



Hewlett Packard
Enterprise

Operations Orchestration

ソフトウェアバージョン: 10.60

Windows

Studioオーサリングガイド

ドキュメントリリース日: 2016年5月 (英語版)

ソフトウェアリリース日: 2016年5月

ご注意

保証

Hewlett Packard Enterprise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、Hewlett Packard Enterprise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、Hewlett Packard Enterpriseからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

© 2005-2016 Hewlett Packard Enterprise Development LP

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更新されます。
- 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/>

このサイトを利用するには、HP Passportに登録してサインインする必要があります。HP Passport IDに登録するには、HPソフトウェアサポートサイトで **[Register]** をクリックするか、HP Passportログインページで **[Create an Account]** 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、HPEの営業担当にお問い合わせください。

目次

『HPE Operations Orchestration Studioオーサリングガイド』へようこそ	7
図解によるOO Studio概要	7
OO Studioの開始 – ワークフローの主なステップ	22
OO Studioウィンドウの外観の調整	23
OO Studioでの各言語の使用 - ローカライズ	24
プロジェクトの使用	28
ワークスペースの管理	28
複数のワークスペースの使用	28
プロジェクトの管理	32
[プロジェクト] ペインを使用したフォルダーの管理	40
OO Studioのソースコントロールの使用	42
OO Studioのソースコントロールとは	42
Subversionソースコントロール管理の使用	44
用語集	44
初期のソースコントロールリポジトリの作成	45
SVNでの複数の作成者による作業	50
ロックポリシーの強制	51
Gitソース管理システムの使用	71
Gitの用語	71
StudioでGitを使用する前に	74
Gitのブランチのサポート	74
Gitリポジトリログについて	75
Gitでの競合の処理	79
認証オプション	81
コンテンツパックの使用	116
コンテンツパックのプロジェクトへのインポート	116
プロジェクトのコンテンツパックと依存関係の管理	118
依存関係	119
構成アイテムの管理	134
構成アイテムの使用	134
構成アイテムの検索	134

構成アイテムフォルダーの使用	135
カテゴリの設定	138
ドメインタームの設定	140
グループエイリアスの設定	143
役割エイリアスの設定	149
スクリプトレットの設定	151
選択リストの設定	156
システムアカウントの設定	160
システム評価子の設定	163
システムフィルターの設定	170
システムプロパティの設定	176
フロー作成 – 基本	180
フローの作成 – 手順	180
新しいフローの作成	184
フロー内でのステップの作成	186
フローの外観の調整	193
フローの変更	196
入力の作成	202
入力ソースの指定	213
入力データの評価	226
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	247
オペレーションの出力の設定	248
ステップの結果の設定	252
出力および結果のフィルタリング	261
変数の使用	279
リターンステップの作成	285
高度な作成	290
フロー内でのサブフローの作成	290
並列分岐ステップのフローの作成	293
マルチインスタンスステップのフローの作成	297
フロー内でのスクリプトレットの使用	307
フローでの正規表現の使用	313

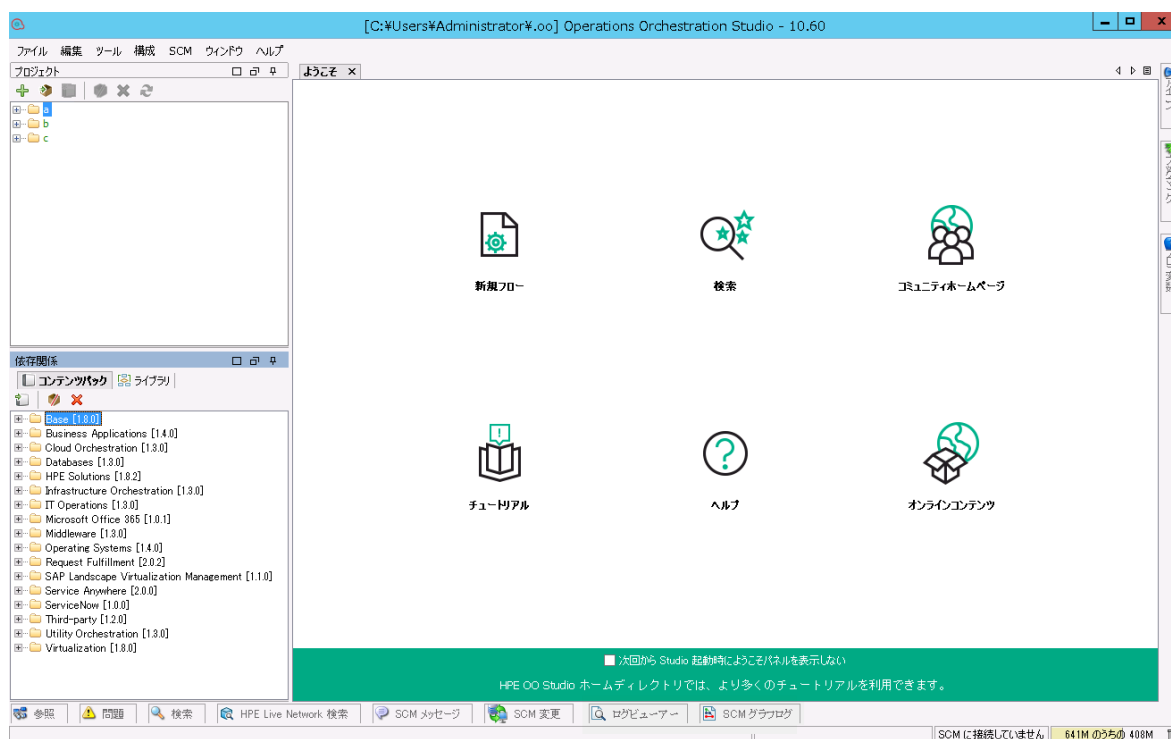
HPE Live Networkの内容をStudioから検索	318
検索の種類	318
hpln-index-generatorツール	319
参考資料	322
コンテンツの検査	324
[問題] ペインでのフローの検査	324
フローのテストとデバッグ	326
ユーザー認証が必要なフローのデバッグ	327
複雑なフローのデバッグ	340
リモートCentralのStudioでのデバッグ	341
一般的なワークフロー	343
リモートデバッガー接続による証明書の自動インポート	347
リモートCentralでのフローのデバッグ	348
コンテンツパックのエクスポート	352
コンテンツパックのバージョン管理のライフサイクル	352
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	353
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	354
フローとオペレーションの管理	366
オペレーションの作成	366
フローまたはオペレーションの検索	375
オペレーションまたはサブフローへの参照の検索および置換	381
グローバル検索および置換テキスト	385
フローおよびオペレーションのコピー	387
ソフトコピーからハードコピーへの切り替え	390
ハードコピー内のプラグインの置き換え	390
フローとオペレーションの使用方法の詳細	391
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	393
フローとオペレーションのバージョン履歴管理	399
フローおよびオペレーションのブックマーク	402
Studioプロパティの設定	408
[割り当て元]と[割り当て先]のデフォルト値を定義する	408
[未指定時]のデフォルト値を[定数を使用する]に変更する	409
コンテキストインスペクターに特殊文字を表示する	410
参考資料	415

トラブルシューティング	420
OO 9.xからアップグレードする際のStudioのトラブルシューティング	420
Studioのユーザーインターフェイスアイテムの場所	420
OO 9.xと10.xのバージョン比較	421
HPE Live Networkのトラブルシューティング	422
GITのトラブルシューティング	428
SVNのトラブルシューティング	429
ログビューアーでのStudioエラーの表示	430

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

HPE 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HPE 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]** フォルダー。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す。
- **[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。詳細については、「[\[問題\] ペイン](#)」(21ページ)を参照してください。
- **[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。詳細については、「[OO Studioのソースコントロールの使用](#)」(42ページ)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。詳細については、「[OO Studioのソースコントロールの使用](#)」(42ページ)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](#)」(71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](#)」(430ページ)を参照してください。

注: [ログビューアー] タブは、[ウィンドウ] メニューの [Studioログビューアー] を選択して有効にした後にのみ表示されます。

- **[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。
- **HPE 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、HPE 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、「[HPE Live Networkの内容をStudioから検索](#)」(318ページ)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。

参考資料

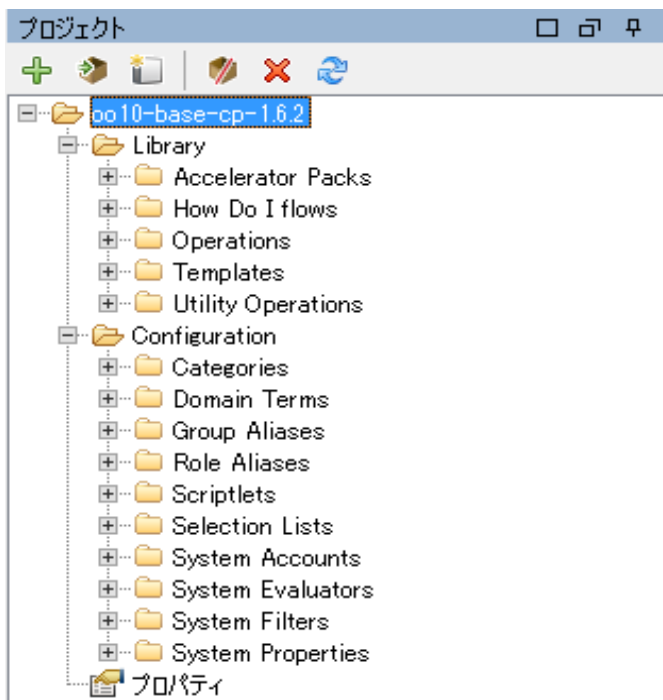
[プロジェクト] ペイン




[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



GUIアイテム	説明
新規プロジェクト 	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。
コンテンツパックを作成 	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](#)」(28ページ)を参照してください。

[依存関係] ペイン

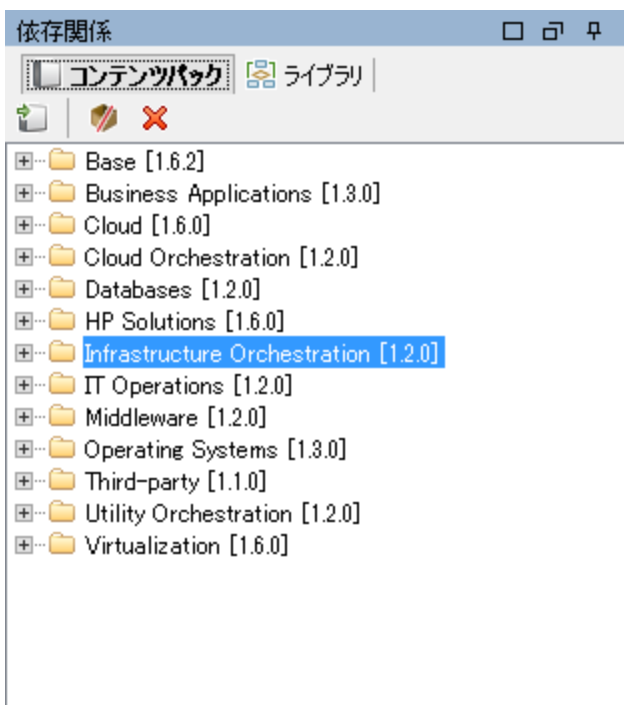
[依存関係] ペインには、使用可能なコンテンツパックが、オペレーションおよびフローが格納されたフォルダーとともに表示されます。

[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。

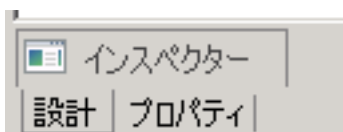


[作成] ペイン

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。

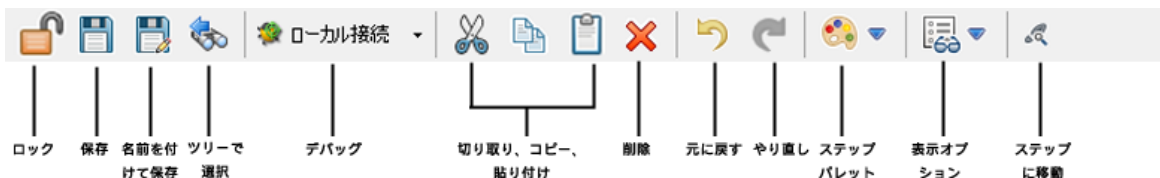
- [設計] タブ。フロー図で作業し、ステップやステップ間のつながりを追加します。
- [プロパティ] タブ。[プロパティ] シートが表示され、フローやオペレーションの選択リスト、フィルター、スクリプトレットなどのプロパティを設定できます。
- [インスペクター] タブ。インスペクターが表示され、個々のステップとトランジションのプロパティを設定できます。








[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。

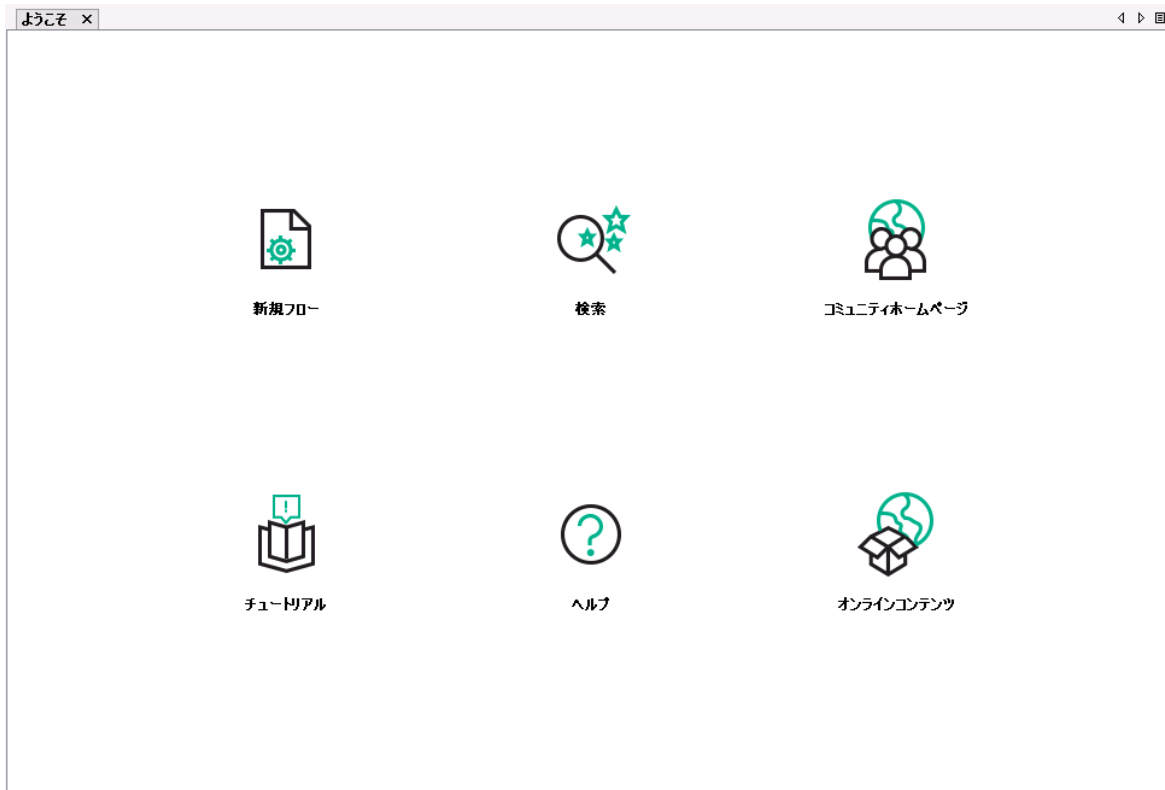


ボタン	機能
 ステップパレット	ステップパレットが開きます。ステップオブジェクトをキャンバスにドラッグします。
 表示オプション	表示オプションパレットの表示
 ツリーで選択	使用するフローまたはオペレーションを選択するためにライブラリツリーを展開します。
 フローのデバッグ	Centralデバッガーの接続を追加または編集します。
 ステップに移動	フロー内の特定のステップに移動できます。移動先のステップの名前を入力するか、ステップの最初の何文字かを入力してリストから選択します。

作成ペインの使用の詳細については、「[フロー作成 – 基本](#)」(180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HPE OOのチュートリアルを表示します。
ヘルプ	クリックしてHPE OO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、HPE OO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起動時にようこそ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うこそ画面が表示されなくなります。

注：HPE OO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いフローの作成の詳細については、「[新しいフローの作成](#)」(184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](#)」(375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
Success 	[Success] リターンステップをフローにドラッグできます。
Diagnosed 	[Diagnosed] リターンステップをフローにドラッグできます。
No Action Taken 	[No Action Taken] リターンステップをフローにドラッグできません。
Failure 	[Failure] 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](#)」(285ページ)を参照してください。

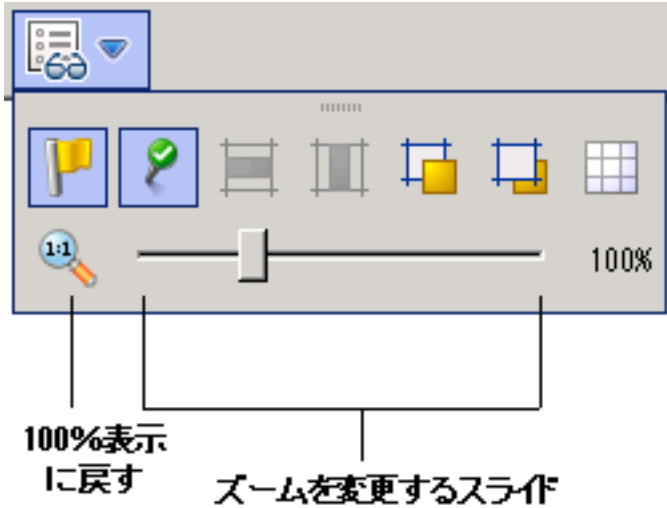
並列分岐ステップとマルチインスタンスステップの詳細については、「[高度な作成](#)」(290ページ)を参照してください。








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。

ボタン	説明
	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](#)」(193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	必須	種類	割り当て元	未指定時	割り当て先
host	<input checked="" type="checkbox"/>		host	ユーザーへのプロンプト (テキストから)	host
badMailPath	<input checked="" type="checkbox"/>		badMailPath	定数の使用: c:\$Program Files*Exc...	badMailPath
exchangePath	<input checked="" type="checkbox"/>		exchangePath	定数の使用: c:\$Program Files*Exc...	exchangePath
Old Threshold	<input checked="" type="checkbox"/>		Old Threshold	ユーザーへのプロンプト (テキストから)	Old Threshold
username	<input type="checkbox"/>		username	ユーザーへのプロンプト (テキストから)	username
password	<input type="checkbox"/>		password	ユーザーへのプロンプト (テキストから)	password
version	<input type="checkbox"/>		version	ユーザーへのプロンプト (選択リストから:)	version

[プロパティ]シートの使用の詳細については、「[入力の作成](#)」(202ページ)および「[オペレーションの出力の設定](#)」(248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	機密データ	必須	種類	割り当て元	未指定時	割り当て先
host	<input type="checkbox"/>	<input type="checkbox"/>	host	host	定数の使用	<未割り当て>
port	<input type="checkbox"/>	<input type="checkbox"/>	port	port	定数の使用	<未割り当て>
command	<input type="checkbox"/>	<input type="checkbox"/>	command	command	定数の使用	<未割り当て>
pty	<input type="checkbox"/>	<input type="checkbox"/>	pty	pty	定数の使用	<未割り当て>
username	<input type="checkbox"/>	<input type="checkbox"/>	username	username	定数の使用	<未割り当て>
password	<input type="checkbox"/>	<input type="checkbox"/>	password	password	定数の使用	<未割り当て>
arguments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	arguments	arguments	定数の使用	<未割り当て>
privateKeyFile	<input type="checkbox"/>	<input type="checkbox"/>	privateKeyFile	privateKeyFile	定数の使用	<未割り当て>
timeout	<input type="checkbox"/>	<input type="checkbox"/>	timeout	timeout	定数の使用	<未割り当て>
characterSet	<input type="checkbox"/>	<input type="checkbox"/>	characterSet	characterSet	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](#)」(202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。

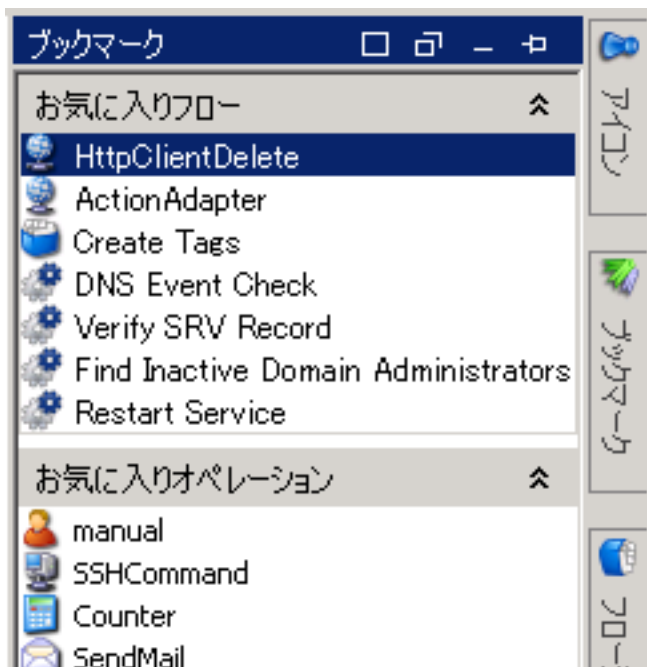
トランジションのインスペクターの使用の詳細については、「[トランジションの作成](#)」(228ページ)を参照してください。

[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。

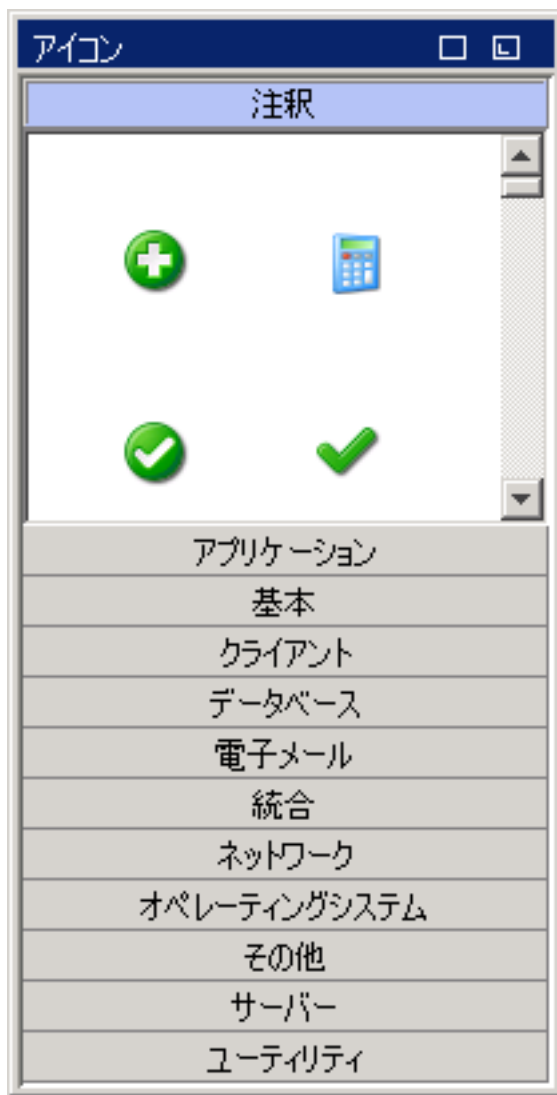
よく使用するフローやオペレーションをライブラリからドラッグして [ブックマーク] ペインに追加することが可能です。 [ブックマーク] ペインから [プロジェクト] ペインにフローおよびオペレーションをドラッグして、プロジェクトにコピーすることもできます。

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](#)」(402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](#)」(196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" に割り当てること...	-		
"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](#)」(279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。

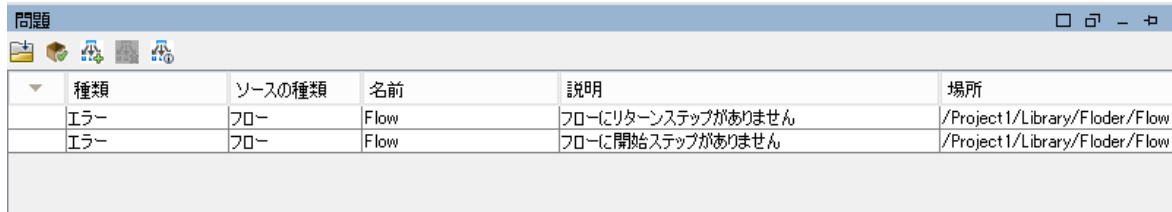
- **[これを使用する項目は?]** - 対象のオペレーションまたはフローから作成されたステップが含まれるフローを識別します。
- **[これが使用する項目は?]** - 対象のオペレーションまたはフローが使用するオブジェクト (選択リスト、グループに割り当てられた権限、システムフィルターなど) を識別します。フローの場合は、フローのステップを作成したときのベースになったオペレーションおよびサブフローが含まれます。

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](#)」(391ページ)を参照してください。

[問題] ペイン

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

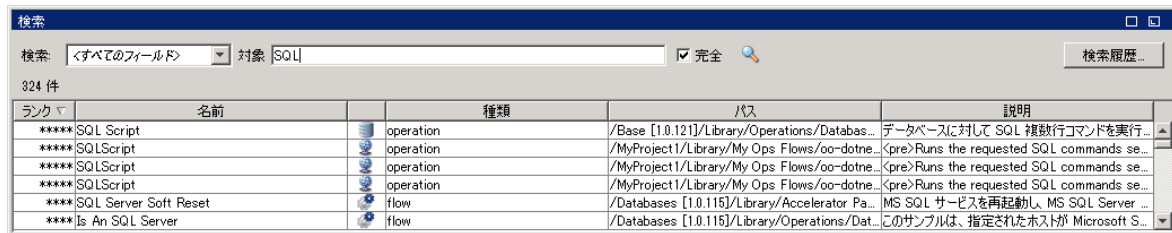


種類	ソースの種類	名前	説明	場所
エラー	フロー	Flow	フローにリターンステップがありません	/Project 1/Library/Floder/Flow
エラー	フロー	Flow	フローに開始ステップがありません	/Project 1/Library/Floder/Flow

[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](#)」(324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。

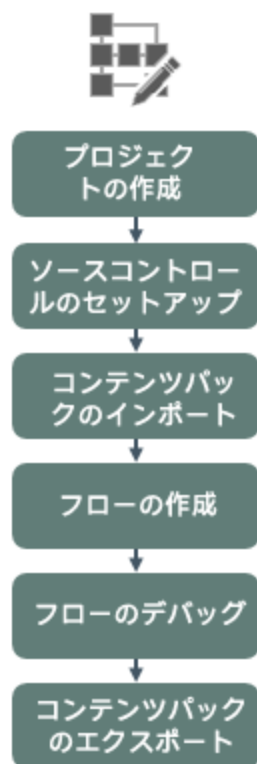


ランク	名前	種類	パス	説明
****	SQL Script	operation	/Base [1.0.121]/Library/Operations/Databas...	データベースに対して SQL 複数行コマンドを実行...
****	SQLScript	operation	/MyProject 1/Library/My Ops Flows/oo-dotne...	<pre>Runs the requested SQL commands se...
****	SQLScript	operation	/MyProject 1/Library/My Ops Flows/oo-dotne...	<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](#)」(375ページ)を参照してください。

OO Studioの開始 – ワークフローの主なステップ

ここでは、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。



ボタンをクリックすると、関連するトピックに移動します。

1. **新規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」\(32ページ\)](#)を参照してください。
2. **ソースコントロール管理のセットアップ** - SVNまたはGitのいずれかのソースコントロール管理システムをセットアップします。
[「OO Studioのソースコントロールの使用」\(42ページ\)](#)を参照してください。
3. **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](#)」(116ページ)を参照してください。

4. **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](#)」(180ページ)および「[高度な作成](#)」(290ページ)を参照してください。

5. **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](#)」(326ページ)を参照してください。

6. **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、OO Centralへ昇格します。

「[コンテンツパックのエクスポート](#)」(352ページ)を参照してください。

OO Studioウィンドウの外観の調整


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


ペインを元のサイズに復元する

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

Studioウィンドウの脇にペインを固定する

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

ペインの固定を解除する

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

ペインのサイズを調整する

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

Studioウィンドウをデフォルトのレイアウトにリセットする

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

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。

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。OO 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

言語コード

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 (user.language)	地域のロケールコード (user.region)	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

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。

例:

- デフォルトのコンテンツパックのプロパティファイルには、**cp.properties**という名前が付いています。
- フランス語のファイルの名前は**cp_fr.properties**です。
- 中国語のファイルの名前は**cp_zh_CN.properties**です。

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。

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。

- `C:\Users\<<ログインユーザーの名前>\.oo\studio.properties`ファイルを開きます。
- 次の行を検索します。

```
user.language=
```

3. **言語コード**表に一覧されている**user.language**コードを使用して言語ロケールを追加します。

例: `user.language=ja`

注: 別の言語インター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

また、コンテンツパックをローカライズできます。「[コンテンツパックのローカライズ](#)」(25ページ)を参照してください。

バージョン10.5x以降では、複数のワークスペースで作業を行うことができます。したがって、各ワークスペースにはそれぞれ固有の**studio.properties**ファイルがあり、これらは**.oo**フォルダーの下に置かれます。

- 新しいワークスペースすべてに同じプロパティを使用する場合、**Studio\conf**の下にある**studio.properties.template**ファイルを編集することができます。
- ワークスペースごとに異なる設定を構成するために、各**studio.properties**ファイルを個別に編集できます。これらのファイルは、各ワークスペース内の**.oo**フォルダーの下にあります。

プロジェクトの使用

プロジェクトは、業務で使用するフロー、オペレーション、フォルダー、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。

ワークスペースの管理

Studioで「ワークスペース」とは、OOプロジェクトを保存しているフォルダーのことです。デフォルトのワークスペースの場所はC:\Users\<<ログインユーザーの名前>\.oolの下ですが、別の場所を指定することもできます。

複数のワークスペースの使用

ソースコントロールを使用する場合など、複数のワークスペースを使用しなければならない場合があります。

SVNでの作業を特定のワークスペースで行い、Gitでの作業を行うために別のワークスペースに切り替えることができます。このとき、SVNワークスペースから接続を解除する必要はなく、さらに変更をコミットしていない場合もSVNで行った作業が失われることはありません。

Studioで複数のワークスペースを使用する場合、プロキシ設定やデバッガー接続、作業しているプロジェクト、使用しているコンテンツパックなどのカスタムのワークスペース固有の設定を簡単に保存できます。

さらに、まったく異なるいくつかのプロジェクトで同時に作業を行い、必要に応じてプロジェクトを切り替えることができます。このようにすると、特にプロジェクトをほかの作成者と共有する場合に、より効率的に作業内容を整理できます。

注: すべてのワークスペースには、それぞれ固有のログファイルを備えた独自のlogsフォルダーが含まれます。

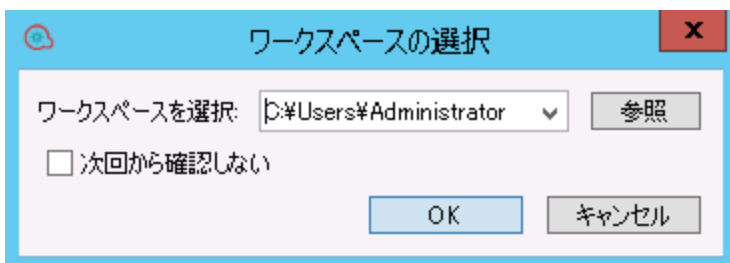
注: Studioで複数のワークスペースを作成する場合は、各ユーザーは読み取りと書き込みのアクセス許可を持っているフォルダーの下にのみワークスペースを作成することをお勧めします。

ワークスペースをパブリックフォルダーの下に作成すると、すべてのユーザーがアクセスできるため、改ざんや機密情報の漏洩が発生しやすくなります。

実行する操作

起動時にワークスペースの定義

Studioを起動すると、[ワークスペースの選択] ダイアログボックスでワークスペースの選択を求めるメッセージが表示されます。



1. 次のいずれかを実行します。
 - リストからワークスペースを選択します。
 - ワークスペースパスを入力します。

注: パスは.ooを含めないようにする必要があります。

- [参照] をクリックして、別の場所からワークスペースを選択します。
2. (オプション)Studioの起動時にこのメッセージが再度表示されないようにし、最近使用したワークスペースを常に開くようにするには、[これをデフォルトとして使用し、次回から確認しない] を選択します。

Studioを次に起動するときには、最後に使用したワークスペースが使用され、[ワークスペースの選択] ダイアログは表示されません。

[ファイル] > [ワークスペースの切り替え] を使用して別のワークスペースに切り替えてからStudioを閉じると、Studioを次に起動したときには、該当するワークスペースが使用されます。

3. [OK] をクリックします。

パスが既存の有効なワークスペースを示している場合、設定、プロジェクト、およびコンテンツパックがロードされ、このワークスペースに設定したSVNまたはGitに接続されます。また、プロキシ、HPLN、デバッガーの各設定についても、このワークスペースでの設定どおりになります。

入力したパスが既存のワークスペースを示していない場合、Studioは新しいワークスペースを作成します。

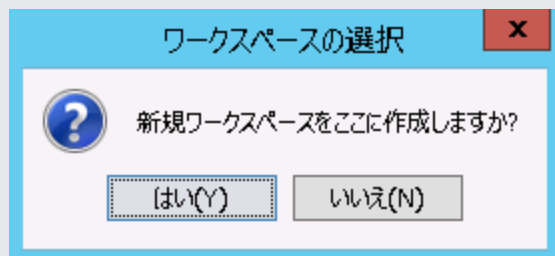
注: このワークスペースが (Studioの別のインスタンスによって) すでに使用されている場合、警告メッセージが表示されます。

別のワークスペースへの切り替え

Studioでの作業中に別のワークスペースに切り替えることもできます。

1. [ファイル]メニューから[ワークスペースの切り替え]を選択します。
2. 次のいずれかを実行します。
 - リストからワークスペースの場所を選択します。
 - [その他]をクリックして、別の場所からワークスペースを指定します。

注: ワークスペースを含まないフォルダーを選択すると、このフォルダーに新しいワークスペースを作成するかどうかを確認されます。

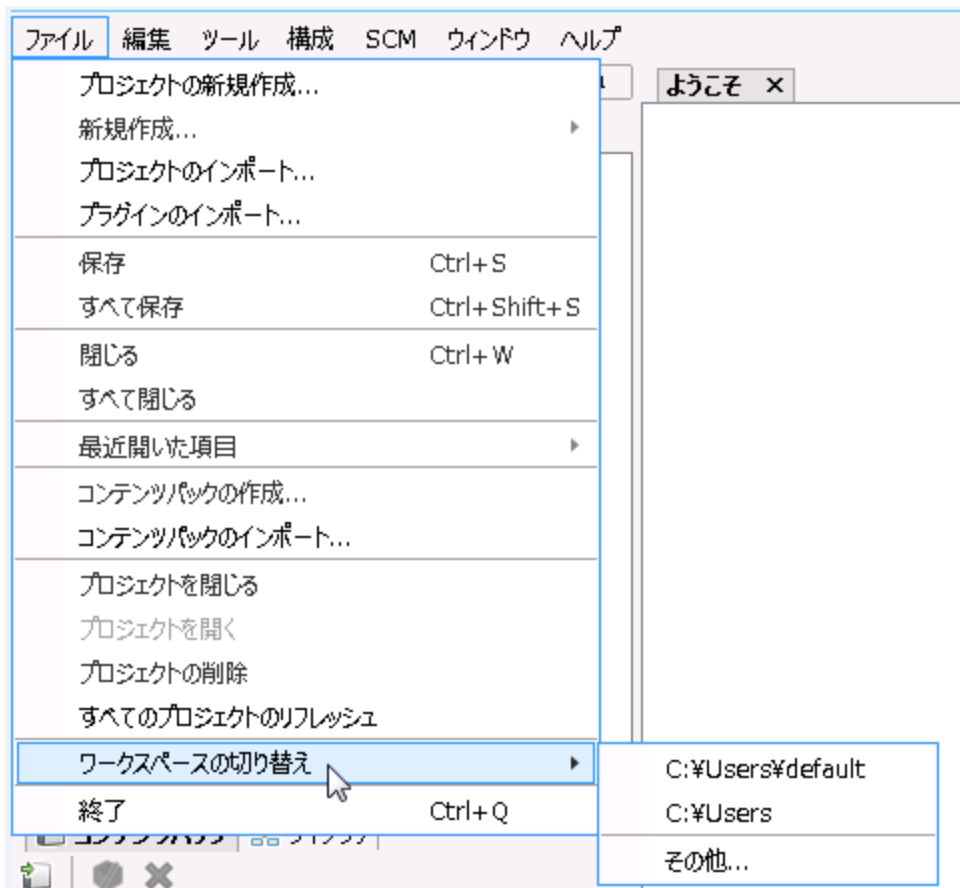


[はい]をクリックして、新しいワークスペースを作成します。

選択したワークスペースが開きます。

参考資料

[ファイル] メニュー



メニューアイテム	説明
ワークスペースの切り替え	リストから切り替えるワークスペースを選択するか、[その他...]を選択して別のワークスペースを指定します。 注: ワークスペースを含まないフォルダーを選択すると、このフォルダーに新しいワークスペースを作成するかどうかを確認されます。

プロジェクトの管理

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

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

Studio 10.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、「[OO Studioのソースコントロールの使用](#)」(42ページ)を参照してください。

重複するUUIDの処理

Studioでは、フロー、オペレーション、構成アイテムごとに一意のUUIDを割り当てる必要があります。ただし、重複するUUIDを持つ同じタイプとサブタイプのアイテムが誤って作成される場合があります。たとえば、SCMで元に戻すオペレーションを何度か実行したときや、Gitでのブランチの操作中に競合を解決したときなどに、このようなことが起こります。

1つのプロジェクト、複数の(2つ以上の)プロジェクト、複数のカスタムコンテンツパック、またはコンテンツパックとプロジェクト間で、重複するUUIDを持つ同じタイプとサブタイプのアイテムが存在する場合、これらはプロジェクトツリーに問題として(赤色で、または下線付きで)表示され、[問題] ペインに表示されます。

注: 同じタイプとサブタイプとは、同じサブタイプのフローを含むフロー、同じサブタイプのオペレーションを含むオペレーション、同じサブタイプの構成アイテムを含む構成アイテムのことです。

重複するUUIDを持つアイテムには次の特性があります。

- これらのアイテムは読み取り専用であるため、移動したり名前を変更したりすることができません。
- これらのアイテムはキャンバスにドラッグできません。
- 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。
- これらのアイテムはエクスポートしたコンテンツパックから除外されます。
- 重複するUUIDを含むフローをデバッグすることはできません。検査でエラーが発生します。
- 参照を使用できません。[これを使用する項目は?] および[これが使用する項目は?]機能が無効になります。

- これらのアイテムはコミットできません。これらのアイテムを含むフォルダーおよびプロジェクトはコミットできませんが、重複するUUIDを持つアイテムをコミットしようとしていることを知らせるエラーメッセージが表示されます。

重複するUUIDを解決するには、重複するUUIDを持つアイテムを削除して、元のアイテムを残します。

両方のアイテムを残す場合は、[コピー/貼り付け]または[重複]コマンドを使用して、異なるUUIDを持つアイテムのコピーを保存してから、コピー元のアイテムを削除します。

コンテンツパック内の重複するUUIDを解決するには、コンテンツパックを削除または閉じます。

実行する操作

プロジェクトを作成する

1. [ファイル] > [新規プロジェクト] を選択します。

注: [プロジェクト] ペインで [新規プロジェクト]  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、「[OO Studioのソースコントロールの使用](#)」(42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、「[OO Studioのソースコントロールの使用](#)」(42ページ)を参照してください。

[ツール] > [検証] メニューからプロジェクトを検証すると、その結果 [問題] ペインにそのプロジェクトの問題が表示されます。たとえば、プロジェクトに重複するUUIDを持つアイテムが含まれていると、これらのアイテムは [問題] ペインに表示されます。

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。

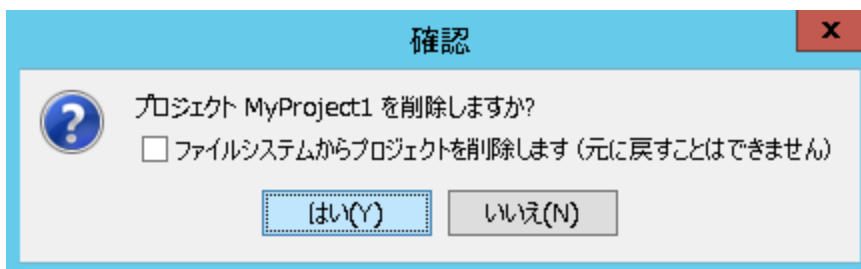
注: [プロジェクト] ペインで [開く]  ボタンをクリックすることもできます。

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](#)」(387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

次のいずれかを実行します。

- [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。
- プロジェクトを右クリックして [プロパティ] を選択します。
- [プロジェクト] 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、プロジェクトの **パブリッシャー**、**バージョン**、**説明**、および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](#)」(44ページ)または「[Gitソース管理システムの使用](#)」(71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。

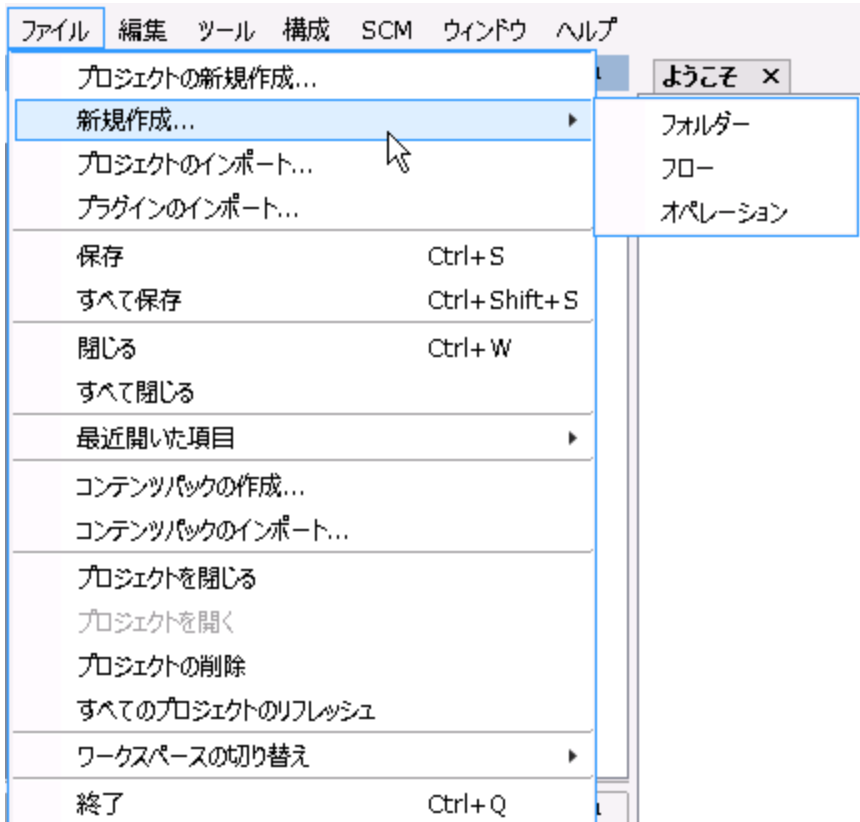
プロジェクト内のオブジェクトのプロパティを表示する

- [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

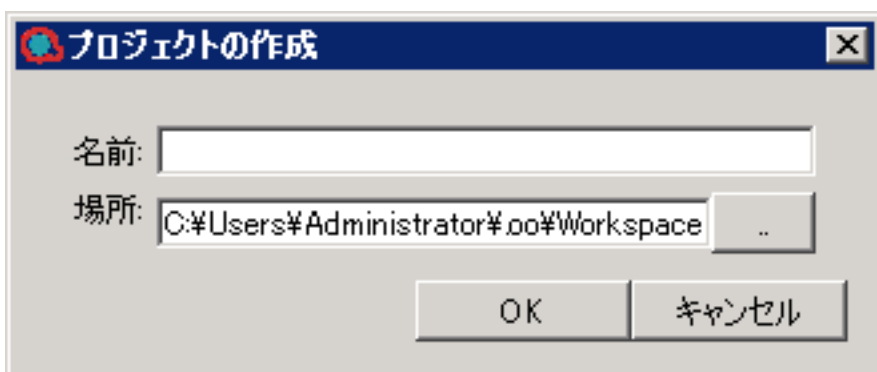
[ファイル] メニュー




メニューアイテム	説明
新規プロジェクト	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」(366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 」(116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	HPE OOを終了します。

[プロジェクトの作成] ダイアログボックス



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	新しいプロジェクトを格納する、ワークスペースの場所を入力します。

- 場所へのパスを入力します。
- 場所を参照するには、参照ボタン  をクリックします。

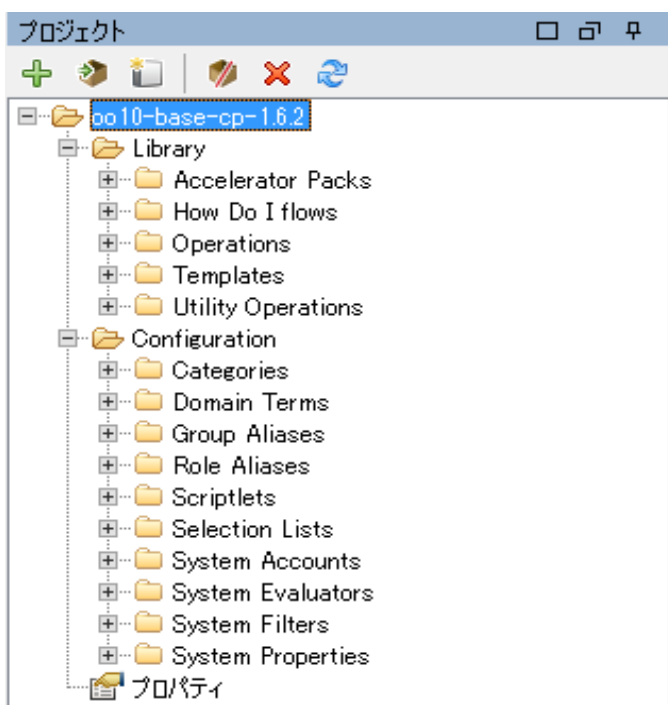
[プロジェクト] ペイン



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






- **Library** フォルダーには、フローとオペレーションが含まれます。
- **Configuration** フォルダーには、オペレーション結果の処理、レポートの作成、およびフローの実行を容易にするために使用できるその他の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



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

[プロジェクト] ペインを使用したフォルダーの管理

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

実行する操作

フォルダーを作成する

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

重要: プロジェクト内に要素 (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。

フォルダー名を変更する

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

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。

注: 重複するアイテムを含むフォルダーまたはフォルダーの階層を削除することはできますが、切り取った後で貼り付けることはできません。

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。

OO 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。

OO Studioのソースコントロールとは

OO 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](#)」(44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](#)」(71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、次の手順を実行します。

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

SCM 接続

種類
 Subversion Git

SCM リポジトリ
 URL

認証
 種類

ユーザー名

パスワード

ユーザー情報
 フルネーム
 電子メールアドレス

重要
 ⚠ Git 機能を使用するには、以下から Git クライアントをインストールする必要があります
[こちらをご覧ください。](#)

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。

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せん。
クローン/接続解除	Gitのみ: 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。 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。
接続解除	<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> <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>
チェックアウト/接続解除	<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>
サーバー証明書を信頼	<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>
資格情報: Windows認証の使用	<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>
ユーザー名	<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>
パスワード	

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、最初に行う操作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。変更を確認するには、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。

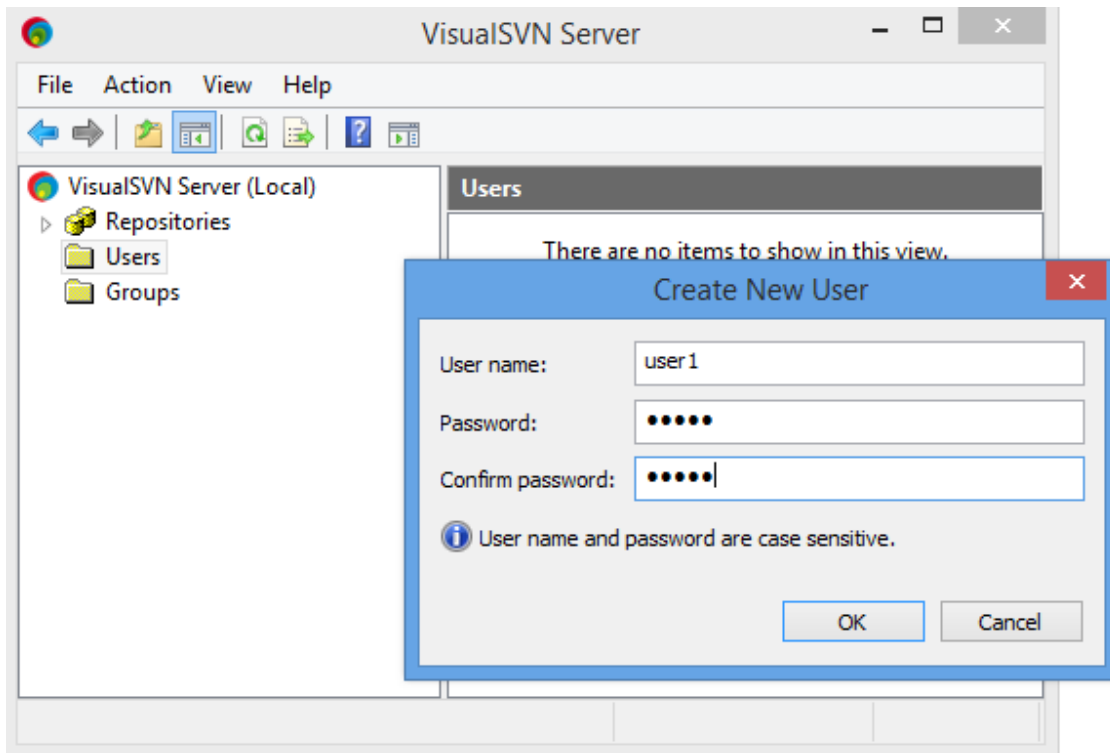
例 1: 複数の作成者のシナリオでのVisualSVNの使用

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する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。

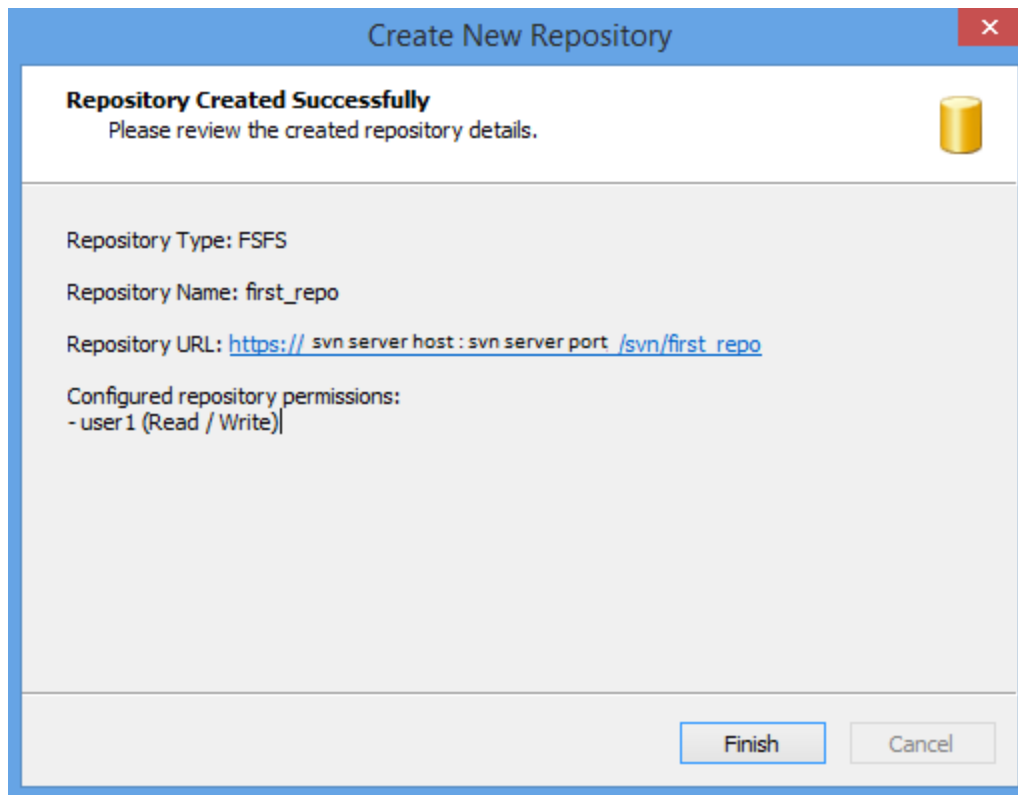
<https://subversion.apache.org/docs/>.

