - DAG が有効かどうか。

レスポンス:

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

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

クラスターが使用不可であり、クラスターサービスが動作していません。

注:

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

2. サポートされているバージョン: 2010.2010

ソースコントロール管理のベストプラクティス

Subversion

- ほかの作成者との競合を防ぐため、ファイルで作業する前に必ずファイルをロックします。作業を開始してからコミットする前にファイルをロックするのではなく、ファイルをロックしてから編集するようにします。ファイルで作業している間に、ほかの作成者がファイルの編集を開始している可能性があります。一部のソースコントロール管理ツールでは、ファイルをロックしようとするまで、そのファイルがロックされていることが表示されません。ロックポリシーの強制を使用すると、アイテムをロックしない限り、フロー作成者はアイテム(フローまたは構成アイテム)を変更できません。これにより、アイテムを編集できる作成者を1人に限定することが可能になります。詳細については、「[ロックポリシーの強制](#)」(66ページ)を参照してください。
- 最新のリリースで常に作業するには、アイテムを更新してから、アイテムをロック、変更、またはコミットするようにします。
- @Action はプロジェクトに格納されません。作成者の Studio リポジトリに格納されて、これらのアクションを使用するオペレーションとの間でリンクされます。別の作成者が作成した @Action を使用するには、その Jar/DLL ファイルを自身の Studio リポジトリに手動でコピーする必要があります。@Action を作成したら、ほかの作成者がインポートできるように、プラグインが含まれたコンテンツパックを作成することをお勧めします。

Git

- Git を使用する前に、現在のワークスペースにコンテンツパックがないことを確認してください。
- 独自の HTTP ベースの Git ソリューションをセットアップする場合は、モジュールを使用するか、[Github](#)、[GitLab](#)、[Gitblit](#)、[Bonobo Git Server](#) などの専用の Git サーバーを選択してください。これらのすべてサーバーに、リポジトリコンテンツの Web 表示、履歴、分岐、詳細なアクセス制御といった利点があります。
- Apache を Git サーバーとして使用する場合は、Apache 用の専用の Git サーバーモジュール [git-http-backend](#) を使用することをお勧めします。

注: WebDAV を基にした HTTP(S) で Git サーバーをセットアップすることはお勧めしません。

HP OO Studio での各言語の使用 - ローカライズ

ローカライズとは、特定の環境 (国または地域) に合わせてソフトウェアを修正することを指します。HP Studio は、次の言語へのローカライズが可能です。

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

Studio では、次のテキスト文字列がローカライズされます。

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

コンテンツパックのローカライズ

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

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

地域のパラメーターはオプションです。

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

プロジェクトのローカライズ

プロジェクトはコンテンツパックと同じ方法でローカライズされません。プロジェクトのデフォルトの言語のみを編集できます。これは、プロジェクトをコンテンツパックにパッケージする際にデフォルトファイル (cp.properties) として保存されます。

次の2つの点に注意してください。

- プロジェクトのテキストは実際にローカライズされるわけではありません。フロー、オペレーション、および構成アイテムの場合、テキストがXMLファイル内に保存され、フォルダーの場合、説明が複数の .properties ファイル (フォルダーごとに1つ) に保存されます。コンテンツパックを作成するときには、テキストは常に同じ cp.properties ファイルに書き込まれます。
- コンテンツパックを展開し、プロジェクトとして Studio にロードすると、デフォルトファイルのみが使用され、ローカライズしたプロパティファイルは使用されません。ファイルが存在しない場合は、すべてのテキストは空白になります。

cp.properties ファイル

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

注: コンテンツパックの構成を変更した後、Studio を再起動してください。

プロジェクトを作成または保存すると、**cp.properties** ファイルは **\resource-bundles\cp.properties** にあるプロジェクトファイルシステムに保存されます。

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

実行する操作

Studio での現在のロケールの変更

デフォルトでは、Studio はコンピューターのデフォルトの言語でインストールされます。

現在のロケールをオーバーライドするには、構成を変更します。

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

言語のロケールコード (<code>user.language</code>)	地域のロケールコード (<code>user.region</code>)	言語
fr	FR	フランス語
de	DE	ドイツ語
ja	JP	日本語
es	ES	スペイン語
zh	cn	中国語

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

Studio の表示言語の設定

Studio は `user.language` プロパティから表示言語を取得します。このプロパティが有効になっている必要があります (`user.region` も有効になっている必要があります)。このプロパティが無効な場合、Studio はシステムロケールを使用します。

現在 Studio でサポートされている表示言語は次のとおりです。

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

さらに、コンテンツパックのローカライズが可能です。詳細については、[「コンテンツパックのローカライズ」 \(45ページ\)](#)を参照してください。

プロジェクトの使用

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

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

プロジェクトの管理



HP 00 Studio でフローの作成を開始するには、プロジェクトで作業している必要があります。

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

Studio 10.x では、プロジェクトは作成者のファイルシステムにローカルに格納されます。ソースコントロール管理機能を使用するには、外部のソースコントロール管理ツールを使用する必要があります。ソースコントロール管理の詳細については、「[ソースコントロールの使用](#)」を参照してください。

実行する操作

.oo フォルダーのデフォルトの場所を変更する

.oo フォルダーには、インポートしたコンテンツパック、使用中のプロジェクト、および他の設定に関する情報が含まれています。デフォルトでは、Studio の .oo フォルダーは Users/<ユーザー名> フォルダーの下にあります。

.oo フォルダーの場所を変更する

1. Studio のインストールフォルダーにある **Studio.l4j.ini** ファイルを開きます。
2. 次のプロパティで場所を変更/追加します。

```
-Duser.home="C:/OO_Home"
```

プロジェクトを作成する

1. [ファイル] > [プロジェクトの新規作成] を選択します。

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

2. [名前] ボックスにプロジェクトの名前を入力します。
3. [プロジェクトの作成] ダイアログボックスで、[場所] ボックス内のパスが前のタスクで設定した **Workspace** フォルダーへの正しいパスになっていることを確認します。パスが **Workspace** フォルダーへの正しいパスでない場合、**Workspace** フォルダーを参照して選択します。
4. [OK] をクリックします。
5. ソースコントロール管理ツールを使用している場合、共有リポジトリにプロジェクトフォルダーを追加します。詳細については、「[HP 00 Studio のソースコントロールの使用](#)」(57ページ)を参照してください。

プロジェクトをインポートする

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

ソースコントロール管理ツールを使用している場合、プロジェクトフォルダーをチェックアウトして、ローカルの作業コピーを作成してから、プロジェクトを Studio にインポートすることが必要になる場合があります。詳細については、「[HP 00 Studio のソースコントロールの使用](#)」(57ページ)を参照してください。

1. [ファイル] > [プロジェクトのインポート] を選択します。

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

2. [プロジェクトディレクトリの選択] ダイアログボックスで、インポートするプロジェクトを参照して見つけます。Studio では複数のプロジェクトを一度にインポートできます。
3. [OK] をクリックします。プロジェクトが Studio の [プロジェクト] ペインに表示されます。

開いたプロジェクトを閉じる

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

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

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

閉じたプロジェクトを開く

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

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

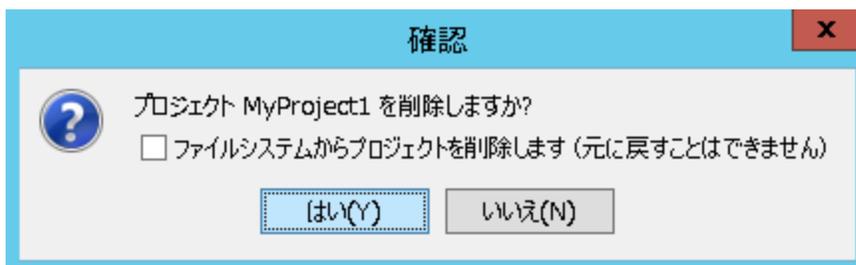
注: [プロジェクト] ペインで [開く]  ボタンをクリックすることもできます。

プロジェクトを削除する

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

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

確認ダイアログが開きます。



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

3. チェックボックスをオンにして、Studio ワークスペースとファイルシステムからプロジェクトを削除します。チェックボックスをオンにしない場合は、プロジェクトは Studio ワークスペースから削除されますが、ファイルシステムからは削除されません。必要な場合は、後でプロジェクトを再インポートすることができます。
4. 確認のダイアログボックスで [はい] をクリックします。

フロー、オペレーション、構成アイテムを移動する

フロー、オペレーション、構成アイテムは、プロジェクト内の別のフォルダーまたは新しいプロジェクトに移動することができます。

注: フローおよびオペレーションを構成アイテムと同時にドラッグアンドドロップすることはできません。フローおよびオペレーションを [Configuration] フォルダーにドラッグアンドドロップしたり、またはこのフォルダーからドラッグアンドドロップすることはできません。

1. [プロジェクト] ペインで、移動するフローまたはオペレーションを選択します。Shift キーまたは Ctrl キーを使用して、複数のアイテムを選択することができます。
2. アイテムを別のフォルダーまたは別のプロジェクトにドラッグアンドドロップするか切り取って貼り付けます。

コンテンツパックのフローとオペレーションをプロジェクトにコピーする

[依存関係] ペインのフローまたはオペレーションは読み取り専用です。これらのフローとオペレーションの編集可能なコピーを作成するには、これらをプロジェクトにコピーします。

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

ヒント: **[依存関係]** ペインから複数のアイテムを素早くコピーするには、アイテムを選択し、Shift キーまたは Ctrl キーを押しながらプロジェクトにドラッグアンドドロップします。

注: フローを構成しているオペレーションもプロジェクトにコピーするには、**[編集]** > **[ディープコピー]** を選択します。詳細については、「[フローおよびオペレーションのコピー](#)」(363ページ)を参照してください。

プロジェクトのプロパティを表示する

- **[プロジェクト]** ペインで **[プロパティ]** オプションをダブルクリックします。
または:
プロジェクトを右クリックして **[プロパティ]** を選択します。
または:
[プロジェクト] ペインの **[プロパティ]** ノードでプロジェクトを右クリックし、**[開く]** を選びます。

[プロパティ] ウィンドウにプロジェクトの情報が表示されます。プロジェクトの **[プロパティ]** ウィンドウを使用して、**パブリッシャーのバージョン**、プロジェクトの**説明**、プロジェクトの**依存関係**を設定できます。

[プロジェクト] ペイン (および **[SCM 変更]** ペイン) の **[プロパティ]** ノードから標準の SCM 操作を実行することもできます。プロジェクトの追加、コミット、古いバージョンに戻す操作、プロパティの履歴の更新を行うことができます。

つまり、プロジェクトの変更全体をコミットせずに、プロジェクトのメタデータの変更 (パブリッシャー、バージョン、または説明の変更) のみをコミットできます。同様に、プロジェクトの変更全体をコミットせずに、依存関係の変更をコミットできます。

これらのオプションの詳細については、「[Subversion ソースコントロール管理の使用](#)」(60ページ)または「[Git ソース管理システムの使用](#)」(84ページ)を参照してください。

Windows エクスプローラーでのオブジェクトの表示

1. **[プロジェクト]** ペインで **[プロパティ]** オプションを選択します。
2. 右クリックしてメニューから **[エクスプローラーで表示]** を選択します。

Windows エクスプローラーが開き、プロジェクトファイルが含まれるフォルダーが表示されます。

プロジェクト内のオブジェクトのプロパティを表示する

- **[プロジェクト]** ペインで、フロー、オペレーション、またはその他のオブジェクトをダブルクリッ

クします。オブジェクトの[プロパティ]ウィンドウが開きます。

- フローが作成キャンバスで開いている場合、フローの[プロパティ]ウィンドウを表示するには、作成キャンバスの左下にある[プロパティ]タブをクリックします。

参考資料

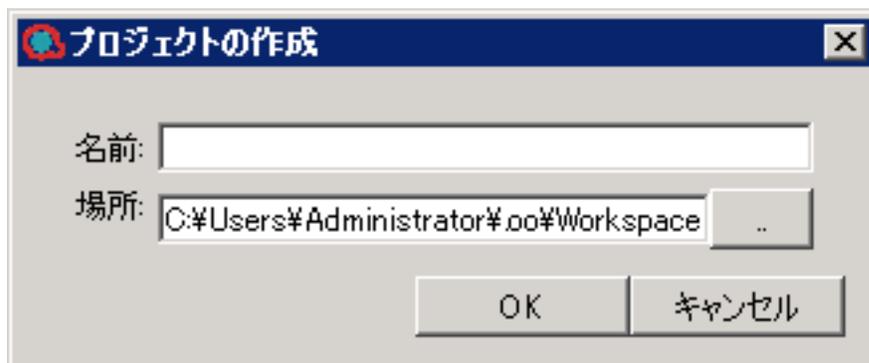
[ファイル] メニュー

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

メニューアイテム	説明
プロジェクトの新規作成	新しいプロジェクトを作成します。
新規作成	新しいファイル、フォルダー、またはオペレーションを作成します。
プロジェクトのインポート	別のワークスペースの既存のプロジェクトを参照してインポートします。
プラグインのインポート	新しいオペレーションのベースとして使用する、アクションプラグインを参照してインポートします。 「オペレーションの作成」(349ページ) を参照してください。

保存	選択された要素を保存します。
すべて保存	更新されたすべての要素を保存します。
閉じる	現在選択されているフローまたは[プロパティ]ウィンドウを閉じます。
すべて閉じる	開いているフローまたは[プロパティ]ウィンドウをすべて閉じます。
最近開いた項目	最近開いた項目を一覧表示します。これには、作成キャンバスで開かれたフローとオペレーション、最近表示された[プロパティ]ウィンドウが含まれます。
コンテンツパックの作成	選択したプロジェクトのコンテンツパックを作成します。
コンテンツパックのインポート	コンテンツパックを参照して.jar形式でインポートします。 「コンテンツパックのプロジェクトへのインポート」(126ページ) を参照してください。
プロジェクトを閉じる	現在選択されているプロジェクトを閉じて、グレー表示にします。
プロジェクトを開く	現在選択されている、閉じたプロジェクトを開きます。
プロジェクトの削除	選択したプロジェクトをワークスペースから削除します。
すべてのプロジェクトのリフレッシュ	[プロジェクト]ペイン内のすべてのプロジェクトを更新します。
終了	HP 00 を終了します。

[プロジェクトの作成] ダイアログボックス



GUI アイテム	説明
----------	----

名前	新しいプロジェクトの名前を入力します。
場所	<p>新しいプロジェクトを格納する、ワークスペースの場所を入力します。</p> <ul style="list-style-type: none"> 場所へのパスを入力します。 場所を参照するには、参照ボタン  をクリックします。

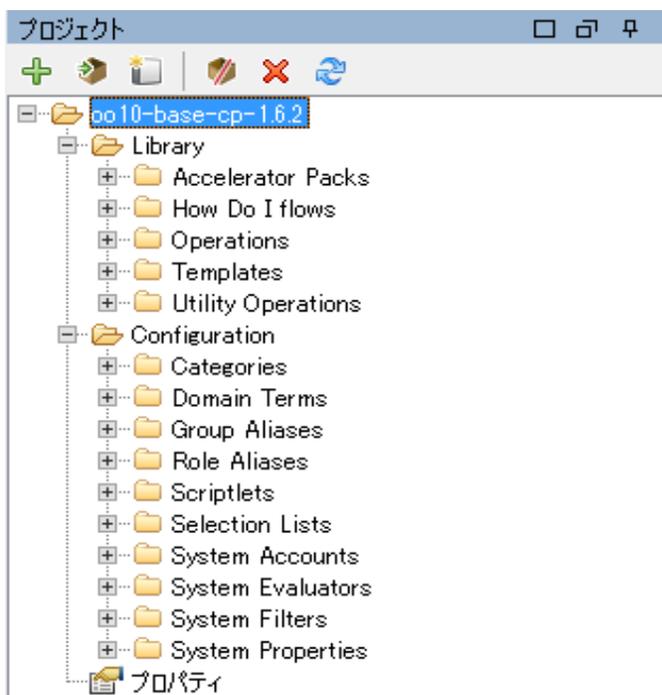
[プロジェクト] ペイン

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

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

注: 既存の構成アイテム構造の下のすべての構成アイテム内にフォルダーを作成できます。

- プロジェクトのプロパティを表示します。



GUI アイテム

説明

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

[プロジェクト] ペインを使用したフォルダの管理

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

実行する操作

フォルダを作成する

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

重要: プロジェクト内に要素 (フォルダ、フロー、オペレーション、または構成アイテム) を作成する必要がある場合は、ファイルシステムでアイテムを作成するのではなく、Studio 内から実行してください。

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

フォルダ名を変更する

プロジェクトツリーでフォルダを右クリックし、[名前の変更] を選択します。

重要: プロジェクト内の要素 (フォルダー、フロー、オペレーション、または構成アイテム) の名前を変更する必要がある場合は、ファイルシステムでアイテムの名前を変更するのではなく、Studio 内から実行してください。

フォルダーを削除する

プロジェクトツリーでフォルダーを右クリックし、**[削除]** を選択します。

重要: プロジェクト内の要素 (フォルダー、フロー、オペレーション、または構成アイテム) を削除する必要がある場合は、ファイルシステムでアイテムを削除するのではなく、Studio 内から実行してください。

フォルダーをコピーして貼り付ける

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

サブフォルダーにドリルダウンする

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

フォルダーの説明を表示する

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

HP 00 Studio のソースコントロールの使用

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

HP 00 Studio のソースコントロールとは

HP 00 Studio では、2つの組み込みのソースコントロール管理ツールが提供されており、作成者はローカル環境で作業した後で変更をパブリックバージョンと同期させることができます。

- **Subversion:** 詳細については、「[Subversion ソースコントロール管理の使用](#)」(60ページ)を参照してください。
- **Git:** 詳細については、「[Git ソース管理システムの使用](#)」(84ページ)を参照してください。

注: これらのツールは同時に使用できないので、どちらか1つのツールのみを選択する必要があります。使用するツールがわからない場合は、システム管理者に問い合わせてください。

実行する操作

ソースコントロールツールを選択する

ソースコントロールツールを選択するには:

1. **[SCM]** > **[接続]** を選択します。
2. **[SCM 接続]** ダイアログボックスで、SCM の種類として**[Subversion]** または **[Git]** を選択します。

参考資料

SCM 接続

The screenshot shows a dialog box titled "SCM 接続" (SCM Connection). It has a blue header bar with a close button (X) on the right. The main area is divided into several sections:

- 種類 (Type):** Radio buttons for "Subversion" and "Git". "Git" is selected. A green callout box points to the "クローン" button with the text: "このボタンはリポジトリのチェックアウトに使用します" (This button is used for checking out the repository).
- SCM リポジトリ (SCM Repository):** A text field for "URL" containing "https://<SCM-path>/trunk" and a "クローン" (Clone) button.
- 認証 (Authentication):** A dropdown menu for "Type" set to "資格情報" (Credentials). Below are text fields for "ユーザー名" (Username) with "admin" and "パスワード" (Password).
- ユーザー情報 (User Information):** Text fields for "フルネーム" (Full Name) with "fg" and "電子メールアドレス" (Email Address) with "another-author@hp.com".
- Important:** A warning icon and text: "In order to use the Git functionality, you must install the Git client from [こちら](#) をご覧ください。" (Please see [here](#) for more information).

At the bottom, there are three buttons: "OK", "キャンセル" (Cancel), and "適用" (Apply).

メニューアイテム	説明
種類	使用するソースコントロール管理ツールとして[Subversion]または[Git]を選択します。
SCM リポジトリ: URL	Subversion または Git リポジトリの URL を入力します。 注: URL では HTTP および HTTPS プロトコルを使用できます。

<p>クローン/接続解除</p>	<p>Git のみ:</p> <p>ローカルマシンの現在のワークスペースのディレクトリで、メイン Git リポジトリから作業コピーにクローンを作成します。</p> <p>クローンを作成した後で、Git パスが無効になり、このボタンが [接続解除] に変わります。</p>
<p>接続解除</p>	<p>メイン Git リポジトリとローカルディレクトリの接続を解除します。複数の Git リポジトリを使用していて、異なるリポジトリに切り替える場合に [接続解除] オプションを使用します。</p> <p>[接続解除] を選択すると、次のメッセージが表示されます。</p> <p>Git から接続解除する前に、すべての変更をコミットし、メイン Git リポジトリにプッシュしていることを確認してください。後でもう一度 Git リポジトリのクローンを作成し、変更を取得することができます。</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>注: 接続解除すると、Git コマンドを実行できなくなります。</p> </div>
<p>チェックアウト/接続解除</p>	<p>Subversion のみ:</p> <p>ローカルマシンの現在のワークスペースのディレクトリ内で SVN リポジトリから作業コピーにソースをコピーします。</p> <p>チェックアウトした後で、SVN パスが無効になり、このボタンが [接続解除] に変わります。</p>
<p>サーバー証明書を信頼</p>	<p>セキュリティで保護されたサーバー (SSL/SSH) を使用している場合は、このチェックボックスを選択します。選択しないと、サーバーにアクセスできません。</p>
<p>資格情報: Windows 認証の使用</p>	<p>オンにした場合、現在ログインしているユーザーを使用して認証を実行します。</p> <p>異なるユーザーで認証を実行するには、このフィールドをオフにし、下のフィールドにユーザー名とパスワードを入力します。</p>
<p>ユーザー名</p>	<p>認証のための資格情報として使用するユーザー名とパスワードを入力します。</p>
<p>パスワード</p>	

Subversion ソースコントロール管理の使用

用語集

作業コピー

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

チェックアウト

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

コミット: リポジトリへの変更の保存

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

追加: SVN リポジトリへの新しいファイルの追加

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

削除: リポジトリからのファイルの削除

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

移動: ファイルまたはディレクトリの名前変更

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

更新: 作業コピーの更新

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

注: ソースコントロール管理ツールには、日付、時刻、コメントなどのフローの履歴データは含まれません。

初期のソースコントロールリポジトリの作成

実稼働環境では、http、https、またはsvnといったプロトコルでアクセスされる専用のSVNサーバーを使用することをお勧めします。

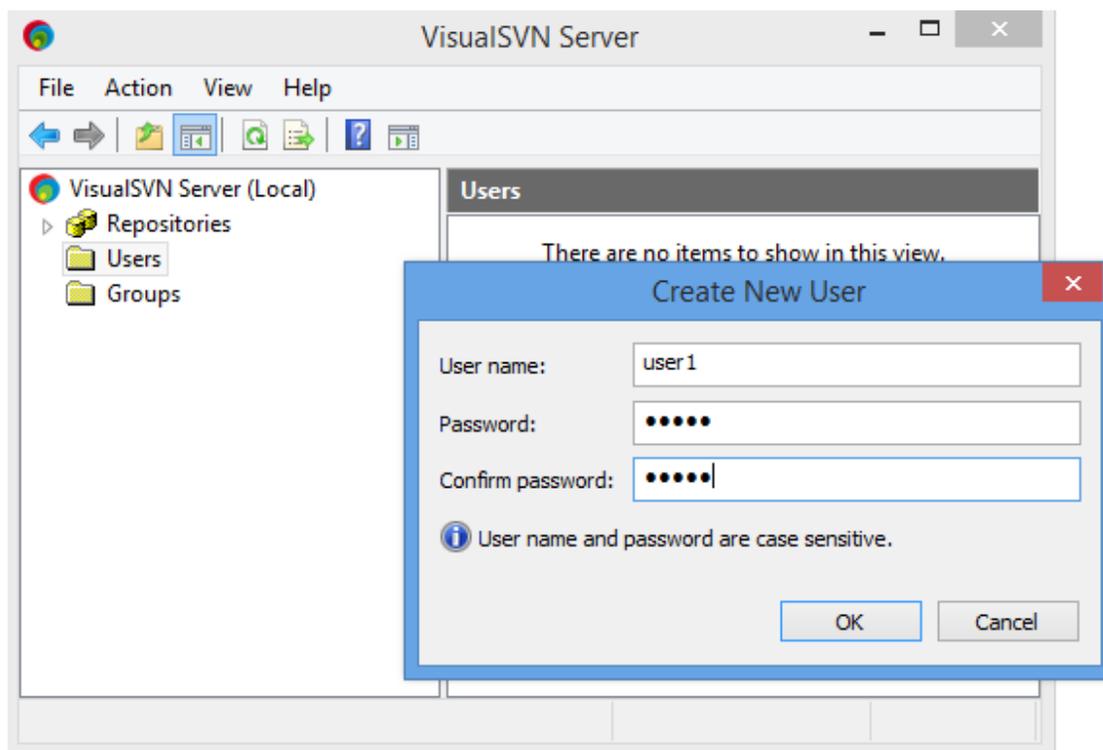
注: 各オペレーティングシステム用のSVNディストリビューションのリストについては、サブバージョンのWebサイト (<http://subversion.apache.org/packages.html>) にアクセスしてください。

例 1: 複数の作成者のシナリオでの VisualSVN の使用

次の例では、複数の作成者のシナリオで、VisualSVNサーバーを構成および使用方法について説明します。

SVNの使用を開始する前に、次のリンクを利用してその概念を詳しく理解することをお勧めします。
<https://subversion.apache.org/docs/>.

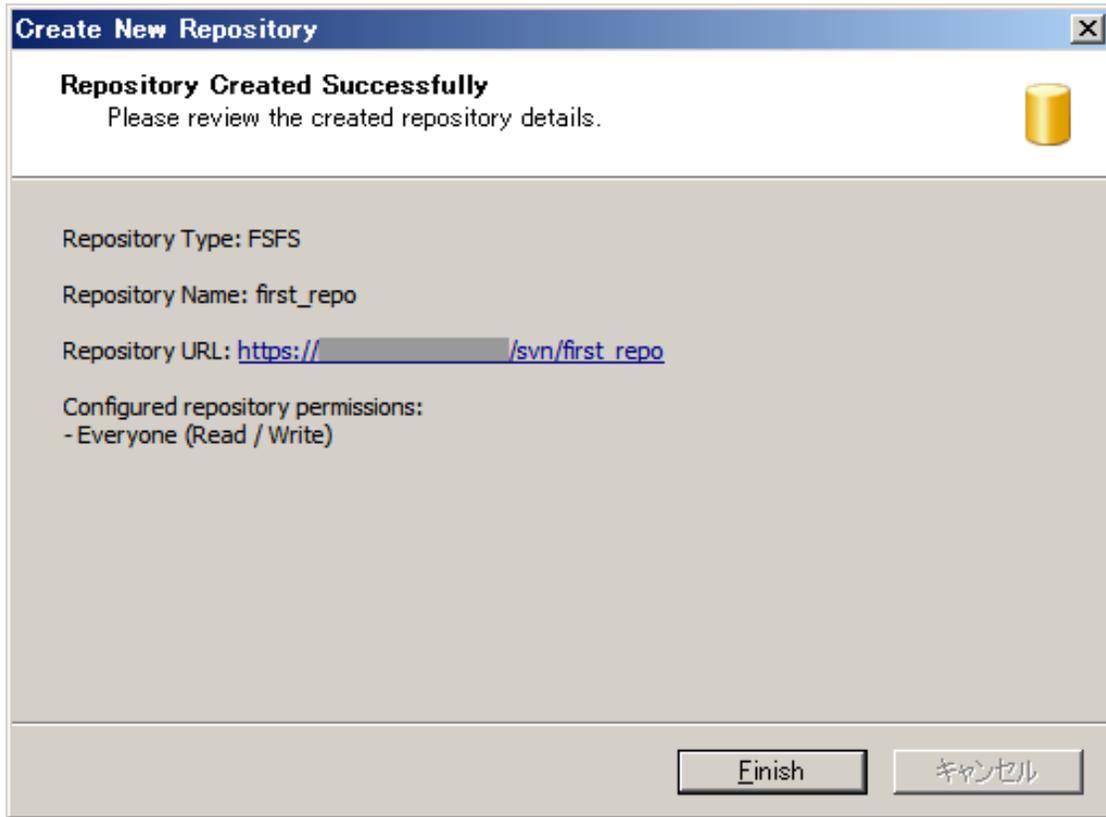
1. 1.8.x SVN 互換バージョンの VisualSVN をダウンロードします。
2. 新しいSVNサーバーユーザーを構成するには、右クリックして **[Users]** > **[Create User]** を選択します。



注: すべての Studio ペルソナは、Visual SVN ユーザーによって識別されます。

3. SVN リポジトリを作成するには、Repository フォルダーを右クリックし、**[Create New**

Repository] を選択して、空のリポジトリを作成します。



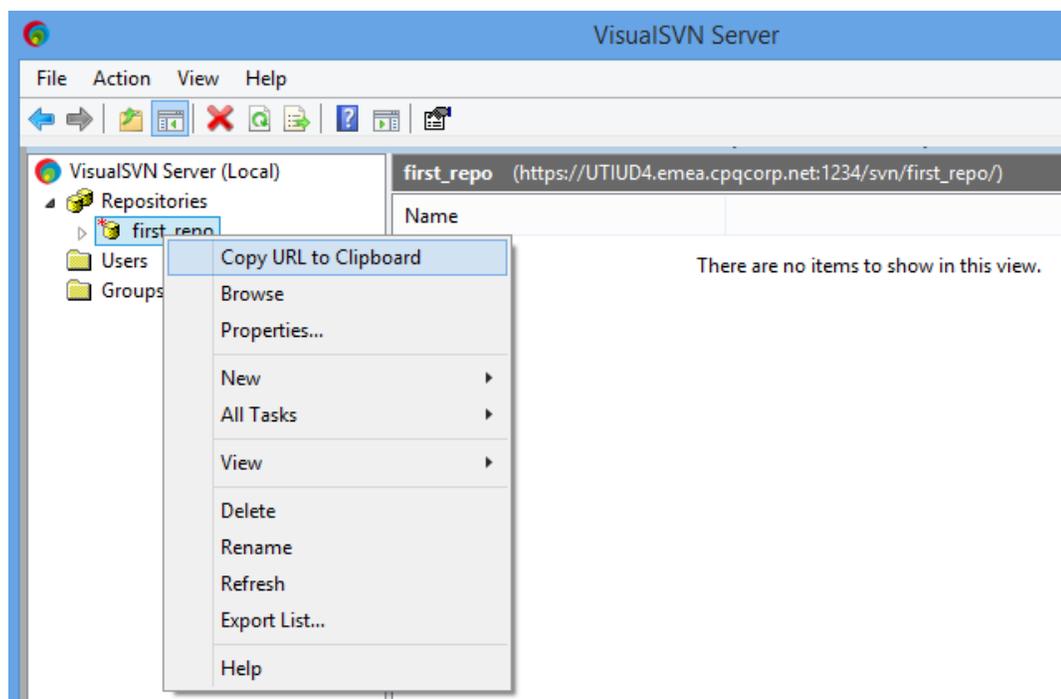
4. Studio で[SCM] メニューから [設定] を選択します。
5. [Subversion] ラジオボタンを選択します。
6. SVN ユーザー名資格情報 (ユーザー名とパスワード) を入力します。



7. HTTPS プロトコルの場合は、[サーバー証明書を信頼] チェックボックスをオンにします。
8. リポジトリをチェックアウトします。

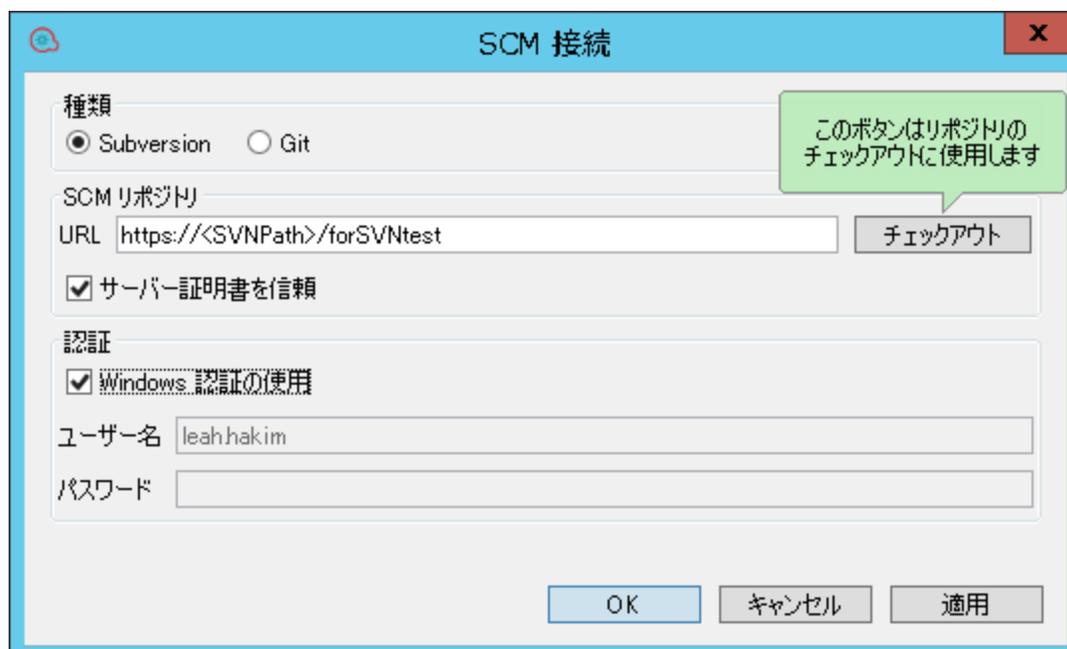
VisualSVN コンソールアプリケーションに直接アクセスできる場合は、次のようにしてリポジトリの URL を取得できます。

- a. VisualSVN でリポジトリの場所を右クリックします。
- b. [Copy URL to Clipboard] を選択します。



VisualSVN コンソールアプリケーションにアクセスできない場合は、SVN 管理者にリポジトリの URL を問い合わせてください。

9. Studio で[SCM]メニューから [接続] を選択します。



10. [SCM リポジトリ]領域の [URL] フィールドで、Ctrl + V を使用して URL を貼り付け、[チェックアウト] をクリックします。

[SCM メッセージ] ペインにチェックアウトメッセージが表示されます。

例 2: 単一ユーザーのシナリオ

次の例は、テストとデバッグを目的とした単一ユーザーのシナリオです。

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

注: 実稼働では、ファイルシステムベース(またはネットワーク共有ベース)のリポジトリを使用しないことをお勧めします。

このリポジトリは、特にテストとデバッグでローカルの単一ユーザーアクセスのみを想定していて、ネットワーク経由での共有時には、セキュリティ上の問題およびパフォーマンス上の問題が発生する可能性があります。

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

SVN リポジトリを作成するには、

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

例:

```
c:\<Studio インストールフォルダー>\<Studio> manageScmRepository.bat SHARED_FOLDER_
FULL_PATH
```

次のように表示されます。

```
Repository was successfully created at <SHARED_FOLDER_FULL_PATH>
```

これで、共有リポジトリフォルダーを Studio で使用する準備が完了です。

SVN での複数の作成者による作業

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

作者がフローを削除、移動、名前変更、または復元しようとする、フローは自動的にロックされます。フローがロックされると、他の作者がこれらのいずれかのアクションを実行しようとする、フローがロックされていることを知らせるメッセージが表示されます。これは、**[ロックポリシーの強制]**が有効になっているときに発生します。詳細については、「[ロックポリシーの強制](#)」(66ページ)を参照してください。

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

SCM リポジトリに存在しないアイテムをロックすることはできません。

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

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

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

注: ロックは特定の SCM ユーザーではなく、ワークスペースに関連付けられます。このため、アイテムをロックしてワークスペースを破棄すると、ワークスペースの再作成後に、前にロックしたアイテムで作業を継続できないだけでなく、ロック解除することもできなくなります。この場合、SVN 管理者に連絡するか、外部の SVN ツールを使用して、アイテムのロックを解除する必要があります。

ロックポリシーの強制

このオプションは、アイテム(フローまたは構成アイテム)がロックされていなければ、フロー作成者がそのアイテムを変更できないようにします。これにより、アイテムを編集できる作成者を 1 人に限定することが可能になります。

このオプションは、作成者が SCM リポジトリをチェックアウトすると自動的に選択されます。

アイテムがロックされている場合、作成者はそのアイテムの手動によるロック解除を選択できます。

作成者は、そのアイテムを変更しロックを解除した後に次のいずれかのオプションを選択するように求められます。

- 変更のコミット
- アイテムをロック解除して編集を継続
- アイテムを元に戻す

次の場合、作成者はロックを取得できません。

- 別の作成者がアイテムをすでにロックしている
- ネットワーク上の問題がある
- アイテムのバージョンが期限切れになっている

上記の場合、[ロックを取得できませんでした]というメッセージが表示され、ローカル変更があるかどうかの確認が求められます。編集を続行するには、[はい]をクリックし、変更を破棄するには、[いいえ]をクリックします。

[ロックポリシーの強制]が有効になっているときには、Studio はプロジェクトの構造を変更するすべてのオペレーションの影響を受けるアイテムのロックを自動的に取得します。たとえば、フロー、オペレーション、またはフォルダーの名前の変更、移動、削除などが該当します。

実行する操作

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

ソースコントロールサーバーで認証を設定する

SCM を使用する際の最初のステップとして、ソースコントロールサーバーでのユーザー認証の設定を行います。

1. [SCM] メニューから [接続] を選択します。

こちら をご覧ください。' (Please see [here](#) to view). At the bottom, there are three buttons: 'OK', 'キャンセル' (Cancel), and '適用' (Apply). A green callout box points to the 'クローン' button with the text: 'このボタンはリポジトリのチェックアウトに使用します' (This button is used for checking out the repository)."/>

2. [種類] で、[Subversion] ラジオボタンを選択します。

- 使用するソースコントロールサーバーに応じて以下のオプションから1つ選択します。
 - Windows 認証の使用:** このオプションは、デフォルトで選択されています。現在ログインしているユーザーを使用して認証を実行します。ファイルベースのソースコントロールリポジトリに適用されます。
 - [Windows 認証の使用]** オプションの選択を解除します。ソースコントロールサーバーで定義されているユーザー名とパスワードを入力します。セキュリティで保護されたサーバー (SSL/SSH) を使用している場合は、**[サーバー証明書を信頼]** を選択します。選択しないと、サーバーにアクセスできません。

注: 資格情報は、SCM 自体へのアクセスのみに使用されます。システムが簡易ネットワークファイル共有をベースにしている場合、HP 00 は共有可能なネットワーク接続をオペレーティングシステムが生成するとみなすので、HP 00 はネットワーク接続用の資格情報をオペレーティングシステムに渡しません。

リポジトリをチェックアウトする

リポジトリには自分のプロジェクトがすべて含まれ、複数の作成者と共有することができます。リポジトリは、ファイルシステムまたは Web サーバー上でホストすることができます。

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

すべてのプロジェクトをチェックアウトするには、次の手順を実行します。

- [SCM]** メニューから **[接続]** を選択します。

- [SCM リポジトリ]** 領域の **[URL]** フィールドで、Ctrl + V を使用して URL を貼り付け、**[チェックア**

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

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

例:

- Web ベースの URL:

```
http://svn.samplehost.com/repo/trunk
```

- リポジトリフォルダーを共有していない場合は、以下を使用して接続します。

```
file:///c:/temp/repo/trunk
```

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

```
file://myshared/repo/trunk
```

[SCM メッセージ] ペインにチェックアウトメッセージが表示されます。**[SCM メッセージ]** ペインに共有フォルダーとファイルのチェックアウト先の場所が表示されます。チェックアウトリポジトリにすでにプロジェクトがある場合は、これらを Studio にインポートして作業することができます。

注: [SCM メッセージ] ペインで、ソースコントロール管理クライアントツールからのメッセージには、エンコードされた URL が含まれる可能性があります。これは標準のエンコード形式によるリポジトリ URL です。このメッセージは、Studio が使用している外部 SVN クライアントツール (SlikSVN) から生成されたものです。URL のエンコードされていないバージョンも、[SCM メッセージ] ペインに記録されます。

3. 次のステップは、新しいプロジェクトの作成または既存プロジェクトのインポートのどちらかになります。**[プロジェクト]** ペインで、**+** ボタンをクリックして新しいプロジェクトを追加するか、**📁** をクリックして既存プロジェクトをインポートするかのどちらかを行います。

SVN からワークスペースをデタッチする

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

1. **[SCM]** メニューから **[接続]** を選択します。
2. [SCM リポジトリ] 領域で、**[デタッチ]** をクリックします。
3. **[OK]** をクリックしてウィンドウを閉じます。

注: SVN リポジトリからデタッチすると、ユーザーによってすでにロックされたフロー、オペレーション、および構成アイテムがすべてロック解除されます。

SVN に変更をコミットする

ローカルで変更を加えたら、[コミット]メニューオプションを使用して、リポジトリに変更内容をチェックインします。変更内容をコミットするにはいくつかの方法があります。

- [SCM]メニューから[すべてコミット]を選択して変更をすべてコミットします。
- [SCM 変更]ペインで、[コミット]ボタンをクリックし、すべての変更をコミットします。
- [プロジェクト]ペインで更新したアイテムを右クリックし、[SCM] > [コミット]を選択します。
- [SCM 変更]ペインで更新したアイテムを右クリックし、[SCM] > [コミット]を選択します。

リポジトリ内のSVNオペレーションの履歴の表示

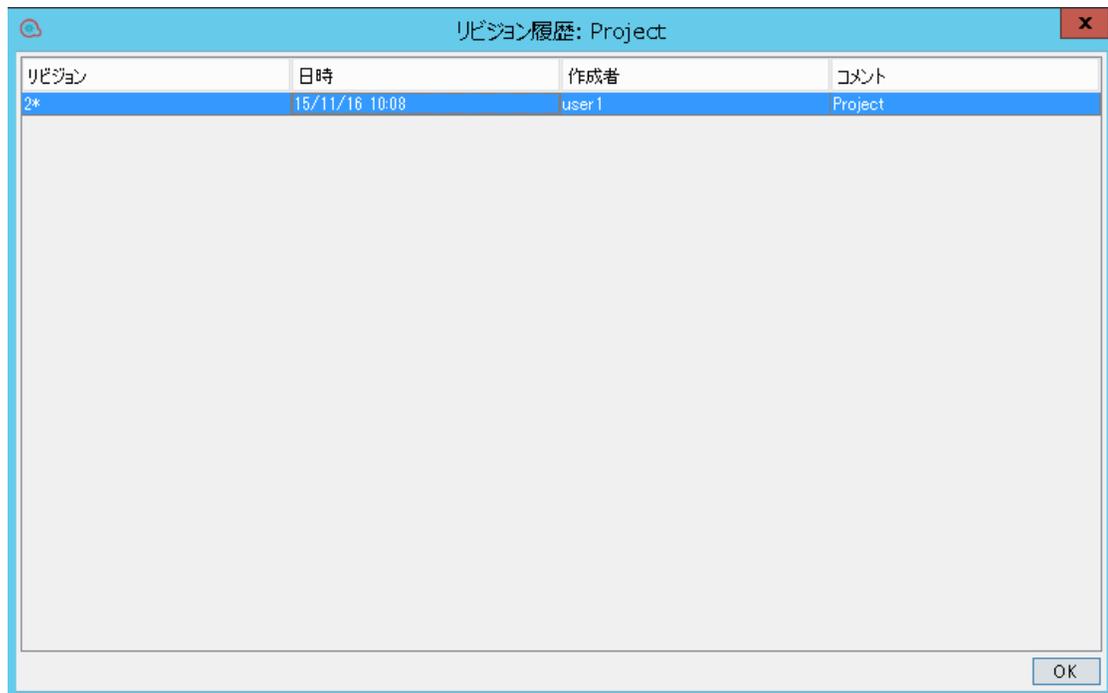
[履歴]オプションには次の情報が表示されます。

- 選択したプロジェクト/アイテムが影響を受けたコミットのリスト。
- 特定のコミットの影響を受けたすべてのファイル。

[履歴]オプションを使用して、プロジェクト/アイテムの特定のバージョンに戻すこともできます。詳細については、「[以前のバージョンのプロジェクト/アイテムに戻す](#)」(70ページ)を参照してください。

1. ファイル/フォルダーを右クリックし、[SCM] > [履歴]を選択します。

[リビジョン履歴]ダイアログボックスが開きます。



以前のバージョンのプロジェクト/アイテムに戻す

前のリビジョンに戻すには、次の手順を実行します。

1. ファイル/フォルダーを右クリックし、[SCM] > [元に戻す] を選択します。
[リビジョン履歴] ダイアログボックスが開きます。
2. 戻す対象のコミットのリビジョンを右クリックします。
3. [このリビジョンに戻す] を選択します。

注: プロジェクトが最初に SVN にコミットされたときに指定された初期リビジョンより前のリビジョン番号に戻すことはできません。

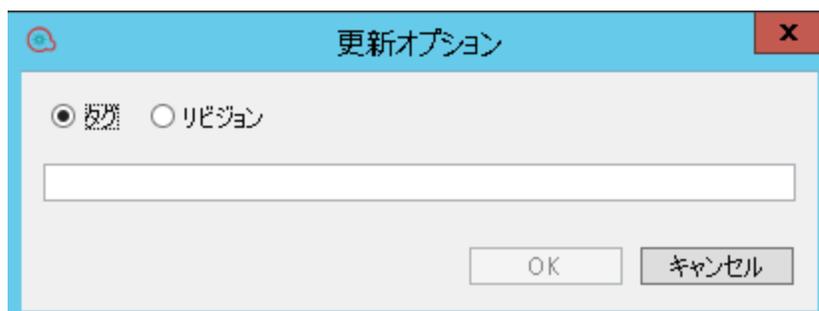
プロジェクト/アイテムのタグ付きのバージョンまたはリビジョンへの更新

コンテンツパック内の特定のフローのトラブルシューティングのために、問題が発生した Central コンテンツパック/プロジェクトの再作成が必要になることがあります。コンテンツパックが使用できない場合は (失われたか新しいバージョンで上書きされた場合)、[更新結果...] オペレーションを使用して特定のバージョンのプロジェクト/アイテムを復元できます。

注: SVN で、個別のフォルダーまたはリポジトリに対して [更新結果...] オペレーションを実行できます。

[更新結果...] オプションは、リポジトリを既にチェックアウトしていて、SCM タグからワークスペースを特定のリビジョンに更新する必要がある場合に使用できます。

1. 更新するコンテンツパック/プロジェクトを選択します。
2. 右クリックして、[SCM] > [更新結果...] を選択します。
[更新オプション] ウィンドウが表示されます。



3. [タグ] を選択し、Central 管理者から受け取った SCM タグを入力します。デフォルトの SCM タグは [Project Name]-[Project version] という形式になっています。
または:
[リビジョン] を選択し、Central 管理者から受け取ったリビジョン番号を入力します。
4. [OK] をクリックします。

バージョン指定されたアイテムの移動/クロス移動

同じプロジェクト内で1つの場所から別の場所にアイテムを移動したり、2つの異なるプロジェクト間でアイテムをクロス移動したりすることができます。

アイテムを移動するには:

1. [プロジェクト] ペインで、移動するアイテムを選択します。**Shift** キーまたは **Ctrl** キーを使用して、複数のアイテムを選択することができます。
2. アイテムを別のフォルダーまたは別のプロジェクトにドラッグアンドドロップします。

または:

Ctrl+X キーを使用してアイテムを切り取り、**Ctrl+V** キーを使用して新しい場所にアイテムを貼り付けます。

[SCM 変更] ペインで、元の場所にあるアイテムのステータスが **[削除されました]** になり、新しい場所のステータスが **[移動されました]** になります。元の場所が括弧で囲まれて表示されま

注:

- バージョン指定されたアイテムを移動するときには、SVM リポジトリでもアイテムの移動がスケジュールされる必要があります。
- アイテムを(ソースプロジェクトと移動先プロジェクトとの間で)クロス移動するときには、適切なプロジェクトでアイテムの移動、削除、または追加をスケジュールする必要があります。
- 必要な場合は、移動/クロス移動操作の結果を SVN に対してコミットできます。

作業コピーをクリーンアップする

場合によっては、Studio のワークスペースで、作業コピーのクリーンアップが必要になることがあります。たとえば、Studio プロセスがクラッシュしたり、I/O エラーが発生したりして、作業コピーがロックされたまま残ったような場合です。

作業コピーをクリーンアップする必要がある場合は、エラーメッセージが表示されます。

ワークスペースをクリーンアップするには:

- [SCM] メニューから [Subversion] > [クリーンアップ] を選択します。

アイテムをロックする

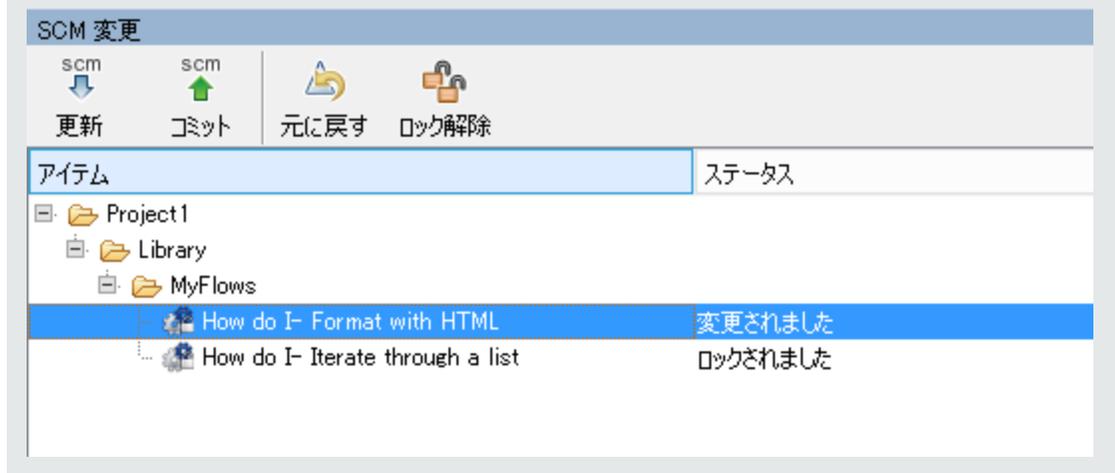
- アイテムの編集ウィンドウの左上隅で開いた鍵のアイコンをクリックします。



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



注: アイテムをロックした後で、[SCM 変更] ペインに表示されるアイテムのステータスが「ロック済み」になります。アイテムがロックされ、変更されている場合 (推奨されません)、ステータスは「変更済み」と表示されます。例:



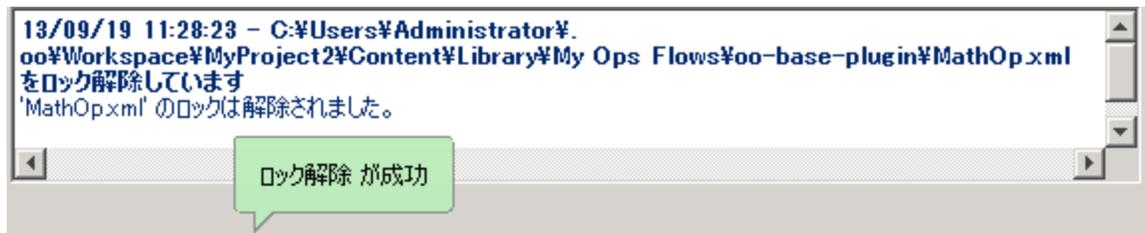
アイテムをロック解除する

- アイテムの編集ウィンドウの左上隅で閉じた鍵のアイコンをクリックします。



ロック解除が成功した場合、アイコンが開いた鍵に変わり、ロック解除されたファイルへの完全パスとユーザーの詳細を示す SCM メッセージが表示されます。





[ロックポリシーの強制] を設定する

- [SCM] メニューから [Subversion] > [ロックポリシーの強制] を選択します。

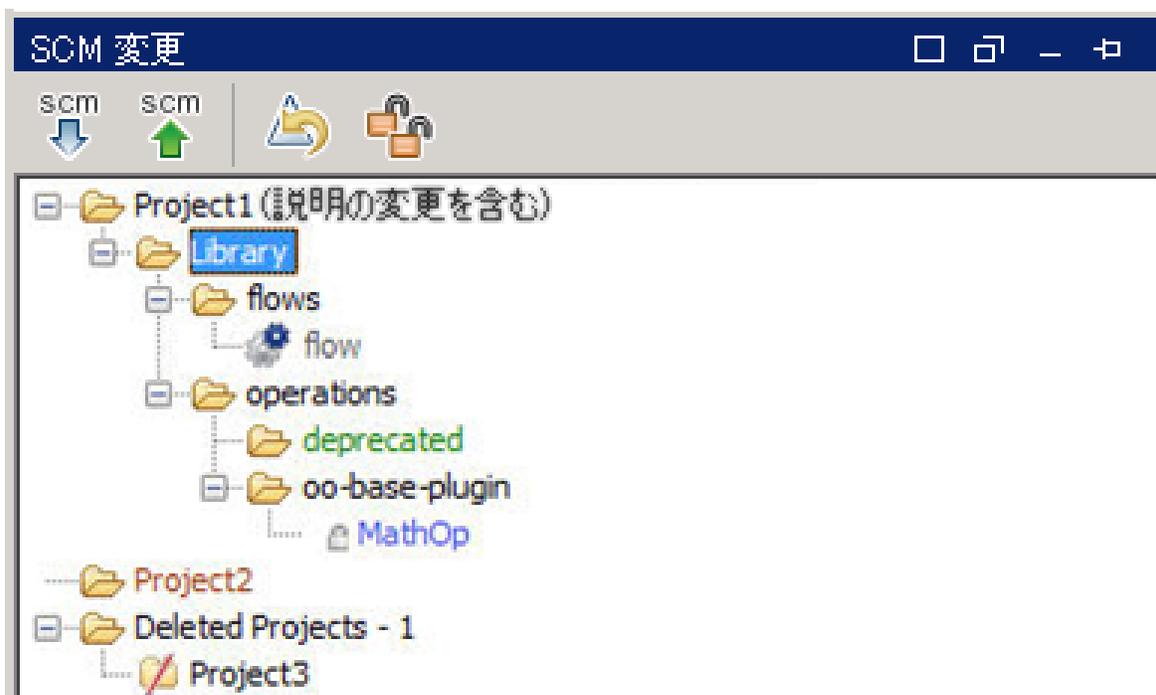
参考資料[SCM 変更] ペイン

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

注: Studio ワークスペースの外部にあるプロジェクトは SCM に追加/コミットされません。それらのプロジェクトは、[プロジェクト] ペインでバージョンなしとして (茶色で) 表示され、[SCM 変更] ペインには表示されません。

変更の種類

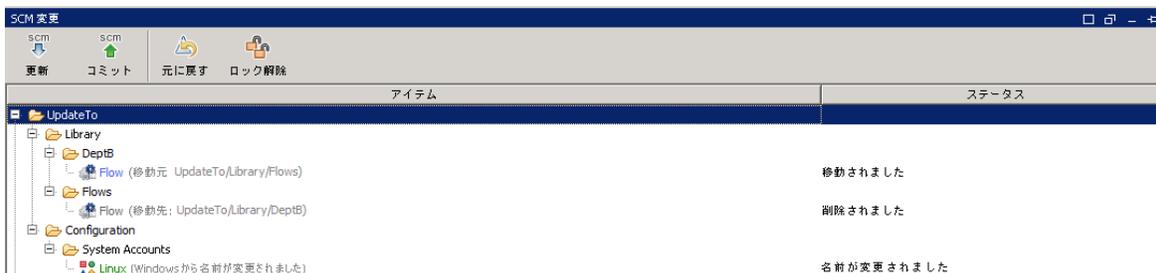
各種変更内容がどのように表示されるか確認できます。



- **flow** という名前の削除済みフロー (グレーで表示)。
- **deprecated** という名前で追加されたフォルダー (緑で表示)。
- **Project2** という名前のバージョンなしプロジェクト (茶で表示)。
- 1つの削除済みプロジェクト。
- また、変更されたアイテム **MathOp** (青で表示) が、ロックインジケータ付きで表示されています。これは、このアイテムはユーザーが編集のためロックされていることを示します。

名前変更および移動されたアイテム

名前変更および移動されたアイテムには、アイテムの移動元または変更前の名前を持つ特別なラベルが付けられます。次の例では、**windows** という名前が **linux** に変更され、フォルダー **flow** が削除されています。



SCM 変更ツールバー

	すべて更新:Studio のワークスペース全体を更新します。
	すべての変更をコミット:[SCM 変更] ペインで表示されるすべての変更をコミットします。変更があった場合にのみ使用可能です。
	すべての変更を元に戻す:[SCM 変更] ペインで表示されるすべての変更を元に戻します。変更があった場合にのみ使用可能です。
	すべてのロックを解除:ロックされたアイテムのロックを解除します。

色分け

Studio では、以下の色で各アイテムを示します。



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

注: デフォルトでは、バージョンなしのファイルが SCM に追加されたときに無視されるファイルパターン (以下に示します) のリストが SVN に用意されています。

```
*.o *.lo *.la *.al .libs *.so *.so.[0-9]* *.a *.pyc *.pyo
```

これは、これらのいずれかのパターンに一致するバージョンなしのプロジェクトまたはフォルダーが、[SCM 変更] パネルでは、黒で (バージョン付きの場合と同様に) 表示されることを意味します。

ただし、そのプロジェクトやフォルダーを SCM に追加して、通常どおり作業することは可能です。

[SCM メッセージ] ペイン

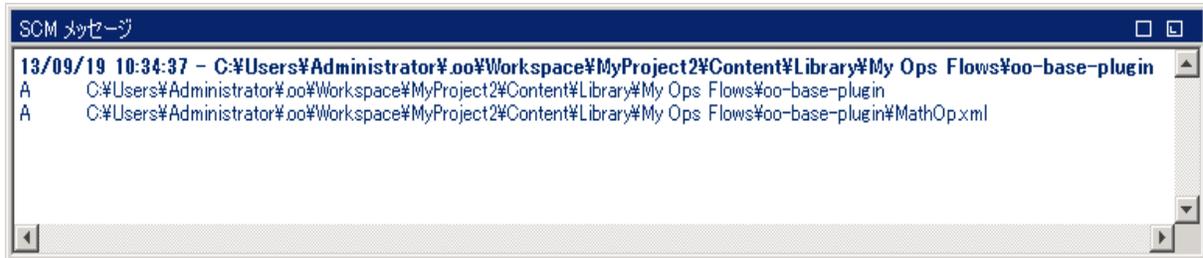
[SCM メッセージ] ペインには、SCM 操作の結果メッセージが表示されます。すべての操作の結果としてメッセージが表示され、色が指定されます。さらに、SCM 操作の結果をユーザーに知らせるポップ

アップメッセージが表示されます。[SCM メッセージ] ペインをクリアするには、ペイン内を右クリックし、[すべてクリア] ボタンをクリックします。

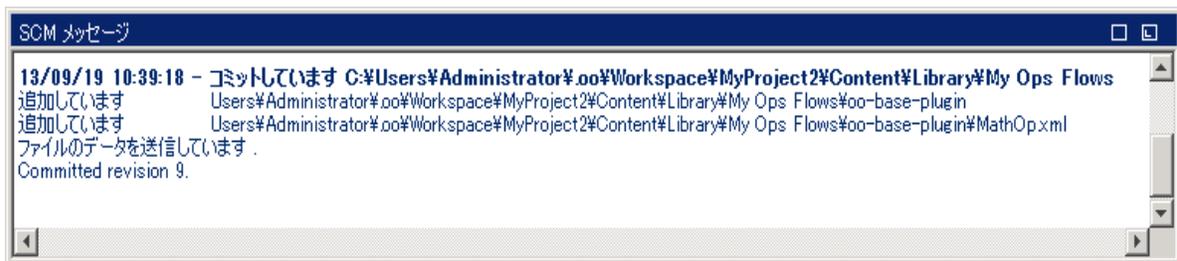
SCM メッセージの例

次の例では、**MathOp** という新しいオペレーションが追加されました。

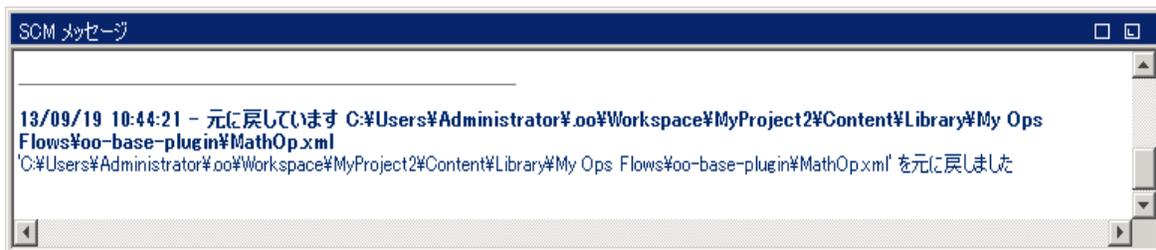
それぞれを追加した後、新しいアイテムが自動的に追加としてマークされます。次の SCM メッセージには、oo-base-plugin フォルダとオペレーションの XML ファイルの MathOp.xml が追加としてマークされています。



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

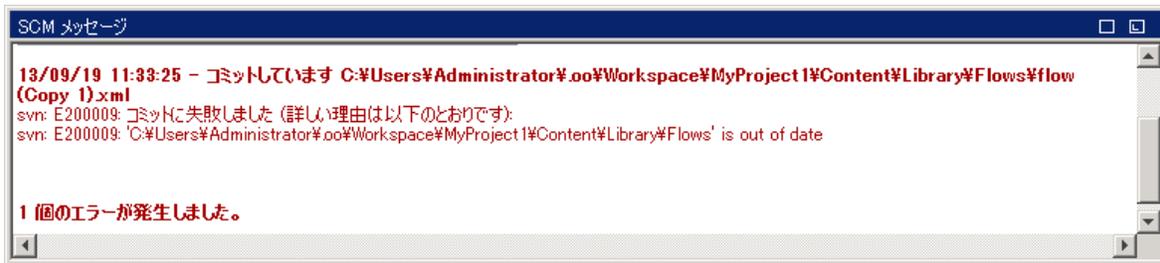


たとえば新しい入力の追加といった変更を **MathOp** に対して行った後、変更を元に戻します。このメッセージには、MathOp.xml への変更が元に戻されたことが示されます。



エラーメッセージ

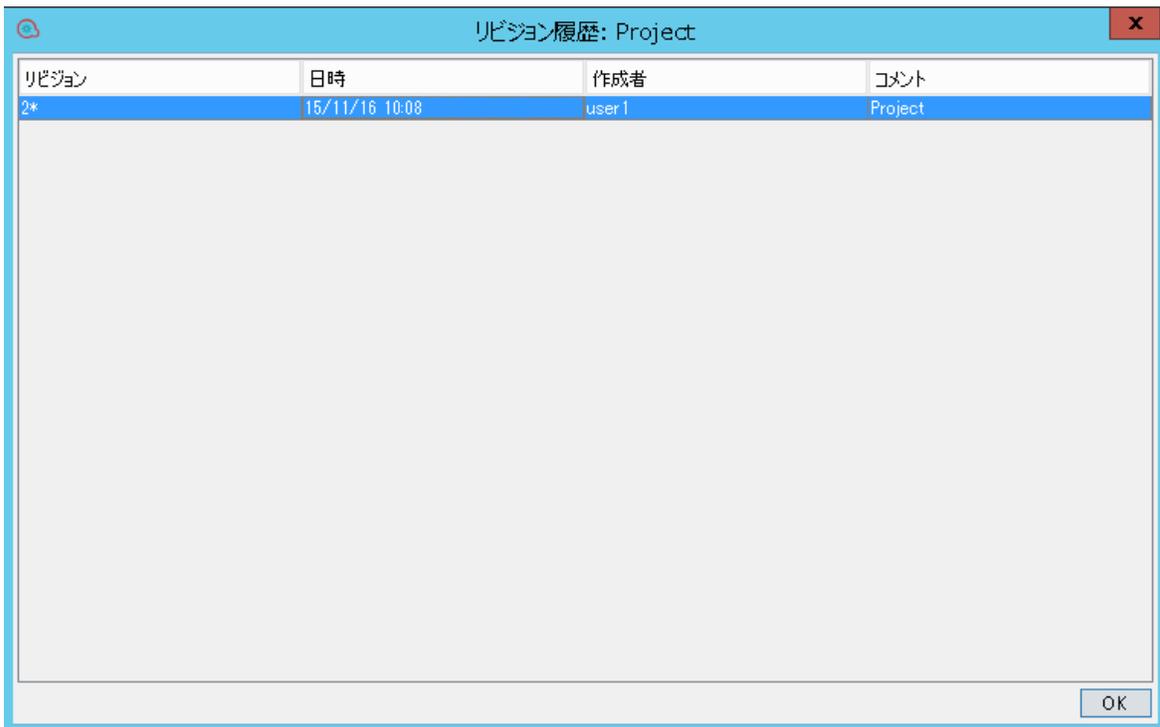
エラーが表示される場合もあります。エラーは赤で表示されます。この例では、作者が変更のコミットを試行しましたが、ファイルが期限切れになっています。



参考資料

リビジョン履歴

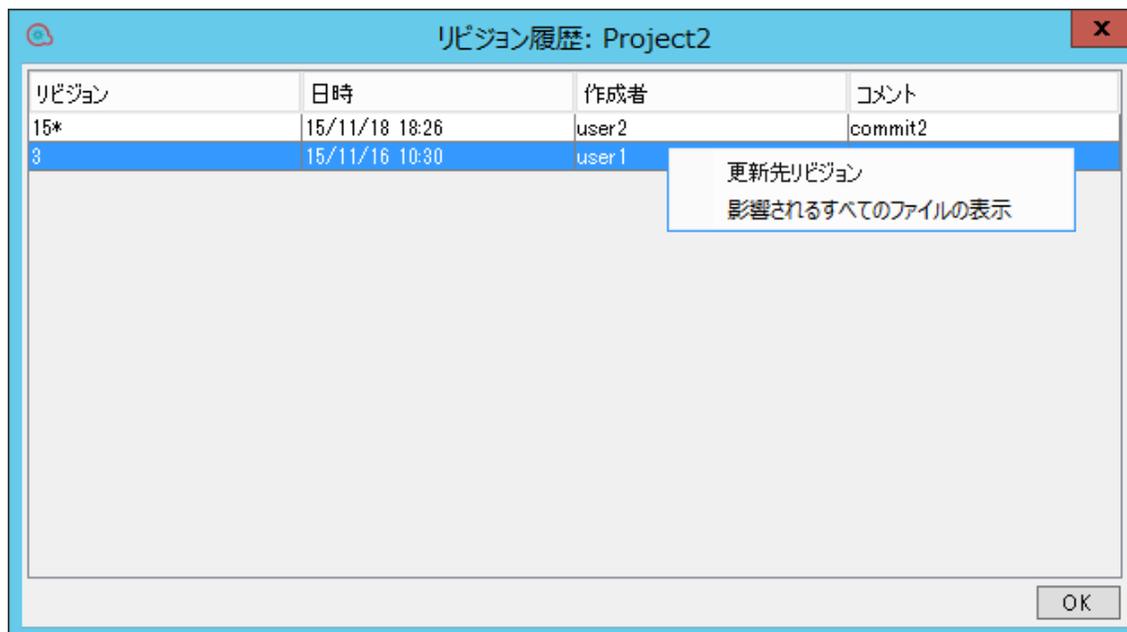
HP OO Studio では、バージョン履歴管理も提供されています。[リビジョン履歴] ペインには、SCM 履歴が表示されます。4つの列に分れ、各コミットが1行ずつ表示されます。



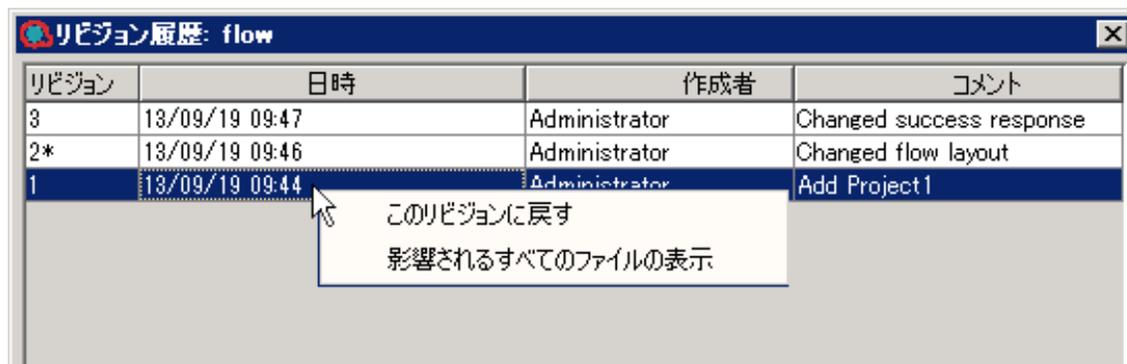
- **リビジョン:** コミットのリビジョン。ローカルコピーのリビジョンにはアスタリスクが表示されます。チェックアウト後、アスタリスクは常に最新のリビジョンを示しますが、別のユーザーが変更をコミットした後かつ更新前の場合は、アスタリスクが示すリビジョンは最新のものではありません。上記のスクリーンショットは、ローカルコピーがリビジョン 1953 であるが、リビジョン 402 でファイルへのコミットが行われたことを示します。
- **日時:** リビジョンがコミットされた日付と時刻。
- **作成者:** リビジョンをコミットした作成者。ユーザー名が表示されます。
- **コメント:** コミット時にユーザーが追加したコメント。

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

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



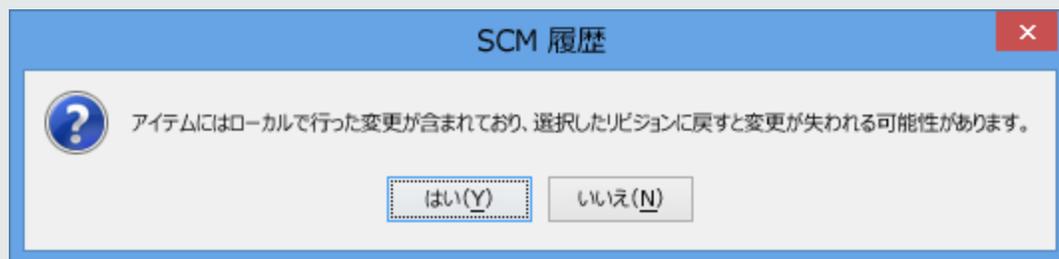
- **このリビジョンに戻す:** 選択したリビジョンと同じ方法でローカルでオブジェクトを変更します。たとえば、リビジョン 333 で 2 つのステップを含み、リビジョン 337 で 3 つのステップを含むフローがある場合、これをリビジョン 333 に戻すと、含まれるステップは再度 2 つになります。ほかの作成者が変更を見ることができるようにするには、フローをコミットする必要があります。以下の例で、アスタリスク (*) の下にあるリビジョンを右クリックします。



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

注: 変更されたアイテムが含まれるフォルダーまたは変更されたフロー/オペレーション/構成アイテムに対して [このリビジョンに戻す] オペレーションを実行しているときには、変更が失われ、ローカルバージョンを使用して解決されるツリーの競合が発生する可能性があります。この

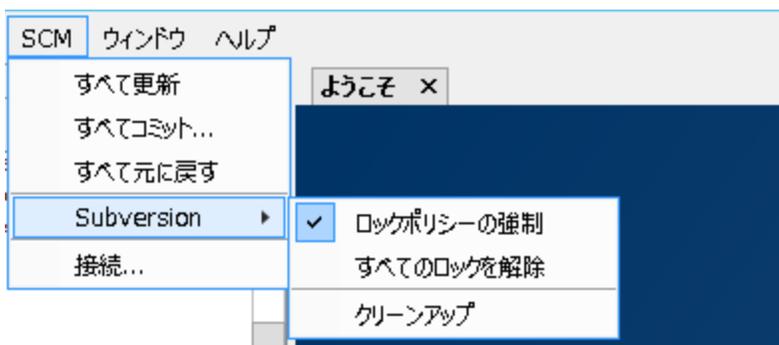
ような場合、次のメッセージが表示されます。



最初に変更を元に戻してから [このリビジョンに戻す] をもう一度実行することをお勧めします。

[ソースコントロール] メニュー

ワークスペース全体に影響するオペレーションが含まれます。



オプション	説明
すべて更新	ワークスペースを直接更新しますが、Studio にプロジェクトがインポートされることはありません。すでにインポートされているプロジェクトは、サーバーから受信した更新を反映します。このコマンドはワークスペース全体を更新するため、ほかの作成者がコミットした新しいプロジェクトが、更新中に取得される場合があります。これらのプロジェクトは Studio にインポートして作業することができます。
すべてコミット	アクティブなプロジェクトの変更をすべてコミットします。[コミット] をクリックすると、コミットのコメントを追加できます。
すべて元に戻す	アクティブなプロジェクトの変更をすべて元に戻します。

オプション	説明
[Subversion] > [ロックポリシーの強制]	<p>このオプションは、アイテム(フローまたは構成アイテム)がロックされていなければ、フロー作成者がそのアイテムを変更できないようにします。これにより、アイテムを編集できる作成者を1人に限定することが可能になります。詳細については、前述の「ロックポリシーの強制」を参照してください。</p>
[Subversion] > [すべてのロックを解除]	<p>ワークスペース内のすべてのロックされたアイテムをロック解除します。警告メッセージが表示されます。例:</p> <div data-bbox="483 716 1377 961" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: center; border-bottom: 1px solid #ccc;">ロック解除 x</p> <p> ロックされている変更されたアイテムが 2 個あります。これらのアイテムのロック解除は、ロックポリシーの強制によって推奨されません。</p> <p>すべてのアイテムのロックを解除して元に戻すには、[すべて元に戻す] を使用します。 ロックを解除してコミットするには、[すべてコミット] を使用します。</p> <p>すべてのアイテムのロックを解除しますか?</p> <p style="text-align: center;"> <input type="button" value="はい(Y)"/> <input type="button" value="いいえ(N)"/> </p> </div> <p>[はい] を選択してすべてのアイテムをロック解除するか、メッセージ内に表示される他のオプションを使用します。</p>
[Subversion] > [クリーンアップ]	<p>Studio のワークスペースで、作業コピーをクリーンアップします。</p>
接続	<p>ソースコントロールサーバーとの認証の実行方法を変更し、SCM タイプを選択して、リポジトリをチェックアウト/デタッチすることができます。</p> <p>Windows 認証: 現在ログインしているユーザーを使用して認証を実行します。ファイルベースのソースコントロールリポジトリに適用されます。</p> <p>[ユーザー名] と [パスワード] の認証: 入力されたユーザー名とパスワードを使用して認証を実行します。</p> <p>サーバー証明書を信頼: セキュリティで保護されたサーバー (SSL/SSH) を使用している場合は、このオプションを選択します。選択しないと、サーバーにアクセスできません。</p>

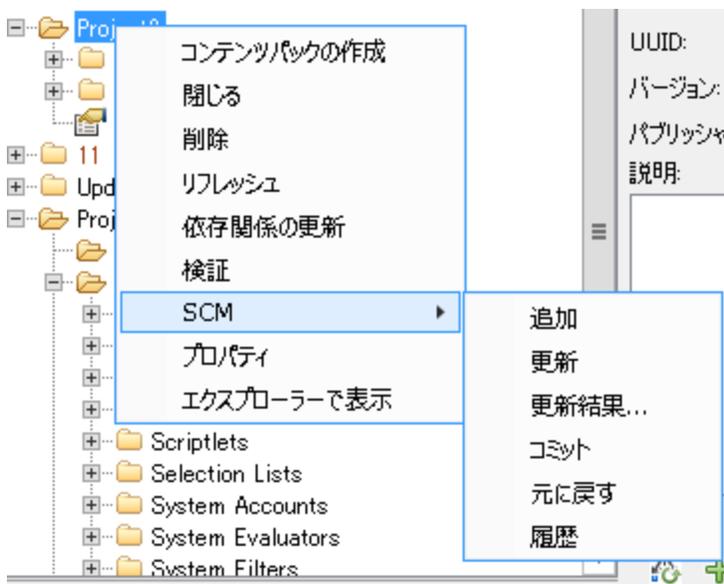
[プロジェクト] ペイン

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

[プロジェクト] ペインのコンテキストメニュー

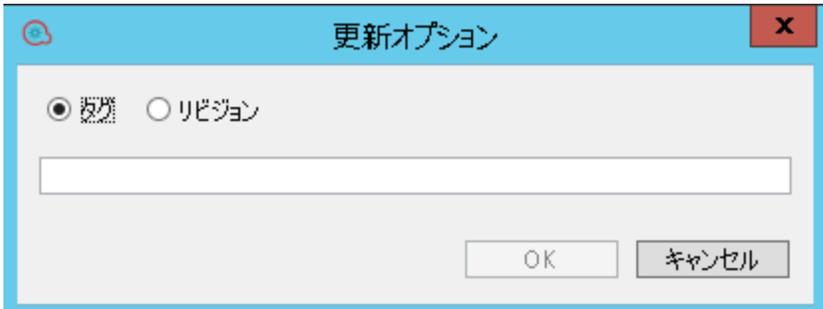
コンテキストメニューには、選択したオブジェクトで使用可能なアクションがその状態に応じてすべて表示されます。

フローを右クリックした際に使用可能なオプションを以下に示します。



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

オプション	説明
追加	ソースコントロールに追加するアイテム (フロー、オペレーション、構成アイテムまたはフォルダー) をマークします。Studio の [追加] では、先祖および子孫オブジェクトがすべて含まれます。つまり、フォルダーが追加されると、子フローと親フォルダーもすべて追加されます。兄弟アイテムは追加されません。Studio 内で作成されたアイテムは自動的に追加され、次のコミットでコミットされます。
コミット	ローカルでの変更をサーバーにコミットします。このオプションは、変更されたアイテムや子アイテムが変更されたフォルダーで使用できます。コミットは再帰的に行われるため、フォルダーをコミットすると、その子アイテムもすべてコミットされます。コミット後、コミットのコメントを追加できます。
更新	選択したアイテムを更新します。このオプションは、ローカルで追加され、まだコミットされていないアイテムに対しては使用できません。再帰的に動作し、すべての子アイテムが更新されます。

オプション	説明
更新結果...	<p>選択したアイテムを特定のバージョンに更新します。このオプションを選択すると、[更新オプション]ウィンドウが表示されます。</p>  <p>タグ名 (Central のデプロイページから取得) またはリビジョン番号 ([リビジョン履歴] ウィンドウから取得) を入力し、[OK] をクリックします。</p>
元に戻す	<p>選択したアイテムのローカルでの変更をすべて元に戻します。フロー、オペレーション、構成アイテムでの変更が元に戻ります。削除されたアイテムが復元されます。</p> <p>重要: アイテムを追加すると自動的に追加済みのマークが付きます。それに対し変更を元に戻すと、追加マークが削除されますが、アイテムは削除されず、Studio とファイルシステムに存在したままになります。メニューの [追加] を使用してアイテムを再び追加するか、アイテムを削除することができます。</p>
履歴	<p>[リビジョン履歴] ウィンドウには、SCM 履歴が表示されます。詳細については、「リビジョン履歴」を参照してください。</p>

プロジェクトの削除

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

Git ソース管理システムの使用

Git は、ソースコントロール管理 (SCM) システムであり、Subversion - SVN ソリューションと並行で実行される代替のソリューションを提供します。Git は、分散バージョンコントロールシステムであり、一部のコンセプトは Subversion とは異なります。

Studio で Git を使用する際の方法論は SVN の方法論と似ています。

10.50 の Git サポートの初期バージョンは、他の IDE ツール (IntelliJ IDEA、Eclipse など) に統合される Git サポートと同じように、基本的なユースケースを対象として使いやすさを実現しています。

Git の用語

追加:Git リポジトリへの新しいファイルの追加

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

クローン

クローンは、リポジトリから作業コピーにソースをダウンロードするために使用します。ソースコントロールサーバーからファイルにアクセスする場合に、最初に行うオペレーションがクローンです。リポジトリのクローンを作成すると、作業コピーが作成され、その内容を編集、削除、追加できるようになります。クローンを作成するには、クローンを作成する Git リポジトリのソースコントロールサーバー URL が必要です。

コミット: リポジトリへの変更の保存

ローカルの作業コピーに変更を加えても、ソースコントロールサーバーには自動的に保存されません。変更を確定するには、その変更をコミットする必要があります。変更はローカルの Git リポジトリに保存されます。その後で、リモート Git リポジトリにそれらをプッシュできます。

削除: リポジトリからのファイルの削除

削除コマンドでは、作業コピー (リポジトリ) からアイテムを削除できます。リモートリポジトリに対して変更をコミットすると、ファイルがリポジトリから削除されます。

ダウンストリーム/アップストリーム

ソースコントロールの観点では、リポジトリからコピーするとき (クローン作成、チェックアウトなど) がダウンストリームになります。情報は「ダウンストリーム」の方向に流れます。

変更を加えるときには、通常はそれらを「アップストリーム」に戻す必要があります。同じソースから情報を取得しているすべてのユーザーが同じ変更で作業できるように変更をリポジトリに反映させます。これは、ソースコントロールの技術的要件というよりは、すべてのユーザーがどのようにして作業を調整できるかという社会的な問題です。開発の枝葉の部分に追従しないようにするには、メインプロジェクトに変更を加える必要があります。

無視

Git でのバージョンコントロールが不要なファイルがあります。これらのファイルは、バックグラウンドで動作し、プロジェクトに追加するときに自動的に実行される Git の無視メカニズムによって処理されます。デフォルトでは、(プロジェクトのホームに対して相対的な) *.idx、*.tmp、*.lock、Content/.metadata のファイルが無視されます。

マージ

マージコマンドは、(履歴が現在のブランチから分岐した時点以降の) コミットからの変更を現在のブランチに組み込みます。

移動

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

プル

プルは、変更を取得してマージする状況を意味します。たとえば、共同作業中のリモートファイルを相手が編集した場合、それらの変更をローカルコピーにプルして、最新の状態にする必要があります。

プッシュ

プッシュは、コミットした変更をローカルリポジトリからリモートリポジトリに送信することを意味します。たとえば、ローカルで何かを変更した場合、それらの変更をプッシュし、他のユーザーがアクセスできるようにする必要があります。

リベース

リベースは、1つのブランチ上でコミットされたすべての変更を取得し、別のブランチ上で再現します。

スタッシュ

スタッシュは、作業ディレクトリとインデックスを安全な場所に保存し、作業ディレクトリとインデックスを最後のコミットに復元します。スタッシュの後に、他のブランチでの作業やコミットなどを行い、元の状態に戻る準備ができたなら、「**git stash apply**」(unstash) オペレーションを実行して、作業コピーを復元します。

作業コピー

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

更新: 作業コピーの更新

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

注: この用語集のいくつかの用語は、github サイト <https://help.github.com/articles/github->

[glossary/](#) から引用されています。

Git の詳細については、次のいずれかのサイトを参照してください。 <https://help.github.com/> または <http://git-scm.com/>。

Studio Git 統合機能を使用するには、Git クライアントバージョン **git-1.9.5-preview20150319** をインストールする必要があります。

1. 次の URL から Git クライアントをダウンロードします。
<https://github.com/msysgit/msysgit/releases/download/Git-1.9.5-preview20150319/Git-1.9.5-preview20150319.exe>
2. **bin** フォルダが **<oo_installation_folder>/studio/Git** のすぐ下になるように、Git クライアントを **<oo_installation_folder>/studio/Git** に保存します。Git インストールウィザードでは、デフォルトのオプションを使用します。

この代わりに、バージョン **git-1.9.5-preview20150319** の Git クライアントインストールがすでにローカルディスク上にある場合は、以下の手順を実行して、その Git インストールを使用するよう Studio に指示します。

1. Studio を閉じます。
2. ユーザーホームフォルダ **C:\Users\<user>\.oo** (Studio ワークスペースの場所) に移動し、**Studio.properties** ファイルを探します。
3. **Studio.properties** ファイルを修正して、ファイルの末尾に次のプロパティを追加します。

```
studio.git.installation.location=<git-1.9.5-preview20150319_installation_folder>
```

例:

```
studio.git.installation.location=C:/Program Files (x86)/Git
```

bin フォルダは、**C:/Program Files (x86)/Git** のすぐ下になります。パスの区切り文字として **/** を使用することに注意してください。

4. **Studio.properties** ファイルを保存し、Studio を起動します。

別のバージョンの Git クライアントがインストールされている場合は、**git-1.9.5-preview20150319** バージョンを Studio で使用する必要があります。

Studio で Git を使用する前に

Git を使用するには、リモート Git リポジトリに接続してそのクローンを作成する必要があります。この方法では、作業可能なリポジトリのコピーを作成し、ローカルリポジトリに対して変更をコミットして、その変更をリモート Git リポジトリにプッシュすることができます。

Studio での Git の操作方法の詳細については、次のビデオを観てください。

ビデオ	学習内容
Connecting to Git and cloning a Git repository (Git への接続と Git リポジトリのクローンの作成)	このビデオでは、いくつかの基本的な Git オペレーションを紹介し、Git リポジトリのクローンを作成し、ローカルリポジトリに対して変更をコミットして、それらをリモートリポジトリにプッシュする方法について学習します。
Working with Git – Basic Operations (Git の使用 – 基本的な操作)	このビデオでは、さまざまなタイプの認証 (SSH) を使用してリポジトリのクローンを作成する方法、およびローカルの変更をスタッシュして、以前にコミットしたバージョンに戻し、スタッシュ解除する方法を見ることができます。
Conflict Handling (競合の処理)	このビデオでは、2人の作成者が同じ Studio ワークスペースで変更を行い、コミットして Git にプッシュしようとしたときに Git で発生する可能性がある競合の処理方法について説明します。
Git branching functionality (Git のブランチ機能)	このビデオでは、Git でさまざまなブランチを操作する方法を学習します。
Multi-authoring (マルチオーサリング)	このビデオでは、Git の高度なマルチオーサリング機能について学習します。

Git のブランチのサポート

ブランチは、Git の強みであり、Subversion と比較したときに異なる機能の 1 つです。最もよく使用されるブランチモデルの 1 つが次のブログに記載されています。<http://nvie.com/posts/a-successful-git-branching-model/>

ブランチ管理は、Git ワークフローの重要な部分です。ブランチを使用すると、変更の追跡や他のユーザーとのコラボレーションが容易になります。Git リポジトリ内のブランチを Studio から直接管理できます。Git のブランチの詳細については、次のサイトを参照してください。

<https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow/>

新しいブランチの作成の詳細については、後の「Git リポジトリ内の新しいブランチの作成」を参照してください。

Git リポジトリログについて

Git リポジトリログには、リモート Git リポジトリに対して最近実行されたコミットと更新がグラフィカルに表示されます。この機能は、他の Git ツールで提供されるログに似ています。

リポジトリログには次の情報が表示されます。

- 重要なポインター (マスター、HEAD、オリジナル/マスター、ブランチ、タグなど)。これを使用して、コミットの実行元のブランチ、コミットのタイプ、リモートブランチにプッシュされたかど

うかを識別できます。

- 先祖関係に基づくさまざまなコミット/更新の関係、およびコミット/更新間のマージの実行方法を示すコミットグラフ。
- [コミット]ウィンドウで入力されたコミットメッセージ。
- コミットの作成者(名前と電子メールアドレス)
- コミットの日付と時刻(日付順)

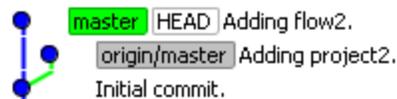
グラフ内の各ノードは、コミット/更新オペレーションを表します。選択した更新方式に応じて(「Git の更新方式」(88ページ)を参照してください)、パスが新しいブランチに分岐されます。

Git の更新方式

リモート Git リポジトリからファイルを更新するときに、使用する更新方式を選択することができます。選択した更新方式に応じてパスが新しいブランチに分岐されます。

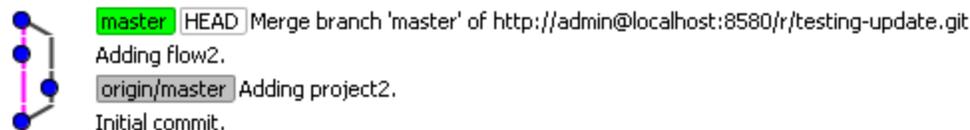
- **更新方式: ブランチデフォルト。** デフォルトの更新方式(常にマージまたはリベース)は、Git リポジトリ構成で次のいずれかの Git キーを使用して構成できます。**pull.rebase** と **branch.<name>.rebase** です。詳細については、<http://git-scm.com/docs/git-config> を参照してください。
- **更新方式: マージ。** マージ更新方式を使用すると、Git がローカルコミットの元の階層を保持し、2つの分岐した「ブランチ」(ローカルブランチとリモートブランチ)を再びマージする特別な「マージ」コミットを作成します。

例:更新の前に次のグラフが含まれているリポジトリの単純な例について見てみましょう。



この場合、ローカルリポジトリとリモートリポジトリが分岐しています。ローカルリポジトリにはリモートリポジトリに含まれていないコミット(メッセージ **Adding flow2** が付けられたコミット)、リモートリポジトリにはローカルリポジトリに含まれていないコミット(メッセージ **Adding project2** が付いたコミット)が含まれています。

2つのコミットをマージした後で、特別なマージコミット **Merge branch 'master' of http://admin@localhost:8580/r/testing-update.git** が作成されます。



マージ操作中に競合が発生した場合、マージコミットは自動的に作成されません。マージが停止します。マージ処理を完了するには競合を解決する必要があります。

この状態では、すべてのリモートの変更がローカルの変更として、[SCM 変更]ペインに表示されます。ユーザーが競合を解決する必要があります。すべての変更を確認し、[SCM 変更]ペインで

「**Commit All Changes**」オペレーションを実行してマージを完了します。詳細については、「[更新オペレーション中の競合の解決](#)」(97ページ)を参照してください。

複数のローカルコミットで複数の競合がある場合は、すべての競合が同時に解決されます。

- **更新方式: リベース。** リベースは変更を統合する代替の方法であり、マージよりも優先される場合があります。マージを使用すると、リポジトリ履歴が水平方向に増加し(すべての個別の「分岐」の情報が保持されるため)、新しいマージコミットが挿入されると垂直方向にも増加します。このため、リポジトリの変更の全体像を把握するのが困難になります。

リベースは、すべてのローカルコミットを変更して、それらがリモートリポジトリからの最近の変更を基にしているように表示します。これにより、リポジトリログのグラフが直線として表示されます。リベース是对話型の操作です。最初に、リモートリポジトリからの最新のコミットの上に最も古い(最も長く使われていない)ローカルコミットをリベースし、次に、前にリベースしたコミットの上に2番目のコミットをリベースし、同様にすべてのローカルコミットをリベースするまで処理を続けます。

特定のコミットのリベース中に競合が発生した場合、リベースが停止し、競合を解決してリベースを続行するか、コミットをスキップするか(すべての変更を破棄します)を選択できます。どちらの場合でも、リベース操作を停止し、初期状態に戻すことができます。これにより、以前にリベースされたすべてのコミットが消えます。

リベースでは、場合によっては同じアイテムについて何度も競合を解決する必要があることに注意する必要があります(異なるコミットをリベースする場合)。

例:

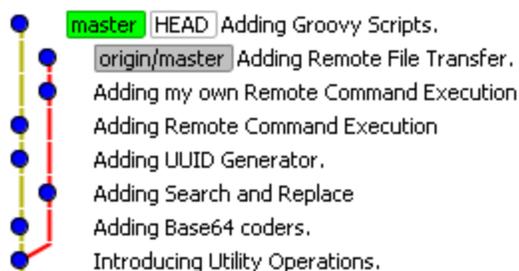
両方のリポジトリ(ローカルとリモート)は、作成者が「**Introducing Utility Operations**」をコミットするまで同一です。次に、ローカルリポジトリの作成者が、「**Base64 coders**」、「**UUID Generator**」、「**Remote Command Execution**」、「**Groovy Scripts**」オペレーションを追加します。

一方で、別の作成者が、リモートリポジトリにコミットをプッシュし、「**Search and Replace**」、独自のバージョンの「**Remote Command Execution**」、および「**Remote File Transfer**」オペレーションを追加します。「**Remote Command Execution**」オペレーションの結果として、両方の作成者が異なるコンテンツでこのオペレーションを追加するために競合が派生します。

更新方式としてリベースを選択して「**Update All**」オペレーションを実行した後で、次のイベントが発生します。

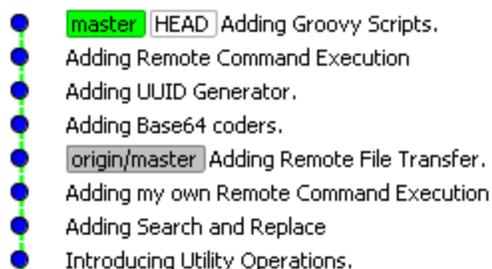
- Studio によって、[競合]ダイアログボックスが表示され、次のメッセージが表示されます。
「**現在のコミットをマージ: 4d10a29 で競合が検出されました。競合を解決してからリベースを続行してください。**」競合内のアイテムは「**Remote Command Execution**」という名前のフローです。
Resolve manually がアイテムに対して選択されます。
- [SCM メッセージ]ペインに、2つのコミット(**Adding UUID Generator** と **Adding Base64 coders**) が正常にリベースされ、競合のためにリベースが停止し、現在のコミットが **Adding Remote Command Execution** であることが表示されます。

この段階で、リポジトリログは次のようになります。



コミット **Adding Base64 coders** と **Adding UUID Generator** は 2 回表示されます。最初は、**Introducing Utility Operations** を基にしたオリジナルの場所で、2 回目はオリジナル/マスターからの最新のコミットを基にしたリストの一番上です。リモートファイル転送現在のブランチ(「マスター」)が引き続き完全な状態になっていることに注意してください。作成者がリベースの停止を選択した場合、ワークスペースはマスターブランチに戻ります。

ユーザーが競合を解決し、リベースの続行を選択したと仮定します。この時点で、2つのコミット(**Adding Remote Command Execution** と **Adding Groovy Scripts**) がさらにリベースされ、リベースが正常に完了したことを説明する SCM メッセージが表示されます。リポジトリログは次のように表示されます。



リベースの後に、ログのグラフが直線になります。リベースの後に、作成者は新しくリベースされたコミットをリモートリポジトリにプッシュすることができます。

注: 移動または名前を変更したアイテム上に(更新中に)競合する変更がある場合、Studio がそれらを自動的に検出して、自動解決を試行します。

たとえば、user1 がフローを移動して変更を加え、user2 が同じフローで競合しない変更を加えたとします。Studio は、この状況を自動的に検出し、それらの変更をフローにマージしようとしています。

Git 更新操作の実行方法の詳細については、「[リモート Git リポジトリからのファイルの更新](#)」(96 ページ)を参照してください。

Git リポジトリログを更新するタイミング

Studio は次の状況で Git リポジトリログを自動的に更新します。

- Studio が起動したとき (Git に接続されます)
- Git リポジトリをチェックアウトした後

- リモート Git リポジトリにファイルをコミットした後
- リモート Git リポジトリからファイルを更新した後
- リモート Git リポジトリからデタッチした後
-  (Git リポジトリログの手動更新) をクリックした後

Git での競合の処理

Git では、自分が加えた変更は「自分」として参照され、異なるユーザーによって実行された変更は「相手」として参照されます。

次のような場合には、ローカルリポジトリを更新しようとしたときに、ファイルのローカルコピー（「自分」によって追加/編集/削除されたもの）とリモート Git リポジトリ内のファイル（「相手」によって追加/編集/削除されたもの）の間に競合が発生することがあります。

- ファイルを自分（「自分」）が削除し、別のユーザー（「相手」）によって変更された場合。
- ファイルが別のユーザー（「相手」）によって削除され、自分（「自分」）が変更した場合。
- 自分（「自分」）と別のユーザー（「相手」）の両方が異なるコンテンツでアイテムを追加した場合。
- 自分（「自分」）と別のユーザー（「相手」）の両方が同じアイテムを変更し、変更を自動的にマージできない場合。

さまざまな競合の種類

作成者の作業中に競合が発生する状況は3つあり、「Update」オペレーションでは3つのすべてが発生する可能性があります。それらの1つは「Unstash」オペレーションでも発生する可能性があります。それぞれの状況で、発生した場合の種類が[競合]ダイアログに表示されます。

- **マージ:** マージオペレーション中の競合。「Update」オペレーションでは、[更新]ウィンドウで選択された更新方式が[マージ]である場合、マージも実行されます。この種類の競合の解決方法の詳細については、「[競合の解決: 自分の内容を受け入れ/相手の内容を受け入れを使用する状況および自分を優先/相手を優先を使用する状況](#)」(92ページ)を参照してください。
- **リベース:** リベース中の競合。「Update」オペレーションでは、[更新]ウィンドウで選択された更新方式が[リベース]である場合、リベースも実行されます。

注: この状況では、「自分」と「相手」の意味はマージの場合と逆になります。そのため、たとえば、作成者が、「自分の内容を受け入れ/自分を優先」を選択した場合、作成者はリモートブランチからの変更を受け入れ/優先し、作成者が「相手の内容を受け入れ/相手を優先」を選択した場合、作成者はローカルブランチからの変更を受け入れ/優先します。

- **Unstash:** スタッシュ解除中の競合。これらの競合は、「Update」オペレーションの前に、作成者がリモートリポジトリに置かれているアイテムと競合状態になっている変更をコミット解除した場合に発生します。

これらの競合は、リベース中に発生した競合と全く同様に動作します（「[リベース: リベース中の競合](#)」。「Update」オペレーションでは、[更新]ウィンドウで選択された更新方式が[リベース]で

ある場合、リベースも実行されます。)」を参照してください。詳細については、「[手動による変更のスタッシュ解除](#)」(101ページ)を参照してください。

競合の解決方法の詳細については、「[更新オペレーション中の競合の解決](#)」(97ページ)を参照してください。

競合の解決: 自分の内容を受け入れ/相手の内容を受け入れを使用する状況および自分を優先/相手を優先を使用する状況

Studio には、次の2つの競合の解決方法が用意されています**受け入れと優先**です。

[**自分の内容を受け入れ/相手の内容を受け入れ**]の解決では、自分/相手からのファイル全体を使用します。一部のファイルが変更されている場合でも、ファイルのマージはありません。

[**自分を優先/相手を優先**]の解決では、選択されたファイル内のできる限り多くの変更を自動的にマージします。マージが不可能な場合は、優先される側からの変更が選択されます。自動的に解決できない他のすべての変更は、Studio で後で手動でマージできます。競合の自動解決および手動解決の例については、「[更新オペレーション中の競合の解決](#)」(97ページ)を参照してください。

たとえば、user1 と user2 という2人のユーザーが同じフローを変更したとします。

- user1 は、特定のステップを変更し、**input1** という名前のフローレベルの入力を追加しました。
- user2 は、同じステップを変更し、**output2** という名前のフローレベルの出力を追加しました。

この場合、user1 が競合を解決する必要があり、次のオプションを使用できます。

- **自分の内容を受け入れ** - user1 のフロー全体が使用され、user1 のバージョンのステップおよび **input1** が使用されますが、**output2** は使用されません。
- **自分の内容を受け入れ** - user2 のフロー全体が使用され、user2 のバージョンのステップおよび **input1** が使用されますが、**output2** は使用されません。
- **自分を優先して手動で解決** - フローは user1 のバージョンのステップを含み、**input1** と **output2** の両方を含みます。
- **相手を優先して手動で解決** - フローは user2 のバージョンのステップを含み、**input1** と **output2** の両方を含みます。

注: 「Update」オペレーション中に移動または名前を変更したアイテム上に競合する変更がある場合、次の例に示すように Studio がそれらを自動的に検出して、自動解決を試行します。

1. User1 が、フローを移動し、いくつかの変更を加えます。
2. User2 が、同じフローでいくつかの競合しない変更を加えます。

Studio がこれらの変更を自動的に検出し、自動的にマージします。競合がないので、両方のユーザーがフローの変更を表示できます。

認証オプション

Git リポジトリに接続するときには、次の2つのいずれかの認証スキームを使用して Git サーバーに接続することを選択できます。資格情報と SSH 秘密キーです。

- **Git リポジトリの資格情報: パスワード認証**
- **秘密キーファイル:** このスキームを使用するには、秘密キーファイルと秘密キーパスフレーズを入力する必要があります。Git コマンドラインから次のコマンドを実行して、**公開キー-秘密キーペア**を生成できます。

```
$ ssh-keygen -t <algorithm_type> -C "your_email@example.com"
```

ここで、`algorithm_type` はプロトコルバージョン 1 の場合は `rsa1`、プロトコルバージョン 2 の場合は `rsa` または `dsa` になります。

注: 秘密キーは OpenSSH 形式になっている必要があります。公開キーの場合、RSA および DSA アルゴリズムがサポートされます。公開キーの拡張子は必ず `.pub` になっている必要があります。

さらに、`studio.properties` ファイル内で既知のホストパラメーターを構成できます。

```
git.ssh.known.hosts.policy=allow|strict|add
```

既知のホストのポリシー。add に設定した場合、**known_hosts** ファイルに含まれていないホストへの接続が許可され、そのホストおよび接続がファイルに追加されます。allow に設定した場合、Studio で任意のホストへの SSH 接続が許可されます。strict に設定した場合、**known_hosts** ファイル内で指定されたホストへの接続のみが許可されます。

```
git.ssh.known.hosts.file
```

known_hosts ファイルの場所です。デフォルト値は `${user.home}/.ssh/known_hosts` です。

[SCM 接続] ダイアログで認証の種類を選択できます。詳細については、[「Git リポジトリへの接続とクローンの作成」\(93ページ\)](#)を参照してください。

プロキシを使用した Git リポジトリへの接続

Git サーバーとの対話処理を行う Studio で使用可能なすべての Git オペレーションは、プロキシ経由で `http/https` 接続 URL を使用して実行できます。プロキシを認証なしで使用することも、基本的な認証スキームを使用することもできます。

詳細については、[「リモート Central でのデバッグ用プロキシの設定」\(331ページ\)](#)を参照してください。

実行する操作

Git リポジトリへの接続とクローンの作成

Git ソースコントロールシステムを使用するには、最初に Git リポジトリに接続し、ローカルファイルシステムのクローンを作成する必要があります。リポジトリのクローンを作成するときには、Studio によって次の手順が実行されます。

- リポジトリを Studio ワークスペースにコピーします。
- クローン作成されたりポジトリ内の各ブランチのリモート追跡ブランチを作成します。
- クローン作成されたりポジトリの現在アクティブなブランチからフォークされる初期ブランチを作成してチェックアウトします。

その後で、ローカルで作業し、変更をコミットして、その変更を Git リモートブランチにプッシュします。

注:

- クローンを作成すると、デフォルトのブランチが常にチェックアウトされます。
- Git サーバーの検証に失敗した場合、エラーメッセージが表示されます。操作の続行または中断を選択できます。

1. **[SCM]** > **[接続...]** を選択します。
[SCM 接続] ダイアログボックスが表示されます。
2. **[種類]** 領域で、SCM の種類として **Git** を選択します。
3. **[SCM リポジトリ]** 領域の **[URL]** フィールドに Git リポジトリの URL を入力するか貼り付けます。

注: URL には、**http**、**https**、**file**、または **ssh** プロトコルを含めることができます。

https を使用するときサーバー証明書が信頼されていない場合、この証明書を信頼するかどうかを尋ねるメッセージが表示されます。" [「リモートデバッガー接続による証明書の自動インポート」 \(332ページ\)](#) の説明に従って、証明書を自動的にインポートすることもできます。

4. **認証: 資格情報または SSH 秘密キー** の 2 種類の認証を使用できます。
 - **資格情報:** Windows 資格情報を使用する場合は、**[種類]** ドロップダウンで、**[資格情報]** を選択し、**[ユーザー名]** フィールドと **[パスワード]** フィールドにユーザー名とパスワードを入力します。
 - **SSH プライベートキー:** SSH 秘密キー認証を使用する場合は、**[種類]** ドロップダウンで、**[SSH 秘密キー]** を選択し、**[秘密キーファイル]** フィールドでプライベートキーファイルを入力/選択し、**[プライベートキーのパスフレーズ]** フィールドにパスフレーズを入力します。

注:  アイコンをクリックして、ディスクの場所から秘密キーファイルを選択します。

5. **ユーザー情報:** **[フルネーム]** フィールドにユーザー名を入力し、**[電子メールアドレス]** フィールドに電子メールアドレスを入力します。この情報はすべての Git コミットオペレーションに添付されるので、これらの両方のフィールドに入力する必要があります。これらのフィールドに入力しない場合、Git はコンピューター名をフルネームとして使用し、`<computer name>@<full computer name>` を電子メールアドレスとして使用します。
6. **[クローン]** をクリックします。

次の形式のメッセージが SCM メッセージペインに表示されます。

```
<date> <time> - Clone https://<full_path>/gitTestRepo.git  
  
Cloning from https://<full_path>/gitTestRepo.git  
to C:\Users\<username>\.oo\Workspace was done successfully.
```

注:

- 空白ではない宛先ディレクトリを使用して **Clone** コマンドが正常に実行されるようにするために、Studio が (C:\...\oo\Workspace\.git の下に) 空白のディレクトリを作成し、Git リポジトリ全体を新しいディレクトリにコピーします。
- Git に初めて接続しようとしたときに、Windows 資格情報を入力している場合は、Git はそれらを使用して Git ライブラリを構成して入力します。

注: Git にはロック機能はありません。これは、Git 競合管理があるので、Git ではファイルをロックする必要がないためです。そのため、エディターのアイテム (フロー、オペレーション、構成アイテム) のロックアイコンは常に無効になっています。SCM リポジトリに接続していないときと同じ動作です。

ローカル Git リポジトリへのファイルの追加

1. プロジェクトツリーでリポジトリに追加するファイルが含まれるプロジェクトを選択します。
2. 右クリックして、**[SCM] > [追加]** を選択します。
3. メッセージが **[SCM メッセージ]** ペインに表示されます。例:

```
Add C:\Users\<username>\.oo\Workspace\Project\Content\Configuration\Scriptlets
```

注: 現在のワークスペースの外部に置かれているプロジェクトを追加することはできません。

バージョン指定されたアイテムの移動/クロス移動

同じプロジェクト内で1つの場所から別の場所にアイテムを移動したり、2つの異なるプロジェクト間でアイテムをクロス移動したりすることができます。

アイテムを移動するには、次の手順を実行します。

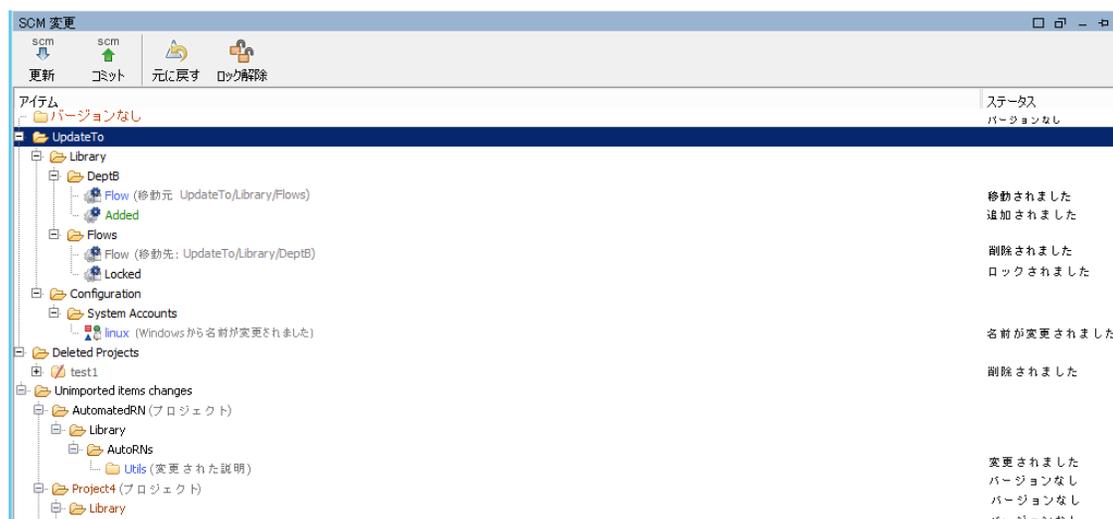
1. **[プロジェクト]** ペインで、移動するアイテムを選択します。 **Shift** キーまたは **Ctrl** キーを使用して、複数のアイテムを選択することができます。
2. アイテムを別のフォルダーまたは別のプロジェクトにドラッグアンドドロップします。

または:

Ctrl+X キーを使用してアイテムを切り取り、**Ctrl+V** キーを使用して新しい場所にアイテムを貼り付けます。

[SCM 変更] ペインで、元の場所にあるアイテムのステータスが **[削除されました]** になり、新しい場所のステータスが **[移動されました]** になります。元の場所が括弧で囲まれて表示されま

す。例:



注:

- 必要な場合は、移動/クロス移動操作の結果を Git に対してコミットできます。

リモート Git リポジトリからのファイルの更新

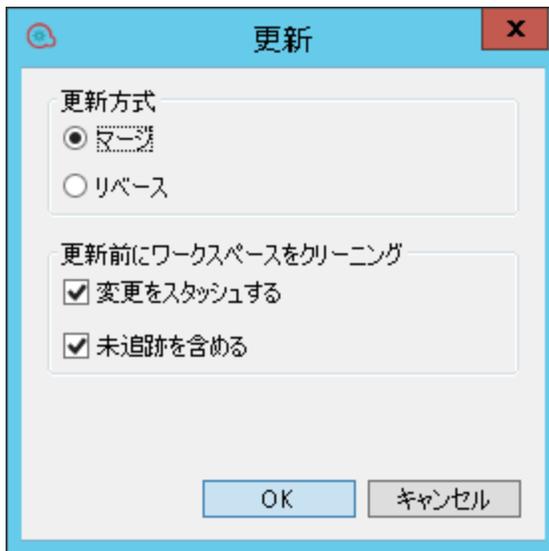
[SCM 更新] オペレーションは、スタッシュ、プル、スタッシュ解除という3つの Git オペレーションを実行します。

スタッシュは、ローカル作業ワークスペースの現在の状態(つまり、変更した追跡されているファイルおよび段階的な変更)を取得し、いつでも再適用できる未完了の変更のスタックに保存します。

ローカルワークスペースに変更が存在し、それらをコミットした場合、Studio でリモート Git リポジトリ内のファイルとの自動マージが試行されます。競合がある場合は、[競合] ダイアログボックスが開き、「更新オペレーション中の競合の解決」(97ページ)の説明に従って競合を解決することができます。

注: SCM > Git メニューの [スタッシュ] および [スタッシュ解除] オプションを使用すると、手動のスタッシュ/スタッシュ解除オペレーションを実行できます。詳細については、「ローカルの変更の手動によるスタッシュ」(100ページ)および「手動による変更のスタッシュ解除」(101ページ)を参照してください。

1. 右クリックして、[SCM] > [更新] を選択します。
2. 次のウィンドウが表示されます。



3. 使用する Git 更新方式を選択します。

- **マージ:** マージ更新方式を使用すると、Git がローカルコミットの元の階層を保持し、2つの分岐した「ブランチ」(ローカルブランチとリモートブランチ)をマージする特別な「マージ」コミットを作成します。
- **リベース:** リベース方式では、すべてローカルコミットを、リモートリポジトリからの最近の変更を基にして実行された場合のようにコミットします。これは、リポジトリログに直線として表示され、マージを使用する場合のように水平方向に増加することがなくなります。

注:

- **pull.rebase** キーと **branch.<name>.rebase** キーを使用してデフォルトの更新方式を変更できます。詳細については、<http://git-scm.com/docs/git-config> を参照してください。
- リベースオペレーションを実行すると、同じファイルの複数の競合が発生する場合があります。

4. **[変更をスタッシュする]** チェックボックスをオンにすると、更新の前にローカルの変更が自動的にスタッシュされます。
5. 追跡されていないアイテム(茶色でマークされるバージョンなしのアイテム)をスタッシュに含めるには、**[未追跡を含める]** チェックボックスをオンにします。

注: リモート Git リポジトリが現在空白になっているときにそのリポジトリから更新しようとした場合、次のエラーメッセージが表示されます。

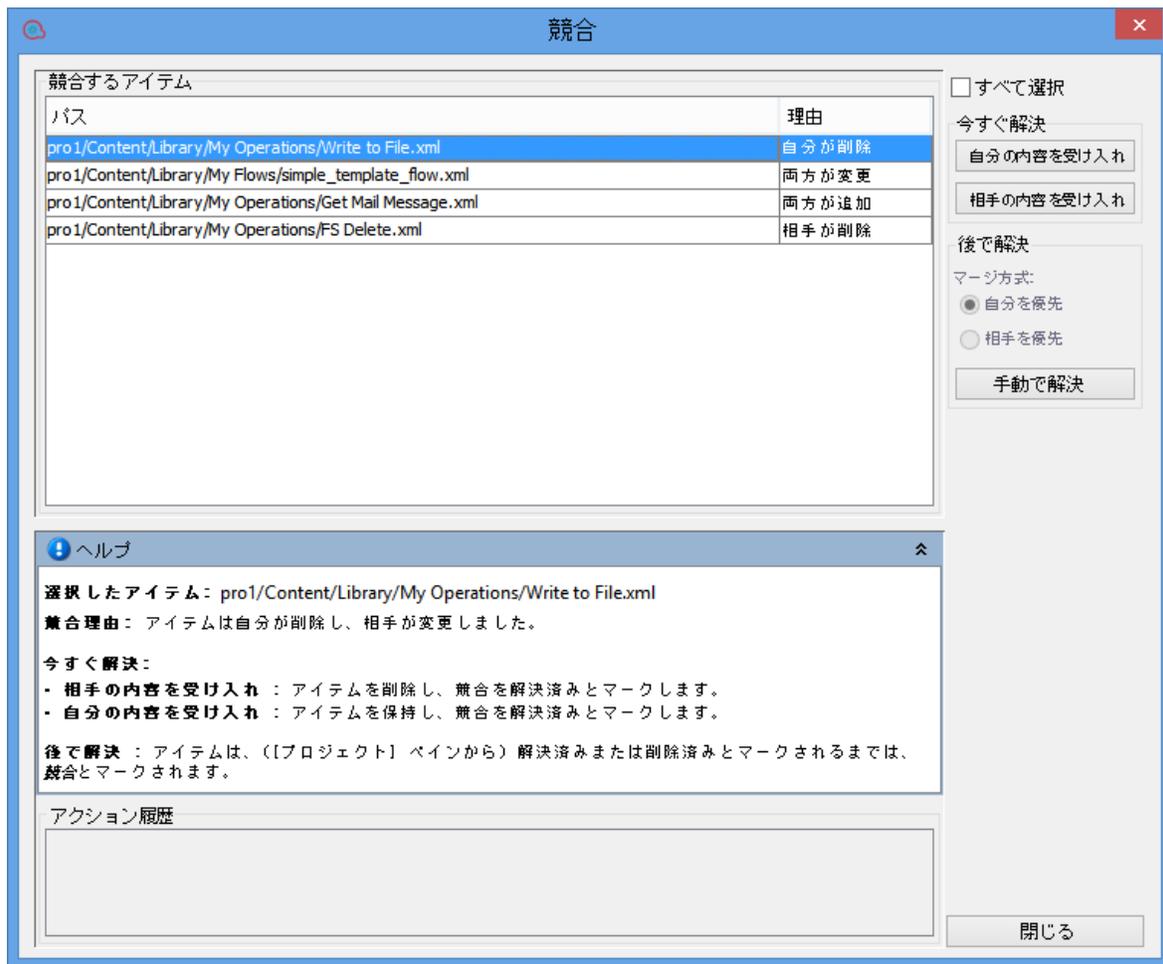
```
Cannot update the current branch as it does not exist in the remote repository.
```

更新オペレーション中の競合の解決

更新オペレーション中の競合は、次のいずれかの状況で表示されます。

- ・ **マージ中:** 作成者がマージ更新方式を選択し、ローカルブランチ上のコミットの間競合する違いが存在し、そのコミットがリモートブランチから取得されている場合。
- ・ **リベース時:** 作成者がリベース更新方式を選択し、ローカルブランチ上のコミットの間競合する違いが存在し、そのコミットされたアイテムがリモートブランチから取得されている場合。
- ・ **スタッシュ時:** 作成者によるローカルの変更がコミットされずにスタッシュされていて、これらのアイテムとリモートブランチから取得されたコミット済みのアイテムの間競合する違いが存在する場合。詳細については、「[ローカルの変更の手動によるスタッシュ](#)」(100ページ)を参照してください。

発生したすべての競合は [競合] ダイアログボックスで表示できます。例:



注: 同じアイテムの複数の競合が存在する場合があります。アイテムを解決済みとしてマークするには、すべての競合を解決する必要があります。

各競合について、競合を今すぐ解決するか後で解決するかを選択できます。

今すぐ解決

- 「自分」が実行した変更 (ローカルリポジトリでコミットされた変更) を適用するには、右側のペインで、**[自分の内容を受け入れ]** を選択します。

または:

「相手」によって実行された変更 (リモートリポジトリからプルして受信した変更) を適用するには、**[相手の内容を受け入れ]** を選択します。

後で解決 (手動)

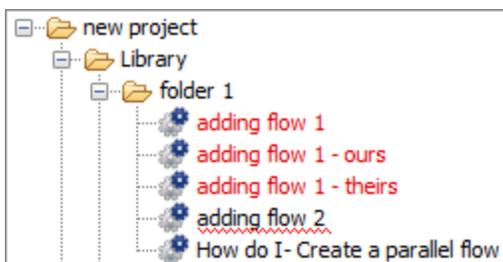
後で競合を解決することを選択した場合は、Studio で手動で競合を解決できるようにアイテムが準備されます。競合していない変更はアイテムに自動的にマージされます。競合している変更は選択した設定 (自分または相手) を基にして置き換えられます。

オリジナルのファイルは、**[プロジェクト]** ペインから解決または削除されるまで競合としてマークされます。

競合の種類が「自分が削除」または「相手が削除」ではない場合、`<filename> - ours` と `<filename> - theirs` という名前の 2 つの新しいアイテムが自動的に作成され、赤で表示されます。

- `<filename> - ours`: アイテムのオリジナルのバージョン (ローカルリポジトリから取得されます)
- `<filename> - theirs`: アイテムのオリジナルのバージョン (リモート Git リポジトリから取得されます)

例:



これらのファイルは、読み取り専用で変更できません。競合を解決した後で、「自分」および「相手」のファイルが自動的に削除されます。

注:

- 解決されていないマージの競合があるときには、Git リポジトリに対するコミットなど、競合しているアイテムに対していくつかのオペレーションを実行できないことがあります。さらに、競合しているアイテムまたはそれらが含まれるフォルダーは移動したり名前を変更したりすることができません。
- 競合するファイルは [問題] ペインに表示されます。
- 自動的にまたは手動による競合のマージによって作成された新しいファイルは、無効な状態になることがあります。

競合を手動で解決するには:

1. 「自分」が実行した変更(ローカルリポジトリでコミットされた変更)を適用するには、右側のペインで、**[自分を優先]**を選択します。
または:
「相手」によって実行された変更(リモートリポジトリからプルして受信した変更)を適用するには、**[相手を優先]**を選択します。
2. **[手動で解決]**を選択します。
3. プロジェクトツリーで競合しているアイテムを右クリックし、**[SCM] > [競合を解決済みとマーク]**、**[SCM] > [自分の内容を受け入れて競合を解決]**、または **[SCM] > [相手の内容を受け入れて競合を解決]** を選択します。
4. リモートリポジトリに対してすべての変更をコミットしてプッシュします。

ローカルの変更の手動によるスタッシュ

スタッシュを使用すると、ローカルの変更の「パッチ」を作成してローカルワークスペース内に保存し、後でいつでもアクセスできます。

スタッシュを作成すると、ワークスペース内のすべての変更が「スタッシュされ」、ローカルの変更が元に戻されます(ワークスペースから削除されます)。

注: 初期化されていない空白のリポジトリに接続されているときに変更をスタッシュしようとした場合、次のエラーメッセージが表示されます。「Before stashing, you must initialize your repository by a first commit」

注: スタッシュオペレーションには、常にワークスペース内のすべての変更が含まれます。

追跡されていないファイルを含めるかどうかを選択することもできます。

注: ワークスペース内にインポートされていないプロジェクトがある場合、スタッシュオペレーションを実行したときにそれらのプロジェクトが含まれます。スタッシュの後には、これらのファイルはワークスペースではなくメモリ内に格納されます。

ただし、空白のフォルダーは残っています。いつでも、変更をスタッシュ解除できます。スタッシュ解除するとスタッシュされていたファイルは元の場所に戻ります。

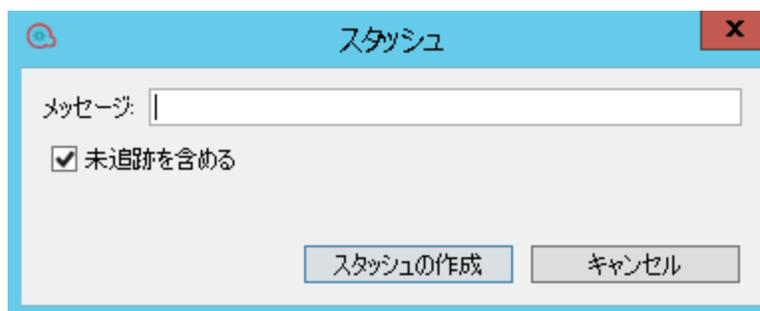
ローカルの変更をスタッシュするには:

1. [SCM] メニューから **[Git] > [スタッシュ]** を選択します。

または:

[SCM 変更] ペインのツールバーで、 をクリックします。

[スタッシュ] ダイアログボックスが開きます。



2. [メッセージ]フィールドに、スタッシュオペレーションを識別するメッセージを入力します。たとえば、「統合オペレーションの変更」と入力します。
3. 追跡されていないファイルをスタッシュに含めるには、[未追跡を含める]チェックボックスをオンにします。
4. [スタッシュの作成]をクリックします。

スタッシュオペレーションが完了すると、[SCM メッセージ]ペインで実行されたオペレーションを表示できます。例:

```
04/14/15 15:13:30 - Stash
ワークスペース内の変更を未追跡のファイルを含めてスタッシュしています。
削除されたプロジェクト: oo-virtualization-project
スタッシュされたパス:
追加: oo-virtualization-project/.gitignore
追加: oo-virtualization-project/Content/Library.properties
追加: oo-virtualization-project/Content/Library/Accelerator Packs.properties
追加: oo-virtualization-project/Content/Library/Accelerator
Packs/Virtualization.properties
追加: oo-virtualization-project/Content/Library/Accelerator
Packs/Virtualization/Get All Snapshots.xml
追加: oo-virtualization-project/contentpack.properties
追加: oo-virtualization-project/pom.xml
```

手動による変更のスタッシュ解除

スタッシュ解除オペレーションを使用すると、以前にスタッシュされた変更を適用できます。このオペレーションは、「**git stash apply**」オペレーションと同じです。[スタッシュ解除]ダイアログボックスには、スタッシュのリストと現在選択されているスタッシュの詳細が表示されます。

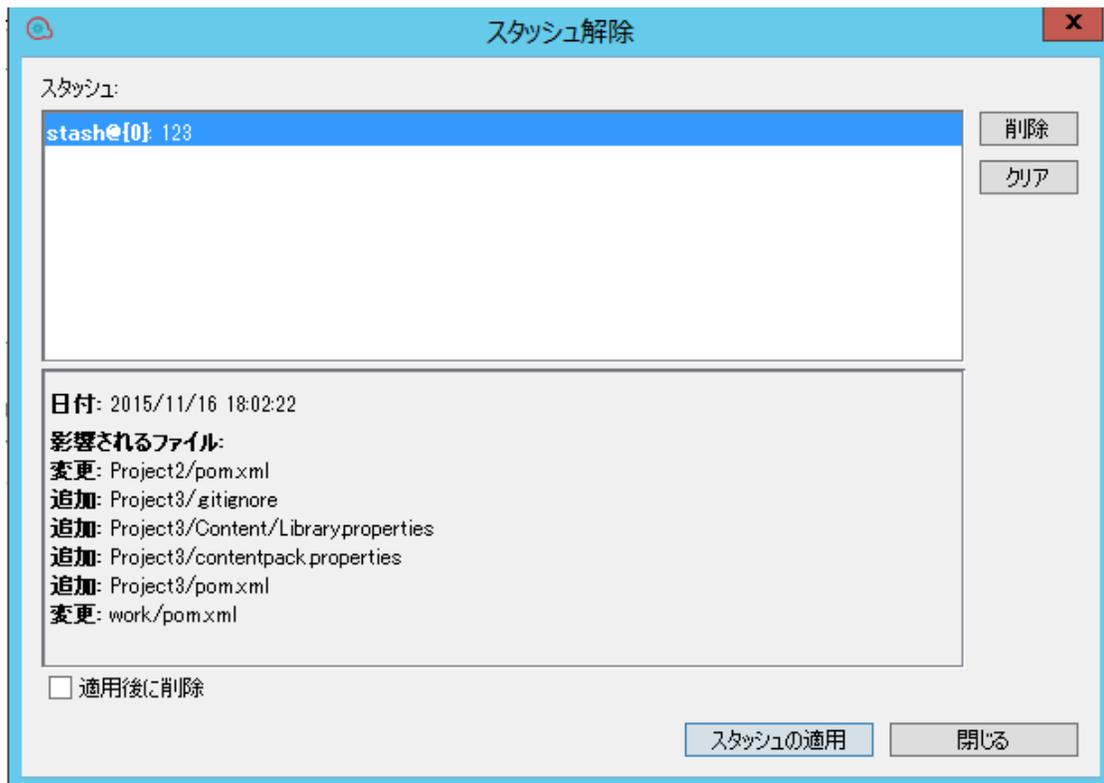
注: スタッシュ解除オペレーションの結果として競合が発生する可能性があるため、スタッシュ解除オペレーションを実行するときには注意してください。

作成者が、リモートリポジトリに置かれているアイテムと競合状態になっている変更をコミット解除した場合、スタッシュ解除オペレーション中に競合が発生します。

変更をスタッシュ解除するには:

1. スタッシュ解除するスタッシュコミットを選択します。
2. [SCM]メニューから [Git] > [スタッシュ解除] を選択します。

[スタッシュ解除] ダイアログボックスが開きます。



3. 現在選択されているスタッシュを削除するには、[ドロップ]を選択します。
4. すべてのスタッシュを削除するには、[クリア]を選択します。
5. 現在選択されているスタッシュを実装するには、[スタッシュの適用]を選択します。[適用後に削除]を選択した場合は、「Apply Stash」オペレーションにより、現在選択されているスタッシュも削除されます。

注: 作成者が、リモートリポジトリに置かれているアイテムと競合状態になっている変更をコミット解除した場合、ここで競合が発生する可能性があります。

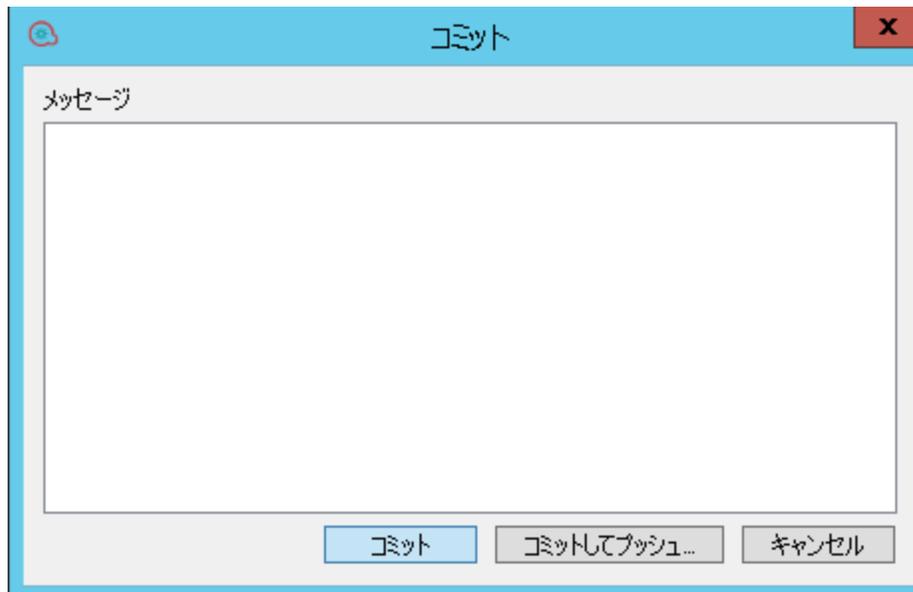
Git リポジトリに対するファイルのコミットとプッシュ

Studio で変更を行った後に変更を記録するには、ローカル Git リポジトリに対してそれらのファイルをコミットし、その後でそれらを Git サーバー上のリモート Git リポジトリにプッシュする必要があります。

変更のコミットを選択し、後でそれらをプッシュすることも、同じステップ内でコミットとプッシュを行うこともできます。

ベストプラクティス: 最新のファイルを確実に使用するには、コミットの前に、[SCM] > [更新] コマンドを使用してリモート Git リポジトリからファイルを更新する(プルする)ことをお勧めします。

1. プロジェクトツリーでローカルリポジトリに対してコミットするファイルが含まれているフォルダーを選択します。
2. 右クリックして、[SCM]>[コミット]を選択します。
[コミット]ダイアログボックスが開きます。



3. [コミット]をクリックして、ローカルリポジトリに対して変更をコミットします。
または:
[コミットしてプッシュ...]をクリックし、ローカル Git リポジトリに対して変更をコミットしてから、それらをリモート Git リポジトリにプッシュ (コピー) します。

注: プッシュの前に、SCM ワークスペース全体が最新の状態になっている必要があります。プッシュオペレーションは、ワークスペース内のすべてのコミットされたファイルをリモート Git リポジトリにプッシュします。コミットされていない変更がある場合、それらはリモートリポジトリにプッシュされません。

メインメニューから [SCM] > [Git] > [プッシュ..] オプションを使用して、変更をリモートリポジトリにプッシュすることもできます。詳細については、「[Git リポジトリに対するファイルのプッシュ](#)」(104ページ)を参照してください。

次のメッセージが [SCM メッセージ] ペインに表示されます。

```
Pushed 1 commit(s) to origin/master
```

注: Git は、空白のフォルダーをリポジトリに格納しません。しかし、Studio では、必要に応じて空白のフォルダーが自動的に作成されるので、プロジェクト内に空白のフォルダーを置くことができます。

Git リポジトリに対するファイルのプッシュ

Studio で変更を行った後に変更を記録するには、ローカル Git リポジトリに対してそれらのファイルをコミットし、その後でそれらを Git サーバー上のリモート Git リポジトリにプッシュする必要があります。

変更のコミットを選択し、後でそれらをプッシュすることも、同じステップ内でコミットとプッシュを行うこともできます。この手順では、リモート Git リポジトリに対して変更をプッシュする方法について説明します。

注: リモートサーバーに変更をプッシュすると、すべてのコミットがプッシュされます。

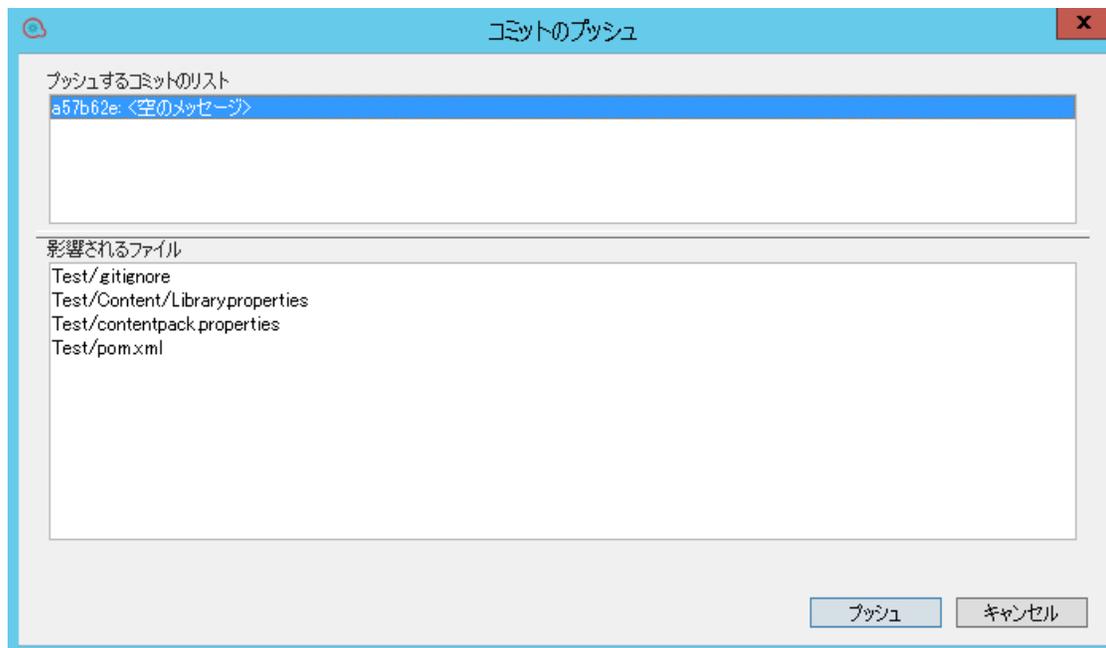
ベストプラクティス: 最新のファイルを確実に使用するには、コミットの前に、**[SCM] > [更新]** コマンドを使用してリモート Git リポジトリからファイルを更新する(プルする)ことをお勧めします。

1. メインメニューから **[SCM] > [Git] > [プッシュ...]** を選択します。

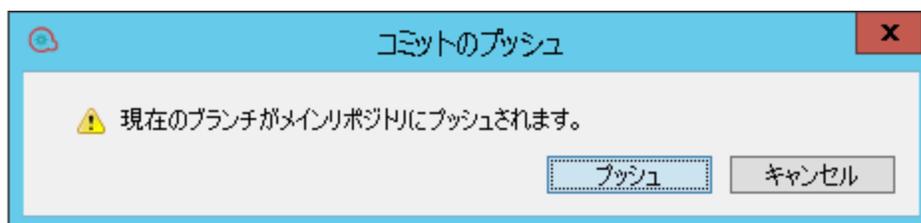
または:

[SCM 変更] ペインで、 をクリックします。

[コミットのプッシュ] ダイアログボックスが開き、プッシュされるコミット (およびコミット ID) のリストと、各コミットで影響を受けるファイルを集計したリストが表示されます。



ブランチがリモートリポジトリでプッシュされない場合、[プッシュ] ウィンドウは次のように表示されます。



2. **[プッシュするコミットのリスト]**でコミットを選択し、**[影響されるファイル]**セクションに影響されるファイルのリストを表示します。

注:

- プッシュの前に、SCM ワークスペース全体が最新の状態になっている必要があります。つまり、まだコミットまたは更新されていないファイルがワークスペース内に存在しないようにする必要があります。
- リモート Git リポジトリが空白のときに、ローカルコミットなしでリモートリポジトリにプッシュしようとする、次のエラーメッセージが表示されます。
「プッシュする前に、リポジトリを最初のコミットで初期化する必要があります」

影響されるファイルのリストが、**[影響されるファイル]**セクションに表示されます。

3. **[すべてプッシュ]**をクリックして、すべてのコミットオペレーションをリモートリポジトリにプッシュします。

確認メッセージが **[SCM メッセージ]** ペインに表示されます。例:

```
04/14/15 13:19:19 - Push  
Push successful  
Pushed 3 commit(s) to origin/master
```

マージまたはリベースの中止

マージ/リベース対象のブランチからの変更を受け入れたくない場合があります。リベースの場合は停止するのが一般的です。

リベースの場合は、リベースしている各ブランチからコミットを処理し、リベースしているブランチに順番に追加します。いくつかのコミットの後に問題が見つかった場合は、リベースをいったん停止してから、再開することができます。

マージの場合は、意図したとおりに解決できなかった競合がワークスペース内に存在する可能性があります。この場合、オペレーションの停止を選択できます。

1. SCM メニューから、**[Git] > [マージまたはリベースの中止]**を選択します。

または:



SCM 変更ツールバーで、**中止** を選択します。

確認メッセージが **[SCM メッセージ]** ペインに表示されます。

リポジトリ内の Git オペレーションの履歴の表示

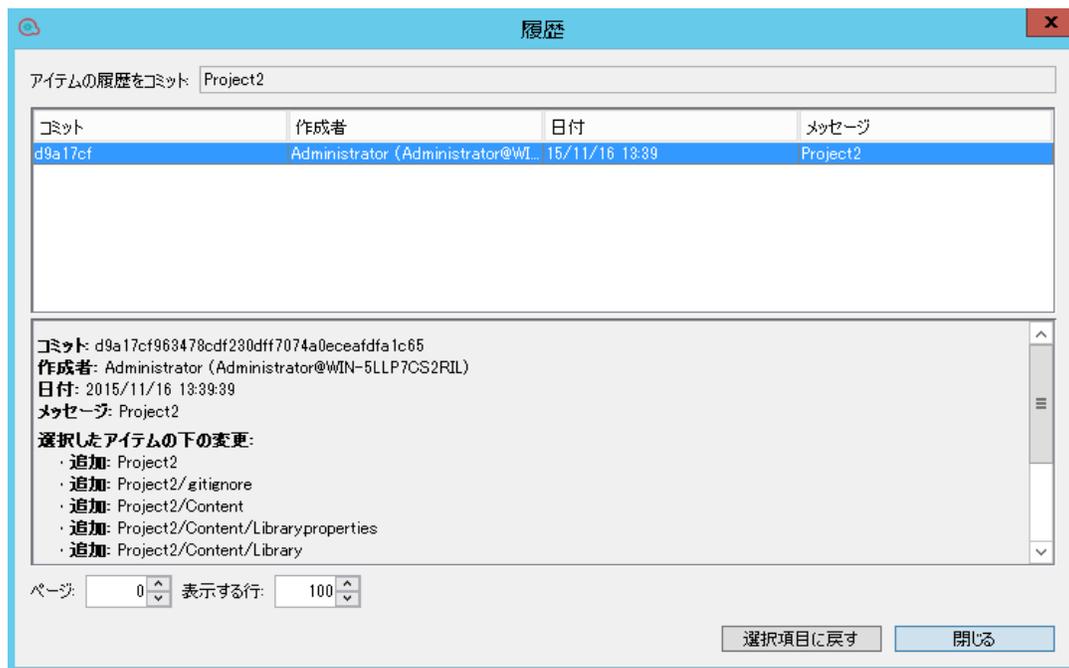
[履歴] オプションには次の情報が表示されます。

- 選択したプロジェクト/アイテムが影響を受けたコミットのリスト。
- 特定のコミットの影響を受けたすべてのファイル。

[履歴] オプションを使用して、プロジェクト/アイテムの特定のバージョンに戻すこともできます。詳細については、「[以前のバージョンのリポジトリに戻す](#)」(106ページ)を参照してください。

1. ファイル/フォルダーを右クリックし、[SCM] > [履歴] を選択します。

[リビジョン履歴] ダイアログボックスが開きます。



以前のバージョンのリポジトリに戻す

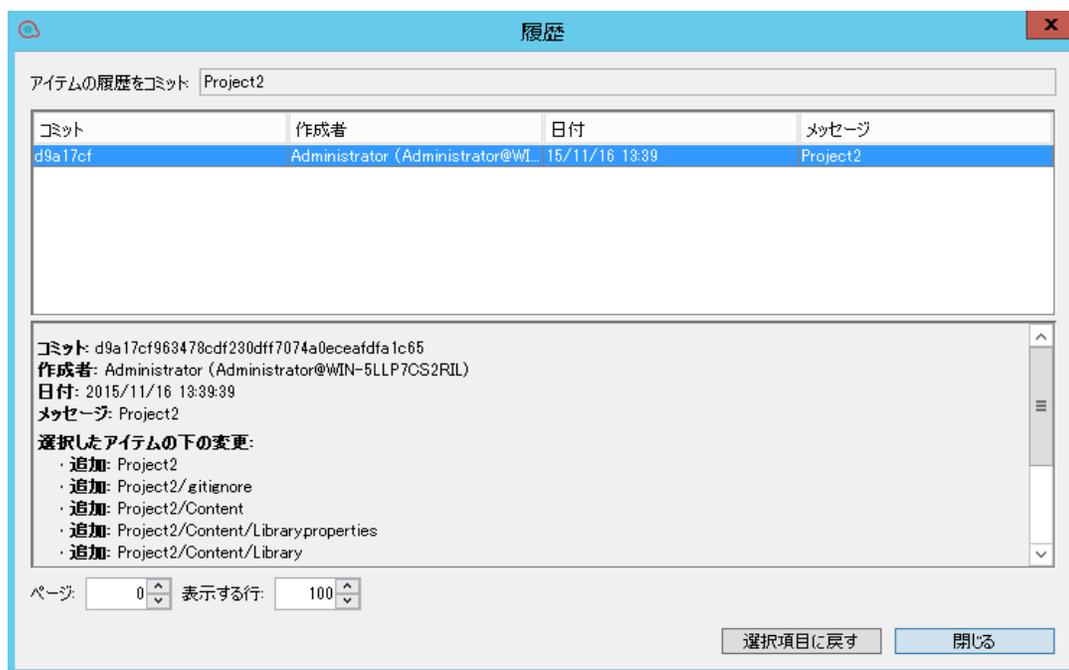
コンテンツパック内の特定のフローのトラブルシューティングのために、問題が発生した Central コンテンツパックの再作成が必要になることがあります。コンテンツパックが使用できない場合は (失われたか新しいバージョンで上書きされた場合)、[履歴] オプションを使用して特定のバージョンのコンテンツパックを復元できます。

注: Git では、リポジトリについてのみ [元に戻す] オペレーションを実行できます。

前のリビジョンに戻すには、次の手順を実行します。

1. ファイル/フォルダーを右クリックし、[SCM] > [履歴] を選択します。

[リビジョン履歴] ダイアログボックスが開きます。



2. 戻す対象のコミットのリビジョンを選択します。
3. [選択項目に戻す] をクリックします。

Git リポジトリから古いバージョンのファイル/フォルダーに戻す

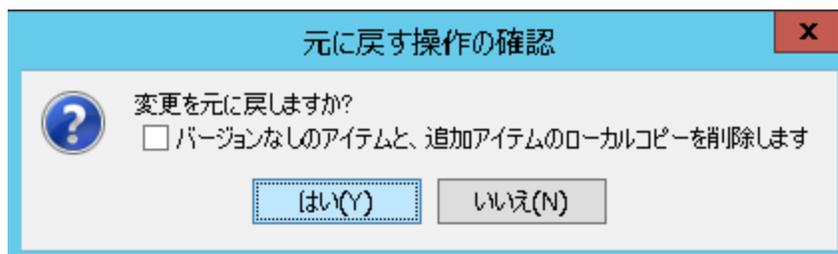
元に戻すと、作成者がアイテムに加えた変更が削除され(フォルダーの場合は、そこに含まれるすべてのアイテムが再帰的に削除されます)、元に戻されたアイテムのオリジナルの状態が、最後に更新されたときの状態に戻されます。

1. プロジェクトツリーから要求されたフォルダー/ファイルを選択します。
2. 右クリックして、[SCM] > [元に戻す] を選択します。

または:

SCM 変更ツールバーで、[すべての変更を元に戻す]  を選択します。

次のメッセージが表示されます。



3. すべてのバージョンなしのアイテムおよび追加されたアイテムのローカルコピーを削除する場合は、このチェックボックスをオンにします。
4. [はい] を選択すると、前のバージョンに戻ります。

プロジェクト/アイテムのタグ付きのコミットへのリセット

コンテンツパック内の特定のフローのトラブルシューティングのために、問題が発生した Central コンテンツパック/プロジェクトの再作成が必要になることがあります。コンテンツパックが使用できない場合は (失われたか新しいバージョンで上書きされた場合)、[タグにリセット...] オプションを使用して特定のバージョンのプロジェクト/アイテムを復元できます。

[タグにリセット...] オプションは、リポジトリを既にチェックアウトしていて、SCM タグからワークスペースを特定のバージョンに更新する必要がある場合に使用できます。

重要: ワークスペース全体がタグが付けられたときの状態に戻ります。

1. [SCM] メニューから [Git] > [タグにリセット...] を選択します。
2. Central 管理者から受け取った SCM タグを入力します。デフォルトの SCM タグは [Project Name]-[Project version] という形式になっています。

注:

- Git の場合、SCM タグに空白を含めることはできません。
- 現在のブランチ上にないタグにリセットしようとする、エラーメッセージが表示されます。

SCM リポジトリログの表示

SCM リポジトリログを表示するには:

1. 画面の下部にある [SCM リポジトリログ] タブを選択します。

リポジトリログが表示されます。例:

グラフと説明	作成者	日付
master HEAD 22	Administrator <Administrator@WIN-5LLP7CS2RIL>	2015-11-16 22:05:55
11	Administrator <Administrator@WIN-5LLP7CS2RIL>	2015-11-16 22:05:02
origin/master プロジェクトの Project2 プロパティをリリースのためにコミットしています。	Administrator <Administrator@WIN-5LLP7CS2RIL>	2015-11-16 17:33:26
Project2-1.0.2 プロジェクトの Project2 プロパティをリリースのためにコミットしています。	Administrator <Administrator@WIN-5LLP7CS2RIL>	2015-11-16 17:33:24
<空のメッセージ>	Administrator <Administrator@WIN-5LLP7CS2RIL>	2015-11-16 14:56:06
<空のメッセージ>	Administrator <Administrator@WIN-5LLP7CS2RIL>	2015-11-16 13:52:37
Project2	Administrator <Administrator@WIN-5LLP7CS2RIL>	2015-11-16 13:39:39
Project2	Administrator <Administrator@WIN-5LLP7CS2RIL>	2015-11-16 13:37:39

2. ローカルリポジトリとリモートリポジトリで異なるバージョンがマージされている場合は、上の例に示すように、リポジトリログに分岐とマージが2つの異なるパスとして表示されます。

Git リポジトリへのバージョンなしのファイルの追加

任意の段階でローカル Git リポジトリにバージョンなしのファイルを追加できます。プロジェクト全体、特定のフォルダー (「Library」または構成アイテムフォルダーの下のフォルダー)、フロー、オペレーション、または構成アイテムのファイルを追加できます。

注:

- 現在のワークスペースの外部に置かれているプロジェクトからファイルを追加することはできません。

- バージョンなしのプロジェクトからファイルを追加するときには、プロジェクト全体がリポジトリに追加されます。

追加したファイルは、[プロジェクト] ペインと [SCM 変更] ペインに緑色で表示されます。

1. 右クリックして、[SCM] > [追加] を選択します。
2. 確認メッセージが [SCM メッセージ] ペインに表示されます。

バージョン指定されたファイルの名前の変更

バージョン指定されたファイルの名前を変更すると、Git リポジトリでもファイル名の変更がスケジュールされます。

名前を変更したアイテムは、[SCM 変更] ペインで青で表示されます。

注: ファイルがバージョン指定されていても、ステータスが [追加されました] になっている場合は、名前の変更操作の後でも、[プロジェクト] ペインと [SCM 変更] ペインの両方でファイルの色が緑色になります。

オペレーション中に Git エラーが発生した場合は、[SCM メッセージ] ペインでエラーを表示できません。

1. 右クリックして [名前の変更] を選択します。
2. ファイルの新しい名前を入力し、Enter を押します。
確認メッセージが [SCM メッセージ] ペインに表示されます。

バージョン指定されたファイルの削除

バージョン指定されたファイルを削除すると、Git リポジトリでもファイルの削除がスケジュールされます。

削除したアイテムは、[SCM 変更] ペインでグレーで表示されます。ファイルを削除した後も、[SCM 変更] ペインから引き続きファイルをリポジトリに対してコミットすることができます。これは、複数の作成者がいる環境で作業する場合に便利です。

オペレーション中に Git エラーが発生した場合は、[SCM メッセージ] ペインでエラーを表示できません。

1. ファイルを右クリックし、[削除] を選択します。
ファイルがローカルリポジトリから削除され、[SCM メッセージ] ペインに確認メッセージが表示されます。
2. ここで、[SCM 変更] ペインからコミットして、リモート Git リポジトリからファイルを削除することができます。

Git リポジトリからのデタッチ

Git リポジトリでの作業が終了したら、リモート Git リポジトリからデタッチすることをお勧めします。

注: デタッチの前に、すべてのローカルの変更をリモート Git リポジトリにコミットしてプッシュしたことを確認してください。

オペレーション中に Git エラーが発生した場合は、[SCM メッセージ] ペインでエラーを表示できません。

1. [SCM] > [設定] を選択します。
2. [SCM リポジトリ] 領域で、[デタッチ] をクリックします。
デタッチする前にすべての変更をコミットしてプッシュすることを勧める情報メッセージが表示されます。
3. [OK] をクリックします。
ローカルリポジトリがリモート Git リポジトリからデタッチされます。

Git リポジトリでの新しいブランチの作成

新しい(ローカル) ブランチを作成すると、現在の HEAD ポインターと同じコミットを最初にポイントしているブランチを使用できます。つまり、新しいブランチには、ブランチを作成するときに Studio で追加されるプロジェクトが含まれます。新しく作成したブランチは、直ちにチェックアウトされ、Studio が新しいブランチに自動的に接続されます。その後のコミットはすべて現在のブランチに対して実行され、後でそのブランチをリモートリポジトリにプッシュすることを選択できます。

注: 初期化されていないリポジトリに接続されているときには、最初のコミットの後にのみローカルブランチが作成されます。

オペレーション中に Git エラーが発生した場合は、[SCM メッセージ] ペインでエラーを表示できません。

1. [SCM] メニューから [Git] > [新規ブランチの作成...] を選択します。
[新規ブランチの作成] ダイアログボックスが開きます。
2. 新しいブランチの名前をフィールドに入力して、[OK] をクリックします。

注: Git リポジトリ名は、標準の Git 命名規則に従い、スペースを含めないようにする必要があります。

ローカル Git リポジトリに新しいブランチが作成され、[SCM メッセージ] ペインにメッセージが表示されます。

3. 同じリモートリポジトリで作業している他の作成者がブランチを使用できるようにするには、[プッシュ] コマンドを使用して現在のブランチをリモートリポジトリにプッシュします。

Git リポジトリでのブランチのチェックアウト

ブランチをチェックアウトすると、既存のブランチに接続できます。既存のブランチに接続すると、新しくチェックアウトしたブランチの状態を反映してワークスペース(およびその中のプロジェクト)が変更されます。

注:

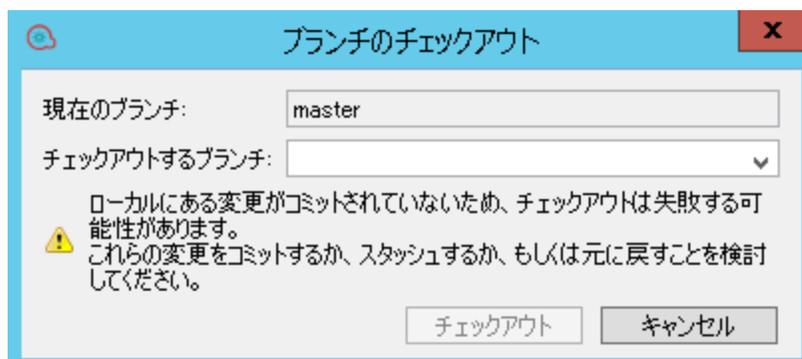
- 新しくチェックアウトをしたブランチ内に存在しているプロジェクトでも、現在のブランチ内に存在していないプロジェクトはすべて削除されます。ただし、引き続きそのブランチを後でチェックアウトできるので、プロジェクトのデータは失われません。
- 両方のブランチ内に存在するプロジェクトは、新しくチェックアウトしたブランチの状態を反映して変更および更新されます。
- 新しくチェックアウトしたブランチ内にもみ存在するプロジェクトは、ディスクにコピーされますが、Studio で自動的にインポートされません。
- ローカルで変更が加えられているブランチのチェックアウトは推奨されません。

ローカルで変更されている同じファイルが、チェックアウトされるブランチ内では異なっている場合、チェックアウトが失敗し、現在のブランチとローカルの変更が保持されます。

また、ブランチがチェックアウトされるとローカルの変更が失われます。

1. [SCM] メニューから [Git] > [既存の分岐のチェックアウト...] を選択します。

[現在のブランチ] ダイアログボックスが開きます。



2. [チェックアウトするブランチ] ドロップダウンから、使用するローカルブランチを選択します。
3. [チェックアウト] をクリックします。

ブランチがチェックアウトされ、すぐに使用できるようになります。リモートリポジトリ名のプレフィックスが付いたブランチのリストが表示されます。たとえば、<master>-newbranch のようになります。

注:

- ブランチがリモートリポジトリにまだプッシュされていない場合は、[ローカルブランチ] の見出しの下にのみ名前が表示されます。
- ローカルでチェックアウトされたことがないリモートブランチ (下の例のブランチなど) をチェックアウトする場合、同じ名前のローカルブランチが作成され、アップストリームがリモートブランチに構成されます。



スラッシュ記号はブランチがリモートリポジトリにあることを意味します。たとえば、**origin/** は、リモートリポジトリ上の origin という名前のブランチを示します。

右向き矢印は、ローカルリポジトリが以前にリモートリポジトリにプッシュされていて、リモートバージョンが存在することを示します。たとえば、**master** → **origin/master** は、ローカルブランチ **master** がリモートリポジトリ上の **origin** という名前のブランチに以前にプッシュされたことを示します。

リモートリポジトリから情報がまだ更新されていないために表示されない追加のリモートリポジトリが存在している可能性があります。この情報を更新するには、取得オペレーションを実行する必要があります。

チェックアウトの後に、[SCM メッセージ] ペインに次のメッセージが表示されます。

```
07/07/15 14:29:28 - Branching  
Successfully checked out branch feature1 with upstream  
origin/feature1.
```

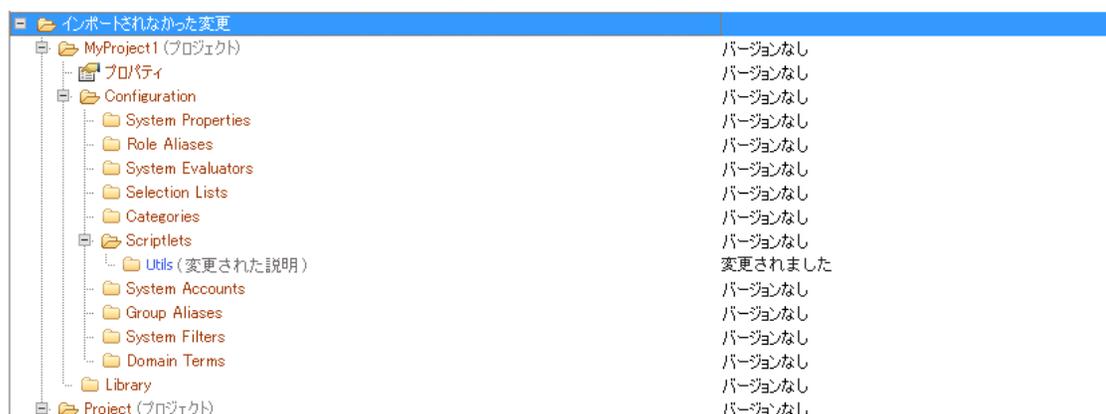
現在のワークスペース内のインポートされていないアイテムの表示

[SCM 変更] ペインで、現在のワークスペース内に存在していて変更されていても、Studio にインポートされていないアイテム (ファイル、フォルダー、またはプロジェクト) を表示できます。[SCM 変更] ペインに、リポジトリのローカルコピー (ワークスペース) と SCM リポジトリの間の変更が表示されます。ファイルが変更されていない場合は、ファイルは表示されません。

この情報は、競合がある場合、またはマージ/リベースを実行し、加えられたすべての変更を理解する必要がある場合に重要です。

インポートされていないアイテムは次のいずれかの状態になっている場合があります。

- **バージョンなし:** アイテムは、ワークスペース内で新しく作成され、Git リポジトリにまだプッシュされていません。
- **変更済み:** アイテムは既に Git リポジトリ内に存在し、変更されています。
 1. [SCM 変更] タブを選択します。
 2. アイテムツリーで、[インポートされなかった変更] ノードを選択します。
変更されたバージョンなしのファイルのリストが表示されます。例:



Item Name	Version Status
MyProject1 (プロジェクト)	バージョンなし
プロパティ	バージョンなし
Configuration	バージョンなし
System Properties	バージョンなし
Role Aliases	バージョンなし
System Evaluators	バージョンなし
Selection Lists	バージョンなし
Categories	バージョンなし
Scriptlets	バージョンなし
Utils (変更された説明)	変更されました
System Accounts	バージョンなし
Group Aliases	バージョンなし
System Filters	バージョンなし
Domain Terms	バージョンなし
Library	バージョンなし
Project (プロジェクト)	バージョンなし

インポートされなかったアイテム変更ノード内の任意のアイテムから次のオペレーションを実行できます。

- **エクスプローラーで表示:** アイテムを右クリックして[エクスプローラーで表示]を選択します。Windows エクスプローラーにアイテムのディレクトリが表示され、そのアイテムが強調表示されます。
- **プロジェクトのインポート (プロジェクトノードからのみ):** プロジェクトノード右クリックし、[プロジェクトのインポート]を選択します。選択したプロジェクトがインポートされ、ワークスペースツリーに表示されます。

Studio プロジェクト構造の下に含まれていない他のすべての変更されたアイテムは、Studio で使用できません。それら进行操作するには外部ツールが必要です。

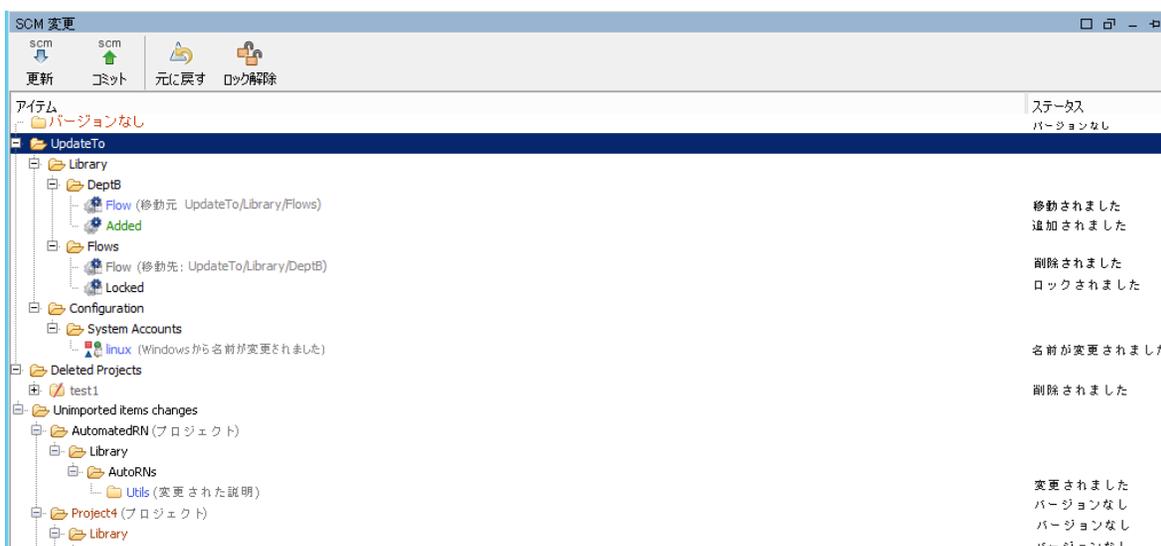
参考資料[SCM 変更] ペイン

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

注: Studio ワークスペースの外部にあるプロジェクトは SCM に追加/コミットされません。それらのプロジェクトは、[プロジェクト] ペインでバージョンなしとして (茶色で) 表示され、[SCM 変更] ペインには表示されません。

変更の種類

[ステータス] 列で、さまざまな種類の変更を確認できます。



変更の種類は、色の指定にも反映されます。次のような色になります。

ステータス	色
追加済み	緑
名前変更済み	青
変更済み	青
競合状態	赤
移動済み (移動元/移動先)	青
削除済み	グレー
バージョンなし	茶

さらに、アイテム名の後にアイテムのステータスを説明するテキストメッセージが括弧で囲まれて表示されます。たとえば、「**moved from <old location>, moved to <new location>, renamed from <old name>**」のように表示されます。これらのエントリからすべての標準の SCM 操作を実行することもできます。

この例では、次の情報を表示できます。

- **test1** という名前の削除済みのプロジェクト (グレーで表示)。
- **Added** という名前で追加されたフロー (緑で表示)。
- **Unversioned** という名前のバージョンなしフォルダー (茶で表示)。
- 変更され、インポートされていないアイテムのリスト。
- いくつかの変更されたアイテム (青で表示)。

- 変更されたフロー (**adding flow 2**)。
- 名前が変更されたフォルダー (**new folder a**)。
- 移動されたフロー (**flow 1234567**)。
- 説明が変更されたフォルダー (**Utils**)。

注: 名前変更および移動されたアイテムには、アイテムの移動元または変更前の名前を示す特別なラベルが括弧で囲まれて表示されます。

状態

[状態] 領域には、現在の Git ブランチの状態と名前が表示されます。

デフォルトでは、現在の Git ブランチは「マスター」です。

次のいずれかの状態になります。

- **準備完了:** デフォルトの状態
- **マージ中:** マージオペレーションの実行中で、まだ解決されていない競合が存在します。
- **マージ中 - すべての競合が解決済み:** マージオペレーションの実行中で、すべての競合が解決されています。
- **インタラクティブリベース:** リベースオペレーションの実行中で、競合が存在しました。

SCM 変更ツールバー

GUI アイテム	説明
	すべて更新: Studio のワークスペース全体を更新します。
	すべての変更をコミット: [SCM 変更] ペインですべての変更をコミットします。変更があった場合にのみ使用可能です。
	変更をプッシュ: [SCM 変更] ペインのすべての変更をメイン Git リポジトリにプッシュします。
	スタッシュ: ローカル作業ディレクトリの現在の状態 (つまり、変更した追跡されているファイルおよび段階的な変更) を取得し、いつでも再適用できる未完了の変更のスタックに保存します。
	スタッシュ解除: 未完了の変更のスタックをスタッシュから取得し、それらのローカル作業ディレクトリに再適用します。

	すべての変更を元に戻す: [SCM 変更] ペインで表示されるすべての変更を元に戻します。
	ブランチ: Git ブランチオペレーションを表示します。アイコンをクリックすると、追加のオプション ([作成]、[チェックアウト]、[マージ]、[リベース]) が表示されます。
	作成: 新しい Git ブランチを作成します。
	チェックアウト: 既存のブランチに接続します。チェックアウトすると、新しくチェックアウトしたブランチの状態を反映してワークスペース (およびその中のプロジェクト) が変更されます。
	マージ: Git がローカルコミットの元の階層を保持し、2つの分岐した「ブランチ」 (ローカルブランチとリモートブランチ) を再びマージする特別な「マージ」コミットを作成します。
	リベースの続行: オリジナルのブランチ内に各コミットに対応する新しいコミット作成することによってプロジェクトの履歴を作成し直します。
	リベース中のコミットのスキップ: 新しいコミットを作成せずにプロジェクトの履歴を作成し直します。
	中止: マージ (マージオペレーションの場合) を中止するか、プロジェクトの履歴の再作成 (リベースオペレーションの場合) を停止します。
	更新: [SCM] 変更ペイン内のすべてのプロジェクト/アイテムを更新します。

状態:	<p>現在のリポジトリの名前 (master など) およびリベースの段階を表示します。</p> <p>空白のリポジトリに接続した場合、(<empty>) というテキストが表示されます。例:</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin: 5px 0;">状態: 🔄 準備完了 (master)</div> <p>リベース中には、インタラクティブリベースというテキストが表示されます。例:</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin: 5px 0;">状態: 🔄 インタラクティブリベース (master)</div> <p>マージ中には、マージ中というテキストが表示されます。例:</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin: 5px 0;">状態: 🔄 マージ中 (master)</div> <p>すべての競合が解決された後にマージを実行すると、マージ中 - すべての競合が解決済みというテキストが表示されます。例:</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin: 5px 0;">状態: 🔄 マージ中 - すべての競合が解決済み (master)</div> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin: 5px 0;"><p>注: マージ処理を完了するには、最終的なコミットを実行して、ファイルをコミット準備完了状態に移行してメイン Git リポジトリにプッシュする必要があります。</p></div>
------------	--

[SCM メッセージ] ペイン

[SCM メッセージ] ペインには、Git 操作の結果メッセージが表示されます。すべての操作の結果としてメッセージが表示され、色が指定されます。さらに、Git 操作の結果をユーザーに知らせるポップアップメッセージが表示されます。[SCM メッセージ] ペインをクリアするには、ペイン内を右クリックし、[すべてクリア] ボタンをクリックします。

表示されるメッセージの種類は、更新方式、スタッシュ、予想される競合、特定のコマンドを実行したときにエラーが発生したかどうかなどの多くの要因によって決まります。

Git メッセージの例

次の例では、現在のワークスペースが更新されます。この処理には、スタッシュ、更新、ファイルの追加が含まれます。

SOM メッセージ

15/11/19 03:24:57 - 更新
ワークスペースの更新を開始しました

15/11/19 03:25:41 - スタッシユ
ワークスペース内の変更を未追跡のファイルを含めてスタッシユしています。
- スタッシユされたパス:
 変更: work/pom.xml

15/11/19 03:25:19 - 追加 C:\Users\Administrator\oo\Workspace\Test

エラーメッセージ

エラーが表示される場合もあります。エラーは赤で表示されます。この例では次のように表示されず。

05/07/15 10:55:16 - Push
http://admin@address.labs.company.com:8180/r/test01.git: cannot open git-receive-pack
There were 1 error(s).

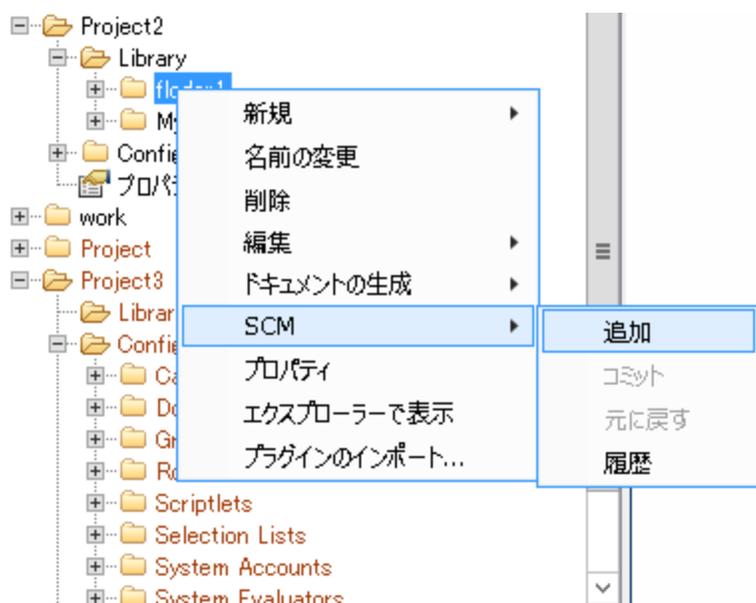
[プロジェクト] ペイン

作業中のプロジェクトと、編集可能なフロー、オペレーション、プロジェクトで使用可能なその他の HP 00 オブジェクトが表示されます。

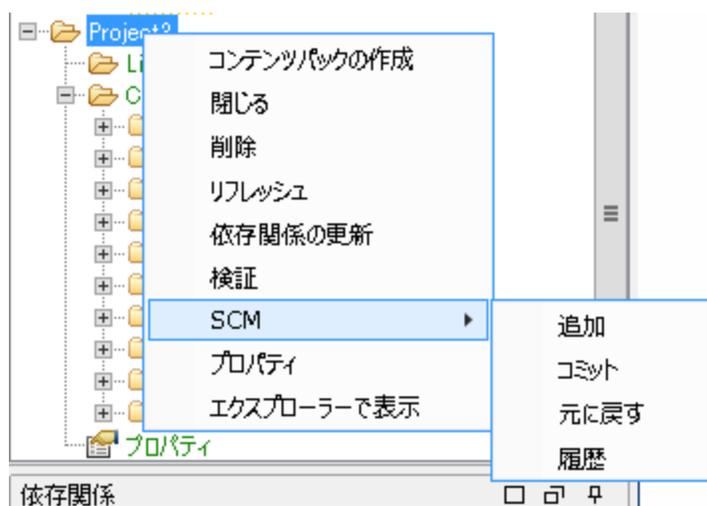
[プロジェクト] ペインのコンテキストメニュー

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

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



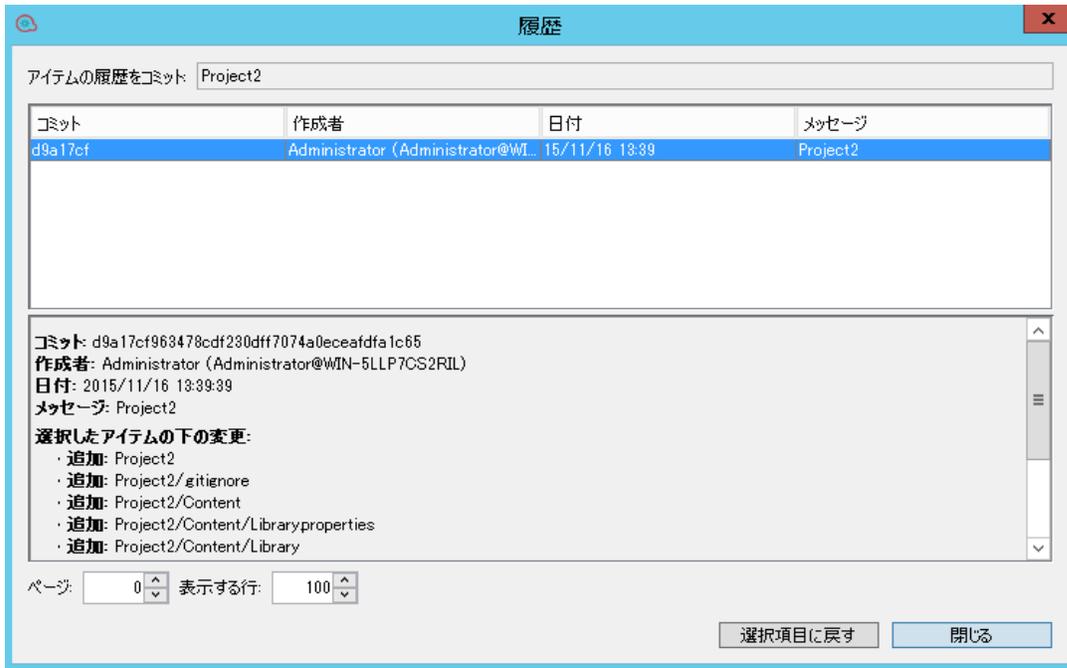
作成者がオブジェクトを変更する場合 (ステップを追加するなど)、ローカルで変更されます。この場合、[コミット] および [元に戻す] が使用可能です。



オプション	説明
追加	ソースコントロールに追加するアイテム(フロー、オペレーション、構成アイテムまたはフォルダー)をマークします。Studio の [追加] では、先祖および子孫オブジェクトがすべて含まれます。つまり、フォルダーが追加されると、子フローと親フォルダーもすべて追加されます。兄弟アイテムは追加されません。Studio 内で作成されたアイテムは自動的に追加され、次のコミットでコミットされます。
コミット	ローカルでの変更をサーバーにコミットします。このオプションは、変更されたアイテムや子アイテムが変更されたフォルダーで使用できます。コミットは再帰的に行われるため、フォルダーをコミットすると、その子アイテムもすべてコミットされます。コミット後、コミットのコメントを追加できます。
元に戻す	選択したアイテムのローカルでの変更をすべて元に戻します。フロー、オペレーション、構成アイテムでの変更が元に戻ります。削除されたアイテムが復元されます。 重要: アイテムを追加すると自動的に追加済みのマークが付きます。それに対し変更を元に戻すと、追加マークが削除されますが、アイテムは削除されず、Studio とファイルシステムに存在したままになります。メニューの [追加] を使用してアイテムを再び追加するか、アイテムを削除することができます。
履歴	[リビジョン履歴] ウィンドウには、SCM 履歴が表示されます。「 リポジトリ内の Git オペレーションの履歴の表示 」(106ページ)を参照してください。

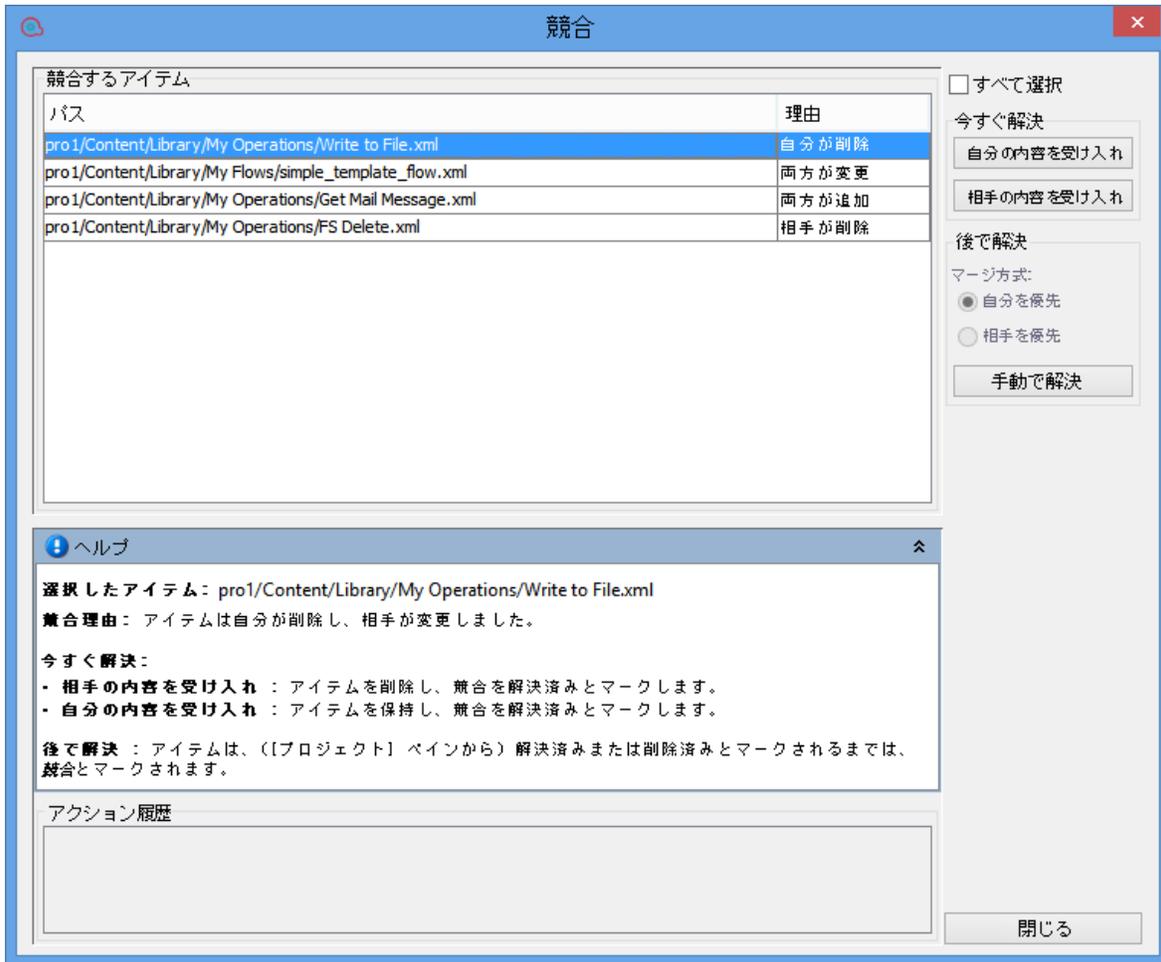
[履歴] ダイアログボックス

[履歴] ダイアログボックスを使用して、プロジェクト/アイテムの以前のリビジョンに戻します。

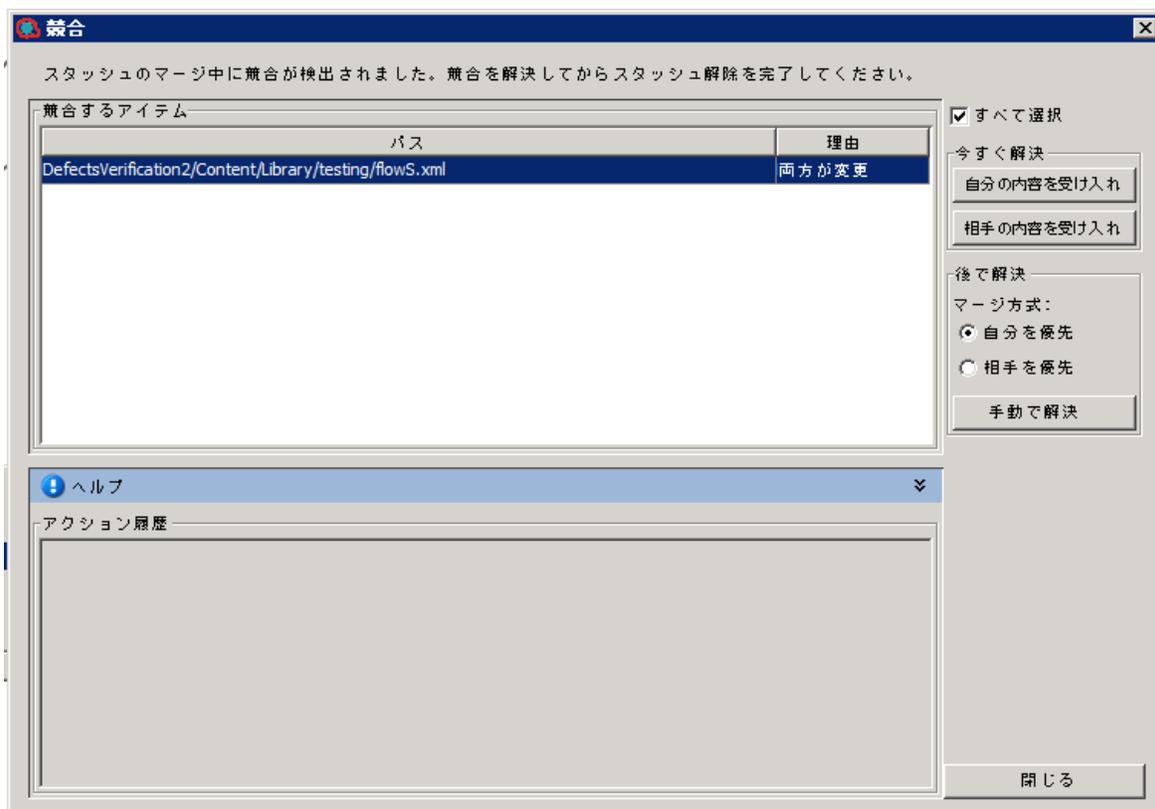


オプション	説明
アイテムの履歴をコミット	プロジェクト/アイテムの名前。
コミット	コミットオペレーションの ID。
作成者	変更をコミットしたユーザーのユーザー名/ID。
日付	コミットオペレーションが実行された日付。
メッセージ	コミットオペレーションが実行されたときに書き込まれたコミットメッセージ。
選択したアイテムの下の変更:	このコミットで実行された変更および変更されたアイテムのリストを。
ページ	リビジョン履歴が1つのページに表示できないほど長い場合は、ページの数および各ページに表示される行数を変更できます。
表示する行	各ページに表示する行数を入力します。
選択項目に戻す	[選択項目に戻す] をクリックすると、プロジェクト/アイテムの選択したコミットがロードされます。

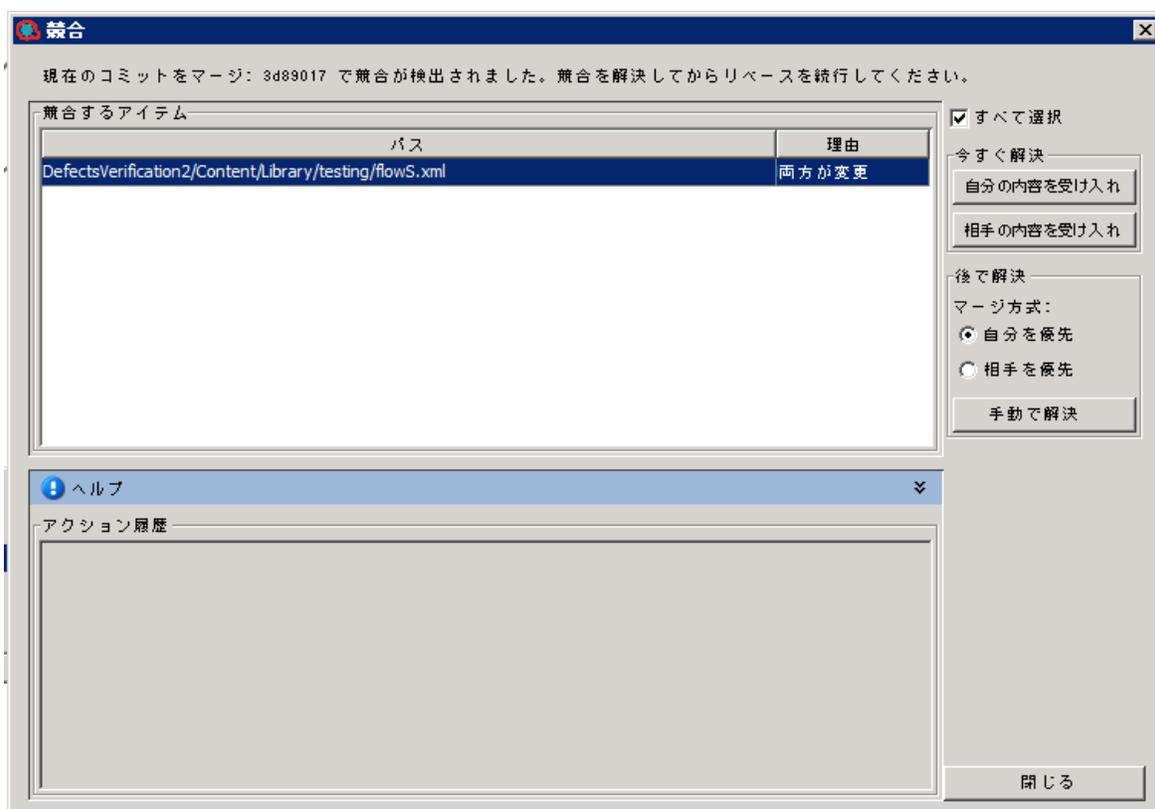
[競合] ダイアログボックス マージの例



スタッシュの例



リベースの例



オプション	説明
競合するアイテム	競合の原因になったプロジェクト内のアイテムおよびそれらのパスのリスト。各アイテムについて、競合の理由が表示されます。
すべて選択	リストからすべての競合アイテムを選択します。
今すぐ解決:	競合が直ちに解決されます。
自分の内容を受け入れ	「自分」が行った変更を直ちに適用します。
相手の内容を受け入れ	「相手」が行った変更を直ちに適用します。
後で解決:	競合は後で解決されます。
手動で解決:	オリジナルのファイルは、[プロジェクト]ペインから解決または削除されるまで競合としてマークされます。
自分を優先	オリジナルファイル内の競合に対して自分が行った変更が適用されます。
相手を優先	オリジナルファイル内の競合に対して相手が行った変更が適用されます。

オプション	説明
ヘルプ	選択したアイテムに関する追加の詳細が表示される [ヘルプ] ペインを表示します。
アクション履歴	すべての実行済みのオペレーションのログを表示します。たとえば、解決済みのアイテム、それらの解決方法、および手動で解決されるアイテムが表示されます。 スクロールバーを使用してログをスクロールすることができます。

後で解決

後で競合を解決することを選択した場合は、Studio で手動で競合を解決できるようにファイルが準備されます。競合していない変更はファイルに自動的にマージされます。競合している変更は選択した設定 (自分または相手) を基にして置き換えられます。

オリジナルのファイルは、[プロジェクト] ペインから解決または削除されるまで競合としてマークされます。

さらに、<filename> - ours と <filename> - theirs という 2 つのファイルが自動的に作成されます。これらはそれぞれ自分および相手のオリジナルバージョンを表します。

競合を解決した後で、「自分」および「相手」のファイルが自動的に削除されます。

手動で解決

競合を手動で解決するには:

1. 右側のペインで [自分を優先] を選択して、「自分」が行った変更を適用します。

コンテンツパックの使用

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

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

コンテンツパックのプロジェクトへのインポート



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

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

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

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

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

注: ネットワークドライブからコンテンツパックをインポートできます。

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

フローは別のプロジェクトから再利用することができます。そのフローで作成したコンテンツパックをインポートします。再利用するフローのコンテンツパックへのパッケージングについては、「[コンテンツパックのエクスポート](#)」(336ページ)を参照してください。

コンテンツパックは、次の複数のソースからダウンロードできます。

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

実行する操作

コンテンツパックのダウンロード

1. HPLN のサイトを開き、Operations Orchestration Community ページ <https://hpln.hp.com/group/operations-orchestration> に移動し、コンテンツパックのリストに移動します。
2. ベースコンテンツパックを、ネットワークドライブ上の場所にダウンロードします。
3. 指定したドメインで必要な別のコンテンツパックを、ネットワークドライブ上の場所にダウンロードします。

Studio の最初の起動時のベースコンテンツパックのインポート

1. **[依存関係]** ペインで、**[インポート]** ボタン  を選択します。

注: コンテンツパックは、**[依存関係]** ペインの **[ツリービュー]** タブまたは **[マージビュー]** タブからインポートできます。

2. コンテンツパックの場所を参照して **[開く]** をクリックします。
3. **[OK]** をクリックします。

注: ベースコンテンツパックをインポートするには数分かかることがあります。

コンテンツパックのインポート

1. **[依存関係]** ペインで、**[インポート]** ボタン  をクリックするか、**[ファイル]** メニューから **[コンテンツパックのインポート]** を選択します。
2. コンテンツパックの場所を参照して **[開く]** をクリックします。

注: 必要に応じて、複数のコンテンツパックを一度にインポートできます。

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

セキュリティ上の注意: Studio は、デジタル署名を検証しません。そのため、デジタル署名を Studio にインポートする前によく知られているツールを使用して、デジタル署名を手動で検証することをお勧めします。

コンテンツパックを解凍してプロジェクトとして開く

旧バージョンの HP 00 からコンテンツをアップグレードする場合、アップグレードツールでコンテンツをコンテンツパックに変換します。次の手順に従って、コンテンツパックを編集可能なプロジェクトとして開きます。

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

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

プロジェクトのコンテンツパックと依存関係の管理

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

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

注: これは、応答または結果を追加する場合のみ推奨されます。入力のみを使用する場合、ステップの内部でそれらを変更できます。

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

依存関係

依存関係は、現在のプロジェクトに含まれているアイテムの参照を含むプロジェクトまたはコンテンツパックです。次の1つ以上のアイテムを使用できます。

- 他のコンテンツパック/プロジェクト内のオペレーションを参照するオペレーションのソフトコピー
- 他のコンテンツパック/プロジェクトから取得されるオペレーショングループのエイリアス
- 他のコンテンツパック/プロジェクトから取得されるオペレーション/フロー/スクリプトレット
- 他のコンテンツパック/プロジェクトから取得されるオペレーション/フロー/出力フィルター
- 他のコンテンツパック/プロジェクトから取得されるオペレーションレスポンスルール評価子
- 他のコンテンツパック/プロジェクトから取得されるオペレーションレスポンスルールフィルター
- 他のコンテンツパック/プロジェクト内のシステムアカウント、選択リスト、ドメインタームからの[未指定時]の割り当てがあるオペレーション、フローまたはステップ入力。
- 他のコンテンツパック/プロジェクト内の入力評価フォーマットおよび入力レコードからそれぞれ取得されるオペレーション、フローまたはステップ入力評価子、ドメインターム。
- 他のコンテンツパック/プロジェクトのオペレーション/フローを参照するフローステップ
- 他のコンテンツパック/プロジェクトからのステップ結果フィルター
- 他のコンテンツパック/プロジェクトからのトランジションの役割のエイリアス

依存関係は、プロジェクトの pom.xml ファイルに保存されます。

依存関係の競合

一般的に、宣言された依存関係のバージョンは、Studio によって自動的に検出される依存関係のバージョンと一致している必要があります。

ただし、依存関係間に次の3つの種類の競合が発生する場合があります。

- **新規競合** - 依存関係がプロジェクト内で宣言されていないときに、Studio でこの依存関係が使用されていることが検出された場合に表示されます。このような競合解決すると、現在のワークスペース内に存在するバージョンとの依存関係が追加されます。
- **更新された競合** - 検出されたバージョンが宣言されたバージョンと異なる場合に表示されます。この競合は、検出されたバージョンが宣言されたバージョンの範囲外にある場合のみ発生します。たとえば、宣言されたバージョンの範囲が 1.2 ~ 1.6 で、検出されたバージョンが 1.3 である場合、1.3 は 1.2 ~ 1.6 の範囲内なので競合はありません。宣言されたバージョンが固定バージョン 1.5 で、検出されたバージョンが異なる固定バージョン 1.6 である場合、競合が存在します。[依存関係の競合の解決] ダイアログボックスでそのような行の [変更の適用] チェックボックスがオンになっている場合、そのプロジェクトの依存関係のリスト内で、[検出されたバージョン] 列から取得したバージョンで依存関係が更新されます。
- **使用されない可能性あり** - プロジェクト内で依存関係が宣言されても、Studio でそれが使用されていることが検出されない場合に表示されます。たとえば、プロジェクト ProjectA で依存関係として追加されているプロジェクト ProjectB があっても、ProjectA のフロー、オペレーション、構

成アイテムでそれが使用されていない場合があります。または、ProjectA 内のフローから参照されている ProjectB 内のシステムプロパティの動的な参照を使用する場合があります。どちらの場合でも、ProjectB の依存関係は **[使用されない可能性あり]** の競合としてタグ付けされます。

この場合、**[変更の適用]** チェックボックスがオンになっていると、この依存関係は現在のプロジェクトの依存関係リストから削除されます。

プロジェクト内の依存関係の更新

プロジェクトに依存関係が含まれている場合、次の「プロジェクトの依存関係の更新」の説明に従ってすべての依存関係を手動で更新する手順を最初に行う必要があります。その後で、依存関係を検証し、依存関係エディターで依存関係を追加、編集、または削除することができます。

実行する操作

コンテンツパックのオブジェクトをプロジェクトにコピーする

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

ヒント: **[依存関係]** ペインから **[プロジェクト]** ペインにオブジェクトをドラッグアンドドロップすることもできます。

コンテンツパックを削除する

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

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

1. **[依存関係]** ペインで **[コンテンツパック]** タブを選択します。
2. コンテンツパックを選択して、**[削除]**  ボタンをクリックします。
3. 確認のダイアログボックスで **[はい]** をクリックします。

コンテンツパックを閉じる

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

以下の場合にコンテンツパックを閉じます。

- ワークスペースに 2 つのバージョンのコンテンツパックが配置されており、一方で作業するにはもう一方を閉じる必要がある場合。2 つのコンテンツパックの UUID は同じであるため、Studio で同時に開くことはできません。

- コンテンツパックからプロジェクトを作成し、プロジェクトで作業するには元のコンテンツパックを閉じる必要がある場合。プロジェクトとコンテンツパックの UUID は同じであるため、Studio で同時に開くことはできません。
1. [依存関係] ペインで [コンテンツパック] タブを選択します。
 2. 閉じるコンテンツパックを選択します。
 3. コンテンツパックを右クリックして [閉じる] を選択します。

注: [依存関係] ペインで [閉じる]  ボタンをクリックすることもできます。

閉じたコンテンツパックを開く

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

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

注: [依存関係] ペインで [開く]  ボタンをクリックすることもできます。

インポートしたコンテンツパックのプロパティを表示する

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

コンテンツパック内のオブジェクトのプロパティを表示する

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

プロジェクトの依存関係の更新

依存関係の更新に使用できる 3 つの方法があります。

1. プロジェクト名を右クリックし、メニューから [依存関係の更新] を選択します。
- または:
2. [プロジェクト] ペインで、[プロパティ] を右クリックします。
 3. メニューから [依存関係の更新] を選択します。

または:

依存関係エディターのツールバーで、 をクリックします。

Studio で新しい依存関係が検出された場合は、その依存関係に対応する新しい行が追加されません。依存関係が既に存在している場合は、依存関係が更新されます。

注: 更新の場合は、Studio によって検出された依存関係と競合しない限り現在のバージョンが保持されます。

4. 現在のバージョンと Studio によって検出されたバージョンの間に競合がある場合は、[依存関係

の競合の解決] ダイアログボックスが表示されます。



- 競合を解決するには、各行の [変更の適用] をチェックボックスをオンにします。

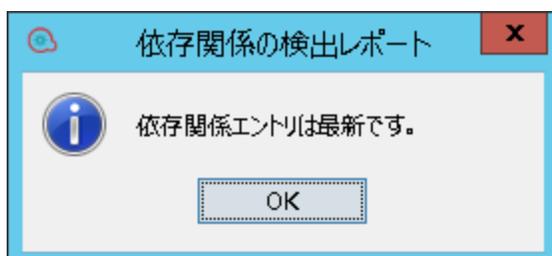
注: 依存関係が {使用されない可能性あり} としてマークされている場合、その依存関係がプロジェクト内で宣言されていても、Studio でその依存関係が使用されているかどうか識別できないことを意味します。たとえば、別のプロジェクト内のシステムプロパティを動的に参照している入力フロー内に存在する場合があります。この場合、Studio はそのシステムプロパティがフロー内で使用されているかどうかを識別できません。

これらの依存関係の場合、[変更の適用] チェックボックスがデフォルトでオフになります。[変更の適用] チェックボックスをオンにして [OK] をクリックすると、現在のプロジェクト内の依存関係リストからその依存関係が削除されます。

- [OK] をクリックします。

[OK] をクリックすると、Studio によって ([変更の適用] 列でマークされたとおりに) 選択した依存関係の適用 (更新、追加、削除) が実行されます。

競合がない場合は、すべての競合が解決された後に、次のメッセージが表示されます。



プロジェクトの依存関係の検証

ここでは、依存関係を手動で検証する方法について説明します。ただし、次のように依存関係の検証が自動的にトリガーされるケースもあります。

- プロジェクトの依存関係が変更され、保存されたとき
- プロジェクトが、個々のプロジェクトとして、または[すべて更新]操作の一環としてSCMから更新されたとき
- プロジェクトがインポートされた後
- プロジェクトが開かれたとき

注: 検証プロセスでは、プロジェクトの依存関係アイテムの範囲が確認され、さらに依存プロジェクトのバージョンがプロジェクトのバージョンと一致しているかどうかも確認されます。

依存関係を手動で検証するには:

- [プロジェクト] ペインで、[プロパティ] を右クリックします。
- メニューから [検証] を選択します。

プロジェクト内に依存関係の警告またはエラーがある場合は、[問題] ペインにそれらが表示されま
す。例:

種類	ソースの種類	名前	説明	場所
警告	依存関係	work	プロジェクトコンテンツパック work に対する Base [1.4.9-SNAPSHOT] 依存関係は	/work
警告	依存関係	work	プロジェクトコンテンツパック work に対する Middleware [1.2.0-SNAPSHOT] 依存...	/work
エラー	依存関係	work	プロジェクトコンテンツパック work に対する Base [1.4.9-SNAPSHOT] 依存関係は	/work
エラー	フロー	Basic Notify (2)	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Notification Opt...	/oo10-base-cp-1.6.2/Library/Utility Operations/Deprecated/Basic Notify (2)
エラー	フロー	Fine with WMI	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Network - Proto...	/oo10-base-cp-1.6.2/Library/Operations/Network/Deprecated/Fine with WMI
エラー	フロー	Check Process By Command Line	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Remote Comma...	/oo10-base-cp-1.6.2/Library/Operations/Operating Systems/Solaris/Process ...
エラー	フロー	Log Analyze	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Linux Log Sever...	/oo10-base-cp-1.6.2/Library/Operations/Operating Systems/Linux/Logs/Sam...
エラー	ステップ	Get Inactive Users	このステップがリンクしているオペレーションに問題があります	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Notification Opt...	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このステップがリンクしているオペレーションに問題があります	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Notification Opt...	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このステップがリンクしているオペレーションに問題があります	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Notification Opt...	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このステップがリンクしているオペレーションに問題があります	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...