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。

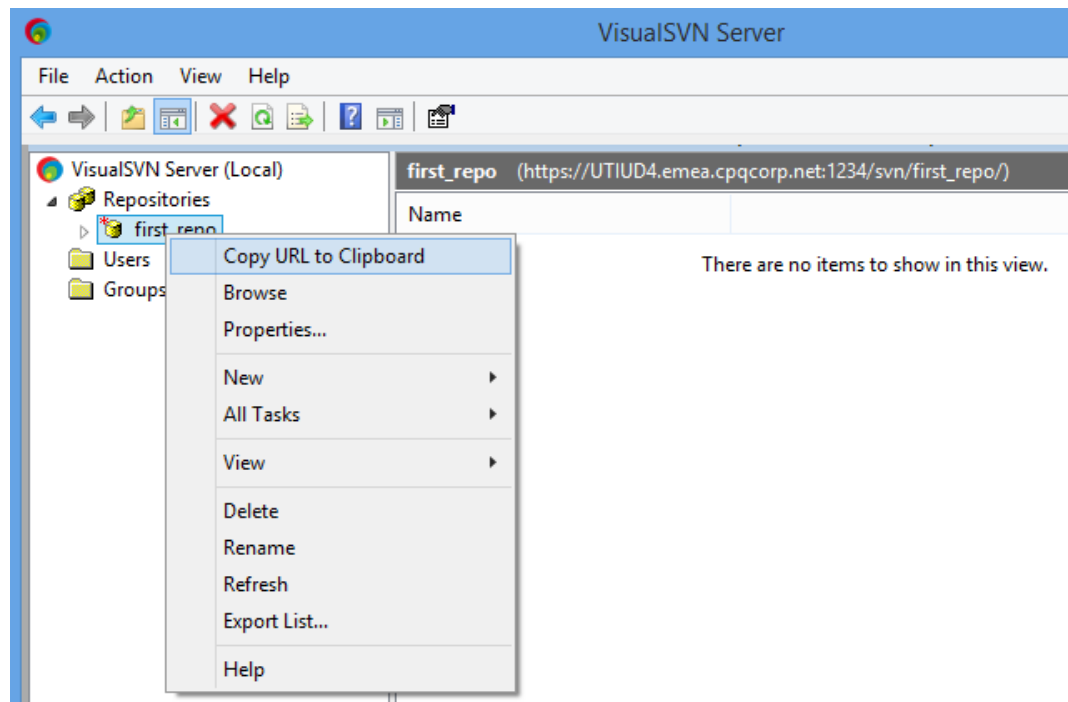


4. Studioで [SCM] メニューから [設定] を選択します。
5. [Subversion] ラジオボタンを選択します。
6. SVNユーザー名 資格情報 (ユーザー名とパスワード)を入力します。

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。

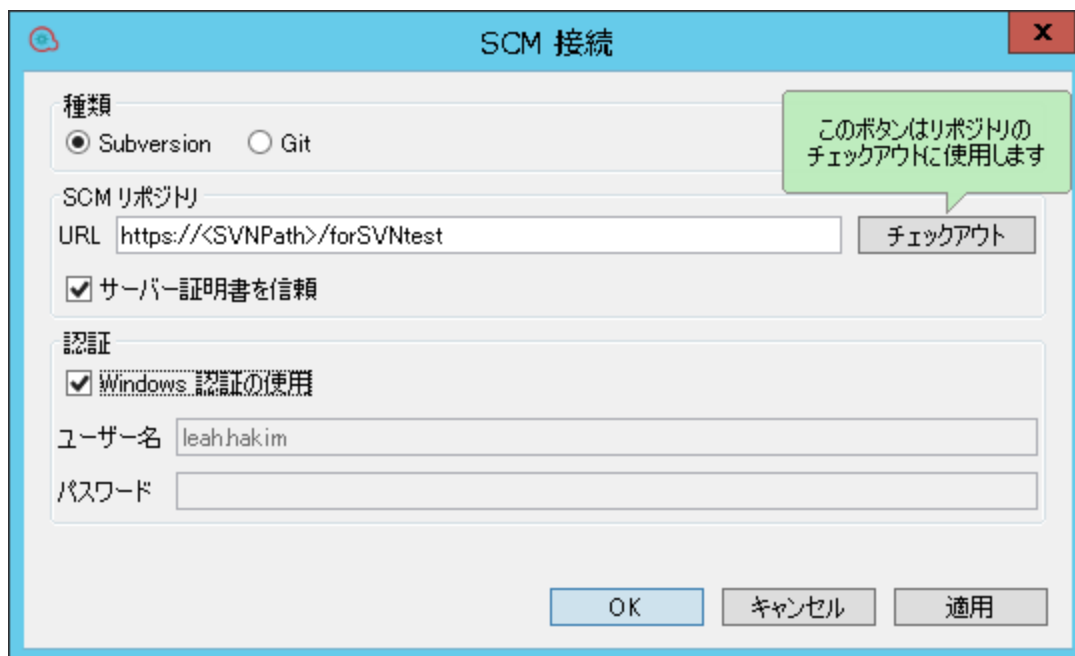
VisualSVNコンソールアプリケーションに直接アクセスできる場合は、次のようにしてリポジトリのURLを取得できます。

- 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、次の手順を実行します。

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**およびローカル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\Tools> manageScmRepository.bat  
-c LOCAL_FOLDER_FULL_PATH
```

次のように表示されます。

```
Repository was successfully created at <LOCAL_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。

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**」(51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いければ、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。

2. [種類] で、[Subversion] ラジオボタンを選択します。
3. 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、HPE OOは共有可能なネットワーク接続をオペレーティングシステムが生成するとみなすので、HPE OO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。

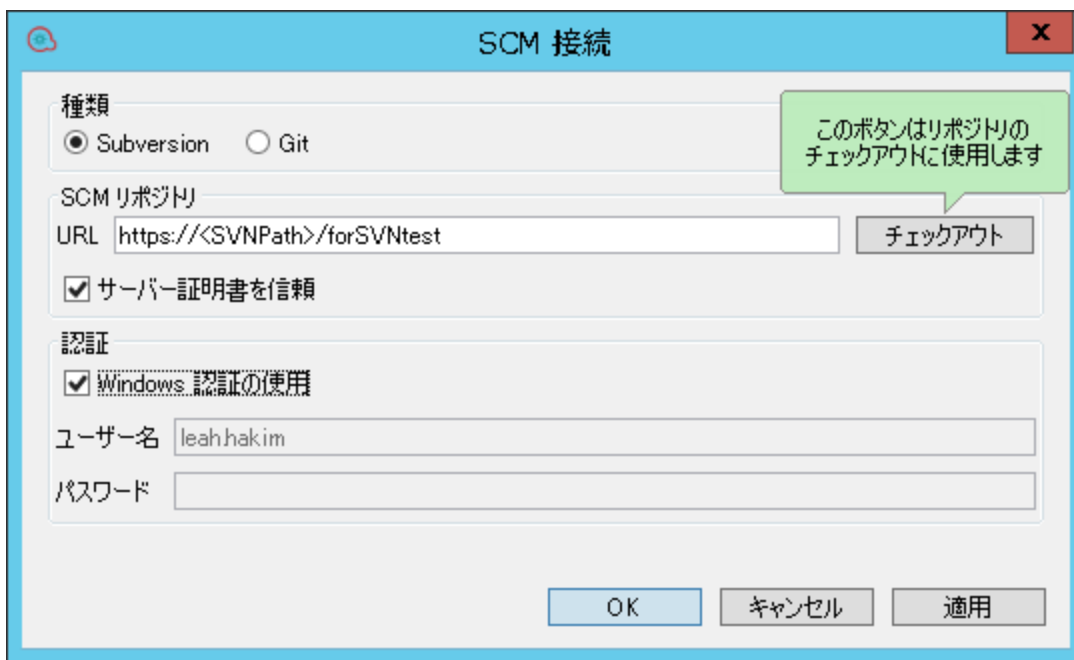
リポジトリをチェックアウトする

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。

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

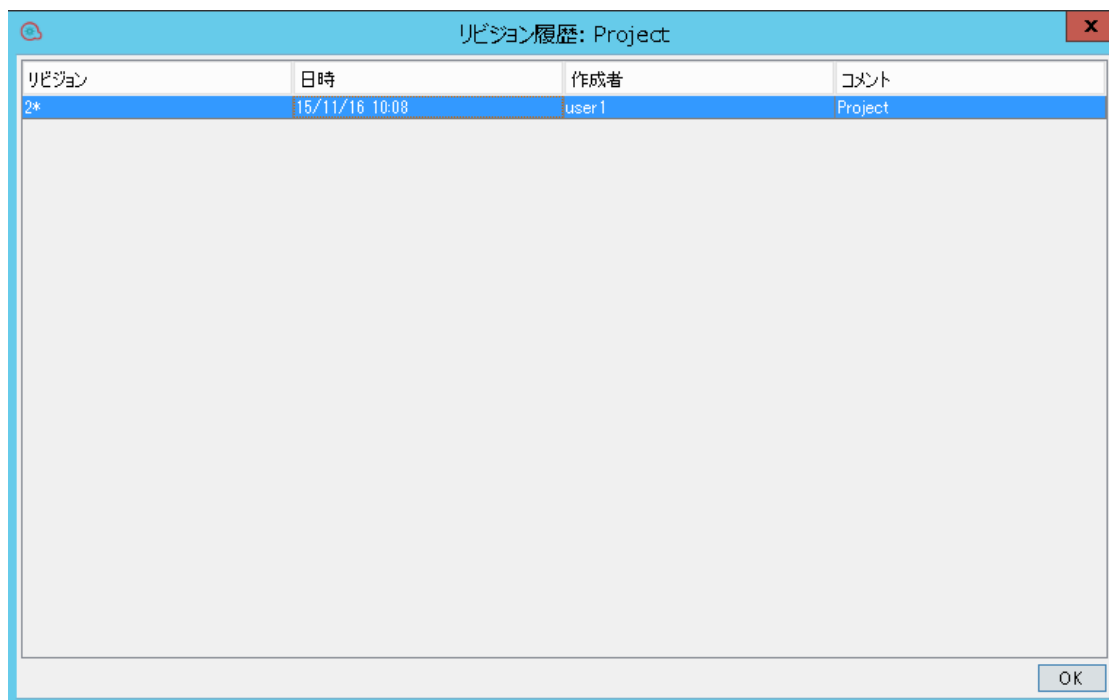
[履歴] オプションには次の情報が表示されます。

- 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/アイテムの特定のリビジョンに戻すこともできます。詳細については、「[以前のリビジョンのプロジェクト/アイテムに戻す](#)」(56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。



以前のリビ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. **[このバージョンに戻す]**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

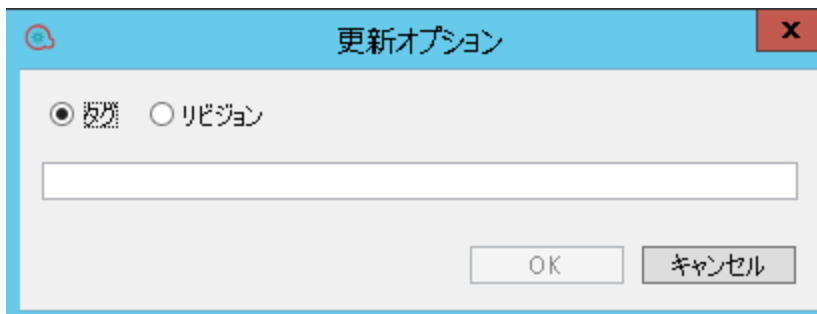
コンテンツパック内の特定のフローのトラブルシューティングのために、問題が発生したCentralコンテンツパック/プロジェクトの再作成が必要になることがあります。コンテンツパックが使用できない場合は(失われたか新しいバージョンで上書きされた場合)、**[更新結果...]** オペレーションを使用して特定のバージョンのプロジェクト/アイテムを復元できます。

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せん。

[更新結果...] オプションは、リポジトリをすで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。

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、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。

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、次の手順を実行します。

- [**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

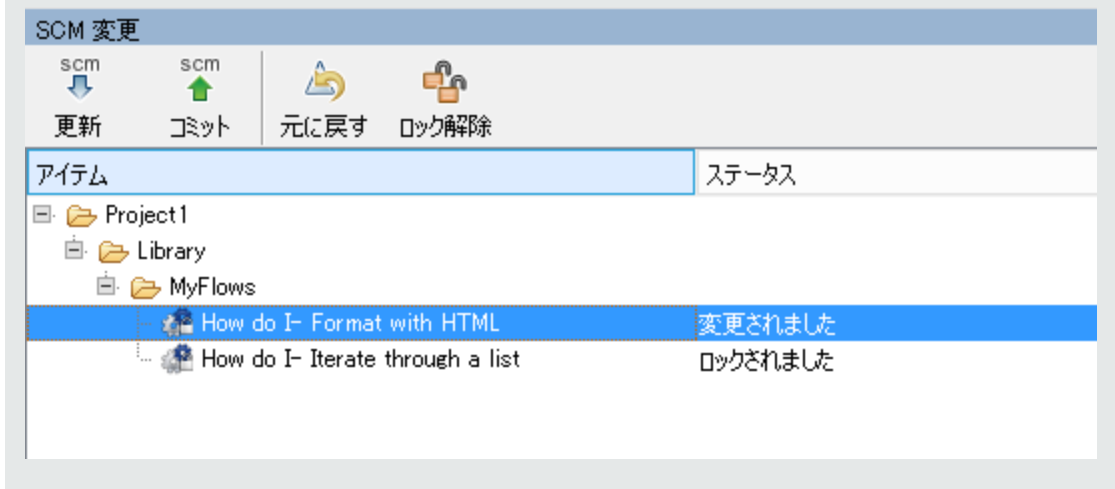
- アイテムの編集ウィンドウの左上隅で開いた鍵のアイコンをクリックします。



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。

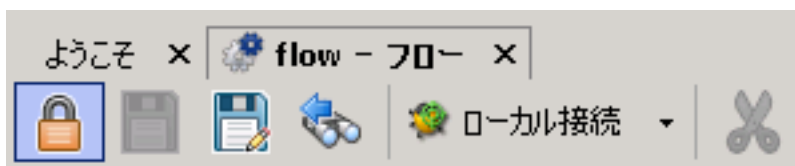


注: アイテムをロックした後で、[SCM変更] ペインに表示されるアイテムのステータスが「ロック済み」になります。アイテムがロックされ、変更されている場合 (推奨されません)、ステータスは「変更済み」と表示されます。例:

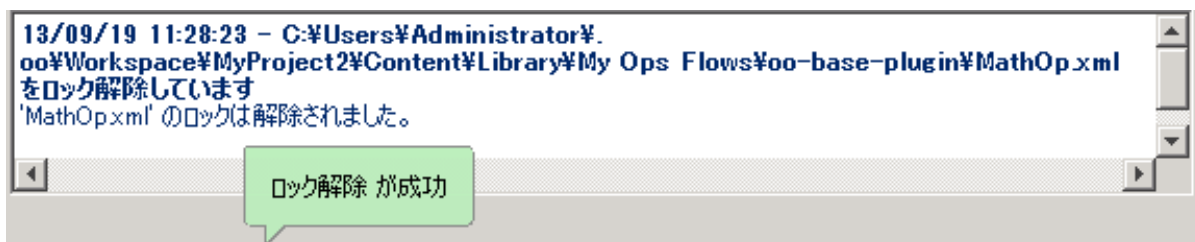


アイテムをロック解除する

- 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。

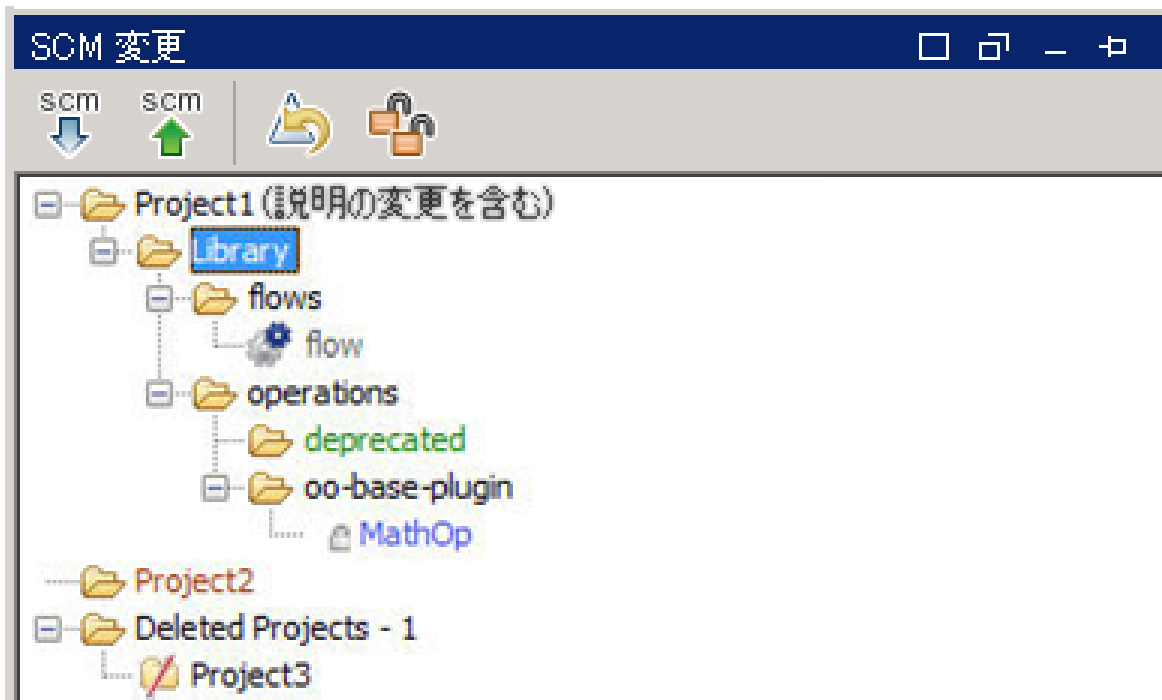
参考資料[SCM変更] ペイン

[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
```

つまり、バージョン管理されていないプロジェクトやフォルダーがこれらのパターンのいずれかに一致する場合、(バージョン管理されている場合のように)[SCM変更] ペインに黒色で表示されます。

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。

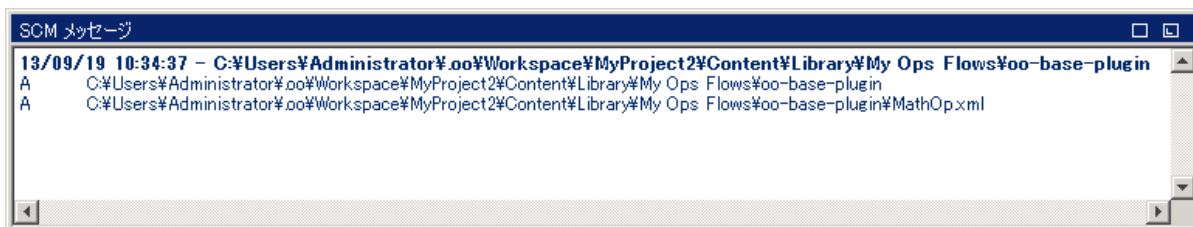
[SCMメッセージ] ペイン

[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。

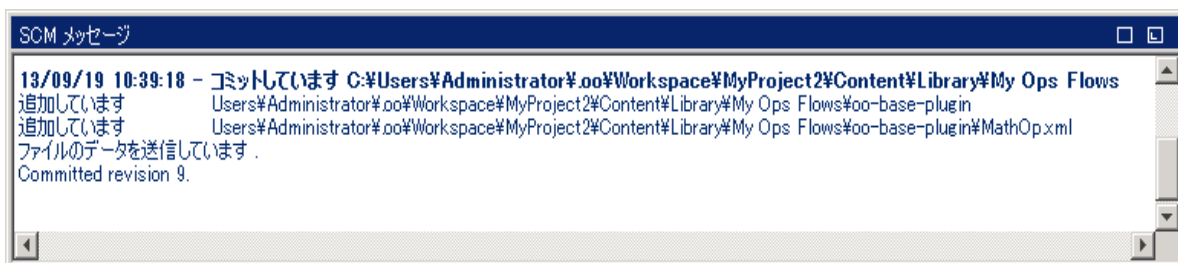
SCMメッセージの例

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。

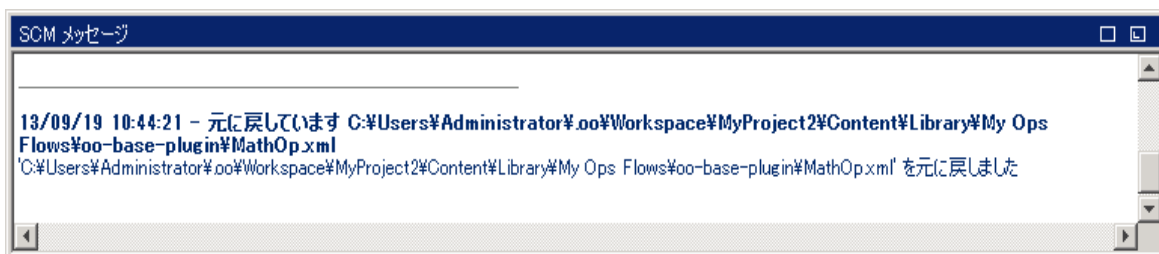
それぞれを追加した後、新しいアイテムが自動的に追加としてマークされます。次のSCMメッセージには、oo-base-pluginフォルダーとオペレーションのXMLファイルのMathOp.xmlが追加としてマークされています。



アイテムをコミットすると、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。

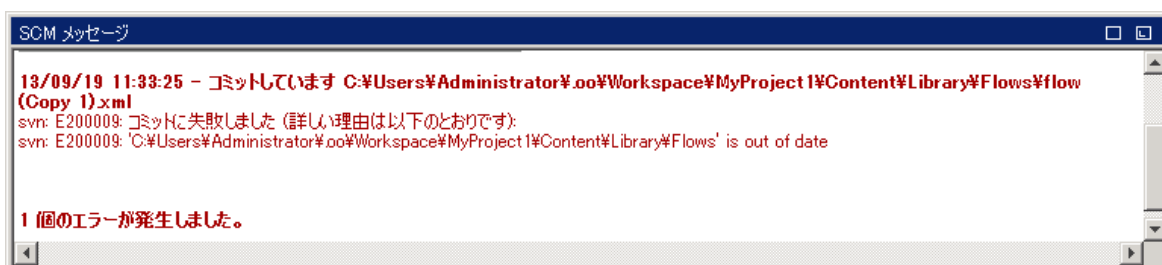


たとえば新しい入力の追加といった変更を**MathOp**に対して行った後、変更を元に戻します。このメッセージには、MathOp.xmlへの変更が元に戻されたことが示されます。



エラーメッセージ

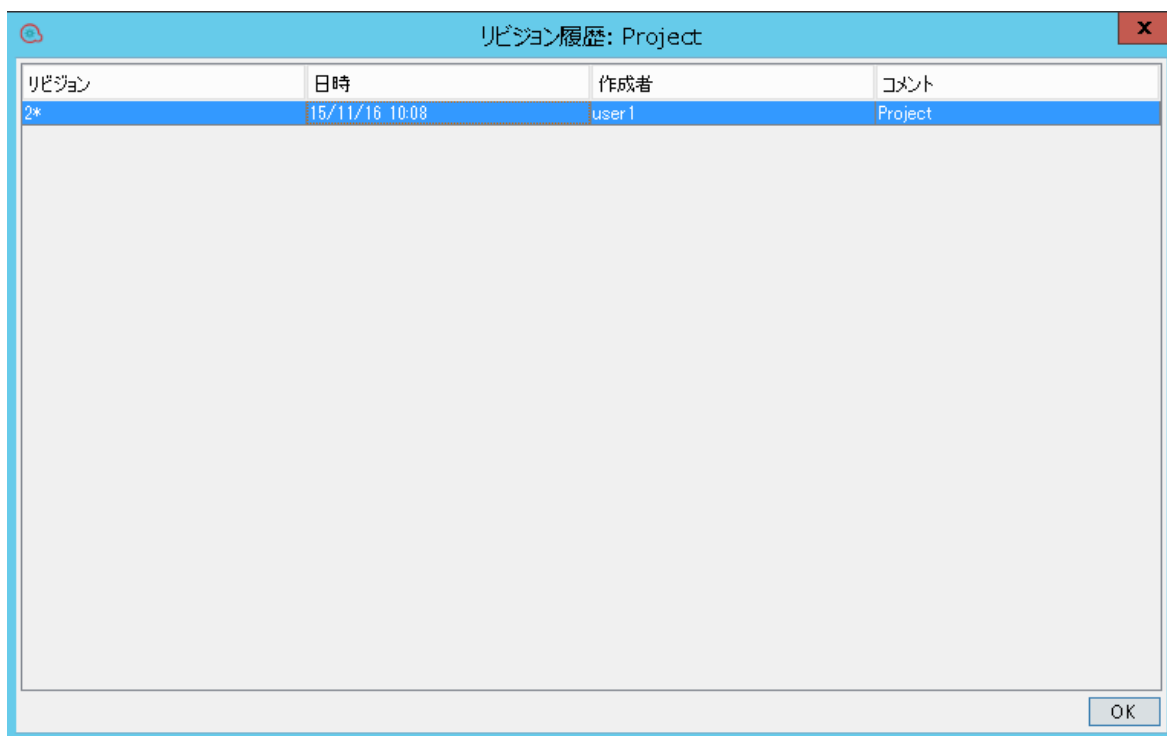
エラーが表示される場合もあります。エラーは赤で表示されます。この例では、作者が変更のコミットを試行しましたが、ファイルが期限切れになっています。



参考資料

リビジョン履歴

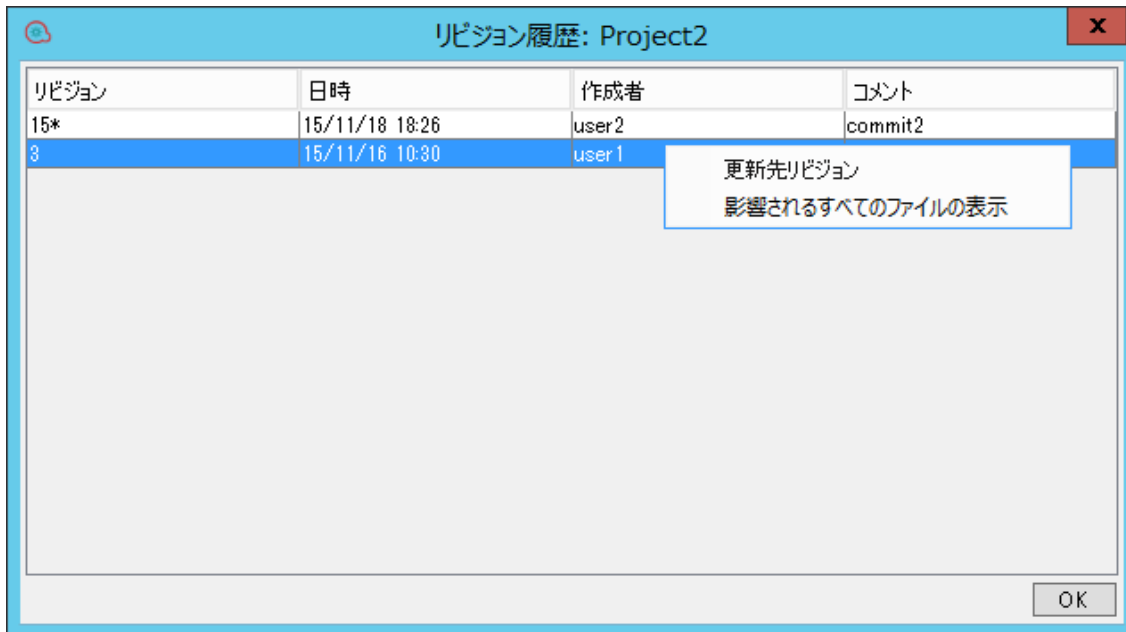
HPE OO Studioでは、バージョン履歴管理も提供されています。[リビジョン履歴] ペインには、SCM履歴が表示されます。4つの列に分れ、各コミットが1行ずつ表示されます。



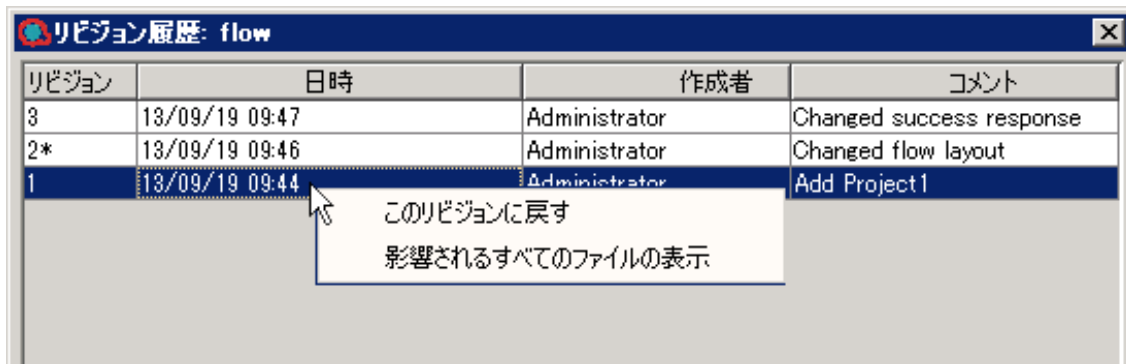
- **リビジョン**: コミットのリビジョン。ローカルコピーのリビジョンにはアスタリスクが表示されます。チェックアウト後、アスタリスクは常に最新のリビジョンを示しますが、別のユーザーが変更をコミットした後かつ更新前の場合は、アスタリスクが示すリビジョンは最新のものではありません。上記のスクリーンショットは、ローカルコピーがリビジョン1953であるが、リビジョン402でファイルへのコミットが行われたことを示します。
- **日時**: リビジョンがコミットされた日付と時刻。
- **作成者**: リビジョンをコミットした作成者。ユーザー名が表示されます。
- **コメント**: コミット時にユ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

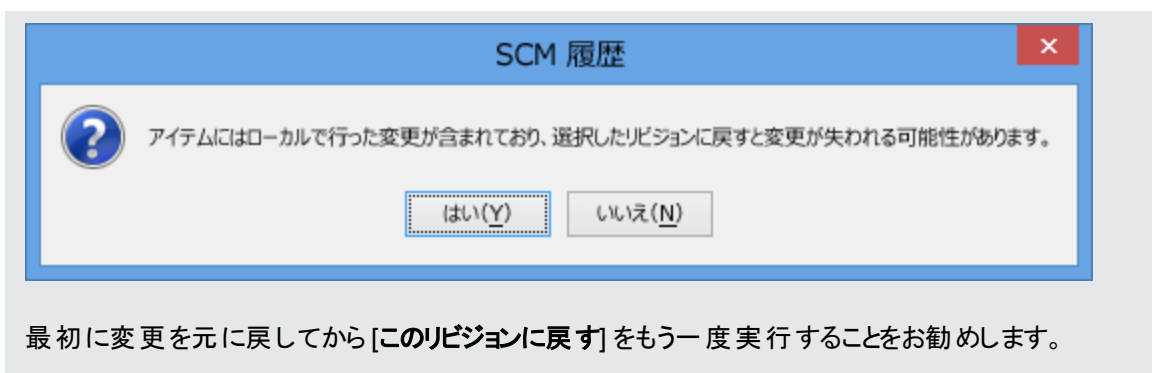


- このリビジョンに戻す:** 選択したリビジョンと同じ方法でローカルでオブジェクトを変更します。たとえば、リビジョン333で2つのステップを含み、リビジョン337で3つのステップを含むフローがある場合、これをリビジョン333に戻すと、含まれるステップは再度2つになります。ほかの作成者が変更を見ることができるようにするには、フローをコミットする必要があります。以下の例で、アスタリスク(*)の下にあるリビジョンを右クリックします。



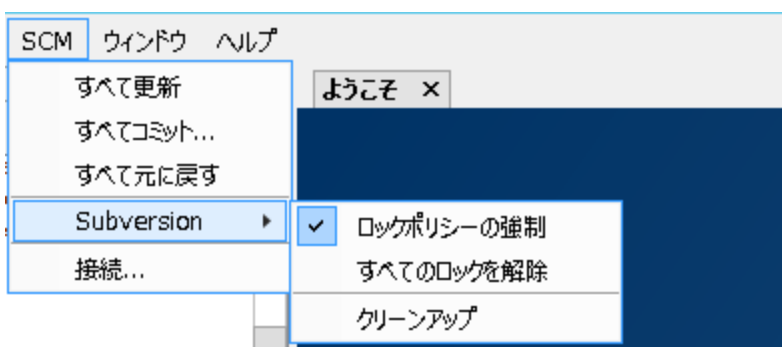
- 影響されるすべてのファイルの表示:** 選択したリビジョンで変更されたファイルをすべて表示します。

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。

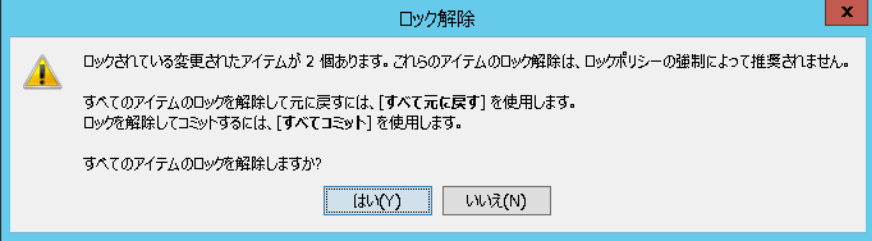


[ソースコントロール] メニュー

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	ワークスペースのディレクトリを更新しますが、Studio1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Studio1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。
[Subversion] > [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。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。

オプション	説明
[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>  <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]	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。
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>

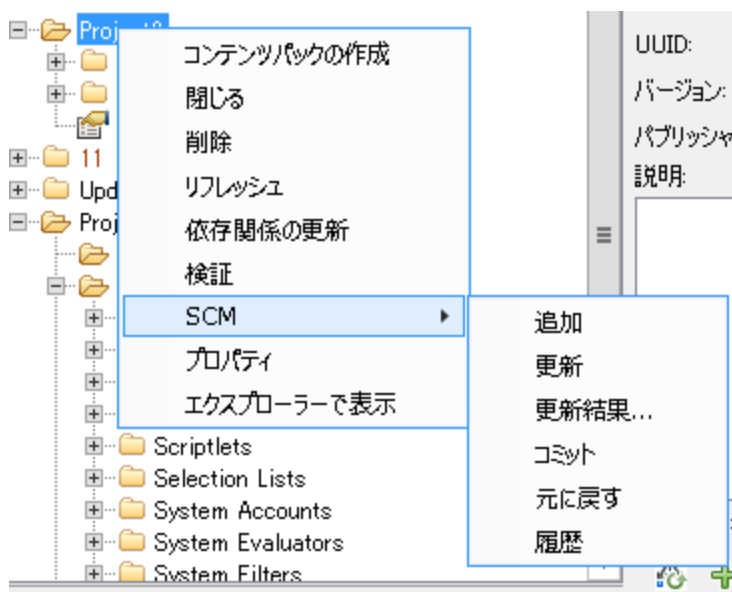
[プロジェクト] ペイン

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HPE 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

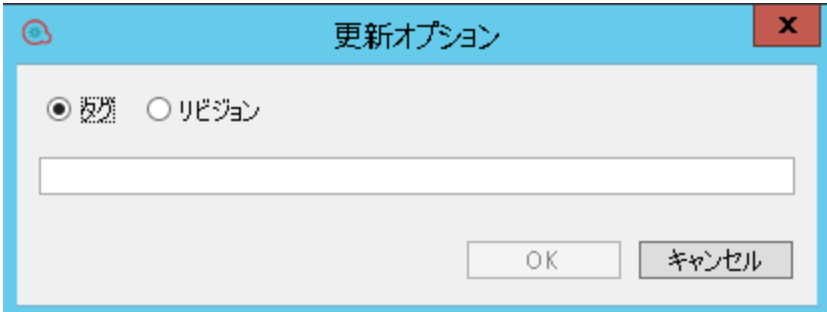
コンテキストメニューには、選択したオブジェクトで使用可能なアクションがその状態に応じてすべて表示されます。

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。
更新結果...	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。

オプション	説明
	 <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、一部の概念は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。

10.5x以降の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の要件というよりは、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クライアントのインストールと構成

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、次の手順を実行して、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。パスの区切り記号として **/** 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。

注: この2番目の方法を選択した場合、次の点に注意する必要があります。

複数のワークスペースを使用している場合に、Gitの場所プロパティを新規ワークスペースのそれぞれに追加するには、**Studio\confstudio.properties.template**にあるテンプレートプロパティファイルを編集する必要があります。そうしないと、新規ワークスペースに切り替えるたびに、**.oolStudio.properties**ファイルで新規ワークスペースのGitの場所を設定する必要があります。

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。これはStudioで検証されているバージョンです。その他のバージョンも正常に動作する可能性はありますが、正式にはサポートされていません。

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ます。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/更新間の関係、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」\(75ページ\)](#)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。

注: すべてのワークスペースには、それぞれ固有のログファイルを備えた独自のlogsフォルダーが含まれます。

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。

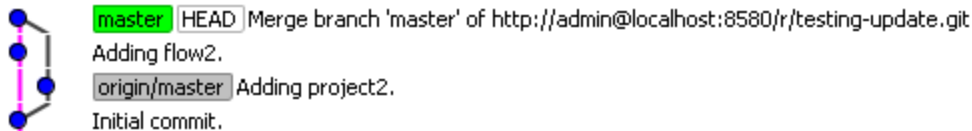
- **更新方式: マージ。** マージ更新方式を使用すると、Gitがローカルコミットの元の階層を保持し、2つの分岐した「ブランチ」(ローカルブランチとリモートブランチ)を再びマージする特別な「マージ」コミットを作成します。

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** 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](#)」(87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。リベースは対話型の操作です。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。

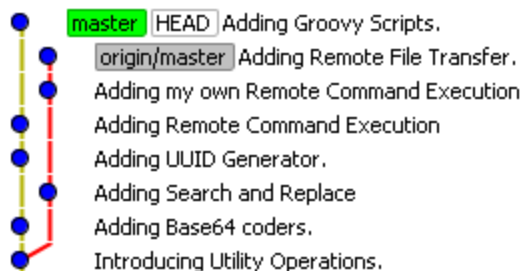
更新方式としてリベースを選択して「**Update All**」オペレーションを実行した後で、次のイベントが発生します。

- StudioI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。

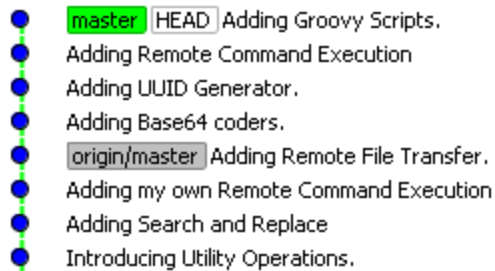
この段階で、リポジトリログは次のようになります。



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

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](#)」(85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](#)」(80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まった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](#)」(90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](#)」(87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](#)」(87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](#)」(82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](#)」(346ページ)を参照してください。

実行する操作

注: SCMオペレーションは、重複するUUIDを持つアイテムでは実行できません。

これを解決するには、次の手順を実行します。

1. **[問題]** ペインに問題のあるアイテムを表示します。
2. 重複するアイテムを削除します。
3. 元のアイテムをそのまま残します。

両方のアイテムを残す場合は、**[コピー/貼り付け]** または **[重複]** コマンドを使用して、異なるUUIDを持つアイテムのコピーを保存してから、コピー元のアイテムを削除します。

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リ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リ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を使用するときに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](#)」(347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]

ボタ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\\.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。

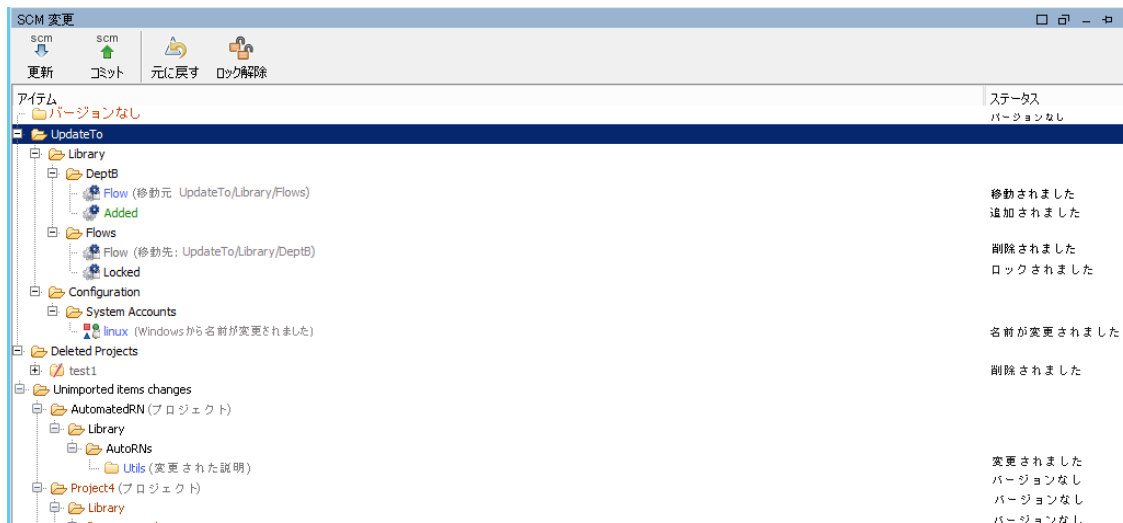
1. **[プロジェクト]** ペインで、移動するアイテムを選択します。**Shift**キーまたは**Ctrl**キーを使用して、複数のアイテムを選択することができます。
2. アイテムを別のフォルダーまたは別のプロジェクトにドラッグアンドドロップします。

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

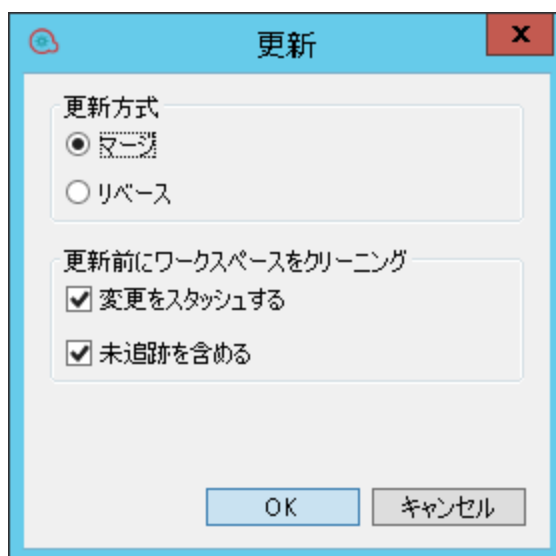
[SCM更新] オペレーションは、スタッシュ、プル、スタッシュ解除という3つのGitオペレーションを実行します。

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」(87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」(89ページ)および「手動による変更のスタッシュ解除」(90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。

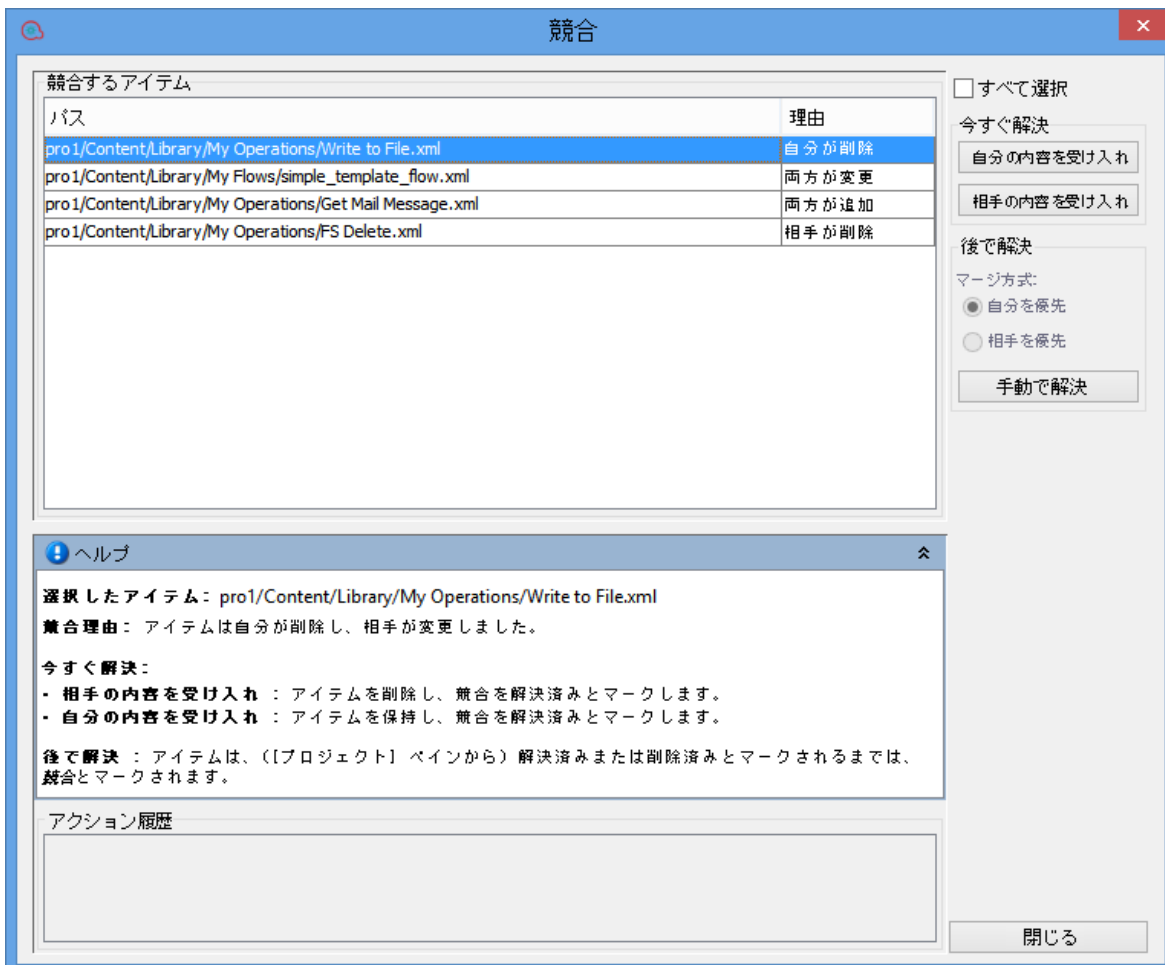
```
Cannot update the current branch as it does not exist in the remote repository.
```

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に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](#)」(89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)

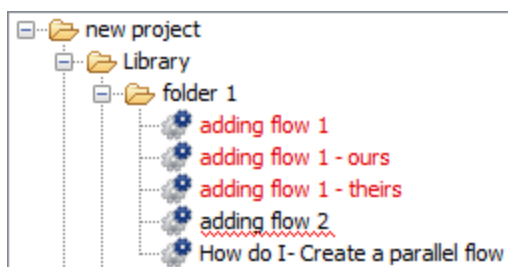
後で競合を解決することを選択した場合は、Studioで手動で競合を解決できるようにアイテムが準備されます。競合していない変更はアイテムに自動的にマージされます。競合している変更は選択した設定 (自分または相手) を基にして置き換えられます。

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、または削除/切り取りできません。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、次の手順を実行します。

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。

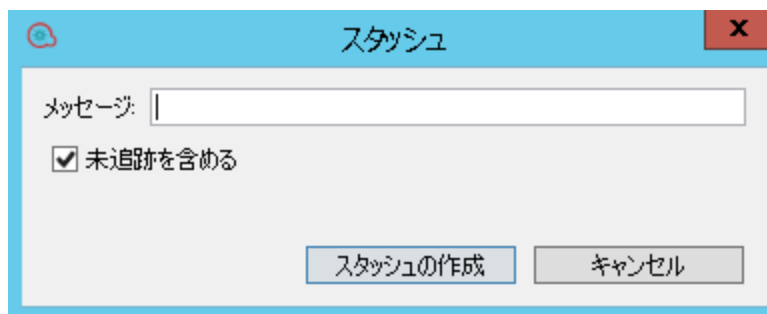
ローカルの変更をスタッシュするには、次の手順を実行します。

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、スタッシュのり

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。

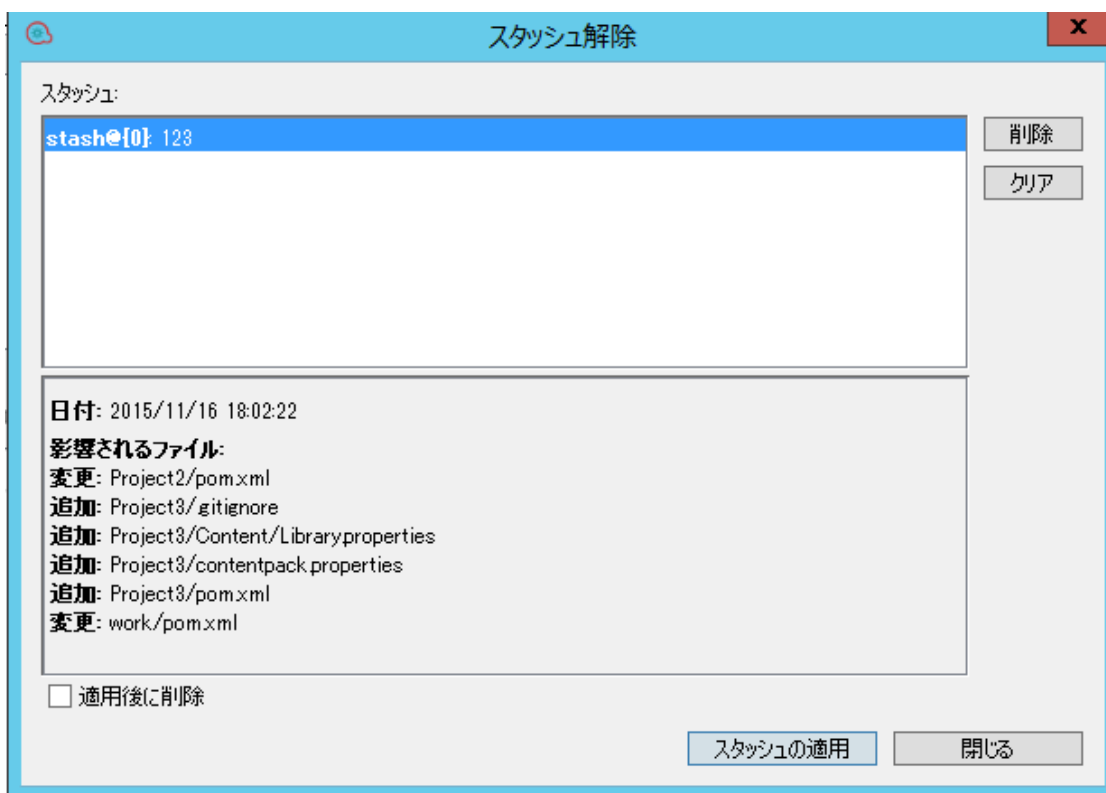
注: スタッシュ解除オペレーションの結果として競合が発生する可能性があるため、スタッシュ解除オペレーションを実行するときには注意してください。

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、次の手順を実行します。

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

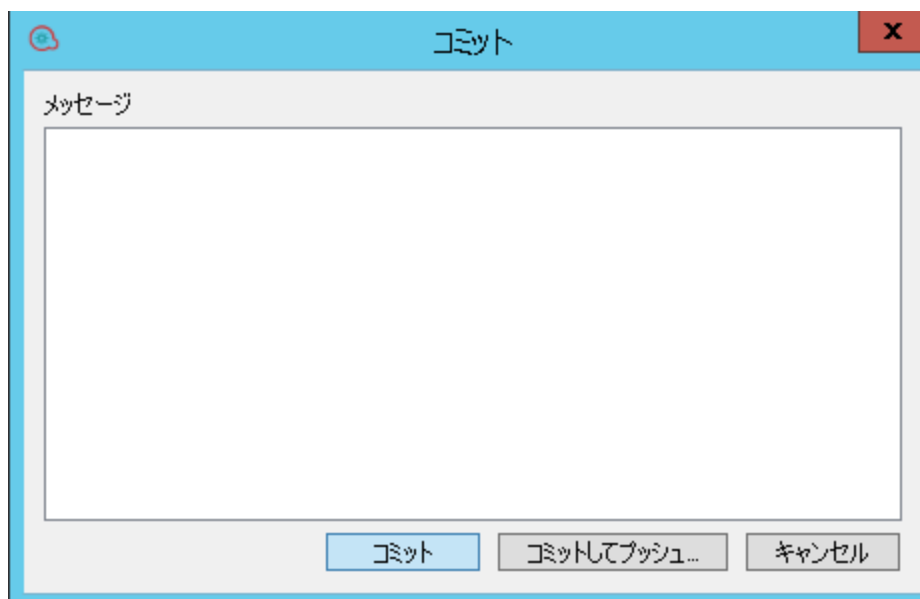
Studioで変更を行った後に変更を記録するには、ローカルGitリポジトリに対してそれらのファイルをコミットし、その後でそれらをGitサーバー上のリモートGitリポジトリにプッシュする必要があります。

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](#)」(93ページ)を参照してください。

注: コミットリスト内に重複するUUIDを持つアイテムがある場合、下線付きのテキストで表示され、次の警告メッセージが表示されます。