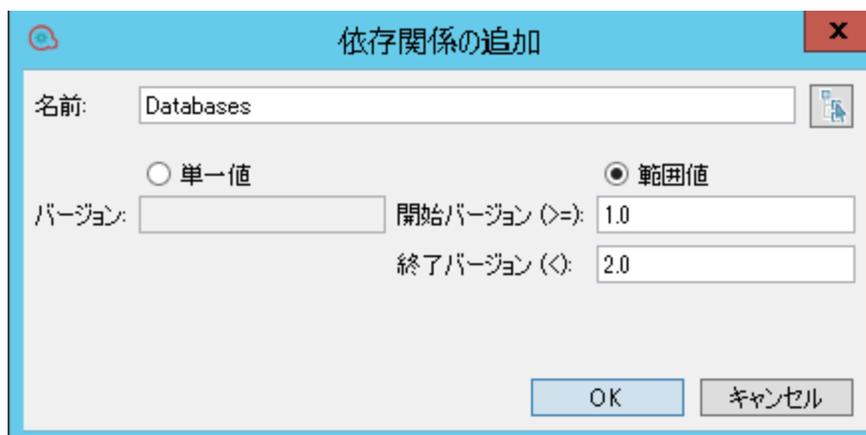
検証の後で、[問題] ペインが依存関係の新しいステータスになるように更新されます。

プロジェクトの新しい依存関係の追加

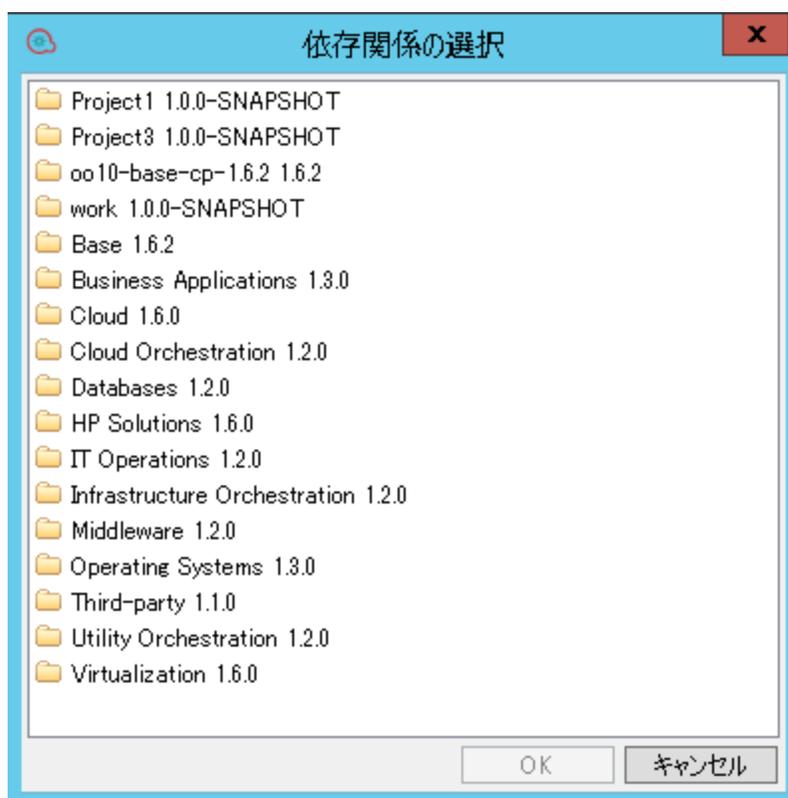
プロジェクトに自動的に追加される依存関係に加えて、新しい依存関係を追加できます(それらが
ワークスペースに存在していない場合も可能です)。

- 依存関係エディターのツールバーで  をクリックします。

[依存関係の追加] ダイアログボックスが開きます。



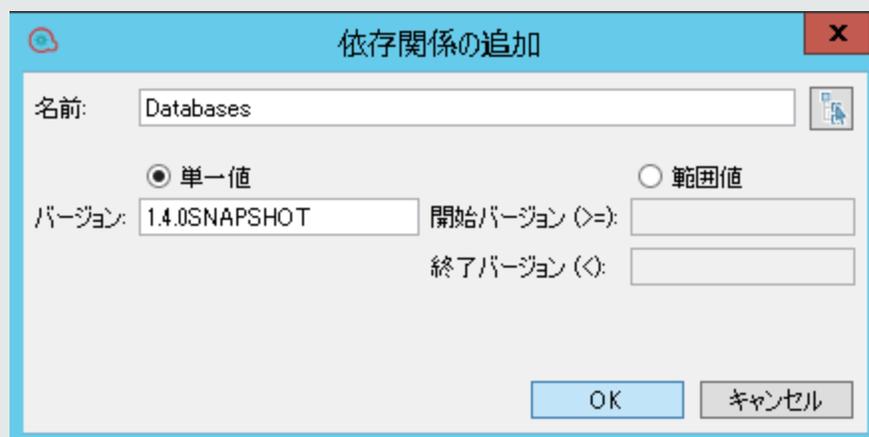
2. [名前] フィールドに、追加する依存関係の名前を入力するか、 をクリックして、現在のプロジェクトおよび Studio にインポートされているコンテンツパックのリストから依存関係を選択します。



たとえば、プロジェクトで Middleware コンテンツパックからの情報を使用している場合は、 をクリックし、タイプ **Middleware 1.2.0-SNAPSHOT** を選択します。

3. [バージョン] フィールドに、追加する依存関係のバージョンを入力します。数値 (最大 99999999) またはスナップショットのバージョン (1.4.0-SNAPSHOT など) を使用できます。

注: 依存関係のリストから選択した場合は、[バージョン]フィールドにバージョン番号が自動的に入力されます。例:



さらに、公式の現在のバージョンがない場合は、スナップショットのバージョンが使用されます。

4. 依存関係でサポートされるバージョンの範囲を定義するには、[範囲値]を選択し、[元]および[先]フィールドに範囲を入力します。たとえば、現在 Base バージョン 1.4.0 を使用しているときに、バージョン 1.2.0 との下位互換性をプロジェクトに持たせる必要がある場合は、[元]フィールドに「1.2.0」と入力し、[先]フィールドに「1.4.0」と入力します。

注: [元]フィールドでは、入力した値およびそれより大きい値が含まれます。[先]フィールドでは、入力した値は含まれません。たとえば、「1.2.0 ~ 1.4.0」は 1.2.0 以上で 1.4.0 未満を意味します。

[先]フィールドにのみ値を入力することでその値未満のすべての値を指定することも、[元]フィールドにのみ値を入力することでその値以上のすべての値を指定することもできます。

依存関係エディターで、範囲は (範囲) というマークが付けられます。

注: インラインエディターでは、[元]フィールドに範囲を入力することで範囲を自動的に定義できます。例: 1.2.0 - 1.4.0。

依存関係の編集

1. 依存関係エディターのツールバーで、編集する依存関係を選択します。
2.  をクリックします。
3. 必要に応じて依存関係の定義を編集します。

依存関係の削除

プロジェクトから依存関係を削除できます。

注: 依存関係はプロジェクトからのみ削除され、ファイルシステムからは削除されません。

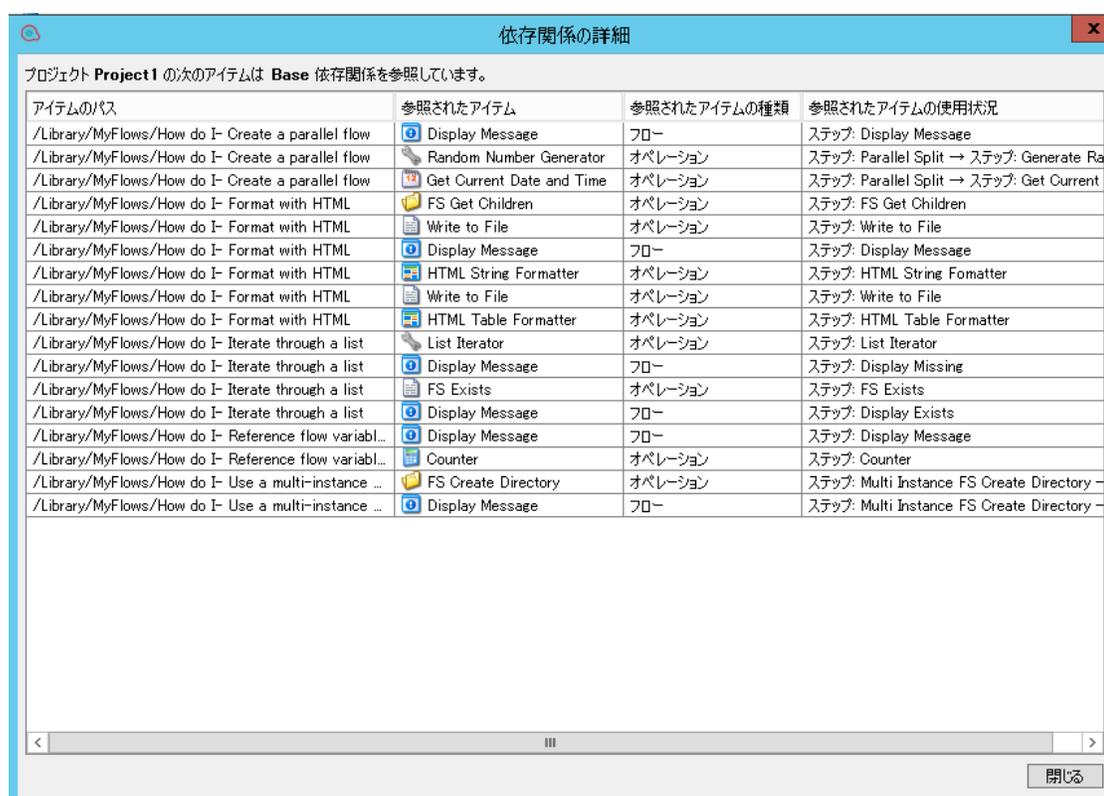
1. 依存関係エディターのツールバーで、削除する依存関係を選択します。
2.  をクリックします。

依存関係の詳細の表示

[問題] ペインで、警告メッセージの原因になっている依存関係を詳しく調べ、必要に応じて依存関係を更新できます。

1. [問題] ペインで、興味があるエラー/警告を選択します。
2. ツールバーで、 をクリックします。

[依存関係の詳細] ダイアログボックスが開き、選択した依存関係を参照しているプロジェクト内のアイテムのリストが表示されます。例:



依存関係の詳細

プロジェクト Project1 の次のアイテムは Base 依存関係を参照しています。

アイテムのパス	参照されたアイテム	参照されたアイテムの種類	参照されたアイテムの使用状況
/Library/MyFlows/How do I- Create a parallel flow	 Display Message	フロー	ステップ: Display Message
/Library/MyFlows/How do I- Create a parallel flow	 Random Number Generator	オペレーション	ステップ: Parallel Split → ステップ: Generate Ra
/Library/MyFlows/How do I- Create a parallel flow	 Get Current Date and Time	オペレーション	ステップ: Parallel Split → ステップ: Get Current
/Library/MyFlows/How do I- Format with HTML	 FS Get Children	オペレーション	ステップ: FS Get Children
/Library/MyFlows/How do I- Format with HTML	 Write to File	オペレーション	ステップ: Write to File
/Library/MyFlows/How do I- Format with HTML	 Display Message	フロー	ステップ: Display Message
/Library/MyFlows/How do I- Format with HTML	 HTML String Formatter	オペレーション	ステップ: HTML String Fomatter
/Library/MyFlows/How do I- Format with HTML	 Write to File	オペレーション	ステップ: Write to File
/Library/MyFlows/How do I- Format with HTML	 HTML Table Formatter	オペレーション	ステップ: HTML Table Formatter
/Library/MyFlows/How do I- Iterate through a list	 List Iterator	オペレーション	ステップ: List Iterator
/Library/MyFlows/How do I- Iterate through a list	 Display Message	フロー	ステップ: Display Missing
/Library/MyFlows/How do I- Iterate through a list	 FS Exists	オペレーション	ステップ: FS Exists
/Library/MyFlows/How do I- Iterate through a list	 Display Message	フロー	ステップ: Display Exists
/Library/MyFlows/How do I- Reference flow variabl...	 Display Message	フロー	ステップ: Display Message
/Library/MyFlows/How do I- Reference flow variabl...	 Counter	オペレーション	ステップ: Counter
/Library/MyFlows/How do I- Use a multi-instance ...	 FS Create Directory	オペレーション	ステップ: Multi Instance FS Create Directory -
/Library/MyFlows/How do I- Use a multi-instance ...	 Display Message	フロー	ステップ: Multi Instance FS Create Directory -

閉じる

4. アイテムをダブルクリックして Studio にロードします。

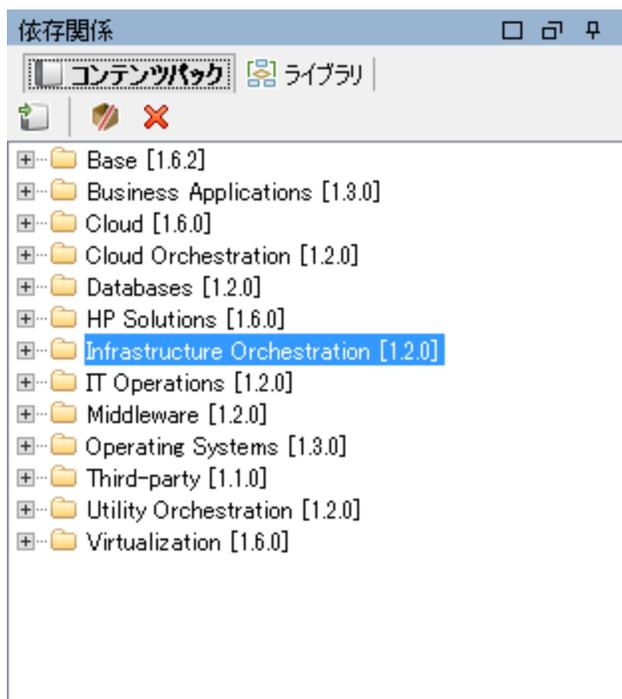
参考資料

[依存関係] ペイン

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

[依存関係] ペインには次の2つのタブがあります。

- **コンテンツパック** - 複数のコンテンツパックごとに複数のツリーが表示されます。このビューから、コンテンツパックのクローズ、削除、またはインポートを実行できます。
- **ライブラリ** - すべてのコンテンツを汎用の [Library] フォルダーと汎用の [Configuration] フォルダーの下にまとめたツリーが1つ表示されます。このビューから、コンテンツパックをインポートできます。



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

コンテンツパックのプロパティ

プロジェクトを右クリックして [プロパティ] を選択すると、プロパティシートが開きます。

プロジェクトのプロパティシートで、すべてのプロパティを更新できます。ただし、コンテンツパックの場合は、プロパティは読み取り専用です。

[依存関係] ペインから同じ情報を表示できます。この情報は、インポートしたコンテンツパック jar ファイルから取得されます。コンテンツパックを HP 00 プロジェクトから作成した場合は、[コンテンツパックの作成] ダイアログボックスに入力した内容から情報が取得されます。

名前:	Base
UUID:	75b8b3d6-d260-43af-acf9-8a142b4feadf
バージョン:	1.6.2
パブリッシャー:	Hewlett-Packard
説明:	Base Content Pack contains flows and operations to be used in every automation use case. Inside you can find technologies such as HTTP Client, Email, Remote Command, XML#JSON processing and many others. Dependencies: Base Content Pack does not depend on any other Content Pack.

GUI アイテム	説明
名前	コンテンツパックの名前は、プロジェクト名から取得されます。このフィールドは読み取り専用です。
UUID	コンテンツパックの一意の ID。
バージョン	コンテンツパックのバージョン。
パブリッシャー	コンテンツパックのパブリッシャーを入力します。この情報は、コンテンツパックの[プロパティ]シートに表示されます。
説明	コンテンツパックの説明を入力します。この情報は、コンテンツパックの[プロパティ]シートに表示されます。

依存関係エディター

コンテンツパック/プロジェクトのプロパティページには、依存関係エディターが表示されます。

既存関係エディターで、現在のプロジェクトのインストールされているすべてのコンテンツパック、およびそれらのバージョンと依存関係を表示できます。必要に応じてこれらの値を編集できます。

依存関係:



名前	バージョン
Base	1.4.0-SNAPSHOT
Middleware	1.2.0-SNAPSHOT
Infrastructure Orchestration	1.2.0-SNAPSHOT

GUI アイテム	説明
名前	コンテンツパックまたはプロジェクトの名前。

バージョン	<p>コンテンツパックのバージョン。次のいずれかを使用してこのバージョンを編集できます。</p> <ul style="list-style-type: none"> • 固定バージョン番号: 例: 1.2.0。これはコンテンツパックの正確なバージョンのみを使用できることを意味します。 • 間隔: 例: 1.2-。1.2 以上の任意のバージョンを使用できることを意味します。 <p>例: -1.21.2 未満の任意のバージョンのみを使用できることを意味します</p> <ul style="list-style-type: none"> • 間隔と範囲: 例: 1.2-3.0.1。1.2 ~ 3.0.1 (3.0.1 を除く) の任意のバージョンを使用できることを意味します
追加 	[依存関係の追加] ダイアログを開き、現在のプロジェクトの新しい依存関係を追加します。
削除 	選択した依存関係を現在のプロジェクトから削除します。
編集 	[依存関係の編集] ダイアログを開き、選択した依存関係の定義を変更します。
依存関係の更新 	現在のプロジェクト内の依存関係の競合を表示して解決します。

[問題] ペインのツールバー



GUI アイテム	説明
	該当する場合は、問題に関連するフローを開きます。
	現在のプロジェクト内の依存関係を更新します。
	現在のプロジェクトの依存関係を追加します。
	現在のプロジェクトから依存関係を削除します。
	現在のプロジェクト内の依存関係の詳細を表示します。

[問題] ペイン内の依存関係

依存関係を検証するときに問題が発生した場合は、[問題] ペインでそれらを表示できます。

発生する可能性がある 3 種類の依存関係の問題があります。[問題] ペインに、各問題に固有の警告メッセージが表示されます。

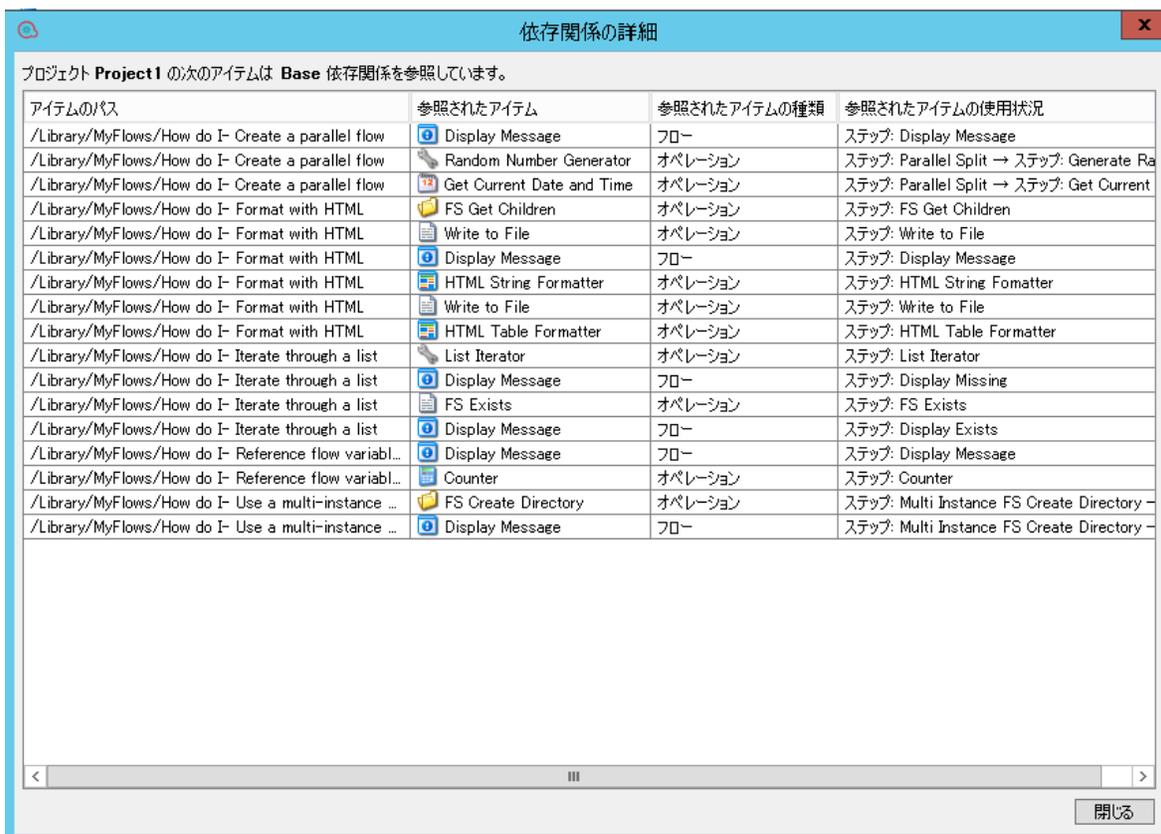
- **存在しない依存関係:**別のコンテンツパック/プロジェクトのアイテムへのリンクがプロジェクトに含まれていますが、そのアイテムが含まれるコンテンツパック/プロジェクトはこのプロジェクトの依存関係リストで宣言されていません。表示
- **未使用の可能性のある依存関係:**プロジェクトの宣言された依存関係に含まれている依存関係が、Studio の依存関係検出ルールで見つかりません。しかし、これらのルールは、システムプロパティの動的な参照を検出しないので、システムプロパティの動的な参照に関する宣言された依存関係が存在している場合やこの依存関係がまったく存在していない場合があります。
- **ワークスペースに含まれていない:**プロジェクトが、Studio ワークスペースに現在インポートされていないプロジェクトまたはコンテンツパックにリンクします。

次の画面は、この 3 つの種類の例を示しています。

種類	ソースの種類	名前	説明	場所
エラー	依存関係	work	このフローはコンテンツパック work に対する 'Base [1.0-SNAPSHOT]' 依存関係は...	/work
警告	依存関係	work	プロジェクトコンテンツパック work に対する 'Base [1.0-SNAPSHOT]' 依存関係は...	/work
エラー	フロー	Basic Notify (2)	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Notification Opt...	/oo10-base-cp-1.6.2/Library/Utility Operations/Deprecated/Basic Notify (2)
エラー	フロー	Ping with WMI	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Network - Proto...	/oo10-base-cp-1.6.2/Library/Operations/Network/Deprecated/Ping with WMI
エラー	フロー	Check Process By Command Line	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Remote Comma...	/oo10-base-cp-1.6.2/Library/Operations/Operating Systems/Solaris/Process ...
エラー	フロー	Log Analyze	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Linux Log Sever...	/oo10-base-cp-1.6.2/Library/Operations/Operating Systems/Linux/Logs/Sam...
エラー	ステップ	Get Inactive Users	このステップがリンクしているオペレーションに問題があります	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Notification Opt...	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このステップがリンクしているオペレーションに問題があります	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このフローは、重複した構成アイテムを使用しています。選択リスト名 'Notification Opt...	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...
エラー	ステップ	Notify	このステップがリンクしているオペレーションに問題があります	/oo10-base-cp-1.6.2/Library/Accelerator Packs/Active Directory/Deprecated/...

GUI アイテム	説明
種類	<p>問題の種類: 警告またはエラー。エラーがあるフローはデバッグできませんが、編集することはできます。</p> <p>注:</p> <ul style="list-style-type: none"> • すべての問題、警告、およびエラーを解決することをお勧めします。 • 複数の作成者がいる環境では、ワークスペースでの 1 人の作成者による変更を他の作成者も使用するので、依存関係の検証を実行したときに警告が生成されないことが重要です。そのため、正しいバージョンの必要なプロジェクト/コンテンツパックをインポートすることが重要です。
ソースの種類	問題が発生したセクション。これは、フローエラー、オペレーションエラー、ステップエラー、構成アイテムエラー、または依存関係の警告の場合があります。
名前	セクションの名前。
説明	依存関係の問題の説明。
場所	アイテムの場所。

[依存関係の詳細] ダイアログボックス



GUI アイテム	説明
アイテムのパス	依存関係を参照しているプロジェクト内のアイテムのパス。
参照されたアイテム	依存関係を参照しているプロジェクト内のアイテムの名前。
参照されたアイテムの種類	依存関係を参照しているプロジェクト内のアイテムのタイプ。
参照されたアイテムの使用状況	アイテムが使用されているフロー内のセクションの名前。

構成アイテムの管理

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

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

重要: プロジェクト内の構成アイテムを削除、作成、または名前変更する必要がある場合は、ファイルシステムでアイテムを削除、作成、または名前変更するのではなく、Studio 内から実行してください。

構成アイテムの使用

構成アイテムの検索

構成アイテムは、UUID、名前、説明の各フィールドで検索できます。種類が <すべてのフィールド> の検索、および種類が <Lucene クエリ> の検索 (名前、説明、UUID を検索クエリ内のフィールドとして使用) も、現在開かれているプロジェクトとコンテンツパックに属している構成アイテムを検索します。

[検索] ペインを使用して、開かれているプロジェクトとコンテンツパックのリスト全体をフルテキスト検索できます。



ランク	名前	種類	パス	説明
*****	Get Config	フロー	/HP Solutions [1.3.0]/Library/Integrations/Hewlett...	<pre>デバイスの構成情報を取得します入力hostname ...
*****	Get Config	オペレーション	/HP Solutions [1.3.0]/Library/Integrations/Hewlett...	<pre>エンクロープターの設定を作成し直すために必要な ...
*****	Config Change with Rollback	フロー	/HP Solutions [1.3.0]/Library/Integrations/Hewlett...	<pre>コマンドによって Cisco ルーターが到達不可能にな ...
**	Configuration Item	ドメインターム	/Base [1.3.0]/Configuration/Domain Terms	IP アドレスの割り振られているマシン、ルーター、またはあら ...
**	Run SCVMM Commands	オペレーション	/Virtualization [1.3.0]/Library/Integrations/Microsoft...	Windows PowerShell を使用して、サービスセンター復元 ...
*	Restart Exchange	フロー	/Business Applications [1.1.0]/Library/Operations/E...	Exchange を安全な方法でリセットします。最初に動作し ...
*	Get Mailbox Database	フロー	/Business Applications [1.1.0]/Library/Operations/E...	1 つ以上のメールボックスデータベースの名前を取得しま ...
*	Get Storage Group	フロー	/Business Applications [1.1.0]/Library/Operations/E...	Exchange サーバーから 1 つ以上のストレージグループの ...
*	Exchange Server Health	フロー	/Business Applications [1.1.0]/Library/Accelerator ...	このフローは、ドメインコントローラーに登録されている Exch ...
*	User Mailbox Health	フロー	/Business Applications [1.1.0]/Library/Accelerator ...	メールボックスの Exchange Server を検索し、サーバーの ...
*	Soft Reset Exchange	フロー	/Business Applications [1.1.0]/Library/Accelerator ...	Exchange を安全な方法でリセットします。最初に診断を ...
*	Run Exchange 2010 Cmdlets	オペレーション	/Business Applications [1.1.0]/Library/Operations/E...	Windows PowerShell を使用して Exchange 2010 コマ ...
*	Test IMAP Connectivity	フロー	/Business Applications [1.1.0]/Library/Operations/E...	Windows PowerShell を使用して Exchange 2010 の ...

構成アイテムの検索方法の詳細については、「[「フローまたはオペレーションの検索」 \(357 ページ\)](#)の「オペレーションまたは構成アイテムの検索」を参照してください。

構成アイテムフォルダーの使用

既存の構成アイテム構造の下のすべての構成アイテム内にフォルダーを作成できます。構成アイテムフォルダーで次の操作を実行できます。

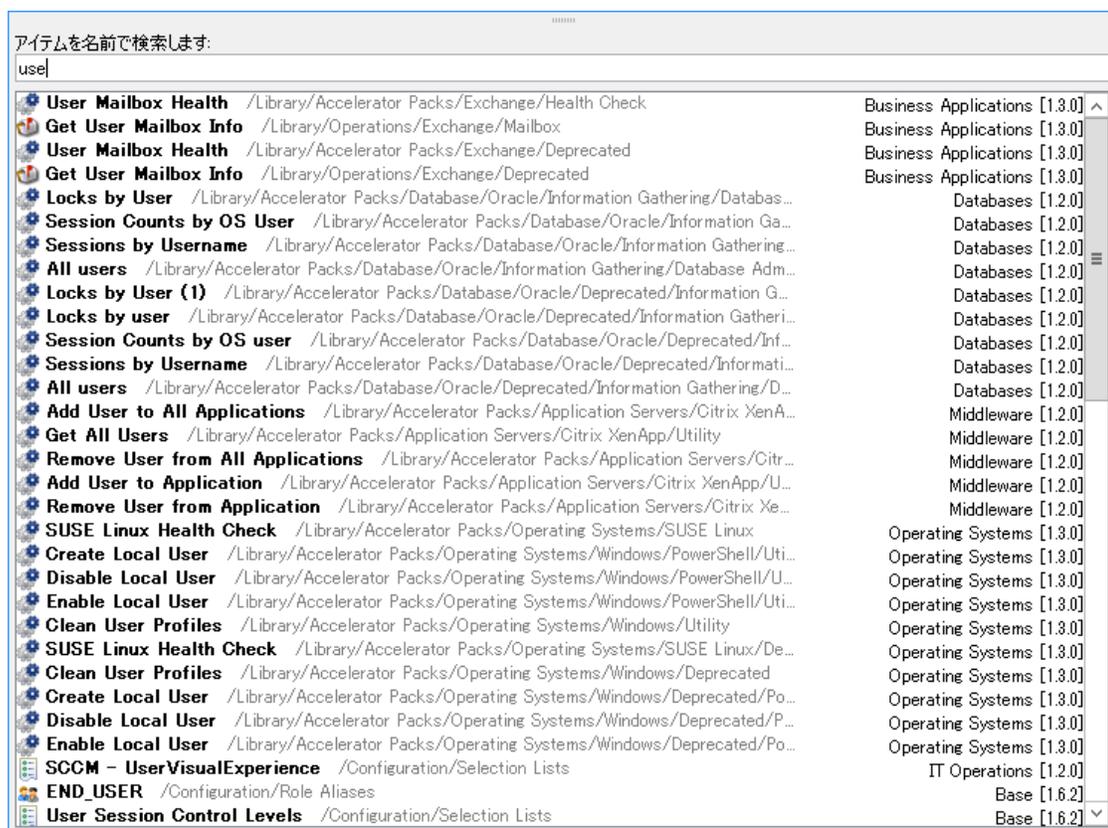
- 各構成アイテムセクション内にフォルダーを作成する
- 1つのプロジェクトから構成アイテムフォルダーをコピーして別のプロジェクトに貼り付ける
- 1つのプロジェクトから別のプロジェクトに構成アイテムフォルダーを移動する
- フォルダーと最上位レベルのセクションの間で構成アイテムを移動する
- 構成アイテムフォルダーを削除する
- 構成アイテムフォルダーの名前を変更する
- 各構成アイテムの説明を設定する

実行する操作

[フローに移動...] オプションを使用した構成アイテムの検索

[アイテムに移動...] オプションを使用して、ワークスペース内の構成アイテムを簡単に見つけて開くことができます。

1. [ツール] > [アイテムに移動...] を選択します。
[アイテムを名前で検索します:] ウィンドウが表示されます。
2. 構成アイテム名の一部を入力します。
ウィンドウに検索結果が直ちに表示されます。たとえば、"use" という文字列が含まれる名前のアイテムを検索する場合、次のような結果が表示されることがあります。



3. 上向き矢印と下向き矢印キーを使用してリスト内を移動し、Enter キーを押して選択したアイテムを開きます。または、ダブルクリックしてアイテムを開くことができます。
4. 現在のフローエディターでリストからオペレーションまたはフローをドラッグアンドドロップしてステップを作成できます。

構成アイテムフォルダーを作成する

1. [プロジェクト] ペインで、**Configuration** フォルダーを展開します。
2. 構成アイテムフォルダーを右クリックし、[新規]>[フォルダー] をクリックします。
3. 新しいフォルダーの名前を入力して、[OK] をクリックします。

このフォルダー内で構成アイテムを追加、コピー、または移動することができます。

構成アイテム/フォルダーの名前を変更する

1. [プロジェクト] ペインで、**Configuration** フォルダーを展開します。
2. 構成アイテムフォルダーを右クリックし、[名前の変更] をクリックします。
3. 新しい名前を入力し、Enter を押します。

1つのプロジェクトから構成アイテムをコピーして別のプロジェクトに貼り付ける

1. [プロジェクト] ペインで、**Configuration** フォルダーを展開します。

2. 構成アイテムフォルダーを右クリックし、[編集]>[コピー]を選択します。
3. 目的のフォルダーで構成アイテムの場所を選択し、[編集]>[貼り付け]を選択します。

構成アイテムフォルダーおよびアイテムを移動する

1. [プロジェクト]ペインで、**Configuration** フォルダーを展開します。
2. 構成アイテムフォルダーを右クリックし、[編集]>[切り取り]を選択します。
3. 2つ目のプロジェクトでフォルダーの場所を選択し、[編集]>[貼り付け]を選択します。
フォルダー全体およびその構成アイテムが新しい場所に移動されます。

構成アイテムフォルダーを削除する

1. [プロジェクト]ペインで、**Configuration** フォルダーを展開します。
2. 構成アイテムフォルダーを右クリックし、[削除]を選択します。

注: フォルダーとその構成アイテムを削除すると、他のアイテムとの間の依存関係が壊れます。

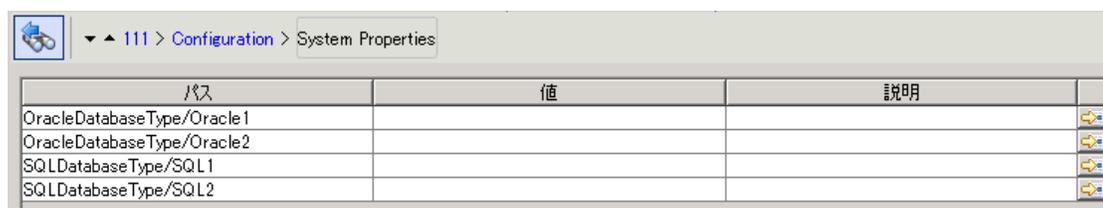
構成アイテムフォルダーの説明を設定する

1. 構成アイテムフォルダーを右クリックし、[プロパティ]を選択します。
2. [説明]ボックスに、構成アイテムフォルダーの説明を入力します。
3. [保存]をクリックします。

すべての構成アイテムとそれらのフォルダー構造を表示する

1. [プロジェクト]ペインで、構成アイテムフォルダーを右クリックします。
2. [プロパティ]を選択します。

すべての構成アイテムとそれらのフォルダー構造が表示されます。たとえば、システムプロパティの場合、次のリストが表示されます。



パス	値	説明
OracleDatabaseType/Oracle1		
OracleDatabaseType/Oracle2		
SQLDatabaseType/SQL1		
SQLDatabaseType/SQL2		

注: このページからフォルダー構造を読み取り専用モードで表示できます。行内の名前を編集することはできません。ただし、行をダブルクリックするか、各行の右にある  アイコンを選択して、個別の構成アイテムを編集できます。

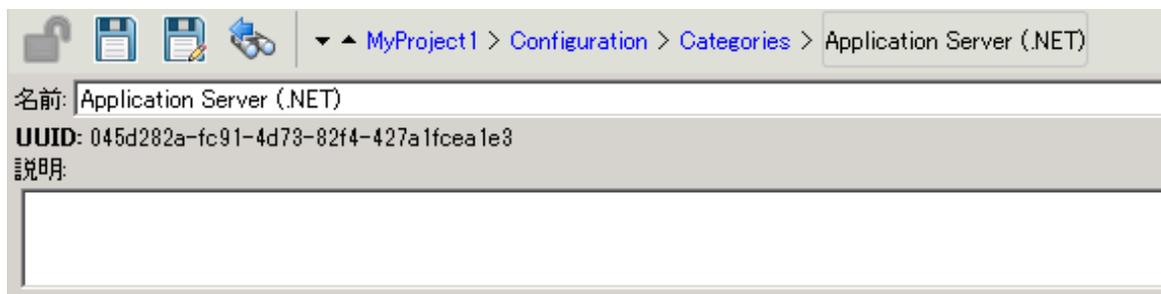
カテゴリの設定

カテゴリとは、フローに割り当てることができる分類のことです。Studio には多数のカテゴリがインストールされていますが、独自のカテゴリの作成もできます。

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

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

カテゴリは **Configuration\Categories** フォルダに保存されます。



実行する操作

カテゴリの作成

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

カテゴリの変更

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

カテゴリの名前の変更

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

カテゴリのコピー

1. [プロジェクト] ペインで、**Configuration** フォルダと **Categories** フォルダを展開します。
2. コピーするカテゴリを右クリックします。

3. [編集]>[コピー]を選択します。
4. コピー先のプロジェクトの**Category** フォルダーに移動します。
5. 右クリックして、[編集]>[貼り付け]を選択します。

カテゴリの削除

カテゴリを削除する前に、[これを使用する項目は?]機能を使用して、ほかのアイテムが依存していないかを確認することをお勧めします。詳細については、「[フローとオペレーションの使用方法的詳細](#)」(366ページ)を参照してください。

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

参考資料

カテゴリエディター



名前: Application Server (.NET)
UUID: 045d282a-fc91-4d73-82f4-427a1fcea1e3
説明:

GUI アイテム	説明
名前	カテゴリの名前です。
説明	(オプション) カテゴリの説明です。

ドメインタームの設定

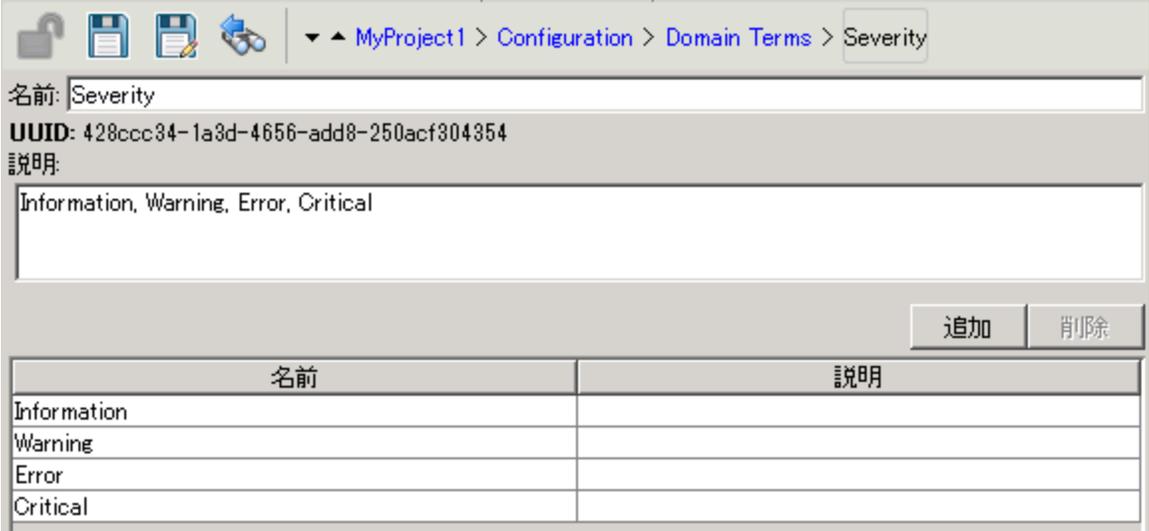
ドメインタームは、フローや入力に割り当てることができる属性です。たとえば、システム内のさまざまな種類のサーバーに対してドメインタームを作成すると、特定の種類のサーバーのみに対してステップを実行させることができます。

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

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

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

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



名前	説明
Information	
Warning	
Error	
Critical	

実行する操作

ドメインタームの作成

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

ドメインタームの値の削除

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

ドメインタームの値の変更

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

ドメインタームのコピー

1. [プロジェクト] ペインで、**Configuration** フォルダーと **Domain Term** フォルダーを展開します。
2. コピーするドメインタームを右クリックします。
3. [編集]>[コピー] を選択します。
4. コピー先のプロジェクトの **Domain Terms** フォルダーに移動します。
5. 右クリックして、[編集]>[貼り付け] を選択します。

ドメインタームの名前の変更

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

ドメインタームの削除

ドメインタームを削除する前に、**[これを使用する項目は?]** 機能を使用して、ほかのアイテムが依存していないかをチェックすることをお勧めします。詳細については、[「フローとオペレーションの使用方法の詳細」\(366ページ\)](#)を参照してください。

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

参考資料

ドメインタームエディター

名前	説明
Information	
Warning	
Error	
Critical	

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

グループエイリアスの設定

RAS グループ

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

RESTful API を使用して、RAS グループをサーバーで定義できます。詳細については、『HP 00 Application Program Interface (API) Guide』を参照してください。

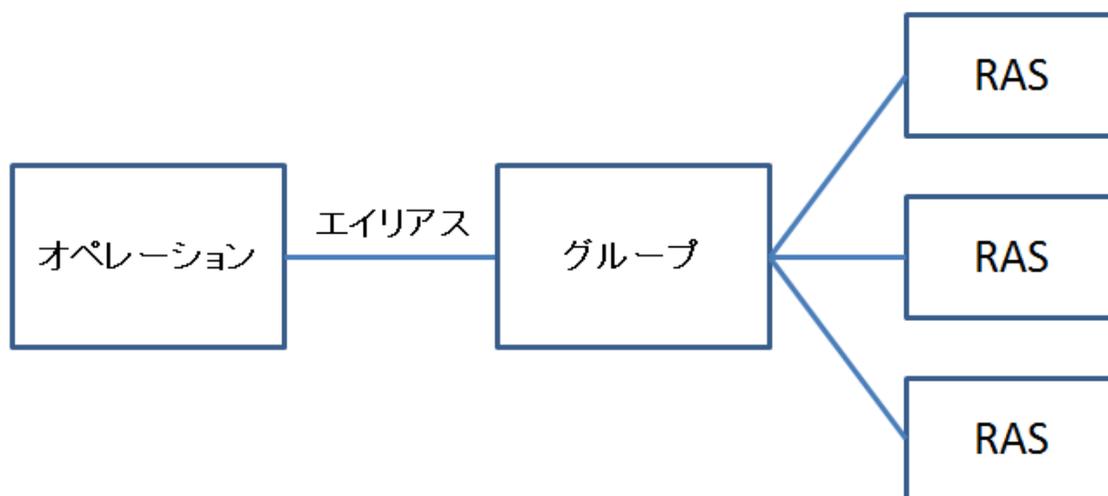
グループエイリアス

グループエイリアスを使用すると RAS に割り当てるオペレーションを作成時と実行時環境とで区別することができます。

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

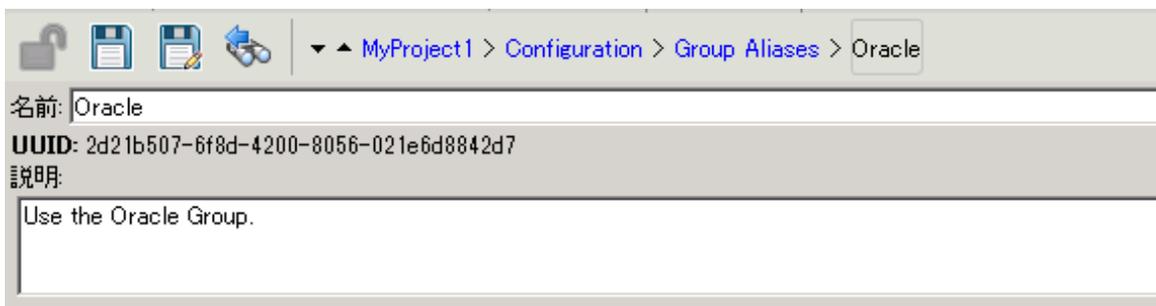
フォールバックとして、グループエイリアスがグループ名と同一の場合、そのグループに自動的にマッピングされます。

オプションとして、トリガーする際にグループエイリアスをオーバーライドし、オペレーションを別の RAS グループにマッピングすることができます。



たとえば、Oracle クライアントで実行する 3 つの RAS のグループがあるとします。Oracle でクエリを実行するオペレーションを作成します。このグループに対するエイリアスを使用することで、HP 00 に、このオペレーションがこのグループのいずれかの RAS で実行する必要があることを知らせます。使用する RAS の選択は実行時に決定されるため、オペレーションで設定する必要はありません。

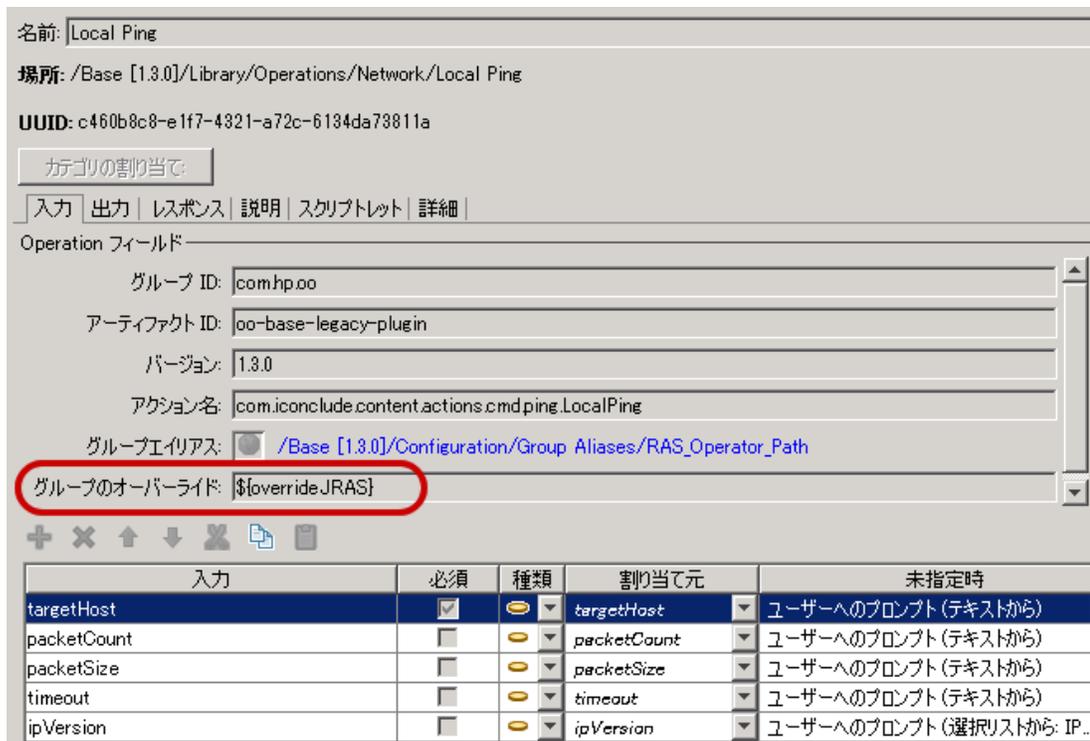
グループエイリアスは **Configuration\Group Aliases** フォルダーに保存されます。



RAS グループのオーバーライド

HP 00 では、次の 2 つの方法で RAS グループの詳細をオーバーライドすることができます。

- 静的 - 次の説明を参照してください。 [「グループエイリアス」 \(151ページ\)](#)
- 動的 - HP 00 が実行時に入力の値を割り当てます。これは、次のいずれかの方法で実行できます。
 - フロー作成者は、Studio の入力インスペクターの [オペレーション] フィールドの **[グループのオーバーライド]** フィールドを使用して RAS を設定できます。(オプション) 現在のグループを別のグループにオーバーライドする必要がある場合、**[グループのオーバーライド]** フィールドに別のグループを入力します。これには静的な値または動的な値を設定できます。たとえば、`${overrideJRAS}` を設定します。この `overrideJRAS` は変数の名前です。実行中に、 [「RAS グループのオーバーライド」 \(152ページ\)](#) で説明されているいずれかの方法を使用して、変数に値を自動的に割り当てることができます。



- Central のスケジューラーで、実行時に自動的に割り当てられる入力フィールドを追加します。

『Central ユーザーガイド』の「フロー実行のスケジューリング」を参照してください。

- Studio でフローに入力フィールドを追加します。「[入力の作成](#)」(205ページ)を参照してください。

RAS グループのオーバーライドの使用例

- **例 1:** 動的な実行環境を変数として定義します。この環境には、テスト環境または実行時の実稼動環境を使用できます。この変数を Studio でフロー入力、ステップ入力、または設定コンテキストとして使用したり、スケジューラーからの入力として使用したりすることができます。
- **例 2:** RAS を使用する SAP クライアントを定義します。このクライアントを Central フローから SAP グループとして呼び出すことができます。

実行する操作

グループエイリアスの作成

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

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

グループエイリアスの名前の変更

1. [プロジェクト] ペインで **Configuration** フォルダと **Group Aliases** フォルダを展開し、ドメインタームをダブルクリックしてエディターを開きます。
2. グループエイリアスを右クリックし、[名前の変更] を選択します。
3. グループエイリアスの新しい名前を入力し、Enter を押します。

グループエイリアスのコピー

1. [プロジェクト] ペインで、**Configuration** フォルダと **Group Aliases** フォルダを展開します。
2. コピーするグループエイリアスを右クリックします。
3. [編集] > [コピー] を選択します。
4. コピー先のプロジェクトの **Group Aliases** フォルダに移動します。
5. 右クリックして、[編集] > [貼り付け] を選択します。

グループエイリアスの削除

1. [プロジェクト] ペインで、**Configuration** フォルダと **Group Aliases** フォルダを展開します。
2. 削除するグループエイリアスを右クリックし、**[削除]** を選択します。

グループのグループエイリアスへのマッピング

Central RESTful API を使用して、実行環境でグループエイリアスをグループにマッピングします。

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

オペレーションでのグループエイリアスの使用

1. 「[オペレーションの作成](#)」(349ページ)の説明に従って、アクションプラグインから新しいオペレーションを作成します。
2. [入力] タブの **[Operation フィールド]** セクションで、**[グループエイリアス]**  ボタンに注意します。

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

3. [構成] > **[グループエイリアス]** を選択し、必要なグループエイリアスを選択して、適切なオペレーションにドラッグし、**[グループエイリアス]** アイコン (地球儀型のアイコン) 上にドロップします。
4. (オプション) 現在のグループを別のグループにオーバーライドする必要がある場合、**[グループのオーバーライド]** ボックスに別のグループを入力します。これには静的な値または動的な値を設定できます。たとえば、`${overrideJRAS}` を設定します。この `overrideJRAS` は変数の名前です。実行中に、[「RAS グループのオーバーライド」](#) (152ページ) で説明されているいずれかの方法を使用して、変数に値を自動的に割り当てることができます。
5. オペレーションを保存します。

オペレーションからのグループエイリアスの割り当ての削除

グループエイリアスをオペレーションに割り当てたら、この割り当てを削除できます。

1. グループエイリアスの割り当て先のオペレーションを開きます。
2. [入力] タブの **[Operation フィールド]** セクションで、**[グループエイリアス]**  ボタンの横にあるリンクを右クリックします。

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

3. 右クリックメニューから [クリア] を選択します。

参考資料

グループエイリアスのエディター

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

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

役割エイリアスの設定

役割とアクセス許可は、Central でユーザーに割り当てられます。ただし、Studio で役割エイリアスを作成することによって、使用する役割の計画を立てることができます。たとえば、役割エイリアスをゲート制御式トランジションに付加することもできます。

コンテンツパックが Central にデプロイされると、ADMINISTRATOR、EVERYBODY、PROMOTER、SYSTEM_ADMIN、END_USER などの役割エイリアスは Central の対応する役割にマッピングされます。

注: Base コンテンツパックの役割エイリアスの一部 (AUDITOR、LEVEL_ONE、LEVEL_TWO、LEVEL_THREE) は、Central に対応する役割がありません。これらの役割エイリアスは非推奨と見なされます。

役割エイリアスは **Configuration\Role Aliases** フォルダーに保存されます。

実行する操作

役割エイリアスの作成

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

役割エイリアスの名前の変更

1. [プロジェクト] ペインで、**Configuration** フォルダーと **Role Aliases** フォルダーを展開します。
2. 名前を変更する役割エイリアスを右クリックし、[名前の変更] を選択します。
3. 役割エイリアスの新しい名前を入力し、Enter を押します。

役割エイリアスのコピー

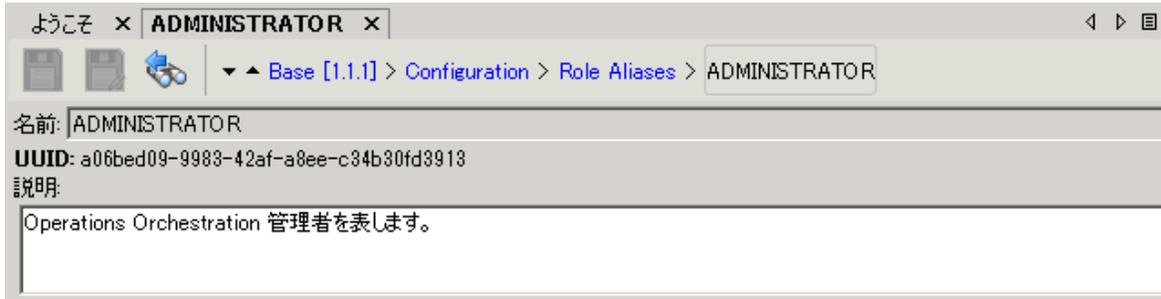
1. [プロジェクト] ペインで、**Configuration** フォルダーと **Role Aliases** フォルダーを展開します。
2. コピーする役割エイリアスを右クリックします。
3. [編集] > [コピー] を選択します。
4. コピー先のプロジェクトの **Role Aliases** フォルダーに移動します。
5. 右クリックして、[編集] > [貼り付け] を選択します。

役割エイリアスの削除

1. [プロジェクト] ペインで、**Configuration** フォルダーと **Role Aliases** フォルダーを展開します。
2. 削除する役割エイリアスを右クリックし、[削除] を選択します。

参考資料

役割エイリアスのエディター



GUI アイテム	説明
名前	役割エイリアスの名前です。
説明	(オプション) 役割エイリアスの説明です。

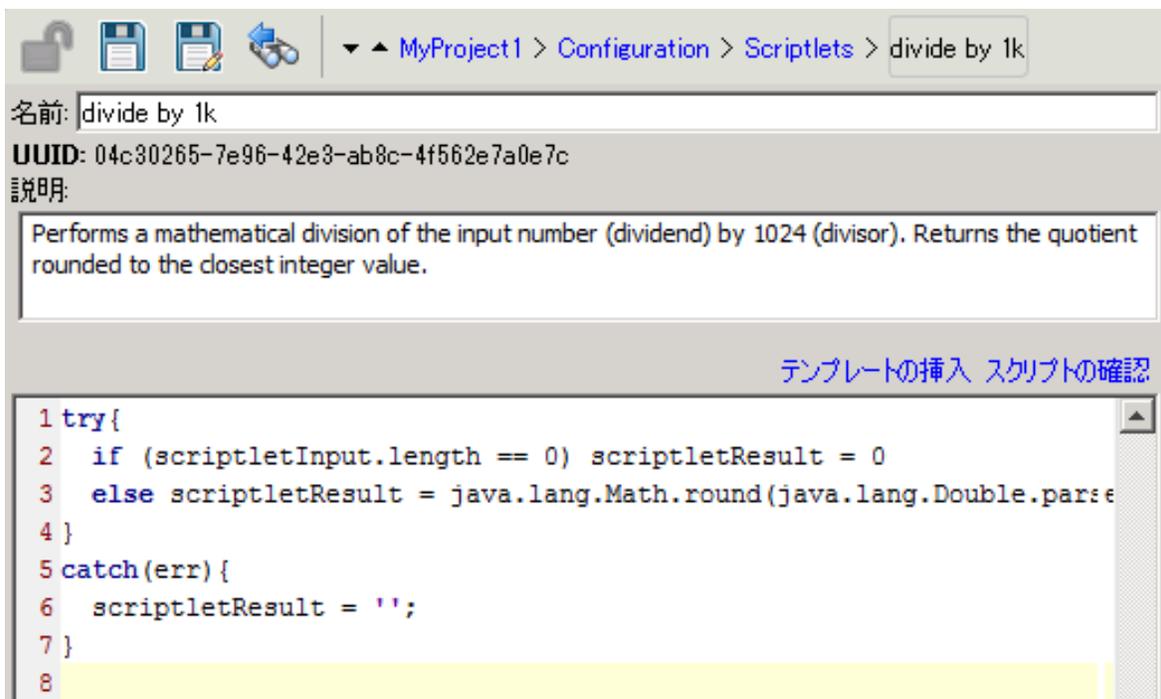
スクリプトレットの設定

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

スクリプトレットを使用すると、次のことを実行できます。

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

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



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

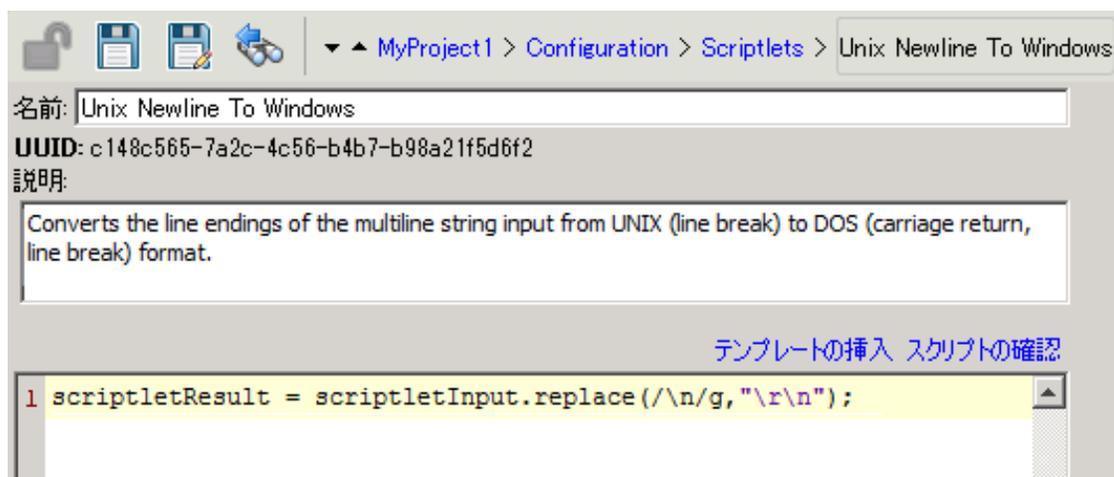
システムスクリプトレットは **Configuration\Scriptlets** フォルダに保存されます。

スクリプトレットの使用の詳細については、[「フロー内でのスクリプトレットの使用」\(297ページ\)](#)を参照してください。

実行する操作

システムスクリプトレットの作成

1. [プロジェクト] ペインで、**Configuration** フォルダを展開します。
2. **Scriptlets** フォルダで、右クリックして **[新規] > [スクリプトレット]** を選択します。
3. スクリプトレットの名前を入力して **[OK]** をクリックします。
スクリプトレットのエディターが開きます。



4. [説明] ボックスに、スクリプトレットの目的を入力します。
5. スクリプトレットを JavaScript で入力します。

注: フォルダーに格納されている変数を参照するには、`${variablename}` 形式を使用し、システムプロパティを参照するには、`${path/system_property}` 形式を使用します。

6. (オプション)[**テンプレートの挿入**]をクリックして、テンプレートのガイドラインに従いスクリプトレットを記述します。
7. [**スクリプトの確認**]をクリックしてエラーがないか確認します。 [「出力および結果のフィルタリング」\(255ページ\)](#)
8. [**保存**]をクリックします。

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

既存のスクリプトレットをシステムスクリプトレットとして保存

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

オペレーション、フロー、ステップのシステムスクリプトレットのフィルタリング

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

3. スクリプトレットを **Scriptlets** フォルダーから **[プロパティ]** シートまたはステップのインスペクターの **[スクリプトレット]** タブの **[スクリプトレット]** アイコンにドラッグします。
[スクリプトレット] タブに、共有のスクリプトレットへの参照があることが表示されます。



システムスクリプトレットの編集

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

スクリプトレットのコピー

1. **[プロジェクト]** ペインで、**Configuration** フォルダーと **Scriptlets** フォルダーを展開します。
2. コピーするスクリプトレットを右クリックします。
3. **[編集]** > **[コピー]** を選択します。
4. コピー先のプロジェクトの **Scriptlets** フォルダーに移動します。
5. 右クリックして、**[編集]** > **[貼り付け]** を選択します。

スクリプトレットの名前の変更

1. **[プロジェクト]** ペインで、**Configuration** フォルダーと **Scriptlets** フォルダーを展開します。
2. 名前を変更するスクリプトレットを右クリックし、**[名前の変更]** を選択します。
3. スクリプトレットの新しい名前を入力し、Enter を押します。

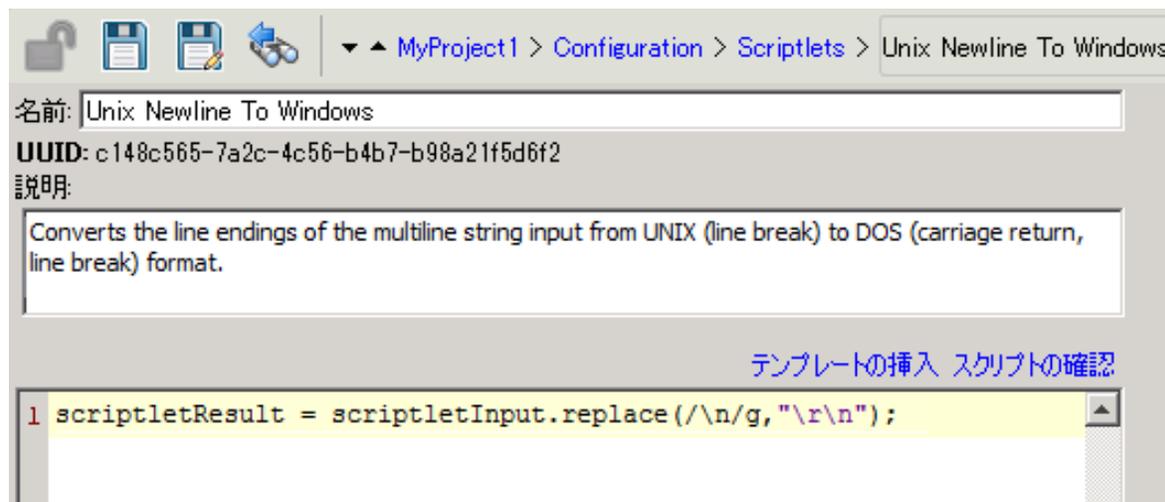
システムスクリプトレットの削除

システムスクリプトレットを削除する前に、**[これを使用する項目は?]** 機能を使用して、ほかのアイテムが依存していないかをチェックすることをお勧めします。詳細については、「[フローとオペレーションの使用方法的詳細](#)」(366ページ)を参照してください。

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

参考資料

スクリプトレットエディター



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

ステータスバー

- カーソルの位置に対応する行列が表示されます。
- 挿入モードで文字を入力している場合は、**Insert** キーの状態が表示されます。上書きモードでは、文字は上書きされます。2つのモードの切り替えには、Insert キーを使用します。

キーワード入力候補

- キーワードの入力中に **Ctrl** と **スペースキー** を押すと、オプションのドロップダウンリストが表示されます。上向き矢印と下向き矢印キーを使用してリスト内を移動して、使用する単語を選択します。アイテムを選択すると、リストは消え、選択した単語の後にカーソルが置かれます。

```
1 var sysvalue_password = scriptletContext.get("var_sa_studio_central_default");  
2 scriptletContext.put("field1", sysvalue_password);  
3 v  
4 void  
5 var  
6 volatile
```

行移動

- ステータスバーの行列アイテムをダブルクリックするか、**Ctrl + G** キーを押します。スクリプトレットウィンドウの左側に番号が付いている範囲内の行にだけ移動できます。

```
1  
2 var sysvalue_password = scriptletContext.get("var_sa_studio_central_default");  
3 scriptletContext.put("field1", sysvalue_password);  
4  
5  
6
```

Go to line number:

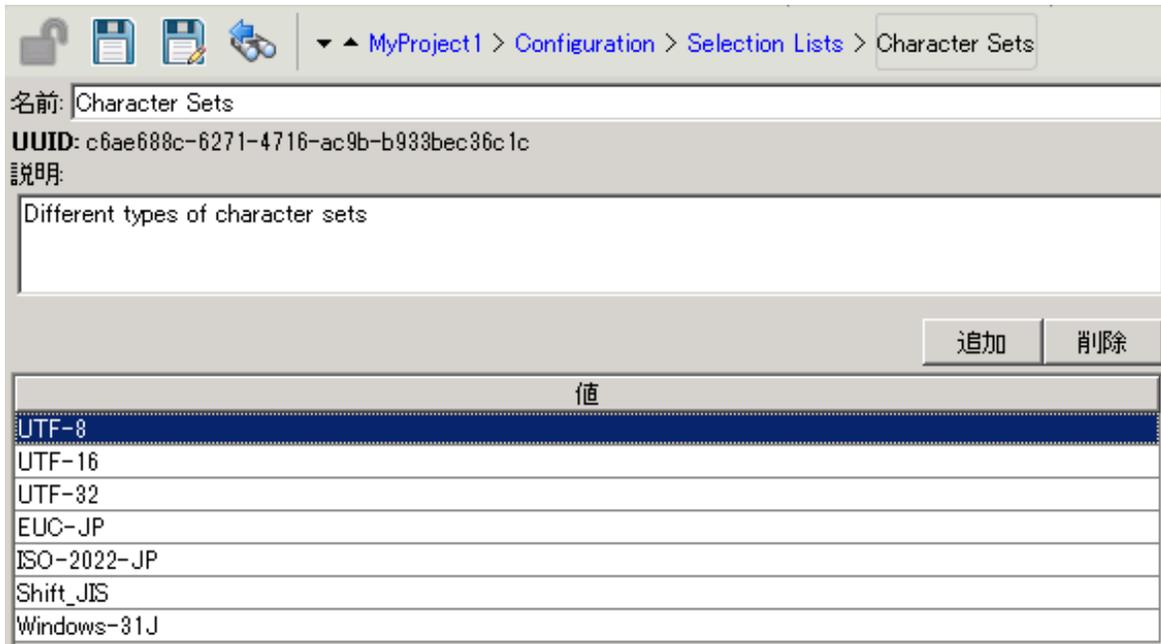
(Between 1 and 6. Press ENTER to continue)

選択リストの設定

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

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

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



実行する操作

選択リストの作成

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

選択リストの値の削除

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

選択リストの値の変更

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

選択リストの名前の変更

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

選択リストのコピー

1. [プロジェクト] ペインで、**Configuration** フォルダと **Selection Lists** フォルダを展開します。
2. コピーする選択リストを右クリックします。
3. [編集] > [コピー] を選択します。
4. コピー先のプロジェクトの **Selection Lists** フォルダに移動します。
5. 右クリックして、[編集] > [貼り付け] を選択します。

選択リストの削除

選択リストを削除する前に、**[これを使用する項目は?]** 機能を使用して、ほかのアイテムがこの選択リストに依存していないかをチェックすることをお勧めします。詳細については、[「フローとオペレーションの使用方法的詳細」\(366ページ\)](#)を参照してください。

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

参考資料

選択リストのエディター

名前: Character Sets
UUID: c6ae688c-6271-4716-ac9b-b933bec36c1c
説明: Different types of character sets

追加 削除

値
UTF-8
UTF-16
UTF-32
EUC-JP
ISO-2022-JP
Shift_JIS
Windows-31J

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

システムアカウントの設定

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

フロー作成者は、フローの作成時にシステムアカウントを使用できます。たとえば、入力ソースを、システムアカウントからの資格情報に設定することができます。[「入力ソースの指定」 \(214ページ\)](#) を参照してください。

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

システムアカウント名はユーザーに一度も表示されることなく、リモートマシンへのアクセスのためユーザーアカウント資格情報がフローに送られます。したがって、資格情報は復号化から保護され、システムアカウント名はユーザーに対して非表示となります。

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

注: <>\\"/;%の文字はシステムアカウント名には使用できません。

The screenshot shows a configuration window for a system account named "John Citizen". The breadcrumb path is "MyProject1 > Configuration > System Accounts > John Citizen". The "名前:" field contains "John Citizen". The "UUID:" field contains "3843ccd4-3206-4f49-8550-c5925596b982". The "説明:" field is empty. Below this is a section for "資格情報" (Credentials). The "ユーザー名:" field contains the template "<ドメイン>\<ユーザー名>". The "パスワード:" field contains a series of asterisks. To the right of the password field is a button labeled "パスワードの割り当て".

実行する操作

システムアカウントの作成

1. [プロジェクト] ペインで、**Configuration** フォルダーを展開します。
2. **System Accounts** フォルダーで、右クリックして [新規] > [システムアカウント] を選択します。
3. 表示されたダイアログボックスで、新しいシステムアカウントの名前を入力して [OK] をクリックします。
4. (オプション) [説明] ボックスに、システムアカウントの説明を入力します。
5. [ユーザー名] ボックスに、次の構文を使用して、このシステムアカウントを表すアカウントのユーザー名を入力します。
<ドメイン>\<ユーザー名>
6. [パスワードの割り当て] ボタンをクリックします。
7. [パスワード] ボックスにパスワードを入力し、[確認 パスワード] ボックスに再度入力します。
8. [保存] をクリックします。

システムアカウントのコピー

1. [プロジェクト] ペインで、**Configuration** フォルダと **System Accounts** フォルダを展開します。
2. コピーするシステムアカウントを右クリックします。
3. [編集] > [コピー] を選択します。
4. コピー先のプロジェクトの **System Accounts** フォルダに移動します。
5. 右クリックして、[編集] > [貼り付け] を選択します。

注: ドラッグアンドドロップを使用して、1つのプロジェクトから別のプロジェクトにシステムアカウントまたはそのシステムアカウントが含まれるフォルダを移動/コピーすることができます。

システムアカウントの編集

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

システムアカウントの削除

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

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

次の場所で DELETE を実行します。/oo/rest/system-accounts/<sa_name>.

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

参考資料

システムアカウントのエディター

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

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

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

システム評価子の設定

システム評価子は、フロー作成者がシステムアカウントを除く任意のデータソースの入力の検証に使用できる文字列形式です。

例:

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

「[入力データの評価](#)」(227ページ)を参照してください。

システム評価子では次のいずれかを使用できます。

- =、!=、Begins with、Contains、Match All Words、Match At Least One Word などの簡単な演算子。
- 正規表現 (詳細については下記を参照) 「[フローでの正規表現の使用](#)」(302ページ)
- スクリプトレット (詳細については、下記を参照) 「[フロー内でのスクリプトレットの使用](#)」(297ページ)

システム評価子は **Configuration\System Evaluators** フォルダーに保存されます。

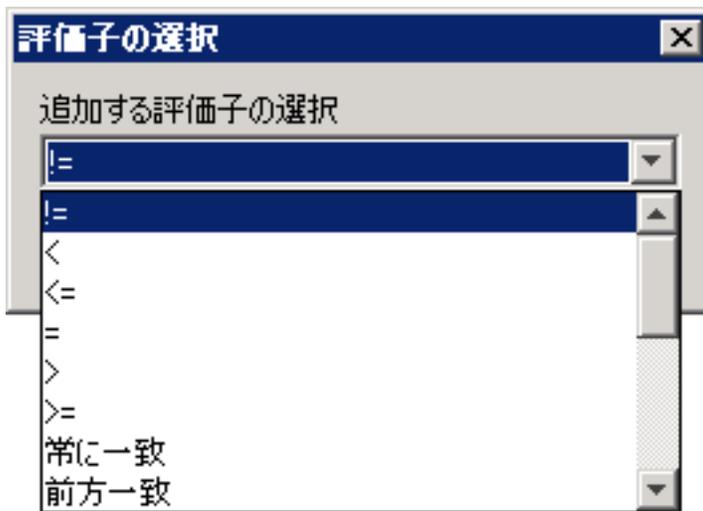
The screenshot shows a configuration window for a system evaluator. It has the following fields and content:

- 名前:** MyEvaluator
- UUID:** e2eff70a-26d9-46e4-8f70-02d1c03a4fa2
- 説明:** 入力値が1より小さいかどうかチェックする。
- 評価子の種類:** 入力文字または数値と比較
- 比較対象:** 1
- フィルター:** フィルター入力のテスト
パターンが一致します (0 回)
- 操作ボタン:** コピー、貼り付け、削除、クリア、クイックコマンド

実行する操作

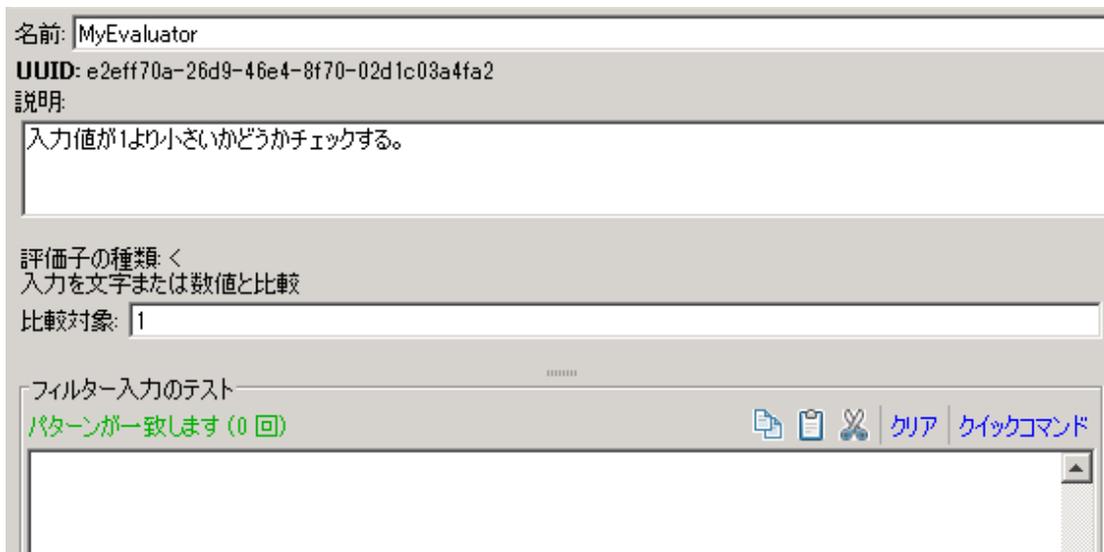
システム評価子の作成

1. [プロジェクト] ペインで、**Configuration** フォルダーを展開します。
2. **System Evaluators** フォルダーで、右クリックして [新規] > [システム評価子] を選択します。
3. [評価子の選択] ダイアログボックスで、評価子の種類を選択し、[OK] をクリックします。



4. 評価子の名前を入力して **[OK]** をクリックします。

評価子エディターが開きます。評価子エディターの外観は、選択した評価子の種類によって異なります。



5. **[説明]** ボックスに、評価子の目的を入力します。
6. 評価子の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。
7. フィルターをテストします。
 - a. **[クリア]** をクリックして **[フィルター入力のテスト]** ボックスの内容をクリアします。
 - b. **[クイックコマンド]** をクリックします。
 - c. 目的のデータを生成するコマンドを入力します。
 - d. **[OK]** をクリックします。コマンドの出力が **[フィルター入力のテスト]** ボックスに表示されます。

フィルターのテストの詳細については、「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。

8. **[保存]** をクリックします。

評価子が **System Evaluators** フォルダーに保存され、入力エディターの **[検査形式]** リストで使用できるようになります。

システム評価子の編集

1. **[プロジェクト]** ペインで、**Configuration** フォルダーと **System Evaluators** フォルダーを展開します。
2. 編集するシステム評価子をダブルクリックします。
3. 評価子の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変更します。
4. **[保存]** をクリックします。

システム評価子のコピー

1. **[プロジェクト]** ペインで、**Configuration** フォルダーと **System Evaluators** フォルダーを展開します。
2. コピーするシステム評価子を右クリックします。
3. **[編集]** > **[コピー]** を選択します。
4. コピー先のプロジェクトの **System Evaluators** フォルダーに移動します。
5. 右クリックして、**[編集]** > **[貼り付け]** を選択します。

システム評価子の削除

システム評価子を削除する前に、**[これを使用する項目は?]** 機能を使用して、ほかのアイテムが依存していないかをチェックすることをお勧めします。詳細については、「[フローとオペレーションの使用方法的詳細](#)」(366ページ)を参照してください。

1. **[プロジェクト]** ペインで、**Configuration** フォルダーと **System Evaluators** フォルダーを展開します。
2. システム評価子を右クリックし、**[削除]** を選択します。
3. 確認ウィンドウで **[はい]** をクリックします。

参考資料

評価子エディター - 標準

評価子エディターの外観は、選択した評価子の種類によって異なります。[評価子の選択] ダイアログボックスで、=、!=、Begins with、Contains、Match All Words、Match At Least One Wordなどの単純な演算子を選択した場合、評価子エディターは次のように表示されます。

名前: MyEvaluator
UUID: e2eff70a-26d9-46e4-8f70-02d1c03a4fa2
説明:
入力値が1より小さいかどうかチェックする。

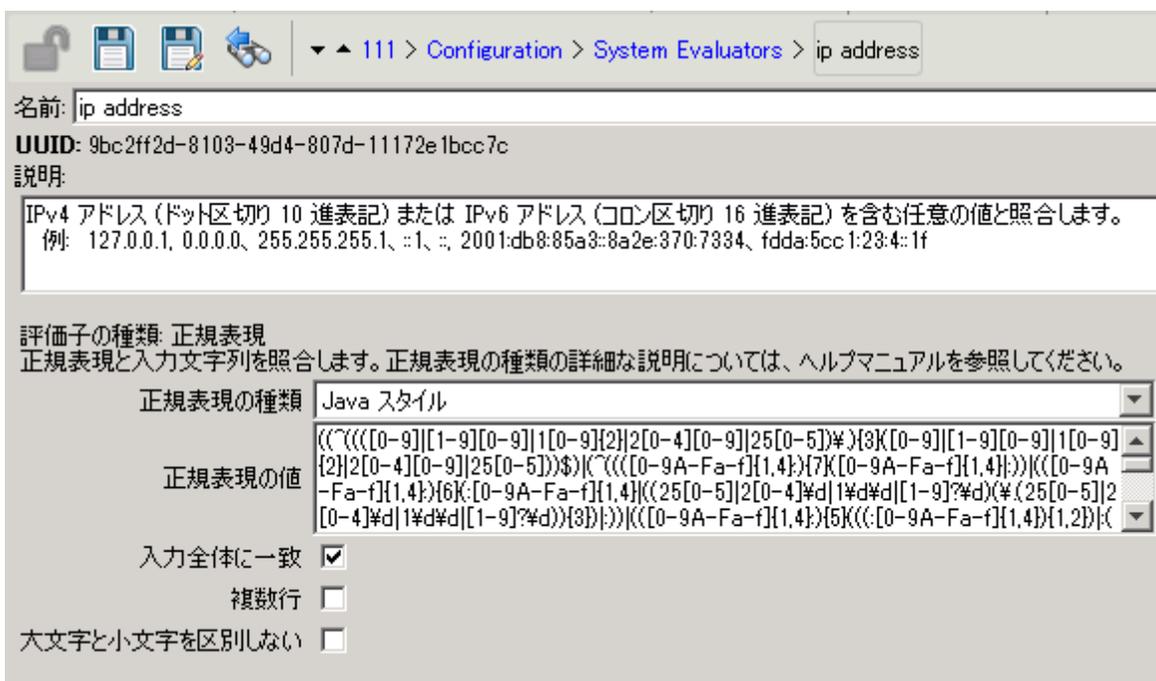
評価子の種類: <
入力を文字または数値と比較
比較対象: 1

フィルター入力のテスト
パターンが一致します (0 回)

GUI アイテム	説明
名前	システム評価子の名前が表示されます。
説明	評価子の説明を入力します。
比較対象	評価子の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。
フィルター入力のテスト	フィルターが想定どおりに動作するかテストするためのデータを配置する場所です。このセクションの GUI アイテムの詳細については、「 出力および結果のフィルタリング 」(255ページ)を参照してください。

評価子エディター - 正規表現

評価子エディターの外観は、選択した評価子の種類によって異なります。[評価子の選択] ダイアログボックスで [正規表現] を選択した場合、評価子エディターは次のように表示されます。

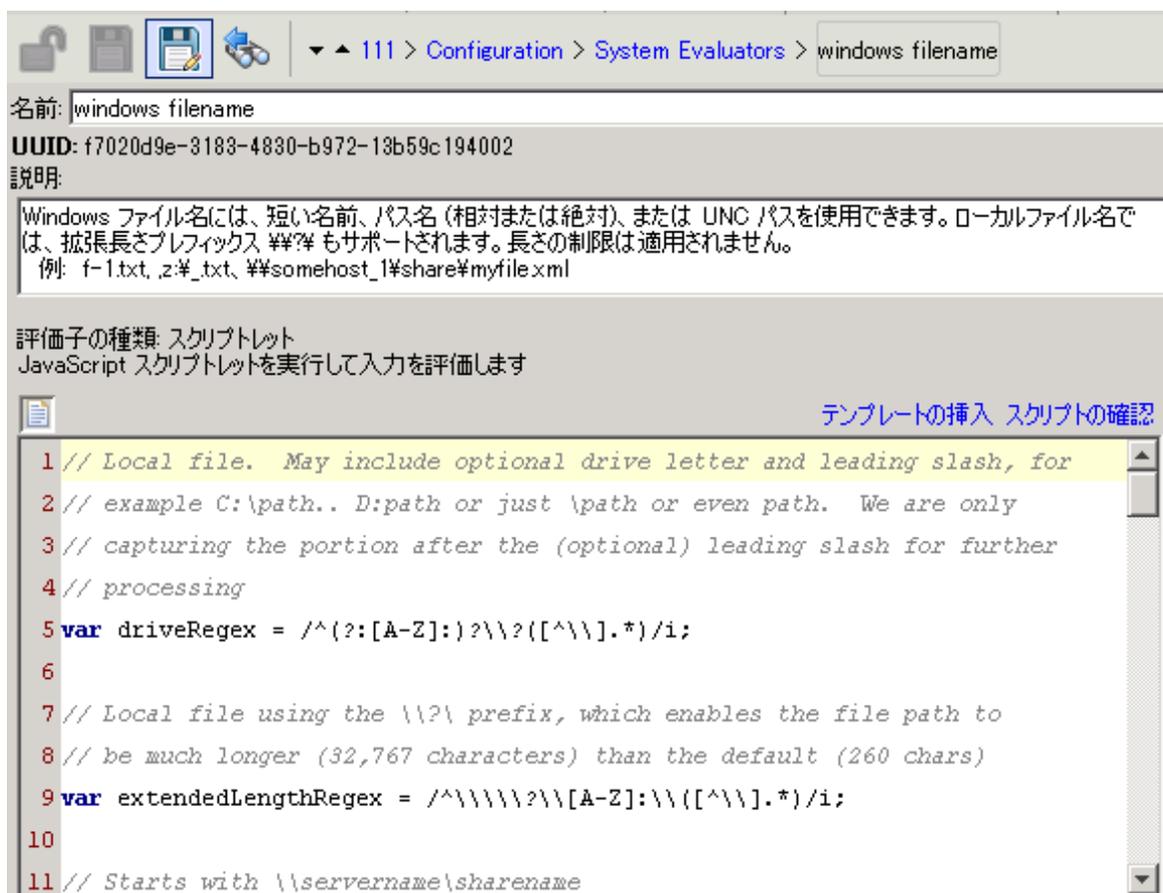


正規表現の作成の詳細については、「[フローでの正規表現の使用](#)」(302ページ)を参照してください。

GUI アイテム	説明
名前	システム評価子の名前が表示されます。
説明	評価子の説明を入力します。
正規表現の種類	正規表現の種類として [Java スタイル] を選択します。ほかのスタイルは使用しないでください。推奨されていません。
正規表現の値	正規表現を入力します。
入力全体に一致	入力全体に評価子を適用するには、これを選択します。
複数行	式で複数の行を使用できるようにする場合は、これを選択します。
大文字と小文字を区別しない	正規表現の大文字小文字を区別しない場合は、これを選択します。

評価子エディター - スクリプトレット

評価子エディターの外観は、選択した評価子の種類によって異なります。[評価子の選択] ダイアログボックスで [スクリプトレット] を選択した場合、評価子エディターは次のように表示されます。



スクリプトレットの作成の詳細については、「[フロー内でのスクリプトレットの使用](#)」(297ページ)を参照してください。

GUI アイテム	説明
名前	システム評価子の名前が表示されます。
説明	評価子の説明を入力します。
スクリプトレットアイコン 	このアイコンを Configuration\Scriptlets フォルダーにドラッグすると、スクリプトレットをフォルダーに保存して再利用できます。
テンプレートの挿入	[テンプレートの挿入] をクリックすると、スクリプトレットを書くのに役立つガイドラインが表示されます。
スクリプトの確認	[スクリプトの確認] をクリックすると、スクリプトレットにエラーがないか確認されます。

システムフィルターの設定

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

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

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

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

名前: TableFilter
UUID: 4b0f1926-57ad-4a62-a64c-01d9880fe815
説明:

入力をテーブルとして解析し、指定された列でソートします

列の区切り記号: スペース 行の区切り記号: 改行

最初の行はヘッダー: 結果の最初の行を削除:

ソート基準列: -1 昇順:

選択行: 0 選択列: 0

選択幅: 1 選択高: 1

フィルターのテスト

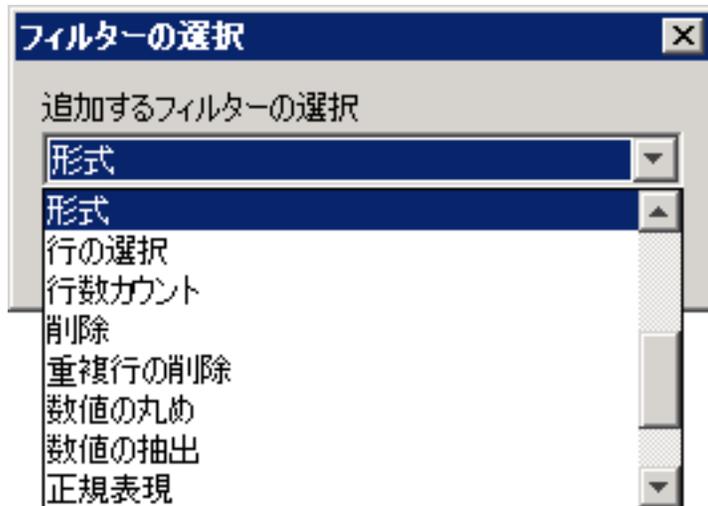
フィルター入力のテスト

クリア クイックコマンド

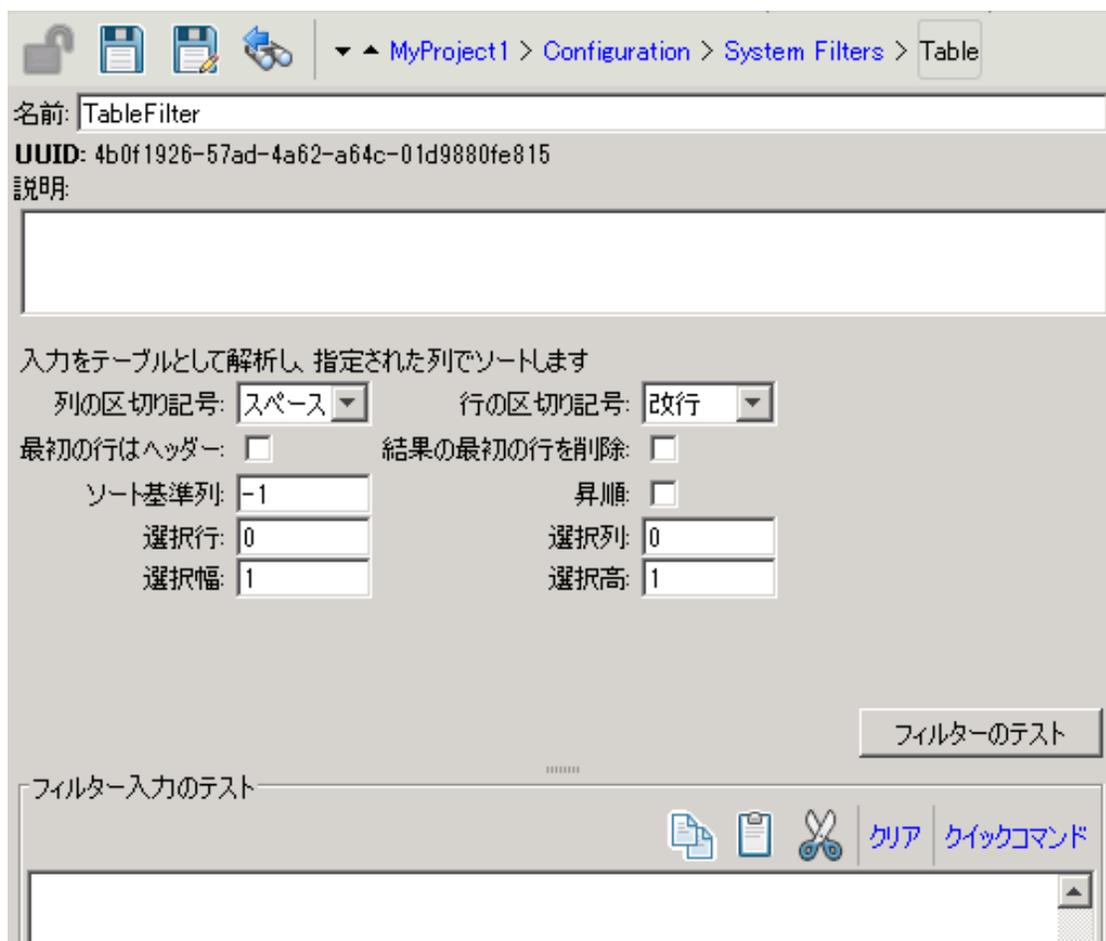
実行する操作

システムフィルターの作成

1. [プロジェクト] ペインで、**Configuration** フォルダを展開します。
2. **System Filters** フォルダで、右クリックして[新規]>[システムフィルター]を選択します。
3. [フィルターの選択] リストで、フィルターの種類を選択します。



4. フィルターの名前を入力して[OK]をクリックします。
フィルターエディターが開きます。フィルターエディターの外観は、選択したフィルターの種類によって異なります。



5. [説明] ボックスに、フィルターの目的を入力します。
6. 出力または結果のフィルターに使用するテキスト、文字列、式の値、またはスクリプトレットを入力します。それぞれのフィルターオプションの詳細については、「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。
7. フィルターをテストします。
 - a. [クリア] をクリックして[フィルター入力のテスト] ボックスの内容をクリアします。
 - b. [クイックコマンド] をクリックします。
 - c. 目的のデータを生成するコマンドを入力します。
 - d. [OK] をクリックします。コマンドの出力が[フィルター入力のテスト] ボックスに表示されます。

フィルターのテストの詳細については、「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。

8. [保存] をクリックします。

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

トラブルシューティング: ローカライズされた Windows のコマンドラインを使用している場合は、クイックコマンドを使用すると、コマンドラインのエンコーディングが原因で、結果が文字化けする可能性があります。

これを回避するには、

1. Studio を閉じます。
2. <インストールフォルダー>/studio/Studio.l4j.ini を開きます。
3. 以下を追加します。

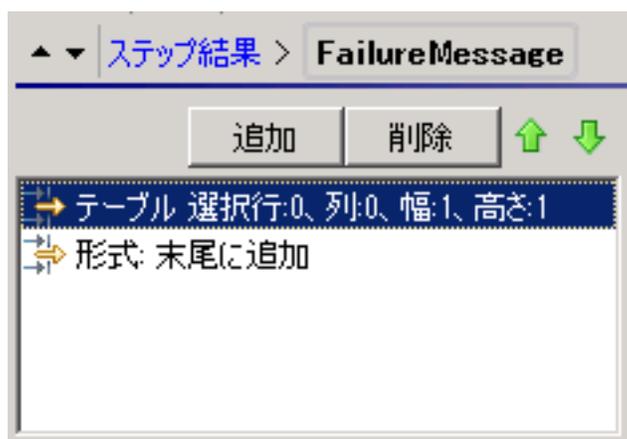
```
-Ddharma.windows.cmd.unicode=true
```

これにより、コマンドラインから読み込まれた文字が unicode に変換されるようになります。

コマンドラインを使用するオペレーションを含むフローの実行が複数ある場合には、このパラメーターを使用するとパフォーマンスに問題が生じる可能性があるので注意してください。フィルターのテストが完了したら、**Studio.l4j.ini** ファイルから削除してください。

既存のフィルターをシステムフィルターとして保存

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



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

出力または結果でのシステムフィルターの使用

1. システムフィルターを使用する出力または結果のフィルターエディターを開きます。
2. [プロジェクト] ペインで、**Configuration** フォルダと **System Filters** フォルダを展開します。

3. 使用するフィルターを、**System Filters** フォルダからフィルターエディターの[フィルター]リストにドラッグします。



システムフィルターの編集

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

システムフィルターのコピー

1. [プロジェクト]ペインで、**Configuration** フォルダと **System Filters** フォルダを展開します。
2. コピーするシステムフィルターを右クリックします。
3. [編集]>[コピー]を選択します。
4. コピー先のプロジェクトの **System Filters** フォルダに移動します。
5. 右クリックして、[編集]>[貼り付け]を選択します。

システムフィルターの削除

システムフィルターを削除する前に、[これを使用する項目は?]機能を使用して、ほかのアイテムが依存していないかをチェックすることをお勧めします。詳細については、「[フローとオペレーションの使用方法的詳細](#)」(366ページ)を参照してください。

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

参考資料

フィルターエディター

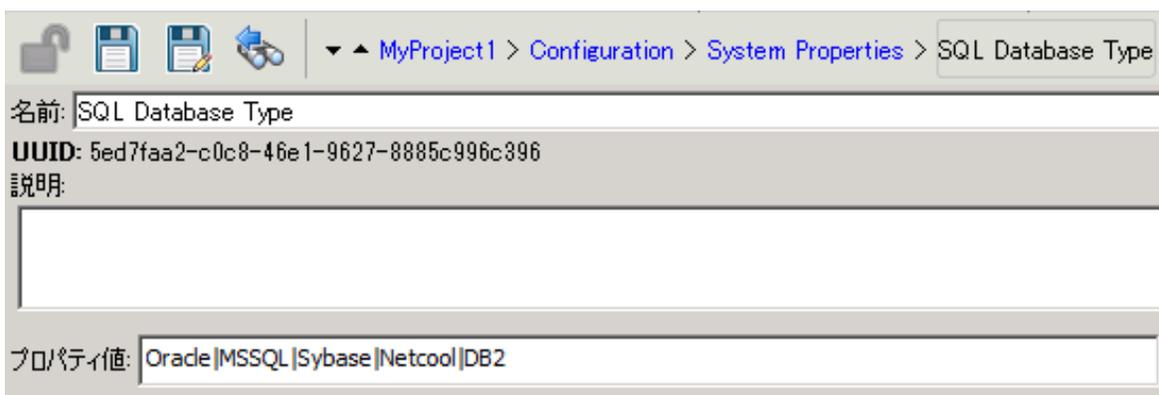
フィルターエディターの外観は、選択したフィルターの種類によって異なります。それぞれのオプションの詳細については、「出力および結果のフィルタリング」(255ページ)の「フィルターオプション」を参照してください。

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

システムプロパティの設定

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

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



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

ベストプラクティス

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

- 適用範囲がグローバルであり、実行時に、フロー実行のコンテキストの一部となるため。結果として、システムプロパティの値を変更すると、既存のオペレーションとフローが壊れる恐れがあります。
- 実行開始時にフローのコンテキストの一部となるため。
- すぐには表示されません。システムプロパティは、Studio Debugger ([システムプロパティ]の[コンテキストインスペクター])と **Configuration\System Properties** フォルダで表示されます。
さらに、フローを作成するときに入力を作成すると、フローの実行時に入力と同じ名前のフロー変数が自動的に作成されます。このため、システムプロパティと同じ名前の空のフロー変数が自動的に作成され、予期しない動作が得られることがあります。
- 同じ名前のフロー変数が優先されます。入力フロー変数からの値を取得可能な場合、つまり、フロー変数が存在しており、それに割り当てられている値がある場合は、フロー変数は、入力値のソースとしてシステムプロパティよりも優先されます。

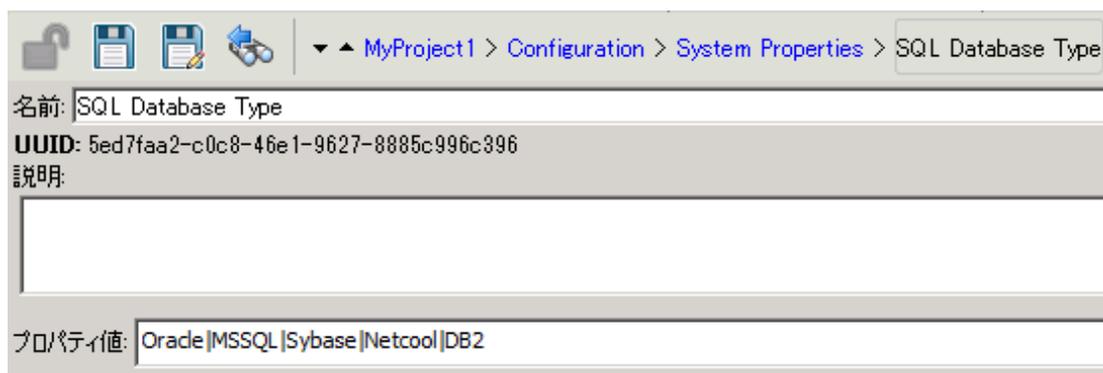
一方、システムプロパティの値は、入力値の割り当てや、システムプロパティの結果によって変更することはできません。値を、これら2種類のソースからシステムプロパティに割り当てると、システムプロパティと同じ名前のフロー変数が作成され、値がそのフロー変数に割り当てられます。

注: システムプロパティを動的に参照する場合は、完全なパスを使用する必要があります。たとえば、**domainName** という名前のシステムプロパティが **staging\domain** というフォルダー構造の下にある場合、**\${staging/domain}** という文字列を使用して参照します。

実行する操作

システムプロパティの作成

1. [プロジェクト] ペインで、**Configuration** フォルダーを展開します。
2. **System Properties** フォルダーで、右クリックして [新規] > [システムプロパティ] を選択します。
3. システムプロパティの名前を入力して [OK] をクリックします。
システムプロパティのエディターが開きます。



The screenshot shows a dialog box titled "SQL Database Type" within the "System Properties" folder of "MyProject1". The "名前:" field contains "SQL Database Type". The "UUID:" field contains "5ed7faa2-c0c8-46e1-9627-8885c996c396". The "説明:" field is empty. The "プロパティ値:" field contains a list of database types: "Oracle|MSSQL|Sybase|Netcool|DB2".

4. (オプション) [説明] ボックスに、システムプロパティの説明を入力します。
5. [プロパティ値] ボックスに、システムプロパティの値を入力します。区切り記号として | を使用します。
6. [保存] をクリックします。
システムプロパティが **System Properties** フォルダーに保存され、フローで使用できるようになります。

フローでのシステムプロパティ値の使用

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

注: オペレーションの [スクリプトレット] タブで必要なコマンドとその構文の詳細については、[テンプレートの挿入] をクリックします。テンプレートで、グローバルコンテキストを使用するために必要なコマンドが参照できます。詳細については、「[フロー内でのスクリプトレットの使用](#)」(297ページ)を参照してください。

システムプロパティの値の変更

システムプロパティの値を変更するには、次のような方法があります。

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

システムプロパティの名前の変更

1. [プロジェクト]ペインで、**Configuration** フォルダと **System Properties** フォルダを展開します。
2. 名前を変更するシステムプロパティを右クリックし、[名前の変更]を選択します。
3. システムプロパティの新しい名前を入力し、Enter を押します。

システムプロパティのコピー

1. [プロジェクト]ペインで、**Configuration** フォルダと **System Properties** フォルダを展開します。
2. コピーするシステムプロパティを右クリックします。
3. [編集]>[コピー]を選択します。
4. コピー先のプロジェクトの **System Properties** フォルダに移動します。
5. 右クリックして、[編集]>[貼り付け]を選択します。

システムプロパティの削除

システムプロパティを削除する前に、[これを使用する項目は?]機能を使用して、ほかのアイテムが依存していないかを確認することをお勧めします。詳細については、「[フローとオペレーションの使用法の詳細](#)」(366ページ)を参照してください。

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

参考資料

システムプロパティのエディター

名前: SQL Database Type
UUID: 5ed7faa2-c0c8-46e1-9627-8885c996c396
説明:
プロパティ値: Oracle|MSSQL|Sybase|Netcool|DB2

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

フロー作成 - 基本

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

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

この章では、単純なフローを作成するために必要な、すべての基本的なステップについて説明します。より複雑なフローの作成については、[「高度な作成」\(282ページ\)](#)を参照してください。

フローの作成 - 手順



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

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

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

ステップ 1: フローの作成

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

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

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

詳細情報とオプションについては、[「新しいフローの作成」\(188ページ\)](#)を参照してください。

ステップ 2: オペレーションをステップとして追加

オペレーションを作成ペインにドラッグし、フローのステップとして使用します。

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

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

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

詳細情報とオプションについては、「[フロー内でのステップの作成](#)」(191ページ)を参照してください。

ステップ 3: リターンステップの作成とフローの終了

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

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

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

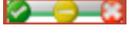
詳細情報とオプションについては、「[リターンステップの作成](#)」(278ページ)を参照してください。

ステップ 4: トランジションの作成

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

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



1. 次のステップに接続するステップ上で、いずれかのレスポンスを表すアイコン  をクリックし、そのレスポンスの結果発生するステップに線をドラッグします。
2. トランジションを修正する必要がある場合は、線をダブルクリックして、トランジションのインスペクターを開きます。
3. ステップの別のレスポンスアイコンに対し、同じ処理を繰り返します。

例では次のようになります。

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

トランジションの詳細については、「[トランジションの作成](#)」(228ページ)を参照してください。

オペレーションのレスポンスの設定の詳細については、「[レスポンスの設定](#)」(233ページ)を参照してください。

ステップ 5: フローへの入力の追加

1. フローを右クリックして [プロパティ] を選択します。
2. オペレーションエディターで、[入力] タブをクリックします。
3. 入力ツールバーの [入力の追加]  をクリックして、入力名を入力します。
4. 目的の行の [種類] 列をクリックし、リストから次のいずれかの値の割り当てタイプを選択します。

- 単一値 

- 値のリスト  - 複数のターゲットに対してオペレーションを実行できます。

5. [割り当て元] フィールドで、入力のデータソースを定義します。

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

詳細情報とオプションについては、「[入力の作成](#)」(205ページ)を参照してください。

ステップ 6: フローへの結果の追加

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

1. ステップのインスペクターで [結果] タブをクリックします。
2. ステップのインスペクターのツールバーの [結果の追加]  をクリックします。
3. [名前] 列に、結果の名前を入力し、キーボードのリターンキーを押します。この名前は、フロー変数またはフロー出力フィールドの名前に使用されます。
4. [ソース] リストから、結果のソースを選択します。たとえば、プライマリ出力を選択します。
5. [割り当て先] リストから、値の保存先としてフロー変数または出力フィールドを指定します。

6. [割り当て操作] リストから、OVERWRITE、APPEND、PREPEND、または算術割り当て操作のいずれかの適切な操作を選択します。

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

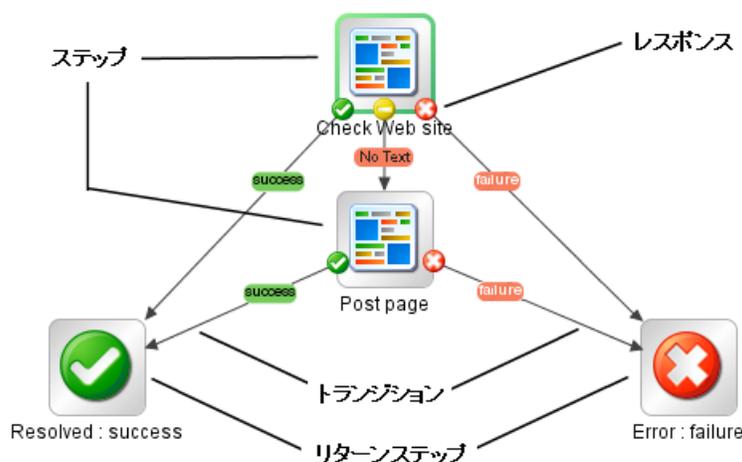
詳細情報とオプションについては、「[ステップの結果の設定](#)」(246ページ)を参照してください。

ステップ 7: フローの保存

[保存] ボタンをクリックします。

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

次の図は各種フロー要素を示しています。



新しいフローの作成

フローを作成するには2つの選択肢があります。

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

重要: プロジェクト内のフローを削除、作成、または名前変更する必要がある場合は、ファイルシステムでアイテムを削除、作成、または名前変更するのではなく、Studio 内から実行してください。

フローを最初から作成

1. フローを作成するプロジェクトを開きます。

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

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

テンプレートを使用したフローの作成

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

1. フローを作成するプロジェクトを開きます。



2. [ようこそ] 画面で、[新規フロー] ボタンをクリックします。
3. 表示されたテンプレートの一覧でいずれかのフローテンプレートを強調表示すると、その説明が表示されます。必要に応じて、スクロールバーをドラッグしてテキストを表示します。

利用できるフローテンプレート	
名前	パス
Restart Solaris Server	/Base [1.0.121]/Library/...
Restart Service	/Base [1.0.121]/Library/...
Network Check	/Base [1.0.121]/Library/...
Counter Average Thresh...	/Base [1.0.121]/Library/...
Check Windows CPU	/Base [1.0.121]/Library/...
Check for Windows Event	/Base [1.0.121]/Library/...

テンプレートの説明:

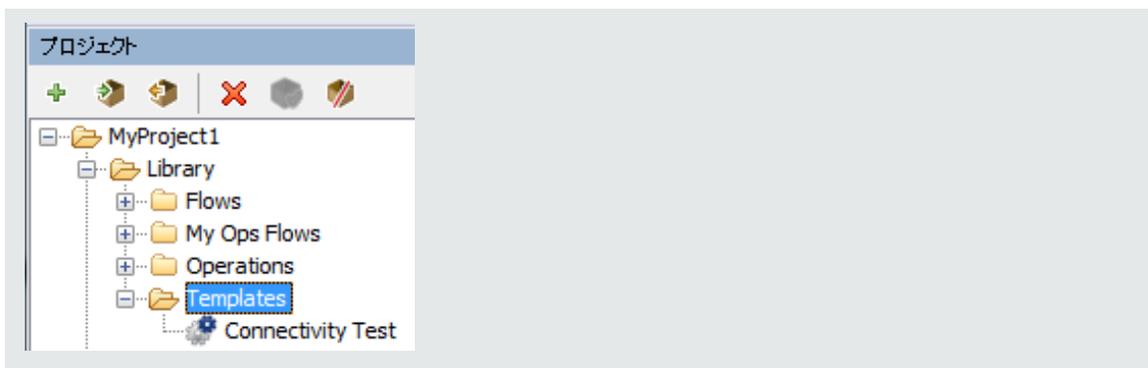
このフローを使用して Windows サービスを再起動します。入力を簡単に変更して、特定のサービスに固有のバージョンを作成するか、セキュリティ資格情報の標準セットを使用するようにできます。入力パラメーター: host - サービスを再起動する必要があるホスト。 service - 再起動する必要があるサービス。 altuser - ホストに接続するためのユーザー名 (例: domain(username))。 altpass - ホスト接続用のパスワード。 レスポンス: failure - サービスが存在しないか、そのステータスを確認できないが、起動できませんでした。 success - サービスが起動されました。



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

ヒント: テンプレートを [依存関係] ペインの **00-Base Content** コンテンツパックの **Templates** フォルダーから入手することもできます。テンプレートをダブルクリックして作成ペインで開きます。フローを修正する場合は、プロジェクトにコピーしてコピーを修正します。

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



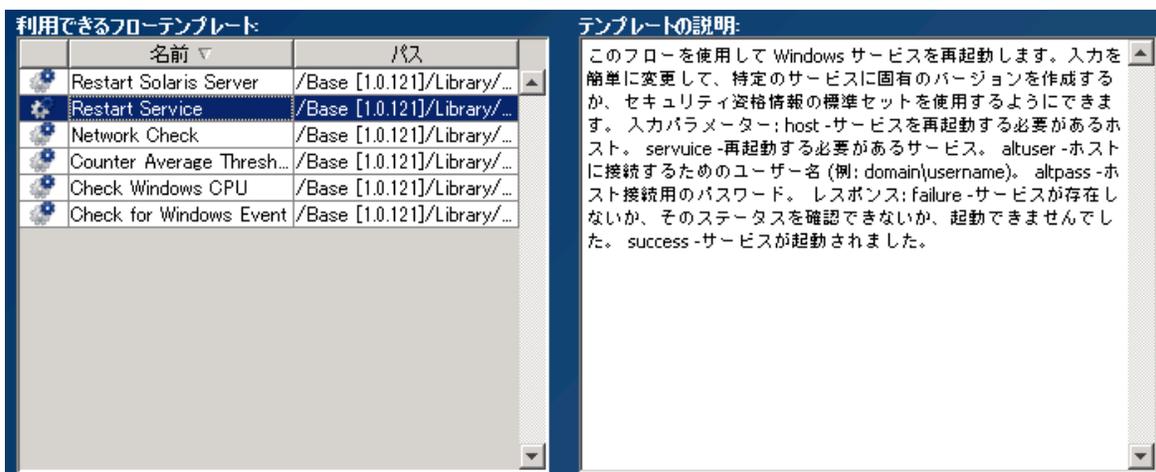
フローの説明の追加

1. 作成ペイン上でフローを右クリックし、[プロパティ]をクリックします。
2. [説明]タブをクリックします。
3. フローの説明を入力して[OK]をクリックします。

参考資料

利用できるフローテンプレート

HP 00 の [ようこそ] 画面で [新規フロー] ボタンをクリックすると、[利用できるフローテンプレート] リストが表示されます。



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

テンプレートの説明	選択したテンプレートの説明が表示されます。
-----------	-----------------------

フロー内でのステップの作成

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

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

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

実行する操作

オペレーションからのステップの作成

1. **[プロジェクト]**ペインまたは**[依存関係]**ペインで、フローに追加するオペレーションを選択します。

注: **[依存関係]**ペインのオペレーションは読み取り専用です。

2. プロジェクトツリーから作成ペインにオペレーションをドラッグします。
3. 必要に応じて、フロー内での機能を表すような名前を付けます (オペレーションの名前は、総称的すぎる場合があります)。
 - a. 名前を変更するステップを右クリックし、**[名前の変更]**を選択します。
 - b. 強調表示されたフィールドに新しい名前を入力します。
4. 必要に応じて、ステップを編集します。詳細については、[「フローの変更」\(200ページ\)](#)を参照してください。

読み取り専用オペレーションを編集可能にするためのプロジェクトへのコピー

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

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

レーションには影響しません。

5. プロジェクトツリーから作成ペインにオペレーションをドラッグします。

注: コンテンツパックからプロジェクトにコピーされたオペレーションは、「ソフトコピー」です。つまり、オペレーションが元々、アクションプラグインからインポートされたものである場合、コピーされたオペレーションは元のオペレーションへの参照を保持したままです。アクションプラグインをアップグレードして、新しいバージョンを呼び出すように元のオペレーションを更新すると、コピーされたオペレーションも自動的に更新されます。詳細については、「[オペレーションの作成](#)」(349ページ)を参照してください。

フロー内からのステップのコピー

ステップをコピーして貼り付けるには、次のいずれかのツールを使用します。

- 作成ペインツールバーの [コピー]  ボタンと [貼り付け]  ボタン
[編集] > [コピー] メニューコマンドと [編集] > [貼り付け] メニューコマンド
- 右クリックメニュー
- キーボードの組み合わせ (CTRL+C、CTRL+V)

ステップへの説明の追加

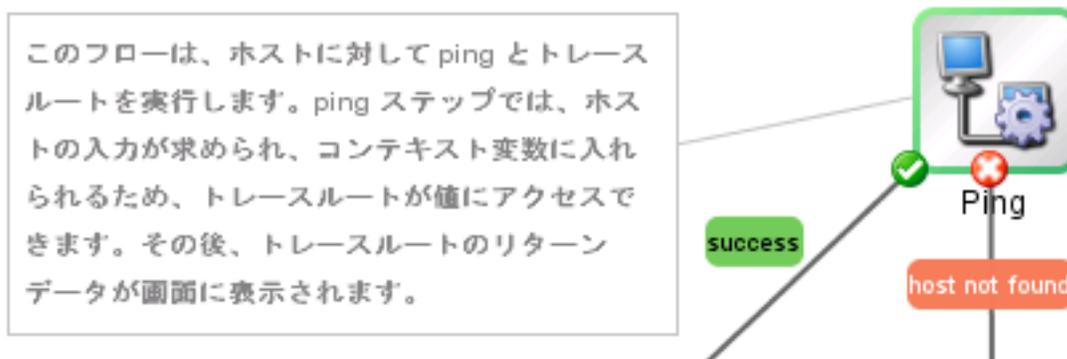
1. 作成ペイン上でステップを右クリックし、[プロパティ] をクリックします。
2. [説明] タブをクリックします。
3. ステップの説明を入力して [OK] をクリックします。

説明入力のベストプラクティスについては、「[作成のベストプラクティス](#)」(25ページ)を参照してください。

ステップへのコールアウトの追加

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

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



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

フローからステップを作成する (サブフロー)

サブフローとは、フロー内フローです。詳細については、「[フロー内でのサブフローの作成](#)」(282 ページ)を参照してください。

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

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

非ブロックステップを作成する

非ブロックステップは、残りのフローをブロックしません。フローの実行時に、非ブロックステップが実行されている間、非ブロックステップの後に配置されているステップが引き続き実行されます。

フローの後続のアクションがこのステップの結果に依存しない場合は、非ブロックステップを行うように選択します。たとえば、ステップで失敗レスポンスが発生した後で、フローによって通知を送信する必要があります。ただし、通知の送信中、フローを待機状態にする必要はありません。

注: 非ブロックステップを含む非階層的フローまたはサブフローでは、フローの実行は非ブロックステップが完了するまで終了しません。ただし、マルチインスタンスステップまたは並列ス

テップを含むフローの一部である場合には、フローの実行は非ブロックステップを待ちません。
マルチインスタンスステップまたは並列分岐ステップは非ブロックにすることもできません。

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^①を取得します。



2. 完了^②レスポンスを次のステップに接続します。

注: 非ブロックステップを通常のステップに戻すには、ステップを右クリックし、**[非ブロックの切り替え]**を再度クリックします。

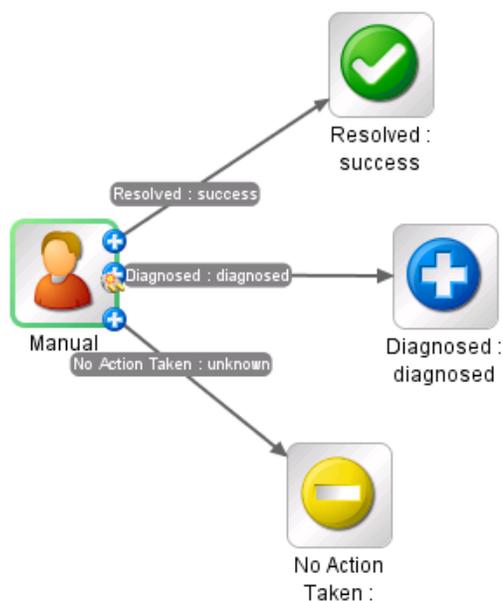
手動ステップを作成する

手動ステップは、操作を選択できるステップです。ユーザーが、実行時に操作を選択する必要があります。

手動ステップを作成するには、手動オペレーションテンプレートを、ベースコンテンツからコピーして、ユーザーが選択できる操作を定義します。

1. **[依存関係]** ペインで、手動オペレーションテンプレートを選択します。
2. **[編集]** > **[コピー]** を選択します。
3. コピーを貼り付ける場所をプロジェクトツリーで選択し、**[編集]** > **[貼り付け]** を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。
4. プロジェクトツリーから作成ペインにオペレーションをドラッグします。

5. ステップで、ユーザーが選択できる操作を追加します。



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

表示ステップの作成

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

表示ステップを作成するには、表示オペレーションテンプレートをベースコンテンツからコピーして、ユーザーに表示する情報を定義します。

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

1. **【依存関係】** ペインで、表示オペレーションテンプレートを選択します。
2. **【編集】** > **【コピー】** を選択します。
3. コピーを貼り付ける場所をプロジェクトツリーで選択し、**【編集】** > **【貼り付け】** を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。
4. プロジェクトツリーから作成ペインにオペレーションをドラッグします。
5. ステップのインスペクターを開いて、**【表示】** タブをクリックします。
6. **【このステップの実行前に常にユーザーに確認する】** チェックボックスをオンにします。
7. **【プロンプトタイトル】** ボックスに、プロンプトのラベル (最大 128 文字) を入力します。
8. **【プロンプトの幅】** ボックスに、プロンプトの幅をピクセル単位で入力します。

9. [高さ] ボックスに、プロンプトの高さをピクセル単位で入力します。
10. [プロンプトテキスト] ボックスに、ユーザーへのメッセージを入力します。
11. [OK] をクリックし、変更を保存します。

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

参考資料

ステップのインスペクター > [表示] タブ

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

ステップ名: SQL Query

入力 | 結果 | 表示 | 説明 | 詳細 | スクリプトレット

このステップの実行前に常にユーザーに確認する

プロンプトタイトル: _____

プロンプトの幅: 0 高さ: 0

プロンプトテキスト

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

プロンプトテキスト

プロンプトウィンドウの本文に表示するメッセージを入力します。メッセージに変数を含めることができます。たとえば、`${dateTime}` のようになります。

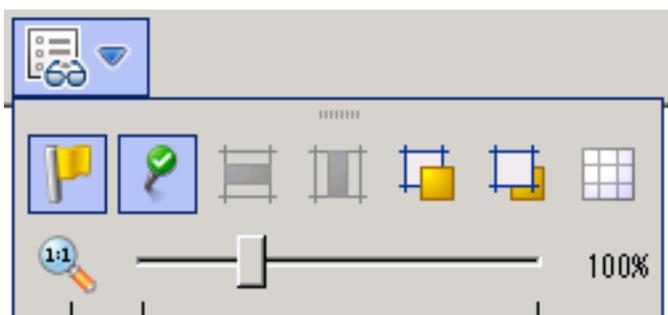
フローの外観の調整

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

作成ペインのツールバーから**表示オプションボタン**



をクリックして、**表示オプションパレット**を表示します。



100%表示
に戻す

ズームを変更するスライダー

実行する操作

ステップのグリッドへのスナップ

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

1. 作成ペインの背景にグリッドが表示されていない場合は、**表示オプションパレット**の**[グリッドの表示/非表示]** ボタン  をクリックします。作成ペインにオペレーションをドラッグすると、グリッドの一番近い線にスナップされます。
2. グリッドのある線から別の線にステップを動かすには、ステップを少し動かしてマウスを放します。

注: デフォルトでは、作成ペインにグリッドは表示されません。**[グリッドの表示/非表示]**  ボ

タンを使用してグリッドを表示するように設定した場合、この状態は Studio を閉じた後では記憶されません。デフォルトの動作を変更して、グリッドがデフォルトで表示されるようにするには、**Studio.properties** ファイルを開き、プロパティ `dharma.studio.ui.activegrid=true` を設定します。

ステップの整列

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

レスポンスラベルおよびアイコンの表示または非表示

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

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

オブジェクトの前面または背面への移動

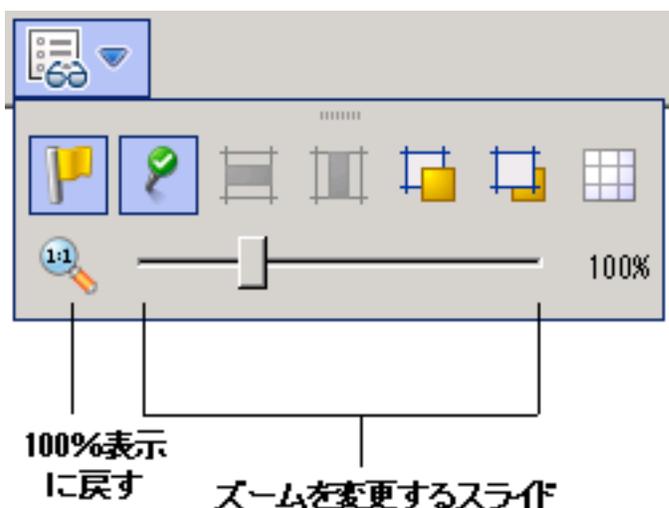
フローでは、作成ペインでオブジェクトを重ねることができます。フロー内に多くのアイテムがある場合、フローが長くなる場合があります。このような場合、最も重要なオブジェクトを前面に移動する必要があります。

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

参考資料

表示オプションパレット

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



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

フローの変更

フローを作成したら、フローを変更できます。たとえば、前に作成したフローをコピーして、多少異なる用途に適応できます。HP 00 で提供されている、[Accelerator Packs] フォルダ内にあるようなすぐに使えるフローを、ニーズに合わせて適応することもできます。

ベストプラクティス

変更する前に、必ずフローをコピーしてください。

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

フローをコピーするときに、オペレーションのプロパティの変更が必要になる可能性がある場合は、[ディープコピー]コマンドを使用することをお勧めします。このコマンドを使用すると、フローとともにオペレーションがコピーされるため、元の内容に影響を与えずに変更できます。「[フローおよびオペレーションのコピー](#)」(363ページ)を参照してください。

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

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

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

実行する操作

フローを開いて編集する

- [プロジェクト]ペインでフローをダブルクリックし、作成ペインで開いて編集します。
- 作成ペインで複数のフローを開くには、Shift キーまたは Ctrl キーを使用して選択し、右クリックして[開く]を選択します。

ステップに移動して編集する

複数のステップが含まれている複雑なフローがある場合は、[ステップに移動]  ボタンを使用して、編集する必要のあるステップに素早く移動します。

1. フローを作成ペインで開きます。
2. 作成ペインのツールバーで、[ステップに移動]  をクリックします。
3. 移動先のステップの名前を入力するか、ステップの最初の何文字かを入力してリストから選択します。
4. 必要に応じて、キーボードの上向き矢印と下向き矢印を使用して、ステップのリスト内を移動します。

開始ステップを変更する

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

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

ステップの名前を変更する

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

ベストプラクティス: ステップの目的を明確に表す名前を指定します。

フローまたはオペレーションの名前を変更する

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

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

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

フロー図のステップのグループを移動する

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

アイコンを変更する

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

1. [アイコン]ペインを開いて[アイコン]タブをクリックします。



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

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

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

ステップのインスペクターでステップを変更する

1. ステップを右クリックし、[プロパティ]を選択します。ステップのインスペクターが開きます。
2. ステップのインスペクターでステップを変更できます。

- ステップの入力を追加または編集する。詳細については、[「入力の作成」\(205ページ\)](#)を参照してください。
- ステップの結果を追加または編集する。詳細については、[「オペレーションの出力の設定」\(243ページ\)](#)を参照してください。
- ステップの説明を追加または編集する。詳細については、[「フロー内でのステップの作成」\(191ページ\)](#)を参照してください。
- ステップのユーザープロンプトを追加または編集する。詳細については、以下の「ステップのユーザープロンプトを表示する」を参照してください。
- ステップのスク립トレットを追加または編集する。詳細については、[「フロー内でのスク립トレットの使用」\(297ページ\)](#)を参照してください。

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

ステップのユーザープロンプトを表示する

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

1. ステップを右クリックし、**[プロパティ]** を選択します。
2. ステップのインスペクターで **[表示]** タブをクリックします。
3. **[このステップの実行前に常にユーザーに確認する]** チェックボックスをオンにします。
4. **[プロンプトタイトル]** ボックスに、プロンプトのラベル (最大 128 文字) を入力します。
5. **[プロンプトの幅]** ボックスに、プロンプトの幅をピクセル単位で入力します。
6. **[高さ]** ボックスに、プロンプトの高さをピクセル単位で入力します。
7. **[プロンプトテキスト]** ボックスに、ユーザーへのメッセージを入力します。
8. **[OK]** をクリックし、変更を保存します。このステップに、表示プロンプトを示す青い矢印が付与されます。



ステップのベースとなるオペレーションを変更する

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

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

参考資料

ステップのインスペクター -> [表示] タブ

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

ステップ名: SQL Query

入力 | 結果 | 表示 | 説明 | 詳細 | スクリプトレット

このステップの実行前に常にユーザーに確認する

プロンプトタイトル:

プロンプトの幅: 0 高さ: 0

プロンプトテキスト

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

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

入力の作成

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

各入力は変数にマッピングされます。その値は以下の方法で設定できます。

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

フロー、オペレーション、またはステップに入力を作成できます。

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

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

注:

- 赤で表示される入力は、必要なオペレーション入力から取得されるステップ入力で、そのステップで削除された入力です。たとえば、次のように表示されます。

queryValueList	<input type="checkbox"/>		queryValueList	ユーザーへのプロンプト (テキストから)		queryValueList
delimiter	<input type="checkbox"/>		delimiter	ユーザーへのプロンプト (テキストから)		delimiter
url	<input checked="" type="checkbox"/>		url	ユーザーへのプロンプト (テキストから)		url

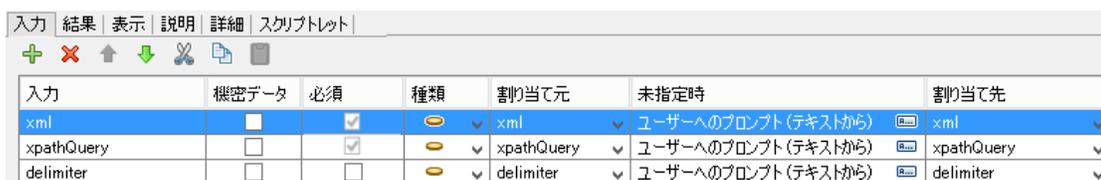
- グレーで表示される入力は、オプションのオペレーション入力から取得されるステップ入力、そのステップで削除された入力です。

- 2つのペインの間にある分割バーをドラッグして各ペインのサイズを変更できます。表示されていないフィールドを表示するには、各ペインの下部にあるスクロールバーを使用します。Studio を次に起動したときに、分割バーは同じ位置にあります。
- [入力] ペインでは常に1つの入力が増強表示されます。入力を削除すると、[入力] ペイン内の最初の入力が自動的に強調表示されます。
- [入力] ペイン内の列はドラッグアンドドロップ操作で移動できます
- 入力は、入力リスト内の位置に従って上から順番に処理されます。

実行する操作

入力を作成する

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



入力	機密データ	必須	種類	割り当て元	未指定時	割り当て先
xml	<input type="checkbox"/>	<input checked="" type="checkbox"/>	xml	ユーザーへのプロンプト (テキストから)	xml	xml
xpathQuery	<input type="checkbox"/>	<input checked="" type="checkbox"/>	xpathQuery	ユーザーへのプロンプト (テキストから)	xpathQuery	xpathQuery
delimiter	<input type="checkbox"/>	<input type="checkbox"/>	delimiter	ユーザーへのプロンプト (テキストから)	delimiter	delimiter

3. 既存の入力がある場合は、新しい入力を追加する行を選択します。たとえば、最初の行を選択した場合、新しい入力は2行目に表示され、リストの2番目のアイテムになります。
4.  ボタンをクリックします。
5. 新しい入力の名前を入力して、[OK] をクリックします。入力が新しい行に表示されます。

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

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

• **単一値** 

- **値のリスト**  - 複数の値が含まれる入力を使用してオペレーションを実行できます。[入力区切り記号] フィールドで指定した区切り文字で区切られたリストで値を指定する必要があります。

8. 入力のインスペクターで入力ソースを指定します。

- 入力と同じ名前の変数から値を割り当てるには、[割り当て元] ドロップダウンの上部に表示されるデフォルト名 (<デフォルト> プレフィックスで示されます) を選択します。
- 別のフロー変数から値を割り当てるには、[割り当て元変数] ドロップダウンで変数名を選択します。
- この入力のソースを定義しない場合は、[割り当て元] ドロップダウンで <未割り当て> を選択します。

入力ソースの種類を定義する方法については、[「入力ソースの指定」\(214ページ\)](#)を参照してください。

注: Studio 10.50 をインストールしているときにオペレーション、フロー、またはステップの新しい入力を作成する場合、[割り当て元]/[割り当て先] のデフォルト値が自動的に <未割り当て> に設定されます。この機能は、**Studio.properties** ファイルの **dharmastudio.ui.inputinspector.assignfrom.selected** プロパティまたは **dharmastudio.ui.inputinspector.assignto.selected** プロパティを使用して変更できます。

このデフォルト値は、新しいステップの入力、および [プロジェクト] または [依存関係] ペインからドラッグされたステップにも適用されます。

Studio.properties ファイルのパラメーターの詳細については、[「Studio プロパティの設定」\(381ページ\)](#)を参照してください。

9. **[未指定時]** リストで、[割り当て元] ボックスで指定したフロー変数が存在しない場合、またはこのフロー変数に値が格納されていない場合に実行する操作を選択します。入力名をクリックすると、右側のペインが自動的に更新されて入力の詳細が表示されます。  をクリックして、[未指定時] ダイアログボックスを開くか、右側のペインのフィールドに値を直接入力します。使用できるオプションは、次のとおりです。
- **定数を使用する:** 入力に使用する定数値を入力します。たとえば、常に使用する IP アドレスを入力します。
 - **ユーザーへのプロンプト:** フローの開始時に、情報を入力するかリストから選択して情報を指定するようにユーザーに求めるプロンプトを設定します。
 - **入力項目: テキスト** - ユーザーに情報の入力を求めるプロンプトを設定します[**ユーザーメッセージ:**] ボックスに表示されるプロンプトを入力します。たとえば、イベント ID の値を入力しますと入力します。

- **入力項目: 選択** - ユーザーに値の選択を求めるプロンプトを設定します。ユーザーが値を選択するときに表示されるリストの種類を選択します。

選択リスト: [名前:] ドロップダウンリストから選択リストを選択します。すべての選択リストが表示されます。

ドメインターム: [名前:] ドロップダウンリストからドメインタームを選択します。すべてのドメインタームが表示されます。

フロー変数: フロー変数の名前を選択します。フロー変数はフローの結果で定義されます。[ソース区切り文字] フィールドにフロー変数の区切り文字を入力します。

注: フローの途中のステップでは、[ユーザーへのプロンプト]を使用しないことをお勧めします。

- **前のステップの結果を使用する:** 入力の [割り当て元] 変数が存在しない場合または変数の値がない場合に使用する、前のステップの結果を選択します。

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

- **システムアカウント:** システムアカウント名と資格情報の種類を選択します。
- **ログインしたユーザーの資格情報:** ログインしたユーザーの資格情報の種類を選択します。

注: [種類] フィールドで、入力を [値のリスト] として定義した場合は、[システムアカウント] フィールドと [ログインしたユーザーの資格情報] フィールドは表示されません。

さまざまなタイプの入力ソースの詳細については、「[入力ソースの指定](#)」(214ページ)の関連するタスクを選択してください。

このフィールドに表示されるデフォルト値は、`studio.properties` ファイルの設定によって決まります。これらの設定の詳細については、「[Studio プロパティの設定](#)」(381ページ)の `dharmastudio.ui.inputinspector.assignfrom.selected` パラメーターを参照してください。

注: オペレーション、フロー、またはステップの新しい入力を作成する場合、[未指定時]のデフォルト値が自動的に[定数を使用する]に設定されます。

10. 入力エディターで入力ターゲットを指定します。
 - 入力と同じ名前の変数に値を割り当てるには、[割り当て先] ドロップダウンの上部に表示されるデフォルト名 (<デフォルト> プレフィックスで示されます) を選択します。
 - 別のフロー変数に値を割り当てるには、[割り当て先] ドロップダウンで変数名を選択します。

- この入力のターゲットを定義しない場合は、**[割り当て先]** ドロップダウンで <未割り当て> を選択します。

11. デフォルトでは、Studio によって入力と同じ名前のフロー変数が作成されます。この変数をフロー内の後続のステップで使用できます。ウィンドウの右側にある **[割り当て先変数]** フィールドでこの名前を変更できます。

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

12. (オプション) 入力の形式が正しいかどうかを確認するには、**[検査形式:]** リストからシステム評価子を選択します。

たとえば、入力が電子メールアドレスの場合、評価子を使って、入力された電子メールの形式が正しいかどうかを確認できます。評価子の作成の詳細については、[「入力データの評価」\(227ページ\)](#)を参照してください。

たとえば、フローでユーザーが電子メールアドレスを入力する必要がある場合、評価子を使って、入力された電子メールの形式が正しいかどうかを確認できます。評価子の作成の詳細については、[「入力データの評価」\(227ページ\)](#)を参照してください。

13. (オプション) 値を記録して診断や監査に使用できるようにするには、**[次の名前で記録:]** リストから **[<実行履歴>]** を選択します。

- **<実行履歴>**

- リスト内のいずれかのドメインアイテム

入力の名前を変更する

1. **[プロパティ]** シート (フローまたはオペレーションの場合) または入力のインスペクター (ステップの場合) を開きます。

2. **[入力]** タブで入力名をダブルクリックするか、**F2** を押して、新しい名前を入力します。

または:

右側のペインで、**[名前]** フィールドに新しい入力名を入力します。

入力を削除する

1. **[プロパティ]** シート (フローまたはオペレーションの場合) または入力のインスペクター (ステップの場合) を開きます。

2. **[入力]** タブで、削除する入力を選択し、入力のインスペクターのツールバーで  **[入力の削除]** をクリックします。削除したステップ入力はイタリック体で表示されます。**Ctrl** キーおよび **Shift** キーを使用して複数の入力を選択できます。

削除したデフォルトの入力を復元する

デフォルトの入力を **[入力]** タブから削除した場合、この入力を復元できます。デフォルトの入力は、ステップのベースとなるオペレーションの一部として作成された入力です。削除したデフォルトの入

力は、入力リストにグレーのイタリック体で表示されます。

1. [入力] タブで、[入力の追加] をクリックします。
2. 復元する入力の正確な名前を入力します。
3. 必要に応じて入力の任意のフィールドを編集できます。
4. [OK] をクリックします。

赤/グレーの色は表示されなくなり、入力は黒で表示されます。

参考資料

入力のインスペクター > ツールバー



GUI アイテム	説明
入力の追加	新しい入力行を追加します。新しい入力の名前を入力して、[OK] をクリックします。デフォルト値を使用して新しい入力の属性が割り当てられます。
入力の削除	選択した入力行を削除します。
上に移動	リスト内の選択した入力行を上に移動します。
下に移動	リスト内の選択した入力行を下に移動します。
切り取り	選択した入力行を現在の場所から削除します。その行を新しい場所に置くには、[貼り付け] オプションを使用します。 注: ステップレベルでオペレーション入力を削除しようとした場合、自動的にイタリック体で表示され、入力の表の一番下に移動されます。必須の入力は赤で表示され、オプションの入力はグレーで表示されます。 [種類]、[割り当て先/割り当て元]、または[未指定時] フィールドを変更することによって入力を元に戻すことができます。
コピー	選択した入力行をコピーします。コピーした行を新しい場所に置くには、[貼り付け] オプションを使用します。
貼り付け	コピーするか切り取った入力行を現在の場所に貼り付けます。入力名が既に存在している場合は、メッセージが表示され、入力は貼り付けられません(これはデフォルトの動作です)。 入力名が既に存在している場合、新しい入力に自動的に <入力> (コピー 1) という名前が付けられます。

注: ステップ、フロー、またはオペレーションの間で複数の入力の切り取り、コピー、貼り付けを行うこともできます。Ctrl キーを押しながら追加の入力を選択し、その行をクリックします。

入力のインスペクター > [入力] タブ

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

入力	機密データ	必須	種類	割り当て元	未指定時	割り当て先
xml	<input type="checkbox"/>	<input checked="" type="checkbox"/>	 xml	xml	ユーザーへのプロンプト(テキストから)	xml
xpathQuery	<input type="checkbox"/>	<input checked="" type="checkbox"/>	 xpathQuery	xpathQuery	ユーザーへのプロンプト(テキストから)	xpathQuery
delimiter	<input type="checkbox"/>	<input type="checkbox"/>	 delimiter	delimiter	ユーザーへのプロンプト(テキストから)	delimiter

GUI アイテム	説明
入力	入力名を表示します。
機密	<p>Central ユーザーインターフェイス、および Studio の内部デバッガーとリモートデバッガーでこの入力を非表示にします。このチェックボックスがオンになっている場合、Central と Studio で入力値が暗号化されてアスタリスクとして表示され、保持されません。</p> <p>このフラグは一時的であり、機密変数から割り当てられると、割り当てられた変数も機密になります。一時的な効果が見られるのはランタイム実行中のみであり、Studio ユーザーインターフェイスには反映されません。</p> <p>スクリプトレットで使用すると、機密データが暗号化された状態で取得されます。</p> <p>割り当てのための式を使用すると、機密性が保持されます。たとえば、<code>\${input1}</code> のようになります。</p>
必須	この入力を必須にします。
種類 	<p>[種類] リストで、入力の値を取得する方法を指定します。次のオプションを選択できます。</p> <ul style="list-style-type: none"> 単一値  値のリスト  - 複数のターゲットに対してオペレーションを実行できます。
割り当て元	入力の値の取得元を指定します。[割り当て元] リストから値の取得元の入力を選択します。

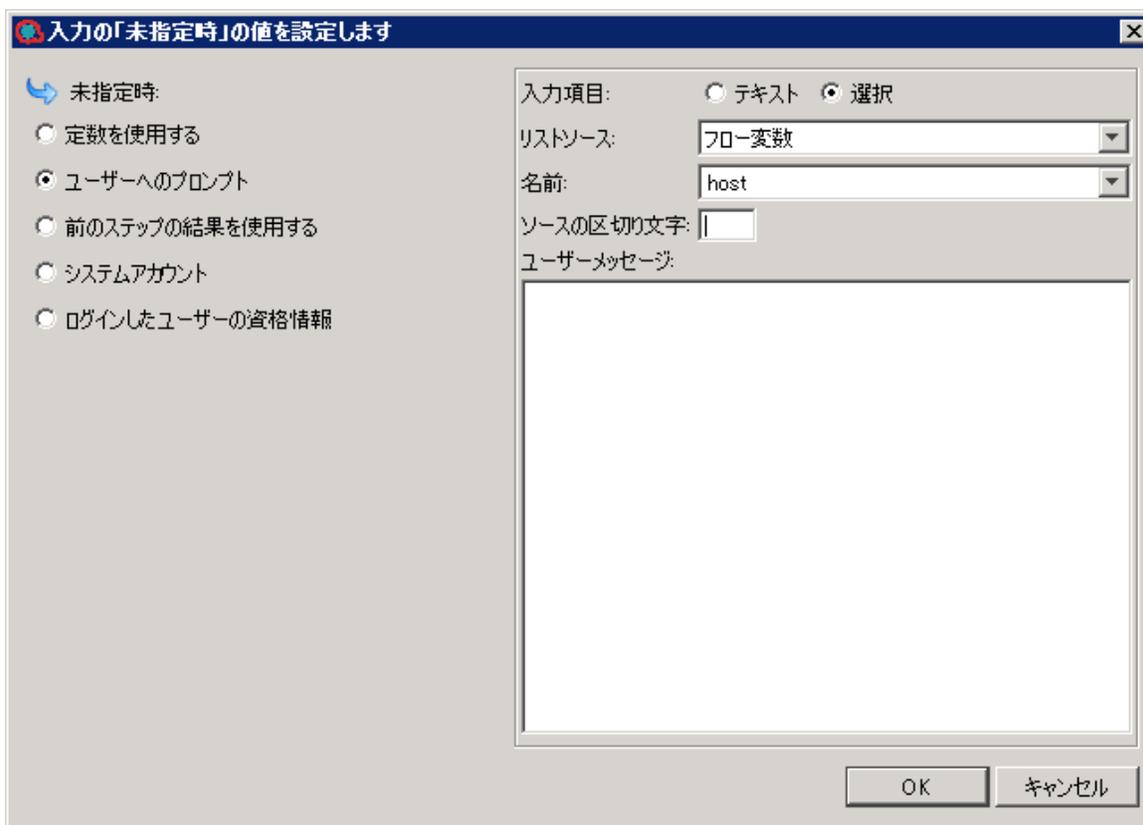
未指定時	<p>[割り当て元] 列で指定したフロー変数が存在しない場合またはその変数に値が格納されていない場合に実行する必要がある操作が表示されます。</p> <p>入力名をクリックすると、右側のペインが自動的に更新されて入力の詳細が表示されます。</p> <p> をクリックして、[未指定時] ダイアログボックスを開くか、右側のペインのフィールドに値を直接入力します。</p>
割り当て先	入力の値を割り当てるフロー変数を選択します。

注: [割り当て元] および [割り当て先] を直接編集するには、フィールド内をダブルクリックし、新しい値を入力します。

入力のインスペクター > [入力] タブ > [未指定時] ペイン/[未指定時] ダイアログボックス

入力のインスペクターの右側に [未指定時] ペインが表示されます。このペインで、[割り当て元] 列で指定したフロー変数が存在しない場合またはその変数に値が格納されていない場合に実行する必要がある操作を定義します。これらのフィールドの一部は、[未指定時] ダイアログボックスにも表示されます。

[未指定時] ペイン



[未指定時] ダイアログボックス

注: [種類] フィールドで、入力を [値のリスト] として定義した場合は、[システムアカウント] フィールドと [ログインしたユーザーの資格情報] フィールドは表示されません。

GUI アイテム	説明
定数を使用する	入力に使用する定数値を入力します。たとえば、常に使用する IP アドレスを入力します。
ユーザーへのプロンプト	<p>フローの開始時に、情報を入力するリストから選択して情報を指定するようにユーザーに求めるプロンプトを設定します。</p> <ul style="list-style-type: none"> • 入力項目: テキスト - ユーザーに情報の入力を求めるプロンプトを設定します[ユーザーメッセージ:] ボックスに表示されるプロンプトを入力します。たとえば、イベント ID の値を入力しますと入力します。 • 入力項目: 選択 - ユーザーに値の選択を求めるプロンプトを設定します。ユーザーが値を選択するときに表示されるリストの種類を選択します。 <p>選択リスト: [名前:] ドロップダウンリストから選択リストを選択します。すべての選択リストが表示されます。</p> <p>ドメインターム: [名前:] ドロップダウンリストからドメインタームを選択します。すべてのドメインタームが表示されます。</p> <p>フロー変数: フロー変数の名前を選択します。フロー変数はフローの結果で定義されます。[ソース区切り文字] フィールドにドメインタームの区切り文字を入力します。</p> <p>注: フローの途中のステップでは、[ユーザーへのプロンプト] を使用しないことをお勧めします。</p>
前のステップの結果を使用する	<p>入力に値がない場合に使用する、前のステップの結果を選択します。</p> <p>たとえば、最初のステップで情報を検索し、この情報をフロー変数として保存してから、2 番目のステップでその情報を表示できます。2 番目のステップでは、最初のステップで作成したフロー変数が使用されます。</p>
システムアカウント:	システムアカウント名と資格情報の種類を入力します。
ログインしたユーザーの資格情報	ログインしたユーザーのユーザー名またはパスワードを入力します。

リストソース	<p>ドロップダウンメニューから次のいずれかを選択します。</p> <ul style="list-style-type: none">• [選択リスト]- あらかじめ定義されているリストのセットから選択する。 [名前] リストで、ユーザーに提示するリストを選択します。 ヒント: リストを作成し、事前定義済みリストのセットに追加することもできます。リストの作成方法については、ユーザープロンプトの選択リストの作成を参照してください。• [ドメインターム]- ドメインタームは、専用の選択リストです。例として、特定の種類のサーバーに対してフローを実行するように指定する場合、システム内の各種サーバーのドメインタームを追加して、あるフローの実行対象にするサーバーの種類をユーザーが選択できるようなユーザープロンプトを作成することが可能です。 [名前] リストで、ユーザーに提示するドメインタームリストを選択します。• [フロー変数]- フロー変数の内容からリストを作成します。 [名前] リストで、リストに含むフロー変数を入力または選択します。異なるフローからフロー変数を参照できます。たとえば、folderA\folderB\flow1 というフォルダー構造の下にあるフローを参照するには、\${folderA/folder/flow1} という文字列を使用します。 [ソース区切り記号] ボックスに、リスト内の各要素を区切る記号を入力します。
ユーザーメッセージ	ユーザーに入力してもらうデータの内容を知らせるプロンプトメッセージを入力します。

入力ソースの指定

フローやオペレーション、ステップでの入力設定時、選択できる入力ソースの指定方法がいくつかあります。

- 「[単一の固定値を入力ソースに指定](#)」
- 「[単一のユーザー入力テキストを入力ソースに指定](#)」
- 「[単一のユーザー選択を入力ソースに指定](#)」
- 「[単一の入力値のリストを入力ソースに指定](#)」
- 「[ユーザーが入力したテキストから取得する入力値のリストを入力ソースに指定](#)」
- 「[単一のユーザー選択を入力ソースに指定](#)」

- 「[入力ソースの指定](#)」
- 「[システムアカウントの詳細としての入力ソースの指定](#)」
- 「[ユーザーのログイン名またはパスワードを入力ソースに指定します。](#)」

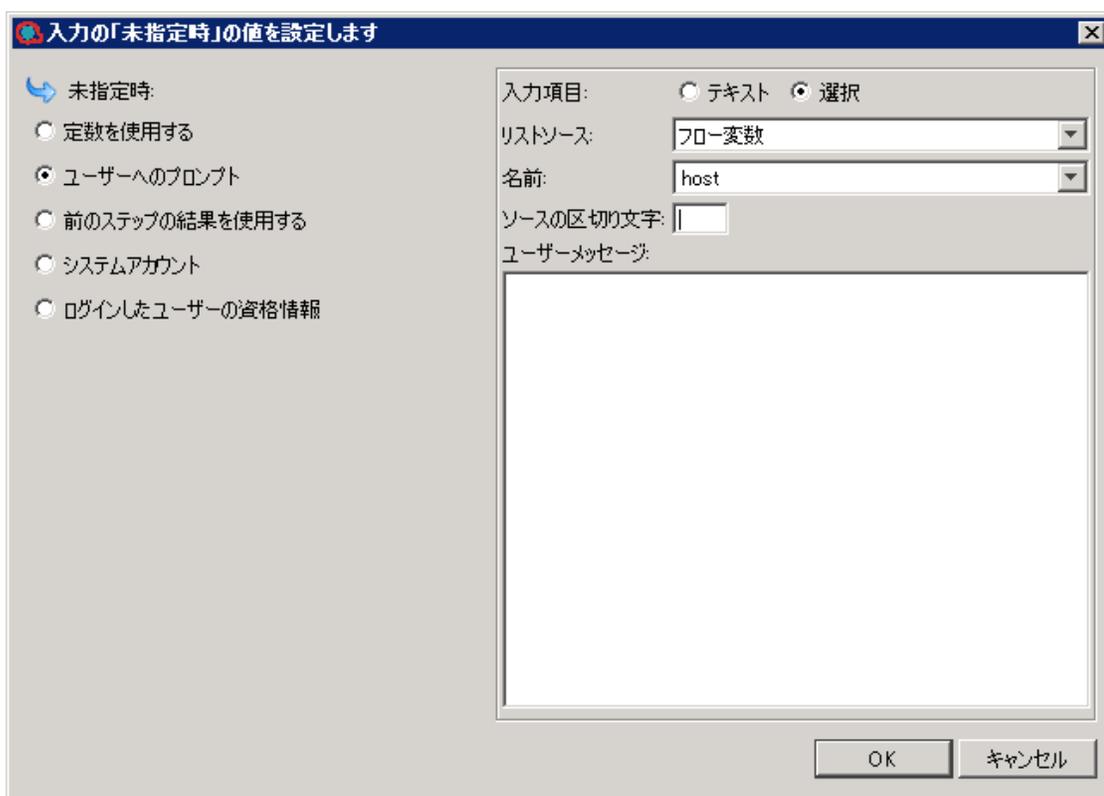
このセクションのタスクは、入力を設定するために行うタスクにおける、オプションのサブタスクです。「[入力の作成](#)」(205ページ)を参照してください。

実行する操作

単一の固定値を入力ソースに指定

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

1. 入力を作成して、種類を[**単一値**]に設定します。
2. [未指定時] ボタン  をクリックして、その行の[未指定時] ダイアログボックスを開くか、右側のペインに値を直接入力します。

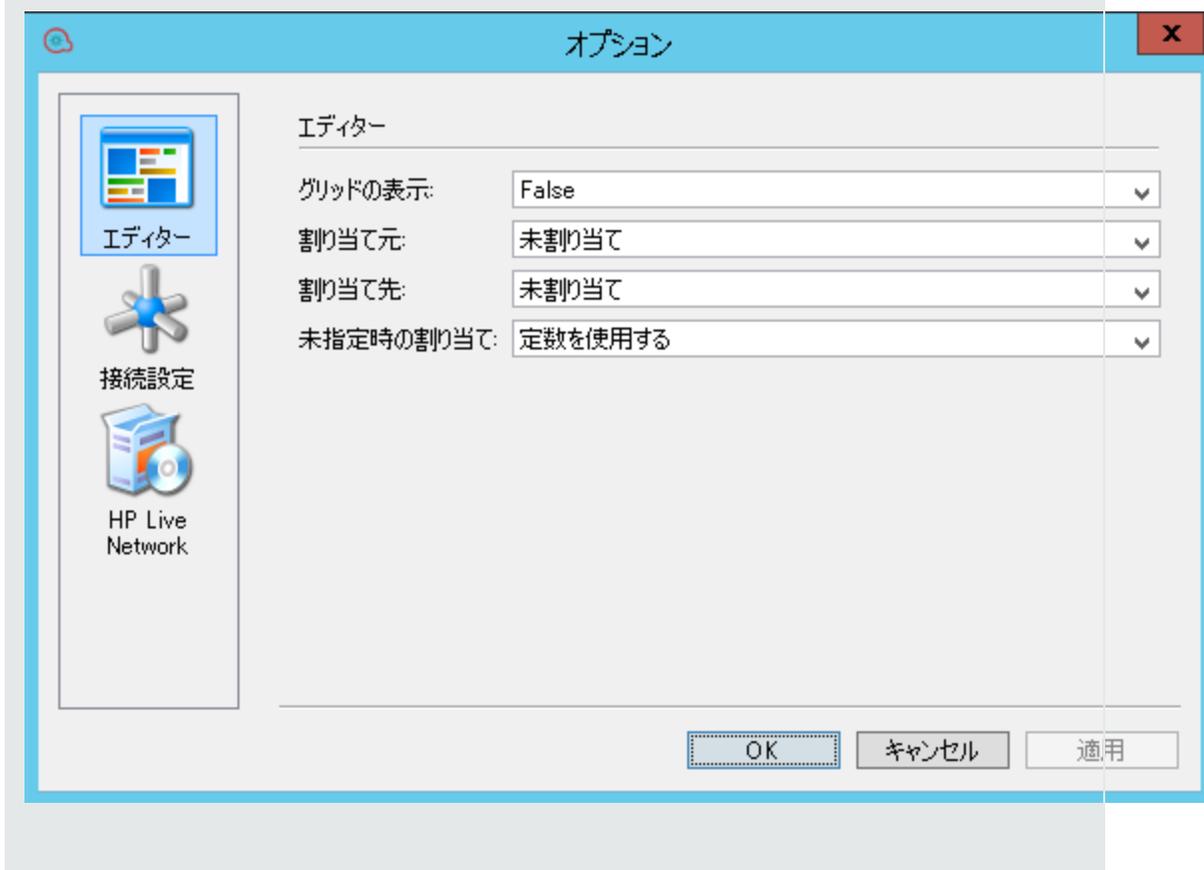


3. デフォルトでは、[割り当て元] フィールドに <未割り当て> が表示されます。
 - 同じ名前の変数の値から入力値を割り当てるには、[割り当て元] ドロップダウンメニューから <デフォルト> - <名前> を選択します。
 - 異なる入力の値から入力値を割り当てるには、[割り当て元] ドロップダウンメニューから入

力名を選択します。

- このフィールドに値を直接入力することもできます。
4. **[未指定時]** リストボックスで**[定数を使用する]**を選択します。これにより、**[割り当て元変数]** ボックスで指定したフロー変数が存在しない場合またはその変数に値が保持されていない場合に実行する操作を定義します。Studio 10.50 をインストールしている場合のデフォルト値は**[定数を使用する]**であり、以前のバージョンの Studio からアップグレードしている場合のデフォルト値は**[ユーザーへのプロンプト]**です。

注: エディターウィンドウの**[構成]**>**[オプション]**でこれらのデフォルトを変更できます。



5. [「未指定時:定数を使用する」構成]セクションで、入力の値 (false など) を入力します。また、テキストと変数の参照を組み合わせることもできます。`${variablename}` の形式で指定します。たとえば、`Ping of ${targethost} succeeded` のようになります。フォルダーに格納されている変数を参照するには、次の形式を使用します。
`${foldername/variablename}`

単一のユーザー入力テキストを入力ソースに指定

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

注: フローの途中のステップで入力ソースをユーザー入力テキストとして指定しないでください。

1. 入力を作成し、[種類] ドロップダウンメニューから [単一値] を選択します。
2. その行の [未指定時] ボタン  をクリックして、[未指定時] ダイアログボックスを開き、[ユーザーへのプロンプト] を選択するか、右側のペインで [ユーザーへのプロンプト] を選択します。Studio 10.50 をインストールしている場合のデフォルト値は [定数を使用する] であり、以前のバージョンの Studio からアップグレードしている場合のデフォルト値は [ユーザーへのプロンプト] です。

注: `Studio.properties` ファイルで `use.constant.default.value.input` パラメーターを設定することによって、このデフォルト値を変更できます。

3. [入力項目:] 領域で、[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、ユーザーに入力してもらいたいデータの内容を知らせるプロンプトメッセージを入力します。

単一のユーザー選択を入力ソースに指定

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

1. 入力を作成し、[種類] ドロップダウンメニューから [単一値] を選択します。
2. その行の [未指定時] ボタン  をクリックして、[未指定時] ダイアログボックスを開き、[ユーザーへのプロンプト] を選択するか、右側のペインで [ユーザーへのプロンプト] を選択します。Studio 10.50 をインストールしている場合のデフォルト値は [定数を使用する] であり、以前のバージョンの Studio からアップグレードしている場合のデフォルト値は [ユーザーへのプロンプト] です。

注: `Studio.properties` ファイルで `use.constant.default.value.input` パラメーターを設定することによって、このデフォルト値を変更できます。

3. [入力項目:] 領域で、[選択] を選択します。
4. [リストソース] ドロップダウンメニューから、次のいずれかを選択します。
 - [選択リスト]- あらかじめ定義されているリストのセットから選択する。
[名前] リストで、ユーザーに提示するリストを選択します。

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

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

うなユーザープロンプトを作成することが可能です。

[名前] リストで、ユーザーに提示するドメインタームリストを選択します。

- [フロー変数]- フロー変数の内容からリストを作成します。

[名前] リストで、リストに含むフロー変数を入力または選択します。異なるフローからフロー変数を参照できます。たとえば、`folderA\folderB\flow1` というフォルダー構造の下にあるフローを参照するには、`${folderA/folder/flow1}` という文字列を使用します。

[ソース区切り記号] ボックスに、リスト内の各要素を区切る記号を入力します。

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

単一の入力値のリストを入力ソースに指定

このような入力により、1つのステップを複数のターゲットで実行できるようになります。たとえば、複数のマシンを対象にしたオペレーティングシステムのヘルスチェックやソフトウェアアップデートのインストールを実行することができます。

1. 入力を作成し、[種類] ドロップダウンメニューから [値のリスト] を選択します。
2. その行の [未指定時] ボタン  をクリックして、[未指定時] ダイアログボックスを開き、[定数を使用する] を選択するか、右側のペインで [定数を使用する] を選択します。Studio 10.50 をインストールしている場合のデフォルト値は [定数を使用する] であり、以前のバージョンの Studio からアップグレードしている場合のデフォルト値は [ユーザーへのプロンプト] です。

注: Studio.properties ファイルで `use.constant.default.value.input` パラメーターを設定することによって、このデフォルト値を変更できます。

3. [入力区切り記号] ボックスに、値のリスト内の各要素を区切る記号を入力します。
4. [定数値] ボックスで、次のいずれかを実行します。
 - 入力の値を、[入力区切り記号] ボックスで入力した記号で区切って入力します。
 - 1つ以上のフロー変数参照を次の形式で入力します。

```
${foldername/flowvariablename1}<区切り記号>${foldername/flowvariablename2}
```

注: 入力した値と変数を同じリストに含めることが可能です。たとえば、`${folder1/flowvariableA}|${folder1/flowvariableB}|10.2.0.200|18.35.100.7` のようにします。

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

ユーザーが入力したテキストから取得する入力値のリストを入力ソースに指定

この種類の入力では、ユーザーは、区切り記号で区切った値のリストを入力する必要があります。たとえば、フローの開始時に、フローでターゲットへの複数のホストアドレスをユーザーが入力するようになることができます。

1. 入力を作成して、種類を **[値のリスト]** に設定します。
2. その行の **[未指定時]** ボタン  をクリックして、**[未指定時]** ダイアログボックスを開き、**[定数を使用する]** を選択するか、右側のペインで **[定数を使用する]** を選択します。Studio 10.50 をインストールしている場合のデフォルト値は **[定数を使用する]** であり、以前のバージョンの Studio からアップグレードしている場合のデフォルト値は **[ユーザーへのプロンプト]** です。
3. **[入力区切り記号]** ボックスに、リスト内の各要素を区切る記号または記号列を入力します。
4. **[未指定時]** リストボックスで **[ユーザーへのプロンプト]** を選択します。
5. **[入力項目:]** 領域で、**[テキスト]** を選択します。
6. **[ユーザーメッセージ]** ボックスに、フローユーザーに対してオペレーションが必要とするデータの種類を説明するプロンプトのテキストを入力します。

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

ユーザーの選択から取得する入力値のリストを入力ソースに指定

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

1. 入力を作成して、種類を **[値のリスト]** に設定します。
 2. その行の **[未指定時]** ボタン  をクリックして、**[未指定時]** ダイアログボックスを開き、**[ユーザーへのプロンプト]** を選択するか、右側のペインで **[ユーザーへのプロンプト]** を選択します。Studio 10.50 をインストールしている場合のデフォルト値は **[定数を使用する]** であり、以前のバージョンの Studio からアップグレードしている場合のデフォルト値は **[ユーザーへのプロンプト]** です。
 3. **[入力区切り記号]** ボックスに、リスト内の各要素を区切る記号を入力します。
 4. **[入力項目:]** 領域で、**[選択]** を選択します。
 5. **[リストソース]** リストで、次のいずれかを選択します。
 - **[選択リスト]** - あらかじめ定義されているリストのセットから選択する。
- [名前]** リストで、ユーザーに提示するリストを選択します。

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

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

[名前] リストで、ユーザーに提示するドメインタームリストを選択します。

- **[フロー変数]**- フロー変数の内容からリストを作成します。

[名前] リストで、リストに含むフロー変数を入力または選択します。

[ソース区切り記号] ボックスに、リスト内の各要素を区切る記号を入力します。

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

前のステップの結果を入力ソースに指定

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

1. 入力を作成し、入力で複数の値を許可するかどうかによって種類を **[単一値]** または **[値のリスト]** のいずれかに設定します。
2. その行の **[未指定時]** ボタン  をクリックして、**[未指定時]** ダイアログボックスを開き、**[前のステップの結果を使用する]** を選択するか、右側のペインで **[前のステップの結果を使用する]** を直接選択します。
3. 入力値が複数值の場合は、**[入力区切り記号]** ボックスに、リスト内の各要素を区切る記号を入力します。

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

システムアカウントの詳細としての入力ソースの指定

単一値として入力の種類を定義する場合は、ユーザーのログイン資格情報を入力ソースとして使用できます。これにより、システムアカウントの資格情報を必要とするタスクをフローで実行できるようになります。

フローを開始したユーザーアカウントで定義されているいずれかの資格情報 (ユーザー名またはパスワード) を入力値のソースとして指定します。

1. その行の **[未指定時]** ボタン  をクリックして、**[未指定時]** ダイアログボックスを開き、**[システムアカウント]** を選択するか、右側のペインで **[システムアカウント]** を直接選択します。
2. **[アカウント名:]** ドロップダウンメニューからオペレーションの資格情報で使用するシステムアカウントを選択し、システムアカウントから抽出するプロパティ (パスワードやユーザー名など) を選択します。これにより、こういったアカウントの資格情報を必要とするタスクをフローで実行できるだけでなく、資格情報をシステムアカウント名で隠しておけるので資格情報の漏洩を防ぐことにもなります。システムアカウントの詳細については、[「システムアカウントの設](#)

[定](#) (165ページ)を参照してください。

3. [資格情報の種類] フィールドで、[ユーザー名]または[パスワード]を選択します。

ユーザーのログイン名またはパスワードを入力ソースに指定します。

資格情報を入力ソースとして使用することもできます。これにより、システムアカウントの資格情報を必要とするタスクをフローで実行できるようになります。

フローを開始したユーザーアカウントで定義されているいずれかの資格情報(ユーザー名またはパスワード)を入力値のソースとして指定します。

注: この入力方法は、Studio 9.x との互換性のみを目的としています。Studio 10.x 用に新しい入力を定義する場合は、「[システムアカウントの詳細としての入力ソースの指定](#)」(220ページ)で説明されている[システムアカウント]オプションを使用してください。

1. その行の[未指定時] ボタン  をクリックして、[未指定時] ダイアログボックスを開きます。
2. [ログインしたユーザーの資格情報] を選択するか、右側のペインで[ログインしたユーザーの資格情報] を直接選択します。
3. [資格情報の種類] ドロップダウンメニューで、[ユーザー名]または[パスワード]を選択します。

システムアカウントの詳細としての入力ソースの指定

単一値として入力の種類を定義する場合は、ユーザーのログイン資格情報を入力ソースとして使用できます。これにより、システムアカウントの資格情報を必要とするタスクをフローで実行できるようになります。

フローを開始したユーザーアカウントで定義されているいずれかの資格情報(ユーザー名またはパスワード)を入力値のソースとして指定します。

1. その行の[未指定時] ボタン  をクリックして、[未指定時] ダイアログボックスを開き、[前のステップの結果を使用する] を選択するか、右側のペインで[前のステップの結果を使用する] を直接選択します。
2. [アカウント名:] ドロップダウンメニューからオペレーションの資格情報で使用するシステムアカウントを選択し、システムアカウントから抽出するプロパティ(パスワードやユーザー名など)を選択します。これにより、こういったアカウントの資格情報を必要とするタスクをフローで実行できるだけでなく、資格情報をシステムアカウント名で隠しておけるので資格情報の漏洩を防ぐことにもなります。システムアカウントの詳細については、「[システムアカウントの設定](#)」(165ページ)を参照してください。
3. [資格情報の種類] フィールドで、[ユーザー名]または[パスワード]を選択します。

参考資料

入力のインスペクター > ツールバー



GUI アイテム	説明
入力の追加 	新しい入力行を追加します。新しい入力の名前を入力して、[OK] をクリックします。デフォルト値を使用して新しい入力の属性が割り当てられます。
行の削除 	選択した入力行を削除します。
上に移動 	リスト内の選択した入力行を上に移動します。
下に移動 	リスト内の選択した入力行を下に移動します。
切り取り  (またはキーボードの Ctrl + X)	選択した入力行を現在の場所から削除します。その行を新しい場所に置くには、[貼り付け] オプションを使用します。 注: 必須の入力を削除しようとした場合、Studio によって自動的にオプションとしてマークされ、イタリック体で表示されます。再び [必須] としてマークすれば入力を復元できます。
コピー  (またはキーボードの Ctrl + C)	選択した入力行をコピーします。コピーした行を新しい場所に置くには、[貼り付け] オプションを使用します。
貼り付け  (またはキーボードの Ctrl + V)	コピーするか切り取った入力行を現在の場所に貼り付けます。入力名が既に存在している場合は、同じ名前に (Copy 1) を付け加えた名前の新しい入力が増加されます。たとえば、host という名前の入力をコピーするか貼り付けた場合、次の 2 つの入力が存在します。host と host (Copy 1) です。

注: ステップ、フロー、またはオペレーションの間で複数の入力の切り取り、コピー、貼り付けを行うこともできます。Ctrl キーを押しながら追加の入力を選択し、その行をクリックします。

入力のインスペクター > [入力] タブ

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

入力	機密データ	必須	種類	割り当て元	未指定時	割り当て先
xml	<input type="checkbox"/>	<input checked="" type="checkbox"/>		xml	ユーザーへのプロンプト (テキストから)	xml
xpathQuery	<input type="checkbox"/>	<input checked="" type="checkbox"/>		xpathQuery	ユーザーへのプロンプト (テキストから)	xpathQuery
delimiter	<input type="checkbox"/>	<input type="checkbox"/>		delimiter	ユーザーへのプロンプト (テキストから)	delimiter

GUI アイテム	説明
入力	入力名を表示します。
機密データ	<p>Central ユーザーインターフェイス、および Studio の内部デバッガーとリモートデバッガーでこの入力を非表示にします。このチェックボックスがオンになっている場合、Central と Studio で入力値が暗号化されてアスタリスクとして表示され、保持されません。</p> <p>このフラグは一時的であり、機密変数から割り当てられると、割り当てられた変数も機密になります。一時的な効果が見られるのはランタイム実行中にのみであり、Studio ユーザーインターフェイスには反映されません。</p> <p>スクリプトレットで使用すると、機密データが暗号化された状態で取得されません。</p> <p>割り当てのための式を使用すると、機密性が保持されます。たとえば、<code>\${input1}</code> のようになります。</p>
必須	この入力を必須にします。
種類 	<p>[種類] リストで、入力の値を取得する方法を指定します。次のオプションを選択できます。</p> <ul style="list-style-type: none"> • 単一値  • 値のリスト  - 複数のターゲットに対してオペレーションを実行できます。
割り当て元	<p>入力の値の取得元を指定します。[割り当て元] リストから値の取得元の入力を選択するか、このフィールドに値を直接入力します。</p> <p>デフォルトでは、このフィールドに <未割り当て> が表示されます。</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>注: 次のような一部の状況では、フローレベルの入力は Central に表示されません。</p> <ul style="list-style-type: none"> • [割り当て元] に値を割り当てず、[未指定時] を [定数を使用する] に設定している場合 • [割り当て元] の値がフローの入力名と異なる場合 </div>
未指定時	<p>[割り当て元] ボックスで指定したフロー変数が存在しない場合またはその変数に値が格納されていない場合に実行する必要がある操作が表示されます。</p> <p> をクリックして、[未指定時] ダイアログボックスを開くか、右側のペインのフィールドに値を直接入力します。</p>

割り当て先	入力の値を割り当てるフロー変数を選択するか、このフィールドに値を直接入力します。 デフォルトでは、このフィールドに <未割り当て> が表示されます。
--------------	---

ステップのインスペクター > [入力] タブ > [未指定時] ペイン/[未指定時] ダイアログボックス

入力のインスペクターの右側に [未指定時] ペインが表示されます。このペインで、[割り当て元] 列で指定したフロー変数が存在しない場合またはその変数に値が格納されていない場合に実行する必要がある操作を定義します。これらのフィールドの一部は、[未指定時] ダイアログボックスにも表示されます。[未指定時] ペインで変更を行った場合、それに合わせて [未指定時] ダイアログボックスが自動的に更新されます。逆の場合も同じです

[未指定時] ペイン

名前: input1 入力の種類: 単一値

割り当て元変数: input1

未指定時:

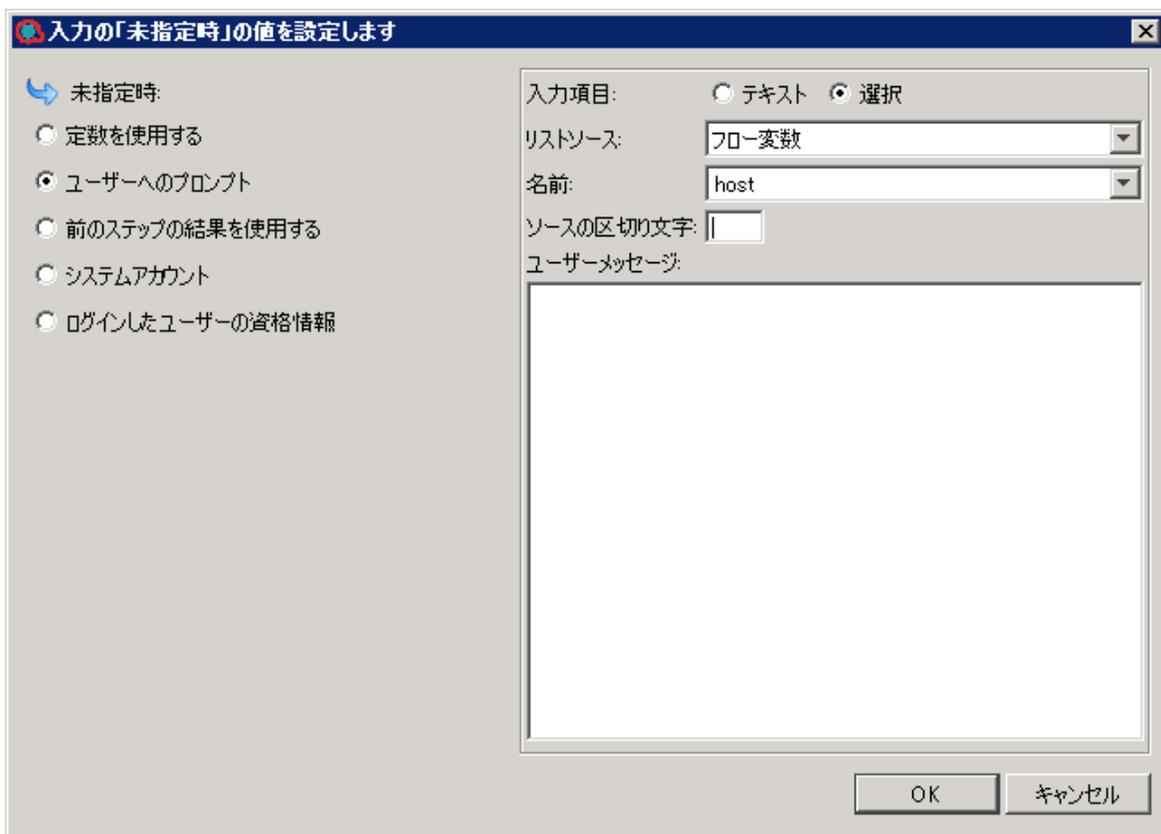
- 定数を使用する
- ユーザーへのプロンプト
- 前のステップの結果を使用する
- システムアカウント
- ログインしたユーザーの資格情報

定数値:

割り当て先変数: input1

機密データ 必須 検査形式: <未検証>

[未指定時] ダイアログボックス



GUI アイテム	説明
入力列	入力名を表示します。
必須列	この入力を必須にします。
種類列 	<p>[種類] リストで、入力の値を取得する方法を指定します。次のオプションを選択できます。</p> <ul style="list-style-type: none"> • 単一値  • 値のリスト  - 複数のターゲットに対してオペレーションを実行できます。