```
There is at least one other item with the same UUID as one of the items included in the commit.Do you want to continue?
```

重複するUUIDを無視してコミットを続行する場合は **[はい]** をクリックし、コミットオペレーションをキャンセルする場合は **[いいえ]** をクリックします。

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。

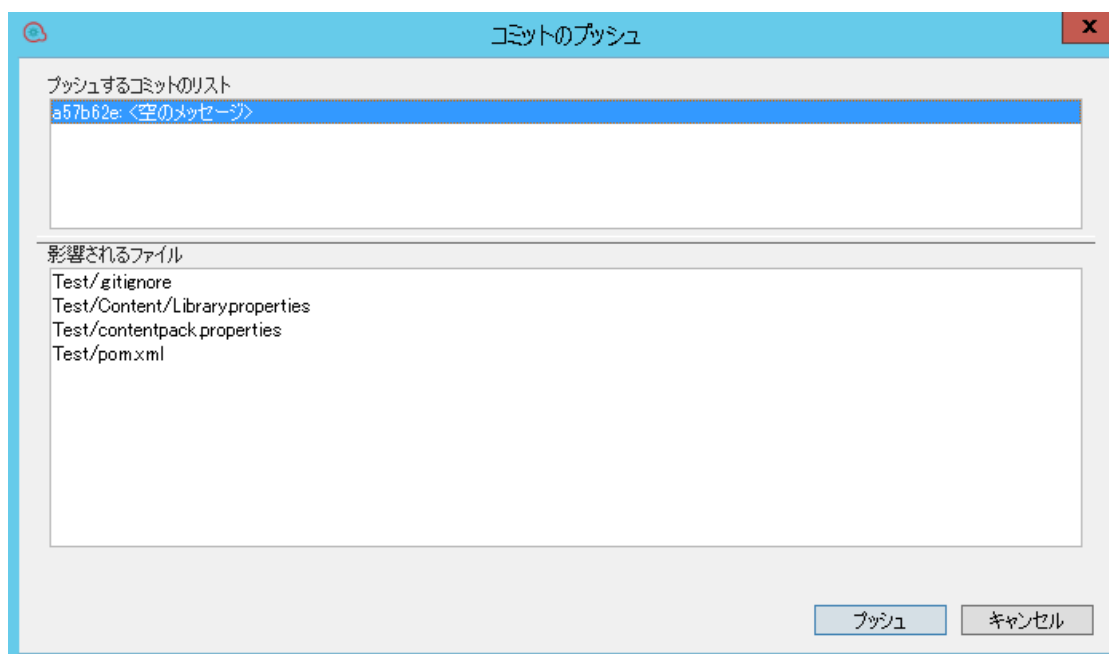
ベストプラクティス: 最新のファイルを確実に使用するには、コミットの前に、**[SCM] > [更新]** コマンドを使用してリモートGitリポジトリからファイルを更新する(プルする)ことをお勧めします。

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

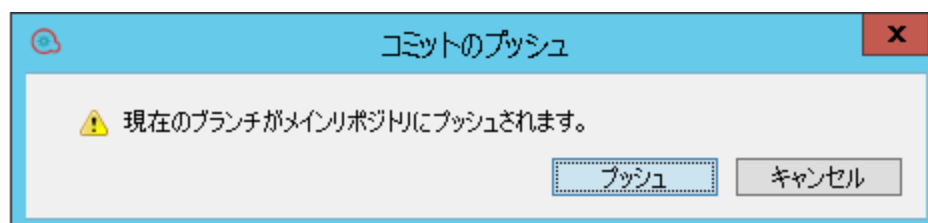
または、

[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

すると、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

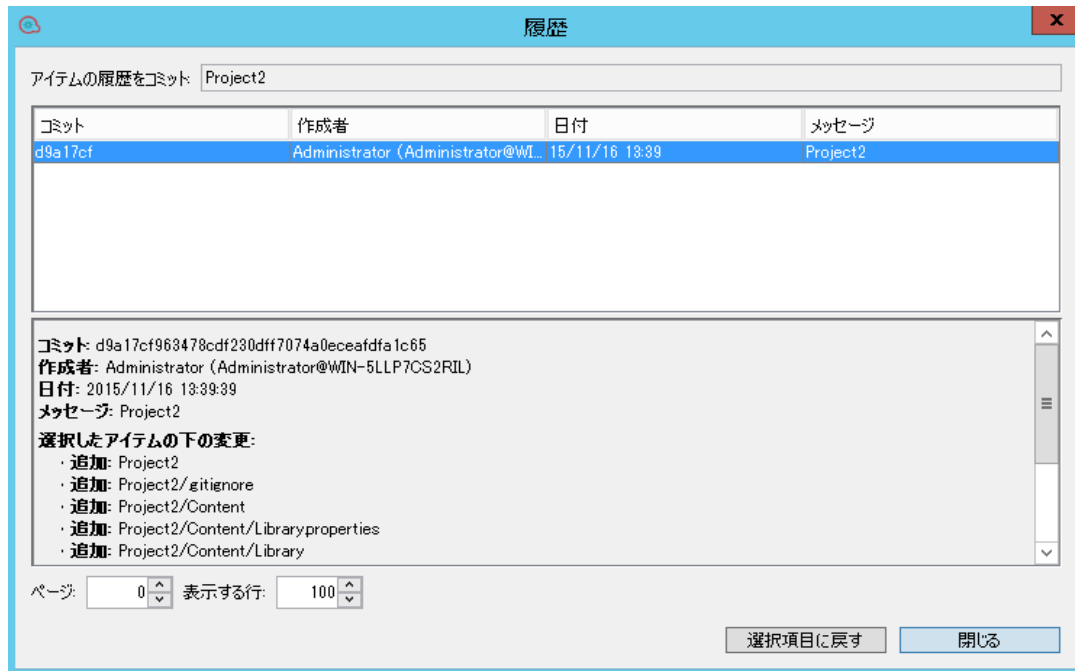
[履歴] オプションには次の情報が表示されます。

- 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/アイテムの特定のリリースに戻すこともできます。詳細については、「[以前のリリースのリポジトリに戻す](#)」(96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。



以前のリビジョンのリポジトリに戻す

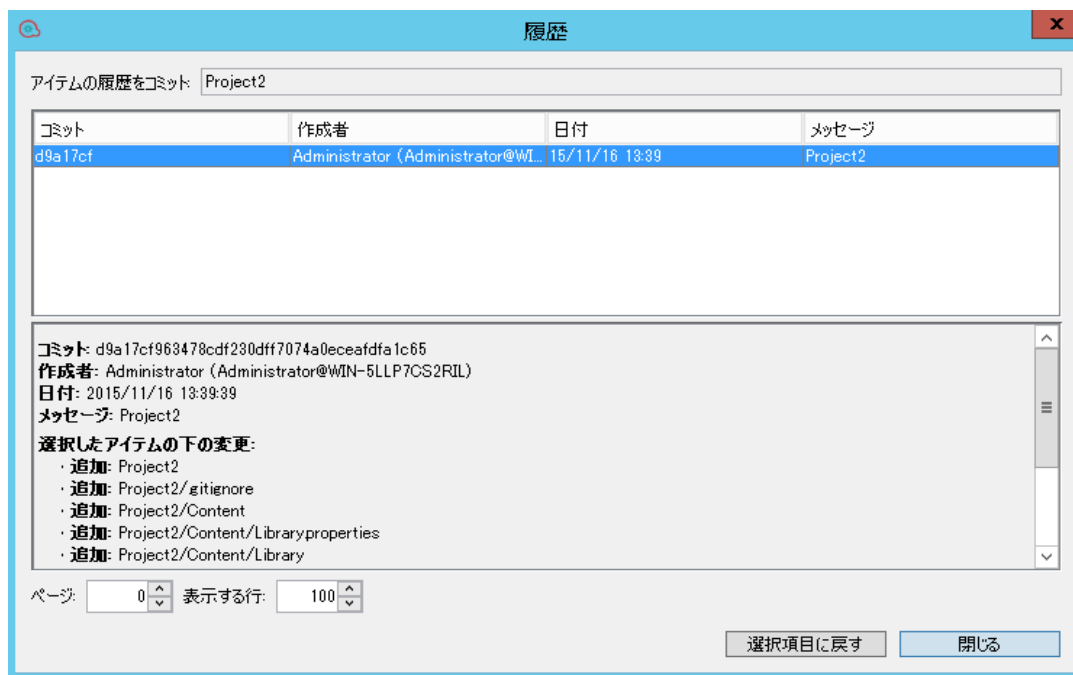
コンテンツパック内の特定のフローのトラブルシューティングのために、問題が発生したCentralコンテンツパックの再作成が必要になることがあります。コンテンツパックが使用できない場合は(失われたか新しいバージョンで上書きされた場合)、**[履歴]** オプションを使用して特定のバージョンのコンテンツパックを復元できません。

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バー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

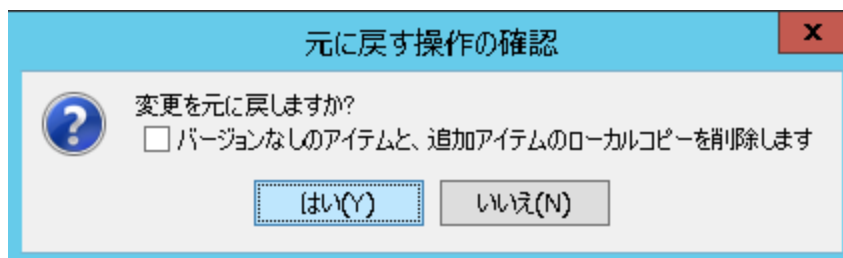
元に戻すと、作成者がアイテムに加えた変更が削除され (フォルダーの場合は、そこに含まれるすべてのアイテムが再帰的に削除されます)、元に戻されたアイテムのオリジナルの状態が、最後に更新されたときの状態に戻されます。

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、リポジトリをすで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と、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、次の手順を実行します。

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

- 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。

- 右クリックして、[SCM] > [追加] 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。

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す。

- 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でも、[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役立ちます。

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す。

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ず。

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す。

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す。

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:

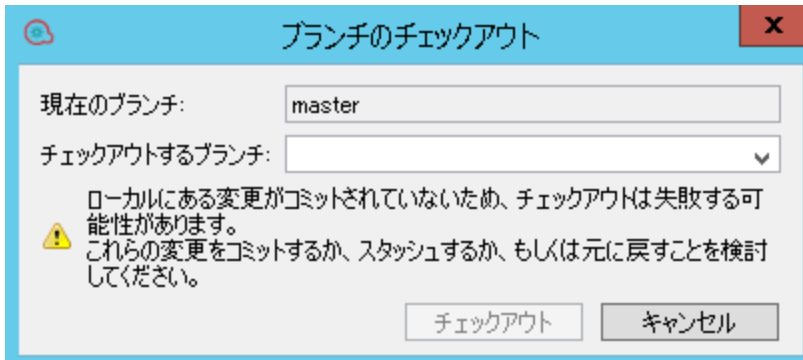
- 新しくチェックアウトしたブランチ内に存在しているプロジェクトでも、現在のブランチ内に存在していないプロジェクトはすべて削除されます。ただし、引き続きそのブランチを後でチェックアウトできるので、プロジェクトのデータは失われません。
- 両方のブランチ内に存在するプロジェクトは、新しくチェックアウトしたブランチの状態を反映して変更および更新されます。
- 新しくチェックアウトしたブランチ内にのみ存在するプロジェクトは、ディスクにコピーされますが、Studioで自動的にインポートされません。
- ローカルで変更が加えられているブランチのチェックアウトは推奨されません。

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] リストから、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

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。

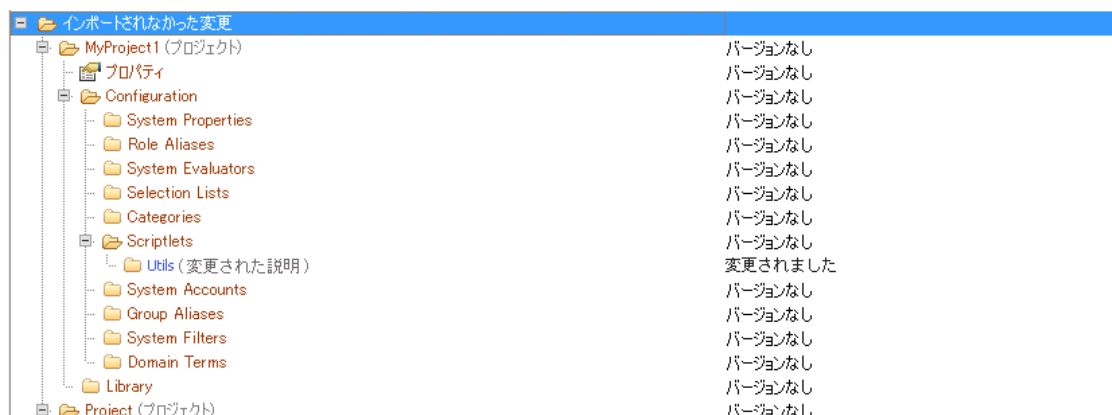
この情報は、競合がある場合、またはマージ/リベースを実行し、加えられたすべての変更を理解する必要がある場合に重要です。

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:** アイテムはすでに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:



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。それらを実行するには外部ツールが必要です。

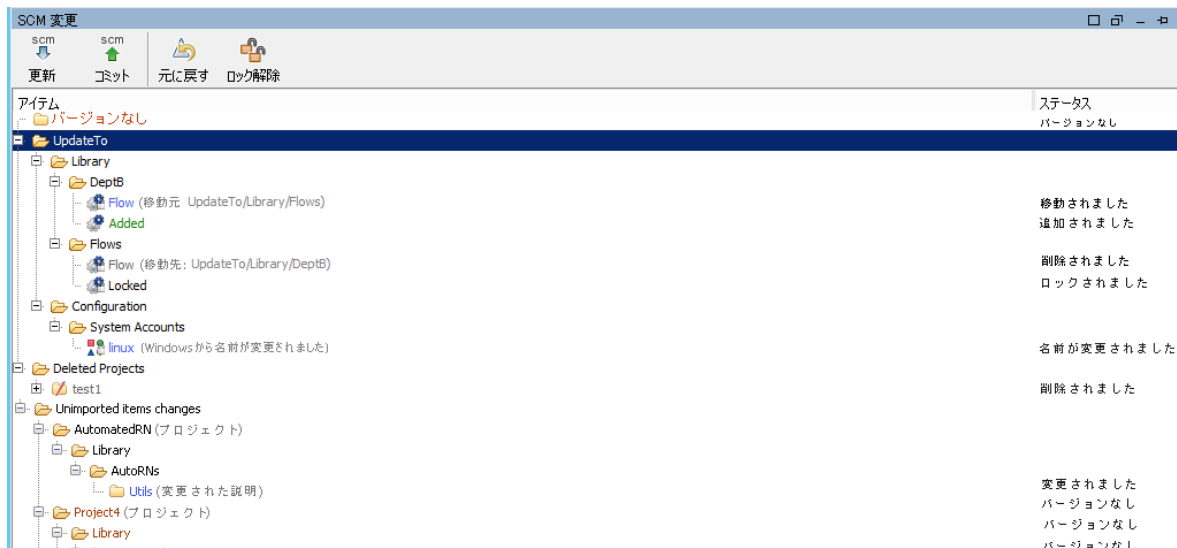
参考資料[SCM変更] ペイン

[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
追加されました	緑
名前が変更されました	青
変更されました	青
競合しました	赤
移動されました (移動元/移動先)	青
削除されました	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、「移動元 <古い場所>、移動先 <新しい場所>、元の名前 <古い名前>」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。

重複するUUIDを持つアイテムがある場合、[SCM変更] ペインに下線付きのテキストで示されます。

上記の例では、次の情報を表示できます。

- **test1**という名前の削除されたプロジェクト (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 data-bbox="423 1472 748 1556" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> 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 data-bbox="423 1671 919 1745" style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> 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>

<p>状態:  マージ中 (master)</p> <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> <p>状態:  マージ中 - すべての競合が解決済み (master)</p> <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>
--

[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アクション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。

SCM メッセージ
<p>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</p>
<p>15/11/19 03:25:41 - スタッシュ 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。 スタッシュされたパス: 変更: work/pom.xml</p>
<p>15/11/19 03:25:19 - 追加 C:\Users\Administrator\oo\Workspace\Test</p>

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ます。

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).

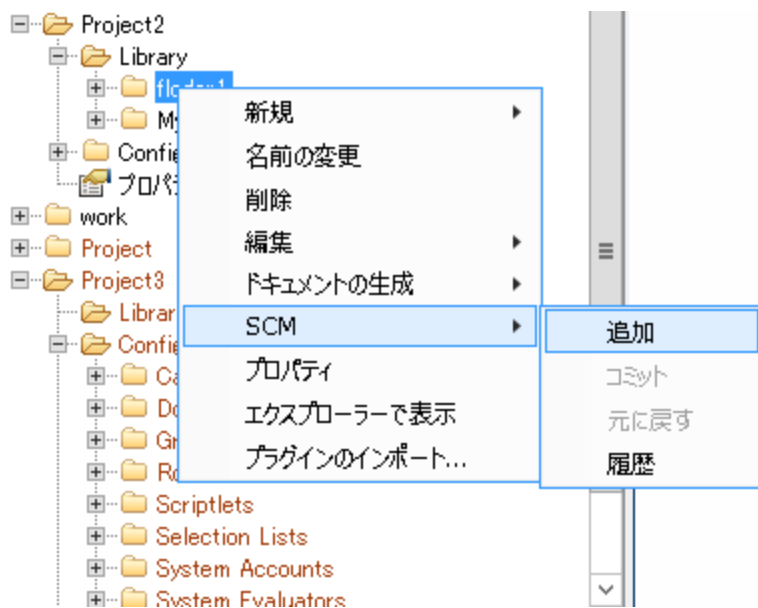
[プロジェクト] ペイン

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HPE OOオブジェクトが表示されます。

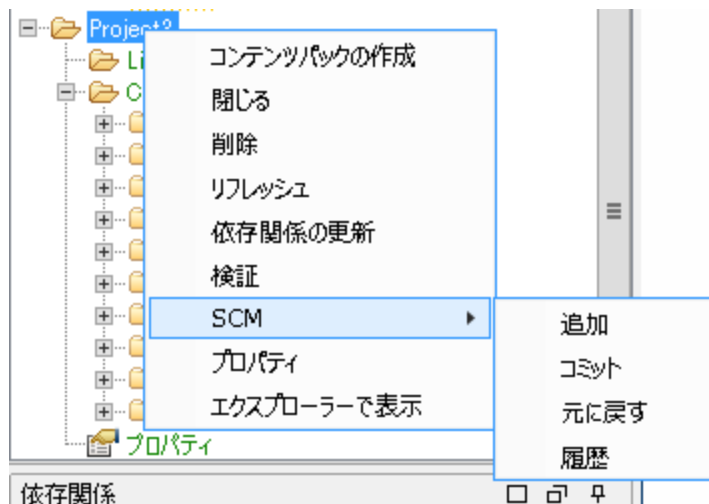
[プロジェクト] ペインのコンテキストメニュー

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。

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。



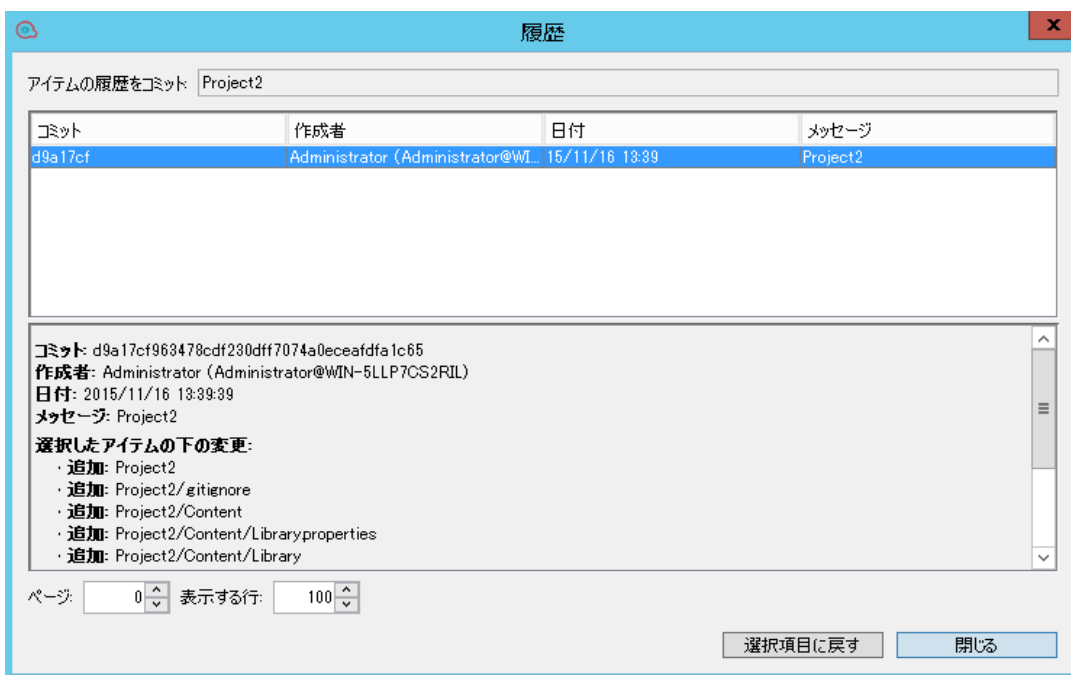
作成者がオブジェクトを変更する場合 (ステップを追加するなど)、ローカルで変更されます。この場合、[コミット] および[元に戻す]が使用可能です。



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	<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ず。</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	[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 」(95ページ)を参照してください。

[履歴] ダイアログボックス

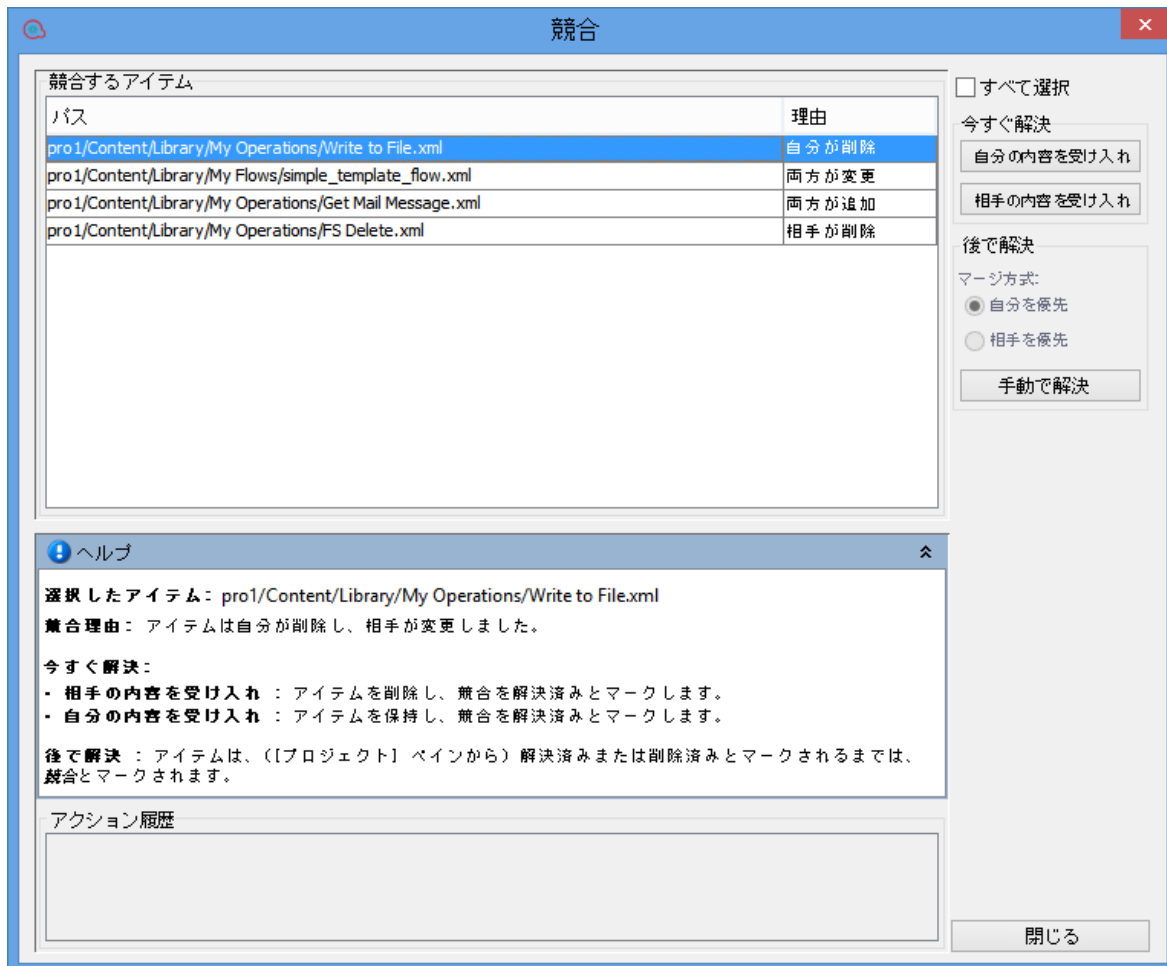
[履歴] ダイアログボックスを使用して、プロジェクト/アイテムの以前のリビジョンに戻します。



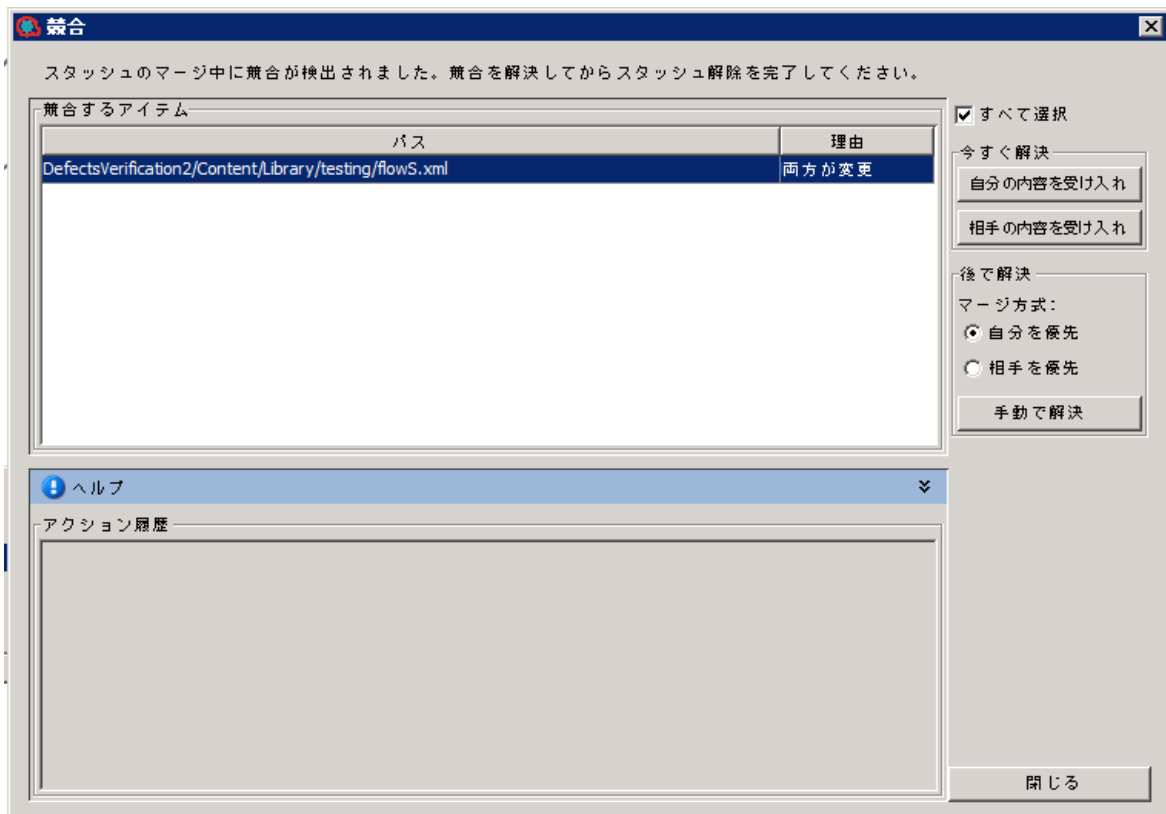
オプション	説明
アイテムの履歴をコミット	プロジェクト/アイテムの名前。
コミット	コミットオペレーションのID。
作成者	変更をコミットしたユーザーのユーザー名/ID。
日付	コミットオペレーションが実行された日付。
メッセージ	コミットオペレーションが実行されたときに書き込まれたコミットメッセージ。
選択したアイテムの下の変更:	このコミットで実行された変更および変更されたアイテムのリストを。
ページ	リビジョン履歴が1つのページに表示できないほど長い場合は、ページの数および各ページに表示される行数を変更できます。
表示する行	各ページに表示する行数を入力します。
選択項目に戻す	[選択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

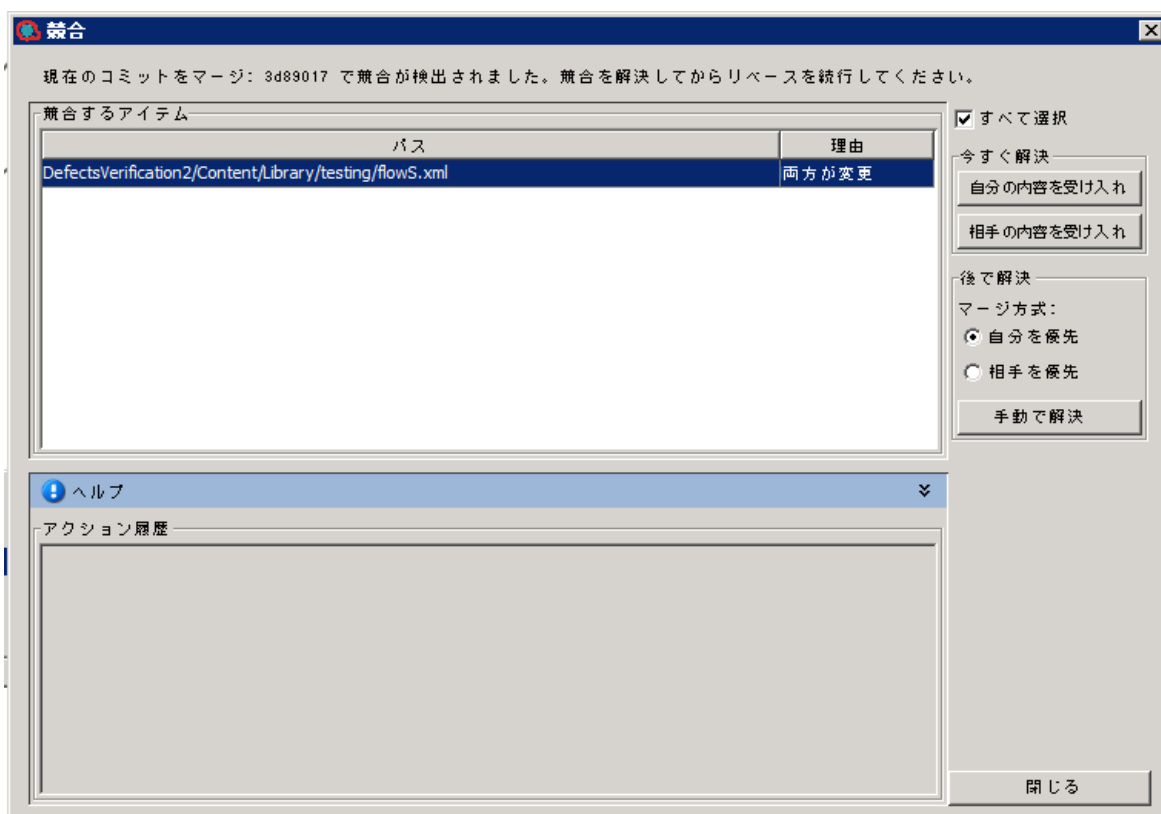
マージの例



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。
<u>今すぐ解決:</u>	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。
<u>後で解決:</u>	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。
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

オプション	説明
	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、次の手順を実行します。

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せん。

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す。

注: ネットワークドライブからコンテンツパックをインポートできます。

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

フローは別のプロジェクトから再利用することができます。そのフローで作成したコンテンツパックをインポートします。再利用するフローのコンテンツパックへのパッケージングについては、「[コンテンツパックのエクスポート](#)」(352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。
- **HPE content** - HPE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す。

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

1. ファイルシステム (Windowsエクスプローラーなど) で、コンテンツパックを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、unzip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。

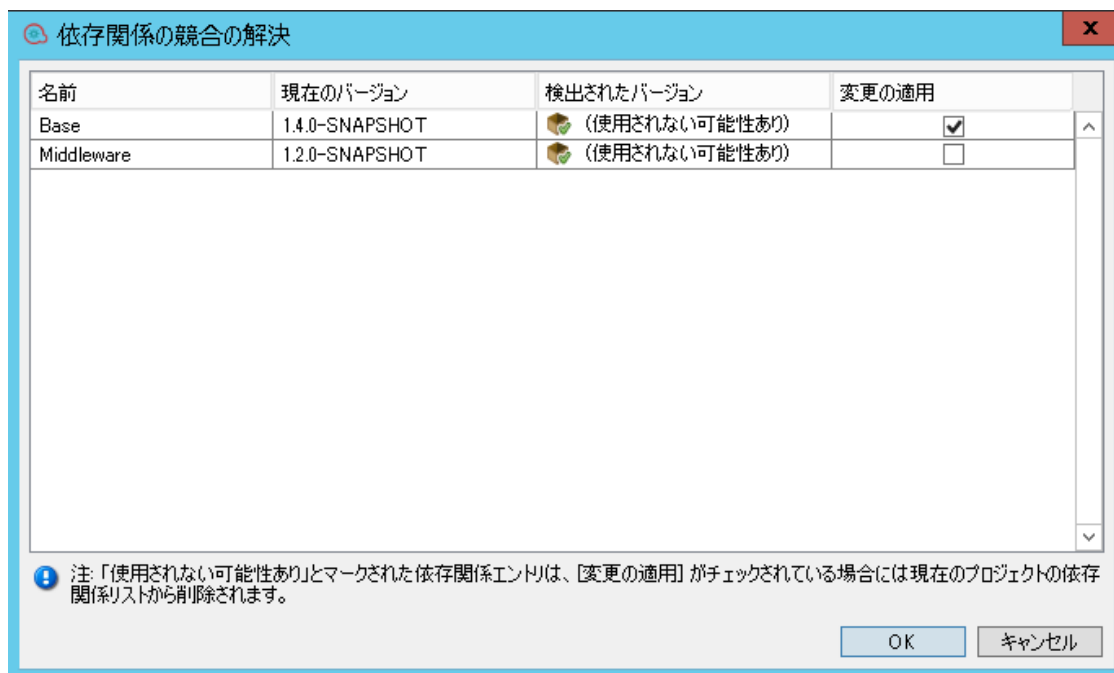
または、

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す。依存関係がすで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、Studio1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Studio1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。



5. 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。

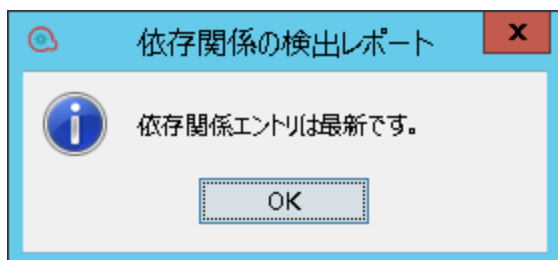
注: 依存関係が「使用されない可能性あり」としてマークされている場合、その依存関係がプロジェクト内で宣言されていても、Studioでその依存関係が使用されているかどうか識別できないことを意味します。たとえば、別のプロジェクト内のシステムプロパティを動的に参照している入力がフロー内に存在する場合があります。この場合、Studioはそのシステムプロパティがフロー内で使用されているかどうかを識別できません。

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。

6. [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、次の手順を実行します。

- [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
通知	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 [14.0-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 'Middleware [12.0-SNAPSHOT] 依存...	/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 [14.0-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/...

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。

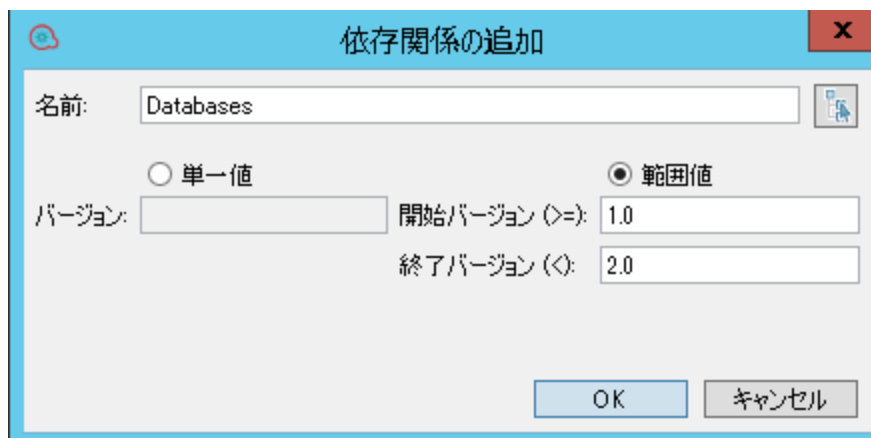
プロジェクトの新しい依存関係の追加


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

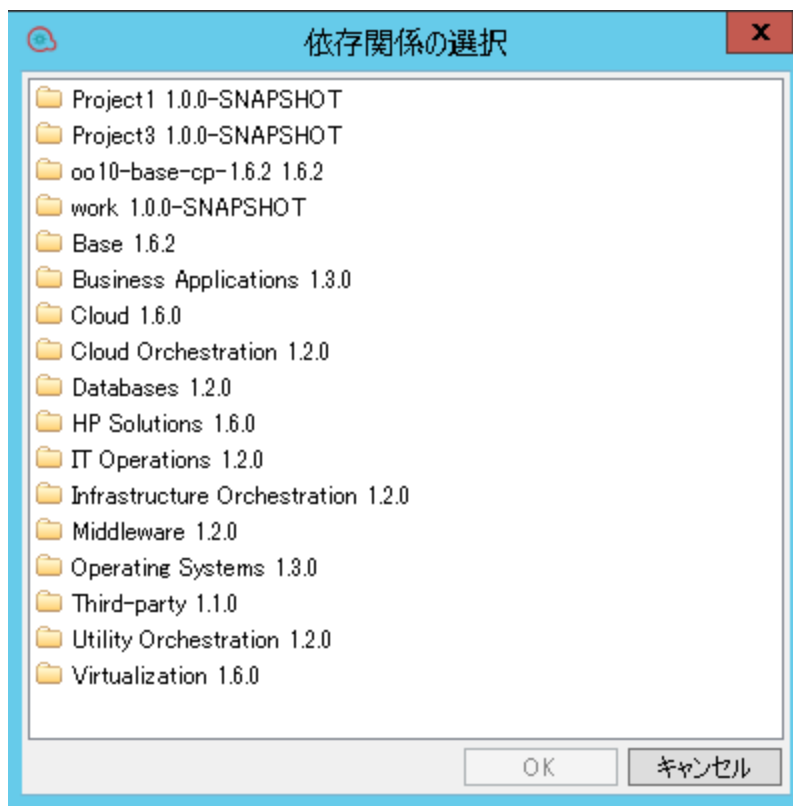
クスペースに存在していない場合も可能です)。


1. 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。



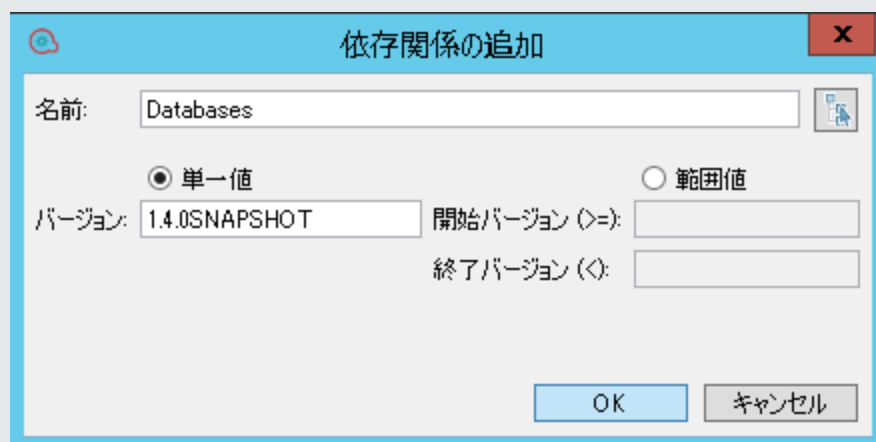
2. [名前] フィールドに、追加する依存関係の名前を入力するか、 をクリックして、現在のプロジェクトおよびStudioにインポートされているコンテンツパックのリストから依存関係を選択します。



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 Formatter
/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

3. アイテムをダブルクリックしてStudioにロードします。

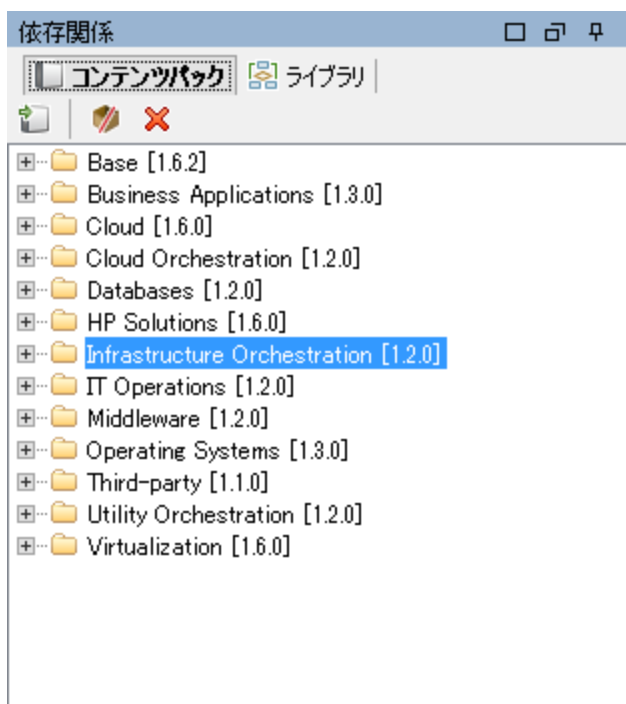
参考資料





[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HPE OOプロジェクトから作成した場合は、[コンテンツパックを作成] 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:</p>

	<p>-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。

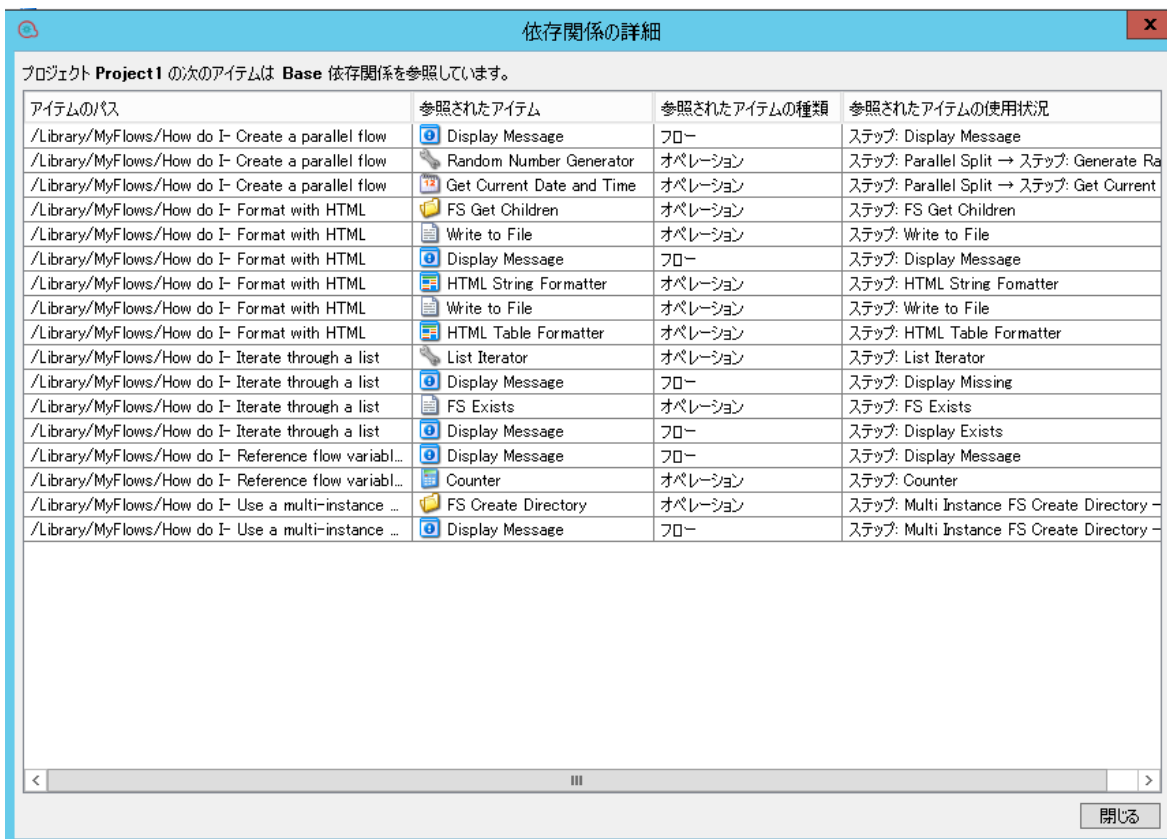
- **ワークスペースに含まれていない:** プロジェクトが、Studioワークスペースに現在インポートされていないプロジェクトまたはコンテンツパックにリンクします。

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
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 [14.0-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 'Middleware [12.0-SNAPSHOT] 依存...	/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 [14.0-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 Conma...	/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/...

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。

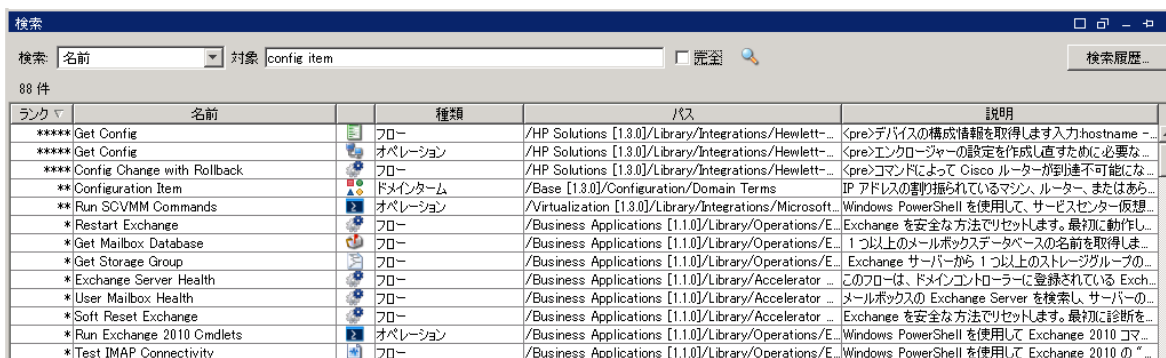
重要: プロジェクト内の構成アイテムを削除、作成、または名前変更する必要がある場合は、ファイルシステムでアイテムを削除、作成、または名前変更のではなく、Studio内から実行してください。

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。



The screenshot shows a search results window with the following table:

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>エンクロージャ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、サービスセンター仮想...
*	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。最初に加動作し...
*	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。最初に加診断を...
*	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](#)」(375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

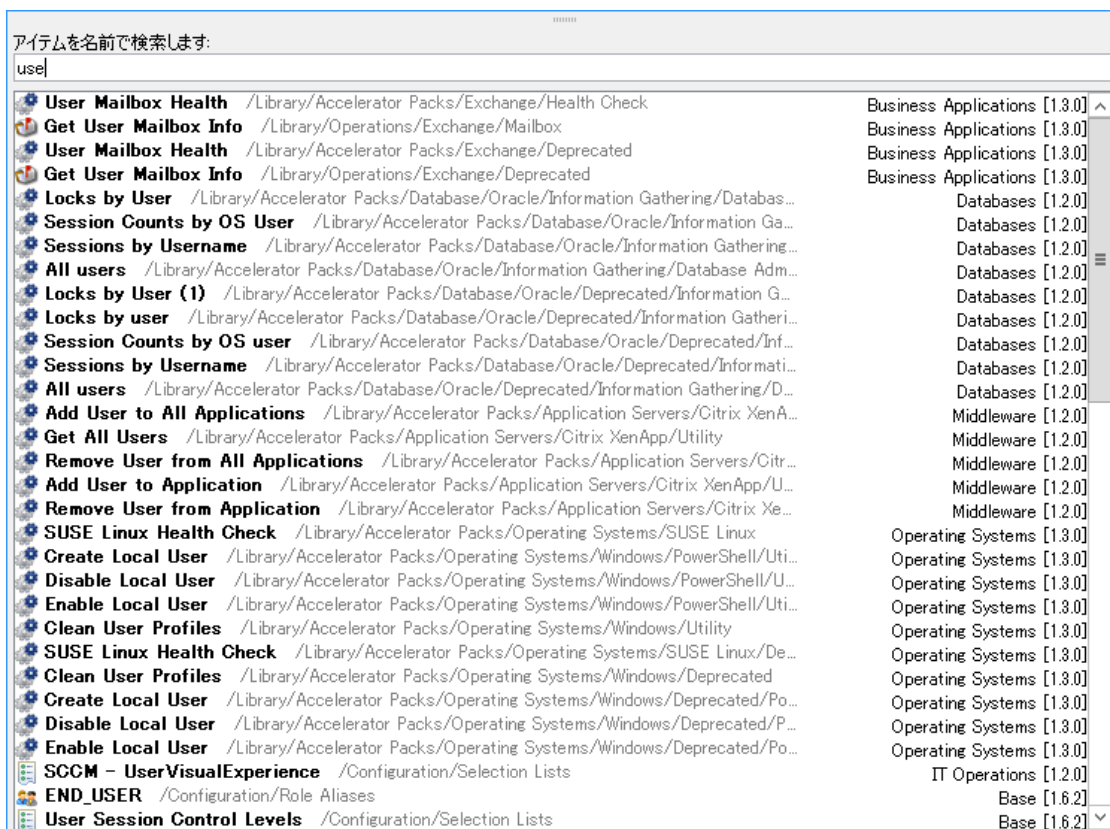
[アイテムに移動...] オプションを使用して、ワークスペース内の構成アイテムを簡単に見つけて開くことができます。

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 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。

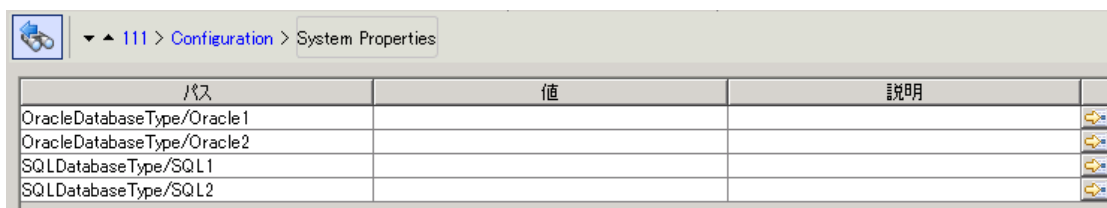
構成アイテムフォルダーの説明を設定する

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。



The screenshot shows a table with columns for 'パス' (Path), '値' (Value), and '説明' (Description). The table lists database types for Oracle and SQL on two different servers.

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。

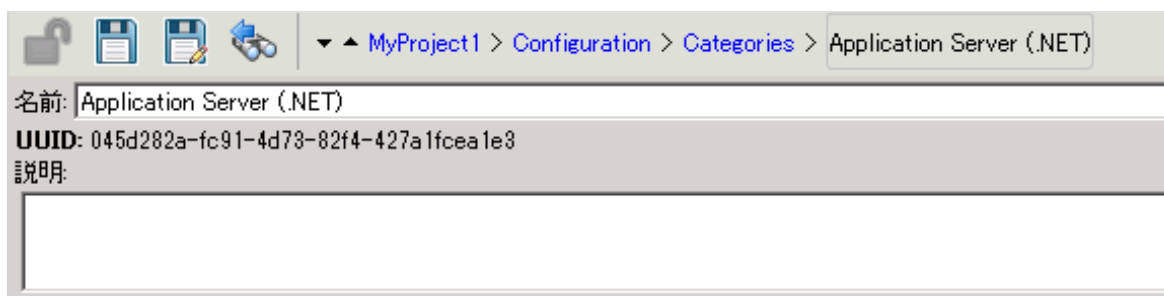
カテゴリの設定

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。Studio1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ことができます。

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



The screenshot shows the 'Application Server (.NET)' category configuration dialog. It includes a name field, a UUID field, and a description field.

名前: Application Server (.NET)
UUID: 045d282a-fc91-4d73-82f4-427a1fcea1e3
説明:

実行する操作

カテゴリの作成

1. [プロジェクト] ペインで、「**Configuration**」フォルダーを展開します。

2. **Categories**フォルダーで、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**」フォルダーと「**Categories**」フォルダーを展開し、カテゴリをダブルクリックしてエディターを開きます。
2. 変更するカテゴリをダブルクリックして新しい値を入力します。

カテゴリの名前の変更

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

カテゴリのコピー

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. 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「**Categories**」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。

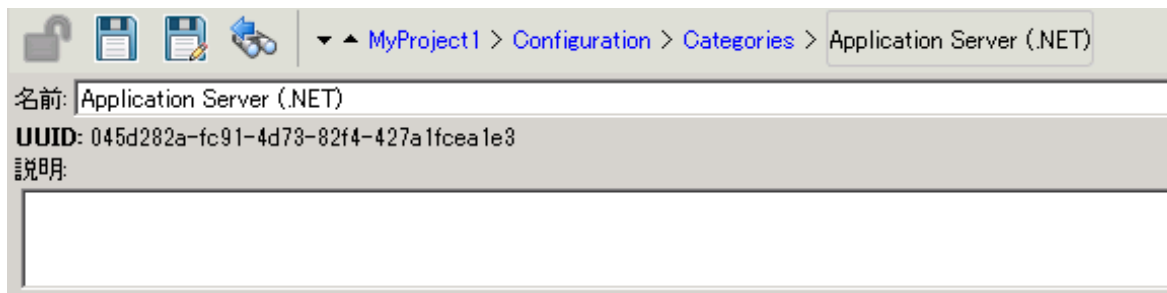
カテゴリの削除

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の詳細](#)」(391 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



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に対してのみステップを実行させることができます。

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

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

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

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

The screenshot shows a configuration window for 'Severity' within the 'Domain Terms' folder. The breadcrumb path is 'MyProject1 > Configuration > Domain Terms > Severity'. The '名前:' field contains 'Severity' and the 'UUID:' is '428ccc34-1a3d-4656-add8-250acf304354'. The '説明:' field contains 'Information, Warning, Error, Critical'. Below the text fields are '追加' and '削除' buttons. A table below shows the current entries:

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**」フォルダーを展開します。
2. 「**Domain Terms**」フォルダーで、**[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**」フォルダーと「**Domain Terms**」フォルダーを展開し、ドメインタームをダブルクリックしてエディターを開きます。
2. 値を強調表示して、**[削除]** をクリックします。

ドメインタームの値の変更

1. [プロジェクト] ペインで「**Configuration**」フォルダーと「**Domain Terms**」フォルダーを展開し、ドメイン

タームをダブルクリックしてエディターを開きます。

2. 変更する値をダブルクリックして新しい値を入力します。

ドメインタームのコピー

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的詳細](#)」(391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 Terms**」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があります。

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、『HPE OO
Application Program Interface (API) Guide』を参照してください。

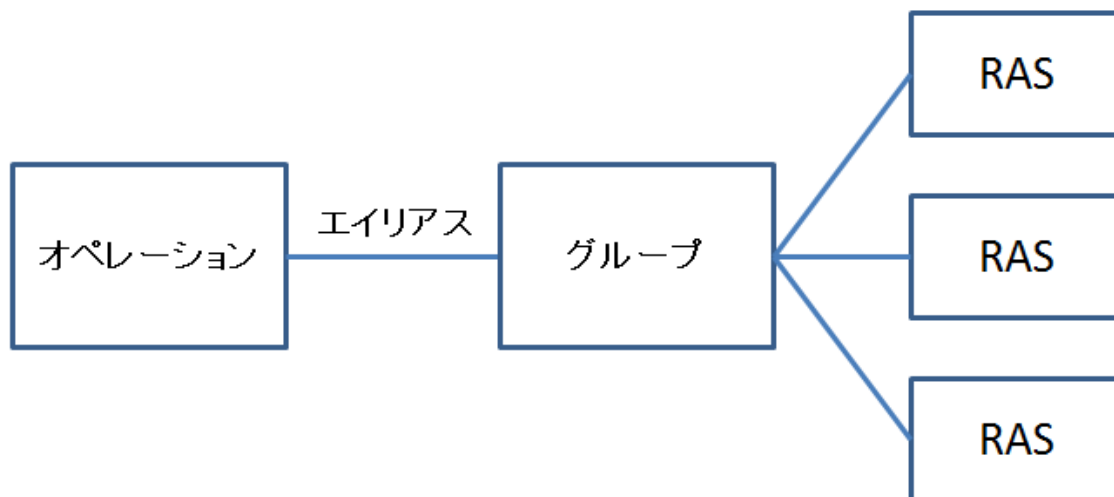
グループエイリアス

グループエイリアスを使用すると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
ができます。

1. 作成時は、作成者はワーカークラス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。

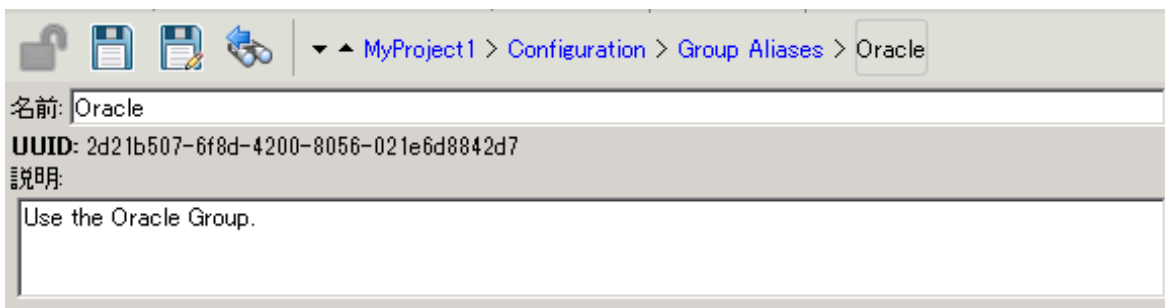
フォールバックとして、グループエイリアスがグループ名と同一の場合、そのグループに自動的にマッピングさ
れます。

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、HPE OOに、このオペレーショ
ンがこのグループのいずれかのRASで実行する必要があることを知らせます。使用するRASの選択は実
行時に決定されるため、オペレーションで設定する必要はありません。

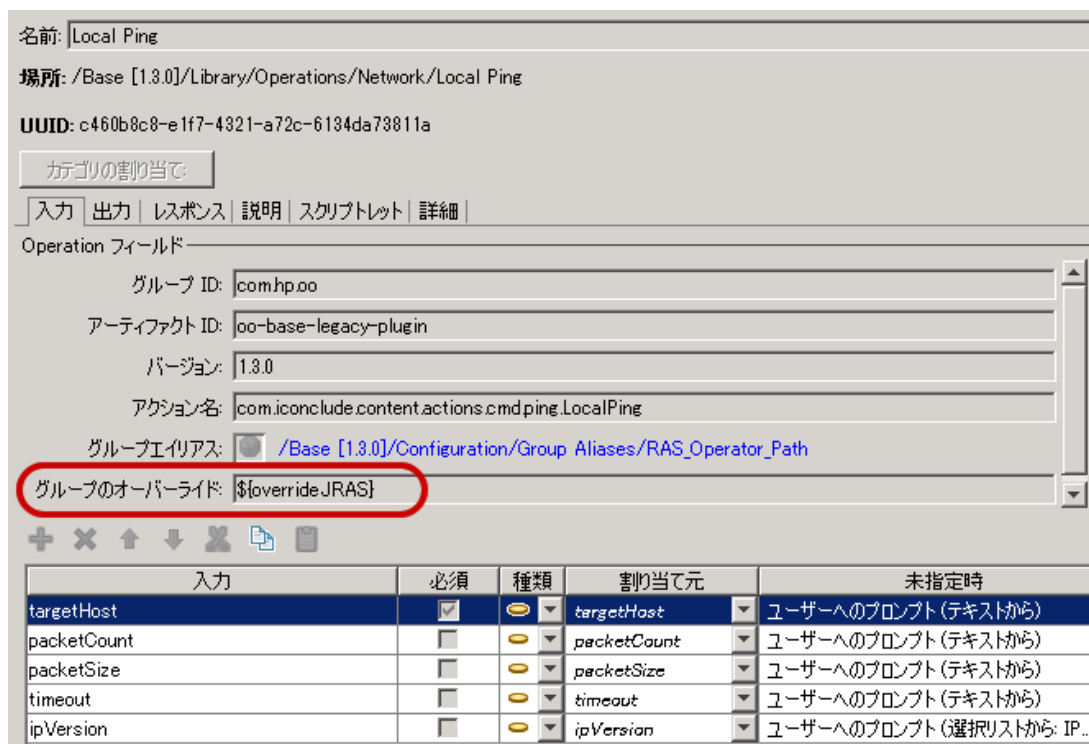
グループエイリアスは「**Configuration\Group Aliases**」フォルダーに保存されます。



RASグループのオーバーライド

HPE OO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」\(144ページ\)](#)
- 動的 - HPE OO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」\(145ページ\)](#)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](#)」(202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**」フォルダーを展開します。
2. 「**Group Aliases**」フォルダーで、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**」フォルダーと「**Group Aliases**」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. 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**」フォルダーと「**Group Aliases**」フォルダーを展開します。
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。

グループエイリアスの削除


1. [プロジェクト] ペインで、「**Configuration**」フォルダーと「**Group Aliases**」フォルダーを展開します。
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、『HPE 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**」(366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:	 /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」**(145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。

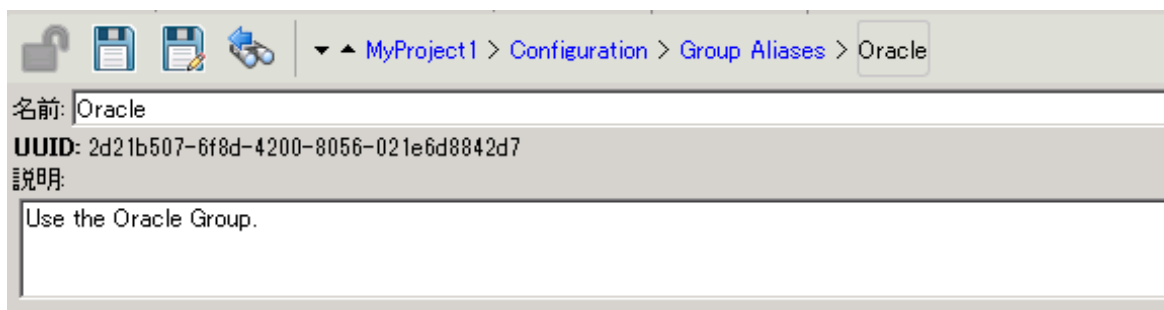
1. グループエイリアスの割り当て先のオペレーションを開きます。
2. **[入力]** タブの **[Operationフィールド]** セクションで、**[グループエイリアス]**  ボタンの横にあるリンクを右クリックします。

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:	 /Base [1.0.121]/Configuration/Group Aliases/RAS_Operator_Path
グループのオーバーライド:	\${overrideJRAS}

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



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

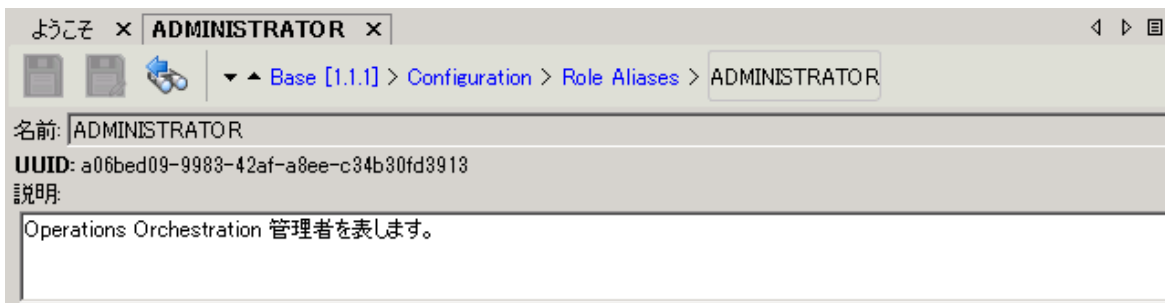
1. **[プロジェクト]** ペインで、「**Configuration**」フォルダーと「**Role Aliases**」フォルダーを展開します。
2. コピーする役割エイリアスを右クリックします。
3. **[編集]** > **[コピー]** を選択します。
4. コピー先のプロジェクトの「**Role Aliases**」フォルダーに移動します。
5. 右クリックして、**[編集]** > **[貼り付け]** を選択します。

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。

The screenshot shows the configuration window for a scriptlet named "divide by 1k". The breadcrumb navigation is "MyProject1 > Configuration > Scriptlets > divide by 1k". The "名前:" field contains "divide by 1k". The "UUID:" field contains "04c30265-7e96-42e3-ab8c-4f562e7a0e7c". The "説明:" field contains the text: "Performs a mathematical division of the input number (dividend) by 1024 (divisor). Returns the quotient rounded to the closest integer value." Below the description, there are two links: "テンプレートの挿入" and "スクリプトの確認". At the bottom, a code editor shows the following JavaScript code:

```
1 try{
2   if (scriptletInput.length == 0) scriptletResult = 0
3   else scriptletResult = java.lang.Math.round(java.lang.Double.parse
4 }
5 catch(err) {
6   scriptletResult = '';
7 }
8
```

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

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

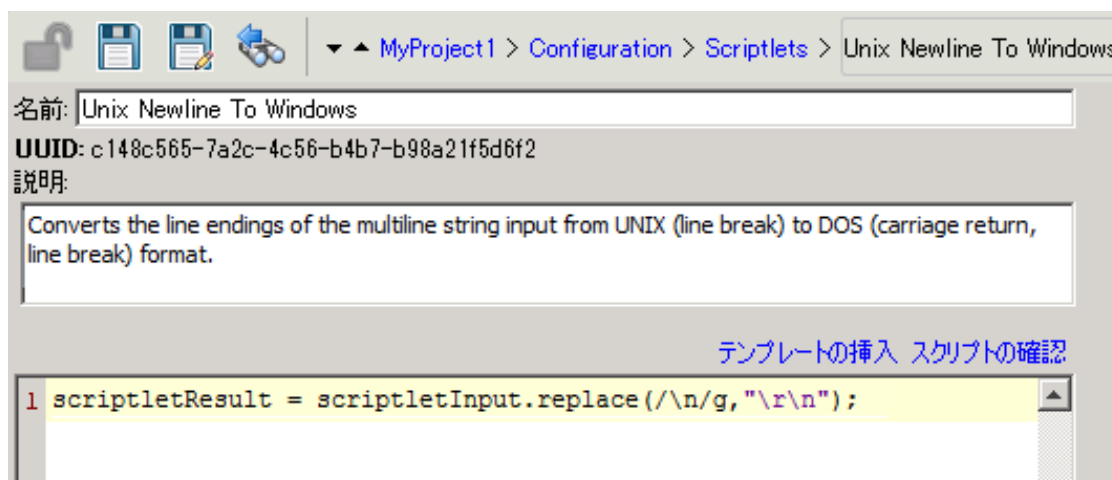
スクリプトレットの使用の詳細については、「[フロー内でのスクリプトレットの使用](#)」(307ページ)を参照してください。

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**」フォルダーを展開します。
2. 「**Scriptlets**」フォルダーで、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](#)」(261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。

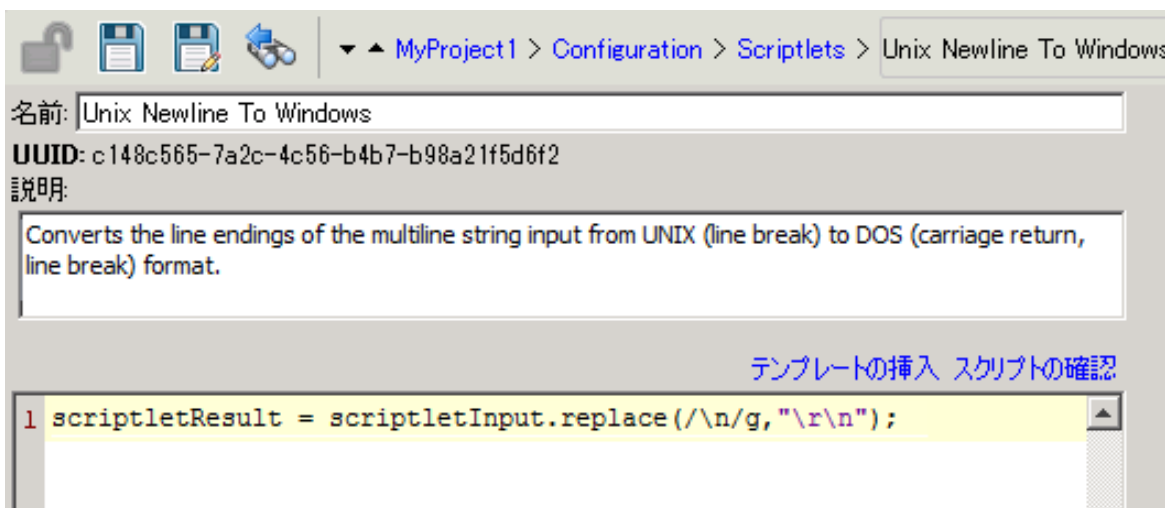
システムスクリプトレットの削除

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](#)」(391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。

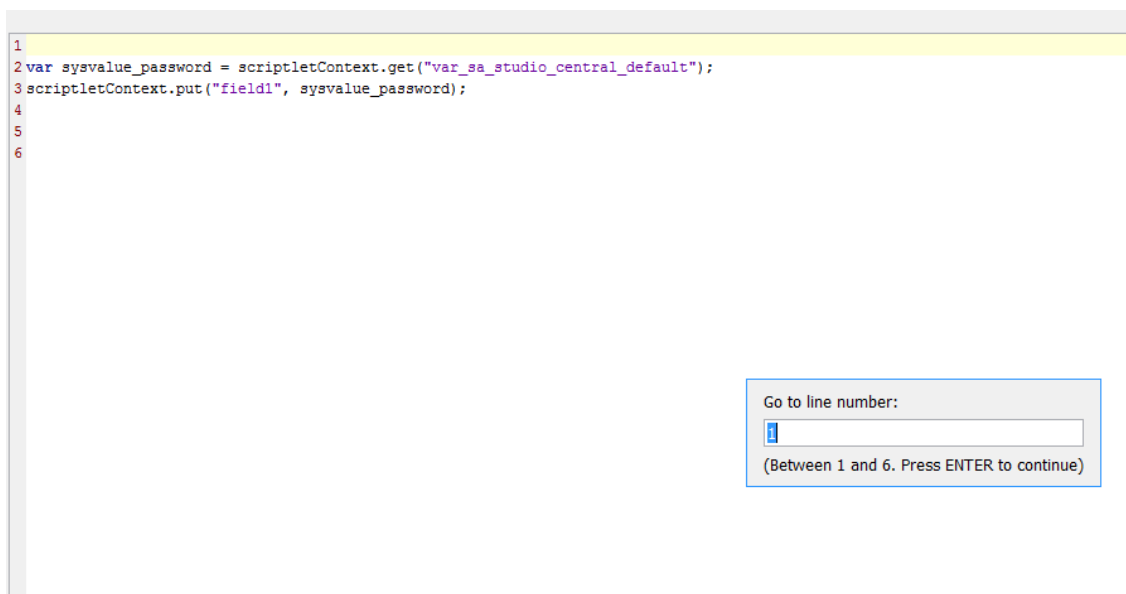
キーワード入力候補

- 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  
2 var sysvalue_password = scriptletContext.get("var_sa_studio_central_default");  
3 scriptletContext.put("field1", sysvalue_password);  
4 v  
5
```

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。

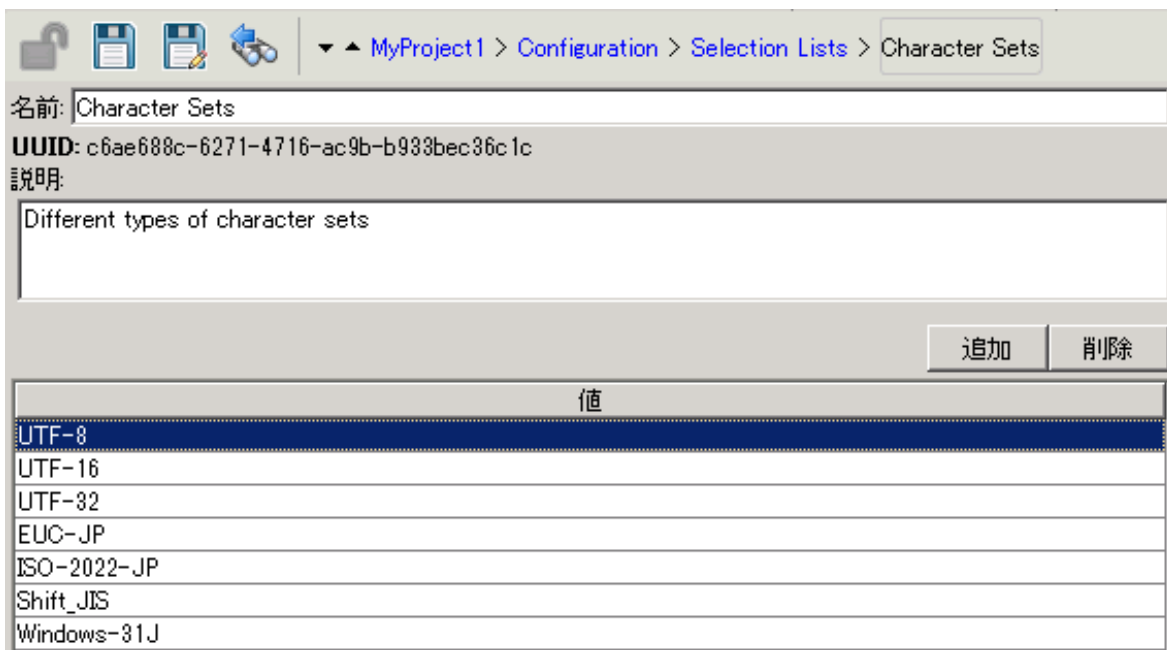


選択リストの設定

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**」フォルダーに保存されます。



注: HP Solutionsコンテンツパックの「Set Selection List」オペレーションを使用して選択リストのカスタム値を追加した場合、選択リスト自体を削除しても、カスタム値がCentralに残ります (oo_content_config_itemおよびoo_config_item_valuesテーブル)。

再デプロイの際には、新しいエンティティがカスタム値に再アタッチされます。

カスタム値を正しく削除するには、次のいずれかの回避策を使用します。

- 古いエンティティをアンデプロイした後、新しいエンティティをデプロイする前に、REST呼び出しを使用することで、構成アイテムを削除します (DELETEを指定)。

または、

REST呼び出しを使用して、カスタム値を削除します (PUTで要求本文にNullを使用)。

実行する操作

選択リストの作成

- [プロジェクト] ペインで、「Configuration」フォルダーを展開します。
- 「Selection Lists」フォルダーで、右クリックして [新規] > [選択リスト] を選択します。
- 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。
- [説明] ボックスに、新しい選択リストの説明を入力します。

5. **[追加]** をクリックして、新しい選択リストの値を追加します。
6. **[値]** 列に、選択リストの値の名前を入力します。
7. **[保存]** をクリックします。

選択リストの値の削除

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

選択リストの値の変更

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

選択リストの名前の変更

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

選択リストのコピー

1. **[プロジェクト]** ペインで、「**Configuration**」フォルダーと「**Selection Lists**」フォルダーを展開します。
2. 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。

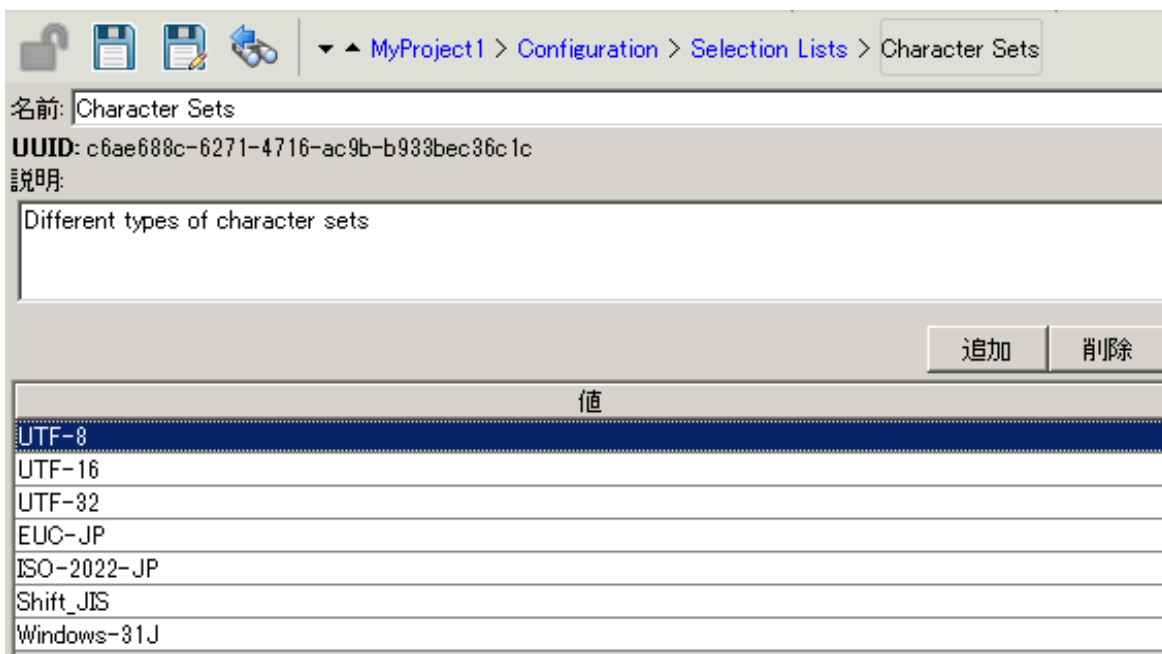
選択リストの削除

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](#) [方法の詳細](#)」(391ページ)を参照してください。

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

参考資料

選択リストのエディター



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。

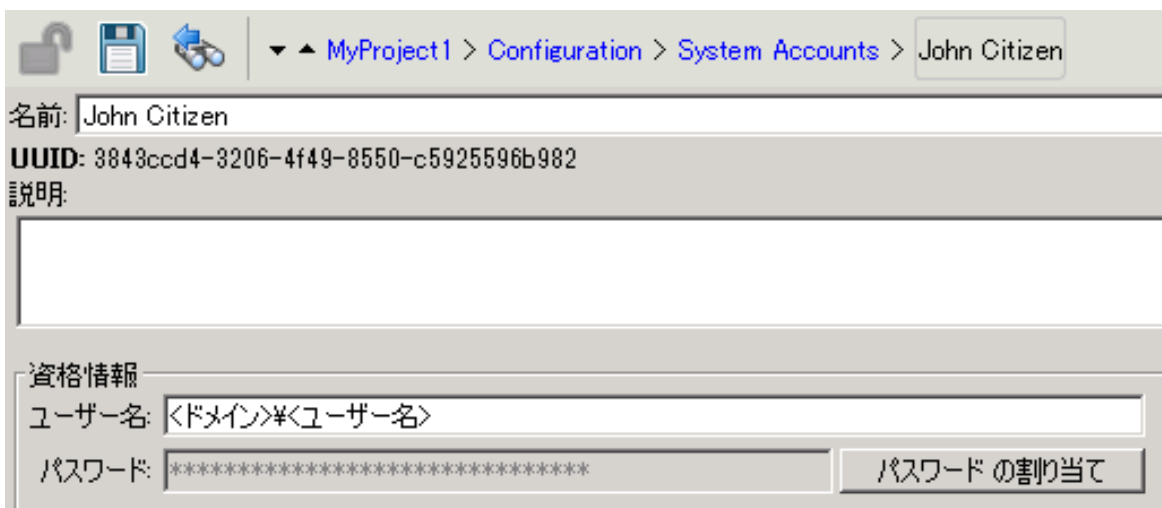
フロー作成者は、フローの作成時にシステムアカウントを使用できます。たとえば、入カソースを、システムアカウントからの資格情報に設定することができます。「[入カソースの指定](#)」(213ページ)を参照してください。

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、『HPE 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. 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. The "資格情報" section has a "ユーザー名:" field with the value "<ドメイン>¥<ユーザー名>" and a "パスワード:" field with asterisks. A "パスワードの割り当て" button is next to the password field.

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**」フォルダーと「**System Accounts**」フォルダーを展開します。
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。

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ーを移動/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**」フォルダーと「**System Accounts**」フォルダーを展開します。
2. 編集するシステムアカウントをダブルクリックします。
3. エディターで変更し、[保存] をクリックします。

システムアカウントの削除

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

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

次の場所でDELETEを実行します。/oo/rest/system-accounts/<sa_name>.

REST APIの使用の詳細については、『HPE 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せん。

[「入力データの評価」\(226ページ\)](#)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」\(313ページ\)](#)
- 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」\(307ページ\)](#)

システム評価子は「**Configuration\System Evaluators**」フォルダーに保存されます。

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。

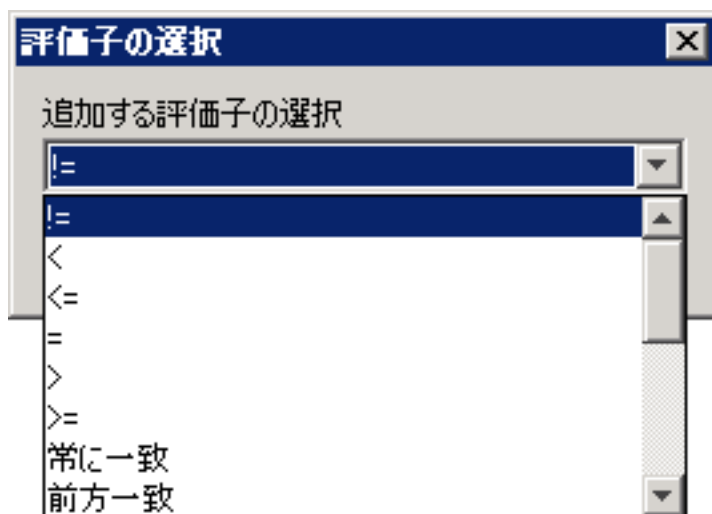
評価子の種類: <
入力を文字または数値と比較
比較対象: 1

フィルター入力のテスト
パターンが一致します (0回)

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。

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) クリア クイックコマンド

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](#)」(261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](#)」(391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」(261ページ)を参照してください。

評価子エディター – 正規表現

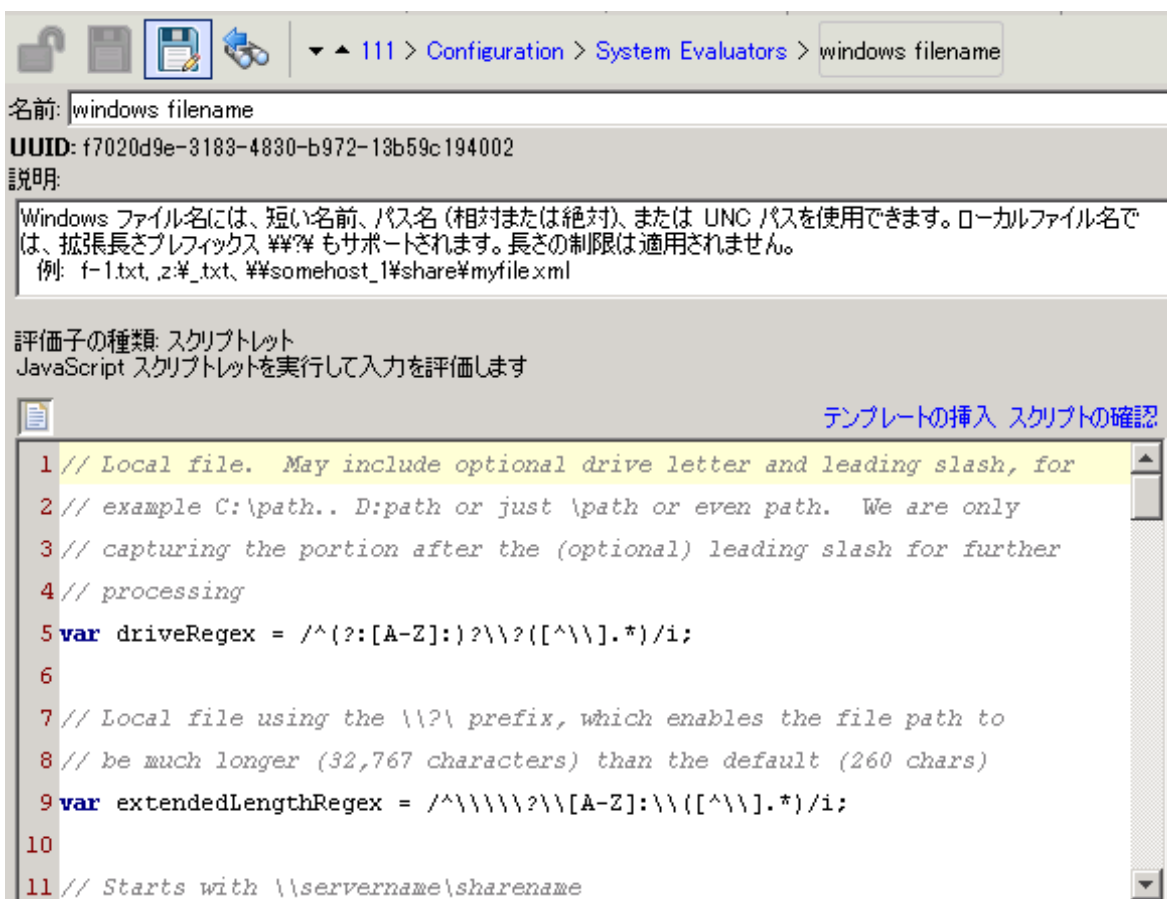
評価子エディターの外観は、選択した評価子の種類によって異なります。[評価子の選択] ダイアログボックスで[正規表現]を選択した場合、評価子エディターは次のように表示されます。

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](#)」(313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](#)」(307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**」フォルダーに保存されます。

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

フィルターへのテスト

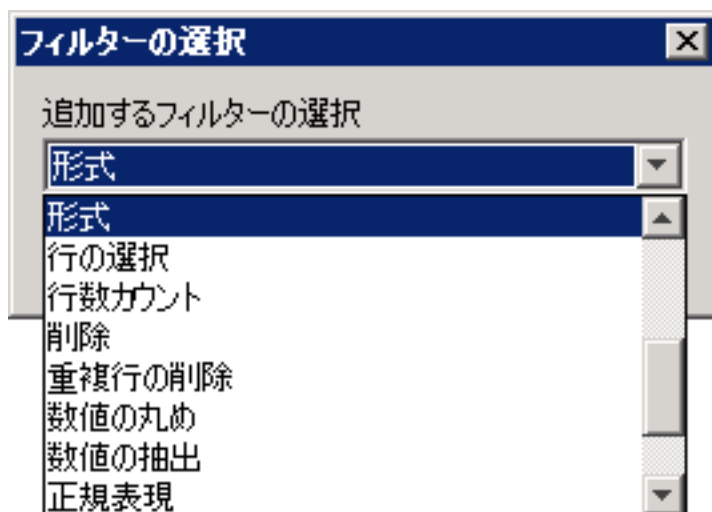
フィルター入力のテスト

クイックコマンド

実行する操作

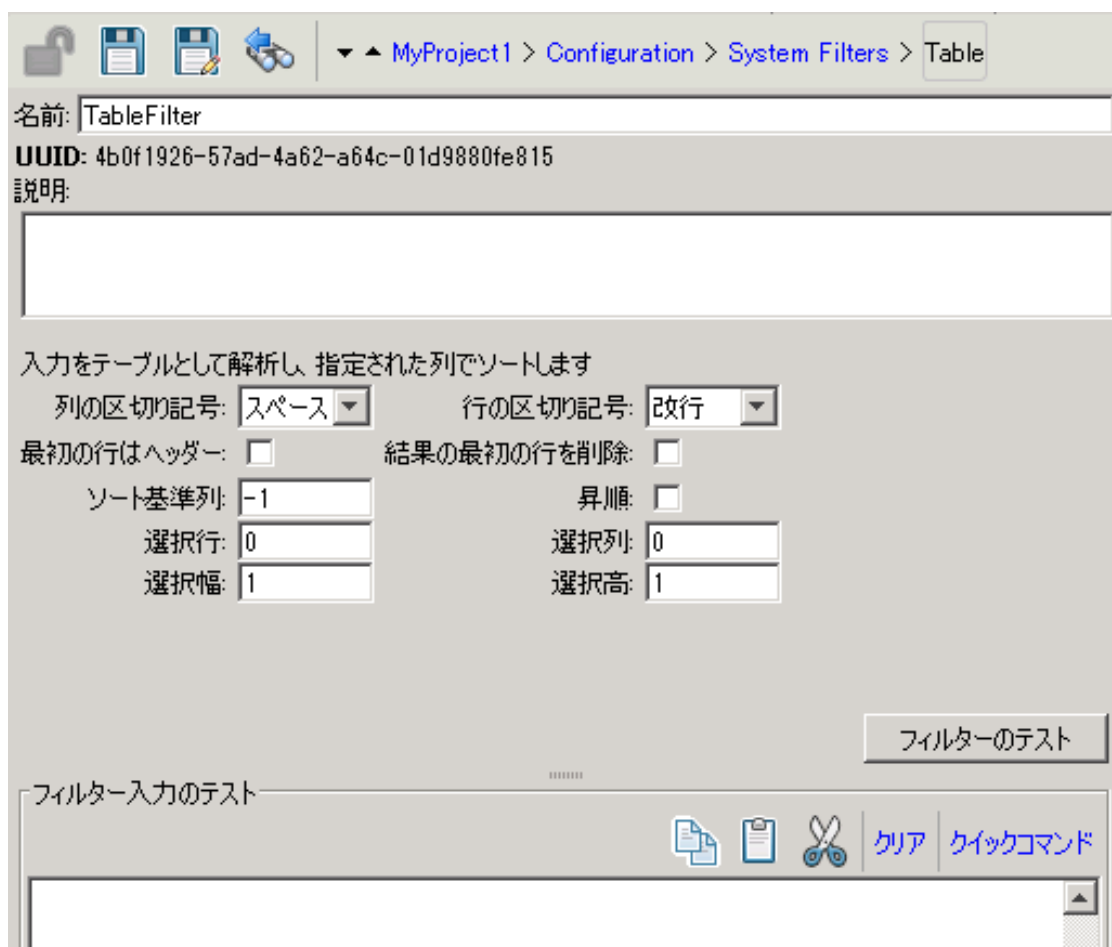
システムフィルターの作成

1. [プロジェクト] ペインで、「**Configuration**」フォルダーを展開します。
2. 「**System Filters**」フォルダーで、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](#)」(261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](#)」(261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。

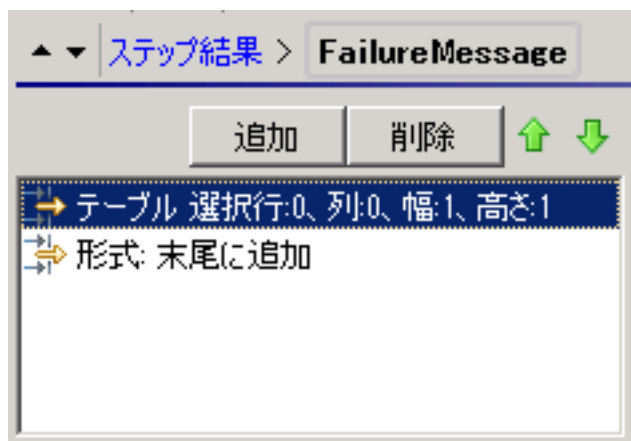
```
-Ddharma.windows.cmd.unicode=true
```

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**」フォルダーと「**System Filters**」フォルダーを展開します。
3. オペレーションのフィルターエディターで、[フィルター] リストからフィルターを「**System Filters**」フォルダーにドラッグします。



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

出力または結果でのシステムフィルターの使用

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



システムフィルターの編集

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

システムフィルターのコピー

1. [プロジェクト] ペインで、「Configuration」フォルダーと「System Filters」フォルダーを展開します。
2. 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

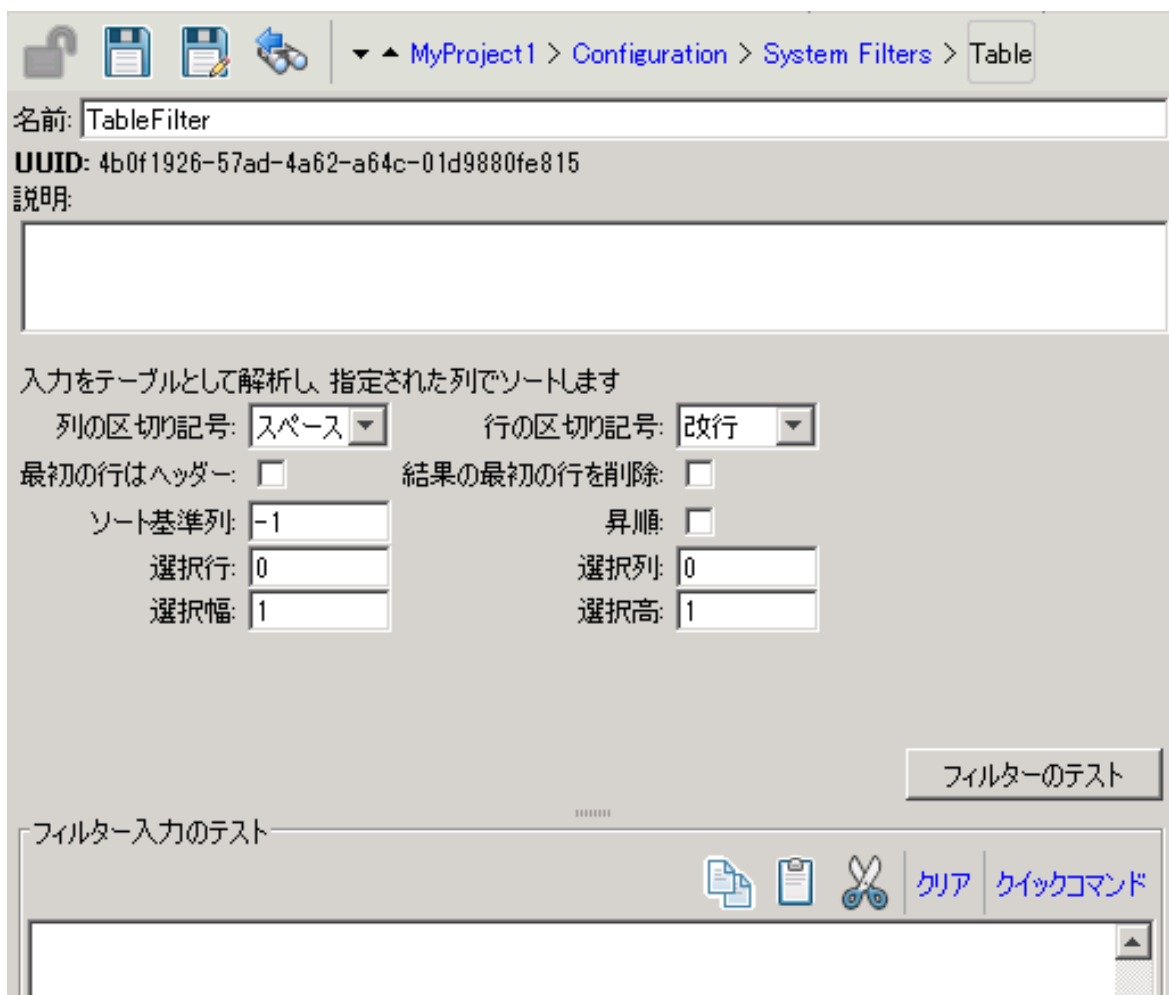
システムフィルターを削除する前に、[\[これを使用する項目は?\]](#)機能を使用して、ほかのアイテムが依存していないかを確認することをお勧めします。詳細については、[「フローとオペレーションの使用方法の詳細」\(391ページ\)](#)を参照してください。

1. [プロジェクト] ペインで、「**Configuration**」フォルダーと「**System Filters**」フォルダーを展開します。
2. システムフィルターを右クリックし、**[削除]**を選択します。
3. 確認ウインドウで **[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」\(261ページ\)](#)の「フィルターオプション」を参照してください。

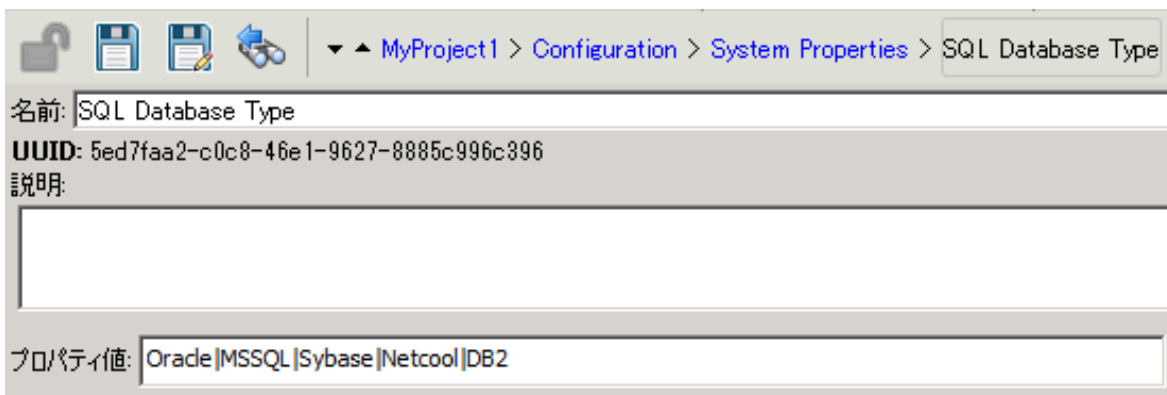


GUIアイテム	説明
名前	システムフィルターの名前が表示されます。
説明	システムフィルターの説明を入力します。

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生じるたびにこの変数を再作成する手間を省けます。システムプロパティの参照はすべて、システムプロパティの値を取得します。

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



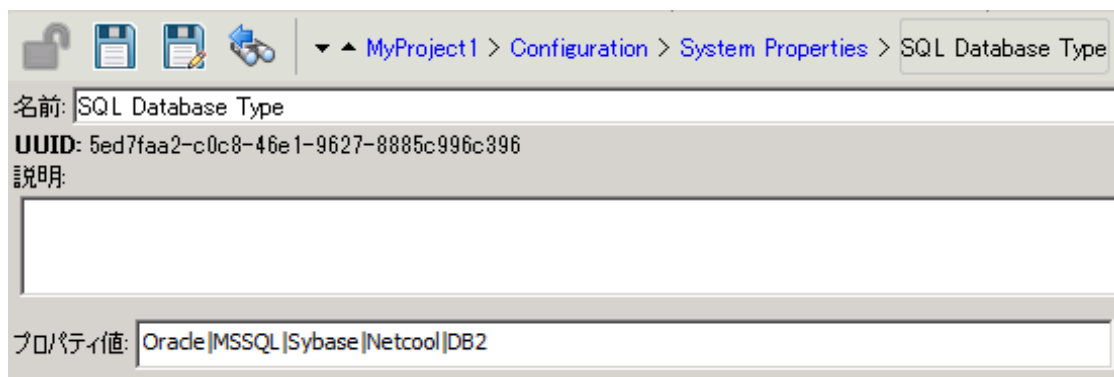
システムプロパティは「**Configuration\System Properties**」フォルダーに保存されます。

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。



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](#)」(307ページ)を参照してください。

システムプロパティの値の変更

システムプロパティの値を変更するには、次のような方法があります。

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

システムプロパティの名前の変更

1. [プロジェクト] ペインで、「**Configuration**」フォルダーと「**System Properties**」フォルダーを展開します。
2. 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**」フォルダーと「**System Properties**」フォルダーを展開します。
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方法の詳細](#)」(391ページ)を参照してください。

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

参考資料

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](#)」(290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](#)」(184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](#)」(186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。**Success**、**Diagnosed**、**No action**、**Failure**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、[**Success**] と [**Failure**] 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](#)」(285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 [Fail] レスポンスアイコンから [Fail] 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 [Success] 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](#)」(202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。

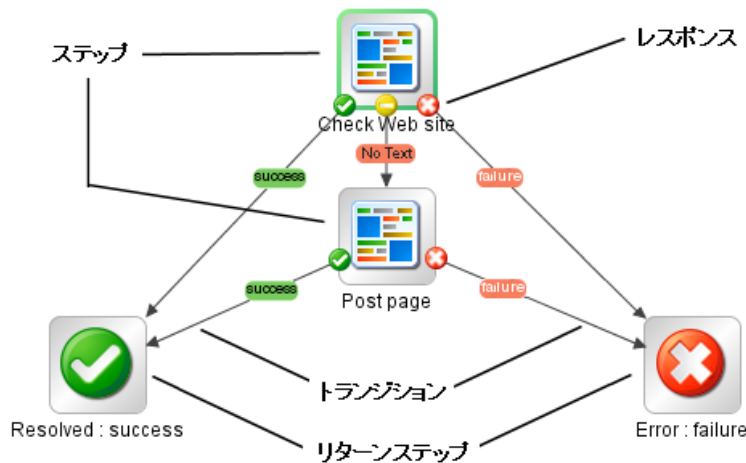
詳細情報とオプションについては、「[ステップの結果の設定](#)」(252ページ)を参照してください。

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。

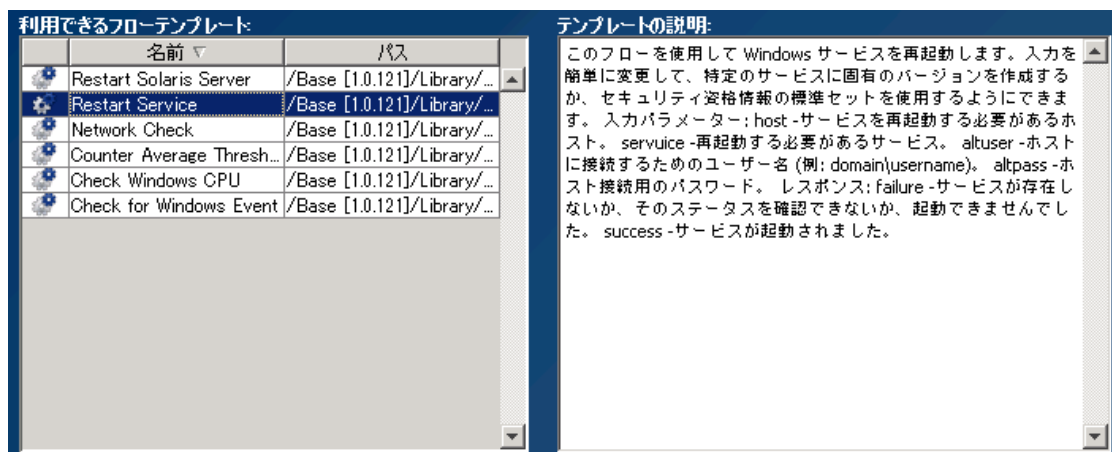
テンプレートを使用したフローの作成

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。



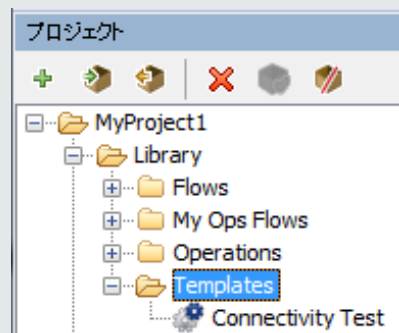
2. [ようこそ] 画面で、[新規フロー] ボタンをクリックします。
3. 表示されたテンプレートの一覧でいずれかのフローテンプレートを強調表示すると、その説明が表示されます。必要に応じて、スクロールバーをドラッグしてテキストを表示します。



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

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



フローの説明の追加

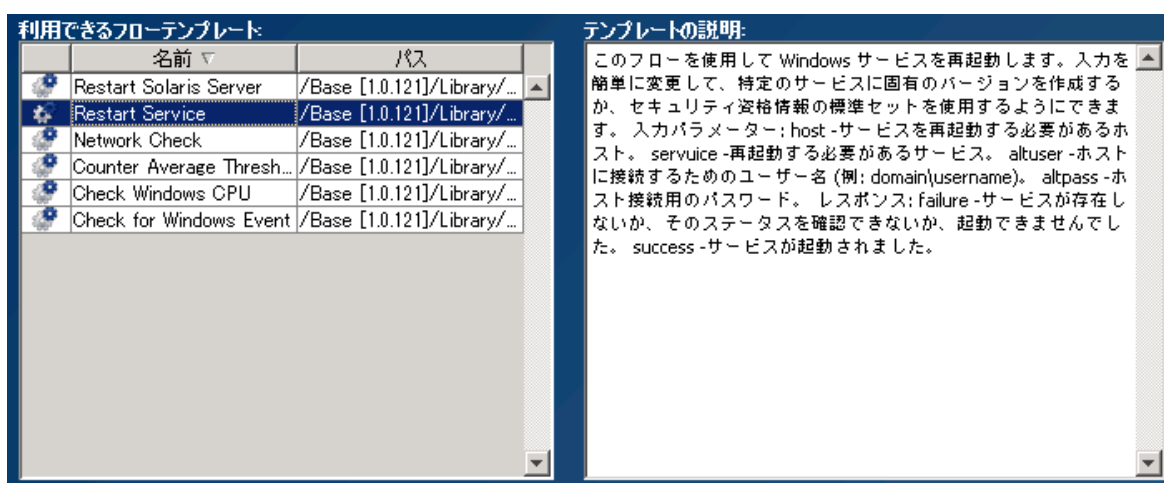
1. 作成ペイン上でフローを右クリックし、**[プロパティ]** をクリックします。

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

参考資料

利用できるフローテンプレート

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



GUI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](#)」(196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](#)」(366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

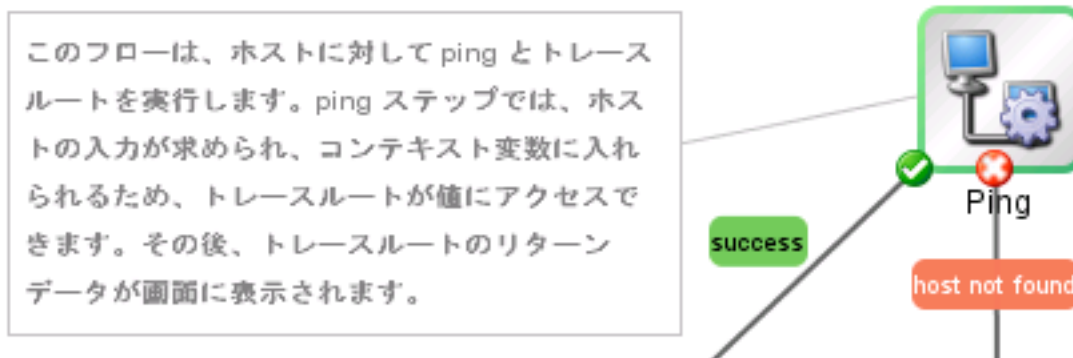
- 作成ペイン上でステップを右クリックし、[プロパティ] をクリックします。
- [説明] タブをクリックします。
- ステップの説明を入力して [OK] をクリックします。




説明入力のベストプラクティスについては、「[Best Practices for Flow Authoring](#)」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](#)」(290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  を取得します。



2. 完了  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。

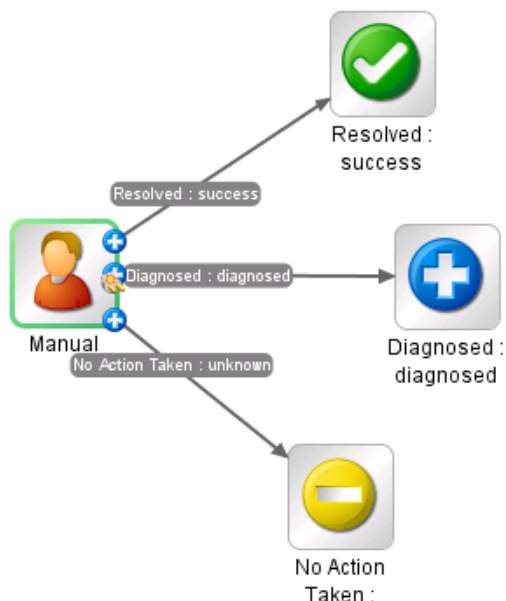
手動ステップを作成する

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。

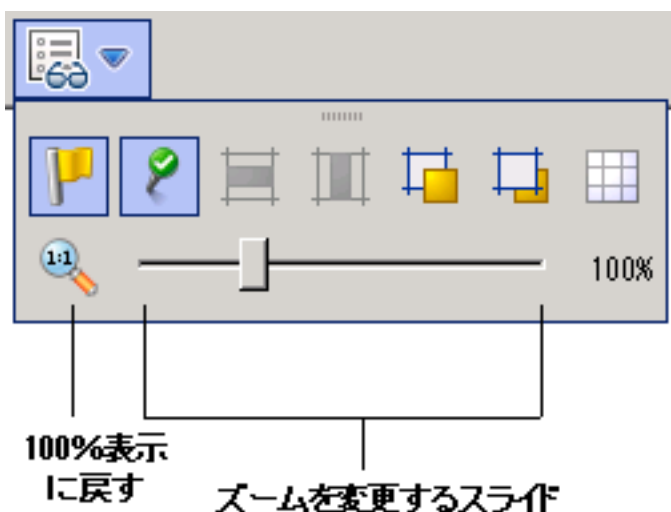
GUIアイテム	説明
このステップの実行前に常にユーザーに確認する	ステップを実行するたびにプロンプト ウィンドウを表示するには、チェックボックスをオンにします。

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。

フローの外観の調整

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


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



実行する操作


ステップのグリッドへのスナップ

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



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

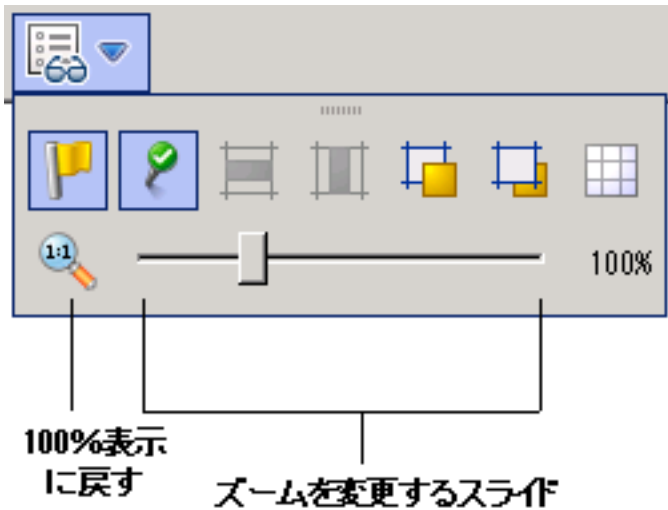
フローでは、作成ペインでオブジェクトを重ねることができます。フロー内に多くのアイテムがある場合、フローが長くなる場合があります。このような場合、最も重要なオブジェクトを前面に移動する必要があります。

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に利用できる作成ペインのグリッドを表示します。ス

ボタン	説明
	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。HPE OOで提供されている、[Accelerator Packs] フォルダー内にあるようなすぐ使えるフローを、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」\(387 ページ\)](#)を参照してください。

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

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


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


実行する操作

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。

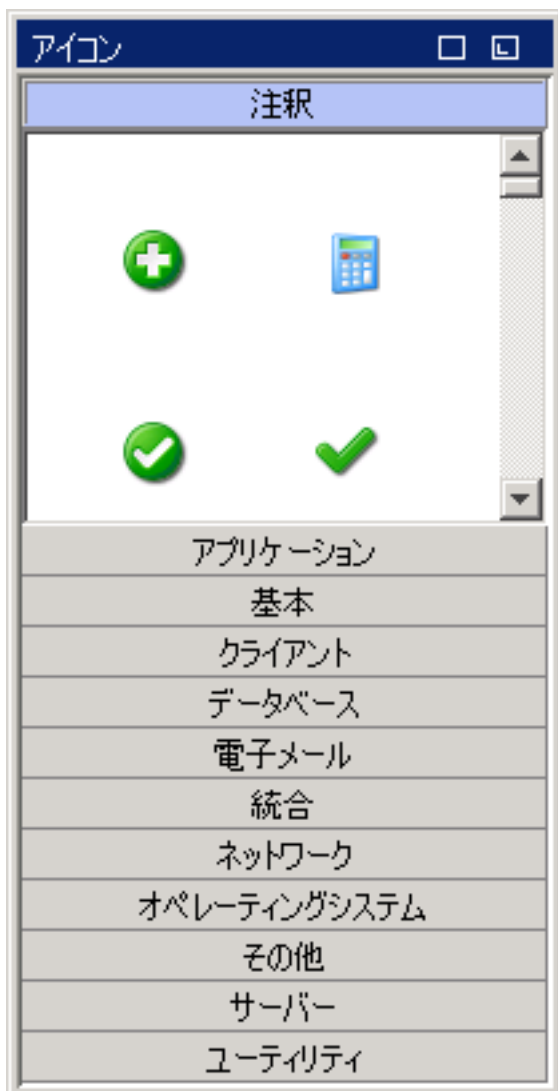
フロー図のステップのグループを移動する

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](#)」(202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](#)」(248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](#)」(186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、「[ステップのユーザープロンプトを表示する](#)」(200ページ)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](#)」(307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:

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するようにする。
- 入力の値を特定の固定の値に設定する。

- 値を別のステップから取得するように設定する。
- フロー変数を入力に割り当てる。フロー変数は、フロー全体で使用できる変数とデータ値の集合に含まれます。

フロー、オペレーション、またはステップに入力を作成できます。

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

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

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で、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つの入力が強調表示されます。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

- [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。
- [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
host	<input type="checkbox"/>	<input type="checkbox"/>	host	host	定数の使用	<未割り当て>
port	<input type="checkbox"/>	<input type="checkbox"/>	port	port	定数の使用	<未割り当て>
command	<input type="checkbox"/>	<input type="checkbox"/>	command	command	定数の使用	<未割り当て>
pty	<input type="checkbox"/>	<input type="checkbox"/>	pty	pty	定数の使用	<未割り当て>
username	<input type="checkbox"/>	<input type="checkbox"/>	username	username	定数の使用	<未割り当て>
password	<input type="checkbox"/>	<input type="checkbox"/>	password	password	定数の使用	<未割り当て>
arguments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	arguments	arguments	定数の使用	<未割り当て>
privateKeyFile	<input type="checkbox"/>	<input type="checkbox"/>	privateKeyFile	privateKeyFile	定数の使用	<未割り当て>
timeout	<input type="checkbox"/>	<input type="checkbox"/>	timeout	timeout	定数の使用	<未割り当て>
characterSet	<input type="checkbox"/>	<input type="checkbox"/>	characterSet	characterSet	定数の使用	<未割り当て>



- 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。

-  ボタンをクリックします。

- 新しい入力の名前を入力して、[OK] をクリックします。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。

- (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。
- [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**  - 単一の値を指定します。
- **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記号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]** リスト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]** リスト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]** リストで <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](#)」(213ページ)を参照してください。

注: Studio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](#)」(408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:] 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:] 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。
[ソース区切り記号] フィールドにフロー変数の区切り記号を入力します。

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を使用されます。

- **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](#)」(213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](#)」(408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] リスト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] リスト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] リストで <未割り当て> を選択します。

11. デフォルトでは、Studio1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](#)」(226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](#)」(226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。削除され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、入力のリストにグレーのイタリック体で表示されます。






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

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 	<p>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。入力名がすで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)。</p> <p>入力名がすで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。</p>

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。

インスペクター						
ステップ名: Compute Size						
入力 結果 表示 説明 詳細 スクリプトレット						
     						
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
host	<input type="checkbox"/>	<input type="checkbox"/>	host	host	定数の使用	<未割り当て>
port	<input type="checkbox"/>	<input type="checkbox"/>	port	port	定数の使用	<未割り当て>
command	<input type="checkbox"/>	<input type="checkbox"/>	command	command	定数の使用	<未割り当て>
pty	<input type="checkbox"/>	<input type="checkbox"/>	pty	pty	定数の使用	<未割り当て>
username	<input type="checkbox"/>	<input type="checkbox"/>	username	username	定数の使用	<未割り当て>
password	<input type="checkbox"/>	<input type="checkbox"/>	password	password	定数の使用	<未割り当て>
arguments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	arguments	arguments	定数の使用	<未割り当て>
privateKeyFile	<input type="checkbox"/>	<input type="checkbox"/>	privateKeyFile	privateKeyFile	定数の使用	<未割り当て>
timeout	<input type="checkbox"/>	<input type="checkbox"/>	timeout	timeout	定数の使用	<未割り当て>
characterSet	<input type="checkbox"/>	<input type="checkbox"/>	characterSet	characterSet	定数の使用	<未割り当て>

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</p>

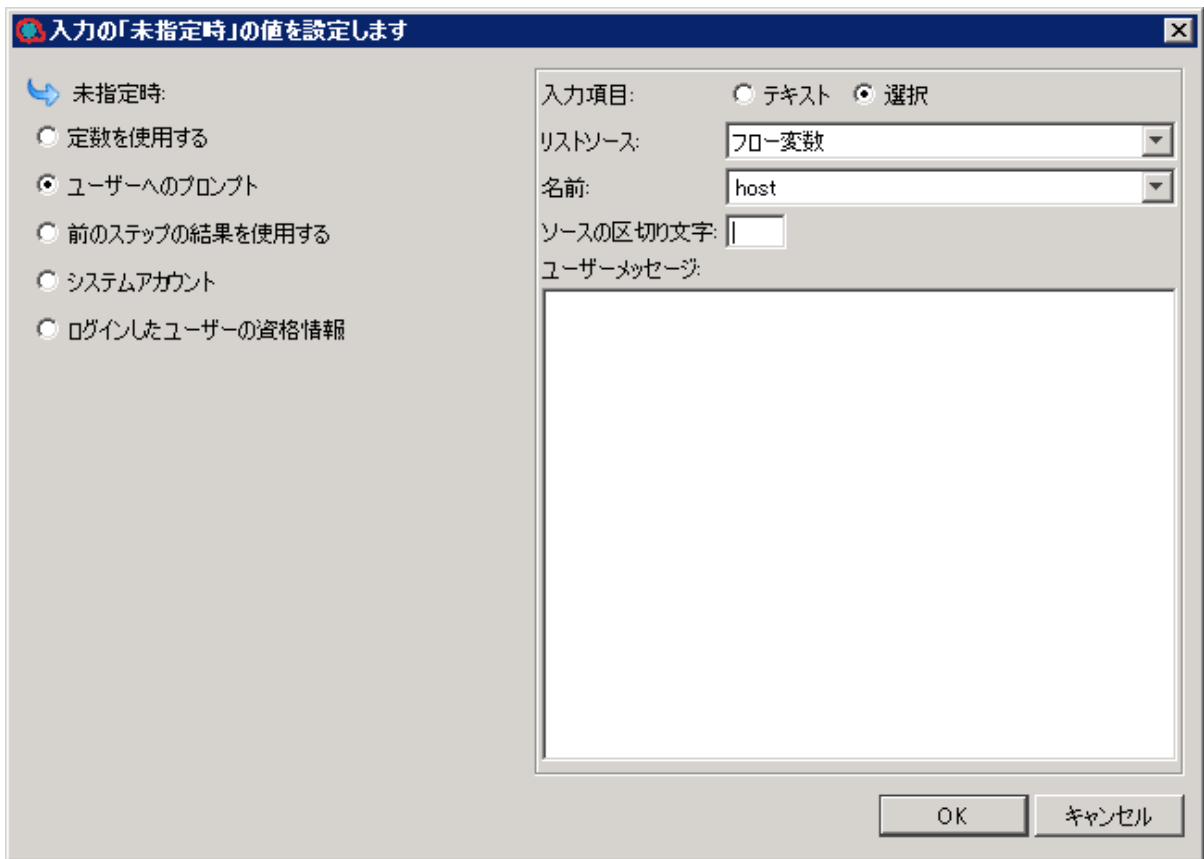
	ば、 $\${input1}$ のようになります。
必須	この入力を必須にします。
種類 	<p>[種類] リストで、入力の値を取得する方法を指定します。次のオプションを選択できます。</p> <ul style="list-style-type: none"> • 単一値  • 値のリスト  - 複数のターゲットに対してオペレーションを実行できます。
割り当て元	入力の値の取得元を指定します。[割り当て元] リストから値の取得元の入力を選択します。
未指定時	<p>[割り当て元] 列で指定したフロー変数が存在しない場合またはその変数に値が格納されていない場合に実行する必要がある操作が表示されます。</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	入り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。</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。[ソース区切り記号] フィールドにドメインタームの区切り記号を入力します。</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>
<p>前のステップの結果を使用する</p>	<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を使用されます。</p>
<p>システムアカウント:</p>	<p>システムアカウント名と資格情報の種類を入力します。</p>
<p>ログインしたユーザーの資格情報</p>	<p>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。</p>
<p>リストソース</p>	<p>リスト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。 <p>[名前] リストで、ユーザーに提示するリストを選択します。</p> <p>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。</p> <ul style="list-style-type: none"> [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。 <p>[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。</p> <ul style="list-style-type: none"> [フロー変数] - フロー変数の内容からリストを作成します。 <p>[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</p>

	<p>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。</p> <p>[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。</p>
ユーザーメッセージ	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。

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」
- ・「前のステップの結果を入力ソースに指定」
- ・「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](#)」(202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。

入力の「未指定時」の値を設定します

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
- システムアカウンント
- ログインしたユーザーの資格情報

入力項目: テキスト 選択

リストソース: フロー変数

名前: host

ソースの区切り文字: |

ユーザーメッセージ:

OK キャンセル

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て使用することもできます。\${variablename} の形式で指定します。たとえば、Ping of \${targethost} succeededのようになります。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す。たとえば、ユーザーは場所の選択をフローの開始時に行うようにすることができます。

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することができます。

1. 入力を作成して、種類を [値のリスト] に設定します。
2. 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. 入力値が複数値の場合は、[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く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](#)」(160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](#)」(219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く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](#)」(160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、Studio1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。入力名がすで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行が追加されます。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タイミングを指定します。

インスペクター						
ステップ名: Compute Size						
入力 結果 表示 説明 詳細 スクリプトレット						
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
host	<input type="checkbox"/>	<input type="checkbox"/>	👉	host	定数の使用	📄 <未割り当て>
port	<input type="checkbox"/>	<input type="checkbox"/>	👉	port	定数の使用	📄 <未割り当て>
command	<input type="checkbox"/>	<input type="checkbox"/>	👉	command	定数の使用	📄 <未割り当て>
pty	<input type="checkbox"/>	<input type="checkbox"/>	👉	pty	定数の使用	📄 <未割り当て>
username	<input type="checkbox"/>	<input type="checkbox"/>	👉	username	定数の使用	📄 <未割り当て>
password	<input type="checkbox"/>	<input type="checkbox"/>	👉	password	定数の使用	📄 <未割り当て>
argyments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	👉	argyments	定数の使用	📄 <未割り当て>
privateKeyFile	<input type="checkbox"/>	<input type="checkbox"/>	👉	privateKeyFile	定数の使用	📄 <未割り当て>
timeout	<input type="checkbox"/>	<input type="checkbox"/>	👉	timeout	定数の使用	📄 <未割り当て>
characterSet	<input type="checkbox"/>	<input type="checkbox"/>	👉	characterSet	定数の使用	📄 <未割り当て>

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	[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。

	<ul style="list-style-type: none"> • 単一値  - 単一の値を選択します。 • 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: 5px;"> <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	<p>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。</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>

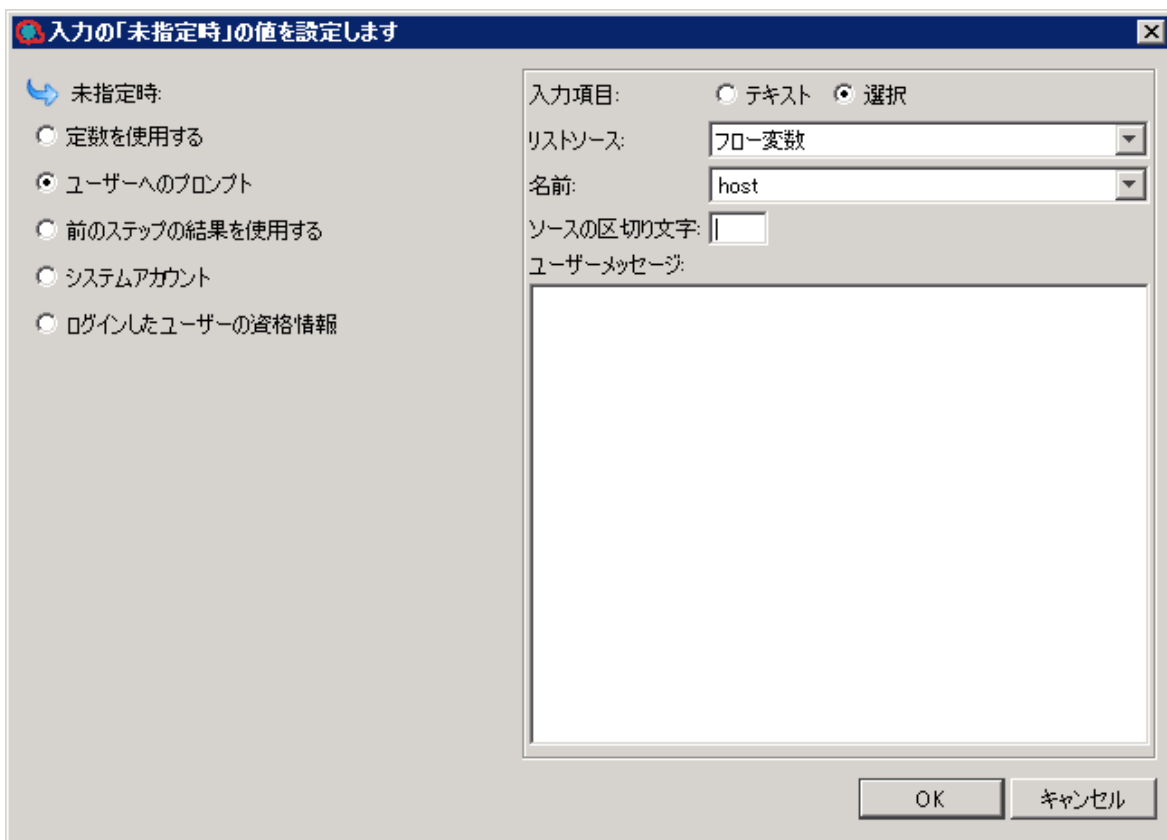
ステップのインスペクター > [入力] タブ > [未指定時] ペイン/[未指定時] ダイアログボックス




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:	定数値:		
<input checked="" type="radio"/> 定数を使用する			
<input type="radio"/> ユーザーへのプロンプト			
<input type="radio"/> 前のステップの結果を使用する			
<input type="radio"/> システムアカウント			
<input type="radio"/> ログインしたユーザーの資格情報			
割り当て先変数:	input1		
<input type="checkbox"/> 機密データ	<input type="checkbox"/> 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。
[割り当て元] 列	<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</p>

	示されます。
[未指定時] 列	<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] 列	入力 の値を割り当てるフロー変数を選択します。
名前	ステップの名前を表示します (読み取り専用)。
入力の種類	入力 の種類を表示します。ここで修正できます。
割り当て元変数	入力 のソースとなるフロー変数の名前を入力または選択します。
未指定時	[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。
「未指定時: <action>」構成	[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。
割り当て先変数	入力 の値を割り当てるフロー変数を選択します。
入力区切り記号	値を区切るために使用する区切り記号を入力します。
必須	この入力を必須にします。
検査形式	<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、Studio1によって有効なIPアドレス形式かどうか確認されます。</p>
次の名前で記録	<p>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 OO 9.xからのレガシーフィールドです。</p> <p>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</p>

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:

- 入力電子メールアドレスの場合、評価子を使って、入力正しい電子メールの形式であるかを確認できます。
- 入力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せん。

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

評価が必要な対象に対する標準のシステム評価子がない場合、追加のシステム評価子を作成することができます。システム評価子の作成の詳細については、「[システム評価子の設定](#)」(163ページ)を参照してください。

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](#)」(313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](#)」(307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: 入力の種類:

割り当て元変数:

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:

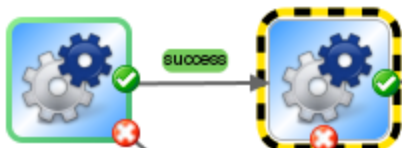
機密データ 必須 **検査形式**

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](#)」(202ページ)を参照してください。

トランジションの作成

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



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

オペレーションのレスポンスの設定の詳細については、「[レスポンスの設定](#)」(233ページ)を参照してください。

注: HPE OO 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HPE OO Centralの結果の概要エリアに表示さ

れます。

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

フロー変数を含む説明を作成する

説明内にフロー変数を使用して、可変の情報を格納できます。たとえば、サーバー名がフロー変数 `servename` 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`-${servename}`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かを選択します。

ゲート制御式トランジションは赤で表示されます。

1. 2つのステップ間にトランジションを作成します。
2. トランジションをダブルクリックしてトランジションのインスペクターを開きます。
3. **[続行前にユーザーのグループを確認]** チェックボックスをオンにします。

4. **[必須の役割エイリアス]** リストから、フローの実行を続けるためにユーザーに割り当てる必要がある役割を選択します。

トランジション後に実行がハンドオフされるようにする

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

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

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

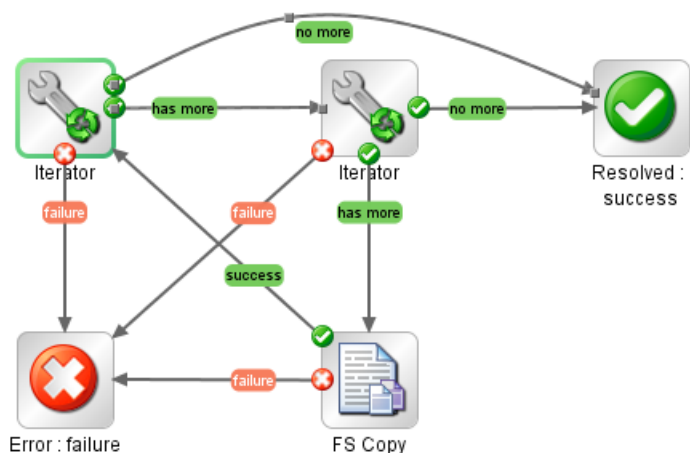
トランジションの完了をフローのROI値でカウントする

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

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

曲線定義点を追加して曲線のトランジションを作成する

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



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

曲線定義点を削除する

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

トランジション名を移動する

トランジション名をクリックして別の場所にドラッグします。

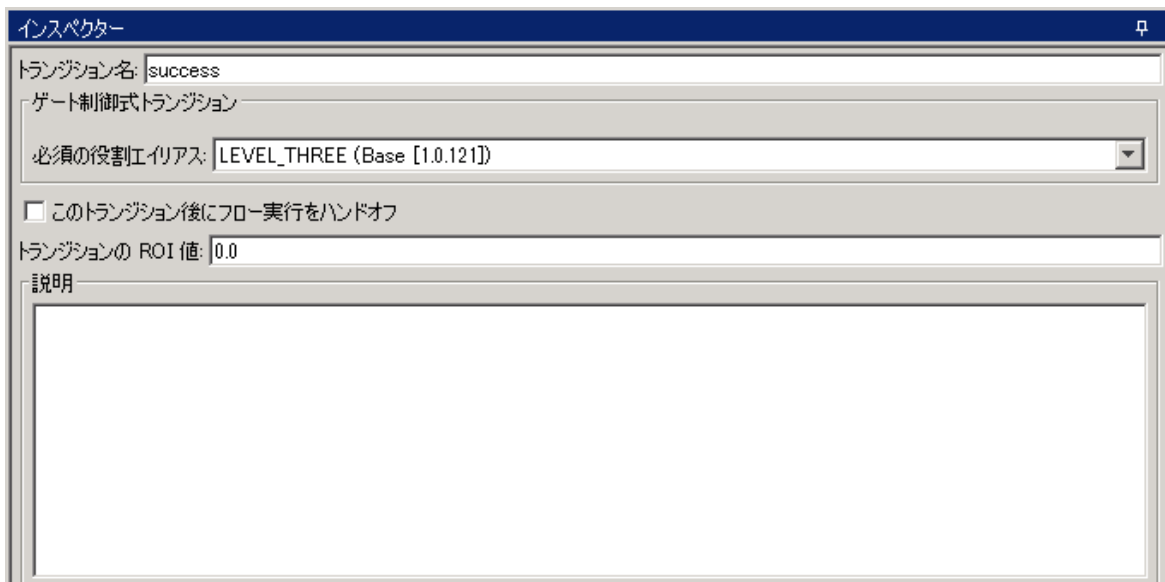
2つのステップ間のトランジションを削除する

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

参考資料

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。






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、HPE 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。

- **Resolved**  - 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。
- **Diagnosed**  - 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。
- **No Action Taken**  - 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、**No Action Taken**  ではなく**Resolved**  を返します。

- **Error**  - 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。

- **More items** 
- **No more items** 
- **Failure** 

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](#)」(285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**Success**レスポンスのみ返すルールを作成できません。

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

オペレーションまたはフローへのレスポンスの追加


1. オペレーション/フローの[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。

- Resolved:
- Diagnosed:
- No action:
- Failure:

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。

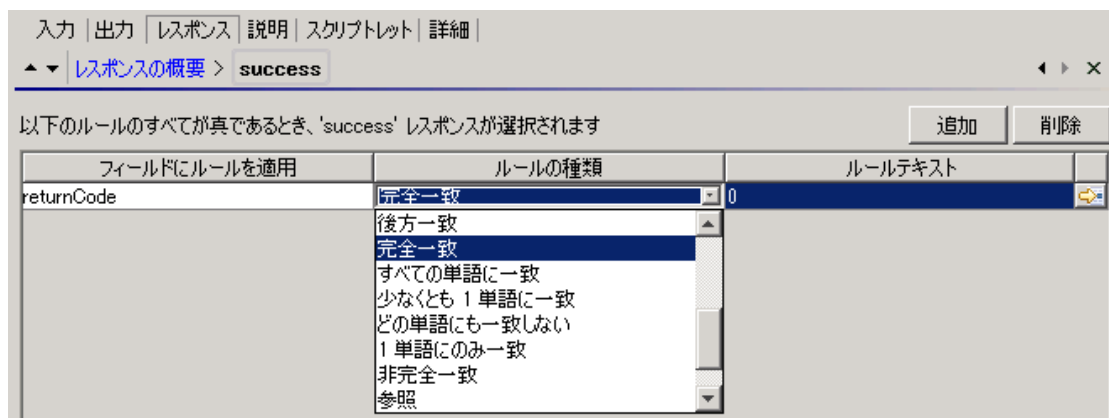
7. レスポンスのルールを作成するには、レスポンスの行の右端にある右向き矢印をクリックします。
詳細については、以下の「レスポンスのルールを作成する」を参照してください。

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。



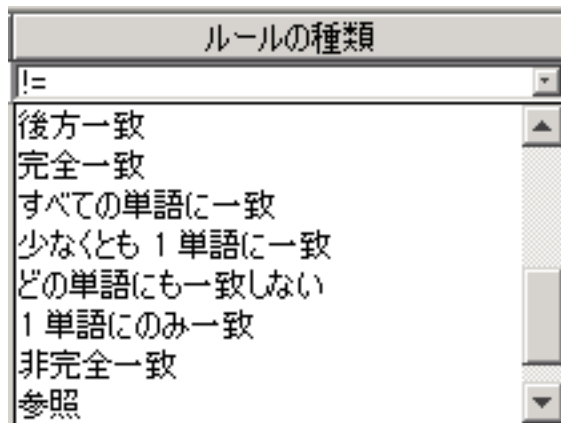
3. **[フィールドにルールを適用]**列で、ルールをテストする値を持つ結果フィールドを選択します。

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](#)」(326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。




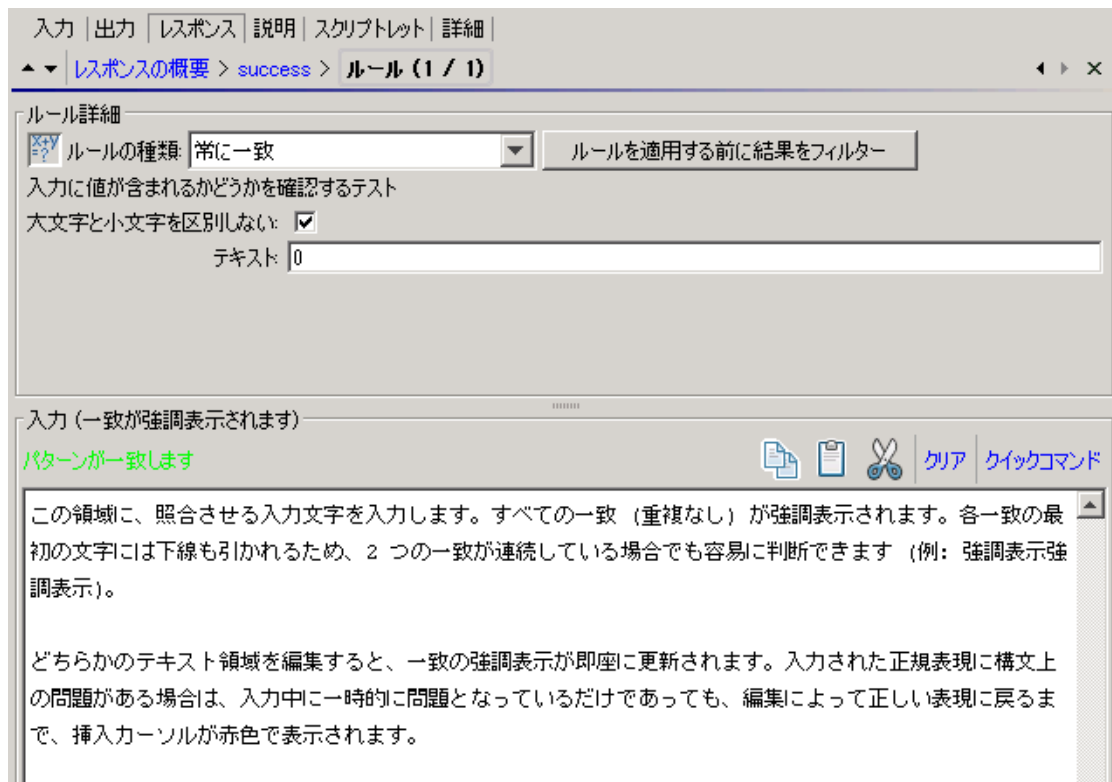
- [=], [!=], [前方一致], [部分一致], [すべての単語に一致], [少なくとも1単語に一致]などの単純な演算子を選択します。
- 正規表現を作成するには、[正規表現]を選択します。
- スクリプトレットを作成するには、[スクリプトレット]を選択します。
- 共有ルールへの参照を作成するには、[参照]を選択します。

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](#)」(307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](#)」(313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](#)」(261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](#)」(313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せん。

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、**Error**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「**Error: threshold not met**」と表示される新しいレスポンスを**Error**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**Error**リターンステップを作成するときに、レスポンスとして「**threshold not met**」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](#)」(285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確認できます。

名前: Network Check
 場所: /test/Library/My Ops Flows/Network Check
 UUID: 1cef472a-083d-4cd2-9928-f16148cc97cf
 カテゴリの割り当て: Network (Base [1.8.0])
 入力 | 出力 | レスポンス | 説明 | スクリプトレット |
 レスポンスの概要

レスポンス	種類
success	
failure	

GUIアイテム	説明
(レスポンスの追加)	新しいレスポンス行を追加します。
(レスポンスの削除)	選択したレスポンス行を削除します。
	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。
種類	レスポンスの種類 (読み取り専用) <ul style="list-style-type: none"> Success/resolved: Diagnosed: No action: Failure:

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確認できます。

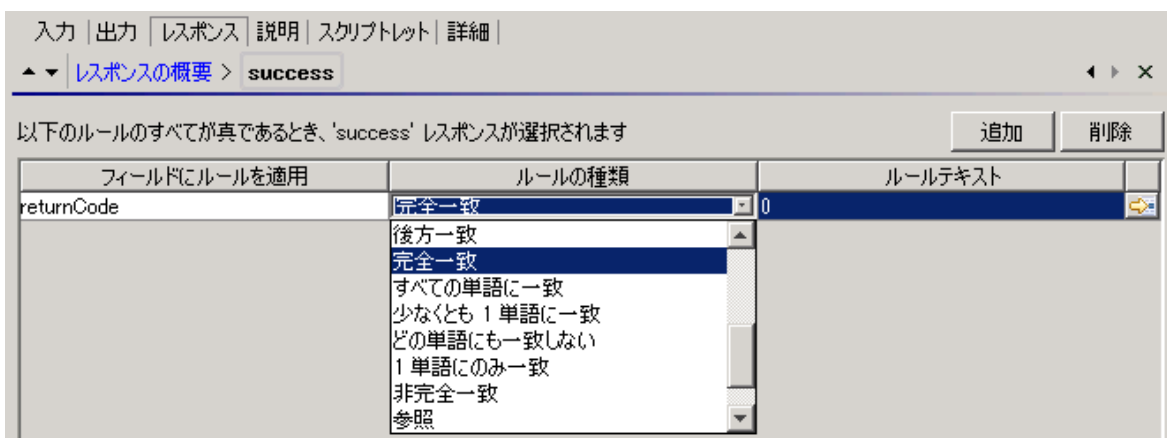
スを指定します。





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"> • Success/resolved: • Diagnosed: • No action: • Failure:
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**Success**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Aに表示されます。
-----------------	---

[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、「**Error: threshold not met**」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。

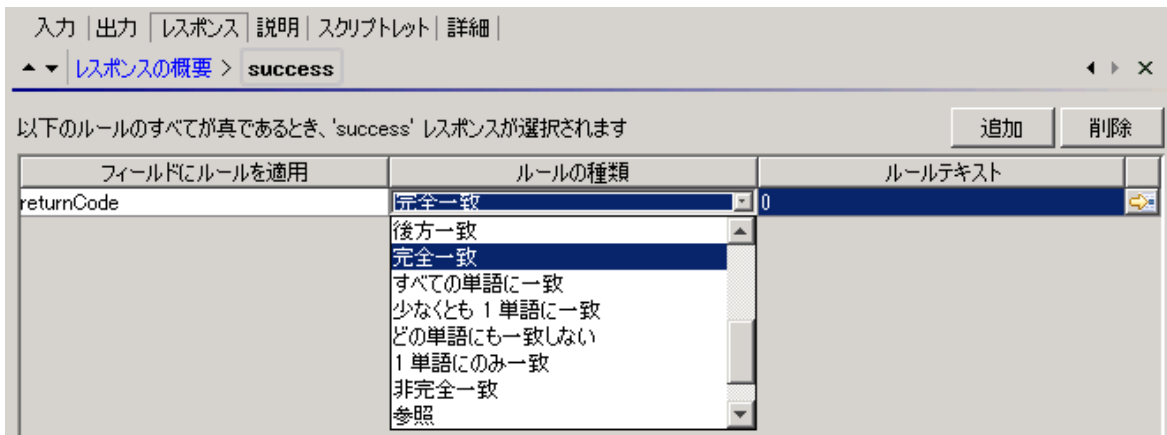


GUIアイテム	説明
レスポンスの追加	新しいレスポンス行を追加します。
レスポンスの削除	選択したレスポンス行を削除します。
	クリックして、選択したレスポンスをリスト内で上または下に移動します。



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"> • Success/resolved:  • Diagnosed:  • No action:  • Failure: 
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**Success**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。

入力 | 出力 | レスポンス | 説明 | スクリプトレット | 詳細

レスポンスの概要 > success > ルール (1 / 1)

ルール詳細

ルールの種類: 常に一致 ルールを適用する前に結果をフィルター

入力が値が含まれるかどうかを確認するテスト

大文字と小文字を区別しない:

テキスト: 0

入力 (一致が強調表示されます)

パターンが一致します クリア クイックコマンド

この領域に、照合させる入力文字を入力します。すべての一致（重複なし）が強調表示されます。各一致の最初の文字には下線も引かれるため、2つの一致が連続している場合でも容易に判断できます（例：強調表示強調表示）。

どちらかのテキスト領域を編集すると、一致の強調表示が即座に更新されます。入力された正規表現に構文上の問題がある場合は、入力中に一時的に問題となっているだけであっても、編集によって正しい表現に戻るまで、挿入カーソルが赤色で表示されます。

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」\(248ページ\)](#)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のかを決定します。

[「ステップの結果の設定」\(252ページ\)](#)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」\(261ページ\)](#)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」\(252ページ\)](#)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」\(261ページ\)](#)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](#)」(261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に行に移動します。
5.  をクリックします。
6. オペレーションを保存します。

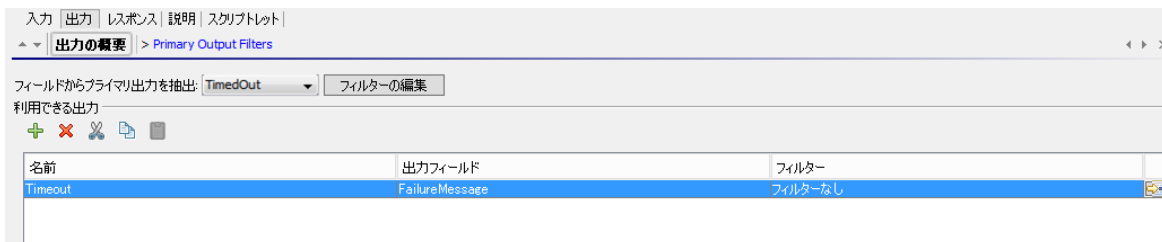
注:

- 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	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

(またはキーボードのCtrl+C)	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。出力名がすで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、受け渡す必要がある追加情報が含まれます。

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](#)」(279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、「[フロー内のサブフローの作成](#)」(290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](#)」(248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](#)」(261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](#)」(248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](#)」(261ページ)を参照してください。

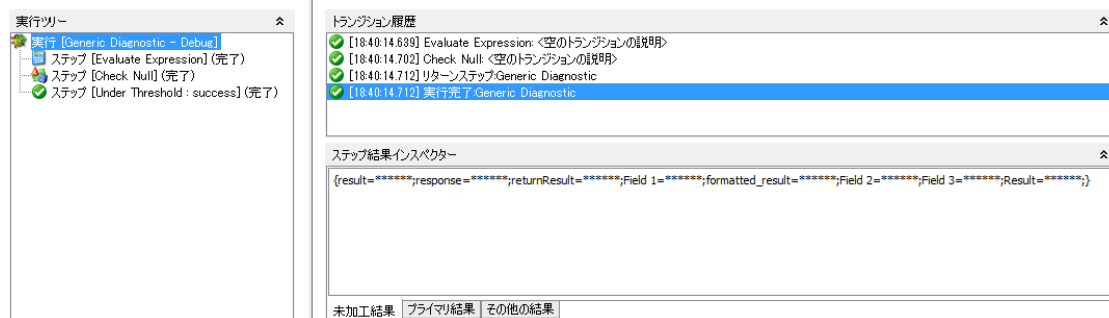
オペレーションの未加工の結果の非表示

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にのみであり、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せん。
- 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。

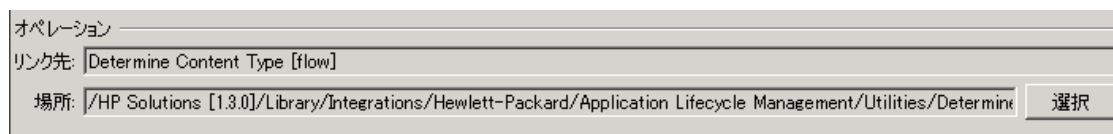
3. 出力の行の[ソース]列をクリックし、リストから目的のフィールドを選択します。
4. ステップを保存します。

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。

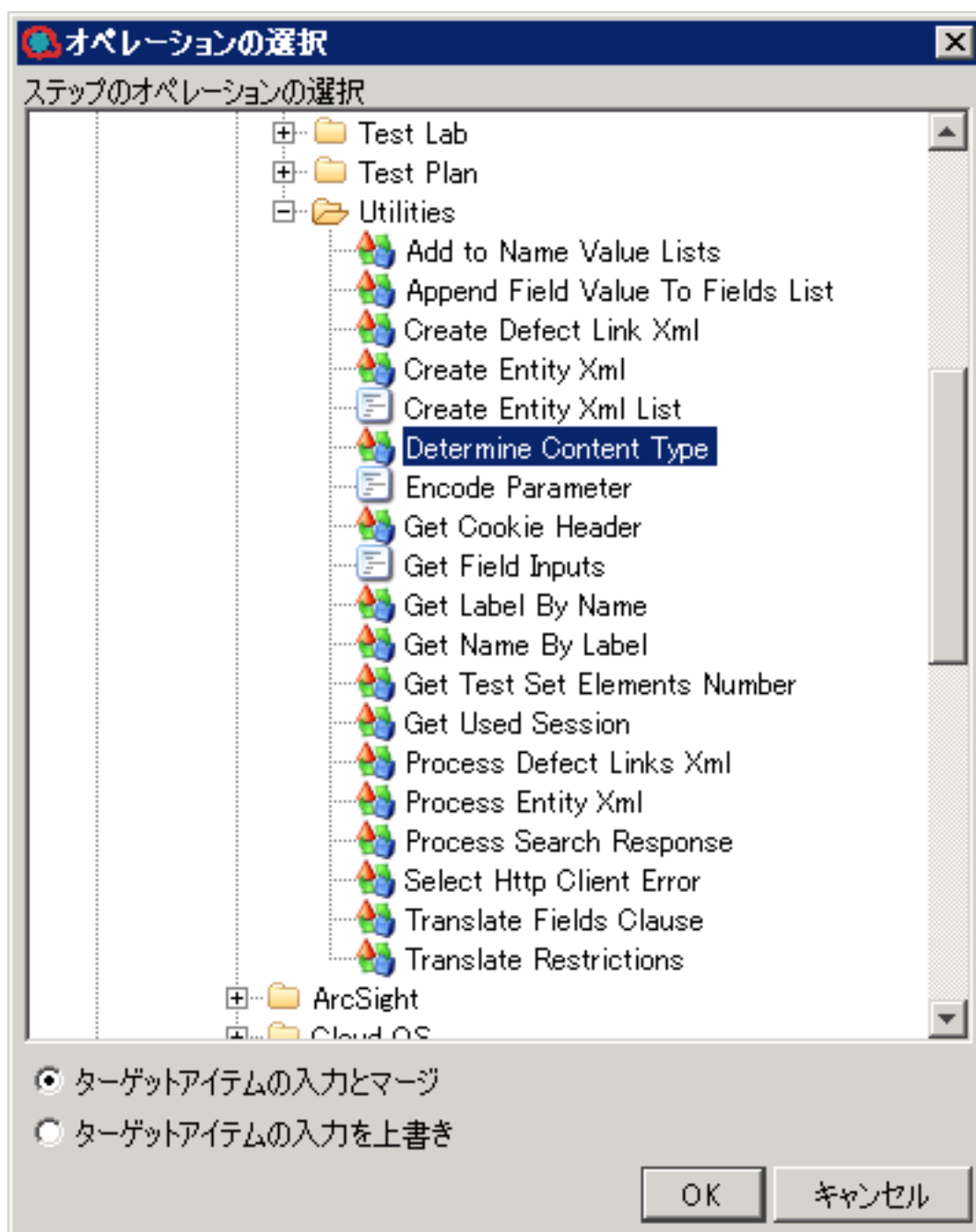
1. 作成ペインでステップをダブルクリックします。
2. ステップのインスペクターで[詳細]タブを選択します。

現在のオペレーションの場所と名前を表示できます。例：



The screenshot shows a UI element titled 'オペレーション' (Operation). It contains two input fields: 'リンク先:' (Link to) with the value 'Determine Content Type [flow]' and '場所:' (Location) with the value '/HP Solutions [1.3.0]/Library/Integrations/Hewlett-Packard/Application Lifecycle Management/Utilities/Determine'. A '選択' (Select) button is located to the right of the location field.