<p>[割り当て元] 列</p>	<p>入力の値の取得元を指定します。[割り当て元] リストから値の取得元の入力を選択します。</p> <p>注: [割り当て元] に値を割り当てず、[未指定時] が [定数を使用する] に設定されている場合、フローレベルの入力は Central に表示されません。</p> <p>[割り当て元] に値を割り当て、[未指定時] が [定数を使用する] に設定されている場合、フローレベルの入力は Central に表示されます。</p>
<p>[未指定時] 列</p>	<p>[割り当て元] ボックスで指定したフロー変数が存在しない場合またはその変数に値が格納されていない場合に実行する必要がある操作が表示されます。</p> <p> をクリックして、[未指定時] ダイアログボックスを開くか、右側のペインのフィールドに値を直接入力します。</p>
<p>[割り当て先] 列</p>	<p>入力の値を割り当てるフロー変数を選択します。</p>
<p>名前</p>	<p>ステップの名前を表示します (読み取り専用)。</p>
<p>入力の種類</p>	<p>入力の種類を表示します。ここで修正できます。</p>
<p>割り当て元変数</p>	<p>入力のソースとなるフロー変数の名前を入力または選択します。</p>
<p>未指定時</p>	<p>[割り当て元変数] ボックスに指定したフロー変数が存在しない場合、またはこのフロー変数に値が格納されていない場合に発生するアクションを選択します。</p>
<p>「未指定時:<action>」 構成</p>	<p>[割り当て元変数] ボックスに指定したフロー変数が存在しない場合、またはこのフロー変数に値が格納されていない場合に発生するアクションの詳細を構成します。このセクションは、[未指定時] リストで選択するアクションによって異なります。</p>
<p>割り当て先変数</p>	<p>入力の値を割り当てるフロー変数を選択します。</p>
<p>入力区切り記号</p>	<p>値を区切るために使用する区切り文字を入力します。</p>
<p>必須</p>	<p>この入力を必須にします。</p>
<p>検査形式</p>	<p>入力の値をシステム評価子で検証します。ドロップダウンリストからシステム評価子を選択します</p> <p>たとえば、IP アドレスを入力するようにユーザーに求める場合は、IP 検査形式を使用します。アドレスが入力されると、Studio によって有効な IP アドレス形式かどうかを確認されます。</p>

次の名前 で記録	診断または監査に値を使用できるようにします。このフィールドは、HP 00 9.x からのレガシーフィールドです。 データベースに入力を記録する際に使用するアイテムを [ドメインターム] リストから選択できます
-----------------	---

入力データの評価

入力の検証には、評価子を使用します。例:

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

Studio では、次の値を検証する標準のシステム評価子を用意しています。

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

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

評価子では次が使用されます。

- =、!=、Begins with、Contains、Match All Words、Match At Least One Word などの簡単な演算子。
- 正規表現 (詳細については下記を参照) [「フローでの正規表現の使用」 \(302ページ\)](#)
- スクリプトレット (詳細については、下記を参照) [「フロー内でのスクリプトレットの使用」 \(297ページ\)](#)

実行する操作

評価子を使用した入力の検証

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

きます。

名前: EventId 入力の種類: 単一値

割り当て元変数: <未割り当て>

未指定時:

- 定数を使用する
- ユーザーへのプロンプト
- 前のステップの結果を使用する
- システムアカウント
- ログインしたユーザーの資格情報

定数値:

割り当て先変数: EventId

機密データ 必須 **検査形式** alphanumeric (oo10-base-cp-1.6.2)

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

入力の作成の詳細については、[「入力の作成」\(205ページ\)](#)を参照してください。

トランジションの作成

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



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

オペレーションのレスポンスの設定の詳細については、「[レスポンスの設定](#)」(233ページ)を参照してください。

注: HP 00 10.00 以降では、トランジションの説明が、1000 バイトに制限されます。

ベストプラクティス

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

実行する操作

ステップ間にトランジションを追加する

1. Studio の作成ペインでフローを開きます。
2. 次のステップに接続する必要があるステップで、レスポンス名をクリックするか、いずれかのレスポンスを表すアイコンをクリックし、そのレスポンスのターゲットステップにドラッグします。
3. トランジションをダブルクリックします。トランジションのインスペクターが開きます。
4. (オプション) トランジションの名前を変更するには、[トランジション名] ボックスに新しい名前を入力します。

5. **[説明]** ボックスに、このトランジションが発生する原因となった先行ステップで起こったことについての説明を入力します。この説明は、フローの実行時に HP 00 Central の**結果の概要**エリアに表示されます。

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

フロー変数を含む説明を作成する

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

1. 2つのステップ間にトランジションを作成します。
2. トランジションをダブルクリックしてトランジションのインスペクターを開きます。
3. **[説明]** ボックスに、ステップのオペレーションまたは実行されたフローの別の場所から取得されたデータを格納しているフロー変数を含めて、説明を入力します。参照の形式は、`${フロー変数名}` を使用する必要があります。

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

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

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

ゲート制御式トランジションは赤で表示されます。

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

トランジション後に実行がハンドオフされるようにする

フローを別のユーザーにハンドオフするように、トランジションを設定できます。この処理が必要と

なるのは、次のステップで別のユーザーからの情報が必要である場合です。

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

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

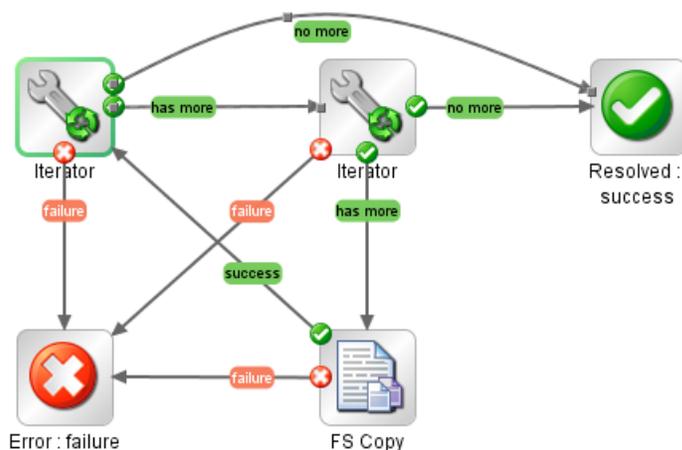
トランジションの完了をフローの ROI 値でカウントする

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

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

曲線定義点を追加して曲線のトランジションを作成する

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



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

曲線定義点を削除する

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

トランジション名を移動する

トランジション名をクリックして別の場所にドラッグします。

2つのステップ間のトランジションを削除する

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

参考資料

トランジションのインスペクター

トランジションのインスペクターで、トランジションの詳細を指定します。

インスペクター

トランジション名: success

ゲート制御式トランジション

続行前にユーザーのグループを確認

必須のグループ: AUDITOR

このトランジション後にフロー実行をハンドオフ

トランジションの ROI 値: 0.0

説明

GUI アイテム	説明
トランジション名	デフォルトでは、トランジション名は起点となるレスポンスの名前 (success、failure など) と同じになりますが、トランジション名は変更することができます。
続行前にユーザーのグループを確認	ゲート制御式トランジションを作成するには、このチェックボックスをオンにします。必須の役割エイリアスが割り当てられている場合のみ、ユーザーは次のステップに進むことができます。
必須の役割エイリアス	フローの実行を続けるためにユーザーに割り当てる必要がある、役割エイリアスを選択します。

このトランジション後にフロー実行をハンドオフ	ハンドオフトランジションは、現在のバージョンではサポートされていません。 トランジション後に別のユーザーにフローをハンドオフするには、このチェックボックスをオンにします。
トランジションの ROI 値	トランジションの値を入力します。これにより、フローの実行中にそのトランジションが発生した場合に、このフローの実行の値にトランジションの値が加算されます。
説明	このトランジションが発生する原因となった先行ステップで起こったことの説明を入力します。この説明は、HP OO Central の結果の概要エリアに表示されます。

レスポンスの設定

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



レスポンスの種類には次の 4 つがあります。

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

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

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

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

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

オペレーションまたはフローのレスポンスを追加、削除、または編集できます。リターンステップを除き、ステップのレスポンスを編集することはできません。リターンステップの詳細については、「[リターンステップの作成](#)」(278ページ)を参照してください。

レスポンスルール

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

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

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

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

実行する操作

オペレーションにレスポンスを追加する

1. オペレーションの [**レスポンス**] タブをクリックします。



2. 新しいレスポンスを追加する行を選択します。たとえば、最初の行を選択した場合、新しいレスポンスは 2 番目の行に表示されます。
3. [**レスポンスの追加**] をクリックし、新しいレスポンスの名前を入力します。
4. オペレーションの実行が失敗した場合に選択されるレスポンスを指定するには、[**失敗時**] 列でそのレスポンスのチェックボックスをオンにします。
5. レスポンスをデフォルトのレスポンスとして指定するには、[**デフォルト**] 列にあるチェックボックスをオンにします。レスポンスのルールで該当するものがない場合、デフォルトのレスポンスが、選択されるレスポンスになります。

6. [種類]列で、レスポンスの種類を選択します。

- 解決済み: 
- 診断済み: 
- 操作なし: 
- 失敗: 

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

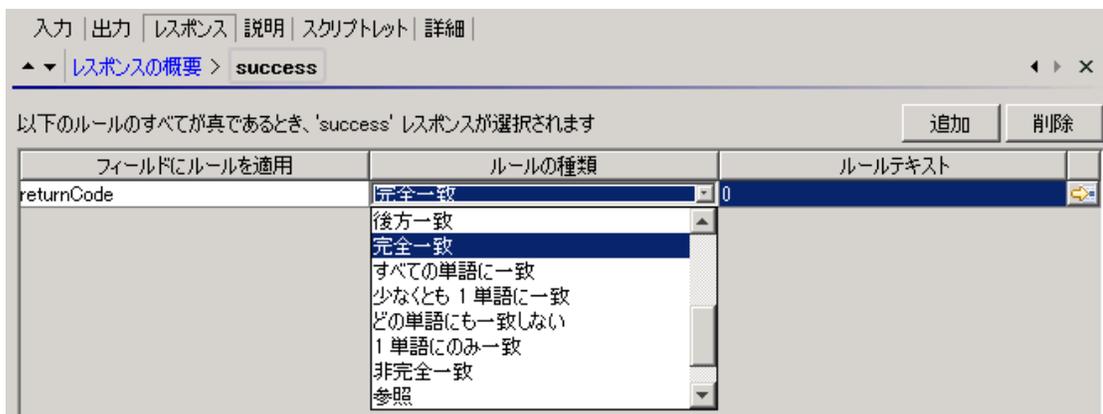
7. レスポンスのルールを作成するには、レスポンスの行の右端にある右向き矢印  をクリックします。詳細については、以下の「レスポンスのルールを作成する」を参照してください。

レスポンスのルールを作成する

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

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

新しいルールが表示されます。

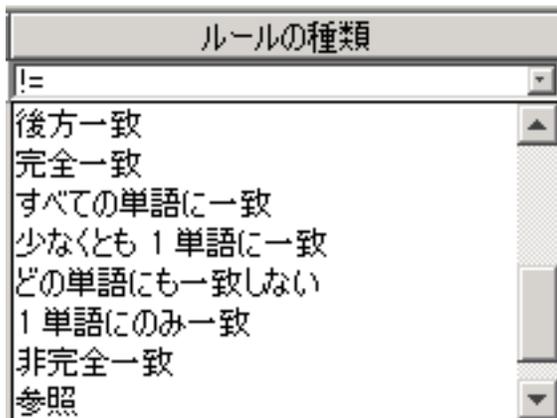


3. [フィールドにルールを適用]列で、ルールをテストする値を持つ結果フィールドを選択します。テストできる結果フィールドには、結果の終了コード、出力文字列、エラー文字列、失敗メッセージ、およびタイムアウトになった true または false が含まれます。

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

これらのフィールドの値を確認するには、Studio Debugger を使ってフローのオペレーションをテストします。Debugger でフローを実行すると、ステップ結果インスペクターにステップの結果が表示されます。Debugger の詳細については、「[フローのテストとデバッグ](#)」(313ページ)を参照してください。

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



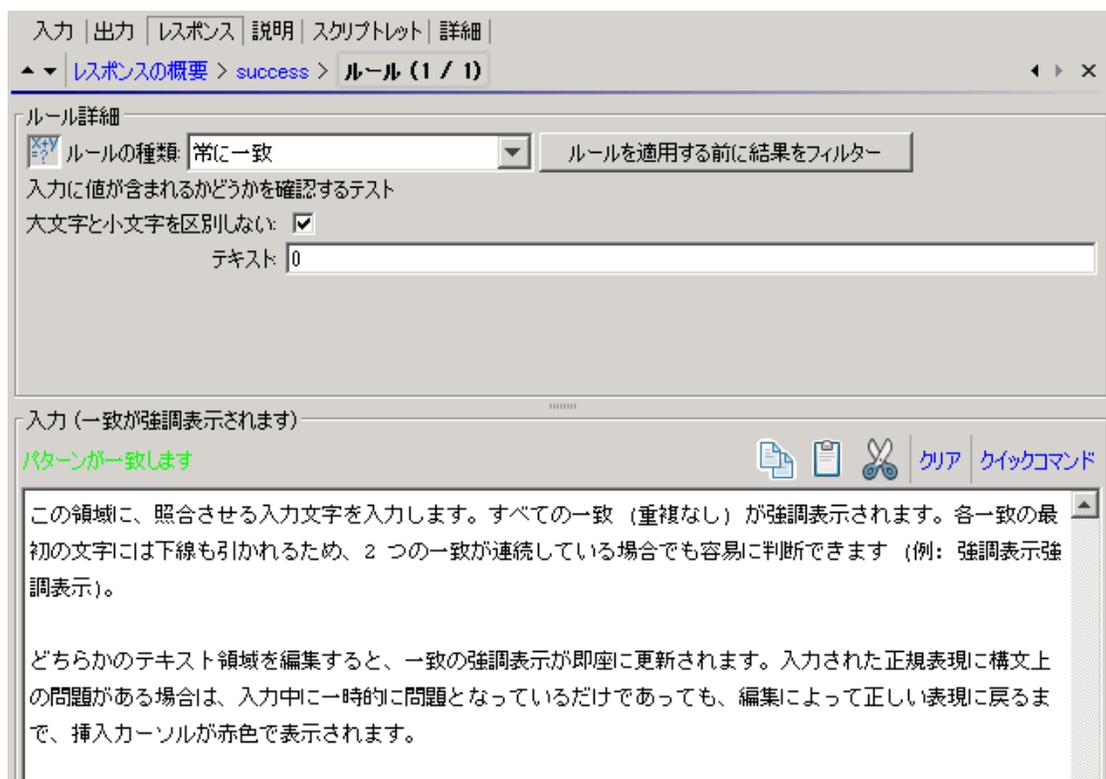
- [=], [!=], [前方一致]、[部分一致]、[すべての単語に一致]、[少なくとも 1 単語に一致]などの単純な演算子を選択します。
- 正規表現を作成するには、[正規表現]を選択します。
- スクリプトレットを作成するには、[スクリプトレット]を選択します。
- 共有ルールへの参照を作成するには、[参照]を選択します。

5. [ルールテキスト]列に、テストで使用するテキストを入力します。

レスポンスルールをフィルタリングおよびテストする

ルール詳細エディターで、次の操作が可能です。

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



注: ルールの種類として[スクリプトレット]を選択している場合、ルール詳細エディターにスクリプトレットエディターが含まれます。スクリプトレットの作成の詳細については、「[フロー内でのスクリプトレットの使用](#)」(297ページ)を参照してください。

ルールの種類として[正規表現]を選択している場合、ルール詳細エディターに正規表現エディターが含まれます。正規表現の作成の詳細については、「[フローでの正規表現の使用](#)」(302ページ)を参照してください。

- 別のルールの種類を使用するには、[ルールの種類]リストでほかの種類を選択します。
- ルールを適用する前に結果をフィルタリングするには、[ルールを適用する前に結果をフィルター]をクリックし、フィルターエディターでフィルターを作成します。
レスポンスルールのフィルターを作成するのは、出力または結果のフィルターを作成するのと同じです。「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。
- ほとんどのルールの種類については、[テキスト]ボックスに、比較をテストするテキストを入力します。大文字小文字を区別しない場合は、[大文字と小文字を区別しない]チェックボックスをオンにします。

正規表現ルールでは、オペレーション結果の正規表現フィルターを作成するのと同じように、正規表現とその用途を指定します。詳細については、「[フローでの正規表現の使用](#)」(302ページ)を参照してください。

結果ボックスにテストの結果が表示されます。[パターンが一致します]または[パターンが一致しません]が表示されて、一致するテキストが強調表示されます。

5. オペレーションのレスポンスの別のルールを使用するには、**[レスポンスの概要]**の横にある上向き矢印または下向き矢印をクリックします。

注: ルールでは、数字で始まる文字列の評価に比較演算子 (=、 !=、 <、 > など) を使用すると、比較されるのは文字列の数字部分のみです。たとえば、 != (等しくない) を使用して "123" と "123Test" を比較する場合、"123" は明らかに "123Test" と同じではありませんので、評価は "false" となります。ただし、**非完全一致**評価子で文字列を比較すると、この問題に対処できます。

フローにレスポンスを追加する

フローにレスポンスを作成したら、作成したレスポンスをフローのリターンステップで使用できるようにします。

たとえば、**エラーリターンステップ**をもたらした結果が、オペレーションの失敗ではなく、要求されるしきい値に到達しない結果だった場合、この結果を反映して「**エラー: しきい値に到達していない**」と表示される新しいレスポンスを**エラーリターンステップ**に作成できます。

1. フローの**[プロパティ]**シートを開きます。
2. **[レスポンス]**タブをクリックします。
3. **[レスポンスの追加]**をクリックし、表示されたテキストボックスに、レスポンスの名前を入力します。たとえば、「しきい値に到達していない」と入力します。
4. **[OK]**をクリックします。

フローに**エラーリターンステップ**を作成するとき、レスポンスとして「**しきい値に到達していない**」を選択できます。リターンステップの詳細については、「[リターンステップの作成](#)」(278ページ)を参照してください。

オペレーションまたはフローからレスポンスを削除する

1. オペレーションまたはフローの**[プロパティ]**シートを開きます。
2. **[レスポンス]**タブをクリックします。
3. レスポンスを選択し、**[レスポンスの削除]**をクリックします。

参考資料

[フロープロパティ]シート > [レスポンス]タブ

フローの**[プロパティ]**シートの**[レスポンス]**タブで、フローのリターンステップに使用できるようにする、発生する可能性があるレスポンスを指定します。たとえば、「**エラー: しきい値に到達していない**」と入力します。



GUI アイテム	説明
レスポンスの追加	新しいレスポンス行を追加します。
レスポンスの削除	選択したレスポンス行を削除します。
	クリックして、選択したレスポンスをリスト内で上または下に移動します。

オペレーションの [プロパティ] シート > [レスポンス] タブ

オペレーションの [プロパティ] シートの [レスポンス] タブで、オペレーションで発生する可能性があるレスポンスを指定します。



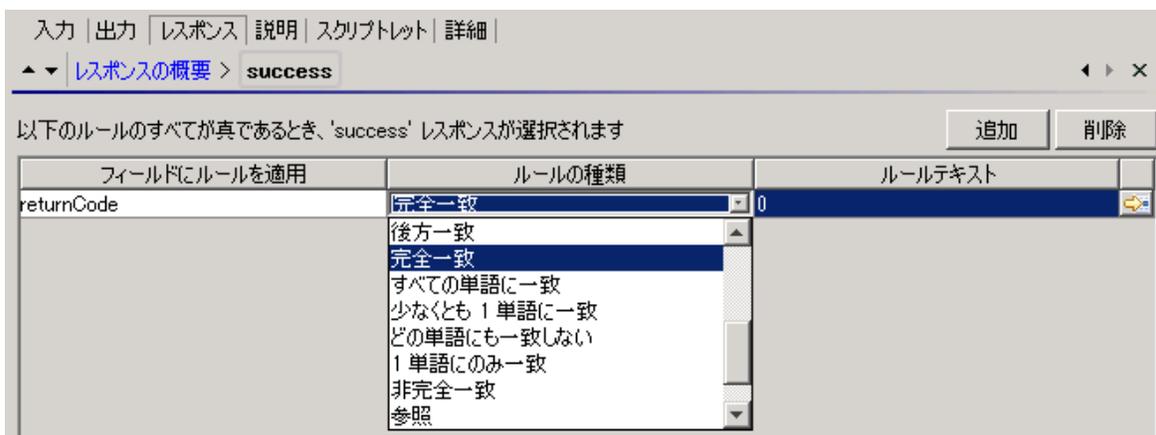
GUI アイテム	説明
レスポンスの追加	新しいレスポンス行を追加します。
レスポンスの削除	選択したレスポンス行を削除します。
	クリックして、選択したレスポンスをリスト内で上または下に移動します。
デフォルト	オンにして、レスポンスをデフォルトのレスポンスとして指定します。レスポンスのルールで該当するものがない場合、デフォルトのレスポンスが、選択されるレスポンスになります。
失敗時	オンにして、オペレーションの実行が失敗した場合に選択されるレスポンスを指定します。

種類	レスポンスの種類を選択します。 <ul style="list-style-type: none"> 成功/解決済み:  診断済み:  操作なし:  失敗: 
ルール	レスポンスに作成したルールを表示します。
	クリックすると、ルールエディターが表示されて、レスポンスのルールを作成できるようになります。

オペレーションの [プロパティ] シート > [レスポンス] タブ > ルールエディター

ルールエディターで、結果の特定の条件が該当する場合のみ発生するように、レスポンスを制限します。

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



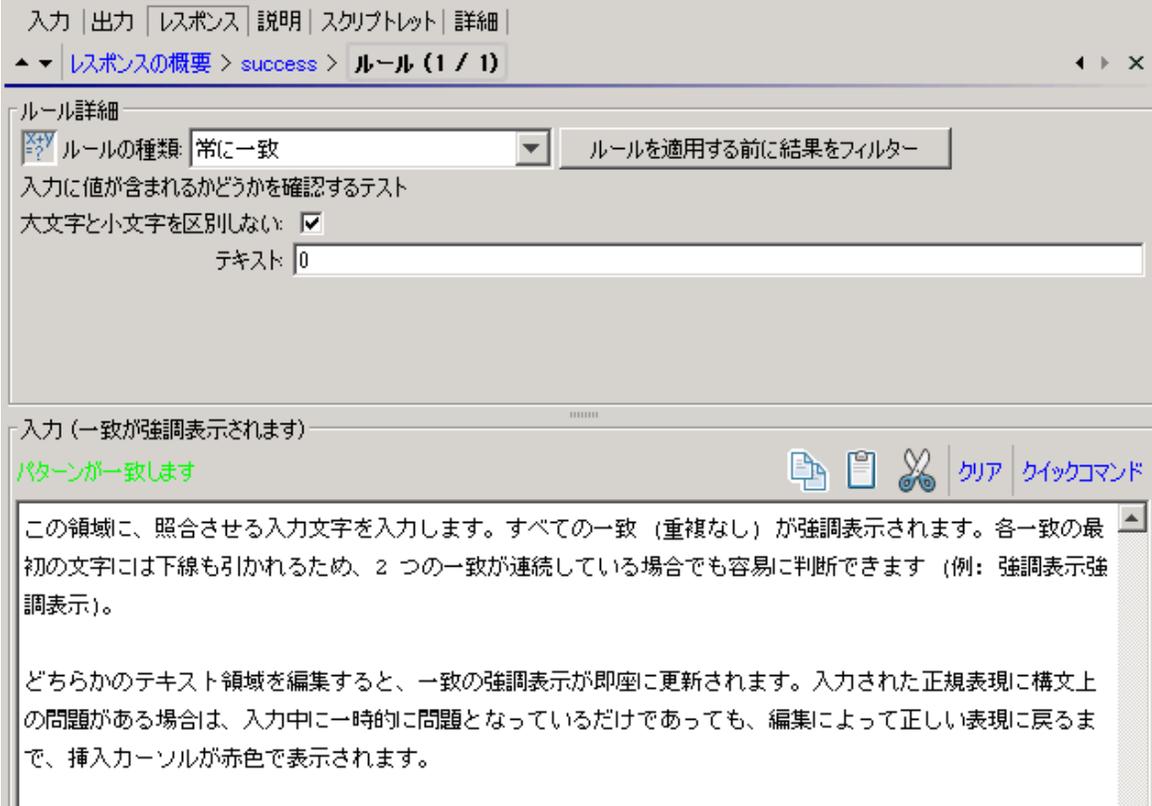
GUI アイテム	説明
レスポンスの追加	新しいレスポンス行を追加します。
レスポンスの削除	選択したレスポンス行を削除します。
	クリックして、選択したレスポンスをリスト内で上または下に移動します。
フィールドにルールを適用	ルールをテストする値を持つ結果フィールドを選択します。テストできる結果フィールドには、結果の終了コード、出力文字列、エラー文字列、失敗メッセージ、およびタイムアウトになった true または false が含まれます。

ルールの種類	フィールド値でテストを行う比較または一致を選択します。
ルールテキスト	テストに使用するテキストを入力します。
	クリックすると、ルール詳細エディターが開いて、ルールをテストおよびフィルタリングできるようになります。

オペレーションの [プロパティ] シート > [レスポンス] タブ > ルールエディター > ルール詳細エディター

ルール詳細エディターで、フィルターをテストしてルールに適用できます。

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



GUI アイテム	説明
ルールの種類	ルールエディターで選択した、ルールの種類を表示します。別のルールの種類を選択することもできます。
ルールを適用する前に結果をフィルター	クリックすると、フィルターエディターが表示されて、ルールを適用する前に結果をフィルタリングできるようになります。

テキスト	比較をテストするテキストを入力します。
大文字と小文字を区別しない	オンにすると、テキストが大文字であるのか小文字であるのか区別されなくなります。
結果ボックス	テストの結果を表示します。[パターンが一致します]または[パターンが一致しません]が表示されて、一致するテキストが強調表示されます。
コピー	結果ボックス内のデータをコピーします。
貼り付け	結果ボックスにデータを貼り付けます。
切り取り	結果ボックス内のデータを切り取ります。
クリア	結果ボックス内のデータをクリアします。
クイックコマンド	フィルターをテストするデータを生成するコマンドを入力します。コマンドの出力は結果ボックスに表示されます。

出力と結果の作成

フロー内で使用するデータを取得する方法の1つに、ステップ結果を使用する方法があります。このデータを割り当てる方法は2つあります。

- 結果内の出力を**フロー変数**に割り当てるときに、この出力をデータとしてフロー内のほかのステップに渡すことができます。
- 結果内の出力を**フロー出力フィールド**に割り当てるときに、この出力をデータとして親フローに渡すことができます。

このプロセスには、次のようなさまざまなステップがあります。

1. プライマリ出力を含め、オペレーションの出力を設定します。
「[オペレーションの出力の設定](#)」(243ページ)を参照してください。
2. フロー内のステップにオペレーションを使用するとき、どのオペレーション出力をステップ結果として使用するのか、つまり、どのオペレーション出力をフロー変数またはフロー出力フィールドに割り当てるとかを決定します。
「[ステップの結果の設定](#)」(246ページ)を参照してください。
3. (オプション) フィルターを作成して、出力または結果の選択の範囲をさらに絞り込むことができます。
「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。

オペレーションの出力の設定

フロー出力を設定するには、まず、オペレーションの出力を設定します。これを行うと、フロー作成者(およびほかのフロー作成者)がこのオペレーションをフロー内で使用するとき、出力をフロー変数に割り当てできるようになります。

エンドユーザーから隠す必要がある機密データが操作に含まれている場合、ステップレベルでこれを定義できます。「[ステップの結果の設定](#)」(246ページ)を参照してください。

オペレーション出力のタイプ

次のようなさまざまなタイプのオペレーション出力があります。

- **未加工結果**は、すべてのオペレーションのリターンコード、データ出力、およびエラー文字列です。
Studio では、オペレーションから作成されたステップの未加工結果を除き、未加工出力は直接表示されません。
- プライマリ出力とその他の出力は未加工出力の一部で、出力として指定したサクセスコード、出力文字列、エラー文字列、障害メッセージなどです。
 - **プライマリ出力**とは、ステップのプライマリ結果として使用される出力です。プライマリ出力は入力に値を提供し、その入力値の割り当てには[前のステップの結果]を使用します。[出力]タブの出力のリストの上にある[出力の概要]にプライマリ出力が表示されます。
 - オペレーションの**セカンダリ出力**とは、プライマリ出力に追加する別の出力です。

ヒント: 出力にフィルターを1つ以上作成すると、出力の選択の範囲をさらに絞り込むことができます。「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。

オペレーション出力の例

オペレーションには通常、そのオペレーションに固有の出力が含まれます。ただし、Libraryの **Accelerator Packs**、**Integrations**、**Operations** の各フォルダーでオペレーションを使用する場合、次の出力がよく見られます。

- **returnResult**
名前付きのフィールドがなく "returns:" が表示された場合、通常これがプライマリ出力です。また、プライマリ出力は、大文字 R (汎用) の **Result** でアクセス可能です。
- **response** (または **returnCode**)
オペレーションのレスポンスの決定に使用するコードまたは文字列。
- **failureMessage**
インフラストラクチャーの内部出力。オペレーションが失敗を返す場合、この出力は例外となります。多くのオペレーションはこの出力を使用しません。

実行する操作

オペレーションのプライマリ出力を指定する

オペレーションを設定する際に、プライマリ出力を指定できます。プライマリ出力を作成すると、そのソースを変更できますが、プライマリ出力なしの状態に戻すことはできません。

この操作はステップレベルでのみ実行できます。

1. [プロジェクト]ペインでオペレーションを右クリックし、[プロパティ]を選択します。

注: 複数のオペレーションを開くには、Shift キーまたは Ctrl キーを使用して選択し、右クリックして[開く]を選択します。

2. [出力]タブを選択します。
3. [フィールドからプライマリ出力を抽出] リストから、ソースフィールドを選択します。たとえば、[FailureMessage]を選択します。

ヒント: 各出力フィールドに表示されるデータの詳細については、オペレーションの[説明]タブをクリックします。

オペレーションにセカンダリ出力を追加する

オペレーションのセカンダリ出力とは、プライマリ出力に追加する別の出力です。

1. [プロジェクト]ペインでオペレーションを右クリックし、[プロパティ]を選択します。
2. [出力]タブを選択します。
3. [出力の追加]をクリックします。
4. 出力の名前を入力します。
5. [出力フィールド] リストから、出力データを取得するフィールドを選択します。
6. 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、該当する行の端にある右向き矢印  をクリックします。

フィルターの作成の詳細については、「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。

出力データを取得するフィールドを変更する

1. オペレーションの[プロパティ]シートを開いて、[出力]タブを選択します。
2. プライマリ出力のフィールドを変更するには、[フィールドからプライマリ出力を抽出] ボックスの右にある下向き矢印をクリックし、リストから目的のフィールドを選択します。
3. セカンダリ出力のフィールドを変更するには、出力の行の[出力フィールド]列をクリックして、リストから目的のフィールドを選択します。

オペレーションから出力を削除する

1. [プロジェクト]ペインでオペレーションを右クリックし、[プロパティ]を選択します。

2. [出力] タブを選択します。
3. 削除する出力を選択して、 をクリックします。

出力の切り取り、コピー、貼り付けを行う

1. オペレーションの[プロパティ]シートを開いて、[出力]タブを選択します。
2. 切り取り/コピーを行う出力を選択します。
3. [出力] ツールバーで、 または  をクリックします。
4. 結果を貼り付ける場所の下の行に移動します。
5.  をクリックします。
6. オペレーションを保存します。

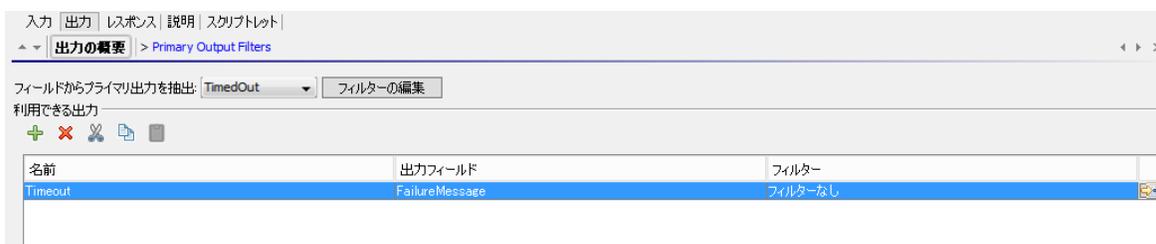
注:

- 他のオペレーションに出力をコピーすることができます。
- 同じ名前の出力を貼り付けることができます。新しい出力に自動的に <出力> (コピー-1) という名前が付けられます。

参考資料

[プロパティ]シート > [出力]タブ

[プロパティ]シートの [出力] タブで、オペレーションのプライマリ出力とセカンダリ出力を指定します。



GUI アイテム	説明
フィールドからプライマリ出力を抽出	プライマリ出力のデータを取得するフィールドを選択します。
フィルターの編集	プライマリ出力のフィルターエディターを表示します。
出力の追加	新しい出力行を追加します。
出力の削除	選択した出力行を削除します。

出力フィールド	セカンダリ出力のデータを取得するフィールドを選択します。
	その行の出力に対してフィルターエディターを表示します。

[プロパティ] シート > [出力] ツールバー



GUI アイテム	説明
出力の追加 	新しい出力行を追加します。新しい出力の名前を入力して、[OK] をクリックします。デフォルト値を使用して新しい出力の属性が割り当てられます。
出力の削除 	選択した出力行を削除します。
切り取り  (またはキーボードの Ctrl + X)	選択した出力行を現在の場所から削除します。その行を新しい場所に置くには、  オプションを使用します。
コピー  (またはキーボードの Ctrl + C)	選択した出力行をコピーします。コピーした行を新しい場所に置くには、  オプションを使用します。
貼り付け  (またはキーボードの Ctrl + V)	コピーするか切り取った出力行を現在の場所に貼り付けます。出力名が既に存在している場合、新しい出力に自動的に <出力> (コピー 1) という名前が付けられます。

ステップの結果の設定

オペレーションではさまざまな出力が生成されますが、出力はフローに自動的に保持されません。保持した場合、不要なデータでフローが低速になり、パフォーマンスに悪影響を与える可能性があります。

ステップのインスペクターの [結果] タブで、必要な結果を指定し、オペレーションの未加工の結果を機密としてマークできます。結果には、オペレーションの出力と伝搬する必要がある追加の情報が含まれています。

結果に機密データのマークを付けることもできます。

次の2つの方法で結果を格納できます。

- 同じフロー内のオペレーション、トランジション、およびプロンプトがアクセスできるフロー変数を作成する。詳細については、[「変数の使用」\(272ページ\)](#)を参照してください。

例: ステップ **LocalPing** は、ターゲットホストが使用可能かどうかを判断し、結果 **PingOutput** に ping オペレーションの出力を格納します。これにより、後続のステップで使用できるフロー変数 **PingOutput** を作成します。

次のステップ **Display** では、変数 **PingOutput** をユーザーに表示します。このステップのプロンプトのテキストは、ping の結果: {PingOutput} として設定されています。

- フローがサブフロー (別のフロー内のステップ) として使用される場合、親フロー内のオペレーション、トランジション、およびプロンプトがアクセスできる**出力フィールド**を作成する。詳細については、「[フロー内でのサブフローの作成](#)」(282ページ)を参照してください。

例: 親フローには、**Windows Health Check**フローがサブフローとして含まれているステップがあります。**Windows Health Check**フローの結果は出力フィールド **HealthCheckOutput** の値として格納されて、メインフローで使用できるようになります。

メインフローには、出力フィールド **HealthCheckOutput** の値を電子メールの本文に表示する、**メールの送信**オペレーションが含まれています。

ステップの結果には次の2種類があります。

- **未加工結果**は、フローのコンテキストで実行されたオペレーションから戻された、すべての未加工データです。ステップの未加工結果とプライマリ結果は、元のオペレーションの未加工出力とプライマリ出力から取得されます。
- ステップエディターの**[結果]**タブで作成する、その他の結果。ステップのインスペクターで、セカンダリ結果を作成して指定できます。

ステップにフローの結果を設定する前に、関連するオペレーションにプライマリ出力が設定されていることを確認します。「[オペレーションの出力の設定](#)」(243ページ)を参照してください。

セキュリティ上の理由から、オペレーションの結果を非表示する必要がある場合もあります。たとえば、オペレーションでランダムなパスワードが生成され、このパスワードをエンドユーザーに見せたくない場合があります。これは**[機密データ]**領域で指定できます。この機能は、オペレーション全体だけでなく個別のステップにも適用できます。

ヒント: 出力にフィルターを1つ以上作成すると、結果の選択の範囲をさらに絞り込むことができます。「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。

実行する操作

ステップにプライマリ結果を作成する

オペレーションにプライマリ出力が設定されています。プライマリ出力は入力に値を提供し、その入力値の割り当てには**[前のステップの結果]**を使用します。

ステップで、このプライマリ出力をフロー変数で取得する(フロー内のほかのステップで使用する)のか、フロー出力フィールドで取得する(親フローに渡す)のかを指定できます。

1. 作成ペインでステップをダブルクリックします。
2. **[結果]** タブを選択し、**[結果の追加]** をクリックします。
3. **[名前]** 列に結果の名前を入力します。キーボードの Return キーを押します。この名前は、フロー変数またはフロー出力フィールドの名前に使用されます。

注: 結果名に「Result」を使用しないでください。

4. **[ソース]** 列のリストから、結果のソースとしてプライマリ出力を選択します。

たとえば、そのオペレーションのプライマリ出力である **[結果フィールド: returnResult]** を選択します。

プライマリ出力の設定の詳細については、「[オペレーションの出力の設定](#)」(243ページ)を参照してください。

ステップにセカンダリ結果を作成する

1. 作成ペインでステップをダブルクリックします。
2. **[結果]** タブを選択し、**[結果の追加]** をクリックします。
3. **[名前]** 列に結果の名前を入力します。キーボードの Return キーを押します。この名前は、フロー変数またはフロー出力フィールドの名前に使用されます。
4. **[ソース]** リストから、結果のソースを選択します。
5. **[割り当て先]** リストで、値の保存先を指定します。
 - フロー変数に値を格納するには、**[Flow Variable]** を選択します。
 - 親フローで値を使用できるようにするには、**[Flow Output Field]** を選択します。
6. **[割り当て操作]** リストから、適切な操作を選択します。
 - **OVERWRITE** - フロー変数またはフロー出力フィールドの現在の値をこの値に置き換えます。
 - **APPEND** - フロー変数またはフロー出力フィールドの現在の値の末尾にこの値を追加します。
 - **PREPEND** - フロー変数またはフロー出力フィールドの現在の値の先頭にこの値を追加します。
 - 4つの算術割り当て操作 **ADD**、**SUB**、**MULTIPLY**、および **DIVIDE** のいずれかを使用して、フロー変数またはフロー出力フィールドの現在の値を算術的に変更します。

たとえば、ステップの結果が 3.14 の場合、**MULTIPLY** を選択すると、フロー変数またはフロー出力フィールドの現在の値と 3.14 が乗算されます。

7. セカンダリ結果の出力データにフィルターを作成するには、該当する行の端にある右向き矢印  をクリックします。

フィルターの作成の詳細については、「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。

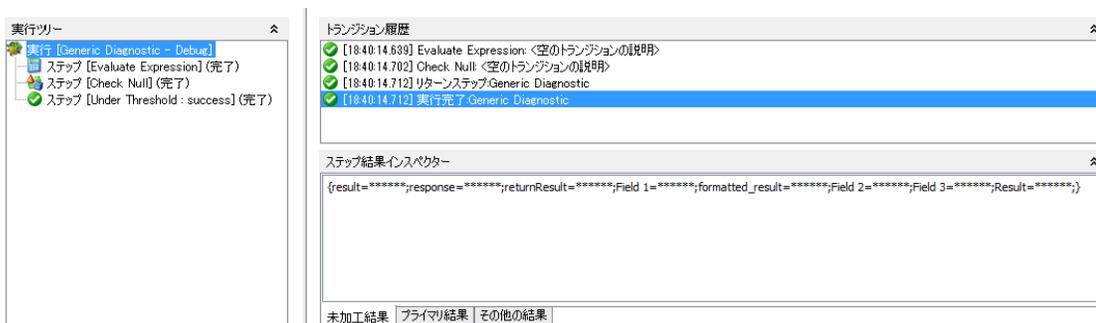
オペレーションの未加工の結果の非表示

1. [機密データ] 領域で、[オペレーションの結果の非表示] チェックボックスをオンにします。これにより、オペレーションの未加工の結果全体が非表示になり、Studio および Central のユーザーインターフェースでこの情報がアスタリスクで表示されます。

このチェックボックスをオンにした後で、オペレーションのプライマリ出力も非表示になり、Studio および Central のユーザーインターフェースでこの情報がアスタリスクで表示されます。

スクリプトレットで使用すると、機密データが暗号化された状態で取得されます。

たとえば、ステップの [式の評価] で [オペレーションの結果の非表示] チェックボックスをオンにした場合、Studio では下のように表示され、Central でも同様に表示されます。



ステップの結果の非表示

1. 作成ペインでステップをダブルクリックします。
2. [結果] タブを選択します。
3. ステップの結果のリストで非表示にするステップの結果を見つけます。
4. [結果] タブで、[機密データ] チェックボックスをオンにします。

これにより、Central ユーザーインターフェース、および Studio の内部デバッガーとリモートデバッガーで結果が非表示になります。結果の値はアスタリスクで表示され、OO データベースに保持されません。

このフラグは一時的であり、機密変数から割り当てられると、割り当てられた変数も機密になります。一時的な効果が見られるのはランタイム実行中のみであり、Studio ユーザーインターフェースには反映されません。

結果が機密の入力または機密の結果から割り当てられている場合は、結果も機密になります。結果が非表示になっているオペレーションの未加工の結果から割り当てられているとき、Studio および Central のユーザーインターフェースで表示にする場合、結果を機密としてマークする必要があります。

スクリプトレットで使用すると、機密データが暗号化された状態で取得されます。

割り当てのための式を使用すると、機密性が保持されます。たとえば、\${input1} のようになります。

ステップの未加工の結果がエンドユーザーに表示されないようにする

1. 作成ペインでステップをダブルクリックします。
2. [結果] タブを選択します。
3. [機密データ] 領域で、[オペレーションの未加工結果の非表示] を選択します。
4. ステップを保存します。

注: ステップにサブフローが含まれている場合は、[オペレーションの未加工結果の非表示] セクションはありません。

結果の切り取り、コピー、貼り付けを行う

1. 作成ペインでステップをダブルクリックします。
2. [結果] タブを選択し、切り取り/コピーを行う結果を選択します。
3. [結果] ツールバーで、 または  をクリックします。
4. 結果を貼り付ける場所の下の行に移動します。
5.  をクリックします。
6. ステップを保存します。

注:

- ステップ結果は別のステップ結果ペインにのみコピーできます。フロー出力は、他のフローまたはオペレーションにコピーできます。
- 同じ名前のステップ結果を貼り付けることができます。新しい結果は、既存の結果と同じになります。

ステップから結果を削除する

1. 作成ペインでステップをダブルクリックします。
2. [結果] タブを選択し、[結果] ツールバーの  をクリックします。
3. ステップを保存します。

結果データを取得するフィールドを変更する

1. 作成ペインでステップをダブルクリックします。
2. [結果] タブを選択します。
3. 出力の行の [ソース] 列をクリックし、リストから目的のフィールドを選択します。
4. ステップを保存します。

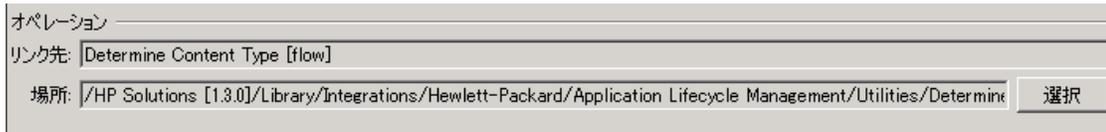
ステップにリンクされたオペレーションの変更

異なるオペレーションを選択してステップにリンクすることができます。新しいオペレーションを選択した後に、既存の入力を新しいオペレーションの入力で上書きするか、新しい入力と既存の入力を

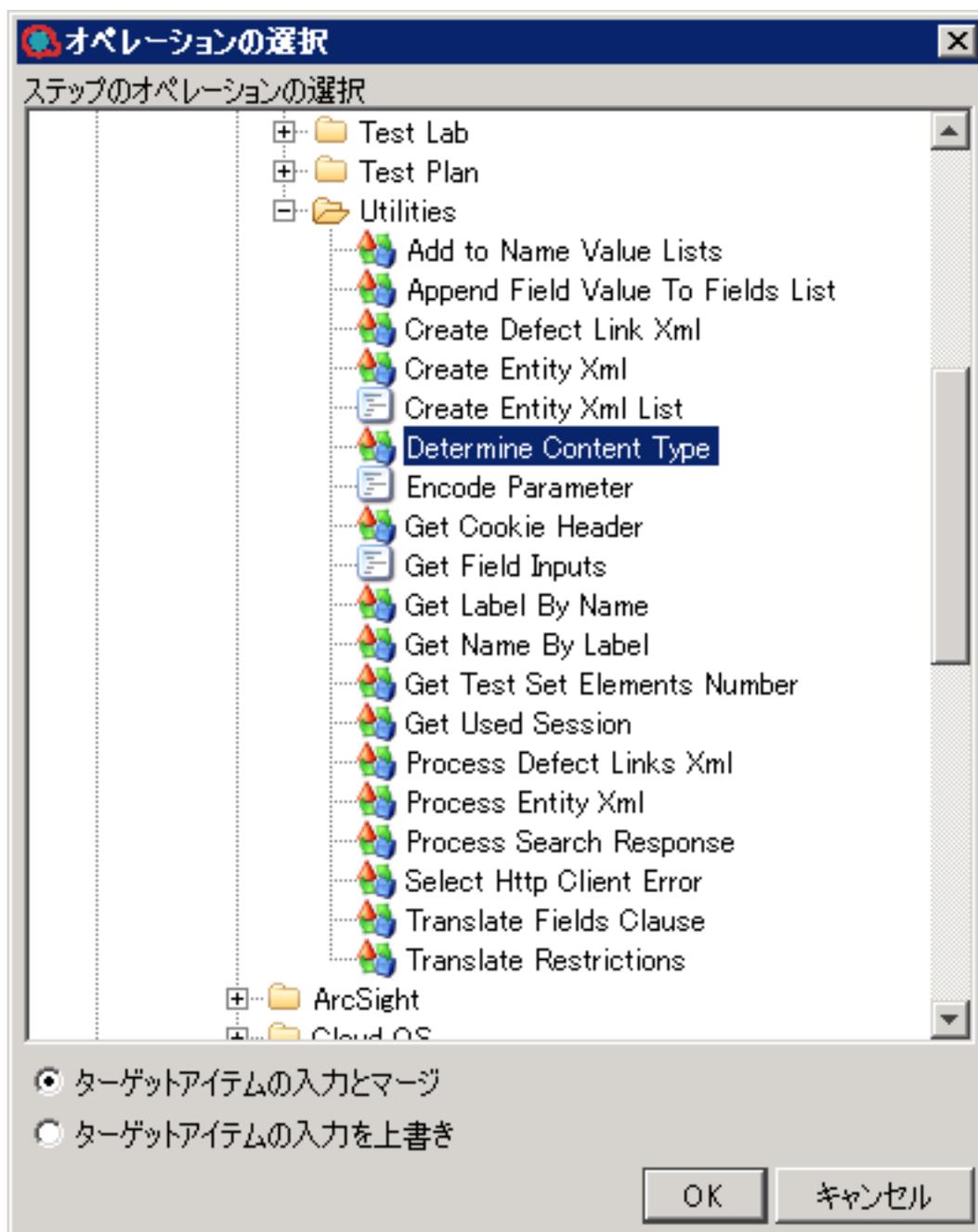
マージするかを選択できます。

1. 作成ペインでステップをダブルクリックします。
2. ステップのインスペクターで **[詳細]** タブを選択します。

現在のオペレーションの場所と名前を表示できます。例:



3. **[選択]** ボタンをクリックします。[オペレーションの選択] ウィンドウが表示されます。



- 新しいオペレーションを参照します。
- 新しいオペレーションの入力を現在の入力に追加するには、[ターゲットアイテムの入力とマージ]を選択します。

注: 入力名が既に存在している場合は、元の入力値が残り、上書きされません。

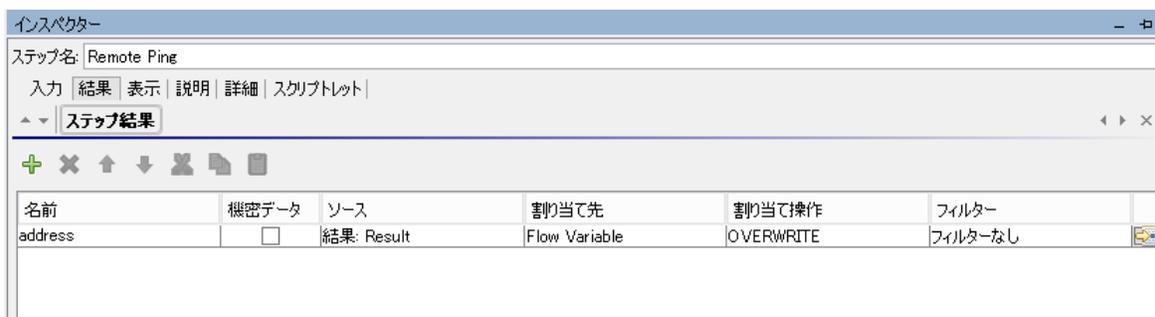
- 現在の入力を削除して新しい入力のみを追加するには、[ターゲットアイテムの入力を上書き]を選択します。

7. [OK] をクリックします。
8. ステップを保存します。

参考資料

ステップのインスペクター > [結果] タブ

ステップのインスペクターの [結果] タブで、フロー変数に保存する出力を指定したり、親フローで使用できるようにする出力を指定します。



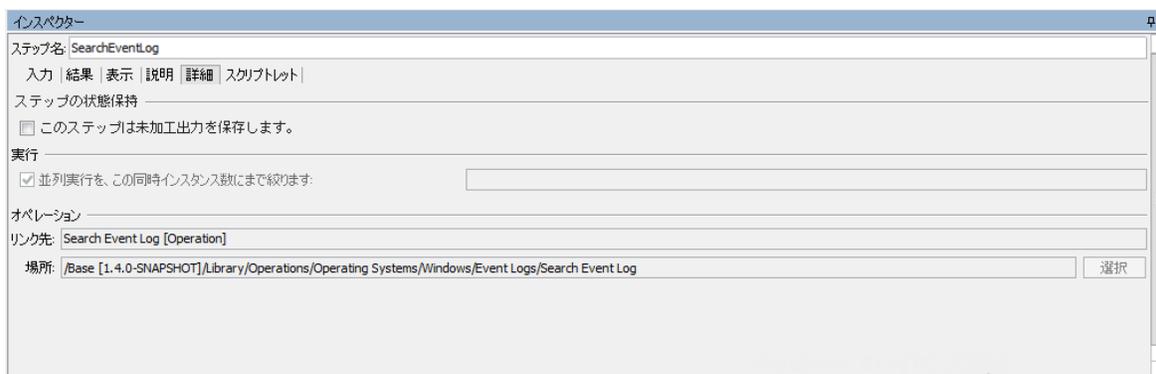
GUI アイテム	説明
名前	結果の名前を入力します。この名前は、フロー変数またはフロー出力フィールドの名前に使用されます。
機密データ	Central ユーザーインターフェイス、および Studio の内部デバッガーとリモートデバッガーでこの結果を非表示にします。結果アスタリスクで表示され、OO データベースに保持されません。
結果の追加	新しい結果行を追加します。
結果の削除	選択した結果行を削除します。
ソース	結果のソースを選択します。
割り当て先	結果の値を格納する場所を選択します。 <ul style="list-style-type: none"> • フロー変数に値を格納するには、[Flow Variable] を選択します。 • 親フローで値を使用できるようにするには、[Flow Output Field] を選択します。

<p>割り当て操作</p>	<p>[割り当て操作] リストから、適切な操作を選択します。</p> <ul style="list-style-type: none"> • OVERWRITE - フロー変数またはフロー出力フィールドの現在の値をこの値に置き換えます。 • APPEND - フロー変数またはフロー出力フィールドの現在の値の末尾にこの値を追加します。 • PREPEND - フロー変数またはフロー出力フィールドの現在の値の先頭にこの値を追加します。 • 4つの算術割り当て操作 ADD、SUB、MULTIPLY、および DIVIDE を使用して、フロー変数またはフロー出力フィールドの現在の値を算術的に変更します。 <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p>たとえば、ステップの結果が 3.14 の場合、MULTIPLY を選択すると、フロー変数またはフロー出力フィールドの現在の値と 3.14 が乗算されます。</p> </div>
	<p>その行の結果のフィルターエディターを表示します。</p>

注: ステップの結果を機密データとしてマークすると、実行の間中機密のままになります (そのため暗号化されます)。ステップの結果は、その結果から割り当てられるすべての入力/ステップの結果にこの動作を継承します。

ステップのインスペクター > [詳細] タブ

ステップのインスペクターの [詳細] タブで、ステップのベースとなるソースオペレーションを変更できる場所を指定します。



GUI アイテム	説明
<p>このステップは未加工出力を保存します</p>	<p>ステップのすべての未加工の結果を保存する必要がある場合は、このチェックボックスをオンにします。</p>
<p>リンク先</p>	<p>ステップのベースとするソースオペレーションを表示します。</p>

場所	ステップのベースとするソースオペレーションの場所を表示します。
選択	[オペレーションの選択] ダイアログボックスを開きます。このダイアログボックスで、ステップのベースとなるオペレーションに移動して、そのオペレーションを選択できます。

出力および結果のフィルタリング

オペレーションの出力や、ステップの結果の抽出や編集を行う際にフィルターを作成することができます。

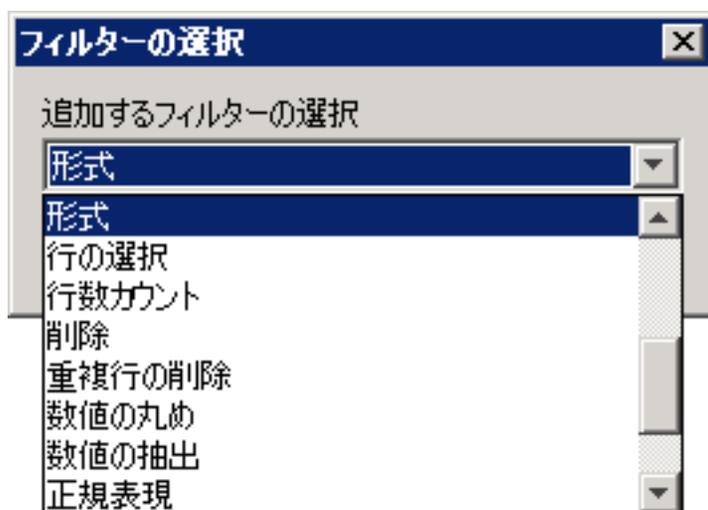
たとえば、サーバーに対する ping オペレーションについて、最大、最小、平均の往復時間のみが必要だとします。未加工出力を3つの出力にフィルタリングして、これらの情報を未加工出力から分離して抽出できます。

実行する操作

フィルターの作成

- フィルターエディターを開きます。この手順は、フィルタリング内容によって異なります。
 - オペレーションのプライマリ出力に1つ以上のフィルターを作成するには、オペレーションの[プロパティ]シートを開き、[出力]タブをクリックして、[フィルターの編集] ボタンをクリックします。
 - オペレーションのセカンダリ出力にフィルターを作成するには、オペレーションの[プロパティ]シートを開き、[出力]タブをクリックして、出力行の端にある右向きの矢印  をクリックします。
 - ステップのセカンダリ結果にフィルターを作成するには、作成ペインでステップをダブルクリックし、[結果]タブをクリックして、結果行の端にある右向きの矢印  をクリックします。
- フィルターエディターで、[追加] ボタンをクリックします。

注: 出力または結果には複数のフィルターを追加できます。
- [フィルターの選択] リストで、フィルターの種類を選択します。



4. フィルターエディターの右上にあるエリアの **[詳細]** で、フィルターを設定します。それぞれのフィルターのオプションの詳細については、[「フィルターオプション」](#) を参照してください。

コマンドラインからのデータを用いたフィルターのテスト

テストを行うには、一部のデータを **[フィルター入力のテスト]** ボックスに貼り付けます。このデータがローカルのコマンドラインコマンドで生成できる場合、次のようにします。

1. 出力または結果のフィルターエディターを開きます。
2. **[クリア]** をクリックして **[フィルター入力のテスト]** ボックスの内容をクリアします。
3. **[クイックコマンド]** をクリックします。
4. 目的のデータを生成するコマンドを入力します。
5. **[OK]** をクリックします。コマンドの出力が **[フィルター入力のテスト]** ボックスに表示されます。
6. 次のいずれかを実行します。
 - **[すべてのフィルターのテスト]** をクリックする
 - テストするフィルターを選択して **[選択したフィルターのテスト]** をクリックする

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

デバッガーからのデータを用いたフィルターのテスト

必要なデータが単純なコマンドラインコマンドで複製できない手段で生成されている場合、次のようにデータをデバッガーからコピーして、**[フィルター入力のテスト]** ボックスに貼り付けます。

1. 出力または結果のフィルターエディターを開きます。
2. **[クリア]** をクリックして **[フィルター入力のテスト]** ボックスの内容をクリアします。
3. デバッガーでフローを実行します。
4. 該当するステップを強調表示します。

5. [ステップ結果インスペクター] ペインで、[未加工結果] タブのコンテンツをコピーします。
6. フィルターエディターで、コンテンツを [フィルター入力のテスト] ボックスに貼り付けます。
7. 次のいずれかを実行します。
 - [すべてのフィルターのテスト] をクリックする
 - テストするフィルターを選択して [選択したフィルターのテスト] をクリックする

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

さまざまな出力や結果のフィルタリング

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

出力または結果でのシステムフィルターの使用

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



フィルターをシステムフィルターとして保存

オペレーションの既存のフィルターを引用し、システムフィルターとして保存できます。結果のシステムフィルターは、作成したオペレーションに依存せず、ほかの出力や結果で再使用できます。

詳細については、「[システムフィルターの設定](#)」(175ページ)を参照してください。

フィルターオプション

大文字と小文字の変換

[**大文字と小文字の変換**] フィルターは、文字列に含まれるすべての文字を大文字または小文字に変更します。[**大文字に変換**] チェックボックスを選択しないでそのままにした場合、このフィルターによりすべての文字が小文字に変更されます。

数値の抽出

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

形式

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

詳細: 形式
入力の前頭や末尾に解決された表現を追加したり、入力を解決された表現で置き換えることで、入力の形式を調整します

テキスト: %n

入力の配置: 末尾に追加

- 先頭に追加
- 末尾に追加
- 置換

1. [テキスト] ボックスに、結果に付加するテキスト、または結果と置換するテキストを入力します。
2. [入力の配置] リストで、次を行います。
 - テキストを既存のテキストの前に付加する場合は、[先頭に追加] を選択する
 - テキストを既存のテキストの後ろに付加する場合は、[末尾に追加] を選択する
 - 出力をテキストに置換するには、[置換] を選択する

行数カウント

[**行数カウント**] フィルターは、結果に含まれる行の総数を出力します。

正規表現

[**正規表現**] フィルターは、正規表現 (regex) を使用して未加工結果をフィルタリングします。

1. **[正規表現の種類]** リストで、**[Java スタイル]** を選択します。ほかのスタイルは使用しないでください。推奨されていません。
2. **[正規表現の値]** ボックスに、正規表現を入力します。
3. **[フィルターの様スタイル]** で、未加工結果にどのようにフィルターを適用するかに応じて、**[入力全体をフィルター]** または **[行単位でフィルター]** を選択します。
4. 正規表現の大文字小文字を区別しない場合は、**[大文字と小文字を区別しない]** を選択します。

正規表現の使用の詳細については、「[フローでの正規表現の使用](#)」(302ページ)を参照してください。

重複行の削除

このフィルターは同一の行を検出し、そのうち1つを残してすべて削除します。

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

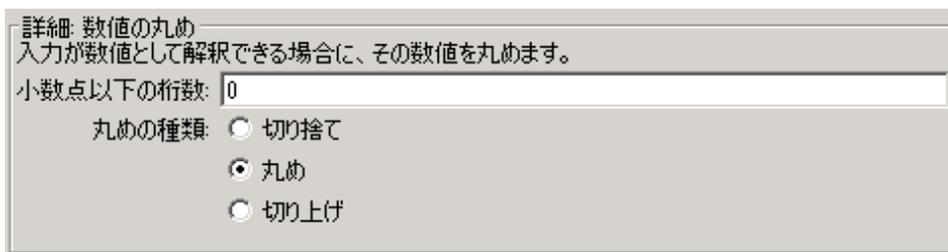
置換

このフィルターは文字列の最初または最後のインスタンスか、すべてのインスタンスを別の文字列と置換します。

1. **[検索]** ボックスに、検索して置換する対象の文字列を入力します。
2. 置換する **[対象:]** 文字列のインスタンスに応じて、**[置換]** から **[先頭]**、**[すべて]**、または **[末尾]** を選択します。
3. **[置換]** ボックスに、対象文字列と置換する文字列を入力します。
4. 検索時に大文字小文字を区別しない場合は、**[大文字と小文字を区別しない]** チェックボックスを選択します。

数値の丸め

このフィルターは、数値を切り捨てまたは切り上げて丸めます。



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

スクリプトレット

このフィルターは、作成したスクリプトレットを使用してデータをフィルタリングします。



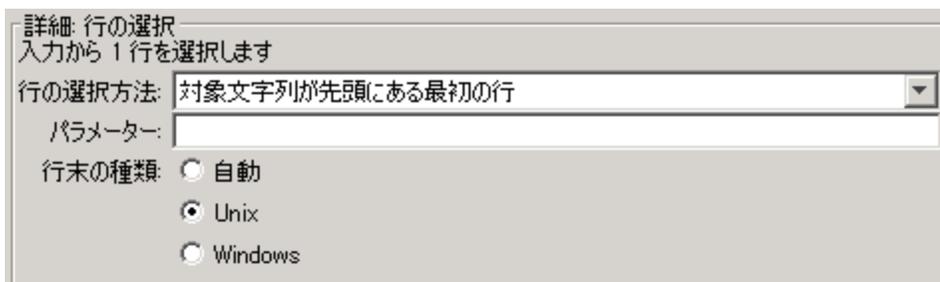
1. フィルターとして使用するのに必要となるスクリプトレットの行を取得するには、**[テンプレートの挿入]** をクリックします。

挿入されたテンプレートは、選択した言語に固有のもので、フロー変数、オペレーション結果、および入力へのアクセスや、フロー変数および結果の設定と加工に最もよく使用するコマンドを含んでいます。
2. スクリプトレットをデバッグするには、**[スクリプトの確認]** をクリックします。

スクリプトレットの詳細については、[「フロー内でのスクリプトレットの使用」\(297ページ\)](#)を参照してください。

行の選択

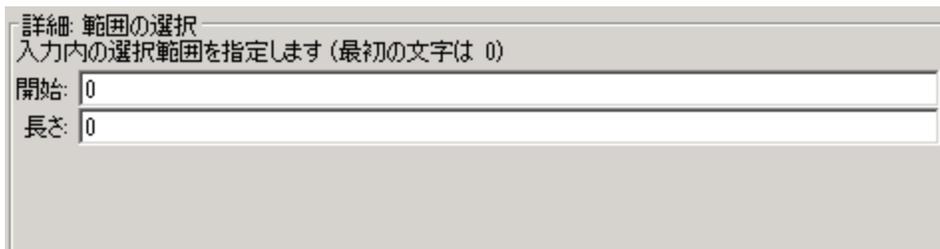
このフィルターは未加工結果から抽出する行を指定します。



1. **[行の選択方法]** リストから、対象とする行の基準を選択します。
2. **[パラメーター]** ボックスに、対象文字列を含む文字列を入力します。
3. **[行末の種類]** グループで、次のいずれかを選択します。
 - フィルタリングしているテキストが、Unix オペレーティングシステム (LF で行を終了) で生成されている場合は、**[Unix]** を選択します。
 - フィルタリングしているテキストが、Windows オペレーティングシステム (CR/LF で行を終了) で生成されている場合は、**[Windows]** を選択します。
 - フィルターで、どちらの種類も行末でも処理できるようにするには、**[自動]** を選択します。デフォルトでは**[自動]** が選択されています。

範囲の選択

このフィルターは、入力データから抽出する文字列を指定します。指定する際に使用する2つの条件は、文字列の長さ (文字数) と、入力データの先頭から見た文字列の最初の文字の位置です。



1. **[開始]** ボックスに、文字列の開始位置をゼロを基準として入力します。
2. **[長さ]** ボックスに、文字列の文字数を入力します。

改行は、フィルタリングしているデータの取得元のオペレーティングシステムに応じて、1文字または2文字としてカウントされます。

ソート

このフィルターは、各行の最初の文字を使用して入力データの行を並べ替えます。

詳細: ソート
改行単位で入力をソートします。
昇順:
数値として扱う:

1. 次のようにして、並べ替えの向きを指定します。
 - 昇順で並べ替える場合、[昇順] チェックボックスを選択します。
 - 降順で並べ替える場合、[昇順] チェックボックスのチェックを外します。
2. ASCII 順でデータを並べ替える場合、[数値として扱う] チェックボックスを選択します。
昇順の ASCII 順は、英語の文字の場合おおまかに次のようになります。
 - 空白
 - 記号
 - 数字
 - アルファベット

削除

このフィルターは、未加工結果の最初または最後から文字を削除します。

注: このフィルターがほかのフィルターの後に適用されると、前のフィルター処理により取得された未加工結果のサブセットの最初または最後から文字を削除します。

詳細: 削除
入力の先頭または末尾から、一致した文字列を削除します
削除方法: 対象文字より前 (対象文字も含む) のすべての文字
削除対象文字:

1. [削除方法] リストから、フィルターで未加工結果から文字を削除する方法を選択します。[削除対象文字] テキストボックスで指定した文字列を削除する際の次のようなオプションを指定できます。
 - [対象文字より前のすべての文字] の文字列
 - [対象文字より前 (対象文字も含む) のすべての文字] の文字列

- [対象文字より後のすべての文字] の文字列
- [対象文字より後 (対象文字も含む) のすべての文字] の文字列

2. [削除対象文字] テキストボックスに、検索する文字列を入力します。

スペースの削除

このフィルターは未加工結果の先頭および末尾から空白文字をすべて削除します。

テーブル

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

詳細: テーブル	
入力をテーブルとして解析し、指定された列でソートします	
列の区切り記号: <input type="text" value="スペース"/>	行の区切り記号: <input type="text" value="改行"/>
最初の行はヘッダー: <input type="checkbox"/>	結果の最初の行を削除: <input type="checkbox"/>
ソート基準列: <input type="text" value="-1"/>	昇順: <input type="checkbox"/>
選択行: <input type="text" value="0"/>	選択列: <input type="text" value="0"/>
選択幅: <input type="text" value="1"/>	選択高: <input type="text" value="1"/>

注: 行の番号設定は 0 ベース (0 [0] から始まる)、列の番号設定は 1 ベースです。

1. [列の区切り記号] リストでは、データを列に分割するための記号を選択します。
2. [行の区切り記号] リストでは、データを行に分割するための記号を選択します。

注: 複数の連続した空白は 1 つの空白と見なされるため、列の右にあると想定していたデータで列が占められることがあります。たとえば、列の区切り記号に空白を指定して `dir` コマンドラインコマンドの出力にこのフィルターを適用する場合は。

3. 最初の行を列ヘッダーとするには、[最初の行はヘッダー] を選択します。
4. 最初の行を削除するには、[結果の最初の行を削除] を選択します。
5. 列で並べ替えを行うには、[ソート基準列] ボックスに列番号を (1 ベースで) 入力します。

ヒント: 値 -1 は列で並べ替えを行わないことを意味します。

6. 昇順を指定するには、[昇順] ボックスを選択します。
デフォルトの並べ替え順序は、降順です。
7. フィルタリングして抽出する行を選択するには、次の操作を行います。
 - [選択行] ボックスに、行番号 (0 ベース) を入力します。

ヒント: -1 を選択すると、データ内のすべての行が選択されます。

- **[選択幅]** ボックスに、抽出する行の列数を入力します。

ヒント: -1 を選択すると、**[選択列]** に指定した列の右にある、データ内の残りの列がすべて選択されます。

8. フィルタリングして抽出する列を選択するには、次の操作を行います。

- **[選択列]** に列番号を入力します。

ヒント: -1 を選択すると、データ内のすべての列が選択されます。

- **[選択高]** ボックスに、抽出する列の行数を入力します。

ヒント: -1 を選択すると、**[選択行]** に指定した行の下にある、データ内の残りの行がすべて選択されます。

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

- **選択行:** 0
- **選択高:** 5
- **選択列:** 2
- **選択幅:** 3

XML フィルター

XML フィルターを使用すると、ステップ内の XML を解析できます。XML は HP 00 デフォルトコンテンツの XML 処理オペレーションのいずれかに XML を渡すフローを作成せずに、ステップの入力または結果から XML を取得することができます。

オペレーションでの XML フィルターの使用と、デフォルトコンテンツでの XML 処理オペレーションの使用は、いくつかの点で異なります。

- オペレーション内でのタスクの完了と、タスクの完了までのフローのインフラストラクチャーの仕様が異なる
- オペレーション内のフィルターには、XML 処理オペレーションにはない制限がいくつかあります。これらの制限については、該当のフィルターについて次のセクションで説明します。入力 XML のフィルタリングにフィルターとオペレーションのどちらを選ぶかは、XML の取得方法によって決定します。

XML フィルターは次のとおりです。

- XML 属性の取得
- XML 要素の取得
- XML 要素値の取得
- XPath クエリ

XML フィルターを説明するため、例では次の XML 例を参照することとします。

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

XML 属性の取得

[XML 属性の取得] フィルターでは、指定する属性の1つまたは複数のインスタンスそれぞれの値を抽出します。フィルターエディターで、属性への要素のパスを指定してフィルターが適用される属性のインスタンスを管理することができます。

属性の単一インスタンスに対する値を取得することも、複数のインスタンスに対する値を表形式で取得することもできます。この表では、列はコンマ区切りとなり、行は改行区切りとなります。

詳細: XML 属性の取得
XML 文書から、要求された属性値をフィルターします。高度な XML フィルター処理には XPath フィルターを使用します。

要素パス:

サブ要素を含む:

属性名:

結果: 1つの一致
 テーブル

1. [要素パス] ボックスに、抽出する値の属性を含む要素のパスを指定します。要素へのパスを部分ごとに区切るには、フォワードスラッシュ (/) を使用します。
フィルターの属性値の取得元の要素のインスタンスを管理するには、[2] や [3] のように指定を追加します。要素の番号付けは、1 ベース ([1] から開始) です。したがって、要素の2つ目のインスタンスを指定するには、[2] を使用します。
2. 指定した要素の子要素を検索するには、[サブ要素を含む] チェックボックスを選択します。
3. [属性名] ボックスに、値を取得する属性の名前を入力します。
4. [結果] で次のいずれかを選択します。
 - 抽出結果を属性の単一インスタンスの値に制限するには、[1つの一致] を選択します。
 - 指定した属性のすべてのインスタンスの値を抽出するには、[テーブル] を選択します。

例: コメントのいずれかでユーザー名を検索するには (XML フィルターのトピックで使った XML の例を使用します)

[要素パス] ボックスに、`/ticket/details/comment` と入力します。

例: 特定のコメント (この例では2つ目のコメント) のユーザー名を取得するには、

1. [要素パス] ボックスに `/ticket/details/comment[2]` と入力します。
2. [属性名] ボックスに、`user` と入力します。
3. [結果] の横の [1つの一致] を選択します。

john と出力されます。

例: 各コメントのユーザー名を検索するには

1. [要素パス] ボックスに `/ticket/details/comment` と入力します。
2. [属性名] ボックスに、`user` と入力します。
3. [結果] の横の [テーブル] を選択します。

次のよう出力されます。

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

XML 要素の取得

[XML 要素の取得] フィルターで、次のいずれかの方法で記述することで、要素全体 (子要素、値、属性を含む) を抽出できます。

- 相対パスまたは絶対パスを指定する。
- 抽出する要素の子要素を指定する。子要素の特定の値を検索することもできます。
- 抽出する要素の属性を指定する。属性の特定の値を検索することもできます。

詳細: XML 要素の取得
XML 文書から指定されたパスの要素をフィルターします。高度な XML フィルター処理には XPath フィルターを使用します。

要素パス:	<input type="text"/>		
子の名前:	<input type="text"/>	値:	<input type="text"/>
属性名:	<input type="text"/>	値:	<input type="text"/>

次の手順で、テキストボックスを1つまたは組み合わせて使用して詳細を入力できます。

1. [要素パス] ボックスに要素の絶対パスを入力します。

パス内の相対パスインジケータは、相対パスインジケータの前の要素の相対位置を表します。

- `../` は、最後に指定した要素の親を示します。
- `./` は、最後に指定した要素です。

例: 例の XML では、`<volume>` および `<company>` は兄弟要素で、両方とも `<customer>` 要素の子です。`<company>` 要素は、次のような相対パスで指定することもできます。

```
/tickets/ticket/customer/volume/./company
```

特定した要素に複数のインスタンスがある場合は、上記の例のようにパスを指定するだけで、要素のすべてのインスタンスが返されます。

パスに角括弧で囲んだ整数を使用して、要素の特定のインスタンスを指定することができます。

例:

/tickets/ticket/details/comment とすると、すべてのチケットのすべてのコメントの詳細が指定されます。

/tickets/ticket/details/comment[2] とすると、各チケットの2つ目のコメントが指定されます。

/tickets/ticket[2]/details/comment とすると、2つ目のチケットのすべてのコメントが指定されます。

2. **[子の名前]** ボックスに、抽出する要素の子である要素の名前を入力します。子要素に値がある場合、**[値]** ボックスに値を入力することで結果を絞り込みます。
 - **[子の名前]** ボックスは、単一レベルの子要素に対してのみ動作します。フィルタリングの結果、指定した子要素の直接の親が返されます。
 - **[値]** ボックスは、簡略化した値を処理します。ここに入力する値は、抽出する要素の子要素の値と正確に一致している必要があります。
3. **[属性名]** ボックスに、抽出する要素に一意の属性名を入力します。さらに結果を絞り込むには、**[値]** ボックスに属性の値を入力します。

例: 例の XML では、customer 要素とそのコンテンツを抽出する複数の方法があります。

- **[要素パス]** ボックスに **/ticket/customer** と入力する
- **[子の名前]** ボックスに、次のように customer の子要素のいずれかを入力する。

company

position

contactInfo

[子] ボックスに **company** と入力した場合、付属の **[値]** ボックスに **myOrg** と入力することができます。

- **[属性名]** ボックスに、次のいずれかを入力します。

名

姓

付属の **[値]** ボックスに、次のようにこれらの属性に対応する値を入力できます。

John

Doe

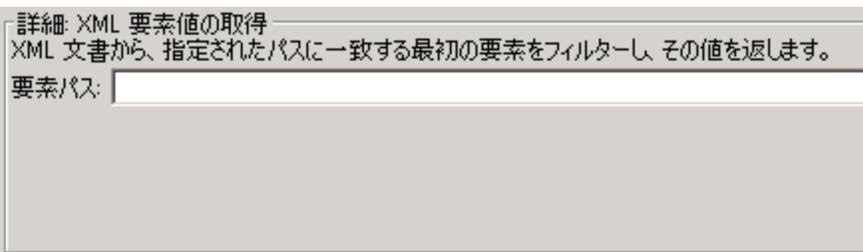
これらのフィルターのそれぞれで、出力は次のような customer 要素になります。

```
<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] フィルターで、特定の要素の値を取得できます。



[要素パス] ボックスに、対象の値の要素へのパスを入力します。

ほかのフィルターと同様、要素に複数のインスタンスがある場合、別のインスタンスを指定しない限り、フィルターは最初のインスタンスを返します。

例: サンプル XML の場合

電子メール要素の値を取得するには、**/tickets/ticket/customer/contactInfo/email** と入力します。

出力は指定した 2 つの電子メールのいずれかです。

jdoe@myorg.com

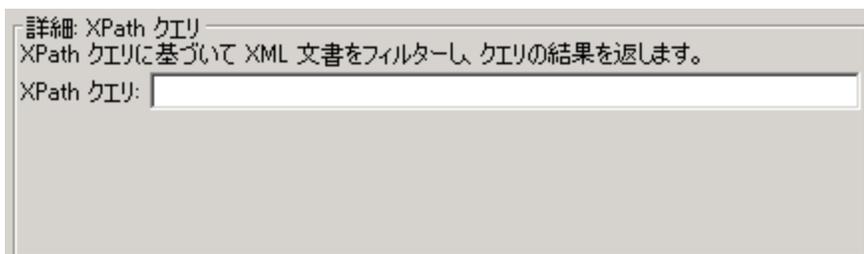
johnsSecondEmail@myorg.com

電子メール要素の特定のインスタンスを指定するには、**/ticket/customer/contactInfo/email[2]** と入力します。

次のように出力されます。johnsSecondEmail@myorg.com

XPath クエリ

[XPath クエリ] フィルターで、[XPath クエリ] ボックスに入力した標準の XPath 構文を使用したクエリで結果からデータを抽出することができます。



[XPath クエリ] ボックスに、XPath 構文を使用してクエリを入力します。

- 角括弧の前のパスで、結果を絞り込むクエリの範囲を指定します。
- 角括弧内には、クエリのフィルタリング部分を入れます。クエリに、複数のフィルターのセットを含めることができます。

例: サンプル XML の場合

次のいずれかのクエリを使用して、40,000 単位を超えるボリュームの顧客を抽出できます。

- この XPath クエリでは、40,000 を超えるボリュームの顧客を抱える企業がすべて検索されます。

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

<volume> 要素は、<company> タグの兄弟のため、要素 <volume> を探すためには、角括弧内のシーケンスを次のようにして、<company> に対するパスを記述します。

```
../
```

- この XPath クエリでは、40,000 単位を超えるボリュームの顧客がすべて検索されます。

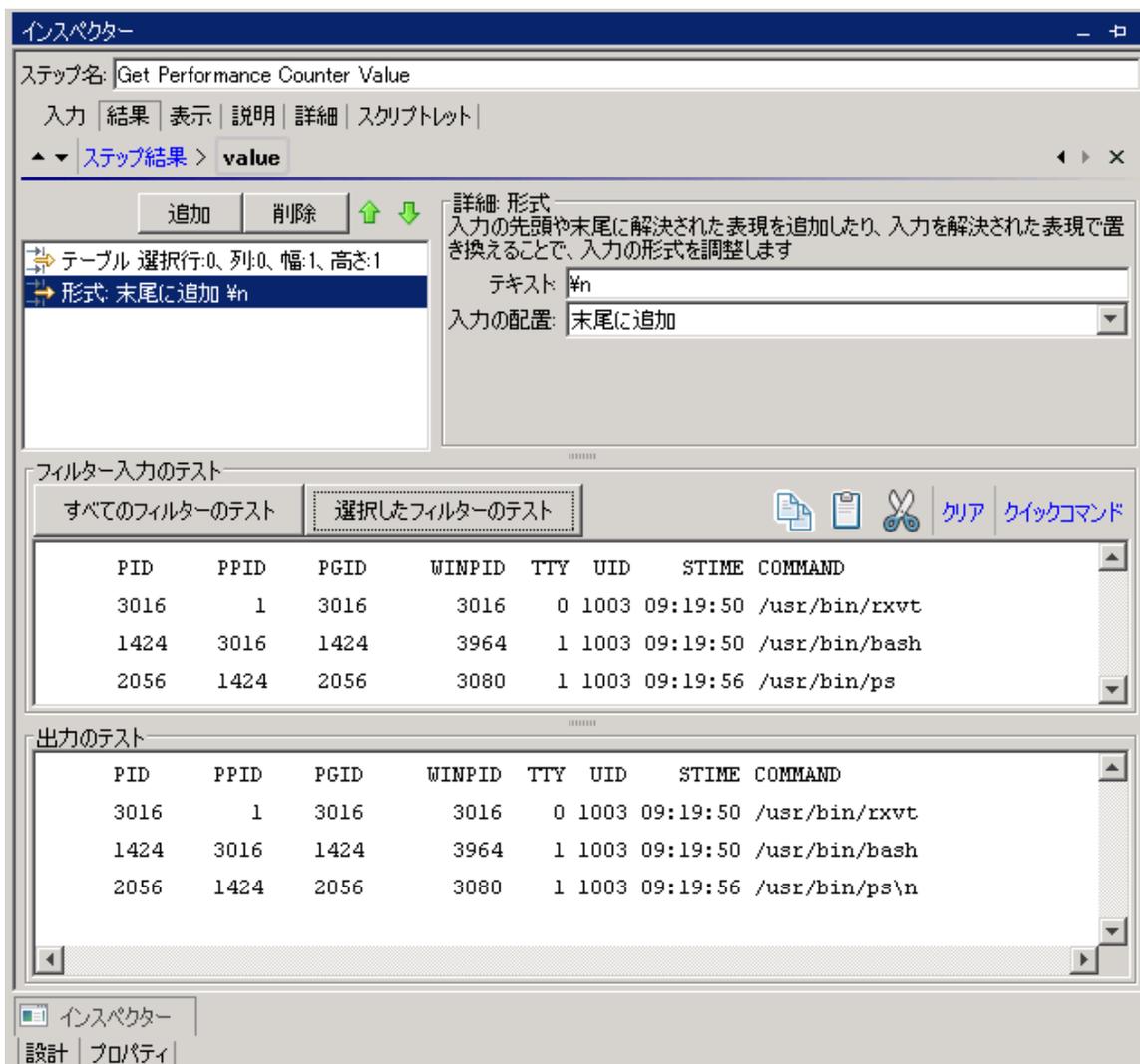
```
/tickets/ticket/customer[volume>40000]
```

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

参考資料

フィルターエディター

左上のフィルターリストに作成したフィルターのリストが表示されます。



フィルターを作成して、フィルターの種類を選択する場合、右上の [詳細:] セクションは、選択したフィルターの種類に従って、フィルターを変更するコントロールが表示されるように変更されます。

GUI アイテム	説明
追加	クリックすると新しいフィルターが追加されます。
削除	クリックすると選択したフィルター (複数も可) が削除されます。
	クリックすると、選択したフィルターがリストの上または下に移動します。フィルターは、リストに表示されている順番で処理されます。
フィルター入力のテスト	フィルターが想定どおりに動作するかテストするためのデータを配置する場所です。

すべてのフィルターのテスト	テストを、出力または結果のすべてのフィルターに適用します。
選択したフィルターのテスト	テストを選択したフィルターに適用します。
コピー	[フィルター入力のテスト] ボックスのデータをコピーします。
貼り付け	[フィルター入力のテスト] ボックスにデータを貼り付けます。
切り取り	[フィルター入力のテスト] ボックスのデータを切り取ります。
クリア	[フィルター入力のテスト] ボックスのデータを消去します。
クイックコマンド	フィルターをテストするデータを生成するコマンドを入力します。コマンドの出力が[フィルター入力のテスト] ボックスに表示されます。
出力のテスト	フィルターが[フィルター入力のテスト] ボックスのテストデータに適用され、フィルタリング結果が[出力のテスト] ボックスに表示されます。

変数の使用

変数を使用すると、フロー内やフロー間でデータを移動できます。

たとえば、サーバーで実行する複数のステップがある場合、最初のステップでサーバーの IP アドレスを取得し、その値をフロー変数に割り当てることができます。以降、同じ名前の入力を持つ後続のステップではすべて、自動的にそのサーバー名が使用されます。

フロー変数

フロー変数は、フロー変数が定義されているフロー内でのみ使用できます。

フロー変数への値の割り当て

次の値をフロー変数に割り当てることができます。

- **ステップの結果** - たとえば、該当項目をカウントするオペレーションが含まれたステップによって、結果をフロー変数に格納します。
- **入力値** - たとえば、IP アドレスを入力値として取得するステップによって、アドレスをフロー変数として格納します。
- **スクリプトレット** - たとえば、ステップのオペレーションから戻されたデータを評価するスクリプトレットによって、データをフロー変数に格納します。

フロー変数の使用

フロー変数およびフロー変数に格納されているデータを、次の場所で参照できます。

- **同じフロー内の別のステップ**
- **並列分岐ステップのレーン内** - 同じレーン内の、または並列分岐ステップより前の先行ステップによってフロー変数に値が書き込まれた場合、レーンステップでフロー変数の値を使用できます。ただし、フロー変数の値が別のレーン内のステップによって変数に書き込まれた場合は、レーン内のステップでこの値を使用することはできません。
- **オペレーションの入力**
- **フロー、ステップ、トランジションの説明** - たとえば、**ping の遅延**オペレーションは、ping の平均応答時間をフィルターにより除外します。このオペレーションに関連付けられているステップはこの平均応答時間をフロー変数 `latency` として保存し、さらにこのステップに続くトランジションがその値をユーザーにレポートすることが可能です。
- **レスポンスルールを使用してテストするデータの一部** - たとえば、フロー変数に格納した値が出力文字列またはエラー文字列に含まれているかどうかを確認します。
- **スクリプトレット内** - スクリプトレットの結果をステップの外部で利用できるようにするには、スクリプトレットでフロー変数を作成 (必要なフロー変数が存在しない場合) し、このフロー変数に結果を割り当てる必要があります。
- **オペレーションのパラメーター内** - オペレーションのパラメーターが値をとる場合、その値が格納されているフロー変数を参照すると、値にアクセスできます。

[**フロー変数**] ペインを使用すると、作成したフロー変数を追跡できます。

グローバル変数

グローバル変数は、グローバルコンテキストに含まれているキー名と値のペアであるため、常にすべてのフロー実行で使用または参照できます。

フロー変数とグローバル変数の名前が同じである場合、その変数名の参照は、グローバル変数ではなく、その名前の (ローカルの) フロー変数にアクセスします。これは、変数に値を割り当てる場合、または変数の値を取得する場合の、いずれの場合にも当てはまります。

入力の値をグローバル変数から取得するように指定している場合、フロー変数はそのグローバル変数の値を使って作成され、この値がフロー変数から入力に提供されます。

実行する操作

入力の値をフロー変数に割り当てる

デフォルトでは、入力の値は、この入力と同じ名前を持つフロー変数に割り当てられます。

1. [**プロパティ**] シート (オペレーションの場合) またはステップのインスペクター (ステップの場合) を開きます。

2. **[入力]** タブで、入力を選択するか、または新規作成します。
3. **[割り当て先]** ドロップダウンメニューで、値を割り当てる変数の名前を指定します。
4. 保存します。

入力の作成の詳細については、[「入力の作成」\(205ページ\)](#)を参照してください。

結果の値をフロー変数に割り当てる

1. **[プロパティ]** シート (オペレーションの場合) またはステップのインスペクター (ステップの場合) を開きます。
2. **[結果]** タブで、該当する結果の行を選択します。
3. **[割り当て先]** リストから **[Flow Variable]** を選択します。
4. **[名前]** で、フロー変数の名前を指定します。
5. **[ソース]** で、値のソースを指定します。

結果の作成の詳細については、[「出力と結果の作成」\(242ページ\)](#)を参照してください。

必要に応じて、厳密な結果を取得するには、結果に1つまたは複数のフィルターを作成します。[「出力および結果のフィルタリング」\(255ページ\)](#)を参照してください。

スクリプトレットの値をフロー変数に割り当てる

スクリプトレットで値を作成してフロー変数に割り当てることもできます。

スクリプトレットに、次の構文のコマンドを含めます。

```
scriptletContext.putLocal("<localflowvariablename>", <value>);
```

<value >には、スクリプトレット内で作成されるオブジェクトまたは変数を指定できます。

[フロー変数] ペインに情報を表示する

[フロー変数] ペインを使用すると、フロー変数へのデータの格納について、次のことを追跡できます。

- 必要なときにデータを使用できるように、フローでフロー変数を使用する仕組み
- フロー変数がデータを取得する場所

[フロー変数] ペインに、この情報がツリー構造で表示されます。[フロー変数] ペインには、現在のフローで使用されているすべてのフロー変数が表示され、各フロー変数の作成と使用が一覧表示されます。フロー内のフロー変数に行った変更は、[フロー変数] ペインに自動的に反映されます。

[フロー変数] ペインを開くには、Studio ウィンドウの右上にある **[フロー変数]** タブをクリックします。



名前	#	✓	✓
altpass	8	✓	✓
→ フロー入力	3	✓	✓
フロー入力 "altpass" として宣言	-	✓	
"altpass" の値は入力 "altpass" に割り当てることができます	-	✓	✓
入力 "altpass" の値を "altpass" に割り当て	-	✓	✓
ユーザープロンプトを使用しないステップ入力	5		
"altpass" の値は "Change Service Status" の入力 "password" に割り当てること...	-		
"altpass" の値は "Does Service Exist" の入力 "altpass" に割り当てることができ...	-		
入力 "altpass" の値を "Does Service Exist" の "altpass" に割り当て	-		
"altpass" の値は "ServiceStatus" の入力 "password" に割り当てることができま...	-		
入力 "password" の値を "ServiceStatus" の "altpass" に割り当て	-		
altuser	8	✓	✓
→ フロー入力	3	✓	✓

[フロー変数] ペインの情報をフィルタリングする

最も関心のあるフロー変数の使用に焦点を絞るため、ペインに表示するフロー変数の使用を選択できます。

1. [フロー変数] ツールバーの [フィルター] ボタン  をクリックして、フィルターボタンを表示します。
2. 表示されたボタンの行で、ボタンをクリックして、各フィルタータイプのオンとオフを切り替えます。各データソースタイプを切り替えると、そのタイプのデータソースが表示されたり、表示から削除されます。
 - フロー入力
 - ユーザープロンプトのステップ入力
 - ユーザープロンプトが設定されていないステップ入力
 - 結果
 - スクリプトレット

フロー変数のリストの参照先の入力を見つける

特定の使用を表示して、その使用を定義するエディターを開くには、[フロー変数] ペインで使用インスタンスを選択します。

- 使用がフロー入力の場合、[入力] タブにフローの [プロパティ] シートが開いて、関連する入力の入力エディターが開きます。

- 使用がステップ入力または結果の場合、選択したステップが含まれたフロー図が開きます。フロー図の下に、入力または結果のエディターが開きます。

ヒント: 使用のリストを上を移動するには、[前へ]  ボタンを使用し、使用のリストを下を移動するには、[次へ]  ボタンを使用します。

グローバル変数を表示する

フロー内のすべてのグローバル変数を表示するには、フローをデバッグします。グローバル変数(フロー変数も同様) および変数の現在の値が、コンテキストインスペクターに表示されます。

コンテキストインスペクターの詳細については、「[コンテンツの検査](#)」(311ページ)を参照してください。

グローバル変数を変更する

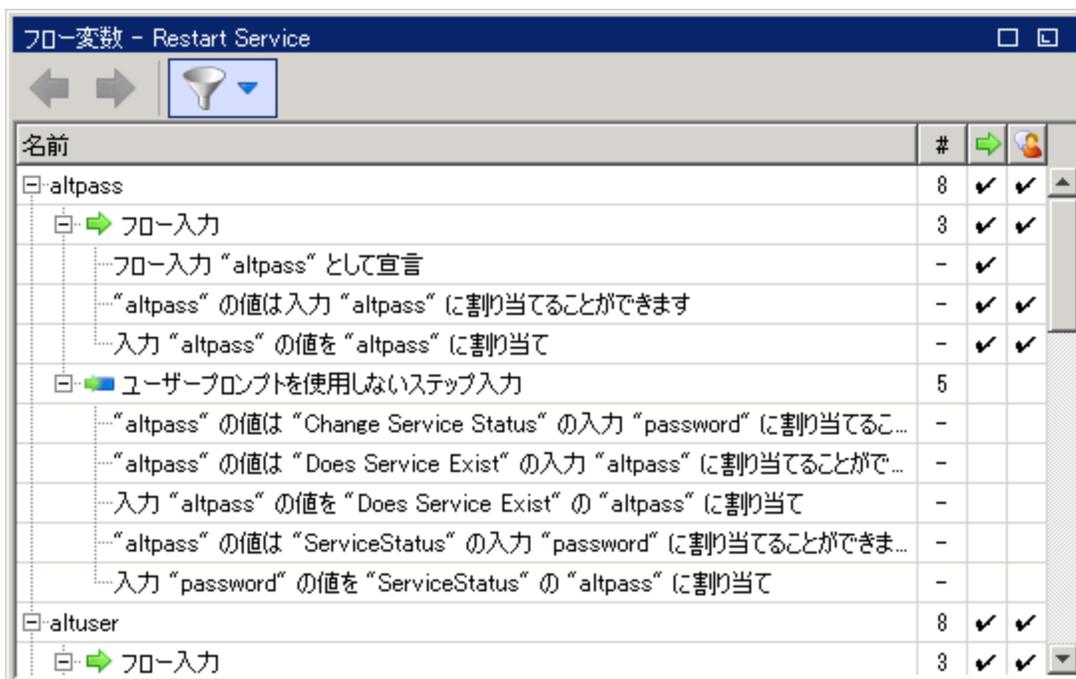
重要: グローバル変数の値を変更する前に、グローバル変数はすべてのフローのすべての実行に使用できることに留意してください。グローバル変数の値を変更すると、そのグローバル変数を使用するほかのフローとオペレーションに影響します。

グローバル変数を変更するには、タスク「入力の値をフロー変数に割り当てる」を完了します。[割り当て先変数] ボックスで、値を割り当てるグローバル変数の名前を入力します。

参考資料

[フロー変数] ペイン

作成ペインでフローを開いている場合、[フロー変数] ペインに各フロー変数がアルファベット順に表示され、フローの各使用の説明(フロー変数を使用できるフロー内の各場所)が表示されます。



GUI アイテム	説明
	クリックすると、使用のリスト内を上に移動します。
	クリックすると、使用のリスト内を下に移動します。
	クリックすると、フィルターボタンが表示されて、[フロー変数] ペインに表示されている情報をフィルタリングできるようになります。
	フロー内でこのフロー変数を使用した回数を表示します。
	フロー変数の特定の使用がフロー入力で発生した場合、チェックされます。
	フロー変数の特定の使用がユーザー入力から値を取得している場合、チェックされます。

[フロー変数] ペインのフィルターボタン

[フロー変数] ツールバーの [フィルター] ボタン をクリックして、フィルターボタンを表示します。

フィルターボタンをクリックして、各フィルタータイプのオンとオフを切り替えます。各データソースタイプを切り替えると、そのタイプのデータソースが表示されたり、表示から削除されます。

フィルターボタン	説明
	フロー入力 - フロー変数が入力内で参照されます。
	ユーザープロンプトを使用したステップ入力
	ユーザープロンプトを使用しないステップ入力
	結果 - フロー変数がステップ結果に関連付けられます。
	スクリプトレット - フロー変数がスクリプトレットで参照されます。

リターンステップの作成

フローには、フローを終了する1つ以上のリターンステップが必要です。



リターンステップの種類には次の4つがあります。

- **解決済み**  - 正常に実行されているフローの標準のリターンステップです。
- **診断済み**  - 何が問題なのかを特定して、通知以外の操作を実行しないことをフローが選択したことを示します。
- **操作なし**  - データを収集したが、診断か修復かを修復フローが決定できない場合に使用されます。

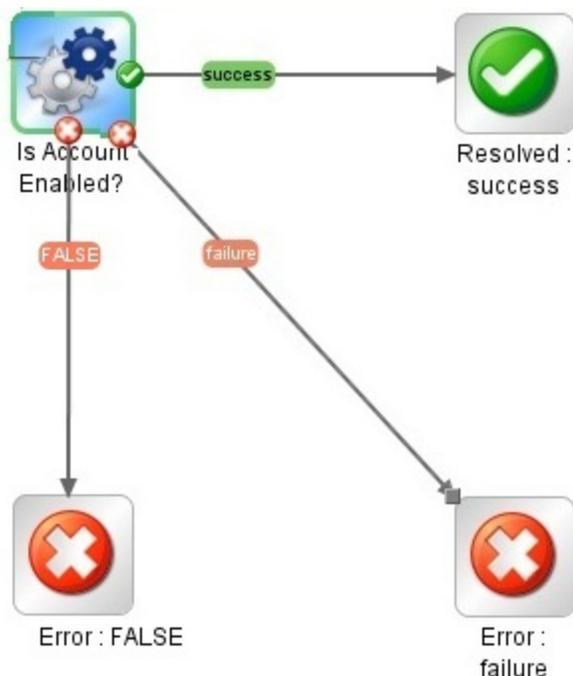
注: データ収集のみを目的としたフローでは、フローが完了すると、**解決済み**  が返され、**操作なし**  は返されません。

- **エラー**  - フローを最後まで実行できない場合に使用されます。たとえば、入力が正しくない場合や、システムに到達できない場合や、フローに問題がある場合に使用されます。

各リターンステップ名で、コロンの後にあるのがリターンステップのレスポンスです。たとえば、**エラー: 失敗** のようになります。このレスポンスを編集できます。たとえば、**エラー: 失敗** をもたらした結果が、オペレーションの失敗ではなく、要求されるしきい値に到達しない結果だった場合、この結果を反映する、**エラー: しきい値に到達していない** などの新しいレスポンスを**エラー: 失敗** ステップに作成できます。

ベストプラクティス

- フロー内に同じ種類の終了ステップが複数ある場合 (複数のエラー終了ステップなど)、失敗の原因を含めるように終了ステップの名前を変更します。
- オペレーションの失敗と否定の結果を混同しないようにします。たとえば、回答が TRUE または FALSE となる質問をオペレーションがした場合、回答 FALSE は失敗と同じことではありません。このような場合、結果が FALSE の場合とオペレーションの失敗の場合の、2つの**エラー**で結果を返す必要があります。



実行する操作

フローにリターンステップを追加する

1. 作成ペインのツールバーで、[ステップパレット] ボタン  をクリックして、ステップパレットを表示します。
2. ステップパレットから、使用するリターンステップのアイコンを作成キャンバスにドラッグします。
3. フローステップからリターンステップへのトランジションを作成します。

リターンステップのレスポンスを変更する

リターンステップのレスポンスは、そのリターンステップに至った結果をより具体的に表すように変更できます。たとえば、フローに複数のエラーレスポンス(「エラー: 失敗」や「エラー: しきい値に到達していない」)がある場合、[エラー]  アイコンを作成キャンバスにドラッグしたときに、必要なレスポンスがエラーリターンステップに含まれない可能性があります。

1. 作成ペインでリターンステップを右クリックし、[レスポンスの選択] を選択します。
2. リターンステップに必要なレスポンスを選択します。たとえば、「エラー: しきい値に到達していない」と入力します。

新しいレスポンスを作成してリターンステップに割り当てる

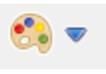
使用可能なレスポンスのリストに必要なレスポンスが含まれていない場合、カスタムのレスポンスを作成できます。

1. 作成ペインでリターンステップを右クリックし、[レスポンスの選択] を選択します。
2. [新規レスポンスの追加] を選択します。
3. ダイアログボックスで、新しいレスポンスの名前を入力し、[OK] をクリックします。

参考資料

ステップパレット

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

テップパレット] ボタン  をクリックして、ステップパレットを表示します。



ボタン	説明
成功 	[解決済み] リターンステップをフローにドラッグできます。
診断済み 	[診断済み] リターンステップをフローにドラッグできます。
操作なし 	[操作なし] リターンステップをフローにドラッグできます。
失敗 	[エラー] リターンステップをフローにドラッグできます。
並列分岐ステップ 	[並列分岐ステップ] をフローにドラッグできます。
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] をフローにドラッグできます。
コールアウト 	[コールアウト] をフローにドラッグして、ユーザーに情報を提供できます。
ドッキングバー 	クリックして、パレットのドッキングとドッキング解除を行います。

高度な作成

この章では、より複雑なフローの作成について説明します。単純なフローの作成については、「[フロー作成 - 基本](#)」(185ページ)を参照してください。

フローの作成時、メモリを無制限に増大させるようなフローは作成しないでください。フローが休止し、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は作成しないでください。この場合、システムでメモリがなくなるまで実行履歴が増大します。

フロー内でのサブフローの作成

サブフローからステップを作成して、フローを単純化できます。次のことが可能です。

- プログラミングタスクを、管理しやすいように小さく分割する
- フローの部分ごとにテストを行う
- 作成したステップを再使用する

たとえば、以下のフローにおいて **Windows Health Check**のステップはサブフローです。



サブフローは、複数のオペレーションが含まれている場合でも1つのステップとして処理されます。

サブフローでは主に、親フローのステップがアクセスする必要があるデータを生成します。フロー内で作成したフロー変数は、そのフロー外では参照できません。ただし、サブフローの結果を**フロー出力フィールド**として保存することにより、サブフローから親フローに値を渡すことが可能です。

ベストプラクティス

- フローは、Studio の最大の大きさである 1024 x 768 の画面上のキャンバスに、1:1 の表示倍率に合わせて作成します。これより大きいフローにすることも不可能ではありませんが、フローが大きくなる場合、そのステップのシーケンスの一部をサブフローに分割できないかをよく確認してください。
- 最上位の親フロー内のすべてのトランジションに説明と名前を付けます。これらのトランジションの説明は、トランジションの先行ステップで発生する内容について説明します。データが実行中に必ず表示されなければならない場合を除き、サブフローのトランジションの説明を追加する必要はありません。

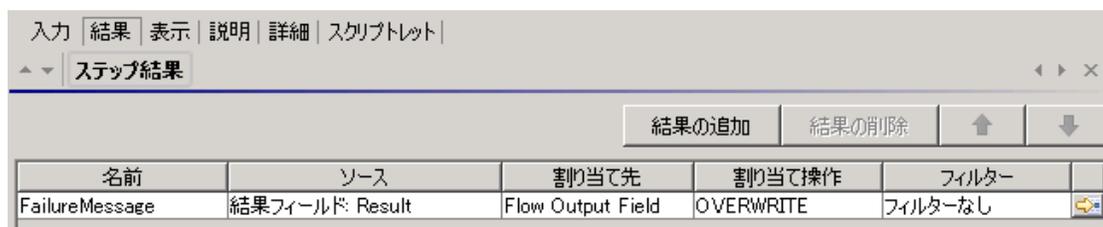
実行する操作

サブフローのあるフローの作成

1. まずフローを作成し、保存します。
2. 親フローとして機能する新しいフローを作成します。
3. [プロジェクト] ペインからこのサブフローを親フローにドラッグし、サブフローからステップを作成します。

サブフローから親フローへのデータの引き渡し

1. 作成キャンバスでサブフローを開き、親フローでデータを使用可能にするステップのインスペクターを開きます。
2. [結果] タブをクリックし、結果を追加します (詳細については、[「ステップの結果の設定」](#) (246 ページ) を参照してください)。
3. 結果データが **フロー出力フィールド** に保存されるように結果を設定します。これによって、サブフロー外でデータが使用可能になります。



- a. [名前] で、フロー出力フィールドの名前を入力します。
- b. [ソース] で、[結果フィールド:Result] を選択します。
- c. [割り当て先] で、[Flow Output Field] を選択します。
- d. 必要に応じてフィルターを作成し、結果をフィルター処理します (詳細については、[「出力および結果のフィルタリング」](#) (255 ページ) を参照してください)。

4. 親フローの作成キャンバスで、サブフローから作成したステップのインスペクターを開きます。
5. **[結果]** タブをクリックし、ステップの結果を作成します。デフォルトでは、この新しい結果により以下が行われます。
 - サブフローのフロー出力フィールドの名前が付いた結果フィールドから値を取得する
 - サブフローのフロー出力フィールドと同じ名前を持つ
 - デフォルトで結果と同じ名前を持ち、このステップに続くトランジションとステップで使用可能になった、フロー変数に割り当てられる

例

1. コマンド操作をコピーし、"dir C:\" を実行するようにします。これに **dir** という名前を付けます。
2. **flowdir** という名前のフローを作成します。
3. **flowdir** フローで、オペレーション **dir** を使用してステップを作成します。
4. **dir** ステップで、オペレーションの出力文字列から得られる結果を追加します。
5. 結果をフロー出力フィールドに割り当て、結果に **foo** という名前を付けます。これで、フローにも **foo** という名前のフロー出力フィールドが追加されました。
6. **parentflow** という名前の別のフローを作成します。
7. **parentflow** で、**flowdir** からステップを作成します。
8. **flowdir** ステップに結果を追加します。

新しい結果の名前は、デフォルトでは **foo** です。結果は **[結果フィールド: foo]** から値を取得し、この値は **foo** という名前のフロー変数にも割り当てられます。これで、サブフローのステップ **dir** の結果 **foo** は、親フローの **flowdir** ステップの後に続くトランジションとステップで使用可能になります。

9. これをテストするには、**flowdir** ステップの後に**基本通知**オペレーションから作成されたステップを追加します。
10. この新しいステップで、以下を行います。
 - a. 入力 **notifyData** を定数を使用する単一値として定義し、定数に **`\${foo}** を指定します。
 - b. 入力 **notifyMethod** を定数を使用する単一値として定義し、定数に **Display** を指定します。
 - c. 入力 **subject** を定数を使用する単一値として定義し、定数を次のように指定します。正常に動作した場合、フロー出力フィールドに「**foo** という **outputString** の内容」と表示。
 - d. フローをデバッグします。

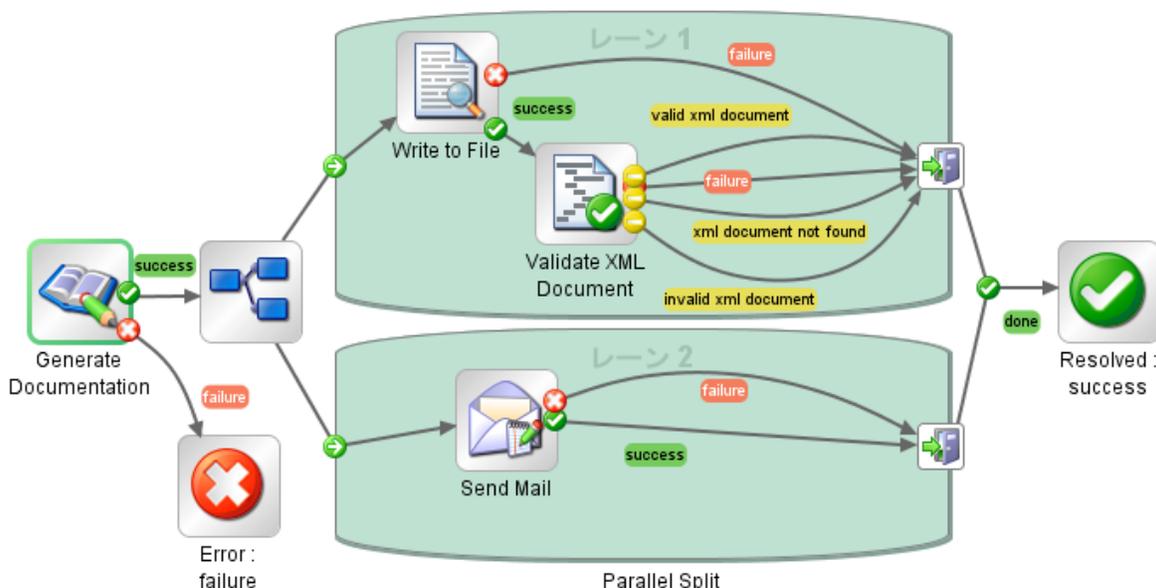
並列分岐ステップのフローの作成

並列分岐ステップは、同時に実行される一連の連続したステップです。各連続ステップは、フロー図内にレーンとして視覚的に表されます。各レーンに含まれるステップを「レーンステップ」と呼びます。フローを実行すると、レーンが同時に開始されます。

並列分岐ステップは、それぞれに独立した異なる処理を同時に実行する場合に最適です。これに対して、マルチインスタンスステップでは、複数の異なる単一入力を使ってインスタンスが同じ処理を実行します。

たとえば、並列分岐ステップを使用すると、XML ファイルの記述と検証を実行すると同時に、担当者
にこの記述と検証について電子メールを送信できます。

- 1 番目のレーンに、ファイルの記述と検証のステップが含まれます。
- 2 番目のレーンによって電子メールが送信されます。



注: 並列分岐ステップは非ブロックにできません。

並列レーンにレスポンスステップを含めることはできません。

実行する操作

並列分岐ステップを作成する

1. 作成ペインのツールバーで、[ステップパレット] ボタン  をクリックして、ステップパ

レットを表示します。

2. ステップパレットから、**[並列分岐ステップ]**  アイコンを作成キャンバスにドラッグします。デフォルトでは、ステップには2つのレーンがあります。
3. 各レーンに含めるステップシーケンスを作成します。
 - a. レーンにステップ(フローまたはオペレーション)を追加します。

注: レスponseステップを並列レーンに追加することはできません。

- b. 各レーンのステップを接続します。
 - c. レーン内の最後のステップを**[レーンエンド]**アイコン  にドラッグします。
4. 並列分岐ステップを残りのフローに接続します。
 - a. 並列分岐ステップが開始ステップではない場合、この並列分岐ステップに先行するステップを**[並列分岐ステップ]**  アイコンに接続します。
 - b. 並列分岐ステップの**[完了]**  レスponseをフローの次のステップに接続します。

レーンの表示順序を変更する

フロー図内のレーンの表示順序を変更できますが、フローを実行すると、すべてのレーンが同時に開始されます。フロー図内の表示順序は、処理の順序には影響しません。

1. 移動するレーンを右クリックします。
2. ドロップダウンメニューから、**[レーンを上に移動]**または**[レーンを下に移動]**を選択します。

並列分岐ステップまたはコンポーネントを移動する

- 並列分岐ステップを移動するには、フロー図内の**[並列分岐ステップ]**  アイコンをクリックしてドラッグします。
- 個々のレーンステップを移動するには、ステップを選択し、レーン内または別のレーンにドラッグします。

並列分岐ステップをコピーする

1. フロー図内で**[並列分岐ステップ]**  アイコンを右クリックし、**[コピー]**を選択します。
2. 作成キャンバスを右クリックし、**[貼り付け]**を選択します。

並列分岐ステップのコンポーネントをコピーする

並列分岐ステップのコンポーネントをコピーするには、次のいずれかのツールを使用します。

- **[編集]** > **[コピー]** メニューコマンドと **[編集]** > **[貼り付け]** メニューコマンド
- 右クリックメニュー

- キーボードの組み合わせ Ctrl+C、Ctrl+V
- 作成ペインツールバーの [コピー]  ボタンと [貼り付け]  ボタン

注: レーンをコピーする場合、[貼り付け] コマンドを実行するときに、カーソルをレーン内に置きます。

レーンを追加する

1. 既存のレーンを右クリックします。
2. ドロップダウンメニューから、[レーンの追加] を選択します。

現在選択しているレーンの下に、新しい空のレーンが追加されます。

レーンを複製する

1. 既存のレーンを右クリックします。
2. ドロップダウンメニューから、[レーンの複製] を選択します。

コピーしたレーンのすぐ下に、同じタイトルを持つ新しいレーンが表示されます。

レーンを削除する

レーンを削除するには、次のいずれかのツールを使用します。

- [編集] > [レーンの削除] メニューコマンド
- 右クリックメニュー
- キーボードの組み合わせ Ctrl+X
- 作成ペインツールバーの [削除]  ボタン

レーンのサイズを変更する

1. レーンの空白部分をクリックして選択します。辺と角にハンドルが表示されます。
2. 辺または角のハンドルをドラッグします。

レーンの名前を変更する

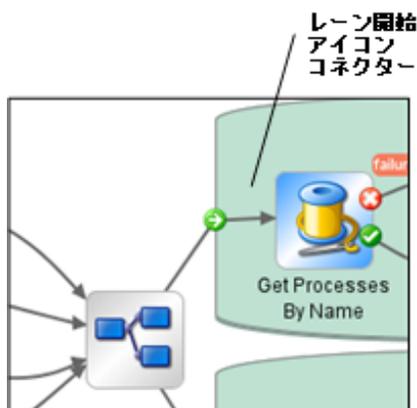
デフォルトでは、レーンの名前は **レーン 1**、**レーン 2** などとなります。

1. レーンを右クリックして [名前の変更] を選択します。
2. 表示されたテキストボックスに、レーンの新しい名前を入力します。

レーンの開始ステップを変更する

レーンの開始ステップには、フローの開始ステップのような緑色の枠がありません。

[レーン開始] アイコン  コネクターを、現在の対象のレーンステップから、レーンの開始ステップにするステップにドラッグします。



並列分岐ステップとの間でデータを移動する

並列分岐ステップが開始すると、ステップの各レーンはグローバルコンテキストのフロー変数のコピー、ローカルコンテキストの変数のコピー、および並列分岐ステップ自体の入力を取得します。各レーンでは、通常のプロールールに従って、ほかのレーンとは関係なく、これらの変数を使用したり、これらの変数を作成、変更、または削除できます。

1つのレーン内のステップから別のレーン内のステップに値を渡すことはできません。各レーン内のステップには、並列分岐ステップ(およびステップのレーン)が開始されたときに利用できるようになった値のみ含まれています。

レーンで実行が終了すると、各コンテキストのフロー変数が、呼び出し元のフロー(並列分岐ステップが含まれているフロー)のコンテキストに戻されてマージされます。マージの順序は、レーンが終了した順序です。このため、2つのレーンが同じフロー変数に書き込む場合、後に終了したレーンによって変数の最後の値が書き込まれます。

並列分岐ステップのレーン内のステップでは、ローカルコンテキストとグローバルコンテキストからデータを取得して、ローカルコンテキストに保存できます。レーンステップは、スクリプトレットで `scriptletcontext.putGlobal()` メソッドを使用する場合のみ、グローバルコンテキストに書き込みできます。`scriptletcontext.putGlobal()` を使用する構文については、オペレーションまたはステップの[スクリプトレット]タブで、**JavaScript** テンプレートを挿入します。

並列分岐ステップをデバッグする

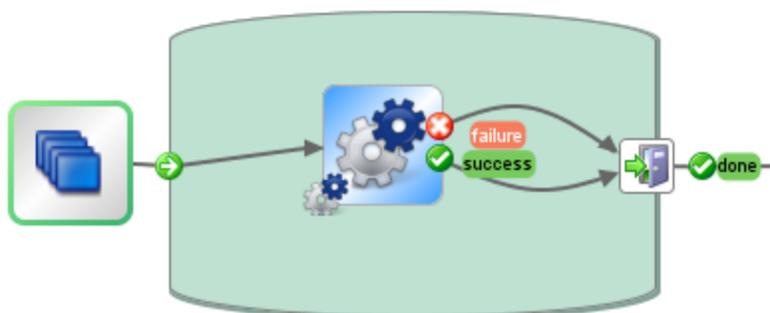
実際の実行時には、フローを実行するとレーンは同時に開始されて実行されますが、デバッガーでテストすると連続して実行されます。デバッガーでのレーンの実行順序は制御できませんが、一意の名前を指定すると、実行された順序を確認できます。

この方法では、デバッガーで実稼働環境でのフローの動作が正確には再現されませんが、デバッガーで並列分岐ステップを連続して実行すると、さまざまな条件を制御しながらテストできます。詳細については、「[複雑なフローのデバッグ](#)」(326ページ)を参照してください。

マルチインスタンスステップのフローの作成

マルチインスタンスステップは、複数のターゲットで同時に実行されるステップです。たとえば、100 台のサーバーで Windows 診断フローを実行する場合、100 台のすべてのサーバーで同時にフローを実行する、マルチインスタンスステップを作成できます。

オペレーションのターゲット(この例では 100 台のサーバー)は、マルチインスタンスステップの入力リストで定義されます。



マルチインスタンスステップ内に、1 つまたは複数のオペレーションまたはサブフローを含めることができます。マルチインスタンスステップ内のオペレーションまたはサブフローは、ターゲットごとに 1 回実行されます。この実行をインスタンスと言います。

各インスタンスは、インスタンスの開始時に、グローバルコンテキストとローカルコンテキストの複製を取得します。実行時に、インスタンスのステップごとに、マルチインスタンスステップ内のグローバル変数、フロー変数、およびフロー出力フィールドを変更できます。

注: いずれかのインスタンスで例外が発生した場合、そのインスタンスは停止されます。ほかのインスタンスは並列で実行されているため、引き続き実行されます。

注: マルチインスタンスステップは非ブロックにできません。

複数のインスタンスレーンにレスポンスステップを含めることはできません。

マルチインスタンスステップと並列分岐ステップの相違点

マルチインスタンスステップでは、各インスタンスが別々のターゲットで同じタスクを実行しますが、並列分岐ステップでは、異なるタスクを実行するように各並列ステップを設定できます。

マルチインスタンスステップでは、実行時にインスタンス数が変化する可能性があります。並列分岐ステップでは、並列ステップの数は一定です。

フローデータの保存

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、グローバル変数、およびフロー出力フィールドは、作成および割り当て元のインスタンスに対してローカルです。これらの変数とフロー出力フィールドの変数は、以下のいずれかの方法でこのデータを残りのフローで使用できるようにしない場合、レーンの終了時に消去されます。

- マルチインスタンスステップの結果にデータをバインドする
- データを保存するように、マルチインスタンスステップにスクリプトレットを作成する

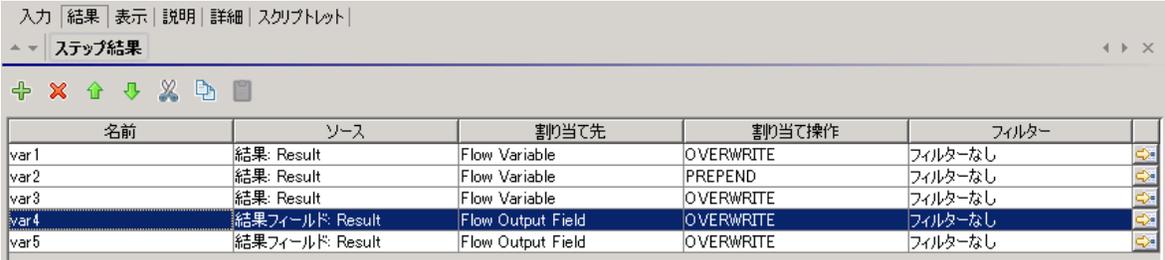
結果を使用したデータの保存

マルチインスタンスステップの終了後にフロー変数のデータを使用できるようにするため、ステップの結果をマルチインスタンスステップに定義して、インスタンスで作成されたフロー変数から値を取得できます。マルチインスタンスステップのステップのインスペクターの[結果]タブで、[ソース]列の[結果 <result>]を選択して、インスタンスで作成されたフロー変数を選択します。

同様の方法で、インスタンスのサブフローで作成されたフロー出力フィールドのデータを保存することもできます。マルチインスタンスステップのステップのインスペクターの[結果]タブで、[ソース]列の[結果フィールド <result>]を選択して、インスタンスで作成されたフロー出力フィールドを選択します。

[割り当て操作] フィールドを設定し、収集した値を使って別の処理を実行できます。たとえば、異なるインスタンスの結果を追加したり、異なるインスタンスの結果をすべて足し合わせたり、より新しいインスタンスで古いインスタンスを上書きしたりできます。

以下の例では、マルチインスタンスステップの結果に5つの変数が設定されています。最初の3つの変数はフロー変数から値を取得し、残りの2つの変数はフロー出力フィールドから値を取得します。



名前	ソース	割り当て先	割り当て操作	フィルター
var1	結果: Result	Flow Variable	OVERWRITE	フィルターなし
var2	結果: Result	Flow Variable	PREPEND	フィルターなし
var3	結果: Result	Flow Variable	OVERWRITE	フィルターなし
var4	結果フィールド: Result	Flow Output Field	OVERWRITE	フィルターなし
var5	結果フィールド: Result	Flow Output Field	OVERWRITE	フィルターなし

Instance1 と **Instance2** の2つのインスタンスが存在し、メインフローのコンテキストは空であり、**Instance2** は **Instance1** の後で終了するとします。これらのインスタンスで以下の変数を指定するとします。

- **Instance1**
 - フロー変数
 - var1 = x
 - var2 = y

```
var3 = w
```

- フロー出力フィールド

```
var1 = z
```

- **Instance2**

- フロー変数

```
var2 = t
```

```
var3 = v
```

- フロー出力フィールド

```
var5 = u
```

マルチインスタンスステップが終了したときに、変数の値は次のようになります。

var1 = NULL (**Instance2** では、この変数の値は存在せず、アクションが上書きされるため)

var2 = t (**Instance2** の値によって **Instance1** の値が上書きされます)

var3 = wv (**Instance2** の値が **Instance1** の値に追加されています)

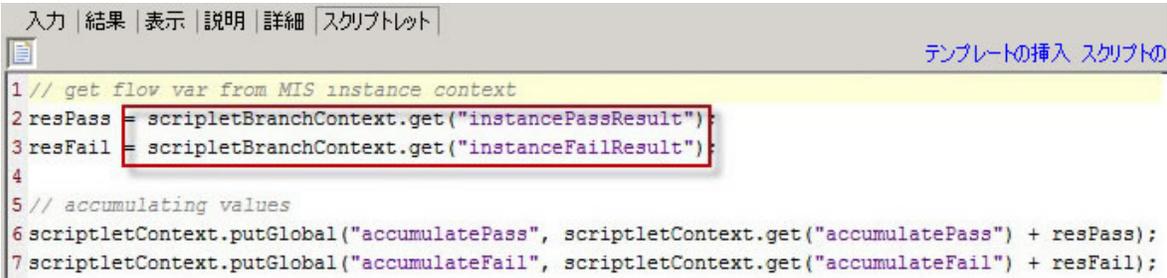
var4 = NULL (**Instance2** では、この変数の値は存在せず、アクションが上書きされるため)

var5 = u

スクリプトレットを使用したデータの保存

ステップで生成したデータを残りのフローで使用できるようにするには、データを収集して、インスタンスの実行が完了した後も引き続き存在する変数として保存する、スクリプトレットを作成するという方法もあります。

ここに示す例では、スクリプトレットを使用して、実行された各インスタンスが成功したのか失敗したのかを追跡し、このデータを累計し、グローバルコンテキストで使用可能な変数として保存しています。



```
1 // get flow var from MIS instance context
2 resPass = scripletBranchContext.get("instancePassResult");
3 resFail = scripletBranchContext.get("instanceFailResult");
4
5 // accumulating values
6 scripletContext.putGlobal("accumulatePass", scripletContext.get("accumulatePass") + resPass);
7 scripletContext.putGlobal("accumulateFail", scripletContext.get("accumulateFail") + resFail);
```

このスクリプトレットは、インスタンスごとに1回ずつ、複数回実行されます。毎回、現在のインスタンス (scriptletBranchContext) の ScriptletContext にアクセスでき、scriptletContext にアクセスすることで親フローコンテキストを変更できます。

scriptletBranchContext は scriptletContext と同じアクセスメソッドを備えています。

スクリプトレットの詳細については、「[フロー内でのスクリプトレットの使用](#)」(297ページ)を参照してください。

アップグレード後のマージ

前のバージョンの HP 00 からアップグレードすると、[複数インスタンスの切り替え] オプションを使って作成されたマルチインスタンスステップがフローに含まれている場合、古いインスタンスを上書きする新しいインスタンスによって、ステップで作成されたグローバル変数が更新されます。

実行する操作

マルチインスタンスステップを作成する

1. 作成ペインのツールバーで、[ステップパレット] ボタン  をクリックして、ステップパレットを表示します。

2. ステップパレットから、[複数のインスタンス]  アイコンを作成キャンバスにドラッグします。

3. [プロジェクト] ペインから、フローまたはオペレーションを複数のインスタンスレーンにドラッグします。

注: 複数のフローおよびオペレーションを複数のインスタンスレーンに追加できます。

レスポンスステップを複数のインスタンスレーンに追加することはできません。

4. 入力を複数の値のリストとして作成して、マルチインスタンスステップのターゲットのリストを設定します。たとえば、フローを実行するサーバーのリストの場合、次の操作を実行します。

- a. ステップの開始時に [複数のインスタンス]  アイコンをダブルクリックし、マルチインスタンスステップのステップのインスペクターを開きます。

- b. 入力を作成します。この例では、入力の名前は **servers** です。

- c. [必須] チェックボックスをオンにして、種類を [値のリスト] に設定します。



入力	必須	種類	ソース
servers	<input checked="" type="checkbox"/>	値のリスト	値: server 1,server 2,server 3

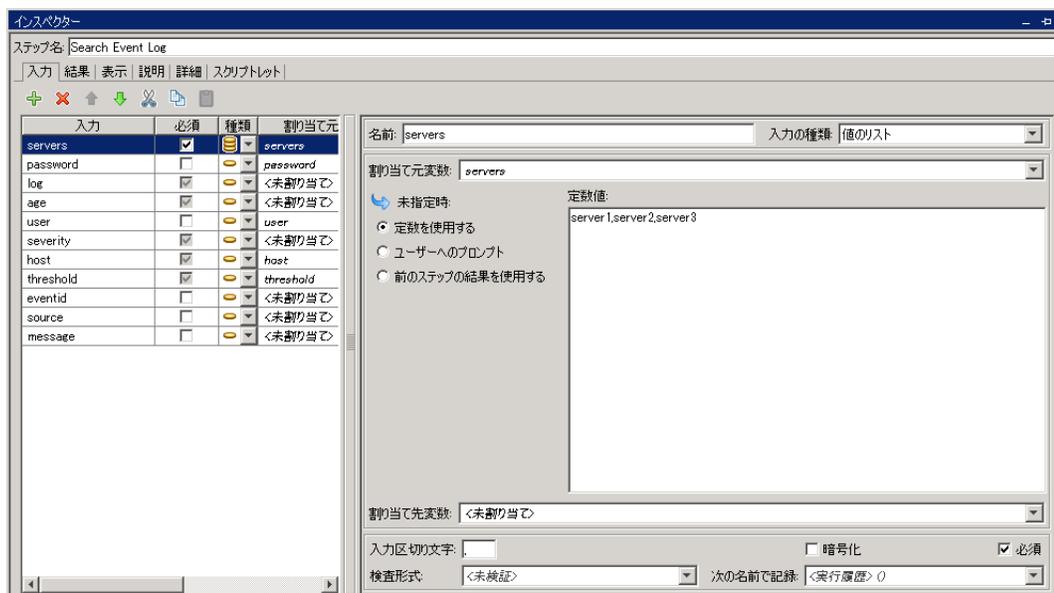
- d. 行の端にある右向き矢印  をクリックして、その行の入力エディターを開きます。

- e. [入力区切り記号] ボックスに、区切り記号 (リスト内の要素を区切る文字 (複数可)) を入力します。

注:

- 複数の文字が含まれる区切り記号を定義するには、疑問符を使用します。例:
"\$%^".
- 区切り記号の一部として疑問符を使用するには、疑問符を2回入力します。たとえば、区切り記号 "\$%" を定義するには、「"\$%^%" と入力します。

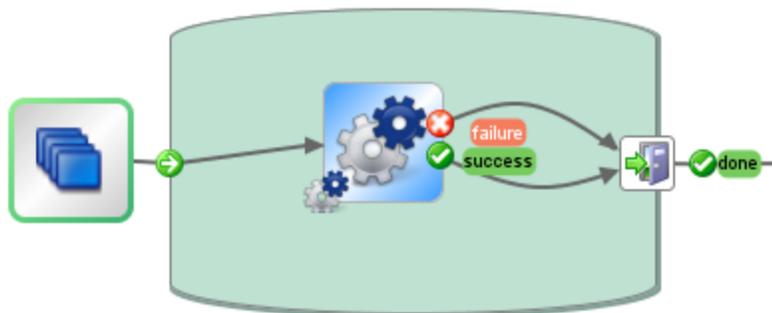
- f. 値のリストを入力する方法を指定します。たとえば、マルチインスタンスステップを多数のサーバーで実行するには、**[定数を使用する]**を選択し、**[定数値]**ボックスでサーバー名を指定します。値のリストを作成する方法には、前のステップの結果を使用する方法や、別のプログラムと統合する方法もあります。



入力の値のリストを作成するオプションの詳細については、「[入力ソースの指定](#)」(214ページ)を参照してください。

5. マルチインスタンスステップの複数の異なる部分を接続します。
- [**レーン開始**] アイコン  を複数のインスタンスレーンの最初のステップに接続します。
 - マルチインスタンスステップに複数のステップが含まれている場合は、これらのステップを接続します。
 - レーン内の最後のステップのすべてのレスポンス行を [**レーン - エンド**] アイコン  を接続します。

にドラッグします。



6. ターゲットのリストを各内部ステップに適用します。
 - a. 複数のインスタンスレーン内の内部ステップごとに、ステップのインスペクターを開いて入力を追加します。
 - b. 入力エディターを開いて、[割り当て元] リストで、ターゲットのリストを保持するために作成した変数を選択します。この例では、この変数は **servers** です。
7. マルチインスタンスステップのさまざまなインスタンスで収集されたデータを保存するには、結果を保存するフロー変数を作成します。

- a. [マルチステップ]  アイコンをダブルクリックし、マルチインスタンスステップのステップのインスペクターを開きます。
- b. [結果] タブをクリックして結果を追加します。
- c. [割り当て先] 列で、結果をフロー変数に割り当てます。
- d. データを保持するフロー変数に名前を指定します。たとえば、**outcome** と指定します。
- e. データをどのように保存するのかを決定します。この例では、各サーバーの結果を保存するため、割り当て操作は **APPEND** になっています。詳細については、以下の「マルチインスタンスステップの出力を保存する」を参照してください。

名前	ソース	割り当て先	割り当て操作	フィルター
outcome	結果: outcome	Flow Variable	APPEND	フィルターなし

8. グローバルコンテキストで作成できるよう、マルチインスタンスステップのさまざまなインスタンスで収集されたデータを保存するには、結果を保存するスクリプトレットを作成します。
 - a. マルチインスタンスステップのステップのインスペクターで、[スクリプトレット] タブをクリックします。
 - b. `scriptletBranchContext` のデータを収集するスクリプトレットを作成して、`scriptletContext` で使用できるようにします。

次の例では、スクリプトレットを使用して、**outcome** 変数のすべての値を累計するようにフローに指示しています。これは、前の手順で選択した **APPEND** アクションと似ています。

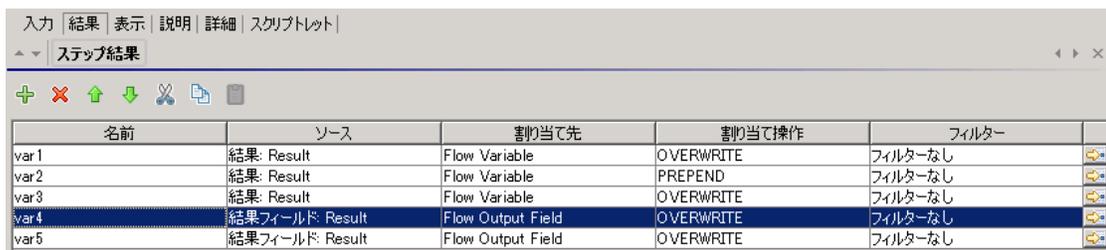
```
入力 | 結果 | 表示 | 説明 | 詳細 | スクリプトレット | テンプレートの挿入 | スクリ  
1 var outcome = scriptletBranchContext.get("outcome");  
2  
3 scriptletContext.putGlobal("outcome",scriptletContext.get("outcome") + outcome);  
4
```

9. マルチインスタンスステップを残りのフローに接続します。
 - a. マルチインスタンスステップが開始ステップではない場合、このマルチインスタンスステップに先行するステップを [複数のインスタンス]  アイコンに接続します。
 - b. マルチインスタンスステップの [完了]  レスポンスをフローの次のステップに接続します。

マルチインスタンスステップの出力を保存する

インスタンスのフロー変数とフロー出力フィールドのデータは、マルチインスタンスステップが完了すると消去されます。このデータを保存するために、データをマルチインスタンスステップの結果にバインドできます。

1. 前述の手順に従って、マルチインスタンスステップを作成します。
2. ステップのインスペクターで [結果] タブをクリックします。
3. 保存する各フロー変数の結果行を追加します。



名前	ソース	割り当て先	割り当て操作	フィルター
var 1	結果: Result	Flow Variable	OVERWRITE	フィルターなし
var 2	結果: Result	Flow Variable	PREPEND	フィルターなし
var 3	結果: Result	Flow Variable	OVERWRITE	フィルターなし
var 4	結果フィールド: Result	Flow Output Field	OVERWRITE	フィルターなし
var 5	結果フィールド: Result	Flow Output Field	OVERWRITE	フィルターなし

4. [名前] 列に、データを保存するフロー変数の名前を入力します。
5. [ソース] 列で、保存するデータのソースであるフロー変数またはフロー出力フィールドを選択します。
 - インスタンスで作成されたフロー変数を選択するには、[ソース] 列で [結果フィールド: <result>] を選択します。
 - インスタンスで作成されたフロー出力フィールドを選択するには、[ソース] 列で [結果フィールド <result>] を選択します。
6. [割り当て操作] 列で、どのようにデータを収集するのかを表すアクションを選択します。

たとえば、すべてのインスタンスを実行するのにかかる時間を計算する場合、[ADD]を選択します。マルチインスタンスステップでチェックされたすべてのサーバーのリストを収集する場合は、[APPEND]を選択します。

7. ステップを保存します。マルチインスタンスステップの実行が終了した後で、作成したフロー変数を残りのフローに使用できるようになります。

マルチインスタンスステップの出力をグローバル変数として保存する

マルチインスタンスステップの出力を保存してフローの外部で使用できるようにするため、この出力をグローバル変数として保存するスクリプトレットを作成できます。

1. ステップの開始時に [複数のインスタンス]  アイコンをダブルクリックし、マルチインスタンスステップのステップのインスペクターを開きます。
2. [スクリプトレット] タブをクリックします。
3. scriptletBranchContext のデータを収集するスクリプトレットを作成して、scriptletContext で使用できるようにします。例:

```
入力 | 結果 | 表示 | 説明 | 詳細 | スクリプトレット
テンプレートの挿入 | スクリプトの
1 // get flow var from MIS instance context
2 resPass = scriptletBranchContext.get("instancePassResult");
3 resFail = scriptletBranchContext.get("instanceFailResult");
4
5 // accumulating values
6 scriptletContext.putGlobal("accumulatePass", scriptletContext.get("accumulatePass") + resPass);
7 scriptletContext.putGlobal("accumulateFail", scriptletContext.get("accumulateFail") + resFail);
```

マルチインスタンスステップで同時に実行できるインスタンスの数を制限する(絞る)

マルチインスタンスステップを多くのインスタンスで同時に実行すると、システムのパフォーマンスが低下することがあります。マルチインスタンスステップで同時にフローを開始できるターゲットの数を制限することでステップの絞りレベルを設定できます。

パフォーマンスは、ターゲットシステム、およびフローですべての 00 のワーカースレッドを使用できるかどうかによって左右されます。

絞りが設定されていない場合、インスタンスの数は入力の数と同じになります。

絞りが設定されている場合、インスタンスの数は絞りのサイズと残りの入力のうちの最小数になります。

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、[並列実行を制限された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オンにします。
3. ボックスに、同時に実行されるステップのインスタンスの最大数を入力します。

マルチインスタンスステップを移動する

1. レーンの開始時に、ステップ全体を表す [複数のインスタンス]  アイコンを選択します。
2. 作成キャンバスでステップをドラッグします。

マルチインスタンスステップのサイズを変更する

1. レーンの空白部分をクリックして選択します。辺と角にハンドルが表示されます。
2. 辺または角のハンドルをドラッグしてレーンのサイズを変更します。

マルチインスタンスステップの名前を変更する

1. レーンの開始時に、[複数のインスタンス]  アイコンを選択します。
2. 右クリックして [名前の変更] を選択します。
3. テキストボックスに新しい名前を入力します。

マルチインスタンスステップをデバッグする

実際の実行時には、複数のインスタンスは同時に実行されますが、デバッガーでテストすると連続して実行されます。つまり、実際の条件下でテストが行われなかったということですが、各インスタンスが終了するまでにかかる時間を調べることができます。

詳細については、「[複雑なフローのデバッグ](#)」(326ページ)を参照してください。

フロー内でのスクリプトレットの使用

スクリプトレット (Nashorn または Rhino JavaScript で記述) は必要に応じてオペレーションに追加します。オペレーションの入力データまたは結果データのいずれかを、オペレーションまたはフローのほかの部分で使用できるように加工する目的で使用します。

スクリプトレットを使用すると、結果の特定のデータをテスト、フォーマット、加工、または抽出できます。

スクリプトレットを使用すると、次のことを実行できます。

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

注: スクリプトレットでシステムプロパティを参照する場合は、完全なパスを使用する必要があります。たとえば、システムプロパティが `folderA\folderB\my_ci` というフォルダー構造の下にある場合、`${folderA/folder/my_ci}` という文字列を使用して参照します。

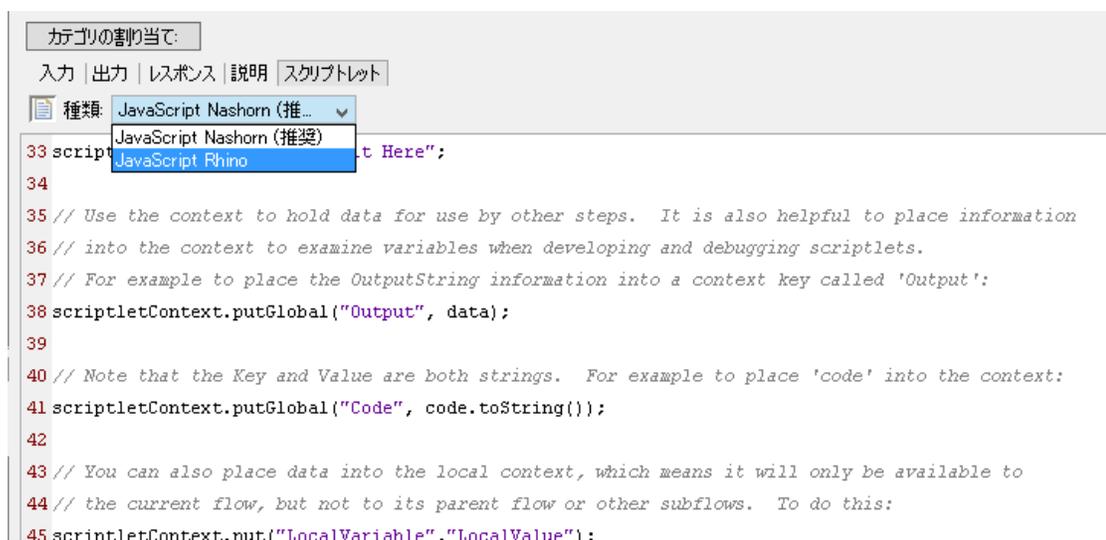
スクリプトレットを書くのに役立つリソース

- スクリプトレットテンプレート (Nashorn または Rhino JavaScript) がスクリプトレットエディターに用意されています。
- デフォルトのスクリプトレットが **Configuration\Scriptlets** フォルダーに用意されています。
- デフォルトコンテンツの既存のスクリプトレットをコピーできます。

実行する操作

テンプレートからスクリプトレットを作成する

1. [プロパティ]シートまたはステップのインスペクターを開きます。
 - オペレーションにスクリプトレットを追加するには、[プロジェクト]ペインでオペレーションを右クリックし、[プロパティ]を選択します。
 - フローにスクリプトレットを追加するには、[プロジェクト]ペインでフローを右クリックし、[プロパティ]を選択します。
 - ステップにスクリプトレットを追加するには、作成ペインでステップをダブルクリックします。
2. [スクリプトレット]タブを選択します。
3. [種類]リストで、スクリプトの記述に使用する JavaScript 言語を選択します。Nashorn (推奨) または Rhino を選択します。



4. [テンプレートの挿入]をクリックします。
5. テンプレートのガイドラインに従って、スクリプトを作成します。
6. [スクリプトの確認]をクリックしてエラーがないか確認します。

7. 保存します。

既存のスク립トレットを使用する

1. [プロパティ]シートまたはステップのインスペクターを開きます。
 - オペレーションにスク립トレットを追加するには、[プロジェクト]ペインでオペレーションを右クリックし、[プロパティ]を選択します。
 - フローにスク립トレットを追加するには、[プロジェクト]ペインでフローを右クリックし、[プロパティ]を選択します。
 - ステップにスク립トレットを追加するには、作成ペインでステップをダブルクリックします。
2. [スク립トレット]タブを選択します。
3. 既存のスク립トレットを別のウィンドウで開きます。
 - **Configuration\Scriptlets** フォルダーのスク립トレットをダブルクリックします。
 - スクリプトレットが含まれているオペレーションを開きます (たとえば、**Operations\Operating Systems\Linux\Red Hat** フォルダーのオペレーション)。
4. スクリプトレットのテキストをコピーして、オペレーション、フロー、またはステップの[スク립トレット]テキストボックスに貼り付けます。
5. 必要に応じて、スク립トレットを変更します。
6. [スクリプトの確認] をクリックしてエラーがないか確認します。
7. 保存します。

スク립トレットでステップまたはフローの結果をフィルタリングする

スク립トレットを使用して、ステップまたはフローの結果をフィルタリングできます。

1. 作成ペインでステップをダブルクリックします。
2. [結果]タブを選択し、フィルタリングする結果を選択します。
3. 結果の行の端にある右向き矢印  をクリックして、フィルターエディターを開きます。
4. フィルターエディターで、[追加] ボタンをクリックします。
5. [フィルターの選択] リストで、[スク립トレット]を選択します。
6. 次のいずれかの方法で、データをフィルタリングするスク립トレットを作成します。
 - [テンプレートの挿入] をクリックし、スク립トレットテンプレートをベースとして使用します。
 - 別のオペレーションまたは **Configuration\Scriptlets** フォルダーの既存のスク립トレットのテキストをコピーして貼り付けます。
フィルターの作成の詳細については、[「出力および結果のフィルタリング」\(255ページ\)](#)を参照してください。

7. **[スクリプトの確認]** をクリックしてエラーがないか確認します。

スクリプトレットでエラーが見つかった場合は、エラーが赤い下線でマークされ、スクリプトレットが含まれるフローも同様になります。

注: 無効なスクリプトレットが含まれるフローは、プロジェクトからコンテンツパックを作成するときには含まれません。「[コンテンツパックのエクスポート](#)」(336ページ)を参照してください。

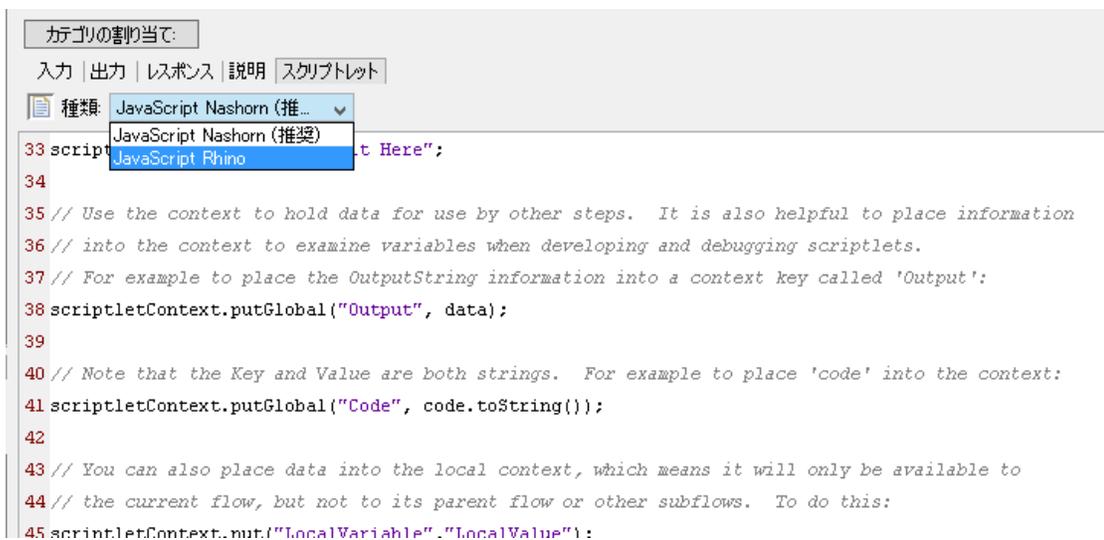
8. フィルターをテストして作業結果を保存します。

ヒント: スクリプトレットオペレーションを作成する場合、スクリプトレットで、スクリプトレットレスポンスとして `success` を指定します。続いて、オペレーションの **[レスポンス]** タブで、デフォルトのレスポンスとして `failure` を選択します。

オペレーションレスポンスのスクリプトレットルールを作成する

スクリプトレットを使用して、オペレーションのレスポンスを制御できます。

1. オペレーションの **[レスポンス]** タブを開いて、レスポンスを選択します。
2. レスポンスの行の右端にある右向き矢印  をクリックして、ルールエディターを開きます。
3. **[ルールの種類]** リストで、**[スクリプトレット]** を選択します。
4. ルールの行の右端にある右向き矢印  をクリックして、ルール詳細エディターを開きます。



```
カテゴリの割り当て:
 入力 | 出力 | レスポンス | 説明 | スクリプトレット
種類: JavaScript Nashorn (推奨)
33 scriptletContext.put("Output", "Here");
34
35 // Use the context to hold data for use by other steps. It is also helpful to place information
36 // into the context to examine variables when developing and debugging scriptlets.
37 // For example to place the OutputString information into a context key called 'Output':
38 scriptletContext.putGlobal("Output", data);
39
40 // Note that the Key and Value are both strings. For example to place 'code' into the context:
41 scriptletContext.putGlobal("Code", code.toString());
42
43 // You can also place data into the local context, which means it will only be available to
44 // the current flow, but not to its parent flow or other subflows. To do this:
45 scriptletContext.put("LocalVariable", "LocalValue");
```

5. 次のいずれかの方法で、スクリプトレットを作成します。
 - **[テンプレートの挿入]** をクリックし、スクリプトレットテンプレートをベースとして使用します。
 - 別のオペレーションまたは **Configuration\Scriptlets** フォルダーの既存のスクリプトレットのテキストをコピーして貼り付けます。

6. スクリプトレットを作成し、[スクリプトの確認] をクリックしてエラーを確認します。
7. 作業結果を保存します。

オペレーション、フロー、ステップのシステムスクリプトレットのフィルタリング

1. システムスクリプトレットを使用するオペレーション、フロー、ステップについて、[プロパティ]シートまたはステップのインスペクターの[スクリプトレット]タブを開きます。
2. [プロジェクト]ペインで、**Configuration** フォルダと **Scriptlets** フォルダを展開します。
3. スクリプトレットを **Scriptlets** フォルダから [プロパティ]シートまたはステップのインスペクターの[スクリプトレット]タブの[スクリプトレット] アイコンにドラッグします。
[スクリプトレット]タブに、共有のスクリプトレットへの参照があることが表示されます。



スクリプトレットを Configuration\Scriptlets フォルダに保存する

1. [プロパティ]シートまたはステップのインスペクターの[スクリプトレット]タブで、保存するスクリプトレットを開きます。
2. [スクリプトレット] アイコンを [プロジェクト]ペインの **Configuration\Scriptlets** フォルダにドラッグします。
3. スクリプトレットの名前を入力します。

参考資料

スクリプトレットエディター

スクリプトレットエディターの外観は、[プロパティ]シートまたはステップのインスペクターの[スクリプトレット]タブで開く場合でも、**Configuration\Scriptlets** フォルダでスクリプトレットをダブルクリックして開く場合でも同じです。

カテゴリの割り当て:

入力 | 出力 | レスポンス | 説明 | スクリプトレット

種類: JavaScript Nashorn (推奨) (推奨) (推奨)

```
33 scriptletContext.putGlobal("Output", data);
34
35 // Use the context to hold data for use by other steps. It is also helpful to place information
36 // into the context to examine variables when developing and debugging scriptlets.
37 // For example to place the OutputString information into a context key called 'Output':
38 scriptletContext.putGlobal("Output", data);
39
40 // Note that the Key and Value are both strings. For example to place 'code' into the context:
41 scriptletContext.putGlobal("Code", code.toString());
42
43 // You can also place data into the local context, which means it will only be available to
44 // the current flow, but not to its parent flow or other subflows. To do this:
45 scriptletContext.put("LocalVariable", "LocalValue");
```

GUI アイテム	説明
スクリプトレットアイコン 	このアイコンを Configuration\Scriptlets フォルダにドラッグすると、スクリプトレットをフォルダに保存して再利用できます。
テンプレートの挿入	[テンプレートの挿入] をクリックすると、スクリプトレットを書くのに役立つガイドラインが表示されます。
スクリプトの確認	[スクリプトの確認] をクリックすると、スクリプトレットにエラーがないか確認されます。

フローでの正規表現の使用

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

正規表現を使用すると、次のことを実行できます。

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

正規表現のワイルドカードと修飾子

正規表現の主なワイルドカードは次のとおりです。

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
\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

正規表現の修飾子は次のとおりです。

修飾子	結果
*	ゼロ (0) 回以上の繰り返しに一致
+	1 回以上の繰り返しに一致
?	ゼロ (0) 回または 1 回の繰り返しに一致
{n}	n 回の繰り返しに完全一致
{n,}	n 回以上の繰り返しに一致
{n,m}	n ~ m 回の繰り返しに一致

[abc]	a、b、cのいずれかに一致
[^abc]	a、b、c以外の任意の文字に一致
[a-c]	a ~ cの間の任意の文字に一致
alb	aまたはbに一致
\	特殊文字をエスケープする(たとえば\.の場合、\.がそれ自身以外のいずれの文字にも一致しないことを意味します)

実行する操作

正規表現を使用してテスト出力をフィルタリングする

- 出力または結果のフィルターエディターを開いて、新しいフィルターを作成します。詳細については、「[出力および結果のフィルタリング](#)」(255ページ)を参照してください。
- [**フィルターの選択**] リストから、フィルターの種類として [**正規表現**] を選択します。右上の [**詳細:**] セクションに、正規表現を作成するコントロールが表示されます。

詳細: 正規表現
正規表現と一致する部分を抽出して、入力文字列をフィルターします。正規表現の詳細な説明については、ヘルプマニュアルを参照してください。

正規表現の種類: Java スタイル

正規表現の値: *

フィルターのスタイル: 入力全体をフィルター
 行単位でフィルター

大文字と小文字を区別しない:

- [**正規表現の種類**] リストで、[**Java スタイル**] を選択します。ほかのスタイルは使用しないでください。推奨されていません。
- [**正規表現の値**] ボックスに正規表現を入力します。

例: パケット損失の数を抽出するには、正規表現 `Lost = \d` を使用します。

この表現で、文字列 "Lost =" とその後に続く任意の数字を検索するよう、HP 00 に指定します。

ワイルドカード `\d` を使用して、任意の文字を検索するよう、HP 00 に指定します。

- [**フィルターのスタイル**] で、未加工結果にどのようにフィルターを適用するかに応じて、[**入力全体をフィルター**] または [**行単位でフィルター**] を選択します。
- 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。
- [**選択したフィルターのテスト**] をクリックしてフィルターをテストします。
- フィルターを保存します。

複数の正規表現を組み合わせて値を抽出する

複数の正規表現を組み合わせて、フィルターで値を抽出できます。

たとえば、Unix の ps コマンドの出力で ps の時間を抽出するには、2 つの正規表現が必要です。1 つ目で出力を ps の行まで絞り込んで、2 つ目で時間を抽出します。

F	S	UID	PID	PPID	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

1. 出力または結果のフィルターエディターを開きます。
2. 正規表現のフィルターを新しく追加します。
3. **[正規表現の値]** ボックスに、1 つ目の正規表現を入力します。
この例では、`.*ps` と入力しています。これにより、"ps" で終わる任意の文字列が抽出されます。

注: 先頭のピリオド [.] を省略しないでください。

4. **[行単位でフィルター]** チェックボックスをオンにします。
5. **[選択したフィルターのテスト]** をクリックします。
[出力のテスト] ボックスで出力されるのは、"ps" の文字が入った行だけとなります。
6. 正規表現の 2 つ目のフィルターを追加します。
7. **[正規表現の値]** ボックスに `\d*:\d*:\d*` と入力します。
この表現は、コロンで区切られた 3 セットの数字を表します。この例では、この表現によって、行から時間が抽出されます。
8. **[選択したフィルターのテスト]** をクリックします。
9. 保存します。

出力のテストに ps 行の時間だけが表示されます。これで、この値を変数に割り当てることができます。

参考資料

フィルターエディター > 詳細: 正規表現

[正規表現] をフィルターの種類として選択すると、右上の **[詳細:]** セクションに正規表現を作成および変更するコントロールが表示されます。

詳細: 正規表現
正規表現と一致する部分を抽出して、入力文字列をフィルターします。正規表現の詳細な説明については、ヘルプマニュアルを参照してください。

正規表現の種類: Java スタイル

正規表現の値: *

フィルターのスタイル: 入力全体をフィルター
 行単位でフィルター

大文字と小文字を区別しない:

GUI アイテム	説明
正規表現の種類	データをフィルタリングする正規表現の種類として [Java スタイル] を選択します。ほかのスタイルは使用しないでください。推奨されていません。
正規表現の値	正規表現を入力します。
フィルターのスタイル > 入力全体をフィルター	未加工結果全体にフィルターを適用するには、これを選択します。
フィルターのスタイル > 行単位でフィルター	各行に個別にフィルターを適用するには、これを選択します。
大文字と小文字を区別しない	正規表現の大文字小文字を区別しない場合は、これを選択します。

HP Live Network の内容を Studio から検索

Studio 作成者は、HPLN プロファイルへのアクセス許可に基づいて、HP Live network 上の関連情報やコンテンツを Studio から直接検索できます。

検索の種類

使用可能な検索の種類は次のとおりです。

- **種類別の検索:** フロー、オペレーション、システムアカウント、システムプロパティ、選択リスト、ドメインターム、システム評価子、システムフィルター、スクリプトレット、グループエイリアス、カテゴリ、役割エイリアスを検索できます。
- **名前で検索:** 名前で検索します。不完全な語には、ワイルドカード文字 * を使用することができます。ワイルドカード * は任意の文字に一致します。名前を指定しないと、いずれの名前も一致になります。この検索では大文字と小文字が区別されません。
- **説明による検索:** 説明を検索します。不完全な語には、ワイルドカード文字 * を使用することができます。ワイルドカード * は任意の文字に一致します。説明を指定しないと、いずれの説明も一致になります。この検索では大文字と小文字が区別されません。
- **入力名で検索:** 入力名を検索します。不完全な語には、ワイルドカード文字 * を使用することができます。ワイルドカード * は任意の文字に一致します。すべてのコンテンツパック内の構成アイテムを除くすべてのアイテムと、検索文字列と一致する入力がある各コンテンツパックの特定のメタデータを一緒に返します。この検索は、構成アイテムには使用できません。
- **UUID による検索:** UUID を完全に一致する文字列で入力します。この検索は、= 演算子に基づいて照合されます。この検索では、その UUID を持つフロー、オペレーション、または構成アイテムが返されます。
- **組み合わせ検索:** 複数の条件で検索できます。これには、AND 演算を使用する、種類、名前、説明、入力のうちの 2 種類以上の条件が含まれます。

注: その他すべてのフィールドが空の検索の種類は許容されていません。この場合、検索ボタンは無効です。

hpln-index-generator ツール

<oo インストールフォルダー>/studio/tools/lib/hpln-index-generator.jar にある hpln-index-generator ツールは、フロー、オペレーション、構成アイテムを説明し、コンテンツパック内の対応するメタ

データを持つインデックスファイル(.json)を生成します。

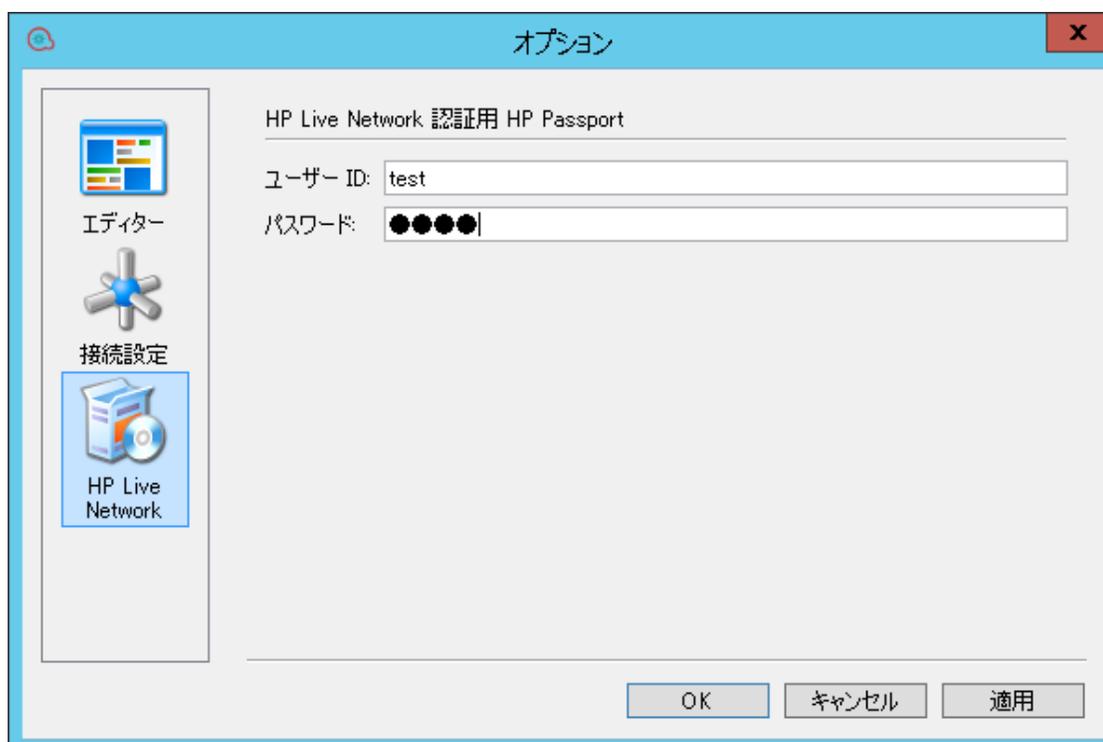
注: Studio で HPLN 検索機能を使用するには、コンテンツパックの.jarファイルと共に.jsonファイルをHPLNにパブリッシュする必要があります。詳細については、「[HPLN Index Generator ツールの実行](#)」(309ページ)を参照してください。

実行する操作

Studio からの HPLN 接続の設定

1. [構成]メニューから[オプション]>[HP Live Network]を選択します。
2. [HP Live Network 設定]ダイアログボックスに、HP Live Network のユーザー ID とパスワードを入力します。

注: このダイアログボックスに入力したユーザー ID とパスワードにより、アクセス許可が決まります。また、検索結果にも影響します。

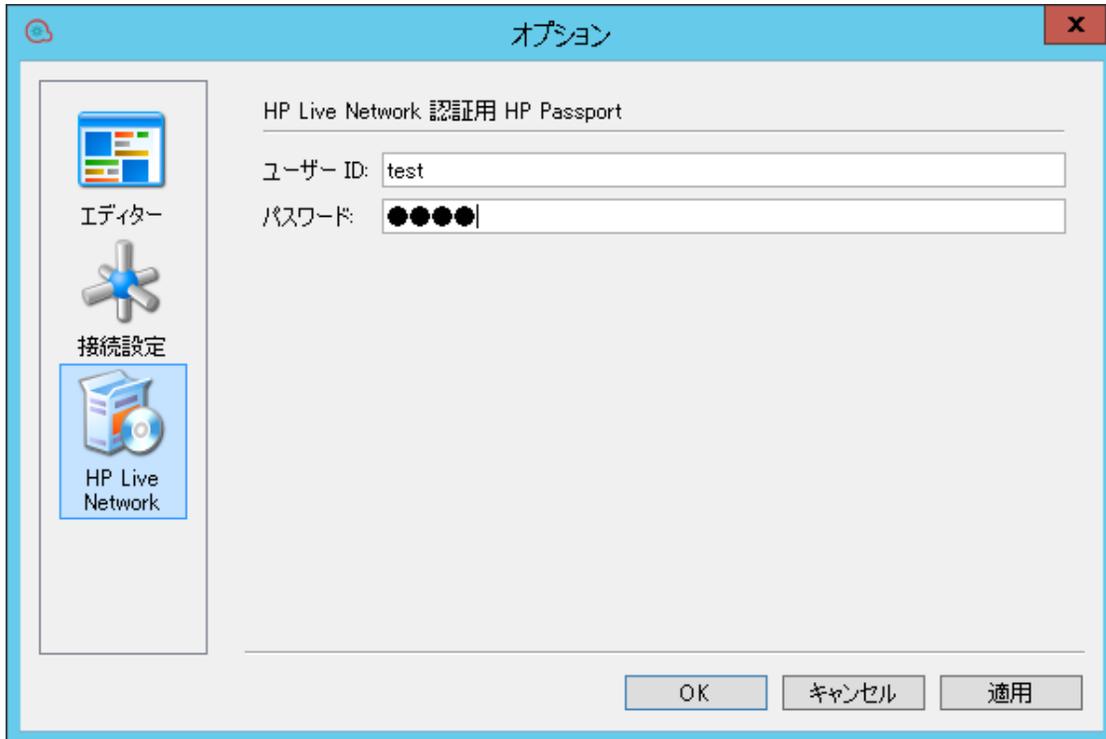


3. [適用]をクリックし、[OK]をクリックします。

Studio からの HP Live Network の検索

1. [構成]メニューから[オプション]>[HP Live Network]を選択します。
2. [HP Live Network 設定]ダイアログボックスに、HP Live Network のユーザー ID とパスワードを入力します。

注: このダイアログボックスに入力したユーザー ID とパスワードにより、アクセス許可が決まります。また、検索結果にも影響します。



3. [適用] をクリックし、[OK] をクリックします。
[HPLN 検索] ペインは、Studio のワークスペースの下部にあり、デフォルトでは開いています。
4. 検索条件を入力し、Enter を押すか、検索ボタンをクリックします。



HP Live Network 検索では、最大 100 件の結果が返されます。

注: \、(、)、& の各文字は、HP Live Network 検索用の名前、説明、入力、UUID 内ではサポートされません。

HPLN Index Generator ツールの実行

1. Windows の [スタート] メニューで、プロンプトフィールドに「cmd」と入力し、コマンドラインウィンドウを開きます。
2. 00 のインストールフォルダーの下にある `studio\tools\lib` に移動します。
3. DOS プロンプトで次のコマンドを入力します。