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。



4. 新しいオペレーションを参照します。
5. 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。

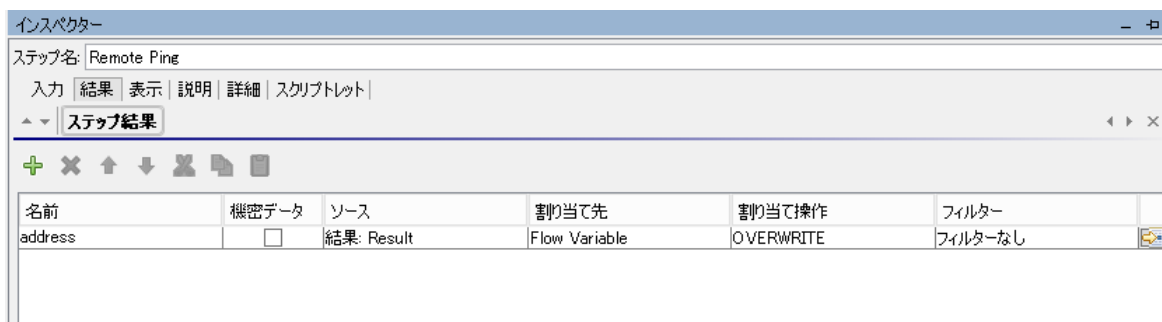
注: 入力名がすでに存在している場合は、元の入力値が残り、上書きされません。

6. 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。



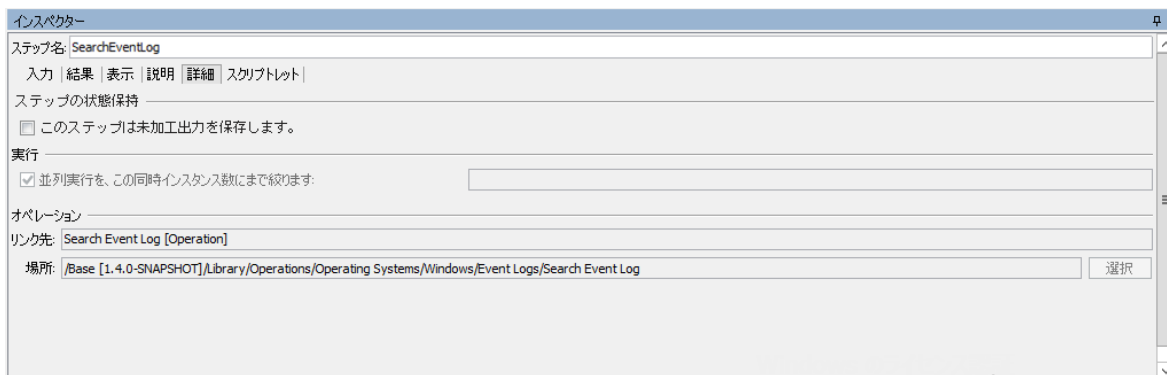
GUIアイテム	説明
名前	結果の名前を入力します。この名前は、フロー変数またはフロー出力フィールドの名前に使用されます。
機密データ	Centralユーザーインターフェース、およびStudioの内部デバッガーとリモートデバッガーでこの結果を非表示にします。結果アスタリスクで表示され、OOデータベースに保持されません。
結果の追加	新しい結果行を追加します。
結果の削除	選択した結果行を削除します。
ソース	結果のソースを選択します。
割り当て先	結果の値を格納する場所を選択します。 <ul style="list-style-type: none"> • フロー変数に値を格納するには、[Flow 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。
割り当て操作	[割り当て操作] リストから、適切な操作を選択します。 <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

	<p>値に置き換えます。</p> <ul style="list-style-type: none"> • 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。 <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>
	<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
このステップは未加工出力を保存します	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。
リンク先	ステップのベースとするソースオペレーションを表示します。
場所	ステップのベースとするソースオペレーションの場所を表示します。

選択

[オペレーションの選択] ダイアログボックスを開きます。このダイアログボックスで、ステップのベースとなるオペレーションに移動して、そのオペレーションを選択できます。



出力および結果のフィルタリング

オペレーションの出力や、ステップの結果の抽出や編集を行う際にフィルターを作成することができます。

たとえば、サーバーに対するpingオペレーションについて、最大、最小、平均の往復時間のみが必要だとします。未加工出力を3つの出力にフィルタリングして、これらの情報を未加工出力から分離して抽出できます。

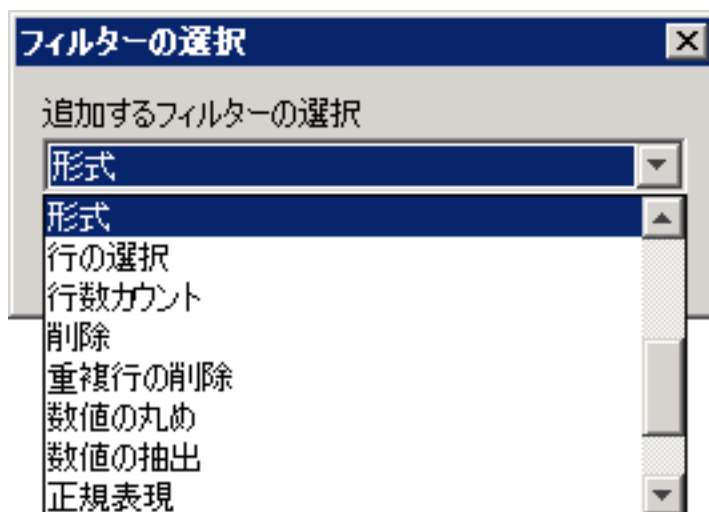
実行する操作

フィルターの作成

1. フィルターエディターを開きます。この手順は、フィルタリング内容によって異なります。
 - オペレーションのプライマリ出力に1つ以上のフィルターを作成するには、オペレーションの[プロパティ]シートを開き、[出力]タブをクリックして、[フィルターの編集] ボタンをクリックします。
 - オペレーションのセカンダリ出力にフィルターを作成するには、オペレーションの[プロパティ]シートを開き、[出力]タブをクリックして、出力行の端にある右向きの矢印をクリックします。
 - ステップのセカンダリ結果にフィルターを作成するには、作成ペインでステップをダブルクリックし、[結果]タブをクリックして、結果行の端にある右向きの矢印をクリックします。
2. フィルターエディターで、[追加] ボタンをクリックします。

注: 出力または結果には複数のフィルターを追加できます。

3. [フィルターの選択] リストで、フィルターの種類を選択します。



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、次のことを実行します。

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](#)」(170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
入力の先頭や末尾に解決された表現を追加したり、入力を解決された表現で置き換えることで、入力の形式を調整します

テキスト: ¥n

入力の配置: 末尾に追加

- 先頭に追加
- 末尾に追加
- 置換

1. [テキスト] ボックスに、結果に付加するテキスト、または結果と置換するテキストを入力します。
2. [入力の配置] リストで、次を行います。
 - テキストを既存のテキストの前に付加する場合は、[先頭に追加] を選択する
 - テキストを既存のテキストの後ろに付加する場合は、[末尾に追加] を選択する
 - 出力をテキストに置換するには、[置換] を選択する

行数カウント

[**行数カウント**] フィルターは、結果に含まれる行の総数を出力します。

正規表現

[**正規表現**] フィルターは、正規表現 (regex) を使用して未加工結果をフィルタリングします。

詳細: 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:

1. [**正規表現の種類**] リストで、[**Javaスタイル**] を選択します。ほかのスタイルは使用しないでください。推奨されていません。
2. [**正規表現の値**] ボックスに、正規表現を入力します。
3. [**フィルターのスタイル**] で、未加工結果にどのようにフィルターを適用するかに応じて、[**入力全体をフィルター**] または [**行単位でフィルター**] を選択します。
4. 正規表現の大文字小文字を区別しない場合は、[**大文字と小文字を区別しない**] を選択します。

正規表現の使用の詳細については、「[フローでの正規表現の使用](#)」(313ページ)を参照してください。

重複行の削除

このフィルターは同一の行を検出し、そのうち1つを残してすべて削除します。

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

置換

このフィルターは文字列の最初または最後のインスタンスか、すべてのインスタンスを別の文字列と置換します。

詳細: 置換
入力に含まれる文字のインスタンスを置き換えます

検索:

対象: すべて

置換:

大文字と小文字を区別しない:

1. [検索] ボックスに、検索して置換する対象の文字列を入力します。
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。

詳細: 数値の丸め
入力の数値として解釈できる場合に、その数値を丸めます。

小数点以下の桁数: 0

丸めの種類: 切り捨て
 丸め
 切り上げ

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

スクリプトレット

このフィルターは、作成したスクリプトレットを使用してデータをフィルタリングします。

詳細: スクリプトレット
JavaScript スクリプトレットを実行してデータをフィルターします

[テンプレートの挿入](#) [スクリプトの確認](#)

```
1 scriptletResult = "Timestamp:" + new java.util.Date() + "\n" + scrip
```

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](#)」(307ページ)を参照してください。

行の選択

このフィルターは未加工結果から抽出する行を指定します。

詳細: 行の選択
入力から 1 行を選択します

行の選択方法: 対象文字列が先頭にある最初の行

パラメーター:

行末の種類: 自動 Unix Windows

1. [行の選択方法] リストから、対象とする行の基準を選択します。
2. [パラメーター] ボックスに、対象文字列を含む文字列を入力します。
3. [行末の種類] グループで、次のいずれかを選択します。
 - フィルタリングしているテキストが、Unixオペレーティングシステム (LFで行を終了) で生成されている場合は、[Unix] を選択します。
 - フィルタリングしているテキストが、Windowsオペレーティングシステム (CR/LFで行を終了) で生成されている場合は、[Windows] を選択します。
 - フィルターで、どちらの種類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。

詳細: 範囲の選択
入力内の選択範囲を指定します (最初の文字は 0)

開始:

長さ:

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. 並べ替えの向きを指定するには、次の手順を実行します。
 - 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:

- [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
- [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、列の並び替えや、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: スペース 行の区切り記号: 改行

最初の行はヘッダー: 結果の最初の行を削除:

ソート基準列: -1 昇順:

選択行: 0 選択列: 0

選択幅: 1 選択高: 1

注: 行の番号設定は0ベース(0 [0] から始まる)、列の番号設定は1ベースです。

1. [列の区切り記号] リストでは、データを列に分割するための記号を選択します。
2. [行の区切り記号] リストでは、データを行に分割するための記号を選択します。

注: 複数の連続した空白は1つの空白と見なされるため、列の右にあると想定していたデータで列が占められることがあります。たとえば、列の区切り記号に空白を指定してdirコマンドラインコマンドの出力にこのフィルターを適用する場合です。

3. 最初の行を列ヘッダーとするには、[最初の行はヘッダー] を選択します。
4. 最初の行を削除するには、[結果の最初の行を削除] を選択します。
5. 列で並べ替えを行うには、[ソート基準列] ボックスに列番号を(1ベースで)入力します。

ヒント: 値 -1は列で並べ替えを行わないことを意味します。

6. 昇順を指定するには、[昇順] ボックスを選択します。
デフォルトの並べ替え順序は、降順です。
7. フィルタリングして抽出する行を選択するには、次の操作を行います。

- [選択行] ボックスに、行番号(0ベース)を入力します。

ヒント: -1を選択すると、データ内のすべての行が選択されます。

- [選択幅] ボックスに、抽出する行の列数を入力します。

ヒント: -1を選択すると、[選択列] に指定した列の右にある、データ内の残りの列がすべて選択されます。

8. フィルタリングして抽出する列を選択するには、次の操作を行います。

- [選択列] に列番号を入力します。

ヒント: -1を選択すると、データ内のすべての列が選択されます。

- [選択高] ボックスに、抽出する列の行数を入力します。

ヒント: -1を選択すると、[選択行] に指定した行の下にある、データ内の残りの行がすべて選択されます。

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

- 選択行: 0
- 選択高: 5
- 選択列: 2
- 選択幅: 3

XMLフィルター

XMLフィルターを使用すると、ステップ内のXMLを解析できます。XMLはHPE OOデフォルトコンテンツのXML処理オペレーションのいずれかにXMLを渡すフローを作成せずに、ステップの入力または結果からXMLを取得することができます。

オペレーションでのXMLフィルターの使用と、デフォルトコンテンツでのXML処理オペレーションの使用は、いくつかの点で異なります。

- オペレーション内でのタスクの完了と、タスクの完了までのフローのインフラストラクチャーの仕様が異なる
- オペレーション内のフィルターには、XML処理オペレーションにはない制限がいくつかあります。これらの制限については、該当のフィルターについて次のセクションで説明します。入力XMLのフィルタリングにフィルターとオペレーションのどちらを選ぶかは、XMLの取得方法によって決定します。

XMLフィルターは次のとおりです。

- XML属性の取得
- XML要素の取得
- XML要素値の取得
- XPathクエリ

XMLフィルターを説明するため、例では次のXML例を参照することとします。

```
<?xml version="1.0" encoding="utf-8"?>
<tickets>
<ticket id="1448" severity="3">
  <customer firstName="John" lastName="Doe">
    <volume>30000</volume>
    <company>myOrg</company>
    <position>CIO</position>
    <contactInfo>
      <email>jdoe@myorg.com</email>
```

```
<email>johnsSecondEmail@myorg.com</email>
<mobile>12065551212</mobile>
<description internal="1">Private contact info</description>
<description>Partial contact info</description>
</contactInfo>
<description>Our best customer</description>
</customer>
<details>
  <description>A simple Test xml</description>
  <comment user="john"> Initially raising ticket</comment>
  <comment user="frank"> Problem diagnosed, not a real issue</comment>
  <comment user="albert">ok, I'm going to close it.</comment>
  <state>Closed</state>
</details>
</ticket>
<ticket id="1886" severity="5">
  <customer firstName="Elaine" lastName="Benson">
    <volume>50000</volume>
    <company>herCompany</company>
    <position>CEO</position>
    <contactInfo>
      <email>ebenson@herco.com</email>
      <mobile>011445551212</mobile>
      <description internal="1">Private contact info</description>
      <description>Partial contact info</description>
    </contactInfo>
    <description>Our other best customer</description>
  </customer>
  <details>
    <description>datastream bug</description>
    <comment user="jack">Customer found bug.</comment>
    <comment user="elsbeth">It is a third-party supplier bug.</comment>
    <state>Closed</state>
  </details>
</ticket>
</tickets>
```

XML属性の取得

[XML属性の取得] フィルターでは、指定する属性の1つまたは複数のインスタンスそれぞれの値を抽出します。フィルターエディターで、属性への要素のパスを指定してフィルターが適用される属性のインスタンスを管理することができます。

属性の単一インスタンスに対する値を取得することも、複数のインスタンスに対する値を表形式で取得することもできます。この表では、列はコンマ区切りとなり、行は改行区切りとなります。

詳細: XML 属性の取得
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用したXMLの例を使用します)

[要素パス] ボックスに、`/ticket/details/comment`と入力します。

例: 特定のコメント (この例では2つ目のコメント) のユーザー名を取得するには、

1. **[要素パス]** ボックスに`/ticket/details/comment[2]`と入力します。
2. **[属性名]** ボックスに、`user`と入力します。
3. **[結果]** の横の**[1つの一致]** を選択します。

johnと出力されます。

例: 各コメントのユーザー名を検索するには

1. **[要素パス]** ボックスに`/ticket/details/comment`と入力します。
2. **[属性名]** ボックスに、`user`と入力します。

3. **[結果]** の横の **[テーブル]** を選択します。

次のように出力されます。

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

XML要素の取得

[XML要素の取得] フィルターで、次のいずれかの方法で記述することで、要素全体 (子要素、値、属性を含む) を抽出できます。

- 相対パスまたは絶対パスを指定する。
- 抽出する要素の子要素を指定する。子要素の特定の値を検索することもできます。
- 抽出する要素の属性を指定する。属性の特定の値を検索することもできます。

詳細: XML 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:			
子の名前:		値:	
属性名:		値:	

次の手順で、テキストボックスを1つまたは組み合わせて使用して詳細を入力できます。

1. **[要素パス]** ボックスに要素の絶対パスを入力します。

パス内の相対パスインジケータは、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的な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。

詳細: XML 要素値の取得
XML 文書から、指定されたパスに一致する最初の要素をフィルターし、その値を返します。

要素パス:

[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`

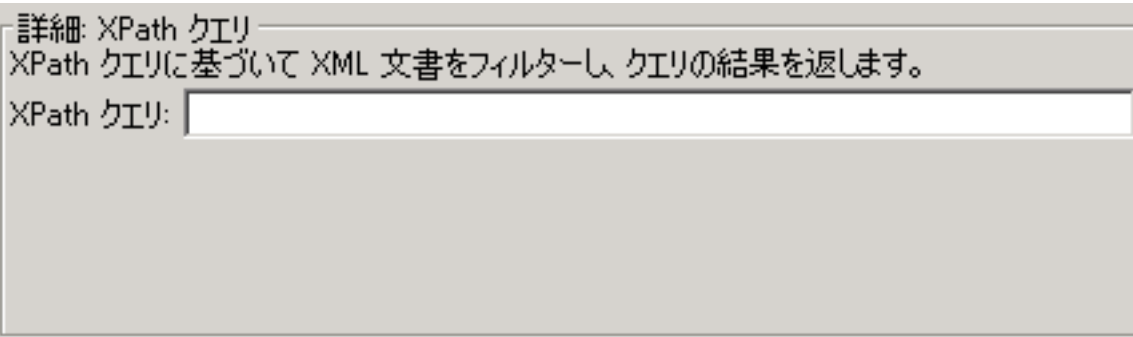
```
johnsSecondEmail@myorg.com
```

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。

次のように入力されます。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設定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。

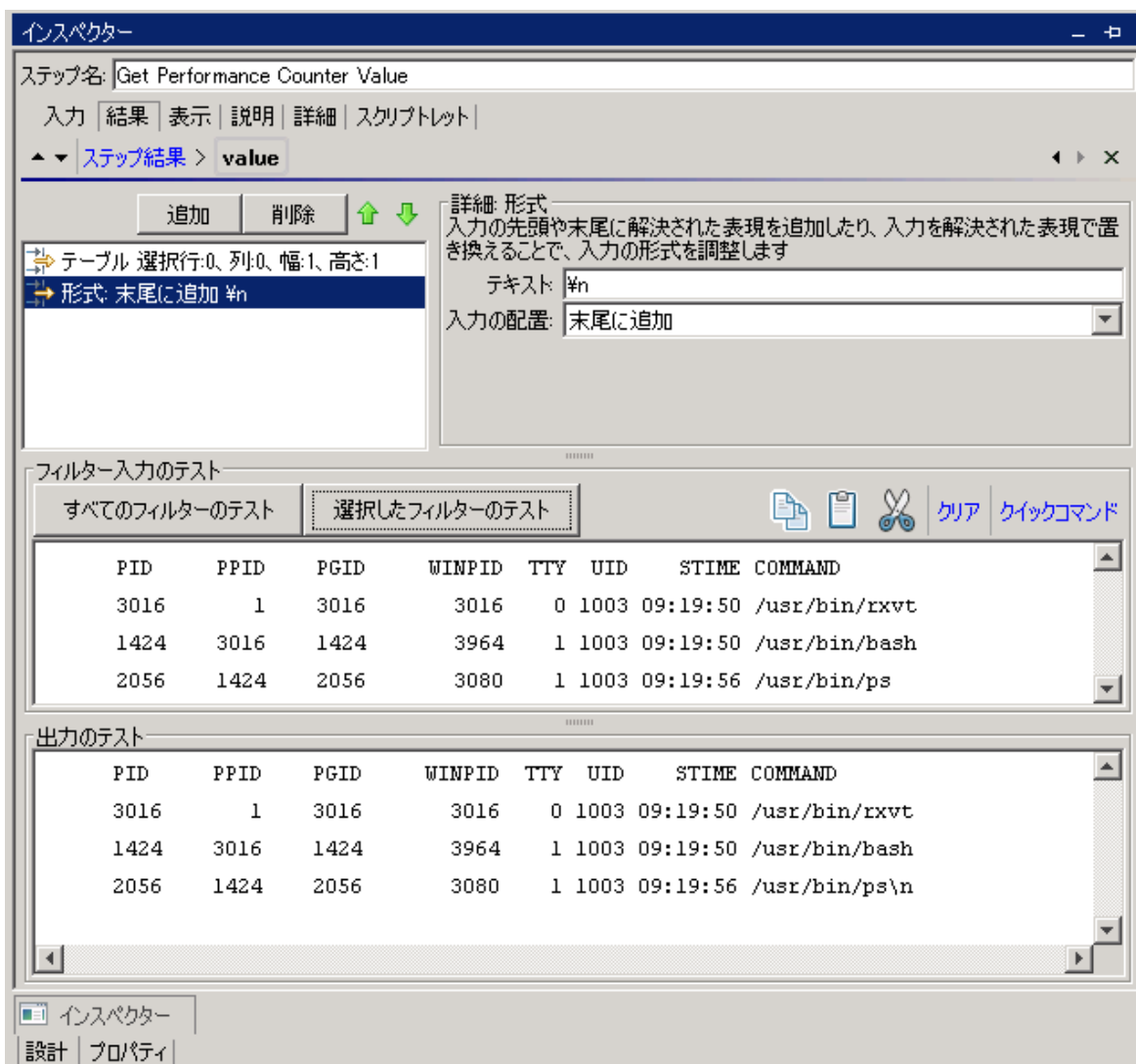
```
/tickets/ticket/customer[volume>40000]
```

<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、同じ名前を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、その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](#)」(202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](#)」(247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](#)」(261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...	-		
-- 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: 使用のリストを上に移すには、[前へ]  ボタンを使用し、使用のリストを下に移すには、[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](#)」(324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。






名前	#	→	👤
altpass	8	✓	✓
→ フロー入力	8	✓	✓
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...	-		
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	8	✓	✓

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が発生した場合、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、チェックされません。

[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。

- **Resolved**  – 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。
- **Diagnosed**  – 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。
- **No Action Taken**  – 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、**No Action Taken**  ではなく **Resolved**  を返します。

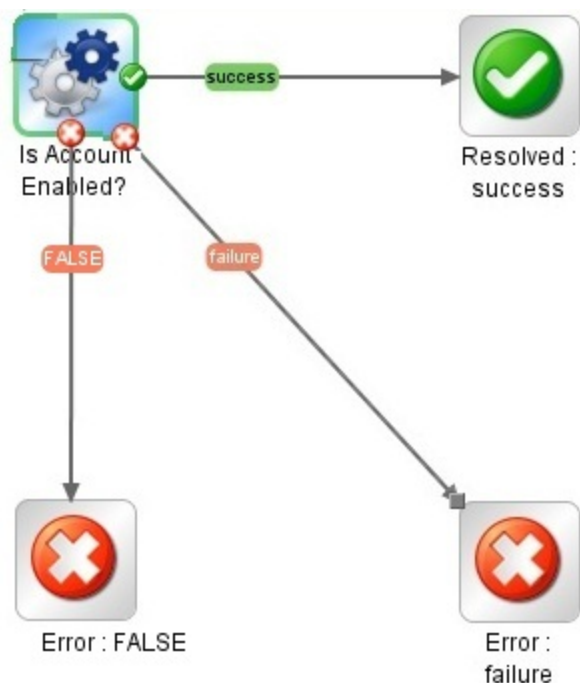
- **Error**  – フローを最後まで実行できない場合に使用されます。たとえば、入力が正しくない場合や、システムに到達できない場合や、フローに問題がある場合に使用されます。

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、**Error: failure** 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、**Error: failure** 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、**Error: threshold not met** などの新しいレスポンスを **Error: failure** 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。

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(**Error: failure**や**Error: threshold not met**)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、「Error: threshold not met」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
Success	[Success] リターンステップをフローにドラッグできます。
Diagnosed	[Diagnosed] リターンステップをフローにドラッグできます。
No Action Taken	[No Action Taken] リターンステップをフローにドラッグできません。
Failure	[Failure] 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。

ボタン	説明
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](#)」(180ページ)を参照してください。

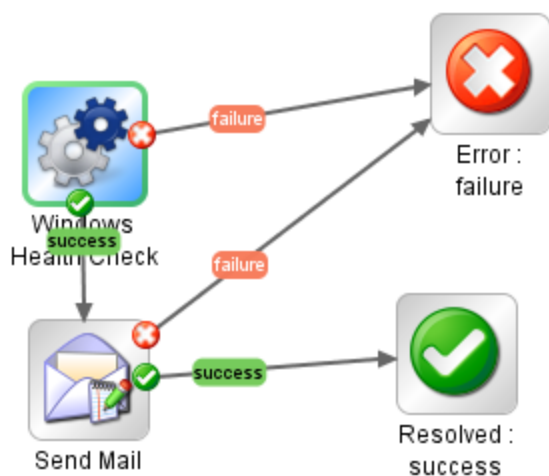
フローの作成時、メモリを無制限に増大させるようなフローは作成しないでください。フローが休止し、一部のタスクを実行した後にまた休止する、といった無限のループを実行するフローなどは作成しないでください。この場合、システムでメモリがなくなるまで実行履歴が増大します。

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のあるデータを生成します。フロー内で作成したフロー変数は、そのフロー外では参照できません。ただし、サブフローの結果をフロー出力フィール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。

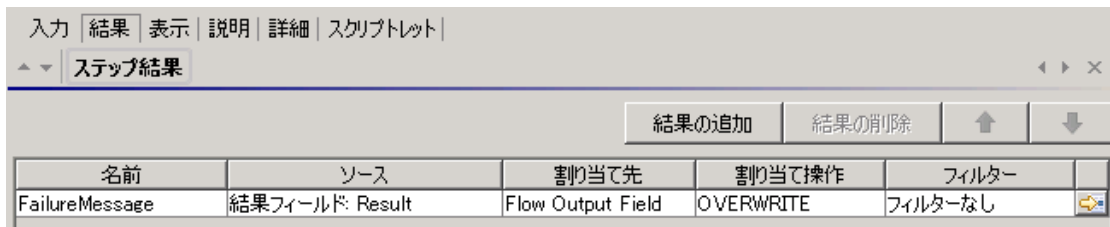
実行する操作

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](#)」(252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」(261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。

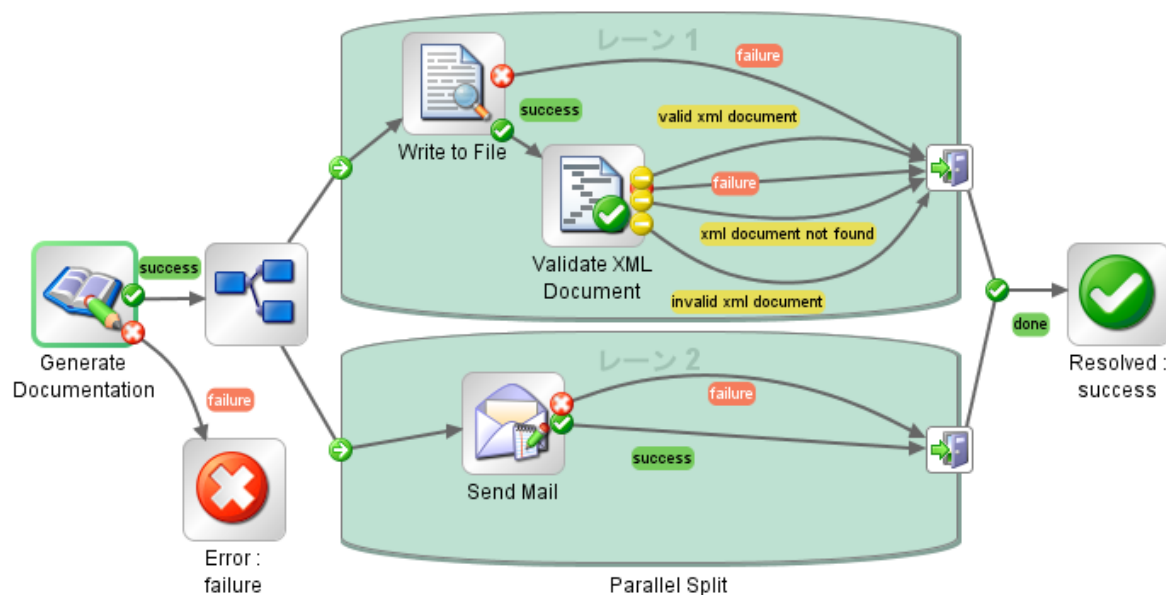
並列分岐ステップのフローの作成

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、担当者は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pons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pons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

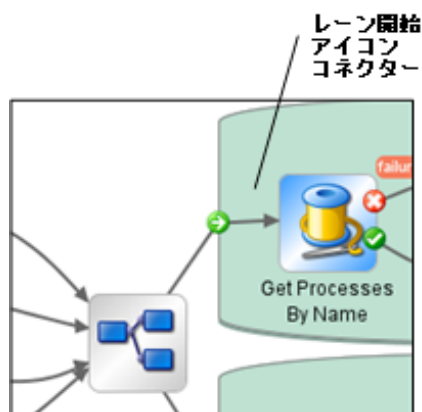
デフォルトでは、レーンの名前はレーン1、レーン2などとなります。

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](#)」(340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。

きわめて多くのターゲットに対し同時にマルチインスタンスのステップを実行したために、インフラストラクチャーの処理が低下した場合、同時に処理可能な値の数を制限することで、マルチインスタンスのステップを絞ることができます。絞りのサイズに定数(整数)を指定するか、その値を $\$(variable)$ から取得することができます。

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

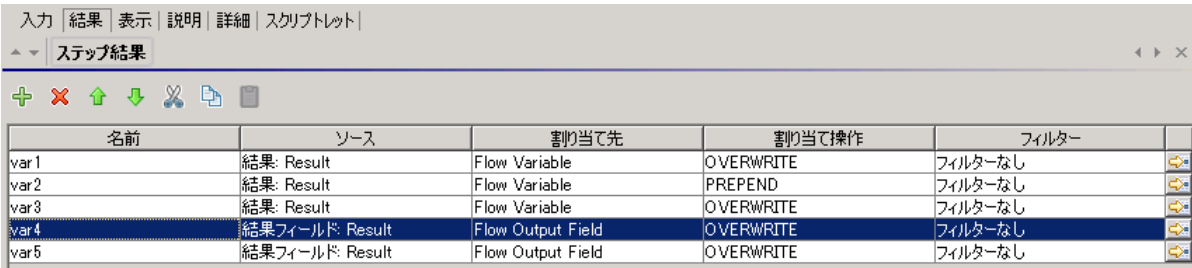
結果を使用したデータの保存

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)
```

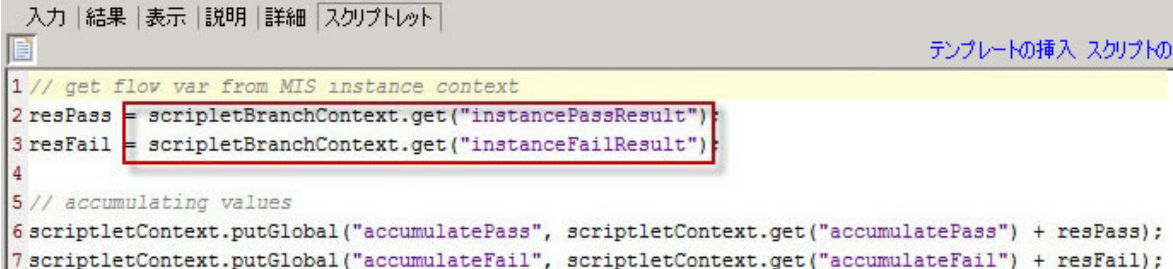
```
var4 = NULL (Instance2では、この変数の値は存在せず、アクションが上書きされるため)
```

```
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。



```
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);
```

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。

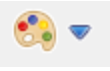
スクリプトレットの詳細については、「[フロー内でのスクリプトレットの使用](#)」(307ページ)を参照してください。

アップグレード後のマージ

前のバージョンのHPE OOからアップグレードすると、**[複数インスタンスの切り替え]** オプションを使って作成された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。

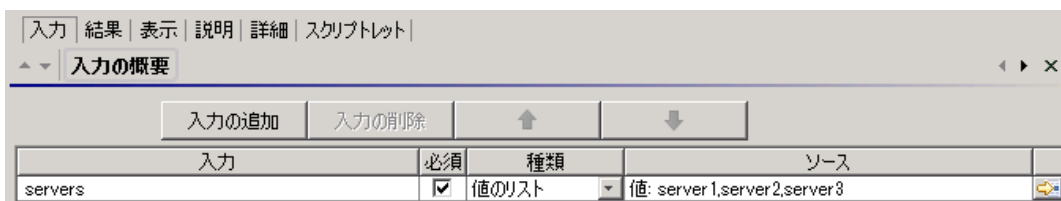
レスポンスステップを複数のインスタンスレーンに追加することはできません。

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。

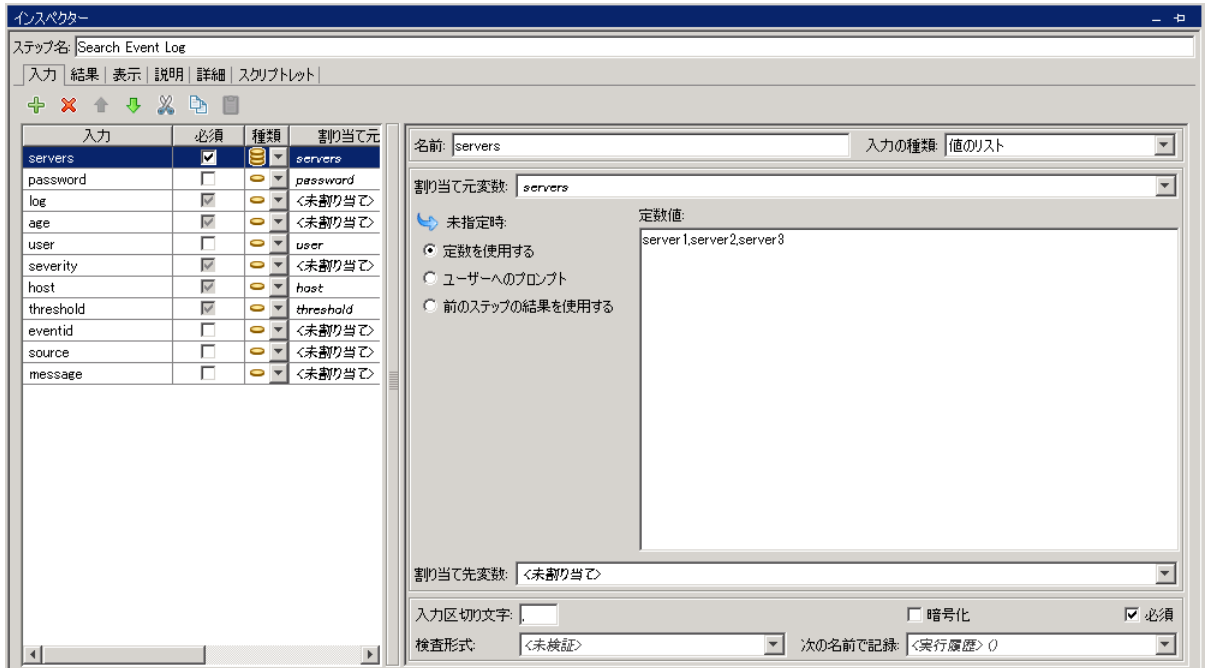


- d. 行の端にある右向き矢印 (👉) をクリックして、その行の入力エディターを開きます。
- e. [入力区切り記号] ボックスに、区切り記号 (リスト内の要素を区切る文字 (複数可)) を入力します。

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](#)」(213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。

- a. [レーン開始] アイコン  を複数のインスタンスレーンの最初のステップに接続します。
- b. 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。
- c. レーン内の最後のステップのすべてのレスポンス行を [レーン - エンド] アイコン  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に、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。

- 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。
- 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。

```
1 var outcome = scriptletBranchContext.get("outcome");  
2  
3 scriptletContext.putGlobal("outcome", scriptletContext.get("outcome") + outcome);  
4
```

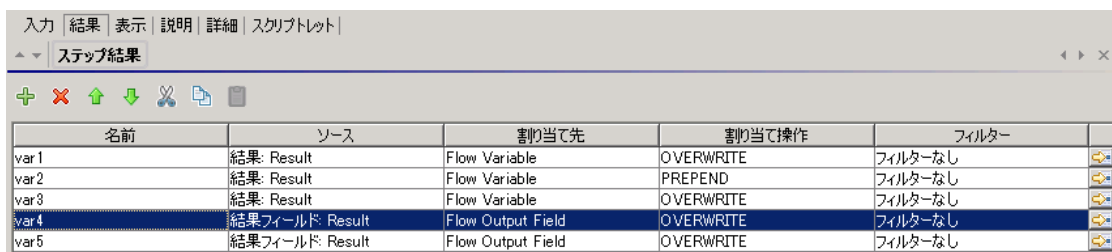
9. マルチインスタンスステップを残りのフローに接続します。

- 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。
- 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。

- 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。
- 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。
- 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
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

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。

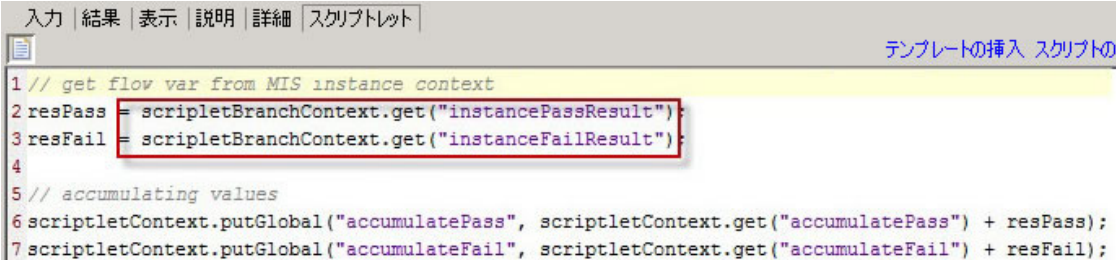
たとえば、すべてのインスタンスを実行するのにかかる時間を計算する場合、[ADD] を選択します。マルチインスタンスステップでチェックされたすべてのサーバーのリストを収集する場合は、[APPEND] を選択します。
7. ステップを保存します。マルチインスタンスステップの実行が終了した後で、作成したフロー変数を残りのフローに使用できるようになります。

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:



```
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OO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。


OO 9.xからアップグレードする際の注意

OO 10.xでは、絞りの動作方法に違いがあります。

- 9.xでは、実行される同時スレッドの数が一定に維持されていました。たとえば、100個のアイテムが存在して絞りを10に設定した場合、マルチインスタンスステップでは常に実行の数が10に維持されます。1つが終了すると、別の1つが開始されます。
- 10.xでは、100のうち10が開始され、その10がすべて終了してから、次の10が開始されます。

パッチ適用をスケジュールする際には、この点を考慮してください。


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いということですが、各インスタンスが終了するまでにかかる時間を調べることができます。

詳細については、「[複雑なフローのデバッグ](#)」(340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。

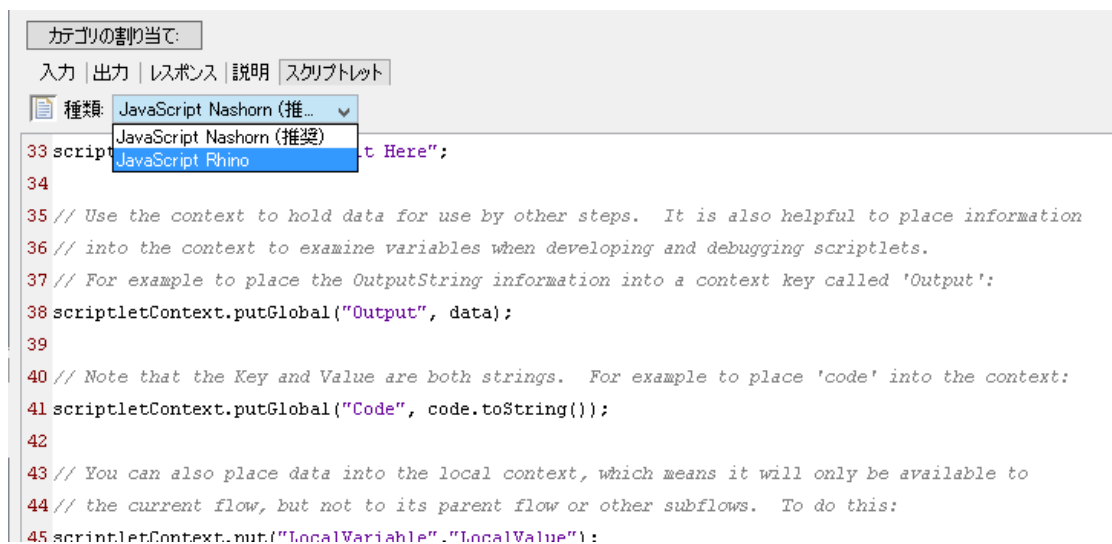
スクリプトレットを書くのに役立つリソース

- 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。


既存のスクリプトレットを使用する

1. [プロパティ] シートまたはステップのインスペクターを開きます。
 - オペレーションにスクリプトレットを追加するには、[プロジェクト] ペインでオペレーションを右クリックし、[プロパティ] を選択します。
 - フローにスクリプトレットを追加するには、[プロジェクト] ペインでフローを右クリックし、[プロパティ] を選択します。
 - ステップにスクリプトレットを追加するには、作成 ペインでステップをダブルクリックします。
2. [スクリプトレット] タブを選択します。
3. 既存のスクリプトレットを別のウィンドウで開きます。
 - 「**Configuration\Scriptlets**」フォルダーのスクリプトレットをダブルクリックします。
 - スクリプトレットが含まれているオペレーションを開きます (たとえば、「**Operations\Operating Systems\Linux\Red Hat**」フォルダーのオペレーション)。
4. スクリプトレットのテキストをコピーして、オペレーション、フロー、またはステップの [スクリプトレット] テキストボックスに貼り付けます。
5. 必要に応じて、スクリプトレットを変更します。

6. **[スクリプトの確認]** をクリックしてエラーがないか確認します。
7. 保存します。

スクリプトレットでステップまたはフローの結果をフィルタリングする

スクリプトレットを使用して、ステップまたはフローの結果をフィルタリングできます。

1. 作成ペインでステップをダブルクリックします。
2. **[結果]** タブを選択し、フィルタリングする結果を選択します。
3. 結果の行の端にある右向き矢印  をクリックして、フィルターエディターを開きます。
4. フィルターエディターで、**[追加]** ボタンをクリックします。
5. **[フィルターの選択]** リストで、**[スクリプトレット]** を選択します。
6. 次のいずれかの方法で、データをフィルタリングするスクリプトレットを作成します。
 - **[テンプレートの挿入]** をクリックし、スクリプトレットテンプレートをベースとして使用します。
 - 別のオペレーションまたは「**Configuration\Scriptlets**」フォルダーの既存のスクリプトレットのテキストをコピーして貼り付けます。

フィルターの作成の詳細については、「[出力および結果のフィルタリング](#)」(261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](#)」(352ページ)を参照してください。

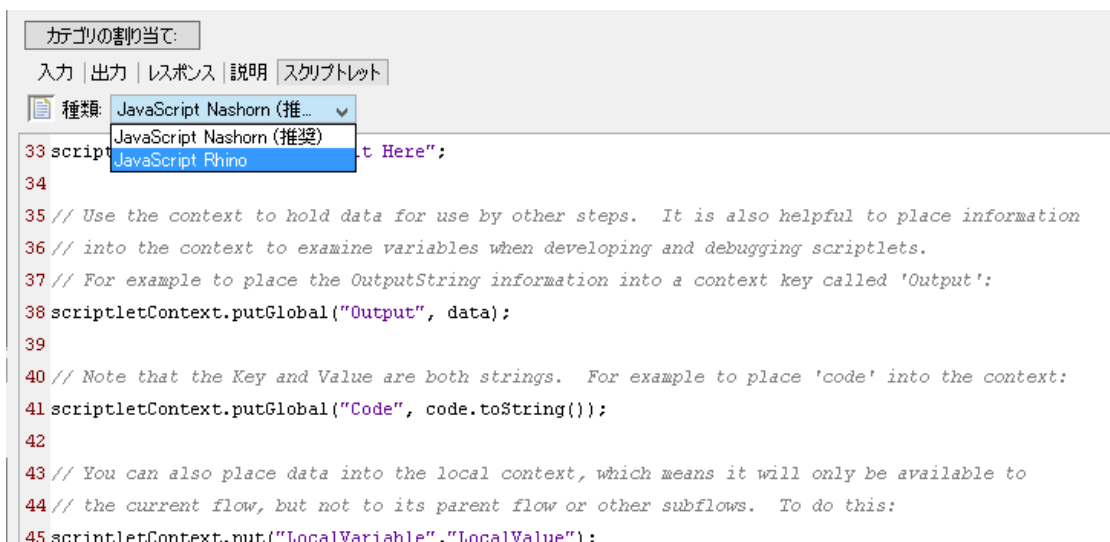
8. フィルターをテストして作業結果を保存します。

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)
JavaScript Nashorn (推奨)
JavaScript Rhino
33 scriptletContext.putGlobal("code", code);
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。

入力 | 出力 | レスポンス | 説明 | **スクリプトレット**

共有スクリプトレットへの参照: [/MyProject1/Configuration/Scriptlets/divide by 1k](#)

カスタムスクリプトレットに切り替え

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

1. [プロパティ] シートまたはステップのインスペクターの[スクリプトレット] タブで、保存するスクリプトレットを開きます。
2. [スクリプトレット] アイコンを [プロジェクト] ペインの「Configuration\Scriptlets」フォルダーにドラッグします。
3. スクリプトレットの名前を入力します。

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」フォルダーでスクリプトレットをダブルクリックして開く場合でも同じです。


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.put("Output", "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");
```