```
hpln-index-generator-<version>.jar --content-pack <arg>
```

<version> は、HPLN Index Generator ツールのバージョン番号で、<arg> はコンテンツパックの jar ファイルです。

プロセスの最後に、コンテンツパック名に .json という拡張子が付いた JSON ファイルが作成されたことを知らせるメッセージが表示されます。例: oo10-virtualization-cp-1.2.1.json。

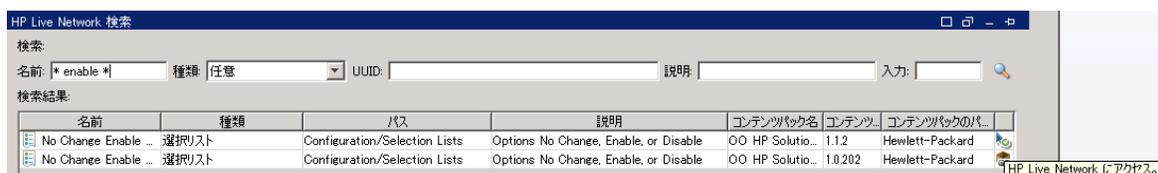
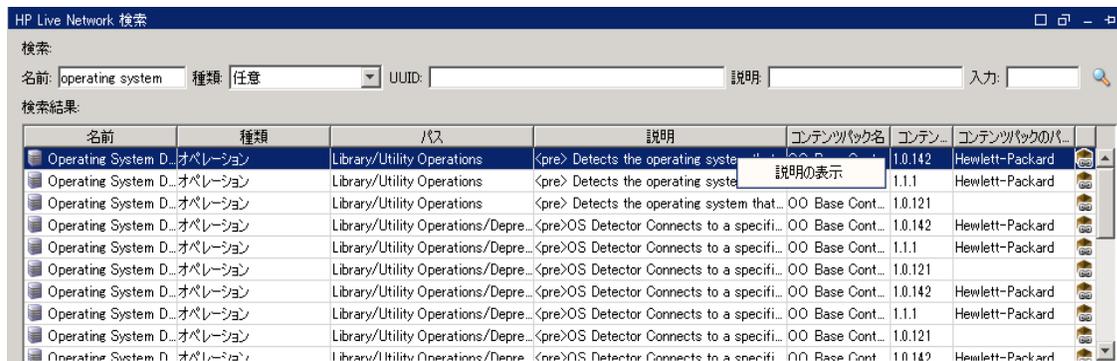
参考資料

HP Live Network 検索 > 検索結果

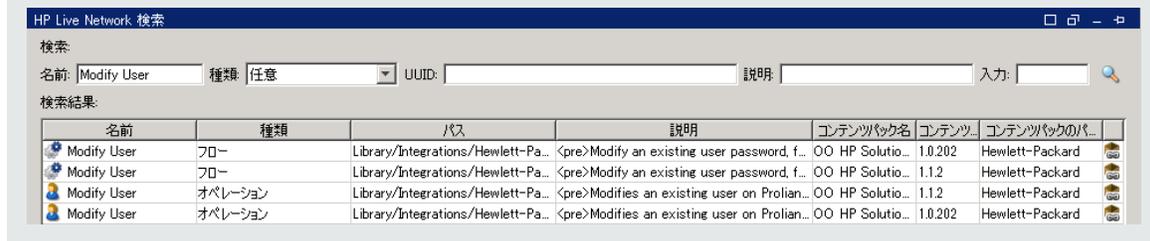
結果は、列名に従ってソートすることができるテーブルに表示されます。さらに、複数の値を使用して結果をフィルタリングすることができます。

行を右クリックすると、検索結果の完全な説明を表示できます。

HP Live Network で、行の末尾にあるリンクボタン  をクリックして、コンテンツの場所に直接移動することもできます。



注: 最近リリースされたコンテンツパックの特定のアイテムを検索する場合、コンテンツパック名でフィルタリングし、コンテンツパックバージョンの降順でソートできます。リスト内の最初のアイテムは、最近リリースされたバージョンのもので、



コンテンツの検査

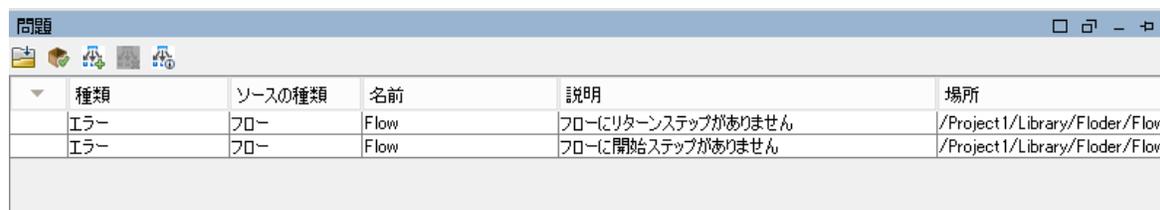
コンテンツをリリースする前に、プロジェクトのフローをテストし検査することが重要です。Studio では、これを行う際に役立つ次のツールを用意しています。

- [問題] ペイン。問題が、発生した場所と内容説明とともに一覧され、この問題を解決するための指針となります
- デバッガー。フローで見つかったエラーや予期しない動作の原因を突き止めるのに役立ちます

[問題] ペインでのフローの検査

フローを実行するには、フローそのもの、そのオペレーション、そしてそのフローで使用するすべてのシステムアカウントが有効である必要があります。

[問題] ペインを使用して、個別のフローやオペレーションに問題がないかチェックしたり、プロジェクト全体を検査したりできます。この場合、プロジェクトのすべてのフロー、オペレーション、システムアカウントが検査されます。



種類	ソースの種類	名前	説明	場所
エラー	フロー	Flow	フローにリターンステップがありません	/Project 1/Library/Floder/Flow
エラー	フロー	Flow	フローに開始ステップがありません	/Project 1/Library/Floder/Flow

有効なフローの要件

フローを有効にするには次の条件を満たすことが必要です。

- 少なくとも1つのステップがある
- ステップの1つが開始ステップとして指定されている
- 各ステップについて、各レスポンスから、ステップが後続ステップにトランジションによって接続されている
- フローの各ステップに何らかの実行処理に至る経路がある
- 値を返してフローを終了するリターンステップがある
- 各入力の値の取得方法が割り当てられている

実行する操作

フローまたはオペレーションの検査

1. [プロジェクト] ペインでフローまたはオペレーションを選択します。
2. [問題] タブをクリックし、[問題] ペインを表示します。
3. [問題] ペインで行をダブルクリックして、アイテムを開いて編集します。

注: 複数のアイテムを開くには、Shift キーまたは Ctrl キーを使用して選択し、右クリックして [開く] を選択します。

プロジェクトのすべてのフローとオペレーションの検査

1. 検査するプロジェクトを開きます。
2. [ツール] メニューから [フローとオペレーションの検査] を選択します。

問題がある場合、その場所と説明が一覧表示され、問題を修復する方法が示されます。

参考資料

[問題] ペイン

[問題] ペインは Studio ウィンドウの下端にある [問題] タブをクリックすると開き、選択したフローやオペレーションが有効かどうかを確認できます。

種類	ソースの種類	名前	説明	場所
エラー	フロー	Flow	フローにリターンステップがありません	/Project1/Library/Floder/Flow
エラー	フロー	Flow	フローに開始ステップがありません	/Project1/Library/Floder/Flow

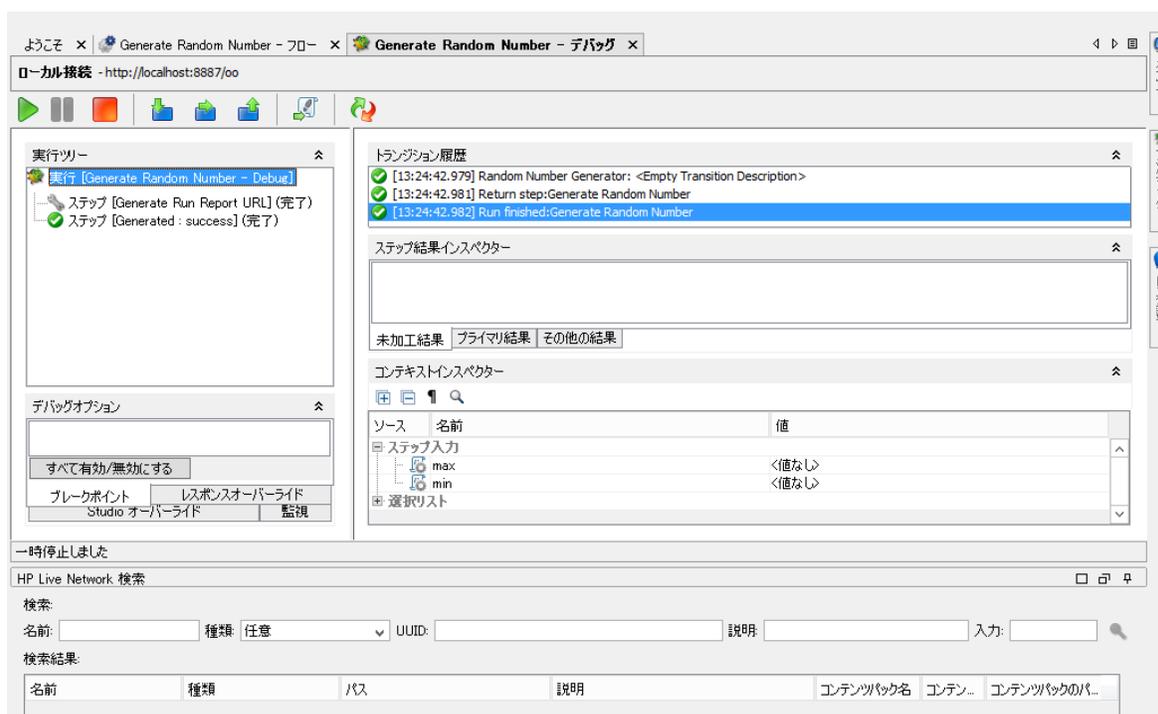
GUI アイテム	説明
ソースの種類	問題がある要素の種類が表示されます。
名前	問題がある要素の名前が表示されます。
説明	問題の説明と、修復する方法が示されます。
場所	問題がある要素の場所が表示されます。

フローのテストとデバッグ



デバッガーは、以下の情報を表示し、フローで見つかったエラーや予期しない動作の原因を突き止めるのに役立ちます。

- 実行したステップを表示するツリー
- 各ステップで生成されたステップ結果とオペレーション出力
- 各ステップのさまざまなコンテキストのフロー変数の値
- 追跡した各トランジションの説明



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。

デバッガーの各ステップから、フロー内の関連するステップに直接ジャンプすることができます。ジャンプすると、フローエディターが開きます。

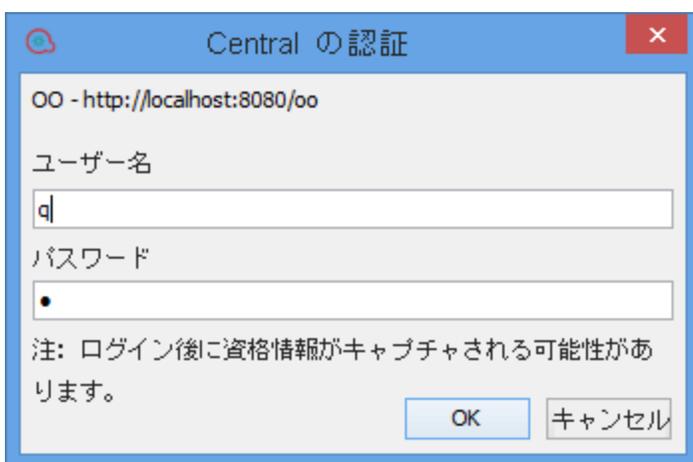
注:

- 暗号化された入力または非表示の入力から取得された結果は ***** 文字で表示されます。これらの値は、フローをトリガーする前にコンテキストインスペクターで編集する必要があります。さもなければ、適用される入力の値は、新しい実行の間、***** になります。

- コンテキストインスペクターで、非表示の出力の名前は表示できますが、それらの値は***** で隠されて表示されます。

ユーザー認証が必要なフローのデバッグ

Central でフローの認証が有効になっている場合 ([**認証を有効にする**] チェックボックス)、フローのデバッグを選択したときに、[Central の認証] ダイアログボックスでユーザー資格情報を入力するように求めるメッセージが表示されます。資格情報入力した後は、[**ユーザー名**] および [**パスワード**] 資格情報が現在のセッション中は記憶されるので、追加のフローをデバッグするときに再入力する必要はありません。



認証の設定の詳細については、『Central ガイド』の「セキュリティ設定のセットアップ」を参照してください。

注: Central で認証が有効になっている場合、ユーザー名にコロンが含まれるユーザーでログインすることはできません。

ベストプラクティス

親フローをデバッグする前に、サブフローをデバッグすることをお勧めします。

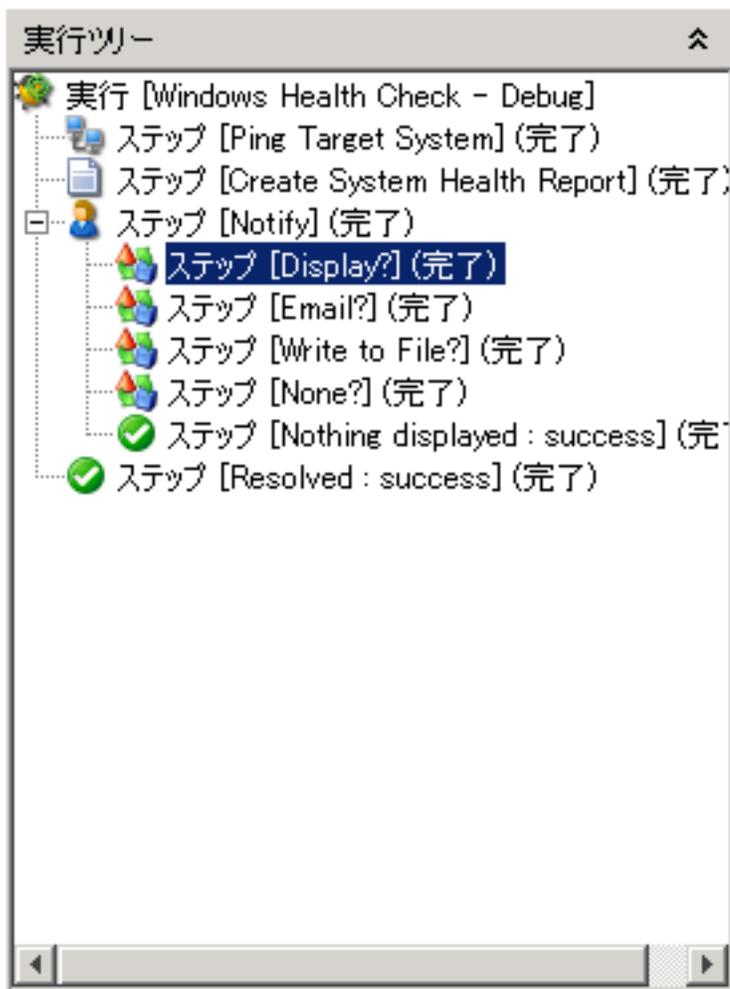
実行する操作

フローのデバッグ

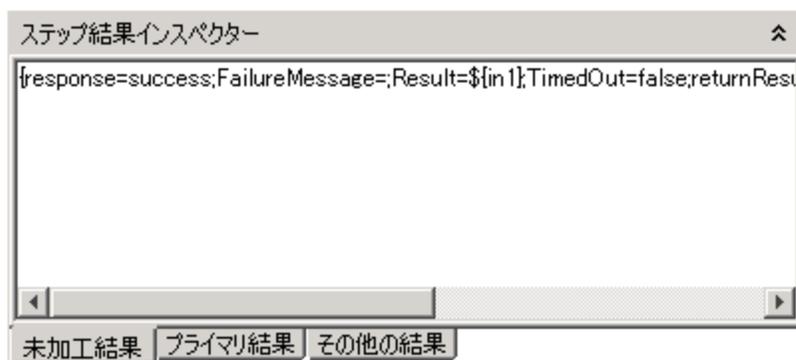
1. [プロジェクト] ペインでフローを右クリックし、[デバッグ] をクリックします。

注: または、作成ペインでフローを開き、[デバッグ]  ボタンをクリックします。

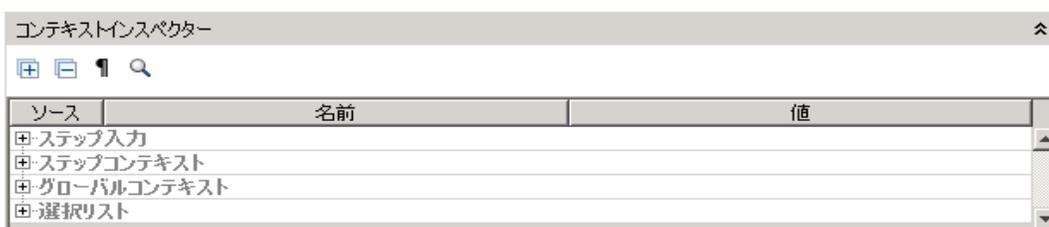
2. フローを最後まで実行するには、デバッガーのツールバーの**[再生]**  ボタンをクリックします。
3. 完了したステップの情報を確認するには、**[実行ツリー]** ペインで目的のステップをクリックします。



4. **[ステップ結果インスペクター]** ペインでは、ステップの未加工結果、プライマリ結果またはフィルタリング済み結果を表示できます。



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



フローのデバッグ - ステップごと

1. デバッガーでフローを開きます。
2. フローをステップごとに実行するには、[ステップオーバー]  ボタンをクリックします。

サブフローへのステップインとサブフローからのステップアウト

これらのアクションを使用すると、フローのデバッグ中にサブフローの実行にステップインまたはステップアウトできます。どちらのアクションも、デバッグ中のフローが一時停止してユーザーのアクションを待っているときに使用できます。

注: 親フローをデバッグする前に、サブフローをデバッグすることをお勧めします。

- ステップのサブフローにステップインするには、[ステップイン]  ボタンをクリックします。デバッガーはサブフローの実行を開始し、サブフローの最初のステップで一時停止します。親フローの先頭で[ステップイン]をクリックすると、デバッガーは親フローの最初のステップで一時停止します。
- サブフローからステップアウトするには、[ステップアウト]  ボタンをクリックします。デバッガーは、現在のサブフロー呼び出しの残りのステップを実行し、サブフローの後の(親フローの)最初のステップで一時停止します。現在のステップが親フロー内にある場合は、このアクションは再開アクションと同様に動作します。

デバッガーでのペインの折りたたみ/展開

ほかのペインにより多くのスペースを使用できるようにするために、デバッガーの一部のペインを折りたたむ必要がある場合があります。

- ペインを折りたたむには、ペインの右上にある上向き二重矢印  をクリックします。
- 折りたたんだペインを元に戻すには、下向き二重矢印  をクリックします。

デバッガーでのフローのリセットと再開

フローのリセットと再開を行うと、フロー変数の値がデバッガーを開いたときの値にリセットされます。

1. デバッガーのツールバーで、[リセット]  ボタンをクリックします。
2. [再生]  ボタンをクリックします。

デバッガーでのフロー変数の値の変更

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

1. デバッガーでフローを開きます。
2. 目的のステップが保留中になるまで、[ステップオーバー]  をクリックします。

注: ステップの前にブレークポイントを設定済みである場合、[再生]  をクリックし、ステップが一時停止するまでフローを実行します。

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

列で検索、ソート、およびフィルタリングできます。

コンテキストインスペクター		
ソース	名前	値
	ステップ入力	
	ステップコンテキスト	
	グローバルコンテキスト	
	選択リスト	

- [ステップ入力] セクションの値は、ステップ開始前に入力に割り当てられた値です。
 - [ステップコンテキスト] セクションの値は、ステップ開始後に更新された値です。
3. このステップで使用されるフロー変数の値を変更するには、[ステップ入力] で目的のフロー変数のリストを見つけ、その値を強調表示し、新しい値を入力して置き換えます。上記の例では、ステップはマルチインスタンスステップです。host フロー変数のリストに別の IP アドレスを追加できます。

4. このステップでアクセス可能だが後続のステップで使用されるフロー変数の値を変更するには、[ステップコンテキスト]に表示される該当フロー変数のリストの値を変更します。
5. フローの実行を続行します。
6. 変更したフロー変数の値を、前回のフロー保存時に設定した値にリセットするには、[リセット]
 ボタンをクリックします。

フローのブレイクポイントの設定

ブレイクポイントは、デバッガーでのフローの実行を自動的に一時停止します。これは、たとえば次のような処理を行う際に便利です。

- フロー変数の値を調べる
- フロー変数の値を変更して、残りの実行でフローが受ける影響を確認する

ブレイ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イクポイントを有効または無効にできます。

1. 作成ペインでフローを開き、ブレイクポイントを設定するステップを右クリックします。
2. [デバッグ]>[ブレイクポイントの設定]を選択します。

フロー図では、ブレイクポイントは黄と黒の枠で囲まれたステップで表されます。



List Appender

3. デバッガーでフローを開きます。

デバッガーの[デバッグオプション]ペインの[ブレイクポイント]タブに、既存のブレイクポイントが表示されます。



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

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

単一ステップのデバッグ実行でレスポンスをオーバーライドする

レスポンスのオーバーライドにより、オペレーションが失敗した場合でも、選択したレスポンスを強制的に指定できます。

レスポンスをオーバーライドすると、デバッガーを終了して入力値を変更せずに、フローの特定のパスをテストできます。

たとえば、フローのステップについて必要な情報がない場合に、このステップが失敗したかどうかに関わらず、フローの残りのステップについてテストが必要な場合があります。ユーザーが介入しない場合に発生する failure レスポンスではなく、目的のレスポンスとトランジションをたどるように強制的に実行を続けることができます。

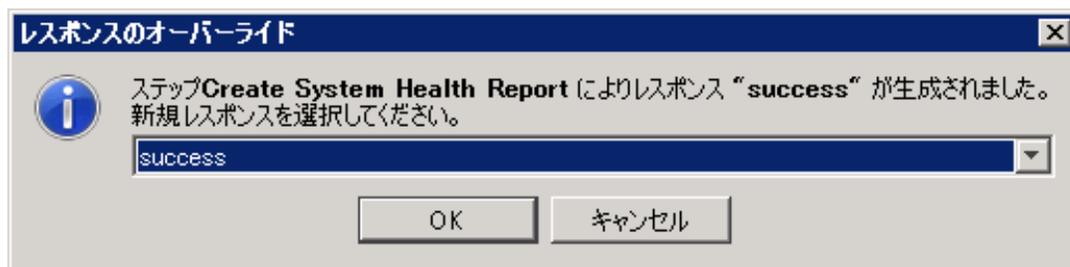
1. 作成ペインでフローを開き、レスポンスをオーバーライドするステップを右クリックします。
2. [デバッグ]>[レスポンスのオーバーライド]を選択し、ステップに強制的に指定するレスポンスをクリックします。
 - **None**
 - **success**
 - **failure**
 - **Prompt**

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

3. デバッガーでフローを開きます。
デバッガーの[デバッグオプション]ペインの[レスポンスオーバーライド]タブに、既存のレスポンスオーバーライドが表示されます。



4. 目的のレスポンスオーバーライドにスクロールします。
 5. 次のいずれかを実行します。
 - オーバーライドに別のレスポンスを選択するには、下向き矢印をクリックしてレスポンスを選択します。
 - 1つのレスポンスオーバーライドを有効にするには、チェックボックスをオンにします。
 - 1つのレスポンスオーバーライドを無効にするには、チェックボックスをクリアします。
 - すべてのレスポンスオーバーライドを有効または無効にするには、[すべて有効/無効にする]をクリックします。
 - すべてのレスポンスオーバーライドをクリアするには、[ツール]メニューから[すべてのレスポンスオーバーライドの削除]を選択します。
 - ステップごとにレスポンスをオーバーライドする場合は、[すべてのレスポンスのオーバーライド]チェックボックスをオンにします。
- すべてのレスポンスをオーバーライドした後にデバッガーでフローを実行すると、ステップごとに該当ステップのレスポンスを手動で選択するよう求められます。



実行ツリーからフローステップへのジャンプ

- [実行ツリー]でステップを右クリックし、[ステップに移動]を選択します。
または:
ツールバーの[ステップに移動]ボタンをクリックします 。
フローエディターが開き、現在のフローと選択したステップが表示されます。

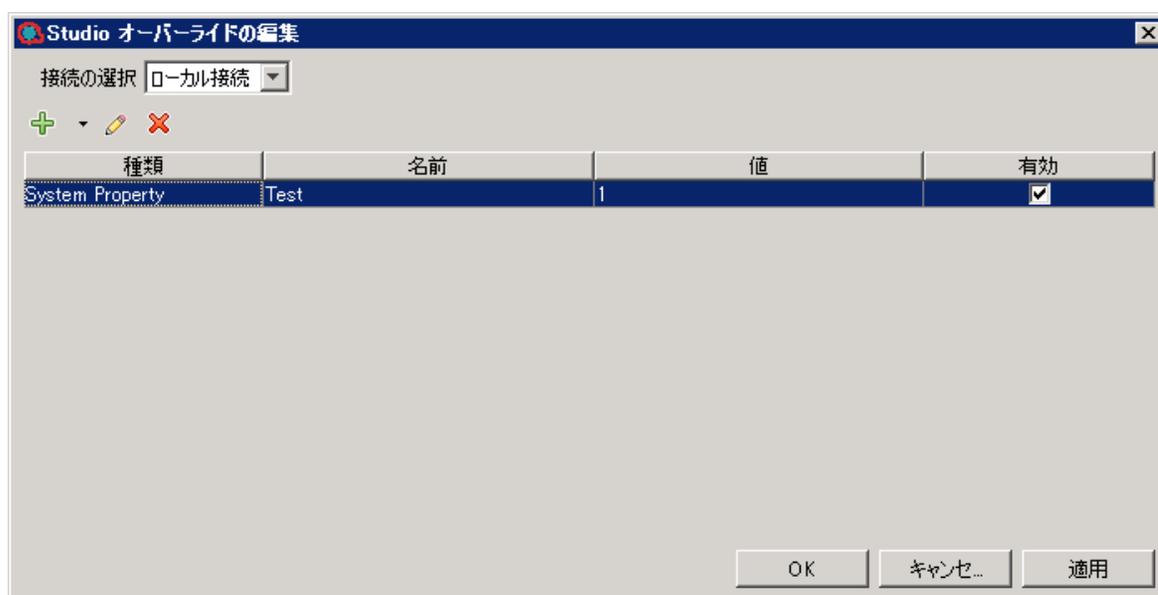
Studio でのオーバーライドの設定

Studio では、システムプロパティとシステムアカウントに対して Central 接続ごとにオーバーライドを定義できます。オーバーライドを作成すると、トリガーする前に、コンテキストインスペクターで値を変更せずに、構成アイテム、システムプロパティ、およびシステムアカウントを変更できます (コンテンツパックから)。たとえば、これらのオーバーライドを使用すると、コンテンツパックからインポートされた読み取り専用の値を調整できます。

注: 個々のオーバーライドは有効または無効にできます。オーバーライドは名前と種類が一意で、Central オーバーライドよりも優先されます。

[Studio オーバーライド] にアクセスするには:

1. [構成] メニューで [Studio オーバーライド] を選択します。



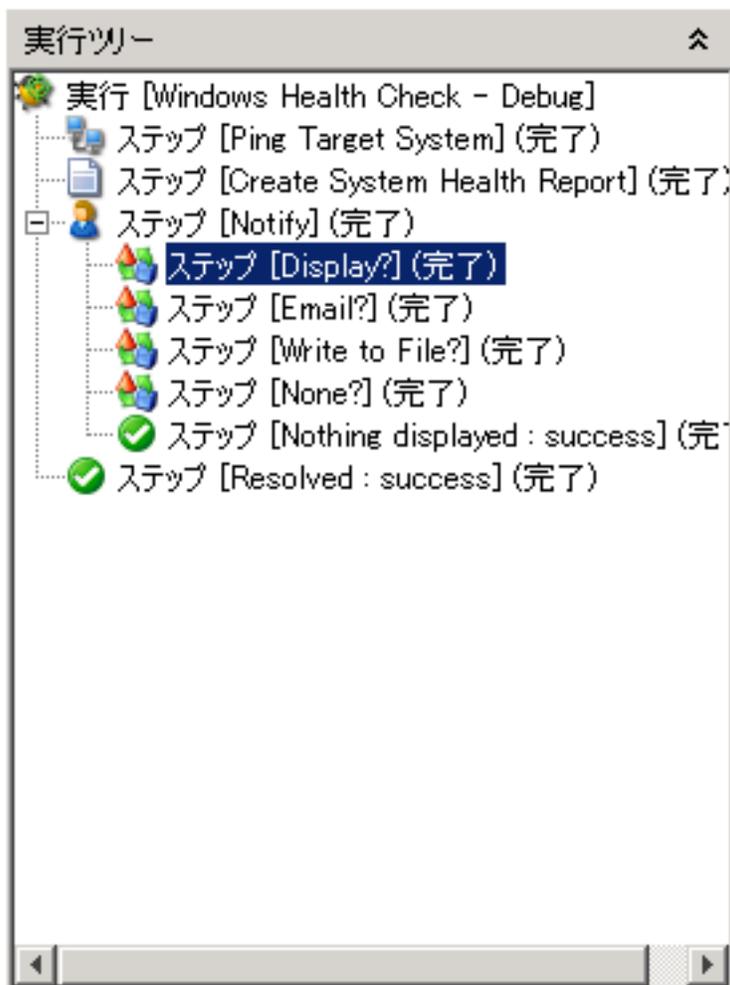
2. [Central の選択] ドロップダウンリストから Central 接続を選択します。Central を選択すると、その Central のオーバーライドが表示されます。
3. 次の操作を実行できます。
 - **+** システムプロパティまたはシステムアカウントのオーバーライドを追加します。
 - **✎** 選択したシステムプロパティを編集します。
 - **X** 選択したシステムプロパティを削除します。
4. 変更を適用するには、[適用] をクリックし、完了したら [OK] をクリックします。

注: システムプロパティを空白の値でオーバーライドした場合、[システムプロパティ] 定義のデフォルト値が使用され、空白の値は使用されません。

注: システムプロパティを参照する場合は、完全なパスを使用する必要があります。たとえば、システムプロパティが `folderA\folderB\my_ci` というフォルダー構造の下にある場合、`${folderA/folder/my_ci}` という文字列を使用して参照します。

参考資料

[実行ツリー] ペイン



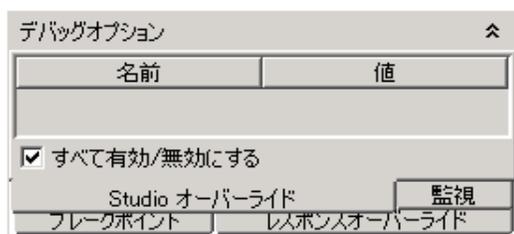
[実行ツリー] ペインには、フローのサブフロー内のステップも含めて、実行する各ステップが表示されます。

実際の実行では、同時に実行されるステップは、デバッガーで順に実行されます。

[デバッグオプション] ペイン



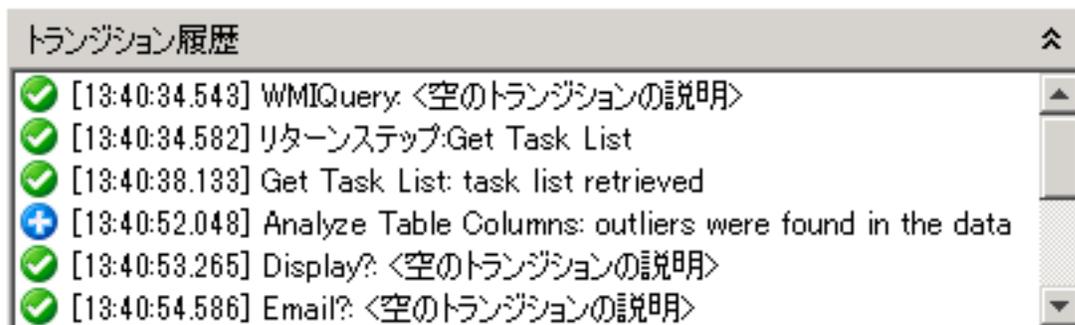
- **[ブレイクポイント]**は、特定のステップで実行を自動的に一時停止し、停止ポイントでのフロー変数の値、実行のパス、または結果を調べるためのフラグです。
- **[レスポンスオーバーライド]**により、オペレーションの結果に関係なく、選択したレスポンスを強制的に指定できます。
- **[Studio オーバーライド]**は、有効な Studio オーバーライドを表示します。このビューでは、必要に応じてオーバーライドを有効および無効にできます。



- **[監視]**では、ユーザーはデバッグ中に変数の監視リストを作成できます。これらの変数は、テック入力、ステップコンテキスト、グローバルコンテキスト、および構成アイテム (選択リスト、システムプロパティ、およびシステムアカウント) など、さまざまなコンテキストのエントリとなります。監視リストでは、名前を変数を追加でき、さまざまなコンテキストと構成アイテムのリストからその名前を持つすべてのエントリが監視に表示されます。

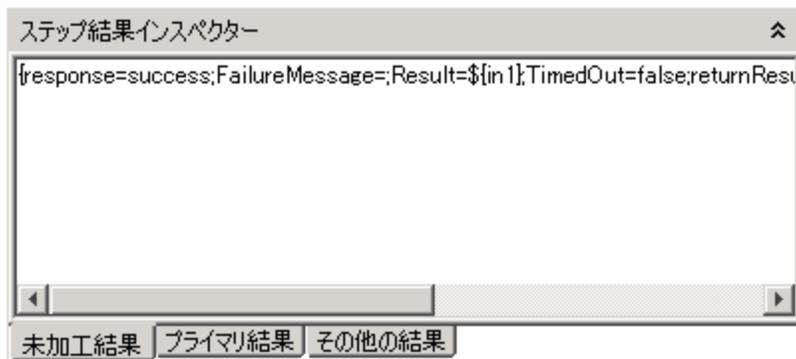
[デバッグオプション]ペインでは、ブレイクポイントとレスポンスオーバーライドが表示され、該当の実行について各要素の削除、有効化、または無効化が可能です。

【トランジション履歴】ペイン



[**トランジション履歴**] ペインでは、実行内で発生したトランジションとその説明が一覧表示されま
す。

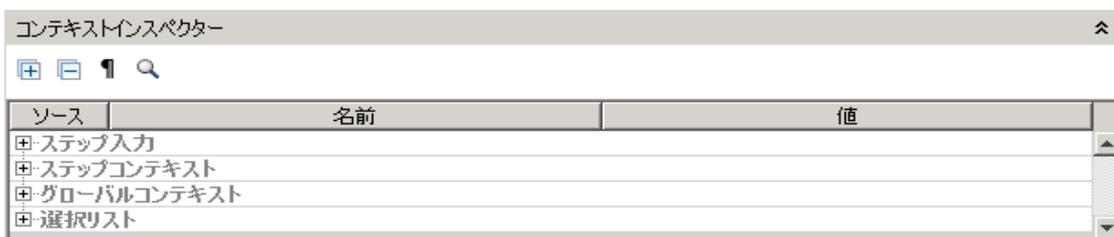
[**ステップ結果インスペクター**] ペイン



[**ステップ結果インスペクター**] ペインには、選択したステップの結果が表示されます。

- 未加工結果 (ステップのオペレーションの結果) を表示するには、[**未加工結果**] タブをクリックします。
- ステップのプライマリ結果を表示するには、[**プライマリ結果**] タブをクリックします。
- 作成した結果がほかにもある場合、そうした結果を表示するには、[**その他の結果**] をクリックしま
す。

[**コンテキストインスペクター**] ペイン



[**コンテキストインスペクター**] ペインには、各ステップについてフロー変数 (グローバルおよびロー
カル) の現在の値が表示されます。

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

- [**ステップ入力**] セクションの値は、ステップ開始前に入力に割り当てられた値です。フロー変数の
値を含むテキストボックスは、色分けして表示されます。

ソース	名前	値
ステップ入力		
ステップコンテキスト		
	max	100
	min	10
	RS_Previous_Response	success
	RS_Previous_Transition	success

- [ステップコンテキスト]セクションの値は、ステップ開始後に更新された値です。
ステップのコンテキストは、ステップのフローと親フローのローカルコンテキストに含まれるフロー変数と値の割り当ての集合です(フローが別のフローのステップである場合、2つのフローの関係はサブフローと親フローになります)。
- [グローバルコンテキスト]セクションの値には、システムプロパティおよび作成されたすべてのグローバル変数の値が表示されます。

ソース	名前	値
グローバルコンテキスト		
	CENTRAL_URL	
	execution_thread_id	
	execution_userid	studio-debugger
	run_id	100100005

デバッガーのツールバー



GUI アイテム	説明	キーボードショートカット
再生	フローを最後まで実行します。	F11
一時停止	デバッガーで実行中のフローを一時停止します。 再生 ボタンをクリックすると、一時停止された時点からフローの実行が再開されます。	ALT + P
停止	デバッガーで実行中のフローを停止します。	ALT + C

ステップオーバー 	フローをステップごとに実行します。	F5
ステップイン 	ステップのサブフローにステップインします。	F6
ステップアウト 	ステップのサブフローからステップアウトします。	F7
ステップに移動 	実行ツリーからフローステップに移動します。	
リセット 	フロー変数の値をデバッガーを開いたときの値にリセットします。	F12

複雑なフローのデバッグ

並列処理ステップを使用したフローのデバッグ

Studio では、並列処理を使用したステップ、複数のインスタンス、並列分岐ステップをデバッグします。並列処理を使用するステップを持つフローが実行時にどのように動作するかについて知るには、Studio デバッガーでフローをテストした後にステージング環境でフローを実行する以外の方法はありません。

並列分岐ステップ、マルチインスタンスステップを含むフローは、こうしたステップを含まないフローをデバッグするのと同じ方法でデバッグします。ただし、デバッガーで実行すると異なる動作となることを考慮する必要があります。

実行する操作

フローの並列分岐ステップのデバッグ

フロー実行では、デバッガーは並列分岐ステップをすべて同時に開始し、終了の順序は、Studio で予測不能な変動要因によって決まります。そのため、デバッガーは、同一のフロー変数への書き込みが競合した場合に、どのレーンが最後にフロー変数に書き込むかなどの考慮事項を予測できません。

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。

並列分岐ステップの詳細については、「[並列分岐ステップのフローの作成](#)」(285ページ)を参照してください。

フロー内のマルチインスタンスステップのデバッグ

フロー実行では、複数のインスタンスが並列で実行され、フローは、ほかのインスタンスの処理中に、1つのインスタンスのレスポンスに続くステップを続行します。

つまり、実際の条件下でテストが行われ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。

マルチインスタンスステップの詳細については、「[マルチインスタンスステップのフローの作成](#)」(289ページ)を参照してください。

リモート Central の Studio でのデバッグ

Studio リモートデバッグ機能を使用すると、HP 00 ユーザーがリモート Central でフロー実行のトラブルシューティングやデバッグを行うことができます。このため、Central 環境で問題を調べる際に Studio を使用することができます。手動で修正をデプロイし、Central フローライブラリを変更する必要はありません。

フローをデバッグする前に、Central に接続するための URL を定義します。詳細については、『HP 00 セキュリティおよびハードニングガイド』を参照してください。

既存のフロー、ローカルで変更したフロー、または新しいフローをデバッグできます。Studio では、そのフローとすべてのサブフローおよびオペレーション (既存、変更した、または新しい) がデバッグされ、Central に送信され実行されます。

フローのデバッグまたは実行はすべて、Studio 構成アイテム、Central 構成アイテム、コンテキストインスペクターの値の組み合わせを用いて、選択した Central で開始されます。

構成アイテムは、Studio または Central から上書きできます。フローのトリガー時に (フローの開始直前)、HP 00 では構成アイテムの初期値を次のように決定します。

- ユーザーがコンテキストインスペクターで構成アイテムを変更した場合、この新しい値がどのオーバーライドとも無関係に使用されます。
- 構成アイテムが (Central でも Studio でも) 上書きされず、コンテキストインスペクターで変更されなかった場合、値がそのまま使用されます。
- 構成アイテムが Central でオーバーライドされるが、Studio オーバーライドではオーバーライドされない場合、Central の値が使用されます。
- 構成アイテムが Studio オーバーライドでオーバーライドされる場合、Studio オーバーライドの値が使用されます。

デバッグ中に一時停止、再開、ステップオーバーなどを行う場合、必要に応じてコンテキストインスペクターで値を変更することができます。

注: Central でオーバーライドできる構成アイテムは、システムプロパティとシステムアカウントだけです。新しいシステムアカウントの構成アイテムのオーバーライドの設定が使用されます。

フローや構成アイテムの変更によって、Central ライブラリが影響を受けたり、変更されることはありません。これらのアイテムは揮発性なので、デバッグセッションに対してのみ可視です。

リモートデバッグセッションによって残されたトレースは、他の実行と同じように、Central 実行エクスペローラー (ドリルダウン)、およびダッシュボードに表示されます。このようなトレースのソー

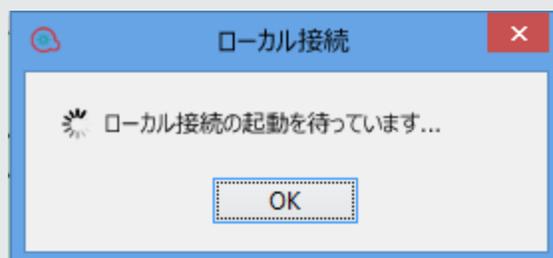
スは、最終的にはデバッガーでの実行が起点であると識別できるので、フィルターで除外できます。ドリルダウンは、Studio から送信されたフローを反映します (最終変更)。

デバッグセッション中は、コンテキストインスペクターで実行のコンテキストを変更できます。

注: リモートデバッガーから開始されたすべての実行では、状態保持レベルが拡張されています。

状態保持の詳細については、『Central ユーザーガイド』の実行ログの状態保持レベルに関する項目を参照してください。

注: 場合によっては、ローカル接続の初期化にかかる時間より早く、Studio が起動することがあります。この場合、Studio が起動してすぐにローカル接続を使用してデバッグを開始すると ([ローカル接続] ボタンに進捗状況のアイコンが表示されています)、次のメッセージが表示されます。



前提条件

• コンテンツパックの整列

リモートデバッグセッションを開始するには、適切なバージョンの Central 関連コンテンツパックを Studio ワークスペースにインポートする必要があります。このため、デバッグセッションの開始前に、デバッグ対象フローで使用するコンテンツパックを Central にデプロイしておく必要があります。

• グループエイリアス

Studio で新しいグループエイリアスを作成した場合は (それらを特定のオペレーションで使用)、Central でグループを手動で構成する必要があります。デバッガーは、グループエイリアスをマッピングするワーカーグループを決定できません。オペレーションの割り当てグループが見つからない場合は (リモートデバッグの実行中)、Central からのトリガーの場合と同様に動作します。つまり、一時停止し、問題についての確認メッセージが表示されます。

• 認証されたユーザー

リモートデバッグのアクセス許可が割り当てられたユーザーだけが、リモート Central でデバッガーをトリガーできます。詳細については、『Central ユーザーガイド』を参照してください。

注: リモートデバッグのアクセス許可を持っているユーザーにはすべてのフローが表示されませんが、システムアカウントは表示されません。システムアカウントは、適切なアクセス許可

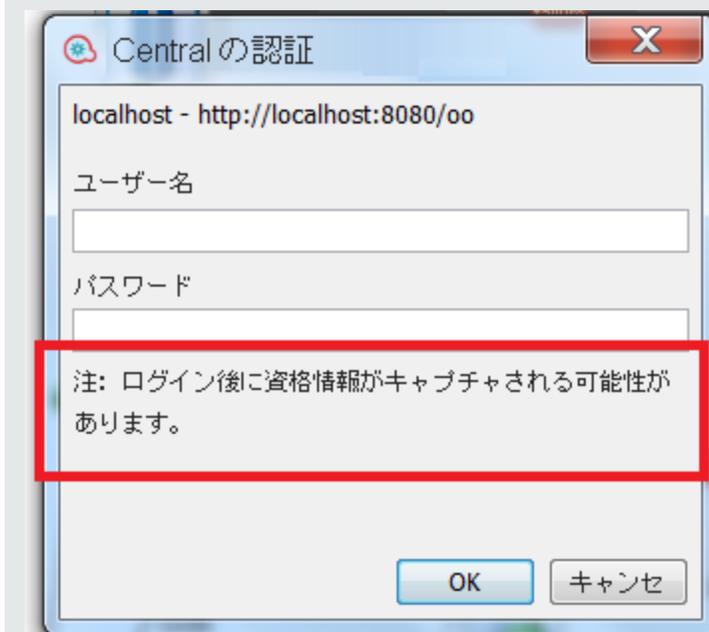
を持っているユーザーのみが使用できます。

注: この制限は、ローカル接続 (リモート Central を構成していない場合に Studio でデフォルトで使用される接続) を使用してデバッグを行うときには適用されません。

一般的なワークフロー

1. リモートデバッグのアクセス許可を含む役割を、ユーザーに割り当てます。
2. Studio のワークスペースを、Central ライブラリ (コンテンツパック、デバッグ対象のフロー) に一致するようにセットアップします。
3. 必要に応じて、Central 接続を構成します。
4. (オプション) デバッグする前に、フローを変更します。
5. デバッグオプションから接続を選択します。デバッグオプションをクリックすると、資格情報の入力が求められます。これらの資格情報は Studio に保存および記憶されます。

注: Central の [システム構成ワークスペース] > [セキュリティ] > [設定] オプションで [ログインしているユーザー資格情報のキャプチャの有効化] チェックボックスがオンになっている場合、Studio リモートデバッガーに次の警告メッセージが表示されます。



注: Central で認証が有効になっている場合、リモートデバッグのために Central にログインするときにユーザー名にコロンが含まれるユーザーでログインすることはできません。

6. Studio 構成アイテム、Central 構成アイテム、コンテキストインスペクターの組み合わせを用いて、実行が行われます。コンテキストインスペクターの詳細については、[「コンテンツの検査」](#)

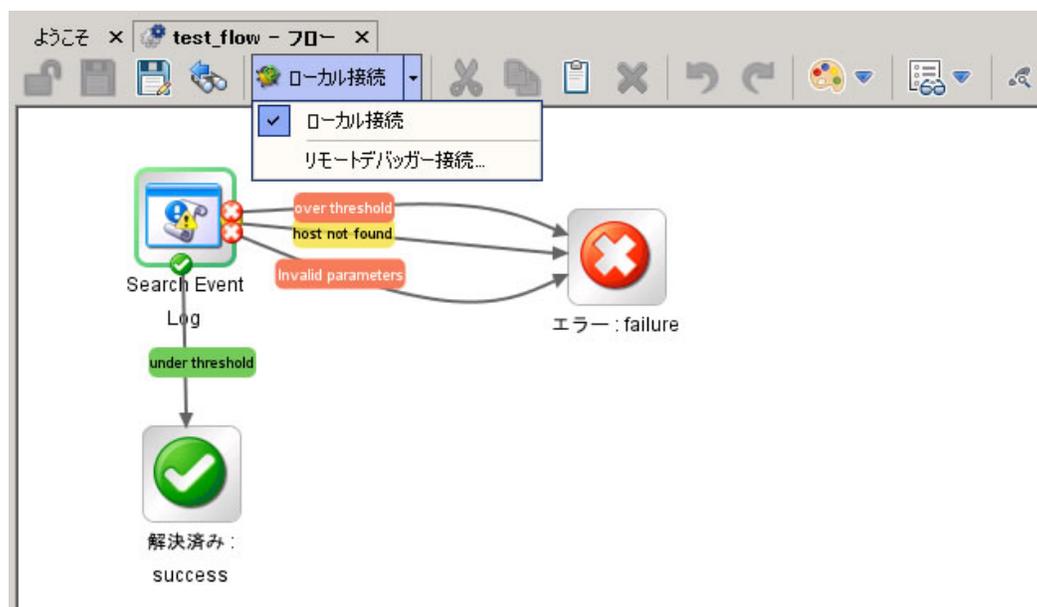
[「査」 \(311ページ\)](#)を参照してください。

7. (オプション) ブレークポイントを設定し、デバッグ中にコンテキストインスペクターで変数と構成アイテムを変更します。
8. 実行が完了します。

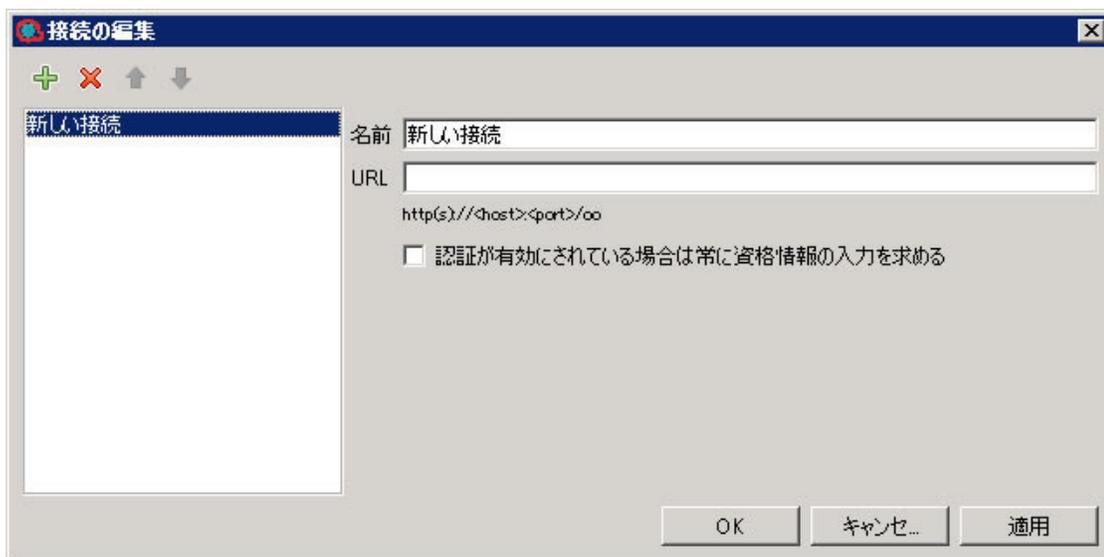
実行する操作

Central 接続の追加または編集

1. Studio で、**[構成]** > **[リモートデバッガー接続]** を選択します。
2. デバッグアイコン  をクリックし、リストからリモート Central への接続を選択します。**[ローカル接続]** がデフォルト接続です。



3. 接続を追加、編集、削除するには、**[リモートデバッガー接続]** をクリックします。**[リモートデバッガー接続]** ダイアログボックスには、使用可能な接続のリストが表示されます。これらの接続の編集、既存の接続の削除、新しい接続の追加が可能です。



- **削除:** 削除する接続を選択し、Delete キーを押すか、赤いマイナス (x) ボタンをクリックします。
- **追加:** 新しい接続を追加するには、緑の (+) ボタンをクリックします。新しい接続の名前とリモート Central の URL を入力します。

注: 接続名と URL に関する検証は行われません。これらの設定はユーザーが確認する必要があります。一意な接続名を割り当てる必要があります。

認証が有効にされている場合は常に資格情報の入力を求める: このオプションを選択し、選択した Central の認証を有効にすると、Studio は常に資格情報の入力を求めます。すでに認証されている場合は、資格情報がすでに認証フォームに入力されています。

注: Studio をリセットせずにユーザー名またはパスワードをリセットする必要がある場合、[リモートデバッガー接続] で、[認証が有効にされている場合は常に資格情報の入力を求める] チェックボックスをオンにします。

LDAP ユーザーとドメインの構成: Central への新しい接続を作成し、LDAP ユーザーを使用して接続した場合、ドメイン、ユーザー名、パスワードを選択するように求められます。

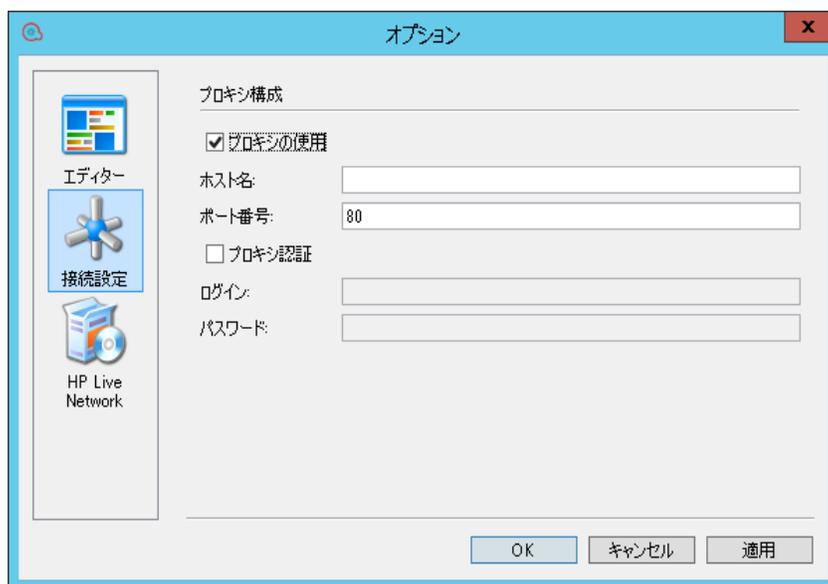
注: `studio.properties` ファイルのリモート接続は構成できません。

リモート Central でのデバッグ用プロキシの設定

Central への接続の設定時に、ホスト、ポート、ユーザー名、パスワードといった HTTP プロキシ情報を設定することができます。プロキシは基本認証スキームをサポートしています。

プロキシをセットアップするには:

1. Studio で、[構成] メニューの [オプション] > [接続設定] を選択します。[プロキシ構成] ダイアログボックスが表示されます。



2. プロキシ情報を入力します。ホストとポート番号を入力してください。完了したら、**[保存]** をクリックします。プロキシ設定がホストとポート番号上で検証されます。

リモートデバッガ接続による証明書の自動インポート

1. HTTPS を使用したリモート Central 接続を選択した場合は、SSL 証明書メッセージが表示されません。

SSL 証明書メッセージには、証明書に署名したエンティティを識別する証明書発行者の識別名および証明書の件名識別名がそれぞれ別の行に表示されます。

証明書が信頼されていない場合、この証明書を信頼するかどうかを尋ねるダイアログボックスが表示されます。

注: Studio は自己署名証明書と CA 署名証明書の両方をサポートしています。

- 証明書を信頼する場合は、**[接続]** をクリックします。Studio 信頼ストアが証明書で更新されるので、Central に接続して SSL ハンドシェイクを成功させることができます。
 - 証明書を信頼しない場合は、**[キャンセル]** をクリックします。Studio 信頼ストアに影響はなく、Central への接続に失敗します。
2. サーバーの URL と証明書が一致しない場合は、**[ホスト名の検証]** ダイアログボックスが表示されます。
 - **[信頼しない]** をクリックした場合は、接続が許可されません。
 - **[常に信頼する]** をクリックした場合、ホスト名の検証の失敗が無視され、接続が許可されます。

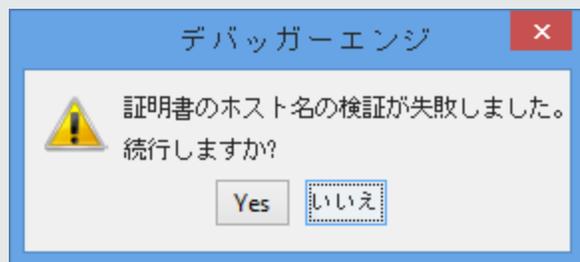
ただし、Studio を再起動すると、この問題の解決が期待されるために、ホスト名の検証メッセージがもう一度表示されます。証明書のホスト名と一致するように URL を修正するか、接続 URL に従って証明書を変更する必要があります。

注: HTTPS を介して Git サーバーに接続するときにも [SSL 証明書] ダイアログボックスが表示されます。

SVN の場合、証明書チェーンの詳細が studio.log に記録されます。

注: Studio インストールフォルダーの **studio.l4j.ini** ファイルで、**self.signed** のデフォルト値は false で、**verify.hostname** のデフォルト値は true です。

ホスト名の検証に失敗した場合、エラーメッセージが表示されます。操作の続行または中断を選択できます。



リモート Central でのフローのデバッグ

1. インポートしたコンテンツパックまたは既存のコンテンツパックからのオペレーションとフローを使用してフローを構築します。
2. フローを開きます。
3. デバッグを実行し、以前にセットアップしたリモート接続を選択します。

注: 実稼働システムの一部である Central でデバッグを実行すると、実稼働データやオペレーションに影響する可能性があります。デバッガービューにバナーが表示され、リモート接続の名前が表示されます。

実行する操作

リモート Central でのフローのデバッグ

フローをデバッグするには:

1. インポートしたコンテンツパックまたは既存のコンテンツパックからのオペレーションとフローを使用してフローを構築します。

2. フローを開きます。
3. **デバッグ**を実行し、以前にセットアップしたリモート接続を選択します。

注: 実稼働システムの一部である Central でデバッグを実行すると、実稼働データやオペレーションに影響する可能性があります。デバッガービューにバナーが表示され、リモート接続の名前が表示されます。

既存のフロー入力の再利用

実行の入力を新しい実行の入力として使用することができます。このため、失敗したフローのトラブルシューティングを行うことができます。入力を再入力したり準備する必要はありません。

現在の Central から実行 ID をロードするか指定して、フローを実行するには:

1. Studio でデバッグするフローを開きます。
2. リモート Central を選択します。
3. 実行の入力を入力します。
4. フローを実行します。
5. **[実行の入力のロード]**をクリックします。

ダイアログボックスが表示され、テキストフィールドに実行 ID が表示されます。Studio は指定された実行 ID の入力をロードします (これらの入力は、現在選択されている Central からロードされます)。入力がない場合は、実行に入力が含まれなかったか、実行が見つかりませんでした。この場合は、メッセージが表示され、現在の実行が変更されなかったことが示されます。

ロードされた入力は、現在のフローの入力と照合されます (名前で)。フローの既存の入力は、対応するロードされた値が空の場合にだけ更新されます (新しい入力は追加されません)。現在の実行の入力が変更されていない場合には、メッセージが表示されます。

コンテキストインスペクターは、変更された入力を強調表示し、新しい値を表示します。

6. 指定した実行 ID の入力で実行をトリガーできます。
種類 (単一値、値のリスト) に関係なく、入力が文字列値としてロードされます。

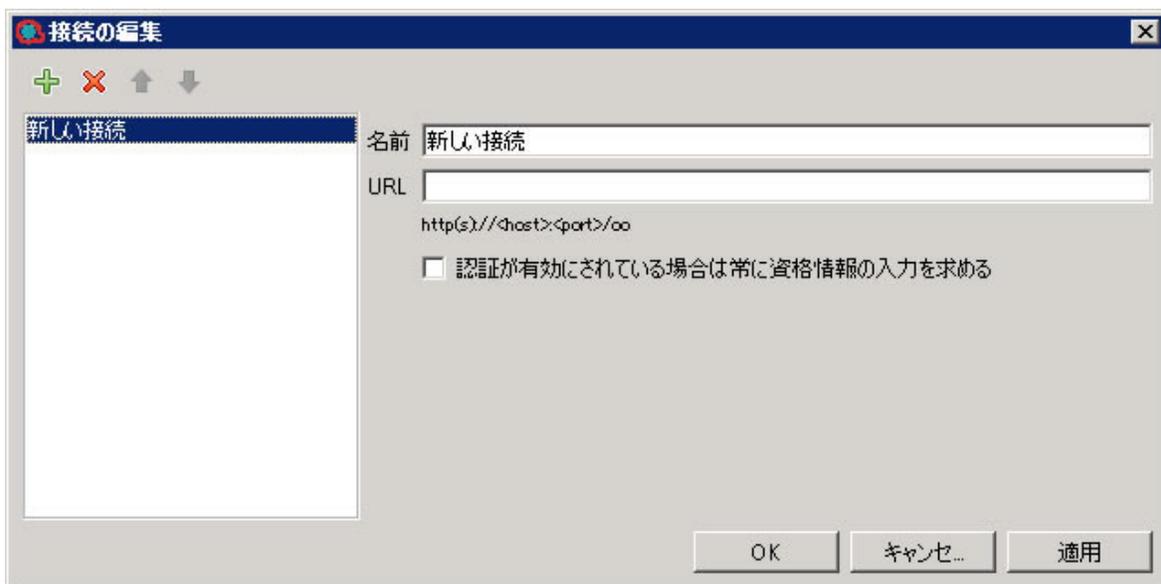
フローの再実行

前の実行と同じ入力を使用して、トリガーされたフローを再実行することができます。デバッグビューの**[再実行]**オプションは最初は無効ですが、このフローの最初の実行後に有効になります。**[再実行]**をクリックすると、ツールバーがビューに代わり、デバッグの準備が完了します。**[再生]**をクリックすると、フローの最終的なプロンプト入力を求めるメッセージは表示されなくなります。新しい実行では、前の実行のフロー入力を使用されます。

参考資料

リモートデバッガー接続

[リモートデバッガー接続] ダイアログボックスには、使用可能な接続のリストが表示されます。



GUI アイテム	説明
接続の追加 	新しい接続を追加します。新しい接続の名前と URL を入力して、[OK] をクリックします。
接続の削除 	選択した接続を削除します。
上に移動 	リスト内の選択した接続を上に移動します。
下に移動 	リスト内の選択した接続を下に移動します。

コンテンツパックのエクスポート

プロジェクトの作成

コンテンツパックの
インポート

フローの作成

フローのデバッグ

コンテンツのリリース

フローの検査が終了したら、フローをデプロイして実行するよう、コンテンツパックにリリースできるようにします。

コンテンツパックは、プロジェクトの成果です。これには、プロジェクトのエンティティと参照 ID が含まれます。コンテンツパックには、フローやオペレーションだけではなく、操作や構成アイテムも含まれています。

コンテンツパックは、Central でのデプロイメントのためにリリースする要素です。

注: 無効なフローやオペレーションは、コンテンツパックには含められません。

プロジェクトからコンテンツパックを作成すると、Studio のデフォルトで固有のバージョン番号が名前に付けられます。その後は、同じプロジェクトから別のコンテンツパックを作成するたびに、次のマイナーバージョン番号が割り当てられます。名前付けメカニズムはさまざまな条件によって変わります。詳細については、「[コンテンツパックのバージョン管理のライフサイクル](#)」(336ページ)を参照してください。

注: コンテンツパックはコンテンツアイテム内のフォルダー含むことができます。

注: Studio 10.20 以降で、構成セクション(システムプロパティ、システムアカウントなど)内のフォルダーを含むプロジェクトから作成されたコンテンツパックは、10.20 より前のバージョンの Studio にインポートすることができません。

ただし、Studio 10.20 以降で、構成セクション内のフォルダーを含まないプロジェクトから作成されたコンテンツパックは、10.20 より前のバージョンの Studio にインポートすることができます。

コンテンツパックのバージョン管理のライフサイクル

デフォルトでは、Studio は、コンテンツパックのバージョン管理を自動的に処理します。コンテンツパックが作成された後は、バージョン番号の最後の桁(通常はマイナー/マイナーセクション)が1つずつ増分され、SNAPSHOT サフィックスが現在のプロジェクトのバージョンに付加されます。つまり、プロジェクトで作業するときには、開発中のバージョンを採用しています。

例:

名前:	Project2
UUID:	ac5c7d79-1040-451c-aaf1-8dc2ee003e6b
バージョン:	1.0.2-SNAPSHOT
パブリッシャー:	Customer
説明:	

[バージョン]フィールドに新しいバージョン番号を入力してコンテンツパックのバージョンを手動で変更することもできます。

注:

- Studio では最後のマイナーバージョンのみが自動的に増分されます。メジャーバージョンを増分させる場合は (たとえば、1.0.1 と 2.0.1 に)、手動で行う必要があります。
- 標準の major.minor.minor.minor.minor...(1.x.x.x.x) 形式以外のバージョン番号を入力した場合は、別のコンテンツパックを作成したときにバージョン番号は自動的に増分されません。
- バージョン番号のマイナーの番号は変数です。つまり、1、1.2、1.2.3.4.5.06、1.1.1.1.1.1.2 などのバージョン番号も自動的に増分されます。

リビジョンコントロールを使用しないコンテンツパックのバージョン管理

Studio でバージョン管理を使用せずに作業する場合、Studio で、新しいコンテンツパックを作成するときにプロジェクトのバージョンを自動的に増分させる方法が用意されています。この方法を使用すると、連続したバージョン番号 (たとえば、1.0.1.1、1.0.1.2、1.0.1.3) が付けられた複数のコンテンツパックを簡単に作成できます。

コンテンツパックの作成ウィザードを開くと、Studio によってコンテンツパックのプロパティでバージョンが読み取られ、-SNAPSHOT サフィックスを削除したバージョン番号が [コンテンツパックのプロパティ] ステップで表示されます。

[依存関係の管理] ステップで **[CP の作成]** をクリックした後で、コンテンツパックが作成され、バージョン番号の最後の桁が自動的に 1 だけ増分されます。

-SNAPSHOT サフィックスがプロジェクトのバージョンに追加されます。これは、プロジェクトで作業するときに開発中のバージョンを採用していることを意味します。

注:

- Studio では最後のマイナーバージョンのみが自動的に増分されます。メジャーバージョンを増分させる場合は (たとえば、1.0.1 と 2.0.1 に)、手動で行う必要があります。

- 標準の major.minor.minor.minor.minor...(1.x.x.x.x) 形式以外のバージョン番号を入力した場合は、別のコンテンツパックを作成したときにバージョン番号は自動的に増分されません。
- バージョン番号のマイナーの番号は変数です。つまり、1、1.2、1.2.3.4.5.06、1.1.1.1.1.1.1.2 などのバージョン番号も自動的に増分されます。

リビジョンコントロールを使用したコンテンツパックのバージョン管理

リビジョンコントロールを使用する場合、新しいコンテンツパックを作成するたびに、Studio によって SCM (SVN または Git) タグが追加されます。このタグはリポジトリ (Git の場合) またはプロジェクト (SVN の場合) の現在の状態のスナップショットを表します。このタグは、コンテンツパック名とバージョン番号を組み込みます。

このタグを使用すると、以前のプロジェクトのバージョンに簡単にリセット/更新し、現在の環境を、コンテンツパックが作成されたときに使用されていたのと同じソースにリセットできます。

この方法で、特定のコンテンツパックのパッチを簡単に作成できます。

注: コンテンツパックリリースを作成する前に、プロジェクトでローカルでコミットされていない変更がないこと、およびプロジェクトのリビジョンが最新の状態になっていることを確認する必要があります。この2つの条件を満たしていない場合、コンテンツパックの作成は失敗します。

ただし、条件を満たしていない場合でも、一時的な "SNAPSHOT" コンテンツパックは作成できます。このコンテンツパックの場合、バージョン番号は増分されず、**古いプロジェクトのバージョンに戻す**操作は実行できません。

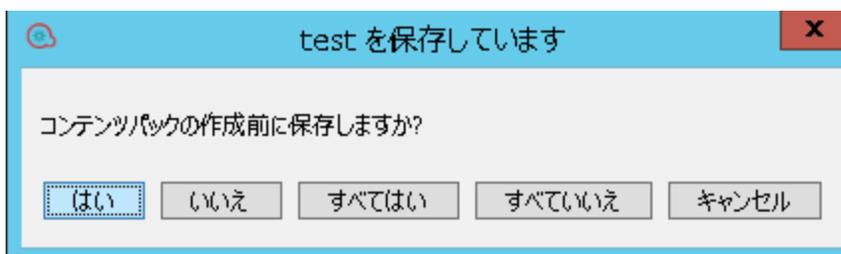
実行する操作

コンテンツパックの作成

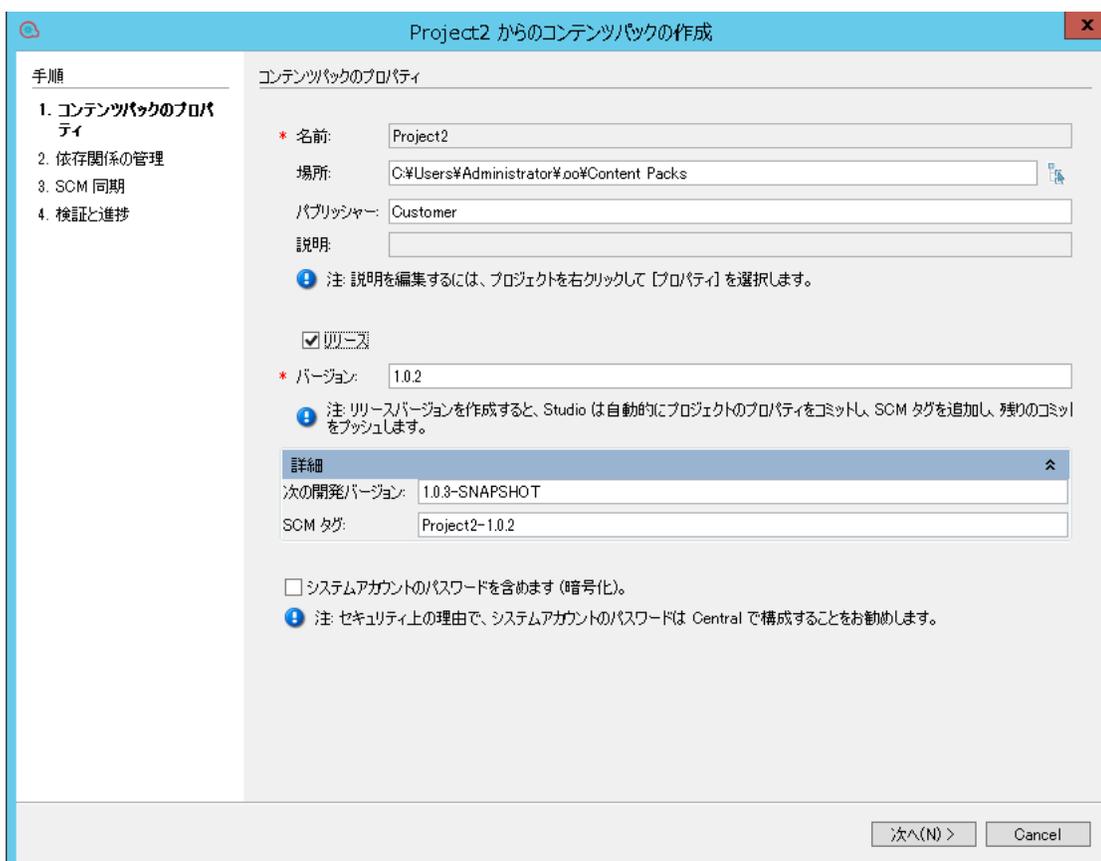
1. [プロジェクト] ペインで、コンテンツパックを作成するプロジェクトを選択します。
2. [ファイル] > [新しいコンテンツパックの作成] を選択します。

注: または、[プロジェクト] ペインの [コンテンツパックの作成]  ボタンを選択するか、プロジェクトを右クリックし、[コンテンツパックの作成] を選択します。

未保存のエディターが開いている場合は、[保存しています] ダイアログボックスで、変更を保存することができます。[すべてはい] をクリックして開いているエディターのすべての変更を保存するか、各変更に対して個別に [はい] か [いいえ] をクリックします。



3. **コンテンツパックの作成ウィザードの [コンテンツパックのプロパティ] ステップで、コンテンツパックの詳細を入力します。**



注:

- [名前] フィールドは、プロジェクト名から直接取得され、変更できません。
- 新しいプロジェクトバージョンは、コンテンツパックの作成ウィザードの [コンテンツパックのプロパティ] ステップの [詳細] タブで設定した開発バージョンです。

4. [場所] フィールドに、コンテンツパックを保存する場所を入力または参照します。デフォルトで、プロジェクトのワークスペースへのパスが選択されています。

注: デフォルトでは、ここに表示されるパスはコンテンツパックを最後に作成した場所で

す。

5. **[パブリッシャー]** フィールドにコンテンツパックのパブリッシャーを入力します。この情報は、コンテンツパックの**[プロパティ]** ページに表示されます。
6. **[説明]** フィールドを編集するには、プロジェクトツリーでプロジェクトを右クリックして**[プロパティ]** を選択します。次に、**[説明]** 領域に説明を入力します。
7. パブリッシュ用のコンテンツパックの最終バージョンを作成するには、**[リリース]** チェックボックスをオンにします。このチェックボックスをオンにすると、**[バージョン]** フィールドでバージョン番号から **-SNAPSHOT** サフィックスが削除されます。

注:

- **[リリース]** チェックボックスが無効になっている場合、ソースコントロール管理システムに接続されていないことを意味します。
- **[リリース]** チェックボックスをオンにした場合、コンテンツパックウィザードのステップ3のように**[SCM 同期]** ウィザードが追加されます。詳細については、[「コンテンツパック作成ウィザード - ステップ 3」 \(347ページ\)](#)を参照してください。

8. **[バージョン]** フィールドに、次のマイナーバージョンが表示されます。新しいコンテンツパックを作成すると、バージョン番号の最後の桁が自動的に増分されます。ただし、異なる番号を入力すれば、この番号を手動で変更できます。詳細については、[「リビジョンコントロールを使用しないコンテンツパックのバージョン管理」 \(337ページ\)](#)を参照してください。

注: コンテンツパックのリリースバージョンを作成するときには、Studio によってプロジェクトのプロパティがコミットされ、SCM タグがコンテンツパックに追加されます。

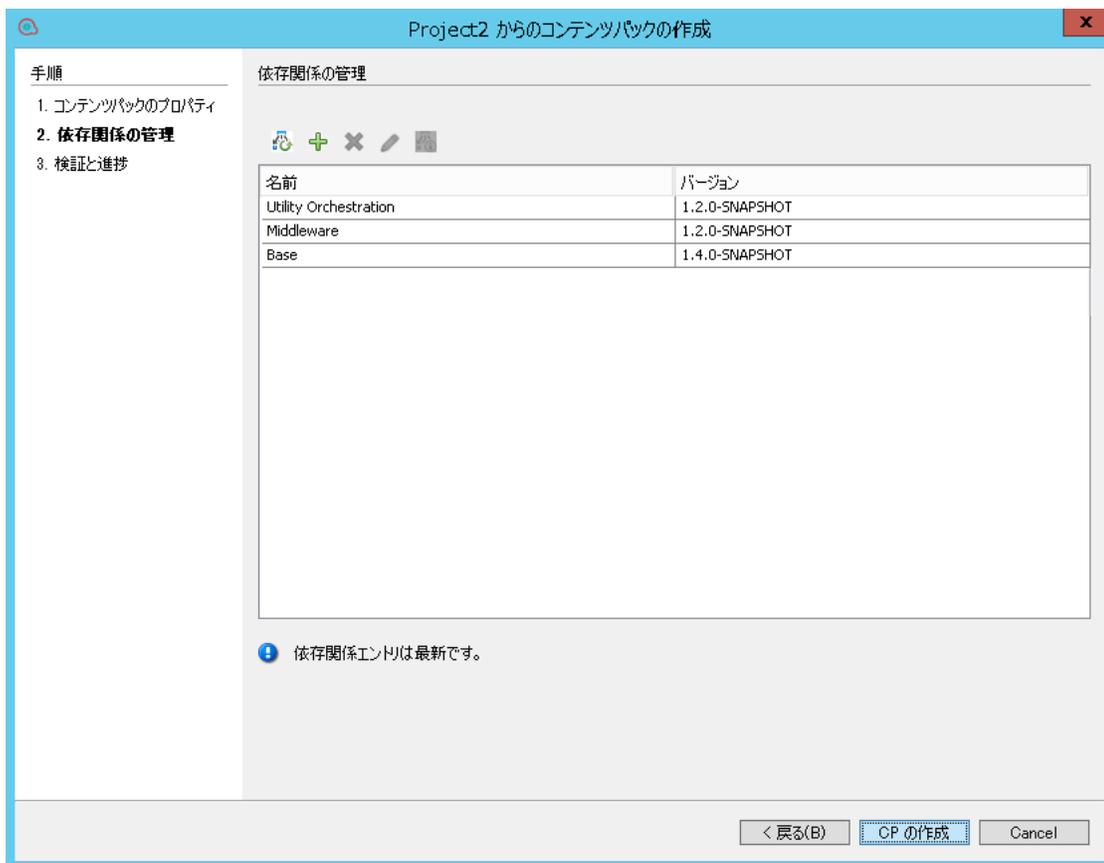
9. **ソースコントロールの場合のみ:** SVN/Git に対する変更をコミットしたときにバージョン番号が増分されるようにする場合は、**[リリース]** チェックボックスをオンにします。
10. **[詳細]** を選択して、次のフィールドを表示します。
 - a. **開発バージョン:** 拡張子 **-SNAPSHOT** が付いた次のバージョンが自動的にこのフィールドに表示されます。必要に応じてこれを変更できます。
 - b. **ソースコントロールの場合のみ: SCM タグ:** デフォルトでは、コンテンツパックをリリースするとき使用される SCM タグ名は `[<projectname>-<version>]` です。しかし、必要に応じてこれを変更し、ソースコントロールでバージョンを識別しやすいように異なる名前を入力できます。
11. コンテンツパックにシステムアカウントのパスワードを含める場合は、**[システムアカウントのパスワードを含めますか?]** チェックボックスをオンにします。コンテンツが Central にデプロイされるときに、ユーザー名とパスワードがデプロイされます。

注: パスワードはコンテンツパック内部で暗号化されます。ただし、これは安全なオプションではないためお勧めしません。代わりに、Central でシステムアカウントを構成するオプションを使用することをお勧めします。詳細については、『Central ユーザーガイド』の

「コンテンツパックの構成アイテムのセットアップ」を参照してください。

12. **[Next]** をクリックします。

依存関係が最新の状態になっていない場合、**[依存関係の管理]** ウィンドウが開きます。



13. このウィンドウで依存関係を管理し、依存関係エディターと同じように、依存関係を追加、削除、編集、更新できます。詳細については、「プロジェクトのコンテンツパックと依存関係の管理」(128ページ)を参照してください。

14. **[Next]** をクリックします。

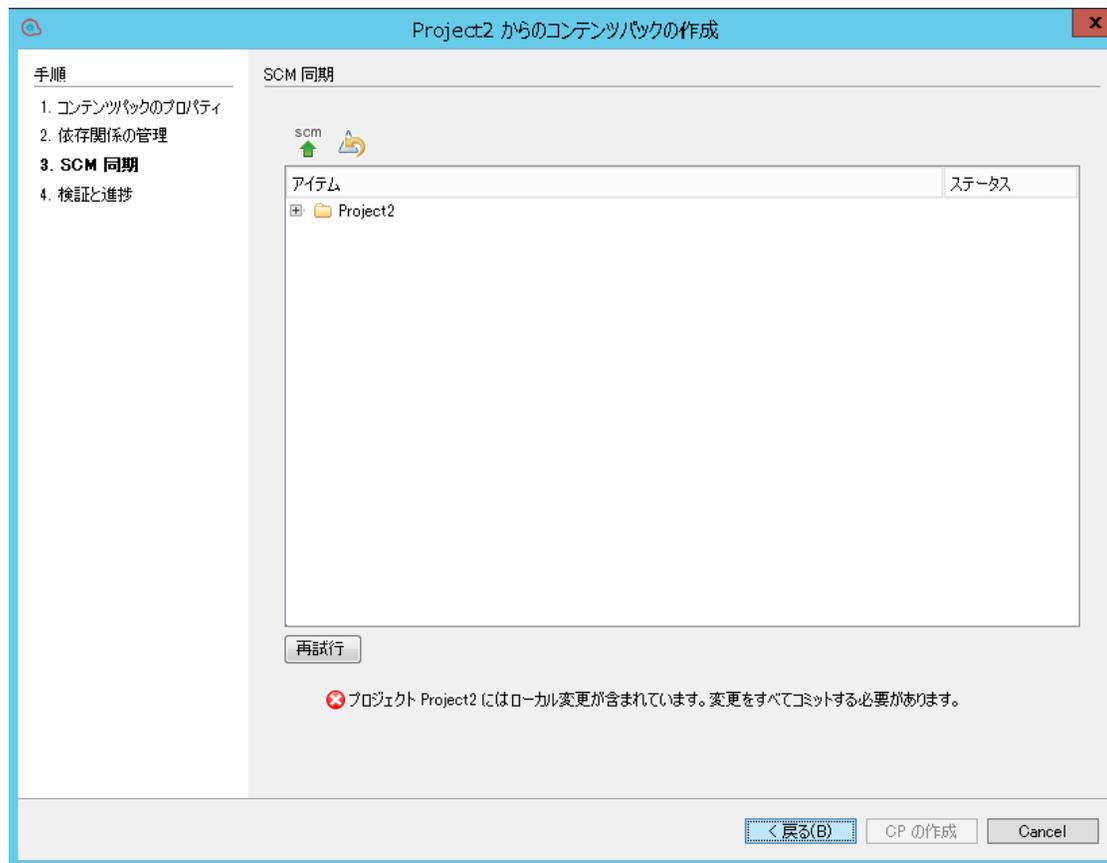
最初の手順で**[リリース]** チェックボックスをオンにした場合は、**[SCM 同期]** ウィンドウが開きます。

注: **[SCM 同期]** ページは、次の場合のみ表示されます。

- プロジェクトが同期されていない場合
- **[依存関係の管理]** ウィンドウでプロジェクトの依存関係が変更された場合
- ワークスペースにコミットされていない変更がある場合

それ以外の場合は、**[次へ]** をクリックした後に**[検証と進捗]** ウィンドウが表示されます。

ローカルバージョンとメインリポジトリの間に SCM 同期の問題がある場合は、[SCM 同期] ウィンドウにそれらの問題が表示されます。続行する前にこれらの問題を解決する必要があります。「[「Git ソース管理システムの使用」\(84ページ\)](#)」に記載されているように、SCM ツールバーを使用して変更をコミット、プッシュ、または元に戻すことができます。



注: [依存関係の管理] ステップで依存関係を変更した場合、それらも [SCM の同期] ステップで変更として表示されます。

SCM の問題を解決した後で、[再試行] をクリックします。他に問題がない場合は、次のメッセージが表示されます。

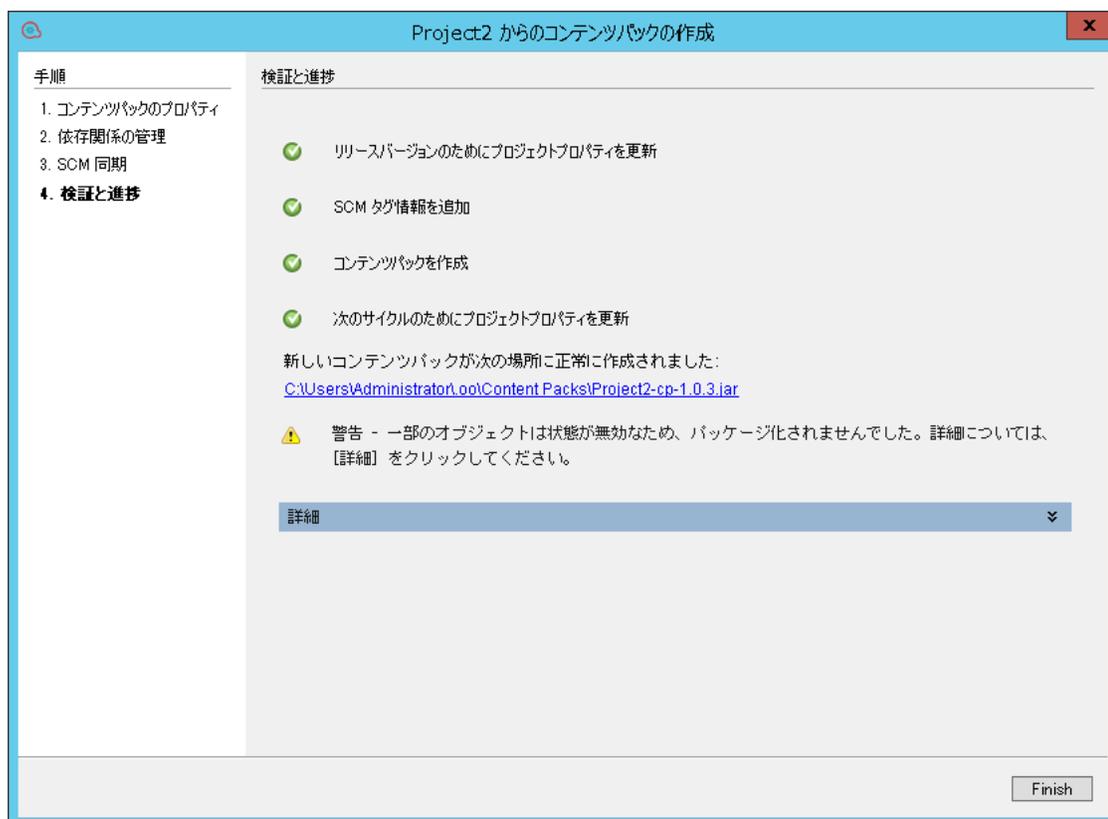
コンテンツパックの作成を続行できます。

15. [CP の作成] をクリックします。

現在のブランチに対して実行したすべてのコミット操作がメイン Git リポジトリにプッシュされ、コンテンツパックが作成されます。

プロジェクト内に無効なフローまたは依存関係の警告がある場合、Studio によってそれらが示されます。[詳細] をクリックして警告を表示します。

無効なフローはコンテンツパックにパッケージ化されません。



コンテンツパックが指定した場所に作成されます。

16. 新しいコンテンツパックが作成された後に、メッセージが表示され、コンテンツパックが作成された場所へのリンクが表示されます。リンクをクリックするとコンテンツパックにアクセスできます。
17. [完了] をクリックして、ウィザードのウィンドウを閉じます。

新しいコンテンツパックは、デプロイ、実行、別のプロジェクトへのインポートができます。

注: コンテンツパックが作成された後に、次のように **contentpack.properties** ファイルにバージョン番号が含まれています。

- **Content Pack: contentpack.properties** にパブリッシュされたバージョンが含まれています。
例: 1.8.3
- **Project:contentpack.properties** に次のスナップショットのバージョンが含まれています。例:
1.8.4-SNAPSHOT

- バージョン番号は pom.xml ファイルに保存されます。pom.xml ファイルには、プロジェクトの依存関係の情報も含まれています。

プロジェクトのバージョンの表示/編集

1. [プロジェクト] ペインで、バージョンを表示/編集するプロジェクトを選択します。

2. [プロジェクト]ウィンドウで[プロパティ]オプションを選択します。

または:

プロジェクトを右クリックして[プロパティ]を選択します。

参考資料

コンテンツパック作成ウィザード - ステップ 1

Project2 からのコンテンツパックの作成

手順

1. コンテンツパックのプロパティ
2. 依存関係の管理
3. SCM 同期
4. 検証と進捗

コンテンツパックのプロパティ

* 名前: Project2

場所: C:\Users\Administrator\%oo%\Content Packs

パブリッシャー: Customer

説明:

注: 説明を編集するには、プロジェクトを右クリックして [プロパティ] を選択します。

リリース

* バージョン: 1.0.2

注: リリースバージョンを作成すると、Studio は自動的にプロジェクトのプロパティをコミットし、SCM タグを追加し、残りのコミットをプッシュします。

詳細

次の開発バージョン: 1.0.3-SNAPSHOT

SCM タグ: Project2-1.0.2

システムアカウントのパスワードを含めます (暗号化)。

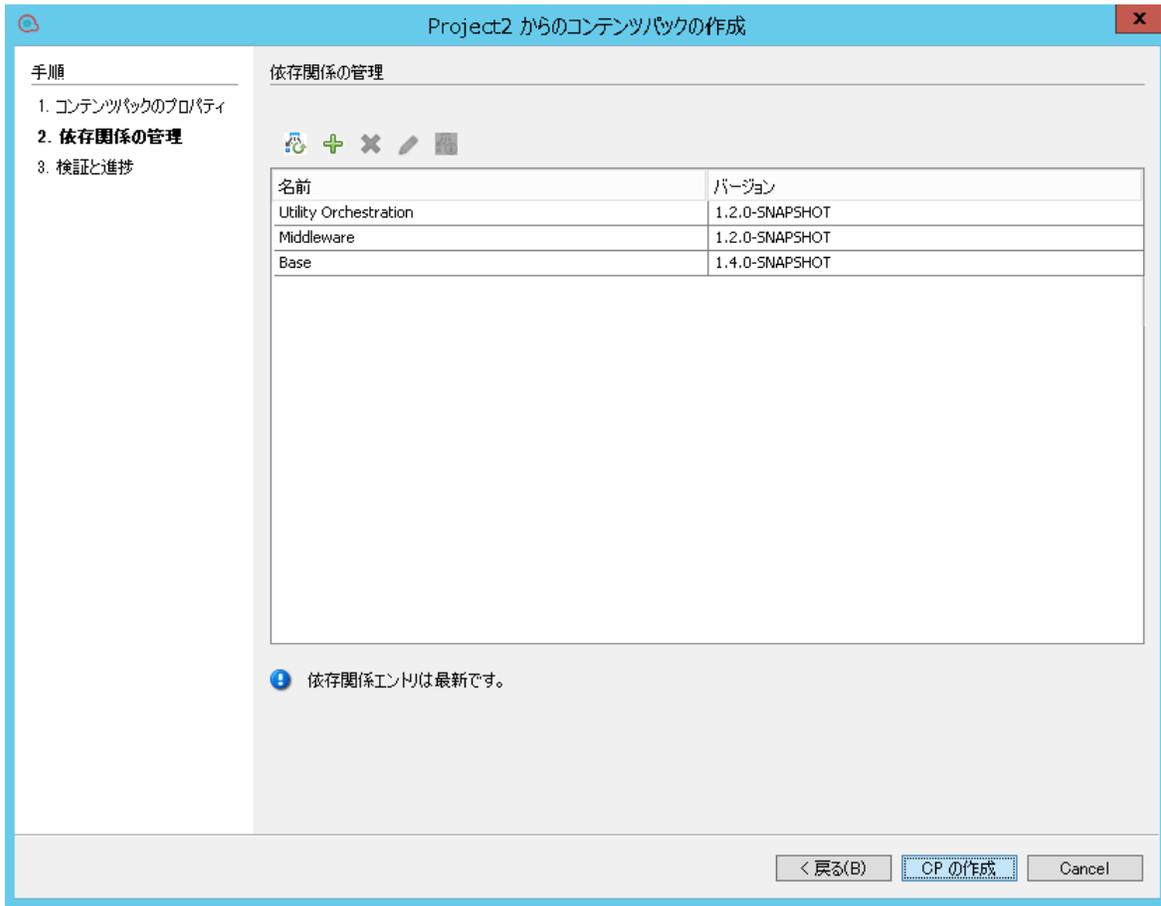
注: セキュリティ上の理由で、システムアカウントのパスワードは Central で構成することをお勧めします。

次へ(N) > Cancel

GUI アイテム	説明
名前	コンテンツパックの名前は、プロジェクト名から取得されます。このフィールドは読み取り専用です。
場所	コンテンツパックを保存する場所を入力または参照します。デフォルトで、プロジェクトのワークスペースへのパスが選択されています。

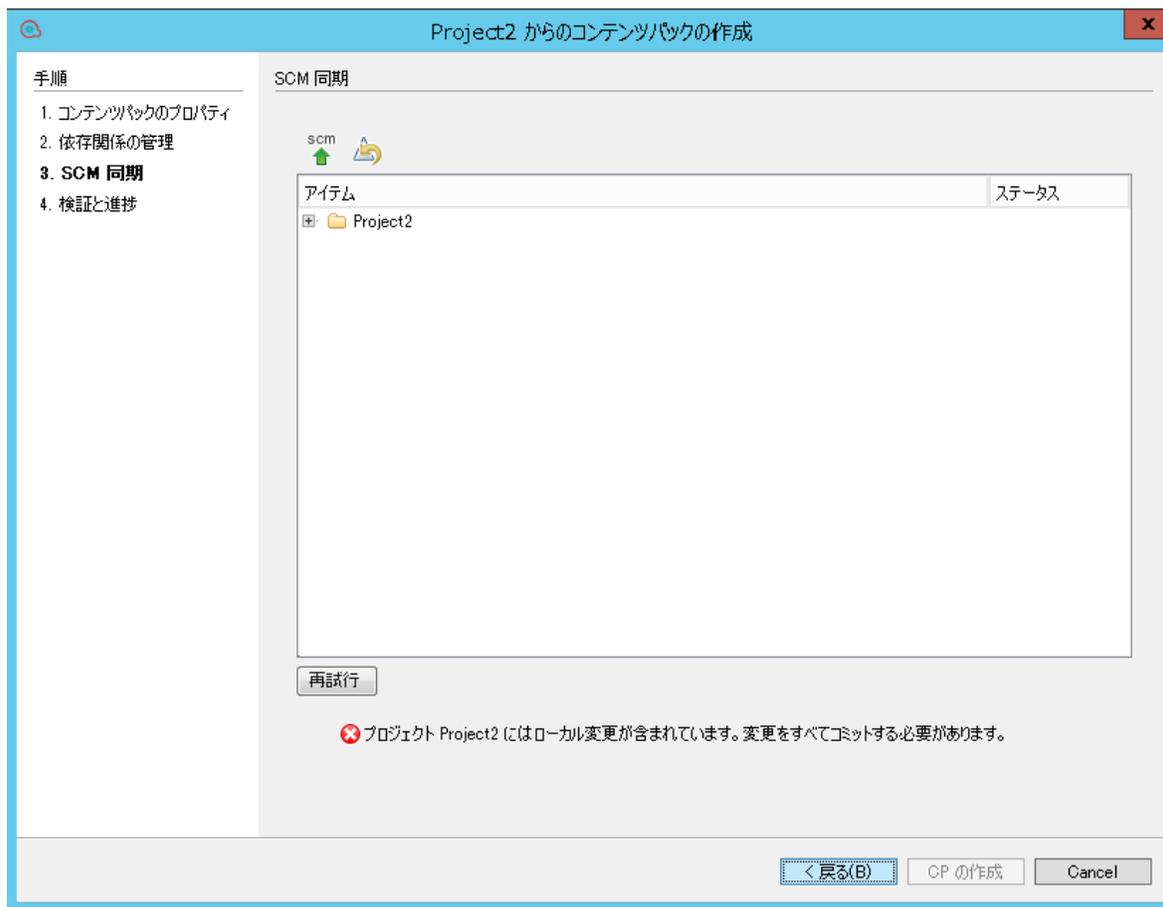
<p>パブリッシャー</p>	<p>コンテンツパックのパブリッシャーを入力します。この情報は、コンテンツパックの[プロパティ]ページに表示されます。</p>
<p>説明</p>	<p>[説明] フィールドを編集するには、プロジェクトツリーでプロジェクトを右クリックして[プロパティ]を選択します。次に、[説明] 領域に説明を入力します。</p>
<p>リリース</p>	<p>SVN/Git に対する変更をコミットしたときにバージョン番号が増分されるようにする場合は、このチェックボックスをオンにします。</p> <p>これにより、プロジェクト (SVN) またはリポジトリ (GIT) の現在の状態を示すタグも作成されます。</p> <p>このチェックボックスをオンにした場合、[SCM 同期] ステップがコンテンツパックの作成ウィザードに追加されます。</p>
<p>バージョン</p>	<p>コンテンツパックのバージョンを入力します。この情報は、コンテンツパックの[プロパティ]ページに表示されます。</p>
<p>詳細</p>	<p>矢印を選択して、SCM フィールドを表示します。</p>
<p>開発バージョン</p>	<p>拡張子 -SNAPSHOT が付いた次のバージョンが自動的にこのフィールドに表示されます。必要に応じてこれを変更できます。</p>
<p>SCM タグ</p>	<p>SCM タグ名として使用するテキストを入力します。このフィールドを使用して、ソースコントロール内のバージョンを識別できません。</p>
<p>システムアカウントのパスワードを含めますか?</p>	<p>コンテンツパックにシステムアカウントのパスワードを含める場合は、このチェックボックスをオンにします。これらのパスワードは暗号化されます。このチェックボックスをオフにした場合は、パスワードはコンテキストボックスには含められません。</p> <p>注: セキュリティ上の理由から、Central でシステムアカウントのパスワードを構成することをお勧めします。</p>

コンテンツパック作成ウィザード - ステップ 2



GUI アイテム	説明
	新しい依存関係を追加します。
	リストから依存関係を削除します。
	依存関係のプロパティを編集します。
	依存関係を更新します。

コンテンツパック作成ウィザード - ステップ 3



GUI アイテム	説明
	すべての変更をコミット: [SCM 変更] ペインですべての変更をコミットします。変更があった場合にのみ使用可能です。
	すべての変更を元に戻す: [SCM 変更] ペインで表示されるすべての変更を元に戻します。

フローとオペレーションの管理

プロジェクトのライブラリに多数のフローとオペレーションを格納できます。この章では、アイテムを検索、コピーする方法、アイテムをブックマークする方法、アイテムの使用状況を確認する方法、および新しいオペレーションを作成する方法といった、このライブラリを管理する方法について説明します。

オペレーションの作成

Studio でオペレーションを作成する方法には次の 3 つがあります。

- 既存オペレーションをコピーおよび編集する。
- オペレーションを既存のプラグインからインポートする。
- Java でアクションプラグインを作成し、そのアクションプラグインを Studio にインポートする。

アクションプラグインからのオペレーションの作成

アクションプラグインは、IActions または @Actions を含む jar ファイルです。アクションプラグインを Studio にインポートし、アクションプラグインのいずれかの操作からオペレーションを作成することができます。

アクションプラグインは、複数の操作を含めることができ、それぞれの操作から 1 つずつオペレーションを作成することができます。

アクションプラグインの開発の詳細については、『アクション開発者ガイド』を参照してください。

アクションプラグインから作成されたオペレーションのコピー

アクションプラグインをインポートして作成したオペレーションをコピーした場合、コピーされたオペレーションは元のオペレーションの参照を保持したままです。アクションプラグインがアップグレードされた場合、元のオペレーションが新しいバージョンを呼び出すように更新されると、コピーされたオペレーションは自動的に更新されます。これを「ソフトコピー」といいます。

ソースオペレーション (このオペレーションのコピー元) は、[詳細] タブに表示されます。

コピーされたオペレーションで自動的に更新されるアイテムは、アクションプラグインへのリンクのみです。元のオペレーションの入力、出力、変数、スクリプトレットなどは、コピーでは更新されません。

有効なオペレーション

有効なオペレーションには次の条件があります。

- UUID ジェネレーターなどの入力を必要としないオペレーションがあること。
- オペレーション結果を表す有効な式へマッピングされるレスポンスが少なくとも1つあること。

新規オペレーションが無効または未完成の場合は、オペレーションの名前が[プロジェクト]ペインに波線の下線付きで表示されます。未完成のオペレーション名の上にカーソルを合わせると、未完成の状態を示すツールチップが表示されます。

ベストプラクティス

作成したオペレーションを使用してフローを作成するほかの作成者にそのオペレーションをわかりやすくするには、オペレーションの[説明]タブに次の情報を追加します。

- オペレーションの内容の説明。
- オペレーションが必要とする入力。入力に必要なデータや適切なデータ形式を作成者が検索できる場所も含む。
- レスポンス。各レスポンスの意味を含む。
- 結果フィールド。各結果フィールドで与えられるデータの説明を含む。
- 次のような、実装に関する追加のメモすべて。
 - プラットフォームまたはアプリケーションのサポート。バージョン情報を含む。
 - フローが相互作用するアプリケーションまたは Web サービス API。
 - その他の環境要件または使用要件

実行する操作

Studio でのオペレーションのコピーと編集

重要: プロジェクト内のオペレーションを削除、作成、または名前変更する必要がある場合は、ファイルシステムでアイテムを削除、作成、または名前変更するのではなく、Studio 内から実行してください。

1. [依存関係] ペインまたは[プロジェクト] ペインで、コピーするオペレーションを選択します。
2. [編集] > [コピー] を選択します。

3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集] > [貼り付け] を選択します。オペレーションは、新しいオブジェクトとして扱われます。
4. オペレーションを検索用途のためにカテゴリに割り当てるには、[カテゴリの割り当て] をクリックし、リストからカテゴリを選択します。
5. 入力を作成するには、[入力] タブをクリックし、[入力の追加] をクリックします。
6. 表示されたダイアログボックスに名前を入力し、[OK] をクリックします。
 - 入力の内容と使用方法の詳細については、[「入力の作成」\(205ページ\)](#)を参照してください。
 - 入力用のデータソースの定義の詳細については、[「入力ソースの指定」\(214ページ\)](#)を参照してください。
7. 出力データを追加し、定義します。

出力データの追加と使用の詳細については、[「オペレーションの出力の設定」\(243ページ\)](#)を参照してください。
8. 必要なレスポンスを作成し、結果をそのレスポンスにマッピングします。

オペレーションにどのレスポンスが選択されるのかを定めるルールの定義については、[「レスポンスの設定」\(233ページ\)](#)を参照してください。
9. [説明] タブをクリックして、テキストボックスに説明を入力します。
10. [OK] をクリックします。

オペレーションがベースとするソースオペレーションの変更

既存のオペレーションをコピーしてオペレーションを作成した場合、ソースオペレーションを変更できます。コピーしたオペレーションは、ソースとして選択したオペレーションのコピーになります。

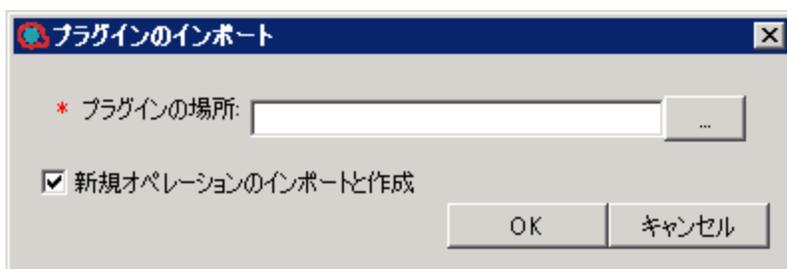
1. コピーしたオペレーションを開き、[プロパティ] を選択します。
2. フォルダーの[プロパティ]シートで、[詳細] タブをクリックします。
3. [ソースオペレーション] で、[選択] ボタンをクリックします。
4. [オペレーションの選択] ダイアログボックスで、コピー元のソースオペレーションに移動して選択し、[OK] をクリックします。
5. 必要に応じて、変更内容に合わせてオペレーションの名前を変更します。
6. 入力の値の割り当てに必要な変更を確認して実行し、以前のオペレーションの入力と新しいオペレーションの入力の違いが反映されるようにします。

アクションプラグインのインポートによるオペレーションの作成

アクションプラグインからオペレーションを作成する最も直接的な方法は、アクションプラグインのインポートと同時にオペレーションを作成する方法です。

1. アクションプラグインを作成してパックし、開発したアクションを含めます。アクションプラグインの開発方法の詳細については、『アクション開発者ガイド』を参照してください。
2. Studio で新しいオペレーションの作成先のフォルダーを右クリックして[プラグインのインポート]を選択します。

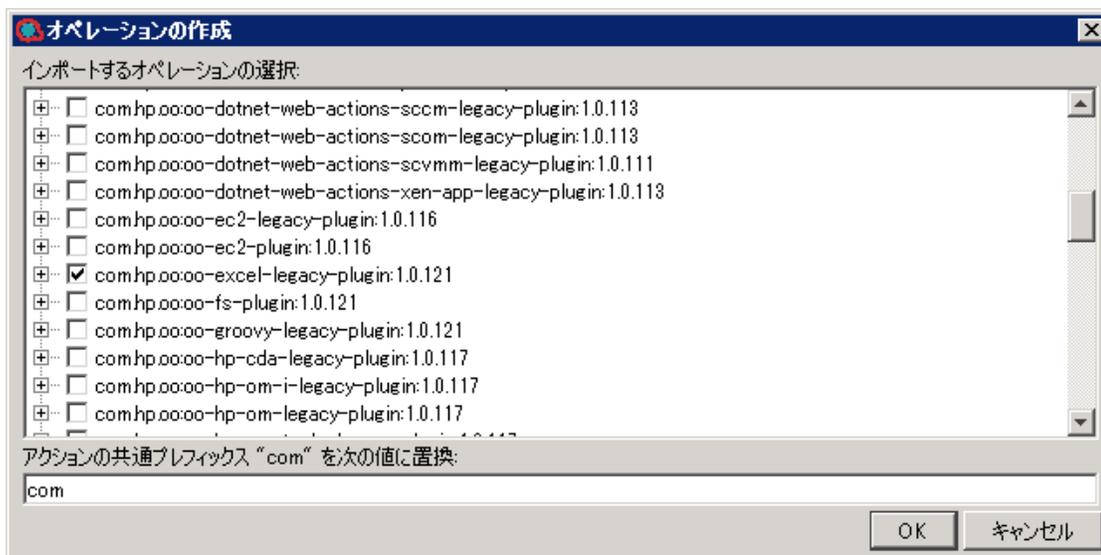
3. [プラグインのインポート] ダイアログボックスで、参照ボタンをクリックして、インポートする HP 00 プラグインを参照して選択します。



注: JAR ファイルまたは POM ファイルによって、1つのプラグイン (Maven アーティファクト) をインポートできるようになりました。プラグインの JAR と ROM の両方のファイルが同じ場所になければなりません。

すでにデプロイされているプラグインをインポートした場合、新しいプラグインが既存のものを置き換えます。

4. [新規オペレーションのインポートと作成] チェックボックスをオンにして [OK] をクリックします。
5. [オペレーションの作成] ダイアログボックスで、必要なアクションを含むプラグインを展開し、オペレーションの作成元として使用するアクションを選択します。



注: プラグインに複数のアクションが含まれている場合、複数のアクションを選択して複数のオペレーションを作成することができます。

選択したそれぞれのアクションに対する新しいオペレーションが、右クリックしたフォルダーに作成されます。

それぞれのオペレーションで、アクションプラグインについての情報が、オペレーションの [プロパティ] ウィンドウの [入力] タブの上部の [Operation フィールド] セクションに表示されます。

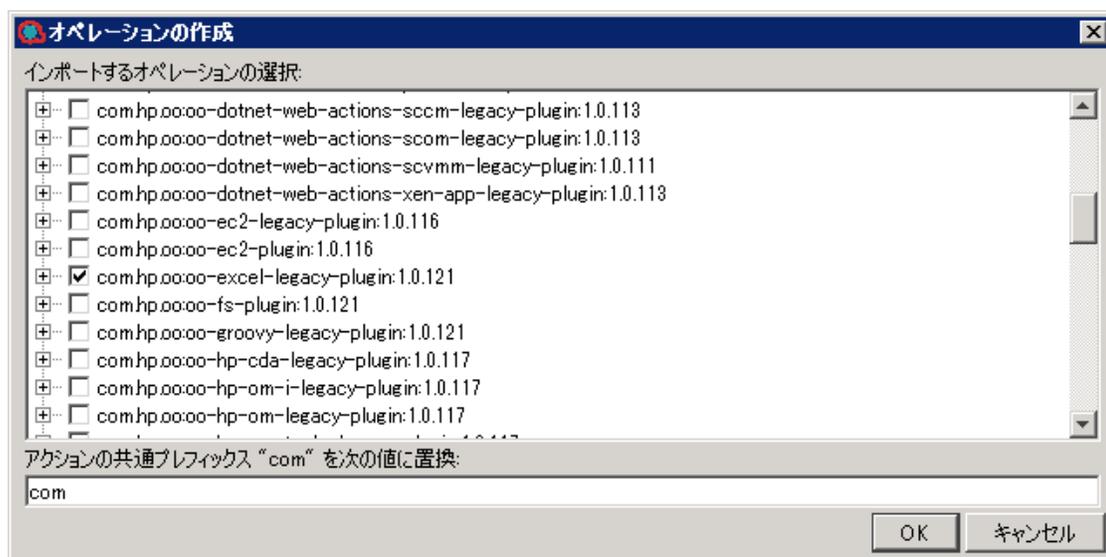
Operation フィールド	
グループ ID:	com.hp.oo
アーティファクト ID:	oo-excel-legacy-plugin
バージョン:	1.0.121
アクション名:	com.iconclude.content.actions.excel.ops.ModifyExcelData

6. オペレーションを保存します。

インポートしたアクションプラグインからのオペレーションの作成

アクションプラグインを Studio のリポジトリにインポートしたら、その中のアクションからオペレーションを作成できます。

1. Studio で新しいオペレーションの作成先のフォルダーを右クリックして [新規] > [オペレーション] を選択します。
2. Studio のリポジトリからプラグインの場所を参照し、[OK] をクリックします。
3. [オペレーションの作成] ダイアログボックスで、オペレーションの作成元のアクションを選択します。



注: プラグインに複数のアクションが含まれている場合、複数のアクションを選択して複数のオペレーションを作成することができます。

選択したそれぞれのアクションに対する新しいオペレーションが、右クリックしたフォルダーに作成されます。

それぞれのオペレーションで、アクションプラグインについての情報が、オペレーションの [プロパティ] ウィンドウの [入力] タブの上部に表示されます。

アクションプラグインのインポート

Studio のリポジトリにアクションプラグインをインポートだけし、後から自分またはほかの作成者がそのアクションプラグインからオペレーションを作成することも可能です。

1. Studio で、[ファイル]>[プラグインのインポート]を選択します。
2. [プラグインのインポート]ダイアログボックスで、Studio のローカルの Maven リポジトリにインポートする HP 00 プラグインを参照して選択します。
3. [OK]をクリックします。作成者はこのプラグインからオペレーションを作成できます。詳細については、「インポートしたアクションプラグインからのオペレーションの作成」を参照してください。

手動オペレーションの作成

手動オペレーションは、操作を選択できるオペレーションです。ユーザーが、実行時に操作を選択する必要があります。

手動オペレーションを作成するには、手動オペレーションテンプレートをベースコンテンツからコピーして、ユーザーが選択できる操作を定義します。

1. [依存関係] ペインで、コンテンツパック内にある手動オペレーションテンプレートを選択します (base-cp/Library/Utility Operations/Manual など)。
2. [編集]>[コピー]を選択します。
3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集]>[貼り付け]を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。
4. オペレーションのプロパティで、ユーザーが選択できる操作を追加します。

表示オペレーションの作成

表示オペレーションとは、ポップアップのプロンプトメッセージに情報を表示しますが、ほかの操作が実行されないオペレーションです。ユーザーは、実行時に[続行]を選択する必要があります。

表示オペレーションを作成するには、ベースコンテンツにある表示オペレーションテンプレート Library/Utility Operations/Display Message をコピーして、ユーザーに表示する情報を定義します。

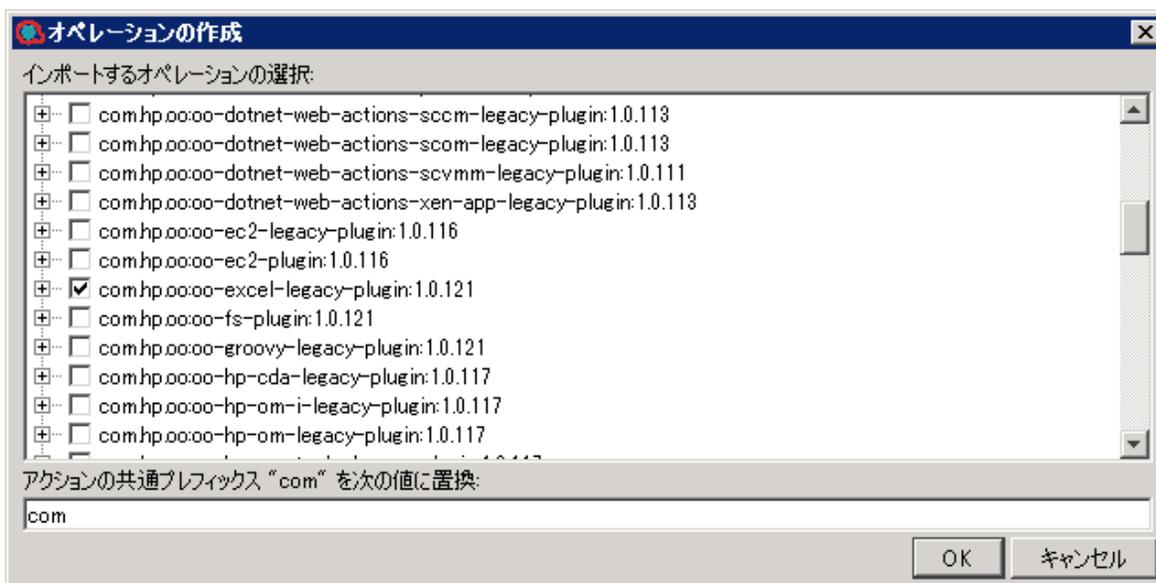
プロンプトメッセージに変数を含めることができます。たとえば、前のステップが終了した時間をユーザーに通知するよう、メッセージに日付/時刻の変数 (`${dateTime}`) を含めることができます。

1. [依存関係] ペインで、表示オペレーションテンプレートを選択します。
2. [編集]>[コピー]を選択します。
3. コピーを貼り付ける場所をプロジェクトツリーで選択し、[編集]>[貼り付け]を選択します。このオペレーションは、新しいオブジェクトとして扱われ、コピー元のコンテンツパックからデタッチされます。
4. オペレーションのプロパティシートで、[表示] タブを選択します。
5. ステップのインスペクターで [表示] タブをクリックします。
6. [このステップの実行前に常にユーザーに確認する] チェックボックスをオンにします。

7. [プロンプトタイトル] ボックスに、プロンプトのラベル (最大 128 文字) を入力します。
8. [プロンプトの幅] ボックスに、プロンプトの幅をピクセル単位で入力します。
9. [高さ] ボックスに、プロンプトの高さをピクセル単位で入力します。
10. [プロンプトテキスト] ボックスに、ユーザーへのメッセージを入力します。
11. [OK] をクリックし、変更を保存します。

参考資料

[オペレーションの作成] ダイアログボックス



GUI アイテム	説明
インポートするオペレーションの選択	必要なアクションを含むプラグインを展開し、オペレーションの作成元として使用するアクションを選択します。

ステップのインスペクター > [表示] タブ

[オペレーションのプロパティ] シートの [表示] タブで、ユーザーに表示するユーザープロンプトを作成できます。

ステップ名:

入力 | 結果 | 表示 | 説明 | 詳細 | スクリプトレット

このステップの実行前に常にユーザーに確認する

プロンプトタイトル:

プロンプトの幅: 高さ:

プロンプトテキスト

GUI アイテム	説明
このステップの実行前に常にユーザーに確認する	ステップを実行するたびにプロンプトウィンドウを表示するには、チェックボックスをオンにします。
プロンプトタイトル	プロンプトウィンドウのタイトルバーに表示するラベル(最大 128 文字)を入力します。
プロンプトの幅	プロンプトウィンドウの幅をピクセル単位で入力します。
高さ	プロンプトウィンドウの高さをピクセル単位で入力します。
プロンプトテキスト	プロンプトウィンドウの本文に表示するメッセージを入力します。メッセージに変数を含めることができます。たとえば、 <code>\${dateTime}</code> のようになります。

プラグインのインポート

プラグインのインポート ✕

* プラグインの場所: ...

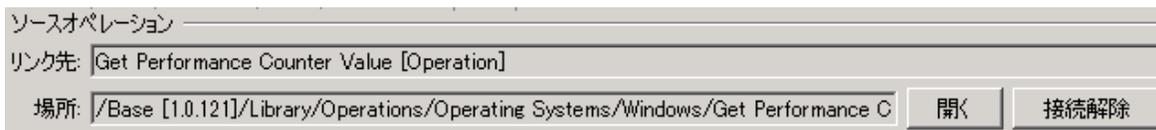
新規オペレーションのインポートと作成

OK キャンセル

GUI アイテム	説明
----------	----

プラグインの場所	インポートする HP OO プラグインを参照して選択します。
新規オペレーションのインポートと作成	このチェックボックスを選択して、インポートしたプラグインから新しいオペレーションを作成します。

オペレーションのプロパティ: [詳細] タブ



GUI アイテム	説明
リンク先	ソースオペレーション (選択したオペレーションのコピー元) が表示されます。
場所	ソースオペレーションの名前が表示されます。
選択	別のソースオペレーションを選択できます。
開く	ソースオペレーションの [プロパティ] シートが開きます。
接続解除	オペレーションを親プラグインからデタッチします。

フローまたはオペレーションの検索

目的のフローまたはオペレーションを見つけるにはいくつかの方法があります。

- [プロジェクト] ペインおよび [依存関係] ペインでフォルダーを参照する
- フォルダー、フロー、オペレーションの説明を表示する
- [検索] タブを使用して検索を実行する
- [フローに移動...] オプションを使用して検索を実行する

実行する操作

フォルダーを参照してフローまたはオペレーションを見つける

フローやオペレーションを探す最も簡単な方法は、フォルダーを参照することです。

フォルダーに適切な名前が付けられ、適切な構造になっている場合は、目的のフローやオペレーションが探しやすくなります。

説明を使用してフローまたはオペレーションを見つける

オペレーションやフローの説明を表示して、目的のものかどうかを確認できます。

- ・ オペレーションの説明を表示するには、作成ペインでオペレーションを開いて **[説明]** タブをクリックします。
- ・ フローの説明を表示するには、作成ペインでフローを開いて **[プロパティ]** (ペインの下部) をクリックし、**[説明]** タブをクリックします。

注: または、**[プロジェクト]** ペインまたは **[依存関係]** ペインで、フローまたはオペレーションを右クリックして **[プロパティ]** を選択します。

ドキュメントを生成してフローまたはオペレーションを見つける

ドキュメントの生成機能では、複数のフローやオペレーションについてのこれらの情報を1つにまとめることができます。ドキュメントの生成機能の詳細については、「[フローおよびオペレーションに関するドキュメントの生成](#) (368ページ)を参照してください。

[検索] タブを使用した操作または構成アイテムの検索

[検索] ペインを使用して、ライブラリ全体をフルテキスト検索できます。名前やそれ以外のフィールドプロパティを検索することにより、フロー、オペレーション、または構成アイテムを検索できます。

1. Studio ウィンドウの下部にある **[検索]** タブをクリックして、**[検索]** ペインを開きます。
2. **[検索]** リストで、検索対象のフィールドを選択します。名前、説明、Lucene クエリ、およびすべてのフィールドによる検索条件があります。

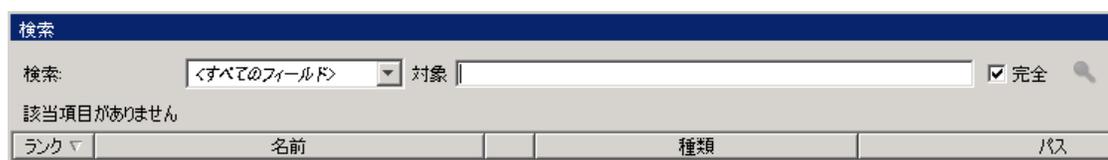
注: 検索にすべてのフィールドを含めるには、**[検索]** リストを **[<すべてのフィールド>]** に設定したままにします。

3. **[対象]** テキストボックスに、検索するテキストを入力します。

検索文字列に空白が含まれている場合、完全一致検索または空白を含む検索のいずれかを指定できます。完全一致検索では、文字列全体を、空白がある場合はそれも含めて、1つの検索値として扱います。空白を挿入すると、空白で区切られた文字列すべてを検索します。

注: 入力した検索文字列の検索は、大文字と小文字を区別しません。

4. 検索での空白の扱い方を定義するには、次のようにします。
 - ・ 空白を1つの検索文字列の一部として扱うよう指定するには、**[完全]** チェックボックスを選択します。
 - ・ 空白を、検索文字列を順に区切るセパレーターとして扱うには、**[完全]** チェックボックスを選択解除します。
5. 検索ボタン  をクリックします。



検索

検索: 対象 完全 

該当項目がありません

ランク	名前	種類	パス
-----	----	----	----

6. **[説明]** タブのテキストを確認し、目的のオペレーションを指定します。

検索結果には、オペレーションの **[説明]** タブから説明が取得され、オペレーションを最大限に活用するための、次のような重要な情報が表示されます。

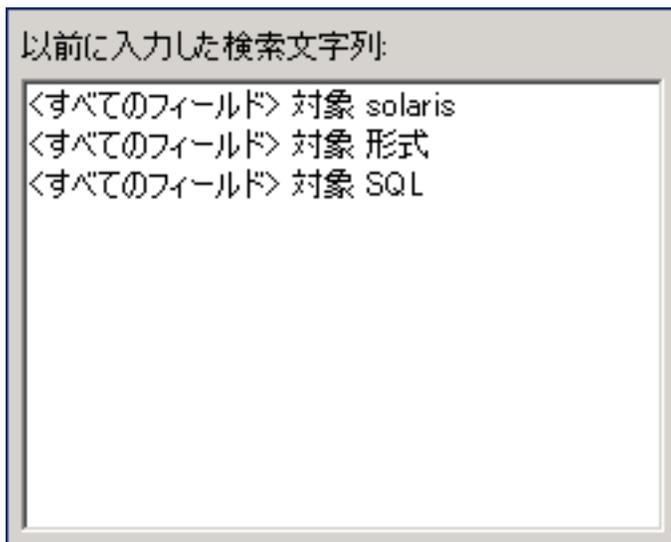
- オペレーションの入力が必要な種類の情報。
- 結果に含まれる情報。
- オペレーションの要件と前提条件。

説明全体を確認する場合は、オペレーションの **[プロパティ]** シートの **[説明]** タブを参照してください。

前回の検索コマンドを使用した検索

1. **[検索]** ペインで、**[検索履歴]** ボタンをクリックします。

[以前に入力した検索文字列] ウィンドウが開きます。



このウィンドウには、以前に使用した検索コマンドのリストが最大 25 個表示されます。

リストの検索コマンドの形式は、「対象<検索テキスト>」です。

2. リストから検索コマンドをダブルクリックして実行します。そのコマンドは、リストの最上位に追加されます。

検索結果の並べ替え

いずれかの列の列見出しをクリックすると、そのパラメーターを基に検索結果が並べ替えられます。

Lucene 構文を使用した検索

より詳細な検索結果を得たい場合、Apache Lucene 構文を使用して検索を行うことができます。

Lucene 検索構文については、Apache Software Foundation の Web サイトを参照してください。

1. Studio ウィンドウの下部にある **[検索]** タブをクリックして、**[検索]** ペインを開きます。
2. **[検索]** ドロップダウンリストで、**[<Lucene クエリ]** を選択します。
3. **[対象]** テキストボックスに、Lucene 検索構文を使用してクエリを入力し、検索ボタン  をクリックします。

最も簡単な Lucene 検索構文は次のようなものです。

<検索可能なフィールド名>:<検索対象の文字列>

検索のヒント

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

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

- フローまたはオペレーションの名前

例:

```
name:Get Temp Dir  
name:Clear Temp Dir
```

- オペレーションのタイプ

例:

```
type:cmd
```

- カテゴリ

例:

```
category:network
```

- 入力の名前

例:

```
inputs:server
```

- フローまたはオペレーション UUID

例:

```
id:1234-3453-3242-32423
```

- フローまたはオペレーションの説明に含まれている文字列

例:

```
description:clear
```

[検索] ペインからのオペレーションへのアクセス

検索結果から直接、オペレーションやフローを操作することができます。オペレーションやフローを開いて編集したり、作成ペインで開いているフローに追加することができます。

- 検索結果でオペレーションの行をダブルクリックして、オペレーションの[プロパティ]シートまたはフロー図を開きます。
- [検索] ペインからフロー図にオペレーションをドラッグし、検索結果のオペレーションからステップを作成します。

[フローに移動...] オプションを使用したフローの検索

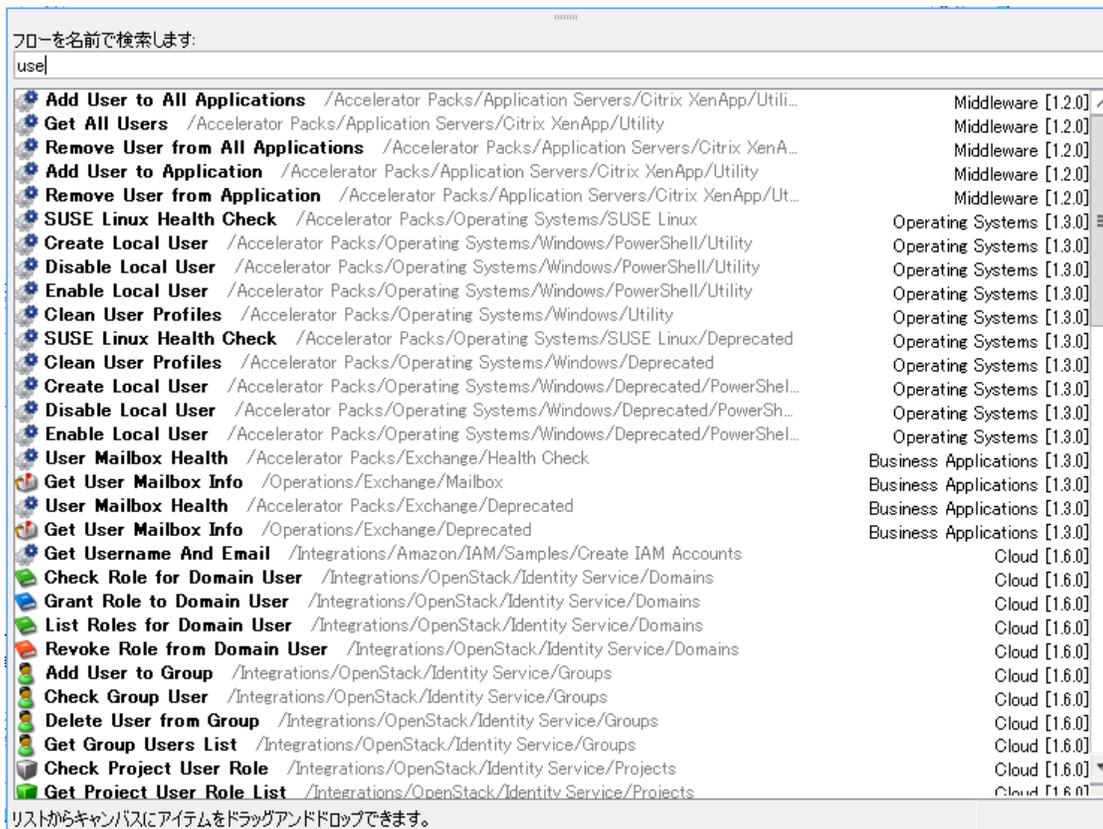
[フローに移動...] オプションを使用して、ワークスペース内のフローを簡単に見つけて開くことができます。

1. [ツール] > [フローに移動...] を選択します。 .

[フローを名前で検索します:] ウィンドウが表示されます。

2. フロー名の一部を入力します。

ウィンドウに検索結果が直ちに表示されます。たとえば、"use" という文字列が含まれる名前のフローを検索する場合、次のような結果が表示されることがあります。



3. 上向き矢印と下向き矢印キーを使用してリスト内を移動し、Enter キーを押して選択したアイテムを開きます。または、ダブルクリックしてアイテムを開くことができます。
4. 現在のフローエディターでリストからフローをドラッグアンドドロップしてステップを作成できます。

参考資料

[検索] ペイン

GUI アイテム	説明
検索	<ul style="list-style-type: none"> • 1つのフィールドで検索を実行する場合は、検索を実行するフィールドを選択します。 • 検索にすべてのフィールドを含めるには、[<すべてのフィールド]を選択します。 • Lucene クエリを使用して検索する場合は、[<Lucene クエリ]を選択します。
対象	検索する文字列を入力します。
完全	<ul style="list-style-type: none"> • 空白を1つの検索文字列の一部として扱うよう指定するには、[完全]チェックボックスを選択する • 空白を、検索文字列を順に区切るセパレーターとして扱うには、[完全]チェックボックスを選択解除します。
ランク	各検索結果のランクが表示されます。星が多いほどランクが高くなります。
種類	フローなどの見つかったアイテムの種類が表示されます。
パス	アイテムの保存先が表示されます。
説明	アイテムの[説明]タブから取得されたアイテムの説明が表示されます。
検索履歴	クリックすると、[以前に入力した検索文字列]ウィンドウが表示され、検索コマンドを再使用できます。

フローおよびオペレーションのコピー

フローやオペレーションをコピーする方法は3つあります。

- フローやオペレーションをコピーする場合は、封印されていないいずれかのフォルダーにコピーを貼り付けることができます。フローをコピーした場合、フローのみがコピーされ、フローを構成するオペレーションはコピーされません。
- [ディープコピー]をクリックすると、フローだけでなくそのフローを構成しているすべてのオペレーションもコピーされます。これは、新しいフローでオペレーションを変更する必要があり、かつ元のオペレーションに影響を与えたくない場合に使用します。
- フローやオペレーションを複製する場合、複製は自動的に元のオペレーションと同じフォルダーに配置され、[<名前> (Copy)]という名前になります。
- フローやオペレーションを切り取ると、そのフローやオペレーションは現在の場所から削除され、別の場所に貼り付けられます。

アクションプラグインから作成されたオペレーションのコピー

ソフトコピー

アクションプラグインの jar ファイルにリンクされたオペレーションをコピーした場合、コピーされたオペレーションは元のオペレーションへの参照を保持したままです。アクションプラグインの jar ファイルがアップグレードされた場合、元のオペレーションを、新しいバージョンを呼び出すよう更新すると、コピーされたオペレーションもすべて自動的に更新されます。これを、**ソフトコピー**と呼びます。

コピーされたオペレーションで自動的に更新されるアイテムは、アクションプラグイン jar ファイルへのリンクのみです。元のオペレーションの入力、出力、変数、スクリプトレットなどは、コピーでは更新されません。

コンテンツパックからオペレーションをコピーできますが、参照は親のコンテンツパックに保持されます。親のプラグインが今後修正された場合、ソフトコピーにもその修正が入ります。場合によっては、オペレーションへの修正を行わないようにすることもできます。この場合、オペレーションを親のプラグインからデタッチできます。ただし、デタッチされたオペレーションを手動で修正する必要があります。

ソフトコピーには次のような長所と短所があります。

- 短所は、元のオペレーションが削除された場合に、コピーが親をなくし、プラグインへのリンクも損失してしまうことです。その場合、新しい親を手動で選択する必要があります。

注: 存在しない親をポイントする孤立したソフトコピーオペレーションを修正するには、オペレーションの [詳細] タブの [選択] ボタンを使用して、親オペレーションを付加します。

- 長所は、元のオペレーションが別のバージョンのプラグインに更新された場合、ソフトコピーも更新されることです。

アクションプラグイン jar ファイルからのオペレーションの作成の詳細については、「[オペレーションの作成](#)」(349ページ)を参照してください。

ハードコピー

10.00 以前のバージョンの 00 では、アクションプラグインにリンクされたオペレーションをコピーすると、**ハードコピー**が作成されました。これは、コピーが、元のオペレーションと同様にアクションプラグインに直接リンクしています。アクションが更新された場合(たとえば JAR の名前またはクラスが変更された場合)、ハードコピーされたオペレーションもすべて更新する必要がありました。

00 10.x では、新しいオペレーションを作成し関連プラグインを選択することで、ハードコピーを作成することができます。この方法では、IAction getTemplate または @Action メタデータに基づいて新しいオペレーションが作成されます。入力や出力の複製も行う、オペレーションのハードコピーを作成することはできません。

ハードコピーには次のような長所と短所があります。

- 長所は、元のオペレーションが削除された場合、コピーには影響せず、親を失うことはありません。
- 短所は、元のオペレーションが別のバージョンのプラグインに更新されても、ハードコピーは更新されないことです。

注: オペレーションエディターの [詳細] タブの [選択] ボタンは、ハードコピーには使用できません。

ソフトコピーからハードコピーへの切り替え

ソフトコピーを親からデタッチし、ハードコピーにすることができます。フローの [詳細] タブで、[接続解除] ボタンをクリックしてオペレーションをデタッチします。確認メッセージが表示されます。プラグインの GAV パラメーターが元の親から取得されます。

ハードコピー内のプラグインの置き換え

特定のプラグインを使用している、ハードコピーされたすべてのオペレーションを検索できます。また、そのオペレーションを選択し、プラグイン GAV パラメーターを変更することができます。

実行する操作

フローまたはオペレーションのコピー

1. [依存関係] ペインまたは [プロジェクト] ペインで、コピーするフローまたはオペレーションを右クリックします。
2. [編集] > [コピー] を選択します。
3. コピーを保存するフォルダーに移動し、[編集] > [コピー] を選択します。

フローまたはオペレーションの複製

フローやオペレーションを複製する場合、複製は自動的に元のオペレーションと同じフォルダーに配置され、[<名前> (Copy)] という名前になります。

1. [依存関係] ペインまたは [プロジェクト] ペインで、コピーするフローまたはオペレーションを右クリックします。
2. [編集] > [複製] を選択します。

フローまたはオペレーションのディープコピー

フローをディープコピーした場合、フローのみではなく、フローを構成するオペレーションもコピーされます。

1. [依存関係] ペインまたは [プロジェクト] ペインで、コピーするフローを右クリックします。
2. [編集] > [ディープコピー] を選択します。

3. フローとそのオペレーションを保存するフォルダーに移動し、[編集]>[貼り付け]を選択します。

フローまたはオペレーションの切り取り

1. [依存関係] ペインまたは[プロジェクト] ペインで、移動するフローまたはオペレーションを右クリックします。
2. [編集]>[切り取り]を選択します。
3. フローやオペレーションを保存するフォルダーに移動し、[編集]>[切り取り]を選択します。

フローとオペレーションの使用方法的詳細

オペレーションやフローが既存のフローでどのように使用されているかを調べることで、その使用方法や実装方法について理解を深めることができます。これは、[参照] ペインでできます。

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

Studio には、次の 2 種類の参照があります。

- オペレーションまたはフローへの参照 - 選択したオペレーションまたはフローから作成されたステップのあるフローの一覧を示します。
- オペレーションまたはフローからの参照 - 選択したオペレーションまたはフローが使用するオブジェクト (選択リスト、グループに割り当てられた権限、システムフィルターなど) の一覧を示します。フローの場合は、フローのステップを作成したときのベースになったオペレーション (サブフローを含む) です。

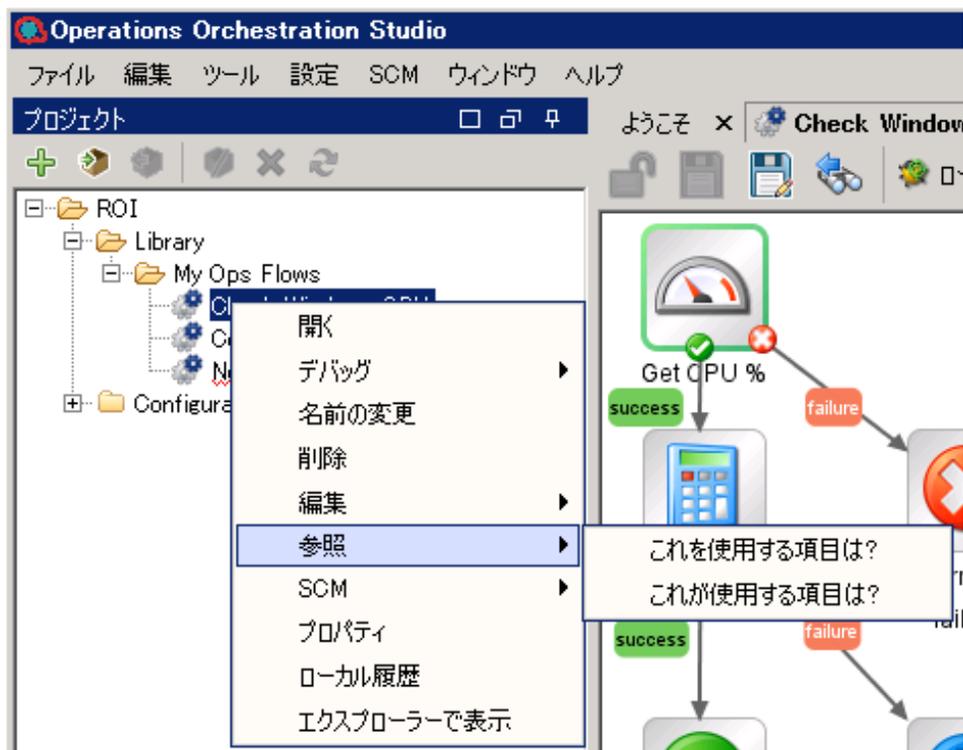
ベストプラクティス

オペレーションまたはフローを変更すると、それを使用しているほかのフローが壊れる場合があります。フローやオペレーションを変更する前に、[参照]>[これを使用する項目は?]を使用して、ほかのフローでこのフローやオペレーションを使用していないかを確認します。

実行する操作

フローまたはオペレーションを使用しているものを識別

1. [プロジェクト] ペインでオペレーションまたはフローを右クリックし、[参照]>[これを使用する項目は?]



[参照] ペインが開き、オペレーションまたはフローの参照元が表示されます。

2. [参照] ペインで行をダブルクリックして、アイテムを開いて編集します。

注: 複数のアイテムを開くには、Shift キーまたは Ctrl キーを使用して選択し、右クリックして [開く] を選択します。

フローまたはオペレーションが使用しているものを識別

1. [プロジェクト] ペインで、オペレーションまたはフローを右クリックします。
2. [参照]>[これが使用する項目は?]

[参照] ペインが開き、オペレーションまたはフローの参照先が表示されます。

ヒント: 参照元であるフローやオペレーションは、コピー、貼り付け、編集が可能なサンプルとして役立ちます。

参考資料

[参照] ペイン

[参照] ペインでは、既存のフローでのオペレーションまたはフローの使用方法を参照できます。



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

GUI アイテム	説明
オブジェクト	選択したフローまたはオペレーションで使用されている、またはこれらが使用するオブジェクトが表示されます。
パス	選択したフローまたはオペレーションで使用されている、またはこれらが使用する場所が表示されます。

フローおよびオペレーションに関するドキュメントの生成

重要: 現在のバージョンでは、実行時のドキュメントの生成機能はサポートされていません。Studio 内からドキュメントを生成できますが、ドキュメントの生成ステップを含むフローを作成しても、このステップは実行時には動作しません。

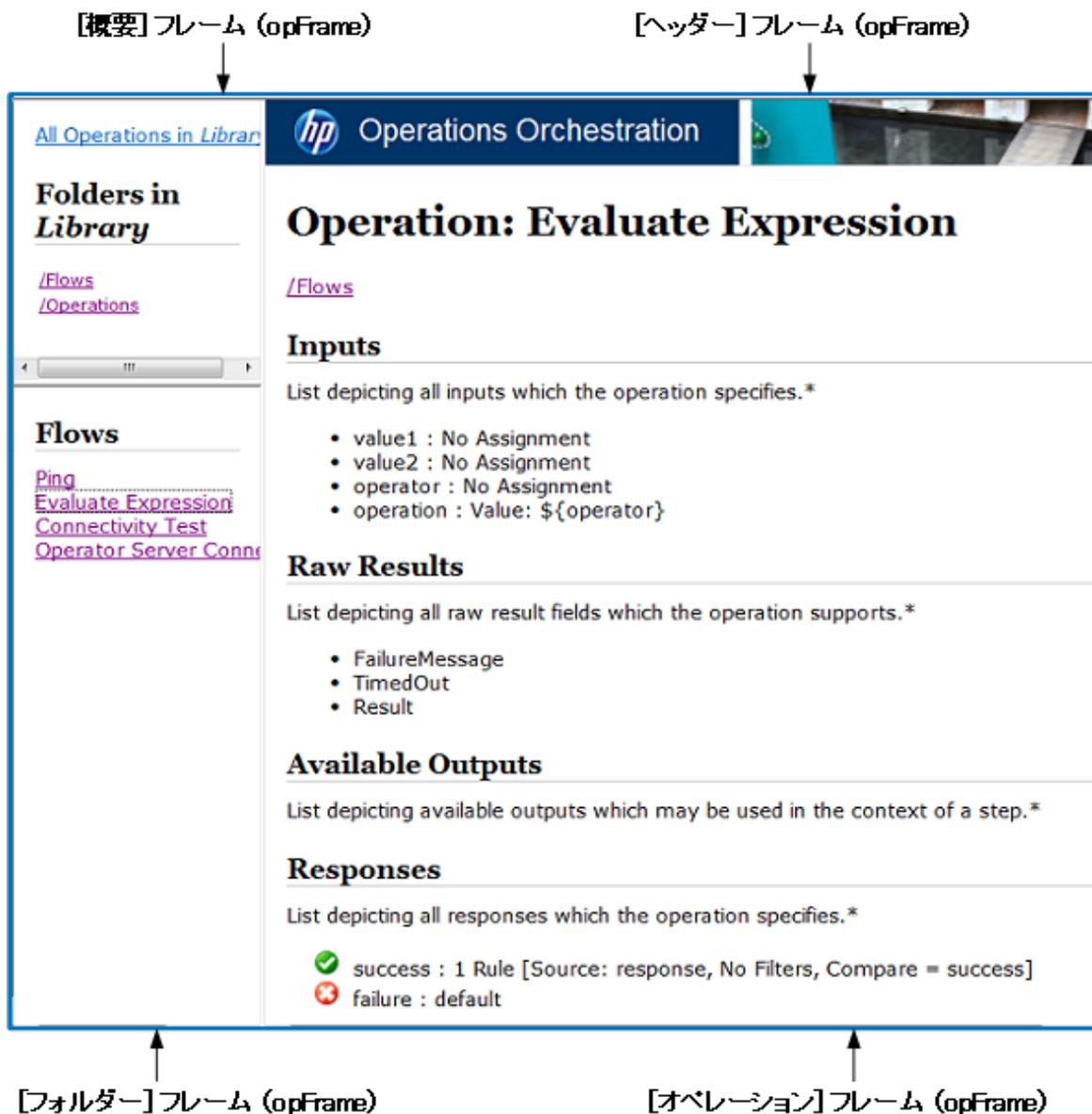
フローやオペレーションをドキュメントにし、ほかのユーザーに詳細情報を提供することができます。

- フローを PNG 画像としてエクスポートする。
- ドキュメントの生成機能を使用して、フローやオペレーションに関する情報を含む HTML ページを作成する。

生成されたドキュメントの構造

ドキュメントを生成すると、**index.html** という HTML ページが作成されます。このページには、次のフレームがあります。

- [概要] フレーム - 左上の [概要] フレームには、ドキュメントを生成したフォルダーに含まれるサブフォルダーのリストが表示されます。[フォルダー] フレームに内容を表示するフォルダーを選択します。
- [フォルダー] フレーム - 左下の [フォルダー] フレームには、[概要] フレームで選択したフォルダー内のフローやオペレーションのリストが表示されます。
- [ヘッダー] フレーム - 右上の [ヘッダー] フレームには、HP 00 バナーが表示されます。
- [オペレーション] フレーム - 右下の [オペレーション] フレームには、フローまたはオペレーションの説明が表示されます。[プロパティ] シートの [説明] タブで入力した情報です。



実行する操作

フローの PNG 画像でのエクスポート

1. フローを作成ペインで開きます。
2. 作成ペインを右クリックし、[PNG にエクスポート]を選択します。
3. 画像の保存先を参照し、[保存]をクリックします。

標準形式でのドキュメントの生成

特定のフローやオペレーションを含むフォルダーのドキュメントや、**Library** フォルダー全体のドキュメントを生成することができます。

1. ドキュメントを作成するフォルダーを右クリックします。

注: ドキュメントを作成できるのは、[コンテンツパック]タブのみです。このオプションは、[ライブラリ]ビューでは利用できません。

2. [ドキュメントの生成] > [標準形式] を選択します。
3. ドキュメントファイルの保存先を参照し、[保存]をクリックします。HTML ファイルの **index.html** が Web ブラウザーで開きます。
4. 旧バージョンの **index.html** を上書きする場合は、[すべてはい]をクリックします。

注: 旧バージョンのドキュメントを上書きしない場合は、[キャンセル]をクリックして手順を繰り返し、ファイルを別の場所に保存します。

カスタム形式のドキュメントの生成

1. ドキュメントを作成するフォルダーを右クリックします。
2. [ドキュメントの生成] > [カスタム形式] を選択します。
3. [ドキュメント生成テンプレートの選択] ダイアログボックスで、ドキュメントの生成に使用するテンプレートを選択します。
4. ドキュメントファイルの保存先を参照し、[保存]をクリックします。HTML ファイルの **index.html** が Web ブラウザーで開きます。

カスタマイズしたドキュメントの生成テンプレートの作成

ドキュメントのテンプレートは **Studio\template** フォルダーに保存されます。テンプレートの末尾には **.vm** が付いていて、テキストエディターで編集できます。テンプレートの詳細については、以降の「参照情報」のセクションを参照してください。

Studio\template フォルダーに作成した新しい **.vm** ファイルが、[ドキュメント生成テンプレートの選択] ダイアログボックスのテンプレートリストに表示されます。



1. 必要な .vm テンプレートのコピーを作成し、そのコピーの名前を変更します。

注意: 元の .vm テンプレートは編集したり、名前を変更しないでください。

2. テキストエディターで、新しいテンプレートに変更を加え、保存します。
3. Studio で、ドキュメントを作成するフォルダーを右クリックして、[ドキュメントの生成]>[カスタム形式]を選択します。
4. [ドキュメント生成テンプレートの選択] ダイアログボックスで、作成したカスタムテンプレートを選択します。

参考資料

.vm テンプレートファイル

Folder_template.vm

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

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

All_folders_template.vm

フォルダーのコンテンツの表を生成します。

- **Header.css** - 全体的なフォント、色などを定義しているスタイルシート。
- **All_ops_template.vm** - すべてのオペレーションのリストを生成し、リンクを作成して **folderFrame** (左下) に表示します。

- **Folder_contents.vm** – 選択したフォルダーのコンテンツのリストを生成し、リンクを作成して **folderFrame** (左下) に表示します。

All_ops_template.vm

すべてのオペレーションのコンテンツおよびすべての子オペレーションのドキュメントの表を生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Op_template.vm** – リンクを作成して **opFrame** (右下) に表示します。

Folder_overview_template.vm

フォルダーのコンテンツについての概要を表形式で生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Folder_contents.vm** – リンクを作成して **folderFrame** (左下) に表示します。

Op_template.vm

単独のオペレーションのドキュメントを生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Folder_template.vm** – リンクを作成して同じフレームに表示します (親フォルダー以下)。
- **Folder_contents.vm** – フォルダーのコンテンツを **folderFrame** に表示します。

Flow_template.vm

単独のフローのドキュメントを生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Flow_template.vm** – リンクを作成して同じフレームに表示します (親フォルダー以下)。
- **Folder_contents.vm** – フォルダーのコンテンツのリストを生成し、リンクを作成して **folderFrame** (左下) に表示します。
- **Op_template.vm** – リンクを作成して **opFrame** (右下) に表示します。

Folder_contents.vm

単独のフォルダーのコンテンツの表を生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Op_template.vm** – リンクを作成して **opFrame** (右下) に表示します。

Header.html

ヒューレット・パッカーのバナー。

Hp_rockwell.css

ヒューレット・パッカーのバナー用のスタイルシート。

Hp_steps_307x39.jpg

ヒューレット・パッカーのバナー用の画像。

Logo_hp_smallmasthead.gif

ヒューレット・パッカーのバナー用のロゴ。

フローとオペレーションのバージョン履歴管理

構成アイテム、フロー、またはオペレーションが保存されるたびに、そのアイテムの新しいバージョンが作成されます。[バージョン履歴] ダイアログボックスにこれらのバージョンが一覧表示され、以下の操作を実行できます。

- アイテムの前のバージョンを表示する
- 前のバージョンを別の名前で保存する
- アイテムを前のバージョンへ復元する
- 2つのバージョンの差異を表示する

注: バージョン履歴機能は、Studio でのローカルな作成作業にのみ適用され、ソースコントロールリポジトリに保存されたバージョンには適用されません。バージョン履歴は、作業内容をソースコントロールリポジトリにコミットする前に使用します。リポジトリで共有するためにフローをコミットした場合、HP 00 のローカルのバージョン履歴機能を前のバージョンに戻す作業に使用することは推奨されません。

ソースコントロールリポジトリでバージョン履歴を処理する方法の詳細については、[「リポジトリ内の Git オペレーションの履歴の表示」\(106ページ\)](#)を参照してください。



実行する操作

構成アイテム、フロー、オペレーションの前のバージョンを開く

アイテムの前のバージョンを開いて表示したときに、そのバージョンに別の名前を付けて保存することができます。

1. 構成アイテム、フロー、またはオペレーションを右クリックし、**[履歴の表示]** を選択します。
2. 必要なバージョンを右クリックし、**[開く]** をクリックします。選択したバージョンが作成ペインで開きます。
3. 開いたバージョンを保存するには、**[名前を付けて保存]**  ボタンをクリックし、一意の名前を付けます。プロジェクトの2つのバージョンが個別に保存されます。
4. **[OK]** をクリックして **[バージョン履歴]** ダイアログボックスを閉じます。

構成アイテム、フロー、オペレーションの以前のバージョンへの復元

この手順により、構成アイテム、フロー、またはオペレーションを前のバージョンに復元できます。現在のバージョンと前のバージョンの両方を保持するには、「構成アイテム、フロー、オペレーションの前のバージョンを開く」を参照してください。

1. 構成アイテム、フロー、またはオペレーションを右クリックし、**[履歴の表示]** を選択します。
2. 復元するバージョンを右クリックし、**[復元]** をクリックします。復元するバージョンが作成ペインで開きます。
3. **[OK]** をクリックして **[バージョン履歴]** ダイアログボックスを閉じます。
4. プロジェクトを保存します。開いたバージョンが現在のバージョンを上書きして保存されます。

バージョンの比較

[バージョン履歴] ダイアログボックスでは、構成アイテム、フロー、オペレーションのバージョンを比較することもできます。現在のバージョンが片側に表示され、前のバージョンが反対側に表示されます。

1. 構成アイテム、フロー、またはオペレーションを右クリックし、**[履歴の表示]** を選択します。
2. Ctrl キーを押しながら現在のバージョン (最上行) と前のバージョンの両方を選択します。
3. 右クリックし、**[比較]** を選択します。アイテムの現在の状態と前のバージョンとの差異が表示されます。
4. **[OK]** をクリックして **[バージョン履歴]** ダイアログボックスを閉じます。

参考資料

[バージョン履歴] ダイアログボックス



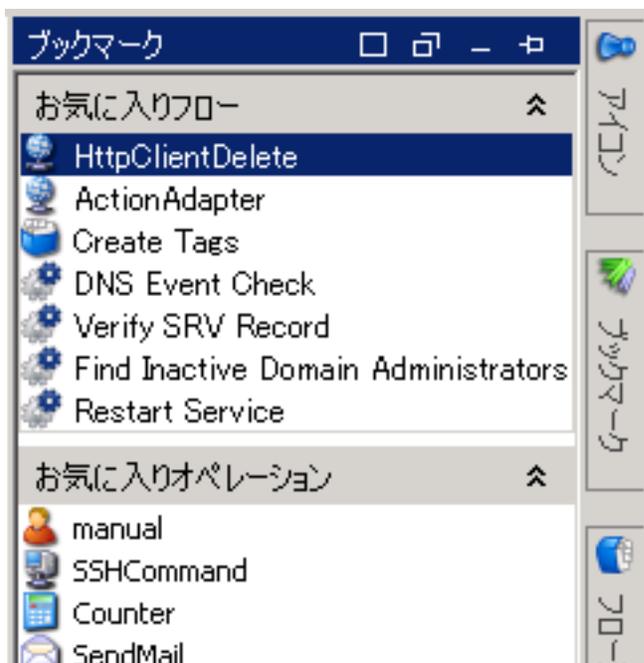
メニューアイテム	説明
バージョン	自動的に生成された、プロジェクトのバージョン番号。
日時	バージョンが変更された日付と時刻。
コメント	加えられた変更の種類。

フローおよびオペレーションのブックマーク

[ブックマーク] ペインを使用すると、使用頻度の高いオペレーションやフローを簡単に見つけて使用できます。ブックマークしたフローやオペレーションは、通常のライブラリ内の場所からも利用できます。

フローやオペレーションを [ブックマーク] ペインに追加すると、作成キャンバスの右クリックメニューにそのフローまたはオペレーションが表示されるようになります。

インストールした Studio から別の Studio へ、ブックマークをエクスポートおよびインポートできます。



実行する操作

ブックマークの追加

1. Studio ウィンドウの右上にある [ブックマーク] タブをクリックして、[ブックマーク] ペインを開きます。
2. ペインを開いたままにしておくには、ペインの右上隅にある [固定] アイコンをクリックします。
3. ライブラリまたは [検索] ペインの結果からフローまたはオペレーションを [ブックマーク] ペインの適切なシェルフにドラッグします。

シェルフの [ブックマーク] ペインへの追加

[ブックマーク] ペインには、デフォルトでフロー用とオペレーション用の 2 つのシェルフがあります。ですが、カスタマイズしたシェルフを追加してブックマークを整理できます。

1. [ブックマーク] ペインのシェルフのタイトルバーを右クリックし、[追加] を選択します。
2. シェルフの名前を入力して [OK] をクリックします。

シェルフの名前の変更

1. 名前を変更するシェルフのタイトルバーを右クリックし、[名前の変更] を選択します。
2. シェルフの新しい名前を入力して [OK] をクリックします。

シェルフの削除

1. 削除するシェルフのタイトルバーを右クリックし、[削除] を選択します。

2. 確認ウィンドウで[はい]をクリックします。

シェルフの表示/非表示

1. 非表示にするシェルフのタイトルバーを右クリックし、[非表示]を選択します。そのシェルフは[ブックマーク]ペインに表示されなくなります。
2. 非表示にしたシェルフを表示するには、[ブックマーク]ペインを右クリックし、[表示]をクリックしてから、非表示にしたシェルフ名を選択します。
3. 非表示にしたシェルフをすべて表示するには、[ブックマーク]ペインを右クリックし、[すべて表示]を選択します。

シェルフの上下移動

上下に移動するシェルフのタイトルバーを右クリックし、[上に移動]または[下に移動]を選択します。

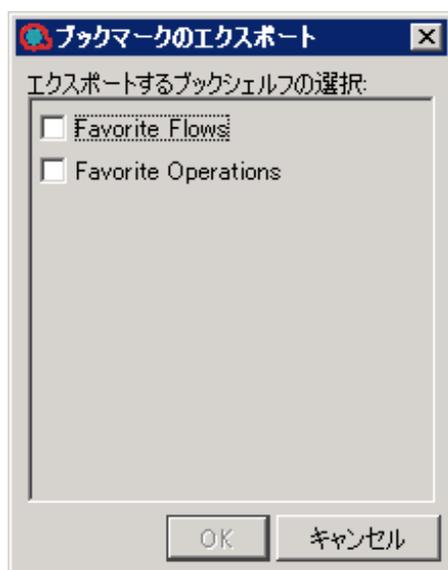
シェルフの折りたたみ/展開

1. シェルフを折りたたむには、シェルフのタイトルバー上にある二重矢印(⌵)をクリックします。シェルフのタイトルは表示されますが、シェルフ内のブックマークは非表示になります。
2. シェルフを展開するには、二重矢印を再度クリックします。

ブックマークのエクスポート

インストールした Studio から別の Studio へ、ブックマークをエクスポートできます。

1. [ブックマーク]ペイン内を右クリックし、[エクスポート]をクリックします。
2. [ブックマークのエクスポート]ダイアログボックスで、エクスポートするシェルフを選択します。

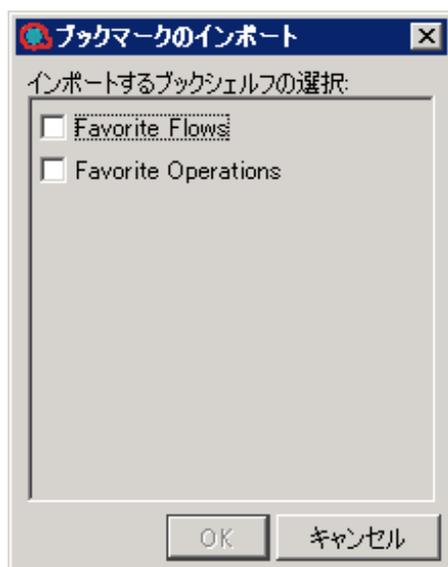


3. [OK]をクリックします。[エクスポートファイルの選択]ダイアログボックスが表示されます。

4. ブックマークの保存先を参照し、ファイルの名前を入力します。
5. [保存] をクリックします。

ブックマークのインポート

1. [ブックマーク] ペイン内を右クリックし、[インポート] を選択します。
2. [インポートファイルの選択] ダイアログボックスで、ブックマークファイルを探して選択し、[開く] をクリックします。
3. [ブックマークのインポート] ダイアログボックスで、インポートするシェルフを選択して [OK] をクリックします。



インポート中、同じシェルフにあるブックマークは UUID を基に統合されます。既存のブックマークは保持され、複製は作成されず、新しいエントリがシェルフに追加されます。

[ブックマーク] ペインからプロジェクトへのフローまたはオペレーションのコピー

[ブックマーク] ペインから [プロジェクト] ペインにフローおよびオペレーションをドラッグして、プロジェクトにコピーすることもできます。

1. [ブックマーク] ペインでフローまたはオペレーションを選択します。
2. [プロジェクト] ペインのプロジェクトにアイテムをドラッグします。

参考資料

[ブックマーク] メニュー

[ブックマーク] ペインで右クリックすると、[ブックマーク] メニューが表示されます。メニューに表示されるアイテムは、右クリックした時に選択されているアイテムによって異なります。

メニューアイテム	説明
追加	新しいシェルフを [ブックマーク] ペインに追加します。
削除	選択したシェルフを [ブックマーク] ペインから削除します。
名前の変更	[ブックマーク] ペインで選択したシェルフの名前を変更します。
上に移動	[ブックマーク] ペインで選択したシェルフを上に移動します。
下に移動	[ブックマーク] ペインで選択したシェルフを下に移動します。
非表示	[ブックマーク] ペインで選択したシェルフを非表示にします。
表示	[ブックマーク] ペインで表示する、非表示のシェルフを選択できます。
すべて表示	[ブックマーク] ペインで選択したすべてのシェルフを非表示にします。
折りたたみ [※]	[ブックマーク] ペインのシェルフを折りたたみます。シェルフのタイトルは表示されますが、ブックマークは非表示になります。
インポート	
エクスポート	

Studio プロパティの設定

[オプション] ダイアログでデフォルトの設定を変更することによって Studio でいくつかのプロパティを構成できます。さらに、**Studio.properties** ファイルでデフォルトの設定を手動で変更することによって Studio でさまざまなプロパティをすべて構成できます。すべての設定の詳細については、[Studio のプロパティ](#) を参照してください。

次のセクションでは、使用頻度が高いパラメーターについて説明します。

[割り当て元] と [割り当て先] のデフォルト値を定義する

デフォルトでは、オペレーションまたはフローステップで新しい入力を作成したときには、入力ソースに値が割り当てられません。フローによっては、フロー変数から入力値を割り当てる必要があります。

同様に、デフォルトでは、Studio はフロー変数に入力値を割り当てません。やはり、フローによっては、フロー変数に入力値を割り当てる必要があります。

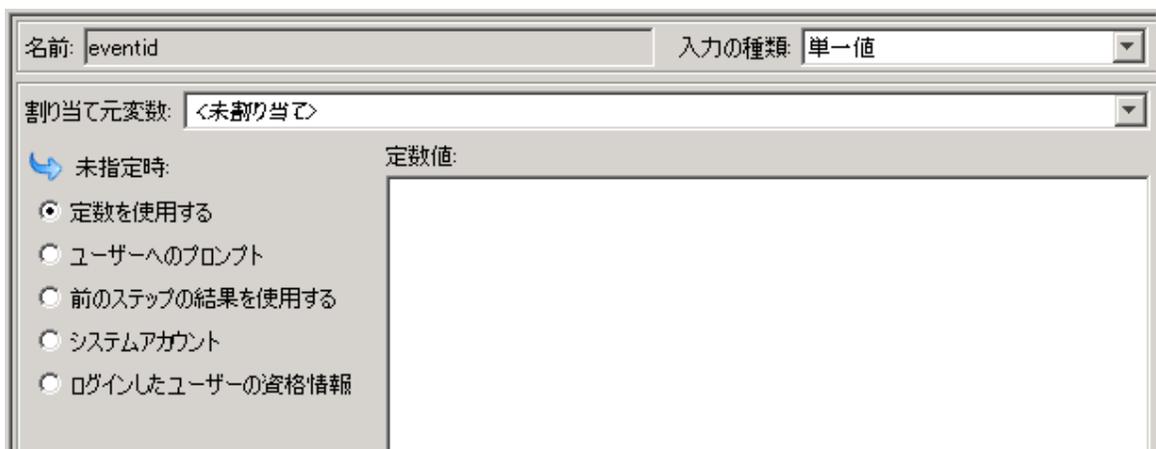
入力値のどちらかまたは両方の入力ソースと同じ名前のフロー変数を割り当てるように Studio のデフォルトの動作を構成できます。この操作は、**Studio.properties** ファイルで行います。

設定の値を **not-assigned** から **assigned** に変更すると、次の操作を行ったときに [割り当て元変数] および [割り当て先変数] に同じ名前の入力のフロー変数が割り当てられます。

- オペレーションまたはフローステップに新しい入力を追加する
- フローに新しいステップを作成する
- 新しいオペレーションを作成する

[未指定時] のデフォルト値を [定数を使用する] に変更する

Studio 10.20 以降では、オペレーションまたはフローステップで新しい入力を作成するときに、デフォルトで [未指定時] パラメーターが [定数を使用する] に設定され、[定数値] に空白の値が指定されます。この割り当てを使用すると、一部の入力が検証に合格しないことがあります。



同様に、[プロジェクト] および [依存関係] ペインからドラッグアンドドロップを使用してフロー内に新しいステップを作成するときにも同じルールが適用されます。ドラッグアンドドロップの前にテキストを使用して[未指定時]パラメーターが[ユーザーへのプロンプト]に設定されているステップ入力の場合、ドラッグアンドドロップの後に、これらの入力が[定数を使用する]に変更され、空白の値が指定されます。[ユーザーへのプロンプト]に設定されていても、[必須]としてマークされているリストの(選択リスト、ドメインインターム、フロー変数)の一部になっているステップ入力は、そのまま変更されません。

オペレーションまたはフローステップで新しい入力を作成されたときに[ユーザーへのプロンプト]を使用するように Studio のデフォルトの動作を構成できます。この場合、ドラッグアンドドロップ操作を使用してフローで新しいステップが作成された場合、[ユーザーへのプロンプト]設定はそのまま変更されません。

デフォルトの動作を[定数を使用する]から[ユーザーへのプロンプト]に変更すると、次のいずれかの操作を実行したときに、選択済みの[未指定時]オプションが[ユーザーへのプロンプト - 入力項目: テキスト]になります。

- オペレーションまたはフローステップに新しい入力を追加する
- フローに新しいステップを作成する
- 新しいオペレーションを作成する

コンテキストインスペクターに特殊文字を表示する

¶ ボタンを使用して、変数上の非表示の書式設定マークのオンとオフを切り替えることができます。たとえば、スペース、改行、特殊文字、^Ms などがあります。デフォルトでは、これは無効になっています。

このスイッチを有効にすると、カテゴリ、ステップコンテキスト、グローバルコンテキスト、選択リスト、システムプロパティ内の値要素で、値に関連付けられているツールチップ内にこれらの特殊文字およびスペースが表示されます、

- studio.properties ファイルで無効なパラメーターまたは空白のパラメーターが使用されている場合、デフォルトのパラメーターが使用されます。
- プロパティの優先される値がツールチップ内に適切に表示されない場合、異なる値に切り替えるか、プロパティを空白のままにすることができます。
- 改行文字 <CR><LF>、<CR>、<LF> の後には、ツールチップ内の新しい行と視覚的な表現が続きます。
 - 垂直タブ文字 \u000B は ^K と表示されます。
 - 改ページ文字 \u000C は ^L と表示されます。

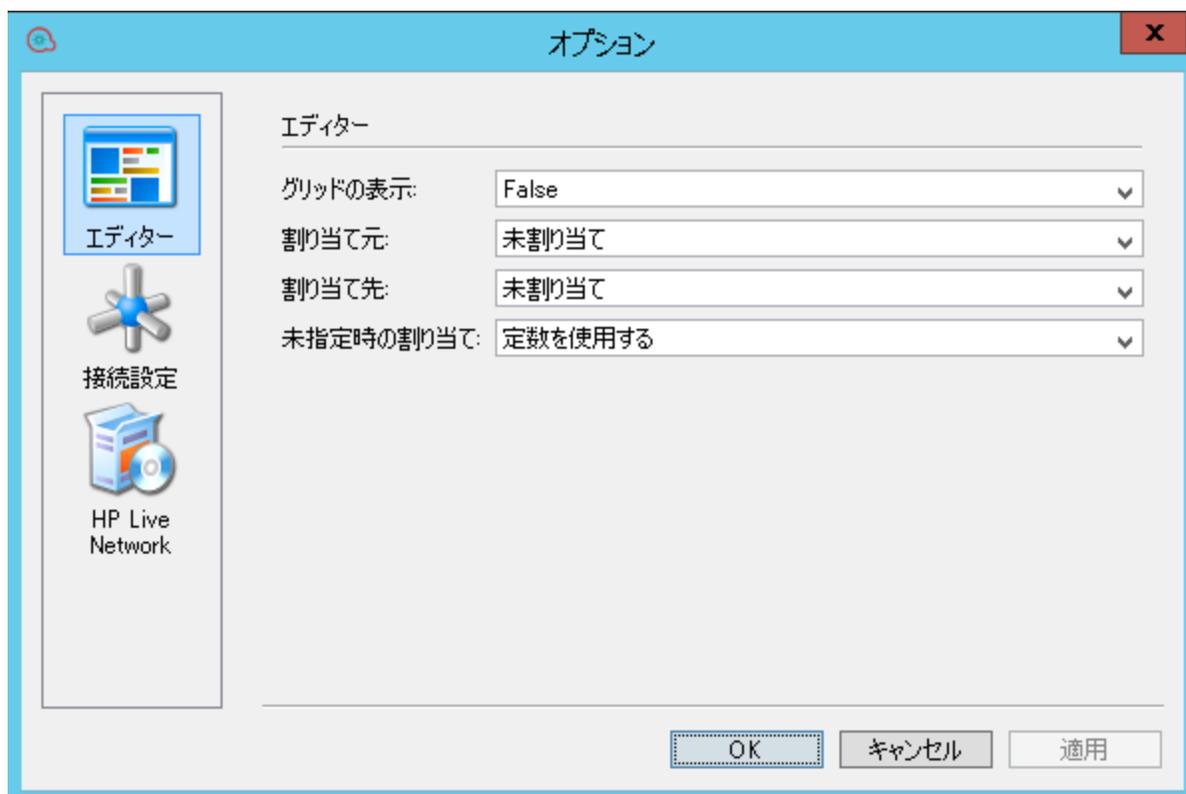
実行する操作

Studio のプロパティを編集します。

1. Windows エクスプローラーで、<HP 00 ホーム> > Studio > conf ディレクトリに移動します。
2. studio.properties ファイルをテキストエディターで開きます。
3. 変更するパラメーターの名前がわかっている場合は、テキストエディターでパラメーターを検索します。
4. パラメーターの名前がわからない場合は、[「Studio プロパティの設定」\(381ページ\)](#)の表を参照して見つけてください。

[オプション] ダイアログを使用して [割り当て元]、[割り当て先]、[未指定時] のデフォルト値を指定する

1. Studio で、[構成] > [オプション] を選択します。
[オプション] ダイアログが開きます。



2. [割り当て元]と[割り当て先]で、次のいずれかを選択します。

未割り当て	入力からまたは入力に値を割り当てません。
割り当て済み	入力と同じ名前のフロー変数の値が入力にまたは入力から割り当てられます。

3. [未指定時]で次のいずれかを選択します。

定数を使用する	[未指定時]フィールドが、[定数を使用する]に設定され、空白の値が指定されます。
ユーザーへのプロンプト	[未指定時]フィールドが[ユーザーへのプロンプト]に設定されません。

[割り当て元]と[割り当て先]のデフォルト値を手動で定義する

- Windows エクスプローラーで、<HP 00 ホーム> > Studio > conf ディレクトリに移動します。
- 編集のために **studio.properties** を開きます。
- 次の行を見つけます。

dharmastudio.ui.inputinspector.assignfrom.selected=not-assigned

dharmastudio.ui.inputinspector.assignto.selected=not-assigned

各パラメーターに、次のいずれかの値を指定できます。

not-assigned	入力からまたは入力に値を割り当てません。
assigned	入力と同じ名前のフロー変数の値が入力にまたは入力から割り当てられます。

注: この動作は Studio 10.50 をクリーンインストールしている場合に当てはまります。バージョン 10.50 にアップグレードした場合、**[割り当て元変数]** および **[割り当て先変数]** のデフォルト値は、前のバージョンで割り当てられた値に自動的に設定されます。

[未指定時] のデフォルト値を [定数を使用する] に変更する

1. Windows エクスプローラーで、<HP 00 ホーム> > **Studio** > **conf** ディレクトリに移動します。
2. 編集のために **studio.properties** を開きます。
3. 次の行を見つけます。

default.behavior.define.input=use_constant

4. 各パラメーターに、次のいずれかの値を指定できます。

use_constant	[未指定時] フィールドが、 [定数を使用する] に設定され、空白の値が指定されます。
prompt_user	[未指定時] フィールドが [ユーザーへのプロンプト] に設定されます。

注: この動作は Studio 10.50 をクリーンインストールしている場合に当てはまります。バージョン 10.50 にアップグレードした場合、**default.behavior.define.input** のデフォルト値は、前のバージョンで割り当てられた値に自動的に設定されます。

フローエディターを開いたときにデフォルトでグリッドが表示されるようにする

これは、すべてのフローエディターのグリッドの表示を制御するグローバルな設定であり、フローエディターを起動するたびにグリッドが表示されるようになります。その後で、個別のフローエディターインスタンスで**[グリッドの表示/非表示]**  ボタンを使用してグリッドを変更できます。

デフォルトでは、フローエディターを開いたときにグリッドは表示されません。

[オプション] ダイアログから、または **studio.properties** ファイルを編集して、グリッドを表示することができます。

[オプション] ダイアログで次の操作を実行します。

1. Studio で、**[構成]** > **[オプション]** を選択します。
2. **[グリッドの表示]** フィールドで、**[True]** を選択してグリッドを表示するか、**[False]** を選択してグリッドを非表示にします。

studio.properties ファイルで次の操作を実行します。

1. Windows エクスプローラーで、<HP 00 ホーム> > **Studio** > **conf** ディレクトリに移動します。
2. 編集のために **studio.properties** を開きます。
3. 次の行を見つけます。

dharmastudio.ui.activegrid=false

4. 設定の値を true に変更します。

dharmastudio.ui.activegrid=true

各ツールチップの文字数を指定する

1. Windows エクスプローラーで、<HP 00 ホーム> > Studio > conf ディレクトリに移動します。
2. 編集のために **studio.properties** を開きます。
3. 次の行を見つけます。

dharmastudio.ui.tooltip.showwhitespacecharacters.max.chars.per.line=5000

4. 必要に応じて値を変更します。

コンテキストインスペクターに特殊文字を表示する

1. Windows エクスプローラーで、<HP 00 ホーム> > Studio > conf ディレクトリに移動します。
2. 編集のために **studio.properties** を開きます。
3. 次の行を見つけます。

dharmastudio.ui.tooltip.showwhitespacecharacters.max.chars.per.line=5000

**dharmastudio.ui.tooltip.showwhitespacecharacters.space= **

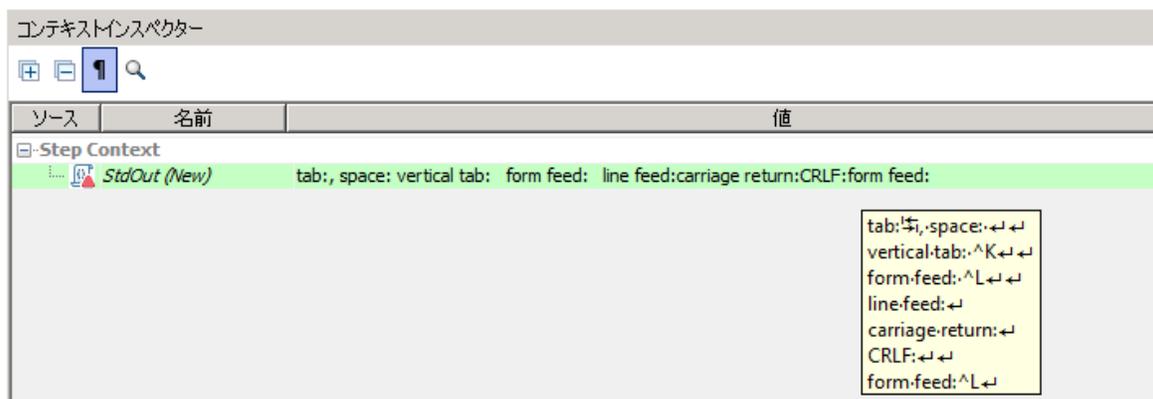
**dharmastudio.ui.tooltip.showwhitespacecharacters.tab=	**

**dharmastudio.ui.tooltip.showwhitespacecharacters.enter=**

**dharmastudio.ui.tooltip.showwhitespacecharacters.otherwhitespace=
**

4. 必要に応じて値を変更します。

たとえば、文字のオーバーライドを定義する場合、コンテキストインスペクターで次のように表示できます。



HPLN 検索を無効にする

デフォルトでは、Studio の下部に HP Live Network パネルが表示されます。 **hpln.enabled** パラメーターの値を変更することで、このパネルを無効にすることができます。

1. Windows エクスプローラーで、<HP 00 ホーム> > Studio > conf ディレクトリに移動します。
2. 編集のために **studio.properties** を開きます。
3. 次の行を見つけます。
hpln.enabled=true
4. 検索を無効にするには、値を false に変更します。
hpln.enabled=false

HPLN 接続/読み取りのタイムアウトを指定する

1. Windows エクスプローラーで、<HP 00 ホーム> > Studio > conf ディレクトリに移動します。
2. 編集のために **studio.properties** を開きます。
3. 次の行を見つけます。
hpln.connection.timeout=180
hpln.read.timeout=300
4. 必要に応じてタイムアウト値を設定します。タイムアウトを設定しない場合は、これらのパラメーターを 0 に設定します。

HPLN 検索結果の最大数を指定する

この設定を使用して、HP Live Network の検索結果パネルに表示される検索結果の最大数を指定できます。

1. Windows エクスプローラーで、<HP 00 ホーム> > Studio > conf ディレクトリに移動します。
2. 編集のために **studio.properties** を開きます。
3. 次の行を見つけます。
hpln.search.results.maximum.count=100
4. 必要な検索結果の最大数を設定します。

参考資料

Studio のプロパティ

パラメーター	デフォルト値	説明
dharmadefault.repository	location=default/repo	デフォルトの Studio ローカルリポジトリ。

パラメーター	デフォルト値	説明
user.language	en	Studio のロケール。これらの値は SCM リポジトリに影響しません。これらが設定されていないかまたは無効な場合、Studio はシステムロケールを使用します。
user.region	US	
dharma.repo.max_history_length	50	オフラインリポジトリの履歴に保存されるアイテムの数。
dharma.studio.ui.lookandfeel	Windows	Studio アプリケーションの外観。
dharma.studio.ui.input.constant.max.chars	8192	定数値の最大文字数を表す正の整数。8193 未満にする必要があります。
dharma.studio.ui.tooltip.showwhitespacecharacters.max.chars.per.line	5000	コンテキストインスペクターに特殊文字を表示します。 ツールチップの行あたりの文字数。
dharma.studio.ui.tooltip.showwhitespacecharacters.space	•	次のいずれかの値です。 •、•
dharma.studio.ui.tooltip.showwhitespacecharacters.tab	↹	次のいずれかの値です。 ↹、⇥、→、¬
dharma.studio.ui.tooltip.showwhitespacecharacters.enter	↵	次のいずれかの値です。 ↵、↲、↶
dharma.studio.ui.tooltip.showwhitespacecharacters.otherwhitespace	•	次のいずれかの値です。 •、•、☯

パラメーター	デフォルト値	説明
dharma.scripting.template.Rhino dharma.scripting.template.Nashorn		<p>このテンプレートは、オペレーションのリターンデータ、入力、およびコンテキストデータへのアクセス方法を示します。</p> <p>例:</p> <p>入力にアクセスするには、単純に [入力] ペインから入力名を参照します。例:myData = inputName;</p> <p>現在のブランチのコンテキストインスタンスにアクセスするには: myBranchData = scriptletBranchContext.get("myBranchVarName");</p> <p>コンテキストキー myContextKey に関連付けられている値を取得するには: myContextData = scriptletContext.get("myContextKey");</p> <p>コンテキストキー myContextKey に関連付けられている値をローカルコンテキストから取得するには: myContextData = scriptletContext.getLocal("myContextKey");</p> <p>コマンドラインまたはスクリプトからのリターンコードにアクセスするには (これは文字列であり、数値処理を行うには整数に変換する必要があります): code = parseInt(scriptletRawResult['Code']);</p> <p>オペレーションからの出力 (stdout など) にアクセスするには: data = scriptletRawResult['Output String'];</p> <p>オペレーションからのエラー文字列 (stderr など) にアクセスするには: error = scriptletRawResult['Error String'];</p> <p>オペレーションによっては異なる複数の結果変数を使用できる場合があります。特定のオペレーションのリストを表示するには、[出力] タブの [出力フィールド] ドロップダウンを開きます。</p> <p>オペレーションのレスポンスを設定するには ([レスポンス] タブのいずれかのレスポンスと一致している必要があります): scriptletResponse = "success";</p> <p>オペレーションの結果を設定するには: scriptletResult = "Your Result Here";</p> <p>このコンテキストを使用して他のステップ内で使用されたデータを格納します。スクリプトレットを開発およびデバッグするときに、情報をコンテキストに配置して変数を調べる場合も役に立ちます。たとえば、次のように OutputString 情報を 'Output' というコンテキストキーに配置します。scriptletContext.putGlobal("Output", data);</p> <p>キーと値はどちらも文字列です。たとえば、'code' をコンテキスト内に配置するには: scriptletContext.putGlobal("Code", code.toString());</p> <p>データをローカルコンテキスト内に配置することもできます。これ</p>

パラメーター	デフォルト値	説明
	<p>は、現在のフローでのみデータを使用することができ、親フローや他のサブフローでは使用できないことを意味します。これを行うには、次のスクリプトレットを使用します。</p> <pre>scriptletContext.put("LocalVariable","LocalValue");</pre> <p>注: フローが 64 K より大きい場合は、フローを保存しようとする、例外が表示されます。さらに、そのフローにアクセスしようとするたびに例外が発生します。これは Studio を再起動した場合でも同じです。</p> <p>長すぎるスクリプトレットを作成しないようにする必要があります。64 K より大きいスクリプトレットをコンパイルした場合、フローを保存しようとする、例外が発生します。そのフローにアクセスしようとするたびに例外が発生します。これは Studio を再起動した場合でも同じです。</p>	
<p>dharma.studio.ui.inputinspector. assignfrom.selected</p>	<p>説明を参照してください</p>	<p>このプロパティを使用して、入力のインスペクターで [割り当て元変数] コンボボックスの初期オプションを設定できます。</p> <p>有効な値:</p> <p>assigned - 以前のバージョンの Studio からアップグレードした場合のデフォルト</p> <p>not-assigned - Studio 10.20 をインストールした場合のデフォルト</p>
<p>dharma.studio.ui.inputinspector. assignto.selected</p>	<p>説明を参照してください</p>	<p>このプロパティを使用して、入力のインスペクターで [割り当て先変数] コンボボックスに初期オプションを表示できます。</p> <p>有効な値:</p> <p>assigned - 以前のバージョンの Studio からアップグレードした場合のデフォルト</p> <p>not-assigned - Studio 10.20 をインストールした場合のデフォルト</p>

パラメーター	デフォルト値	説明
default.behavior.define.input	説明を参照してください	このプロパティを使用して、フローで入力を定義するときを使用するデフォルトの動作を設定できます。 有効な値: prompt_user - 以前のバージョンの Studio からアップグレードした場合のデフォルト use_constant - Studio 10.20 をインストールした場合のデフォルト
dharma.studio.ui.activegrid	false	true に設定すると、フローエディターでデフォルトでグリッドが表示されます。
engine.assigner.trigger.repeatInterval	10.	
community.home.page.link	https://hpln.hp.com/group/operations-orchestration	コミュニティホームページ
git.history.page.size	100	1 つの Git 履歴ページに表示するアイテムの数を指定します。
git.history.changes.length	500	Git 履歴の [選択したアイテムの下の変更] セクションと [その他の変更] セクションに表示する変更のデフォルトの数を指定します。
hpln.enabled	true	HPLN 検索を有効にするかどうかを指定します。
hpln.connection.timeout	180	HPLN 接続のタイムアウト (秒単位) を指定します。タイムアウトを無制限にする場合は、0 を指定します。
hpln.read.timeout	300	HPLN 読み取りのタイムアウト (秒単位) を指定します。タイムアウトを無制限にする場合は、0 を指定します。
hpln.connection.url	https://api.hpln.hp.com/hpln	HPLN 接続の URL を指定します。
hpln.authentication.service.uri	rest/authenticate	HPLN 認証サービスの URI を指定します。

パラメーター	デフォルト値	説明
hpln.content.packages.service.uri	rest/contentpackages	HPLN コンテンツパッケージサービスの URI を指定します。
hpln.content.offerings.service.uri	rest/contentofferings	HPLN コンテンツ提供サービスの URI を指定します。
hpln.search.service.uri	rest/contentpackages/search	HPLN 検索サービスの URI を指定します。
hpln.search.service.search.param	search	HPLN 検索サービスの検索パラメーターを指定します。
hpln.search.service.size.param	pageSize	HPLN 検索サービスのサイズパラメーターを指定します。
hpln.search.service.provider.param	provider	HPLN 検索サービスのプロバイダーパラメーターを指定します。
hpln.search.service.product.name.param	primaryproduct	HPLN 検索サービスの製品名パラメーターを指定します。
hpln.search.service.product.version.value	10.20	HPLN 検索サービスの製品バージョンパラメーターを指定します。このバージョン番号は Studio のバージョンと同じです。たとえば、Studio のバージョンが 10.20 である場合、hpln.search.service.product.version.param は、10.20 になります。
hpln.search.results.maximum.count	100	HPLN 検索結果の最大数を指定します。この値は 100 未満にする必要があります。
local.debugger.prompt.for.run.user	true	true に設定すると、ローカルデバッガーの実行ユーザーを指定するように求めるメッセージを作成者に表示します。
online.content.packs.link	https://hpln.hp.com/group/operations-orchestration-content-packs	オンラインコンテンツ

注: `studio.properties` ファイルにプロパティがない場合、デフォルト値では、値を入力するように求めるメッセージがユーザーに表示されます。

トラブルシューティング

HP 00 9.x からアップグレードする際のト ラブルシューティング

Studio のユーザーインタフェースアイテムの場 所

HP 00 9.x での作業に慣れているユーザーが、Studio のユーザーインタフェースアイテムを見つけられない場合、以下のトラブルシューティングのヒントが対象アイテムを見つけるのに役立ちます。

リポジトリはどこにありますか？

HP 00 ではリポジトリは使用されなくなりました。ファイルはユーザーのファイルシステムにローカルで保存されるため、共同作業ではソースコントロールアプリケーションを使用することをお勧めします。

[チェックイン] ボタン、[チェックアウト] ボタン、[自分の変更/チェックアウト] ペインはどこにありますか？

ソースコントロールに接続している場合、[SCM 変更] ペインからコミットとチェックアウトを実行可能です。「[ソースコントロールの使用](#)」を参照してください。

[プロジェクト] ペインと [依存関係] ペインに同じアイテムが含まれているように 見えるのはなぜですか？

[プロジェクト] ペインと [依存関係] ペインは以下の点で異なります。

- [プロジェクト] ペインには、プロジェクトで使用できる編集可能なフロー、オペレーション、その他の HP 00 が含まれます。
- [依存関係] ペインには、読み取り専用のフロー、オペレーション、その他の HP 00 が含まれます。プロジェクト内でこれらのオブジェクトを使用できますが、編集はできません。これらのオブジェクトを編集する場合は、[プロジェクト] ペインにコピーします。

オペレーションを作成できないのはなぜですか？

この機能は存在し、オペレーションを作成することは可能です。削除されたのは、組み込みオペレーション (HTTP、SSH、コマンドラインなど) を作成するオプションです。これらは既存のテンプレートオペレーションからコピーする必要があります。

HTTP、SSH、コマンドラインなどの組み込みオペレーションを作成することはできません。既存のテンプレートオペレーションからこれらをコピーし、プラグイン内のアクションから新しいオペレーションを作成する必要があります。詳細については、「[オペレーションの作成](#)」(349ページ)を参照してください。

sleep スクリプトレットを作成できないのはなぜですか?

Sleep スクリプトレットは非推奨になりました。HP 00 10.x では、Rhino または Nashorn でスクリプトレットを記述する必要があります。

カテゴリというドメインタームはどこにありますか?

`Configuration\Categories` フォルダが追加され、カテゴリを保存してフローを分類することができるようになりました。これは、`カテゴリドメインターム`の代わりに使用します。

HP 00 9.x と 10.x のバージョン比較

タスク	HP 00 9.x での処理	HP 00 10.x での処理方法
オペレーションの作成	[新規]>[オペレーション]メニューオプションを使用し、オペレーションの種類を選択します。	アクションプラグインをインポートするか、インポート済みのアクションプラグインからオペレーションを作成します。 「オペレーションの作成」 (349ページ)を参照してください。
共有リポジトリへのフローのチェックイン	Studio で[チェックイン]ボタンを使用します。	プロジェクトをローカルに保存し、ソースコントロール管理ツールを使用して、共有リポジトリにコミットします。 「HP 00 Studio のソースコントロールの使用」 (57ページ)を参照してください。
フローのデプロイと実行	Central のアプリケーションでフローを開き、フローを実行します。	フローをコンテンツパックとしてリリースし、API 経由で HP 00 にデプロイします。 『HP 00 インストールガイド』および『HP 00 Application Program Interface (API) Guide』を参照してください。

タスク	HP 00 9.x での処理	HP 00 10.x での処理方法
マルチインスタンスステップの作成	ステップを右クリックし、[複数インスタンスの切り替え] オプションを選択して、そのステップをマルチインスタンスステップに変更します。次に、ステップの異なるターゲットについて複数のループを作成します。	ステップパレットの[複数のインスタンス] アイコンを作成キャンバスにドラッグします。1つまたは複数のサブフローまたはオペレーションを複数のインスタンスレーンに追加し、値の入力リストでステップの複数のターゲットを設定します。 「マルチインスタンスステップのフローの作成」(289ページ)を参照してください。
オペレーションのアクションの作成	IAction 実装クラスを作成し、コンパイルして .dll または .jar ファイルを作成します。この .dll または .jar ファイルを自分の Web サービスにコピーし、その Web サービスを Studio にインポートします。	アクションプラグインを作成してパッケージ化し、Studio にインポートして、そこから新しいオペレーションを作成します。 「オペレーションの作成」(349ページ)を参照してください。
フローの分類に使用するカテゴリの作成	Configuration\Domain Terms フォルダー内の カテゴリ とよばれるドメインタームで、新しいカテゴリ用に新しい行を追加します。	Configuration\Categories フォルダー内に新しいカテゴリを作成します。

HPLN のトラブルシューティング

HP Live Network 検索を実行しようとしたときにエラーが表示された場合はどうすればよいでしょう

HP 00 ではリポジトリは使用されなくなりました。ファイルはユーザーのファイルシステムにローカルで保存されるため、共同作業ではソースコントロールアプリケーションを使用することをお勧めします。

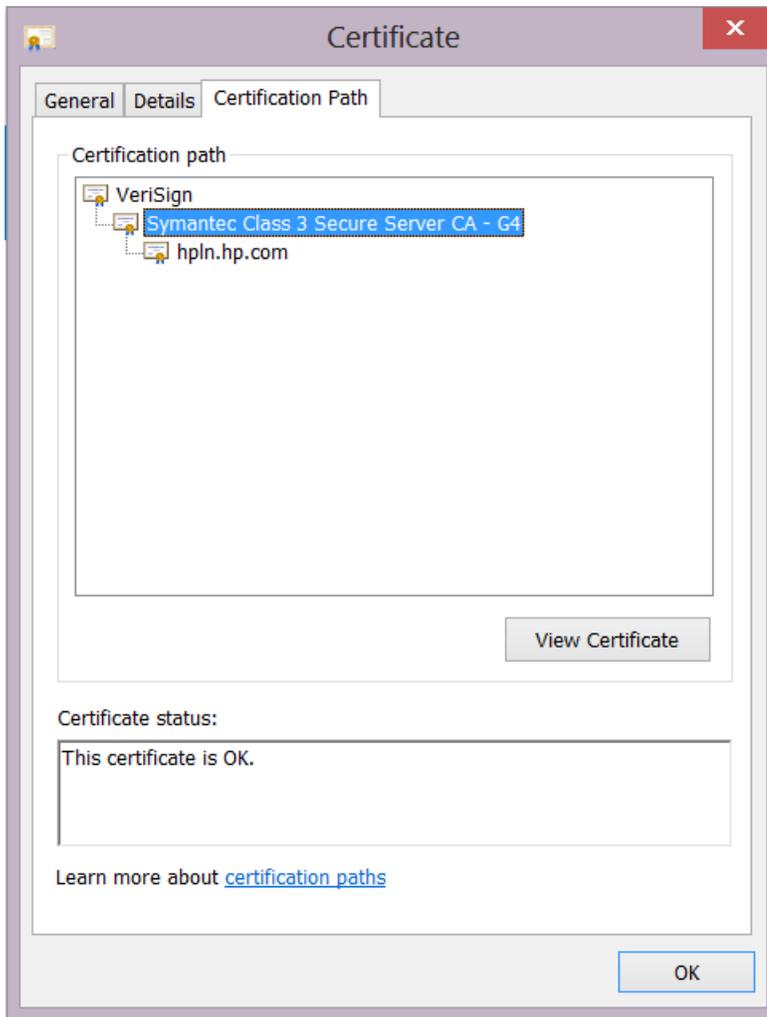
HP Live Network 検索を実行したときに次のエラーが表示されることがあります。

```
sun.security.validator.ValidatorException:PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to
find valid certification path to requested target
```

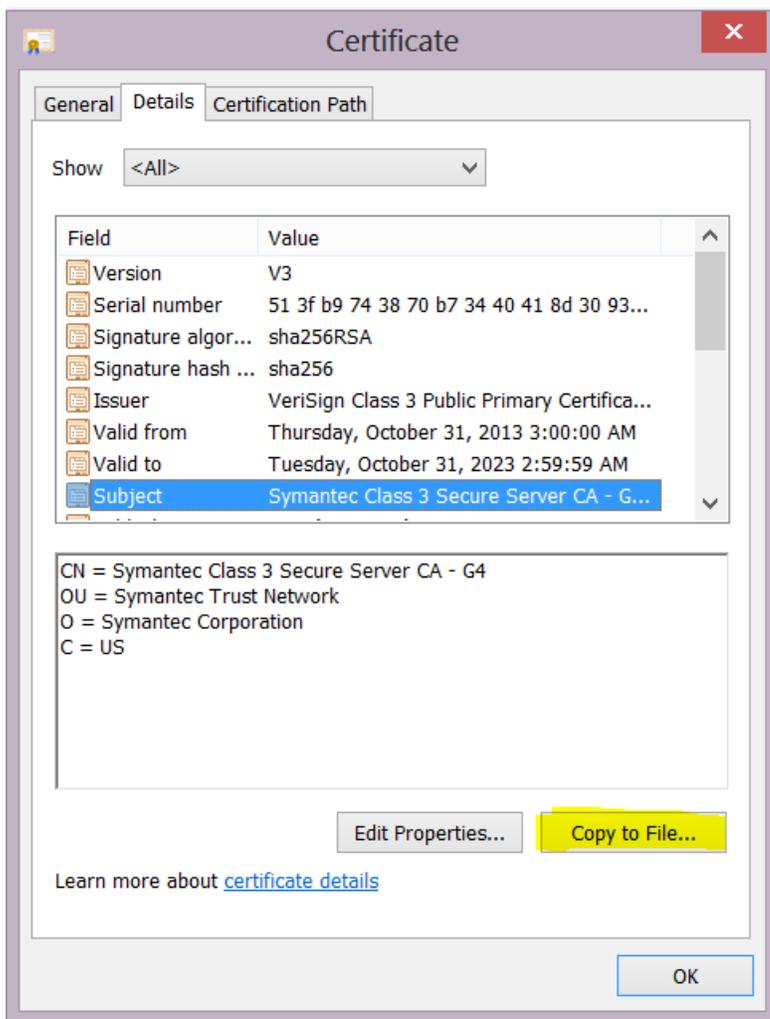
この問題を解決するには:

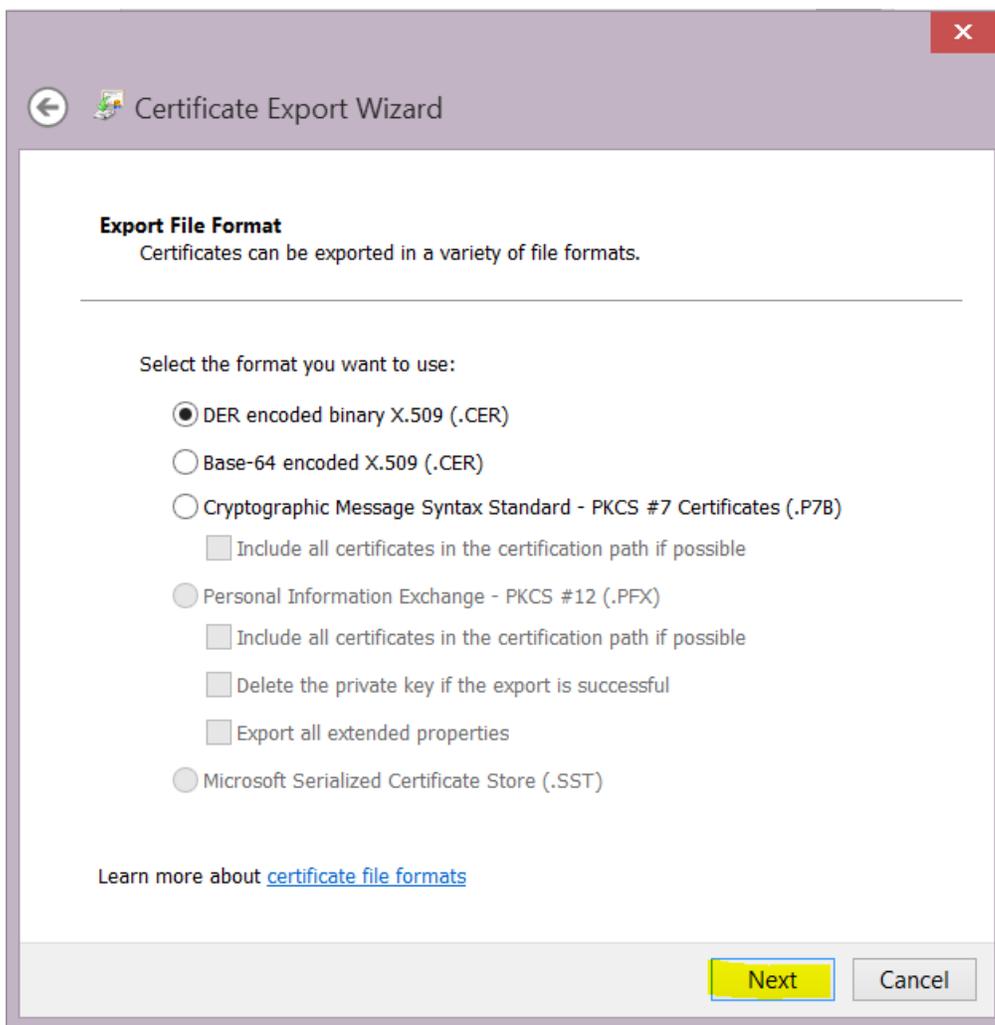
1. Studio を閉じます。
2. HP Live Network ルート証明書 (Symantec Class 3 Secure Server CA - G4 という名前のチェーン

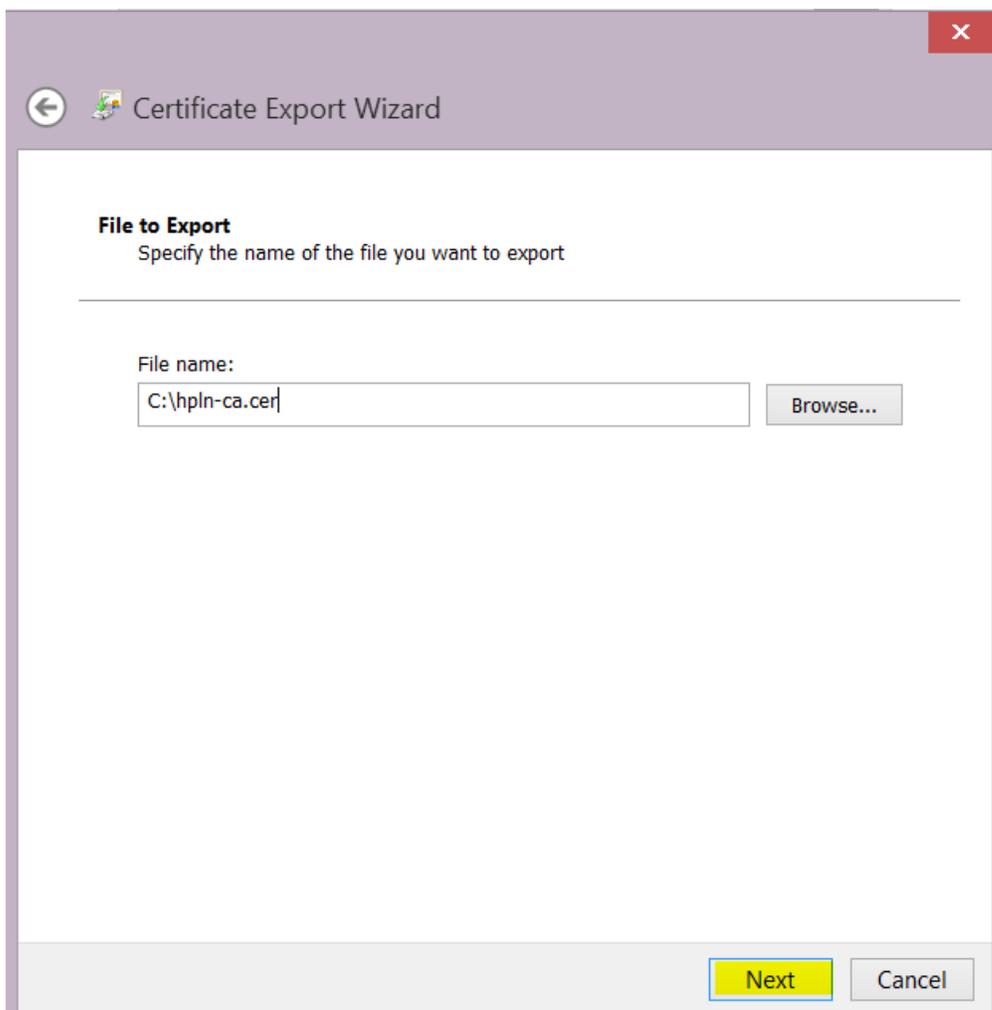
の 2 番目の証明書) を DER エンコードされたバイナリ X.509 の形式で **hpln-ca.cer** という名前のファイルにエクスポートします。

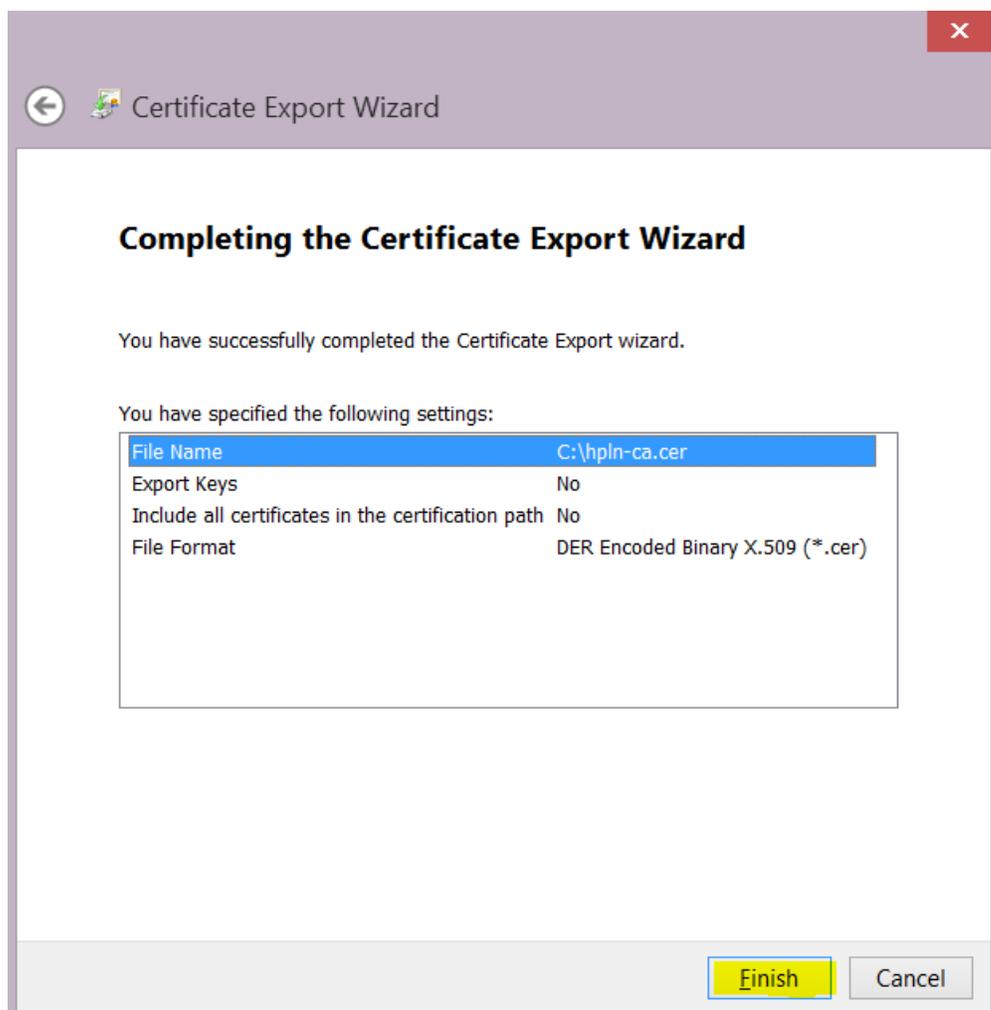


以下に示すように、ブラウザを使用して <https://api.hpln.hp.com/hpln> にアクセスして、ルート証明書をエクスポートすることができます。









- 証明書ファイル **hpln-ca.cer** を <oo インストールフォルダー>\studio\var\security フォルダにコピーします。
- 次のコマンドを使用して証明書を Studio の信頼ストアにインポートします。

```
<oo インストールフォルダー>\java\bin\keytool -import -alias hplnroot -keystore client.truststore -file hpln-ca.cer
```

注:

- キースタアのデフォルトのパスワードは **changeit** です。デフォルトのパスワードを変更した場合、要求されたときには、<oo インストールフォルダー>\studio\Studio.l4j.ini ファイルの **Djavadoc.net.ssl.trustStorePassword** パラメーターの値を使用します。
- Studio 信頼ストアのパスワードを変更するには、“oo” フォルダの Studio.properties ファイルに **client.truststore.password** プロパティを追加し、非表示形式でパスワードの値を指定します。

```
client.truststore.password=={OBFUSCATED}6L9+NqBjKYp5heuvMEzg0g==
```

このプロパティが定義されていない場合、Studio はシステムプロパティ **javax.net.ssl.trustStorePassword** にフォールバックして、信頼ストアのパスワードを取得します。

5. Studio を再起動します。

GIT のトラブルシューティング

Git の操作を実行しようとしたときに "System cannot find specified file" というエラーが表示された場合はどうすればよいでしょう

Git SCM 操作を時刻したときに、次のエラーが表示される場合があります。

```
"Cannot run program "C:\<oo_installation_folder>\studio\Git\bin\git" (in directory "C:\<user_home_folder> \.oo\Workspace"):  
CreateProcess error=2, The system cannot find the file specified "
```

1. 次の URL から Git クライアントをダウンロードします。
<https://github.com/msysgit/msysgit/releases/download/Git-1.9.5-preview20150319/Git-1.9.5-preview20150319.exe>
2. **bin** フォルダーが **<oo_installation_folder>/studio/Git** のすぐ下になるように、Git クライアントを **<oo_installation_folder>/studio/Git** に保存します。Git インストールウィザードでは、デフォルトのオプションを使用します。

この代わりに、バージョン **git-1.9.5-preview20150319** の Git クライアントインストールがすでにローカルディスク上にある場合は、以下の手順を実行して、その Git インストールを使用するよう Studio に指示します。

1. Studio を閉じます。
2. ユーザーホームフォルダー **C:\Users\<user>\.oo** (Studio ワークスペースの場所) に移動し、**Studio.properties** ファイルを探します。
3. **Studio.properties** ファイルを修正して、ファイルの末尾に次のプロパティを追加します。

```
studio.git.installation.location=<git-1.9.5-preview20150319_installation_folder>
```

例:

```
studio.git.installation.location=C:/Program Files (x86)/Git
```

bin フォルダーは、**C:/Program Files (x86)/Git** のすぐ下になります。パスの区切り文字として **/** を使用することに注意してください。

4. **Studio.properties** ファイルを保存し、Studio を起動します。

別のバージョンの Git クライアントがインストールされている場合は、**git-1.9.5-preview20150319** バージョンを Studio で使用する必要があります。

ログビューアーでの Studio エラーの表示

ログビューアーは、発生したすべてのエラーを Studio 内の 1 つの場所で表示できる便利なツールです。エラーは、**Studio.log** ログファイルに表示される時と同じ方法で表示されます。

ログビューアーには、次の 2 種類のエラーが表示されます。

- 致命的なエラー (Studio.log では FATAL のラベルが付けられます)
- エラー (Studio.log では ERROR のラベルが付けられます)

注: 警告 (WARN) と情報 (INFO) のメッセージはログビューアーには表示されませんが、**Studio.log** ファイルには表示されます。

[ウィンドウ] メニューの [**Studio ログビューアー**] オプションが選択されている場合、Studio を開いたときに [ログビューアー] タブが Studio ウィンドウの下部にあるタブの 1 つとして表示されます。

実行する操作

[ログビューアー] タブの有効化

1. [ウィンドウ] メニューで [**Studio ログビューアー**] を選択します。