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。

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
a b	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」(261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 =" とその後続く任意の数字を検索するよう、HPE OOI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、HPE OOI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。

7. **[選択したフィルターのテスト]** をクリックしてフィルターをテストします。
8. フィルターを保存します。

複数の正規表現を組み合わせて値を抽出する

複数の正規表現を組み合わせて、フィルターで値を抽出できます。

たとえば、Unixのpsコマンドの出力でpsの時間を抽出するには、2つの正規表現が必要です。1つ目で出力をpsの行まで絞り込んで、2つ目で時間を抽出します。

F	S	UID	PID	PPID	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。

HPE 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、HPE 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。

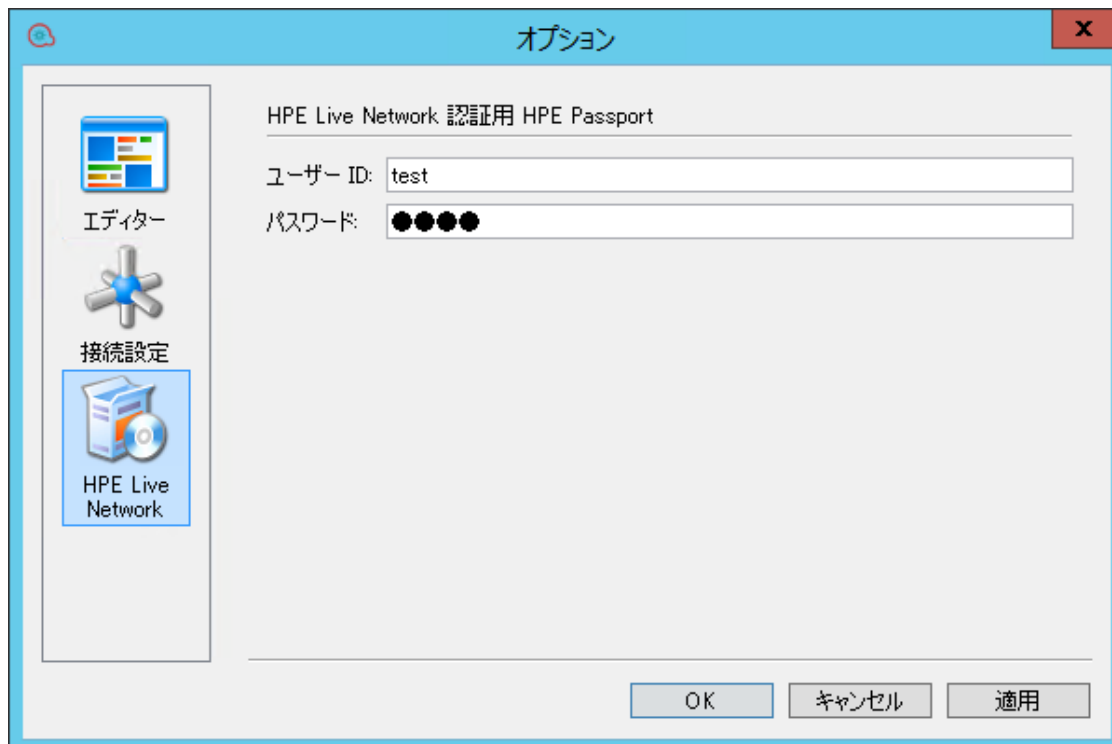
注: StudioでHPLN検索機能を使用するには、コンテンツパックの.jarファイルとともに.jsonファイルをHPLNにパブリッシュする必要があります。詳細については、「[HPLN Index Generatorツールの実行](#)」(321ページ)を参照してください。

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]>[HPE Live Network]を選択します。
2. [HPE Live Network設定]ダイアログボックスに、HPE 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。

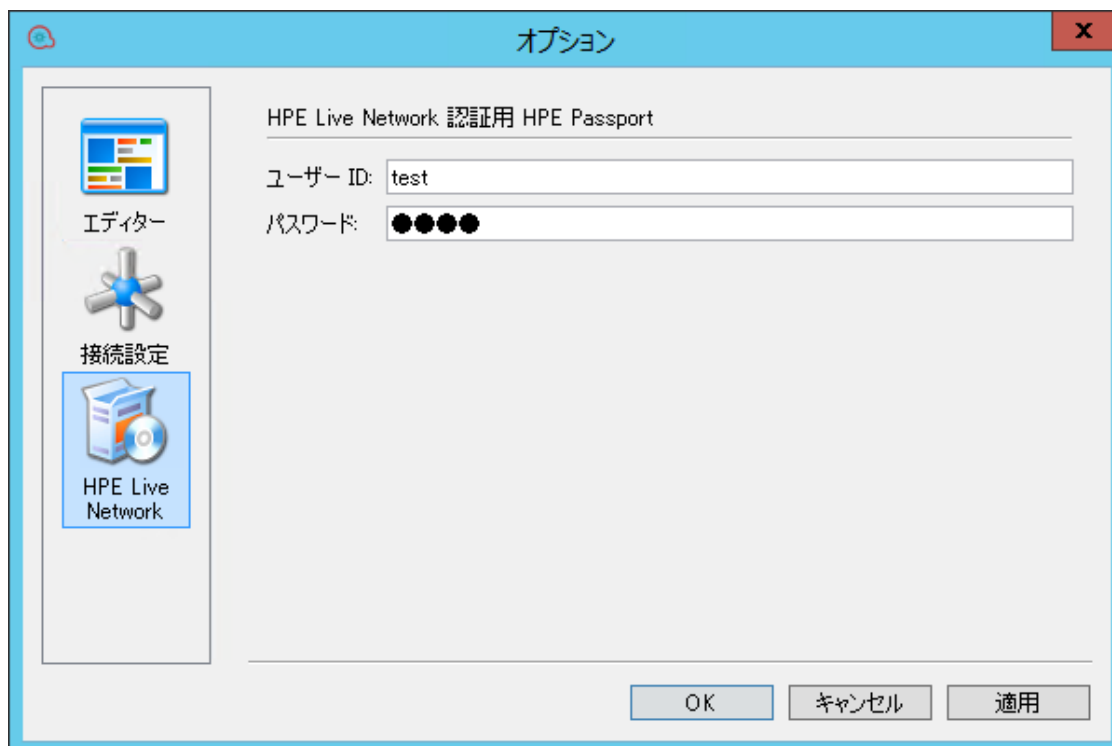


3. [適用] をクリックし、[OK] をクリックします。

StudioからのHPE Live Networkの検索

1. [構成] メニューから [オプション] > [HPE Live Network] を選択します。
2. [HPE Live Network設定] ダイアログボックスに、HPE Live NetworkのユーザーIDとパスワードを入力します。

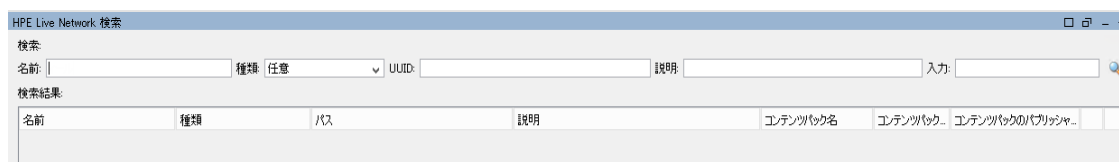
注: このダイアログボックスに入力したユーザーIDとパスワードにより、アクセス許可が決まります。また、検索結果にも影響します。



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。



HPE 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、HPE 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. OO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-1.7.0.json。

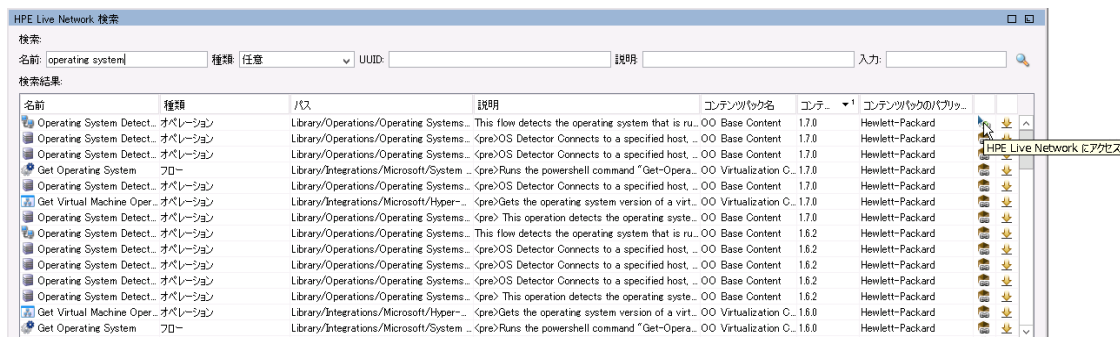
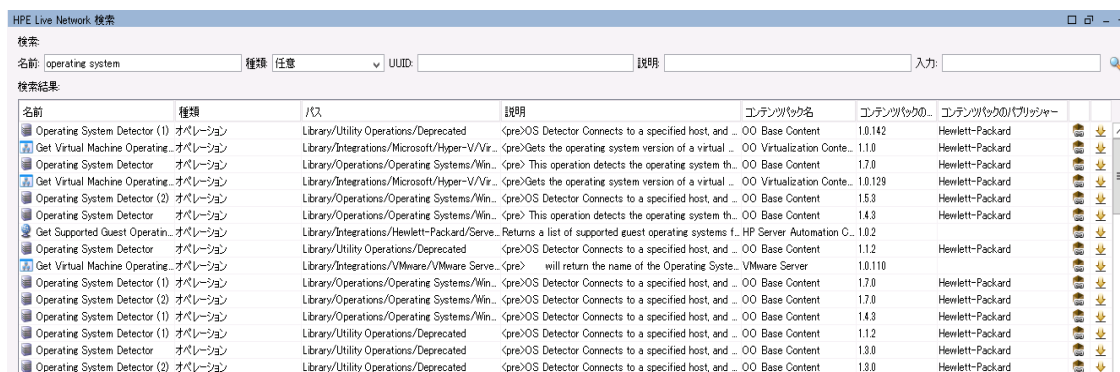
参考資料

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。

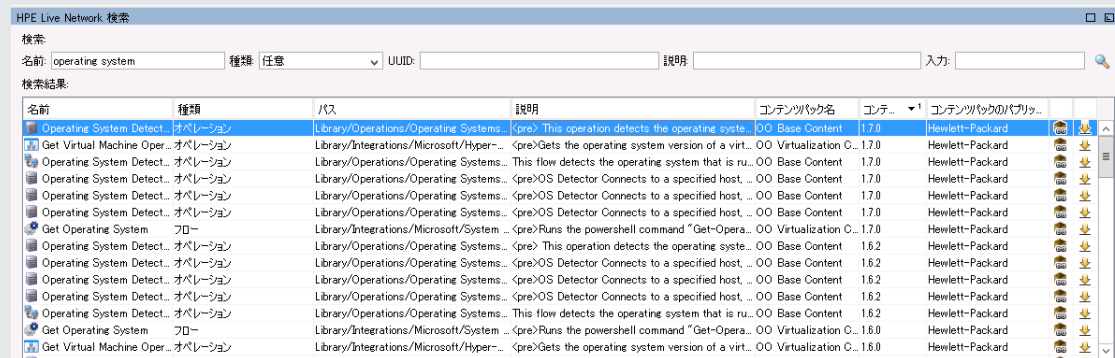
行を右クリックすると、検索結果の完全な説明を表示できます。

HPE 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す。



コンテンツの検査

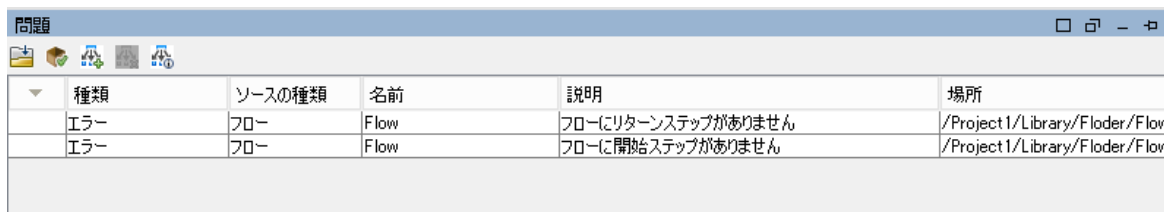
コンテンツをリリースする前に、プロジェクトのフローをテストし検査することが重要です。Studioでは、これを行う際に役立つ次のツールを用意しています。

- **[問題] ペイン**。問題が、発生した場所と内容説明とともに一覧され、この問題を解決するための指針となります
- **デバッガー**。フローで見つかったエラーや予期しない動作の原因を突き止めるのに役立ちます

[問題] ペインでのフローの検査

フローを実行するには、フローそのもの、そのオペレーション、構成アイテム、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、構成アイテム、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	/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

さまざまな種類の問題が一覧されます。以下にその一部を示します。

- フローに開始またはリターンステップがない
- オペレーションが見つからない
- 複数のオブジェクトのUUIDが同じである
- <変数>が存在しない選択リストを参照している
- トランジションソースステップに、リンクされたオペレーションがない

有効なフローの要件

フローを有効にするには次の条件を満たす必要があります。

- 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。

プロジェクトおよびプロジェクト依存関係内のすべてのアイテムの検査

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 [検証] 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。

重複するUUIDの解決

1. [問題] タブをクリックし、[問題] ペインを表示します。
2. 重複するUUIDエラーが含まれる行をダブルクリックします。

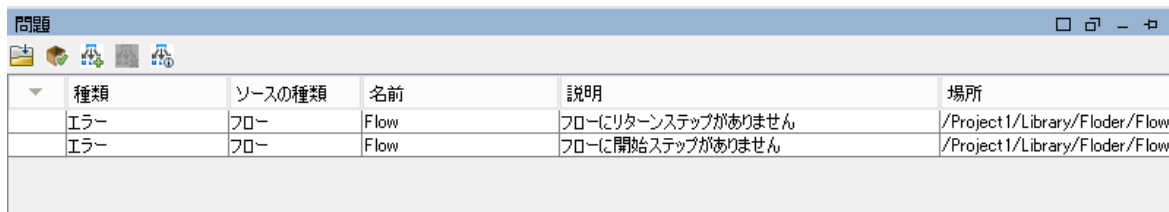
重複するUUIDを持つアイテムが開きます。次に、そのアイテムを削除して、その後は元のアイテムを使用します。

両方のアイテムを残す場合は、[コピー貼り付け] または [重複] コマンドを使用して、異なるUUIDを持つアイテムのコピーを保存してから、コピー元のアイテムを削除します。

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。



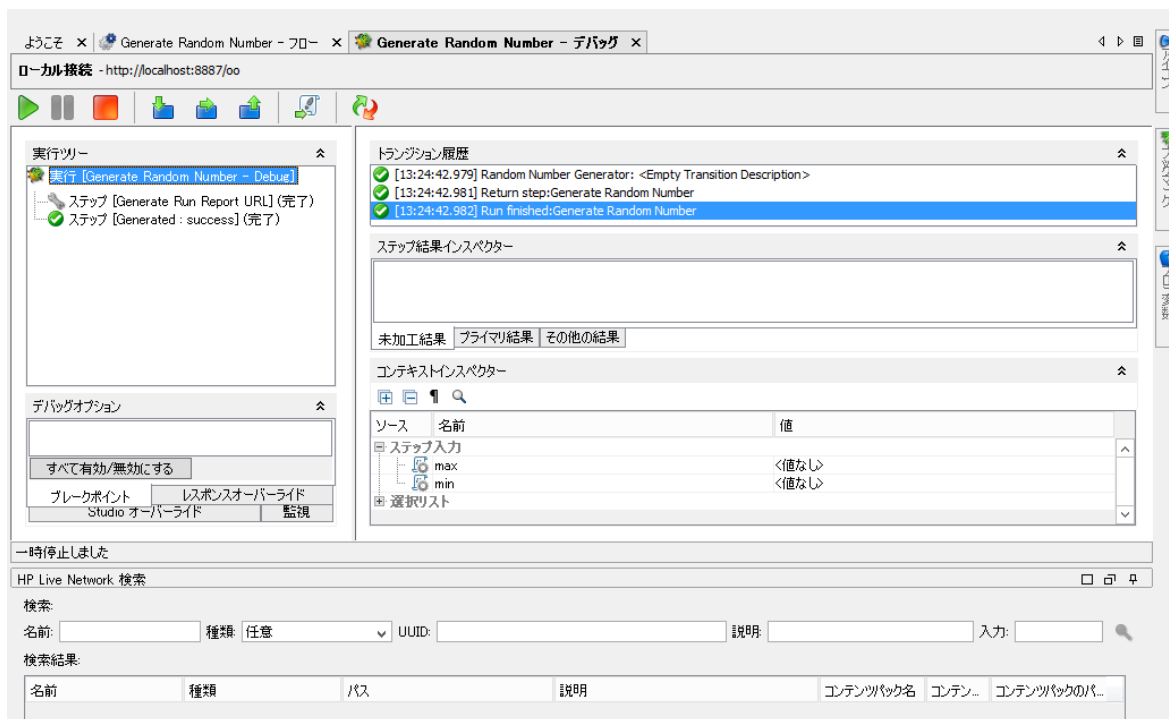
種類	ソースの種類	名前	説明	場所
エラー	フロー	Flow	フローにリターンステップがありません	/Project1/Library/Floder/Flow
エラー	フロー	Flow	フローに開始ステップがありません	/Project1/Library/Floder/Flow

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。

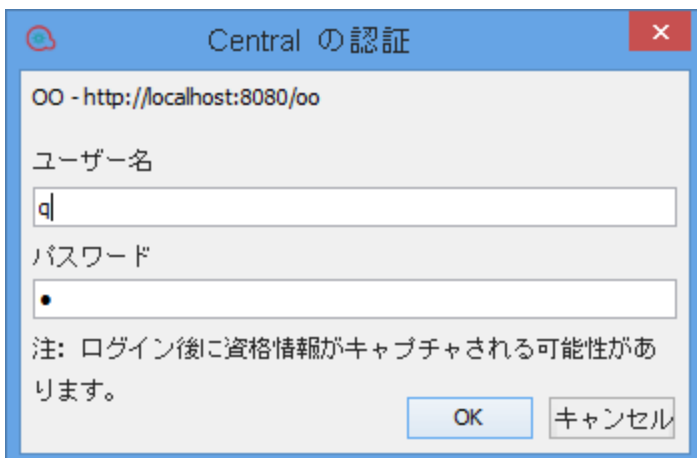
デバッガーの各ステップから、フロー内の関連するステップに直接ジャンプすることができます。ジャンプすると、フローエディターが開きます。

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。

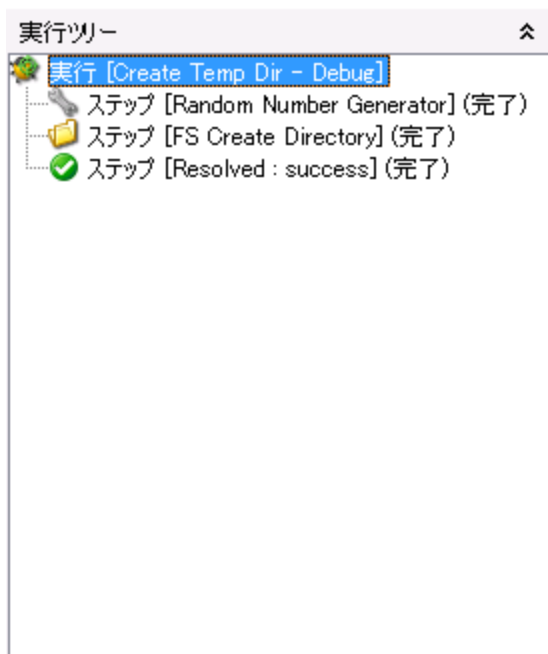
実行する操作

フローのデバッグ

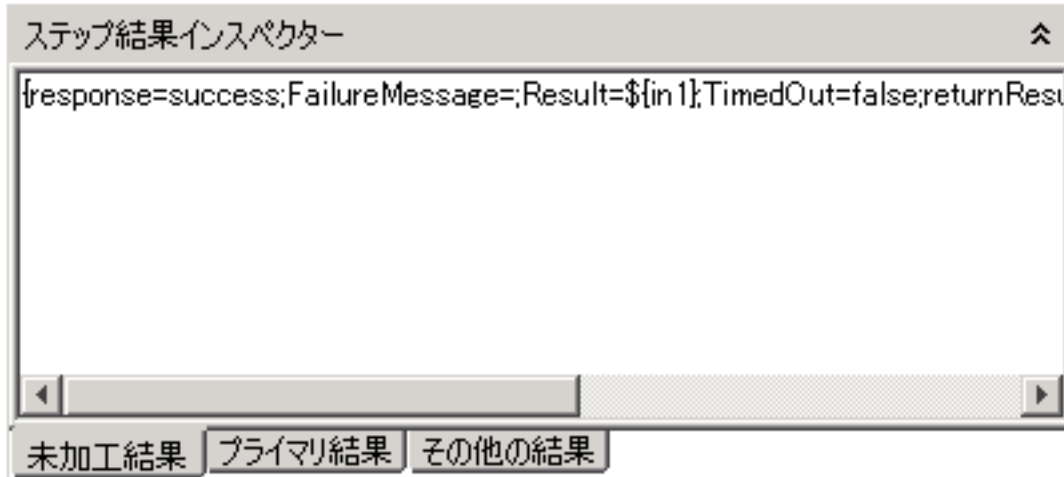
1. [プロジェクト] ペインでフローを右クリックし、[デバッグ] をクリックします。

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。

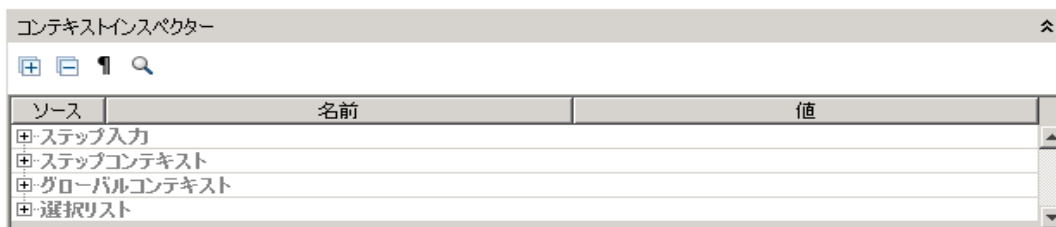
2. フローを最後まで実行するには、デバッガーのツールバーの[再生]  ボタンをクリックします。
3. 完了したステップの情報を確認するには、[実行ツリー] ペインで目的のステップをクリックします。




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。

- 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。

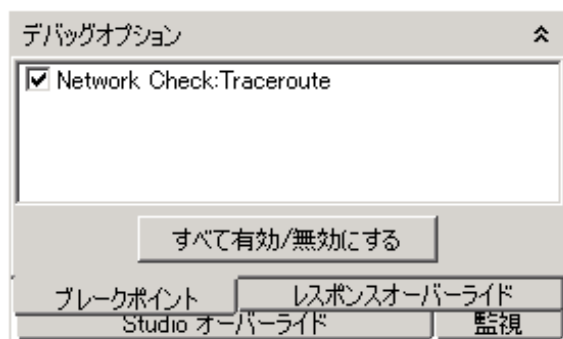
- 作成ペインでフローを開き、ブレイクポイントを設定するステップを右クリックします。
- [デバッグ] > [ブレイクポイントの設定] を選択します。

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。



- 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。



- 次のいずれかを実行します。

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

単一ステップのデバッグ実行でレスポンスをオーバーライドする

レスポンスのオーバーライドにより、オペレーションが失敗した場合でも、選択したレスポンスを強制的に指定できます。

レスポンスをオーバーライドすると、デバッガーを終了して入力値を変更せずに、フローの特定のパスをテストできます。

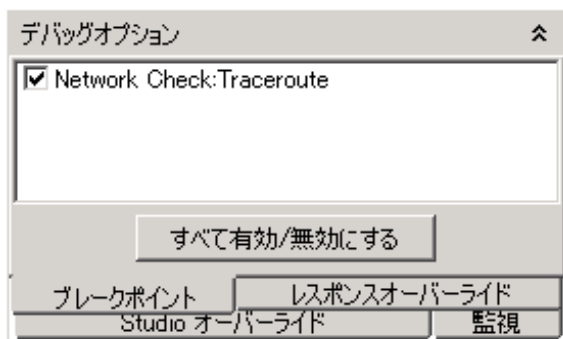
たとえば、フローのステップについて必要な情報がない場合に、このステップが失敗したかどうかに関わらず、フローの残りのステップについてテストが必要な場合があります。ユーザーが介入しない場合に発生するfailureレスポンスではなく、目的のレスポンスとランジションをたどるように強制的に実行を続けることができます。

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。

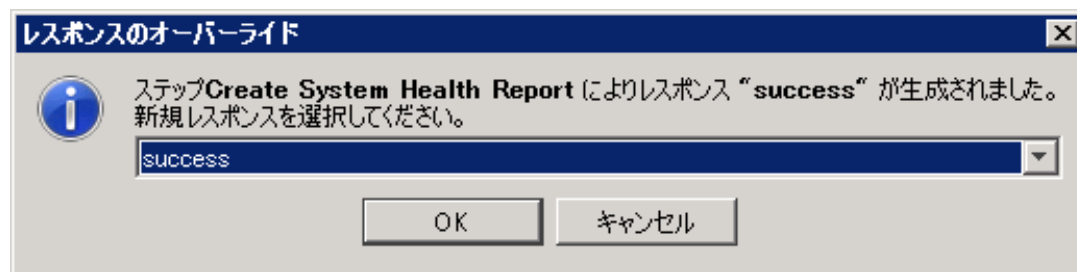
3. デバッガーでフローを開きます。

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。

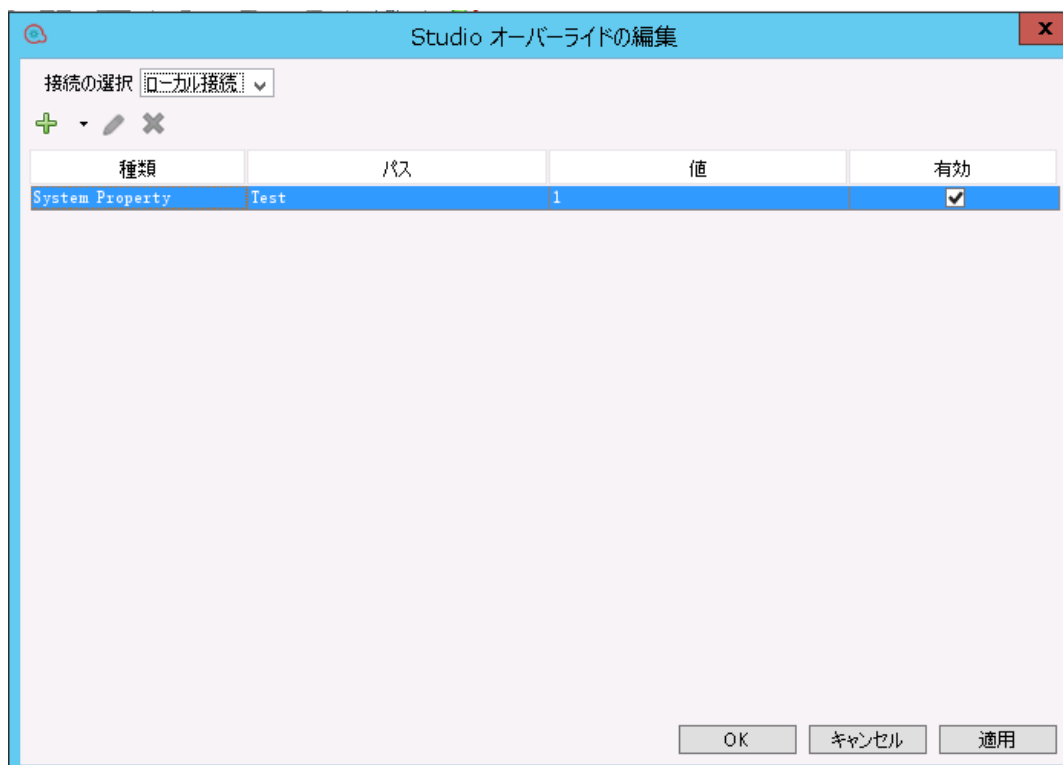
Studioでのオーバーライドの設定

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、次の手順を実行します。

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。

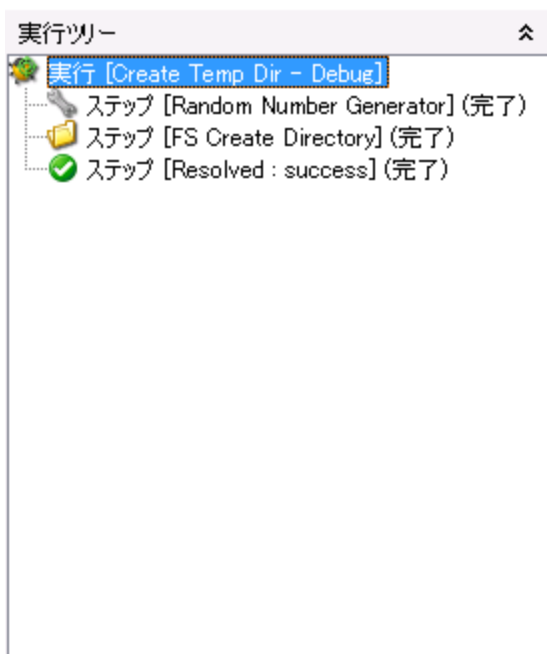
-  システムプロパティまたはシステムアカウントのオーバーライドを追加します。
 -  選択したシステムプロパティを編集します。
 -  選択したシステムプロパティを削除します。
4. 変更を適用するには、**[適用]**をクリックし、完了したら**[OK]**をクリックします。

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

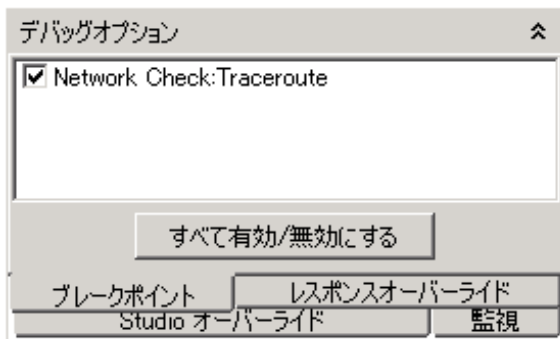
[実行ツリー] ペイン



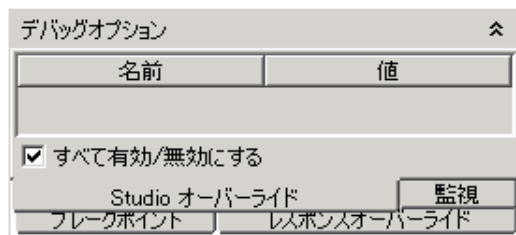
[実行ツリー] ペインには、フローのサブフロー内のステップも含めて、実行する各ステップが表示されます。

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



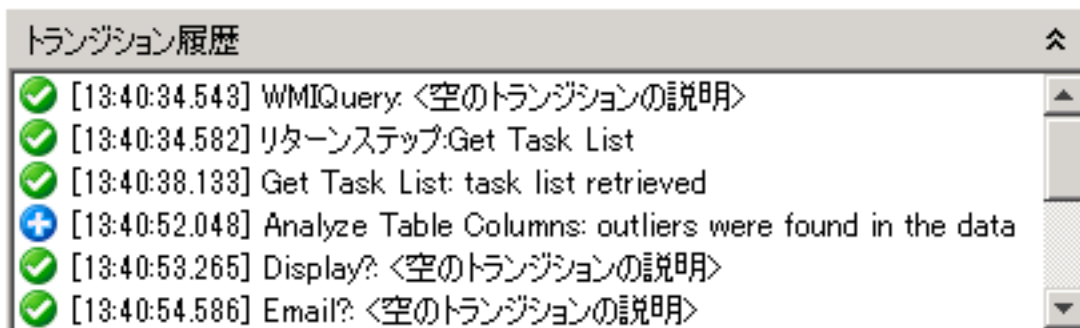
- [ブレークポイント] は、特定のステップで実行を自動的に一時停止し、停止ポイントでのフロー変数の値、実行のパス、または結果を調べるためのフラグです。
- [レスポンスオーバーライド] により、オペレーションの結果に関係なく、選択したレスポンスを強制的に指定できます。
- [Studioオーバーライド] は、有効なStudioオーバーライドを表示します。このビューでは、必要に応じてオーバーライドを有効および無効にできます。



- [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、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、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。

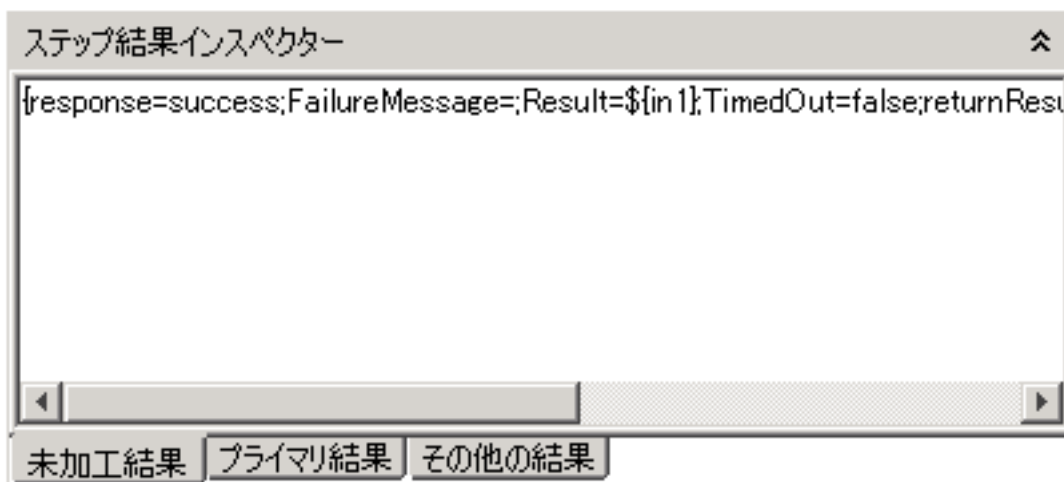
[デバッグオプション] ペインでは、ブレークポイントとレスポンスオーバーライドが表示され、該当の実行について各要素の削除、有効化、または無効化が可能です。

[トランジション履歴] ペイン



[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。

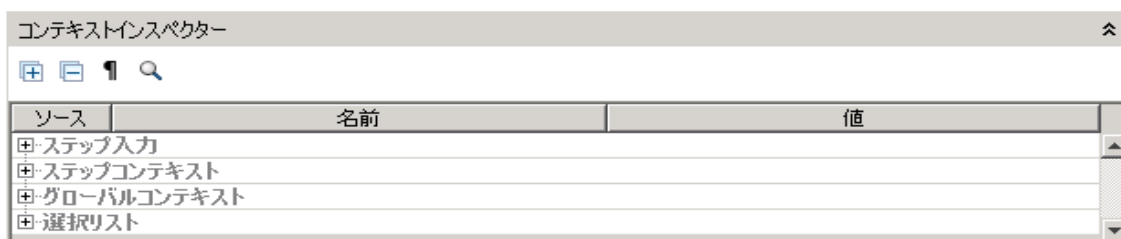
[ステップ結果インスペクター] ペイン



[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
設定	実行のパラメーターを設定します。[実行設定]ダイアログボックスが表示されます。	
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](#)」(293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なかったことですが、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](#)」(297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、HPE OO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、『HPE OO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)、HPE OO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。

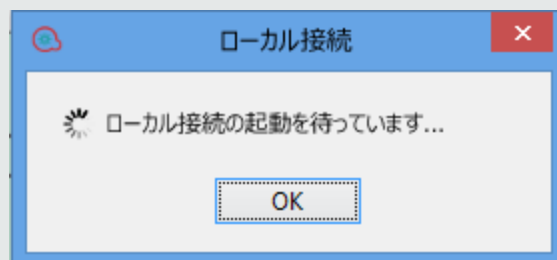
リモートデバッグセッションによって残されたトレースは、他の実行と同じように、Central実行エクスプローラー(ドリルダウン)、およびダッシュボードに表示されます。このようなトレースのソースは、最終的にはデバッガーでの実行が起点であると識別できるので、フィルターで除外できます。ドリルダウンは、Studioから送信されたフローを反映します(最終変更)。

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すが、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。

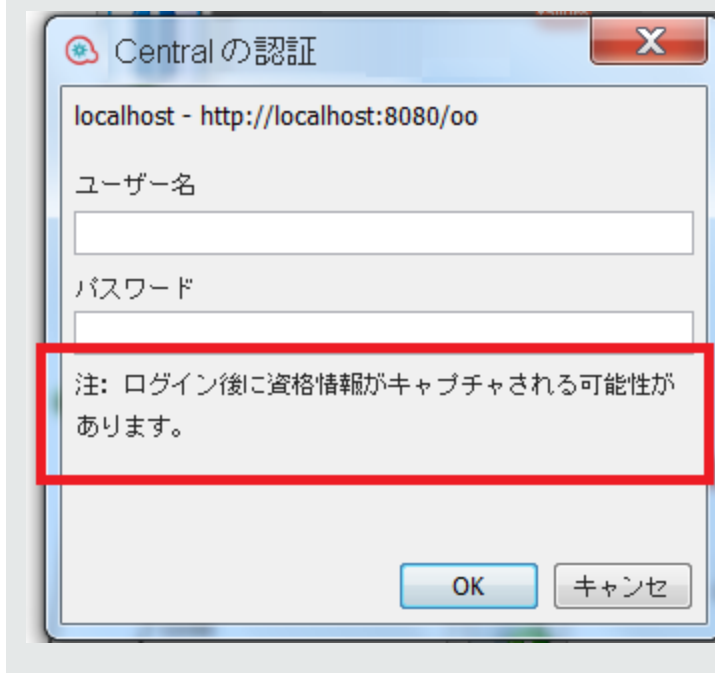
注: この制限は、ローカル接続(リモートCentralを構成していない場合にStudioでデフォルトで使用される接続)を使用してデバッグを行うときには適用されません。

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、資格情報の入力が必要

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の資格情報のキャプチャーを有効にする] 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。

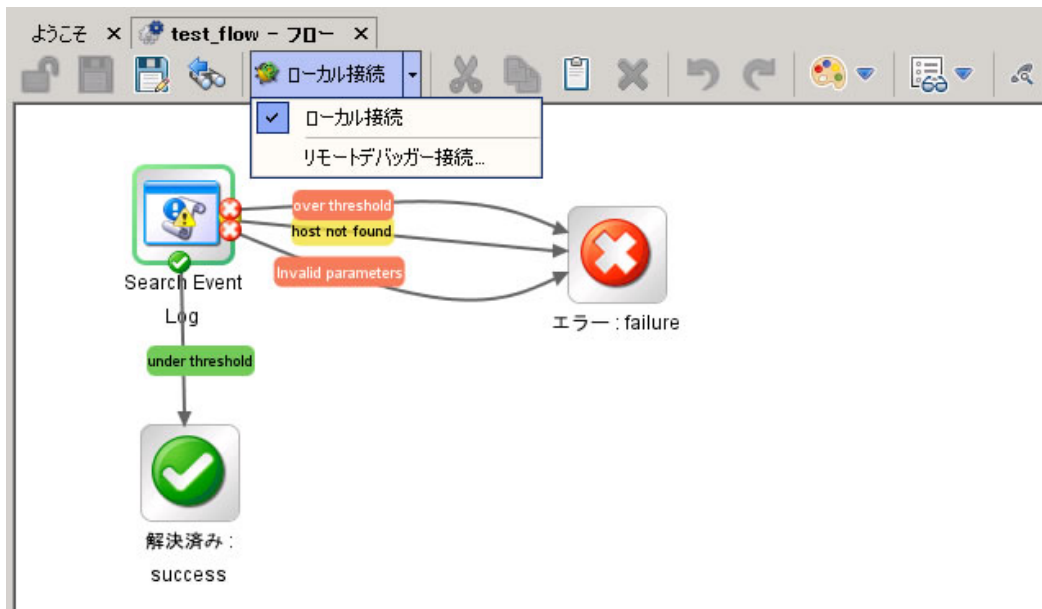
6. **Studio構成アイテム**、**Central構成アイテム**、**コンテキストインスペクター**の組み合わせを用いて、実行が行われます。コンテキストインスペクターの詳細については、「[コンテンツの検査](#)」(324ページ)を参照してください。
7. (オプション) ブレークポイントを設定し、デバッグ中にコンテキストインスペクターで変数と構成アイテムを変更します。
8. 実行が完了します。

実行する操作

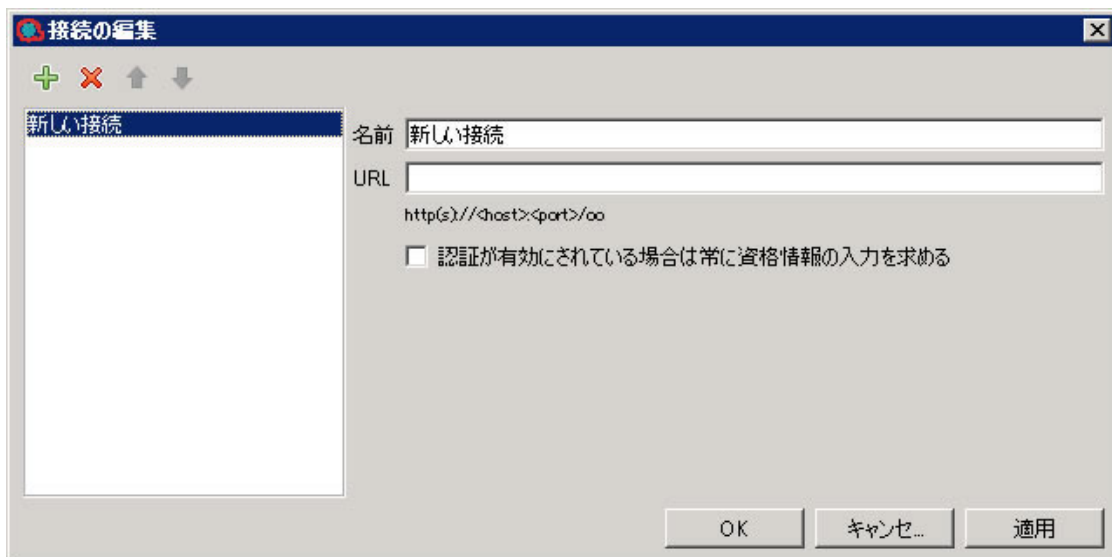
Central接続の追加または編集

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行います。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。

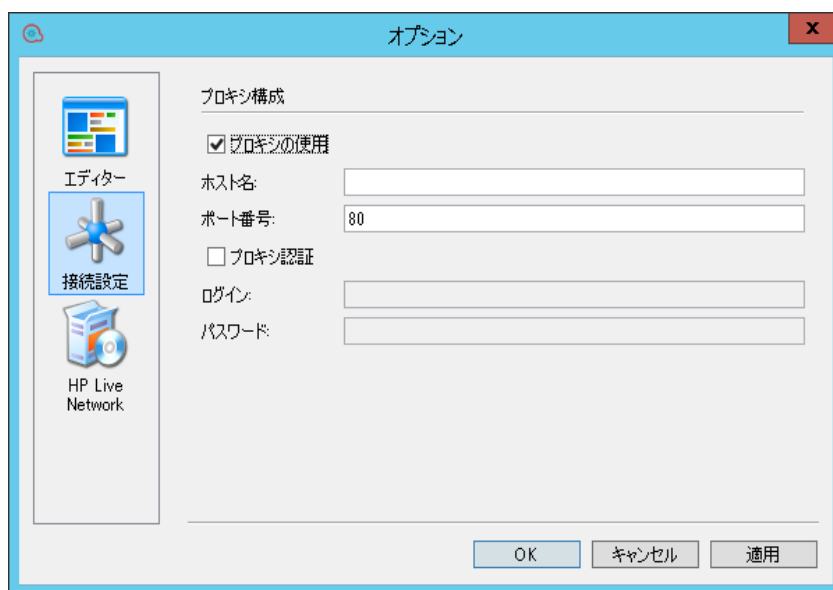
注: `studio.properties`ファイルのリモート接続は構成できません。

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、次の手順を実行します。

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 TrustStore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 TrustStore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せん。

- [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。

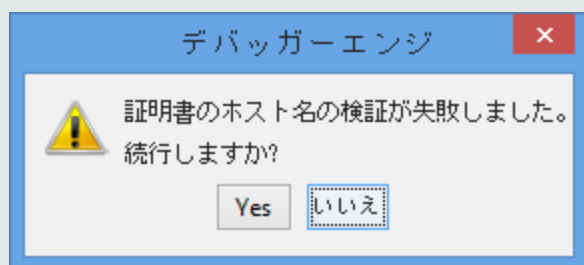
ただし、Studioを再起動すると、この問題の解決が期待されるために、ホスト名の検証メッセージがもう一度表示されます。証明書と一致するようにURLを修正するか、接続URLに従って証明書を変更する必要があります。

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、次の手順を実行します。

1. インポートしたコンテンツパックまたは既存のコンテンツパックからのオペレーションとフローを使用してフローを構築します。
2. フローを開きます。
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、次の手順を実行します。

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。

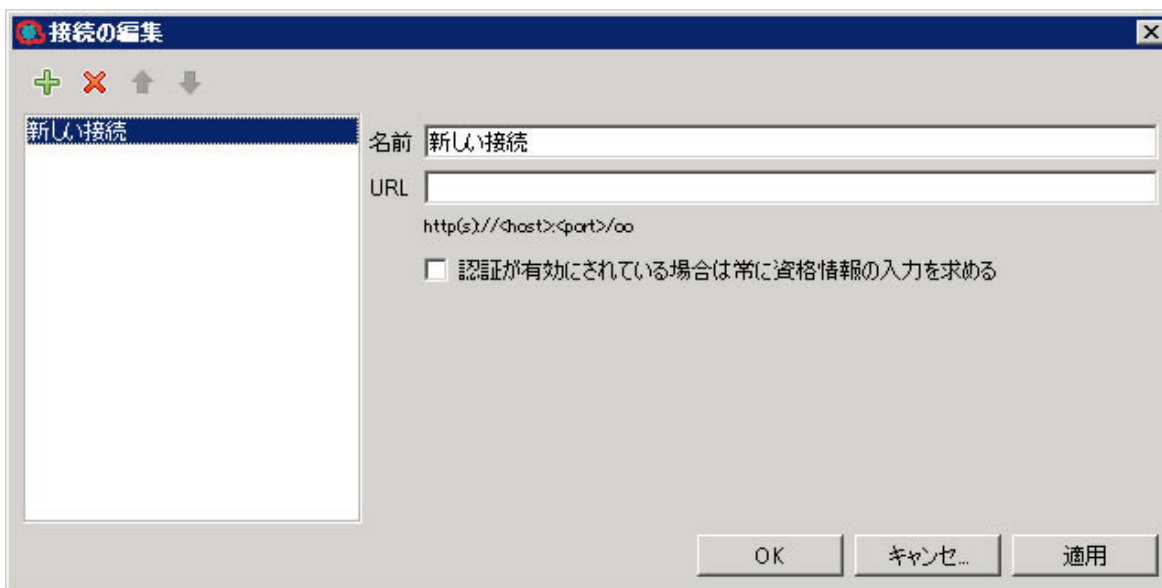
フローの再実行



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キスポート

フローの検査が終了したら、フローをデプロイして実行するよう、コンテンツパックにリリースできるようになります。

コンテンツパックは、プロジェクトの成果です。これには、プロジェクトのエンティティと参照IDが含まれます。コンテンツパックには、フローやオペレーションだけでなく、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: 重複するUUIDを持つフローまたはオペレーションなどの無効なフローまたはオペレーションは、コンテンツパックに含めることができません。

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](#)」(352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を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:



The screenshot shows a window titled 'ようこそ x Project2 x'. It contains a table with the following information:

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、1.1.1.1.1.1.1.2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。

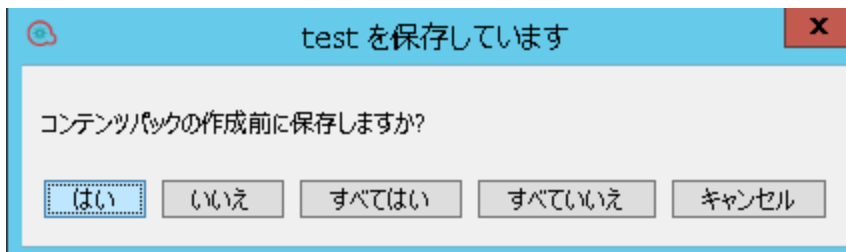
実行する操作

コンテンツパックの作成

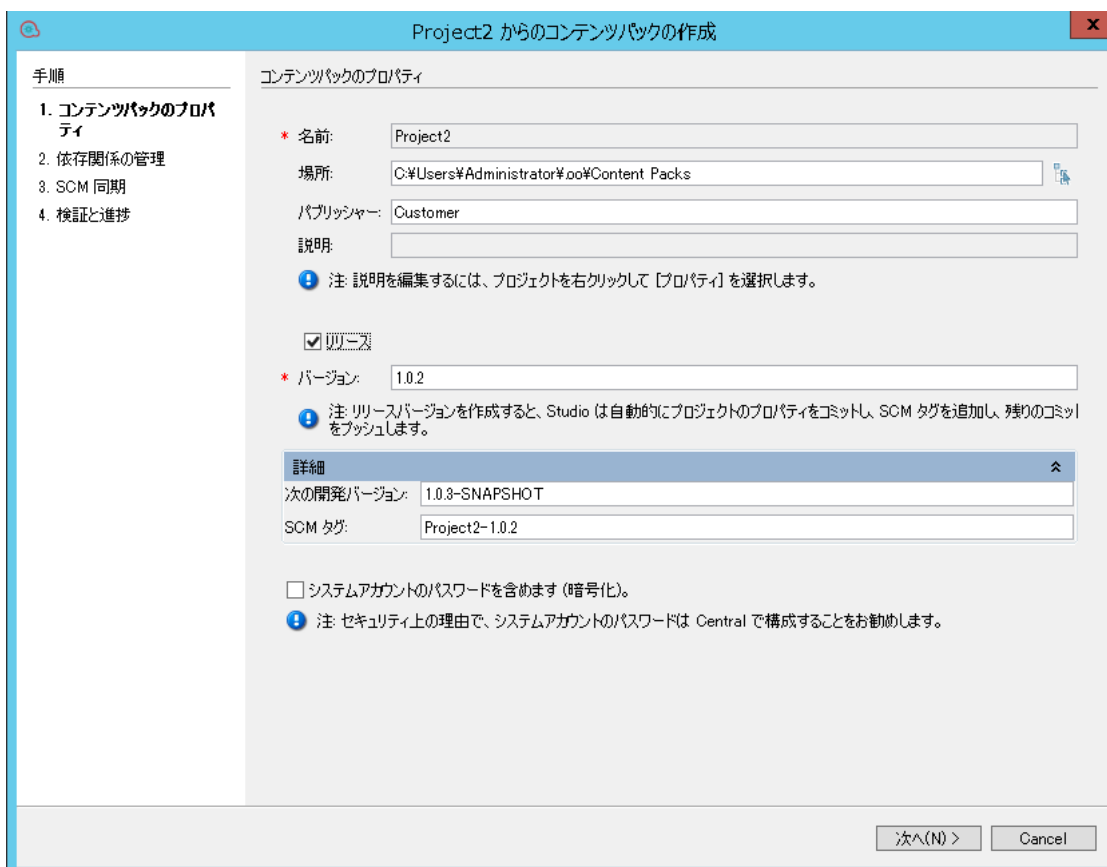
1. [プロジェクト] ペインで、コンテンツパックを作成するプロジェクトを選択します。
2. [ファイル] > [新しいコンテンツパックの作成] を選択します。

注: または、[プロジェクト] ペインの [コンテンツパックを作成] 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「Content Pack Properties」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](#)」(364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](#)」(353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。

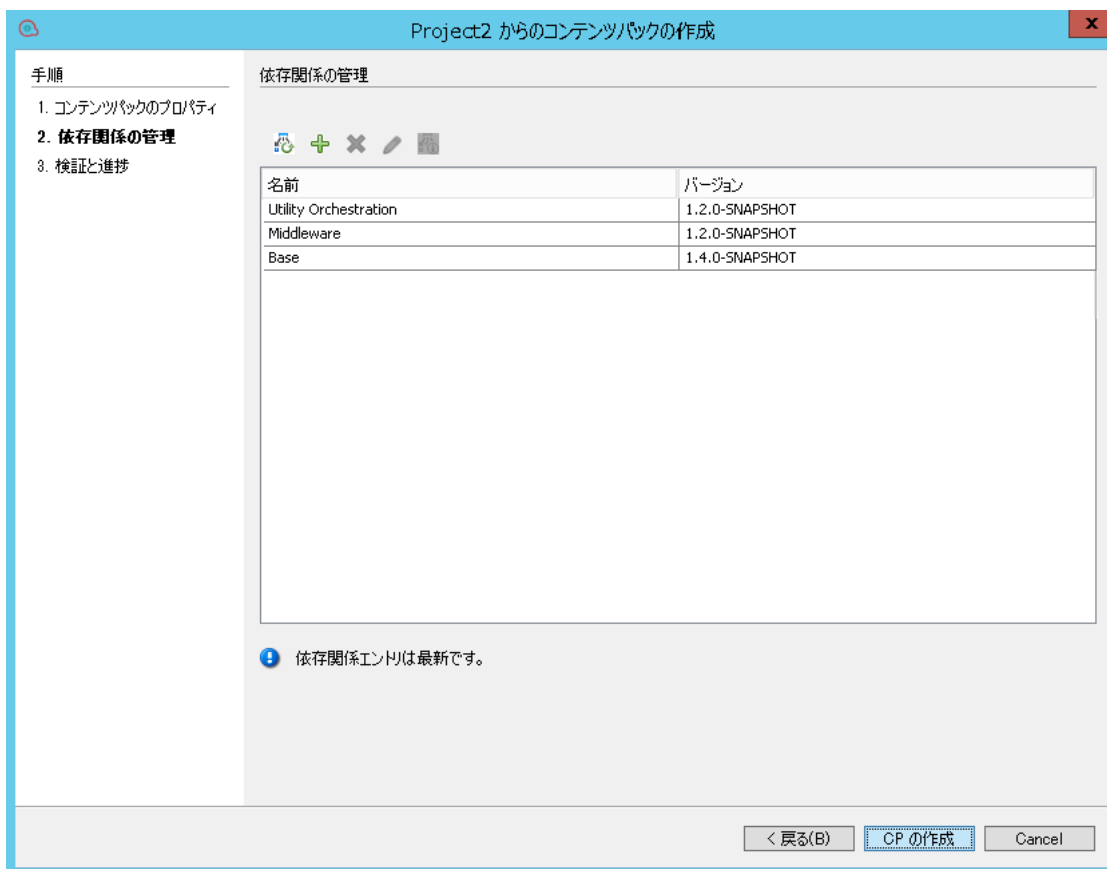
9. **ソースコントロールの場合のみ:** SVN/Gitに対する変更をコミットしたときにリビジョン番号が増分されるようにする場合は、[リリース] チェックボックスをオンにします。
10. [詳細] を選択して、次のフィールドを表示します。

- 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](#)」(118 ページ)を参照してください。

14. **[Next]** をクリックします。

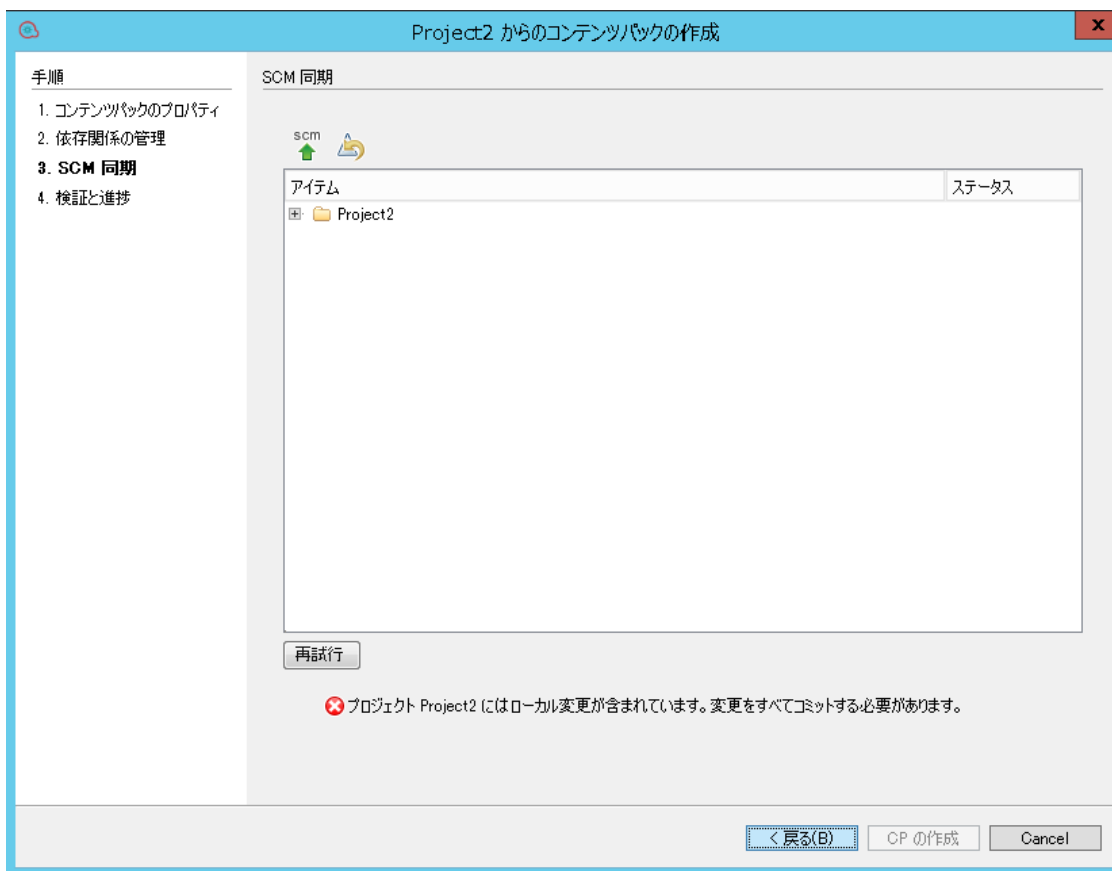
最初の手順で **[リリース]** チェックボックスをオンにした場合は、**[SCM同期]** ウィンドウが開きます。

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](#)」(71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。

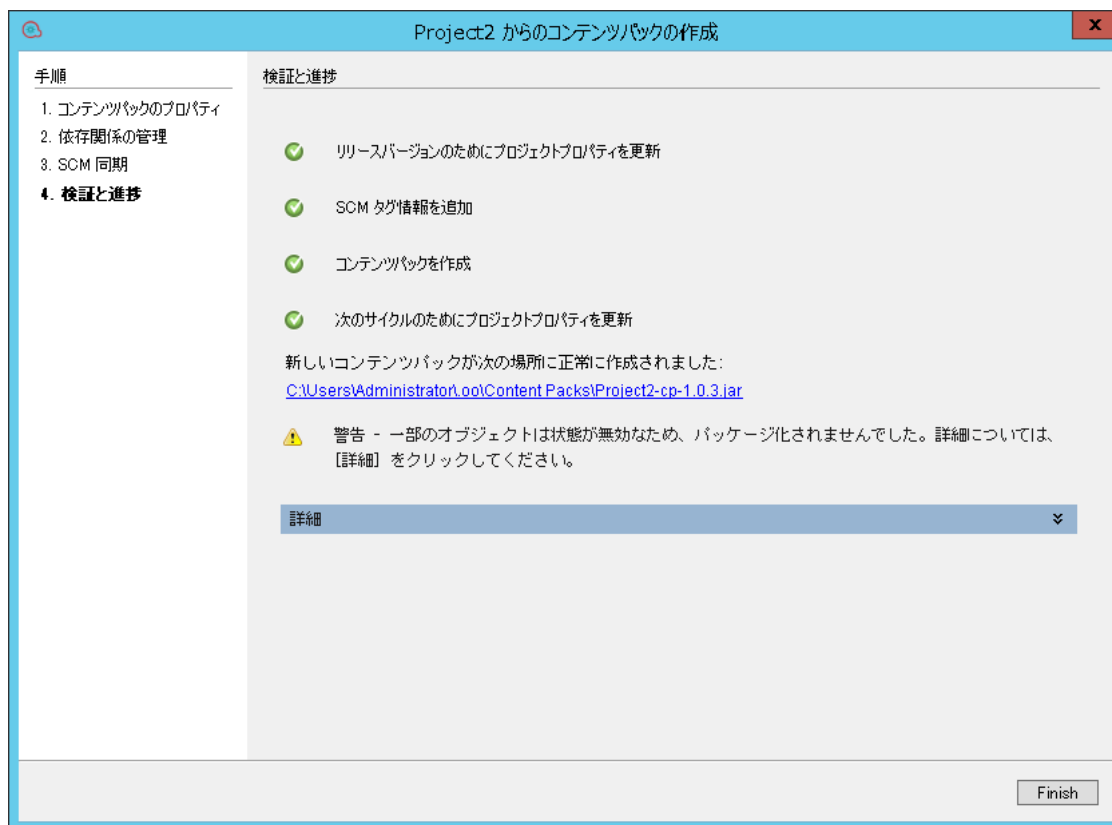
コンテンツパックの作成を続行できます。

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、StudioI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。

- **コンテンツパック:** `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
- **プロジェクト:** `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] ペインで[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\AppData\Local\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)。

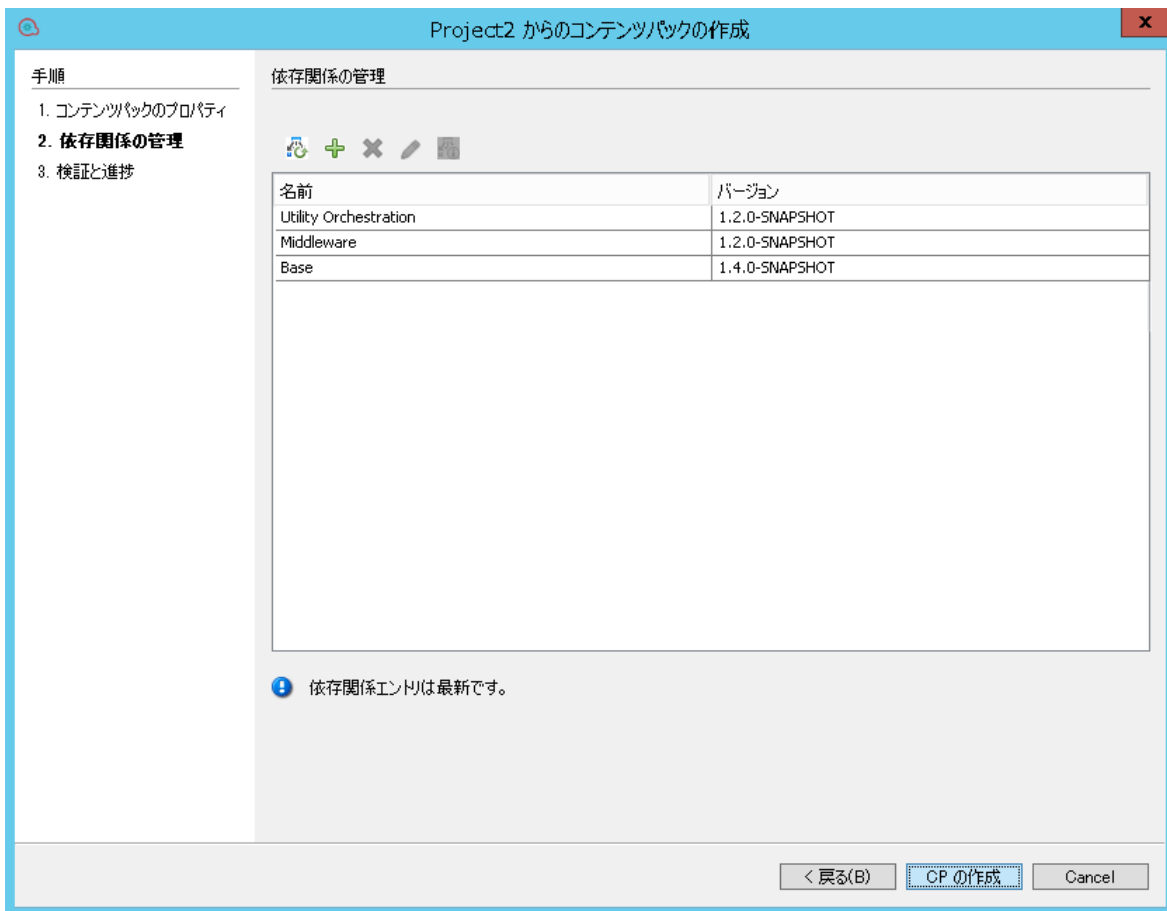
注: セキュリティ上の理由で、システムアカウントのパスワードは Central で構成することをお勧めします。





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。

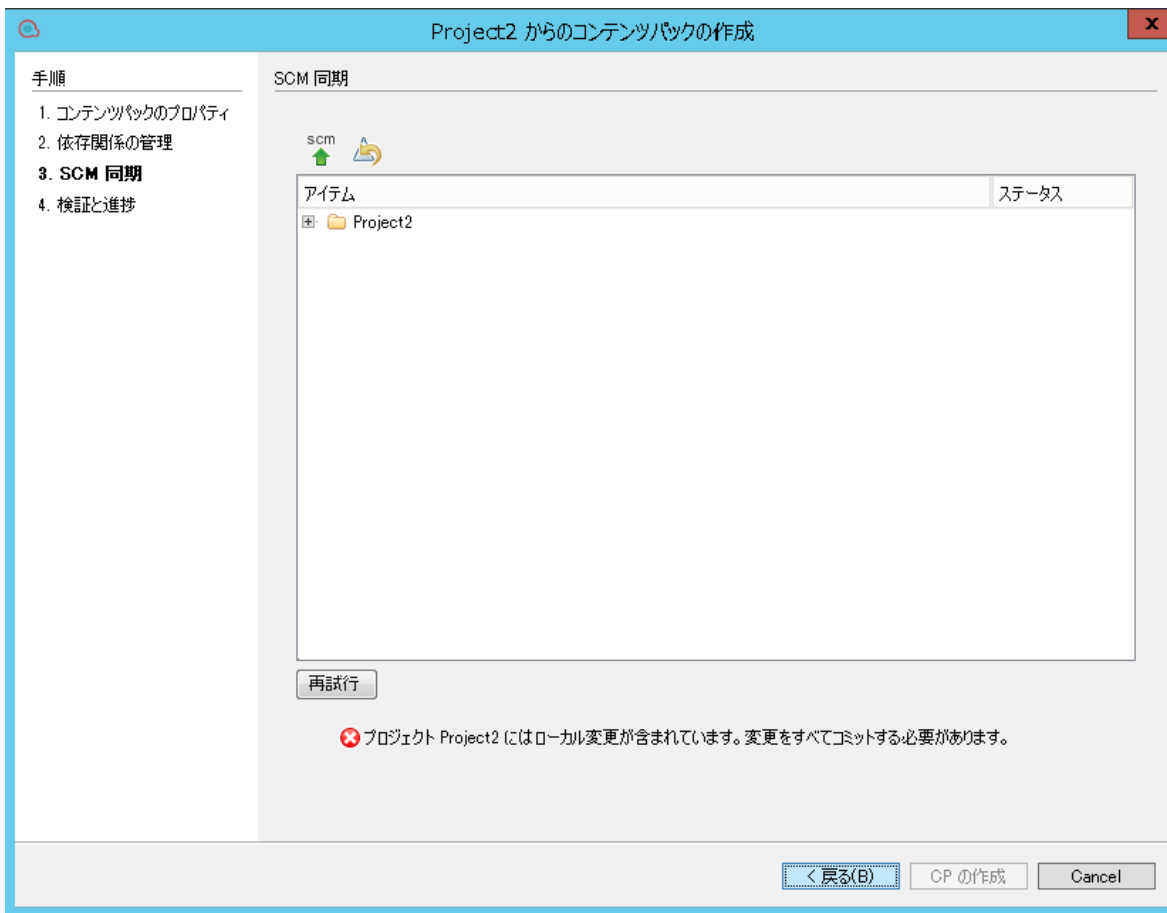
場所	コンテンツパックを保存する場所を入力または参照します。デフォルトで、プロジェクトのワークスペースへのパスが選択されています。
パブリッシャー	コンテンツパックのパブリッシャーを入力します。この情報は、コンテンツパックの[プロパティ]ページに表示されます。
説明	[説明]フィールドを編集するには、プロジェクトツリーでプロジェクトを右クリックして[プロパティ]を選択します。次に、[説明]領域に説明を入力します。
リリース	<p>SVN/Gitに対する変更をコミットしたときにリビジョン番号が増分されるようにする場合は、このチェックボックスをオンにします。</p> <p>これにより、プロジェクト (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>
バージョン	コンテンツパックのバージョンを入力します。この情報は、コンテンツパックの[プロパティ]ページに表示されます。
詳細	矢印を選択して、SCMフィールドを表示します。
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。
SCMタグ	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す。
システムアカウントのパスワードを含めますか?	<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Studio1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Studio1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」(202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」(213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」(248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

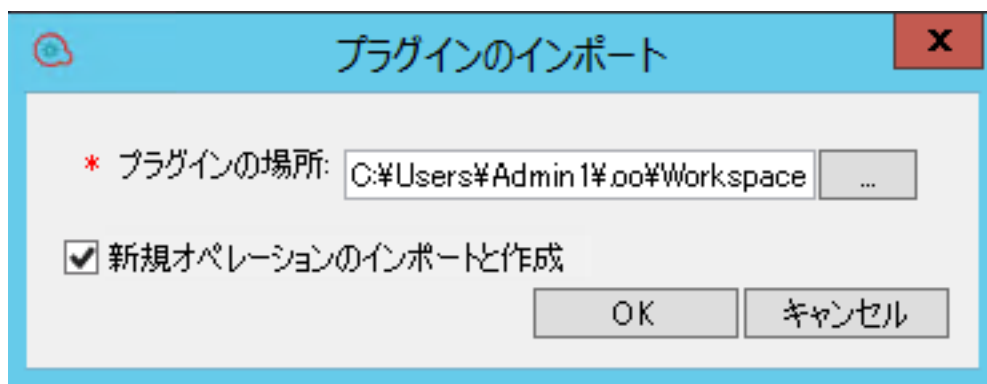
既存のオペレーションをコピーしてオペレーションを作成した場合、ソースオペレーションを変更できます。コピーしたオペレーションは、ソースとして選択したオペレーションのコピーになります。

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。

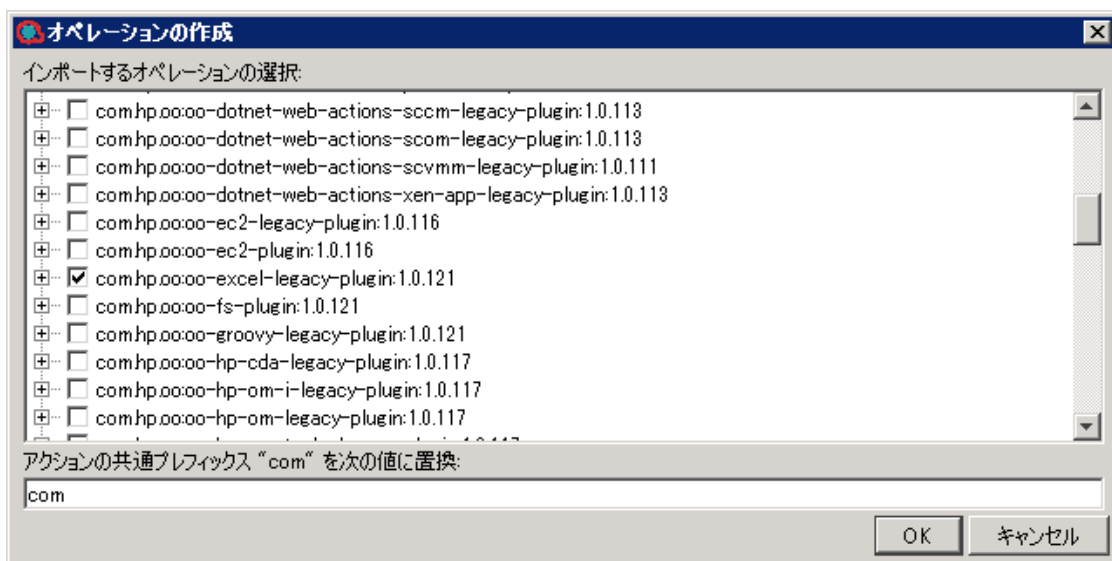
1. アクションプラグインを作成してパックし、開発したアクションを含めます。アクションプラグインの開発方法の詳細については、『アクション開発者ガイド』を参照してください。
2. Studioで新しいオペレーションの作成先のフォルダーを右クリックして[プラグインのインポート]を選択します。
3. [プラグインのインポート] ダイアログボックスで、参照ボタンをクリックして、インポートするHPE OOプラグインを参照して選択します。



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。

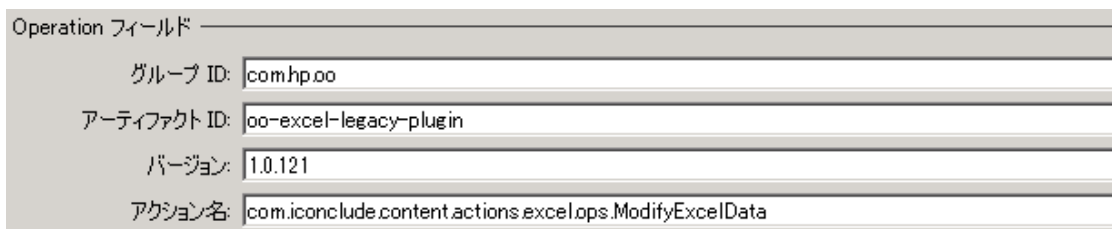
4. [新規オペレーションのインポートと作成] チェックボックスをオンにして [OK] をクリックします。
5. [オペレーションの作成] ダイアログボックスで、必要なアクションを含むプラグインを展開し、オペレーションの作成元として使用するアクションを選択します。



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。

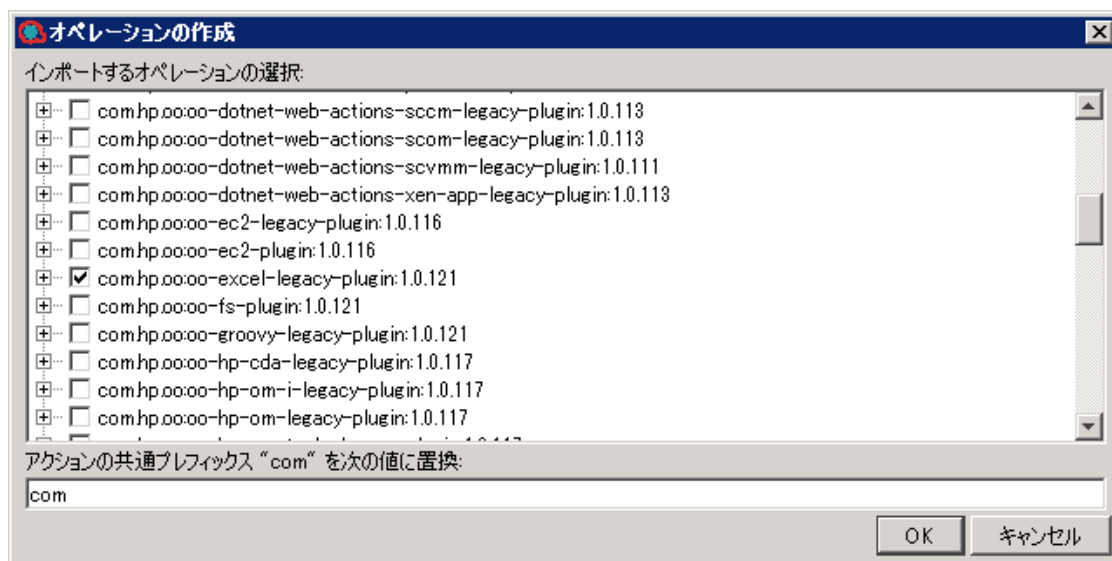


6. オペレーションを保存します。

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HPE OOプラグインを参照して選択します。
3. [OK] をクリックします。作成者はこのプラグインからオペレーションを作成できます。詳細については、「インポートしたアクションプラグインからのオペレーションの作成」を参照してください。

手動オペレーションの作成

手動オペレーションは、操作を選択できるオペレーションです。ユーザーが、実行時に操作を選択する必要があります。

手動オペレーションを作成するには、手動オペレーションテンプレートをベースコンテンツからコピーして、ユーザーが選択できる操作を定義します。

1. **[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。

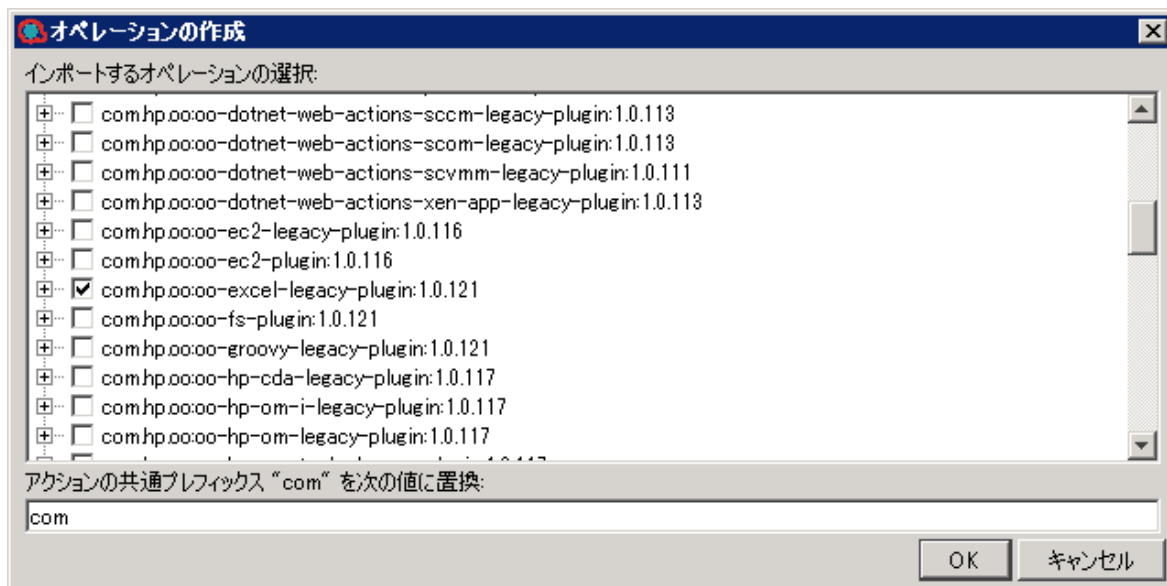
表示オペレーションを作成するには、ベースコンテンツにある表示オペレーションテンプレート Library/Utility Operations/Display Messageをコピーして、ユーザーに表示する情報を定義します。

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:

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

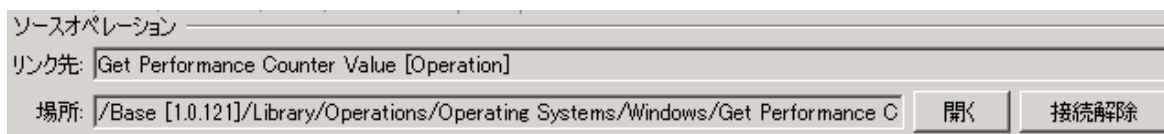
プラグインのインポート
X

* 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

GUIアイテム	説明
プラグインの場所	インポートするHPE 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](#)」(393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を検索する

[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。

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。

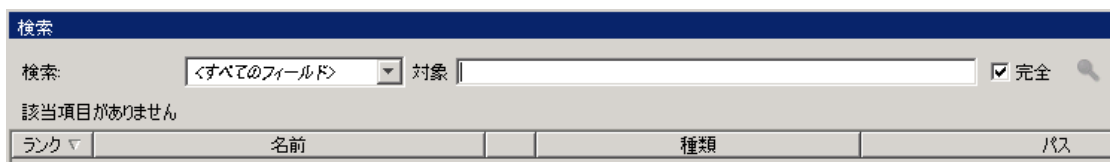
3. **[対象]** テキストボックスに、検索するテキストを入力します。

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、次の手順を実行します。
 - 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。



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。

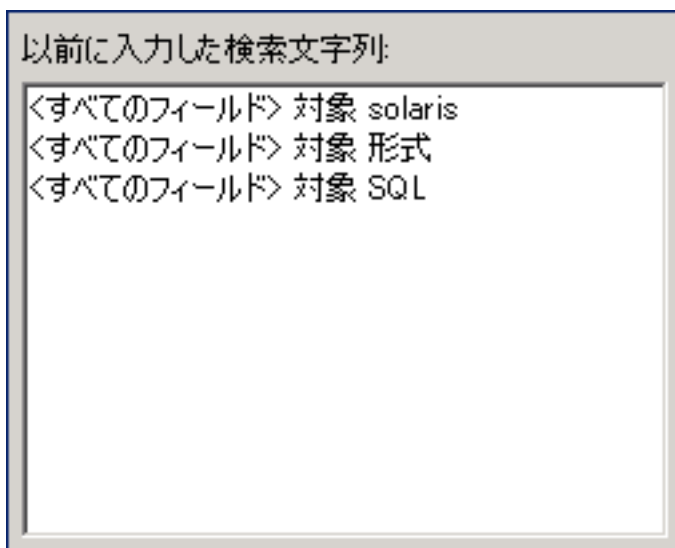
- オペレーションの入力が必要な種類の情報。
- 結果に含まれる情報。
- オペレーションの要件と前提条件。

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

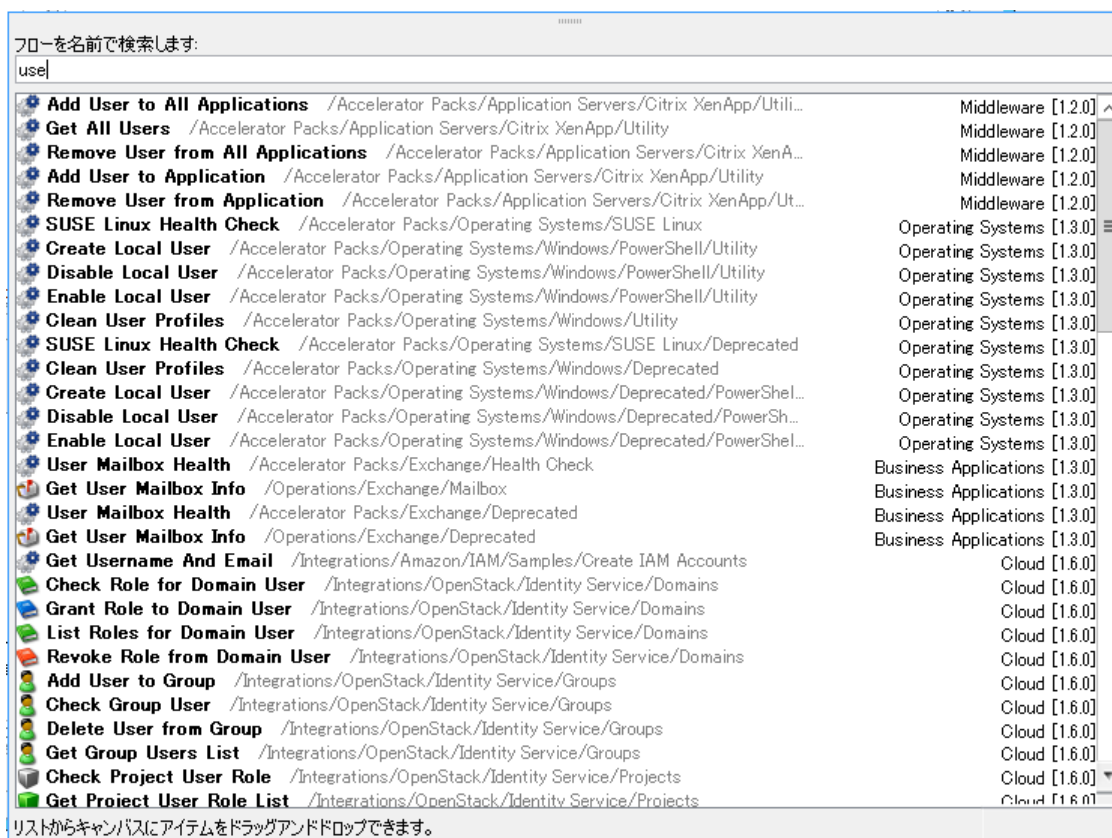
[フローに移動...] オプションを使用して、ワークスペース内のフローを簡単に見つけて開くこともできます。

1. [ツール] > [フローに移動...] を選択します。

[フローを名前を検索します:] 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。

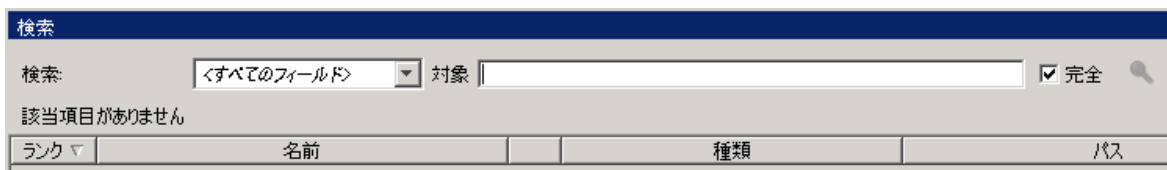
ウィンドウに検索結果が直ちに表示されます。たとえば、"use" という文字列が含まれる名前 のフローを検索する場合、次のような結果が表示されることがあります。



- 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。
- 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。

	<ul style="list-style-type: none">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せん。
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。

オペレーションまたはサブフローへの参照の検索および置換

オペレーションまたはサブフローへの参照をすべてのオープンプロジェクトに渡ってグローバルに変更したり、または特定のフロー内で変更したりする必要がある場合があります。

たとえば、フロー内に**Change Status**という名前のサブフローへの参照があり、この参照を**Update Status**という名前で新しく作成したサブフローに変更する必要がある場合があります。この操作は、**オペレーションまたはサブフローへの参照の置換**機能を使用して実行できます。

実行する操作

オペレーションまたはサブフローへの参照の検索および置換

1. [編集] > [置換]を選択するか、**Ctrl+R**を押します。


[フローまたはオペレーションの参照の置換]ダイアログが開きます。

2. **[検索対象]** フィールドに、検索するオペレーションまたはサブフローの完全パスを入力するか、貼り付けます。例: **/My Project/Library/My Ops Flows/Restart Service**。

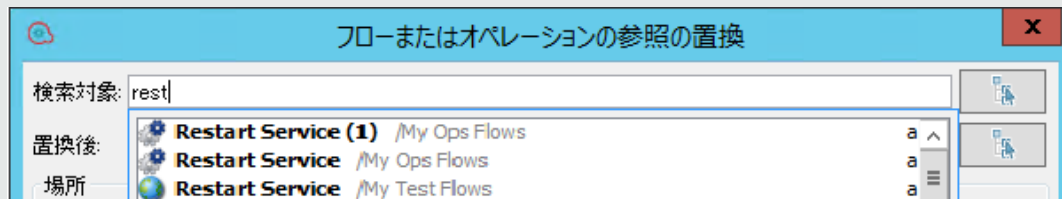
または、



をクリックして、プロジェクトツリーからサブフローまたはオペレーションを選択します。

3. 同様に、**[置換後]** フィールドに、元のオペレーションまたはサブフローと置き換えるのに使用するオペレーションまたはサブフローの完全パスを入力するか貼り付ける、または  をクリックしてプロジェクトツリーからサブフローまたはオペレーションを選択します。

注: **[検索対象]** フィールドおよび **[置換後]** フィールドの両方に文字を入力すると、フィールドに検索結果が自動的に入力されます。たとえば、"rest" という文字列を入力すると、次のような結果が表示されることがあります。

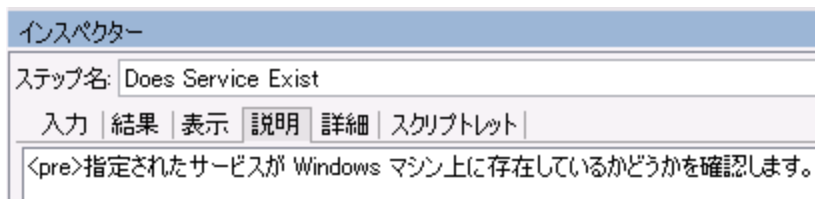


4. **[場所]** 領域で、検索の範囲を選択します。
 - すべてのオープンプロジェクト
 - 選択したフォルダー
 - 現在のフロー
5. 上記で選択した条件 (すべてのオープンプロジェクトのサブフロー、選択したフォルダー、現在のフローのいずれか) に基づいてソースアイテムの検索にサブフローを含めるには、**[サブフローを含める]** を選択します。
6. **[入力]** 領域で、次のいずれかのオプションを選択します。
 - **ステップ入力のマージ:** ソースアイテムの入力とターゲットアイテムの入力をマージする
 - **ステップ入力の上書き:** ターゲットアイテムの入力をソースアイテムの入力で上書きする
7. オペレーションをポイントしているステップを置き換えると、(ステップのインスペクターの **[結果]** タブで定義したとおりに) ステップの結果がオペレーションの出力のいずれかにマッピングされます。

この出力を削除すると、結果はマッピングされなくなります。

これらの結果を削除するには、**[結果]** 領域で **[マッピングされていない結果の削除]** を選択します。

8. **[追加のオプション]** 領域で、該当するチェックボックスをオンにします。
 - **ステップのスクリプトレットの削除**: ターゲットアイテム内のステップのスクリプトレットが不要になった場合にこれを削除するには、このチェックボックスをオンにします。
 - **ステップの説明の置換**: **[説明]** タブで、ターゲットアイテムのステップの説明をソースアイテムのステップ説明に置き換えるには、このチェックボックスをオンにします。



- **ステップ名の置換**: ターゲットアイテムのステップ名をソースアイテムのステップ名で置き換えるには、このチェックボックスをオンにします。
9. 検索を開始するには、**[次を検索]** をクリックします。

次にアイテムと一致する箇所を示すメッセージが表示されます。たとえば、**Does Service Exist?**というステップの検索を実行すると、次の結果が生成されます。



```
Found match in flow abc/Library/My Ops Flows/Restart Service
```

10. アイテムのこの箇所を置き換えるには、**[置換]** をクリックします。(**[場所]** 領域に指定されている場所で) アイテムと一致するすべての箇所を置き換えるには、**[すべて置換]** をクリックします。

注: フローが開いているときに**[フローまたはオペレーションの参照の置換]** ダイアログを開くと、**[検索対象]** フィールドに現在のステップの参照が自動的に入力されます。

参考資料

参照の置換

GUIアイテム	説明
検索対象	探しているオペレーションまたはサブフローの完全パスを入力するか貼り付ける、または  をクリックして、プロジェクトツリーからサブフローまたはオペレーションを選択します。
置換後	元のオペレーションまたはサブフローと置き換えるのに使用するオペレーションまたはサブフローの完全パスを入力するか貼り付ける、または  をクリックしてプロジェクトツリーからサブフローまたはオペレーションを選択します。



場所	ソースアイテムを検索する場所を選択します。
すべてのオープンプロジェクト	Studioで現在開いているすべてのプロジェクト内のソースアイテムを検索します。
選択したフォルダー	選択したフォルダー内のみのソースアイテムを検索します。 ボタンを使用してフォルダーを選択します。 
現在のフロー	現在のフロー内のみのソースアイテムを検索します。
サブフローを含める	上記で選択した条件 (すべてのオープンプロジェクトのサブフロー、選択したフォルダー、現在のフローのいずれか) に基づいてソースアイテムの検索にサブフローを含めます。
入力	
ステップ入力のマージ	ソースアイテムの入力とターゲットアイテムの入力をマージします。
ステップ入力の上書き	ターゲットアイテムの入力をソースアイテムの入力で上書きします。
結果	
マップされていない結果の削除	ターゲットアイテムのステップのマップされていない結果を削除します。詳細については、「 オペレーションまたはサブフローへの参照の検索および置換 」(381ページ)のステップ8を参照してください。
追加のオプション	
ステップのスク립トレットの削除	ターゲットアイテム内のステップのスク립トレットが不要になった場合にこれを削除するには、このチェックボックスをオンにします。
ステップの説明の削除	ステップのドキュメントプロパティを、新しいフローまたはオペレーションのドキュメントで置き換えるには、このチェックボックスをオンにします。
ステップ名の置換	ターゲットアイテムのステップ名をソースアイテムのステップ名で置き換えるには、このチェックボックスをオンにします。

グローバル検索および置換テキスト

ビッグプロジェクトとは、多数のフローまたはオペレーションが含まれているという意味です。一般に、フローは非常に複雑です。入力値としてのテキスト、またはステップ、フロー、オペレーションの説明内のテキストなど、テキストの変更が必要になるたびに、作成者はすべてのフロー、ステップ、オペレーションを確認する必要があります。

参考資料

参照の置換

GUIアイテム	説明
検索対象	探しているオペレーションまたはサブフローの完全パスを入力するか貼り付ける、または  をクリックして、プロジェクトツリーからサブフローまたはオペレーションを選択します。
置換後	元のオペレーションまたはサブフローと置き換えるのに使用するオペレーションまたはサブフローの完全パスを入力するか貼り付ける、または  をクリックしてプロジェクトツリーからサブフローまたはオペレーションを選択します。

場所	ソースアイテムを検索する場所を選択します。
すべてのオープンプロジェクト	Studioで現在開いているすべてのプロジェクト内のソースアイテムを検索します。
選択したフォルダー	選択したフォルダー内のみのソースアイテムを検索します。  ボタンを使用してフォルダーを選択します。
現在のフロー	現在のフロー内のみのソースアイテムを検索します。
サブフローを含める	上記で選択した条件 (すべてのオープンプロジェクトのサブフロー、選択したフォルダー、現在のフローのいずれか) に基づいてソースアイテムの検索にサブフローを含めます。
入力	
ステップ入力のマージ	ソースアイテムの入力とターゲットアイテムの入力をマージします。
ステップ入力の上書き	ターゲットアイテムの入力をソースアイテムの入力で上書きします。
結果	
マップされていない結果の削除	ターゲットアイテムのステップのマップされていない結果を削除します。 詳細については、「 グローバル検索および置換テキスト 」(385ページ)のステップ8を参照してください。
追加のオプション	
ステップのスク립トレットの削除	ターゲットアイテム内のステップのスク립トレットが不要になった場合にこれを削除するには、このチェックボックスをオンにします。
ステップの説明の削除	ステップのドキュメントプロパティを、新しいフローまたはオペレーションのドキュメントで置き換えるには、このチェックボックスをオンにします。
ステップ名の置換	ターゲットアイテムのステップ名をソースアイテムのステップ名で置き換えるには、このチェックボックスをオンにします。

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](#)」(366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OOでは、アクションプラグインにリンクされたオペレーションをコピーすると、**ハードコピー**が作成されました。これは、コピーが、元のオペレーションと同様にアクションプラグインに直接リンクしています。アクションが更新された場合(たとえばJARの名前またはクラスが変更された場合)、ハードコピーされたオペレーションもすべて更新する必要がありました。

OO 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、元のエ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、元のエ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] > [貼り付け] を選択します。

フローとオペレーションの使用法の詳細

オペレーションやフローが既存のフローでどのように使用されているかを調べることで、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。

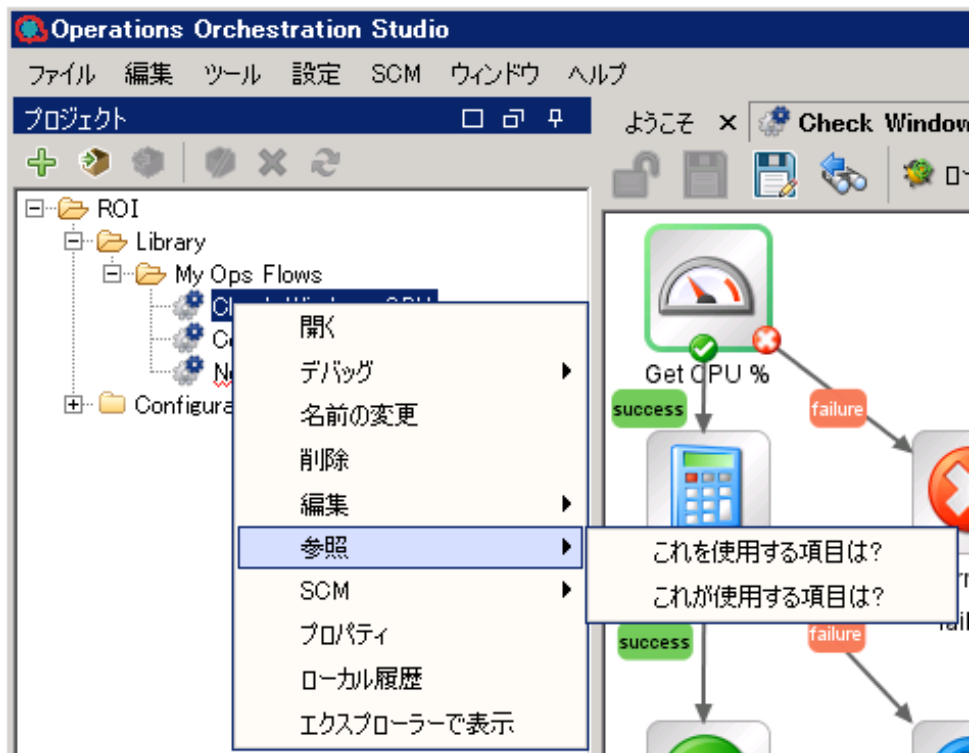
ベストプラクティス

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?] を選択します。



[参照] ペインが開き、オペレーションまたはフローの参照元が表示されます。

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?] を選択します。

[参照] ペインが開き、オペレーションまたはフローの参照先が表示されます。

ヒント: 参照元であるフローやオペレーションは、コピー、貼り付け、編集が可能なサンプルとして役立ちます。

参考資料

[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。

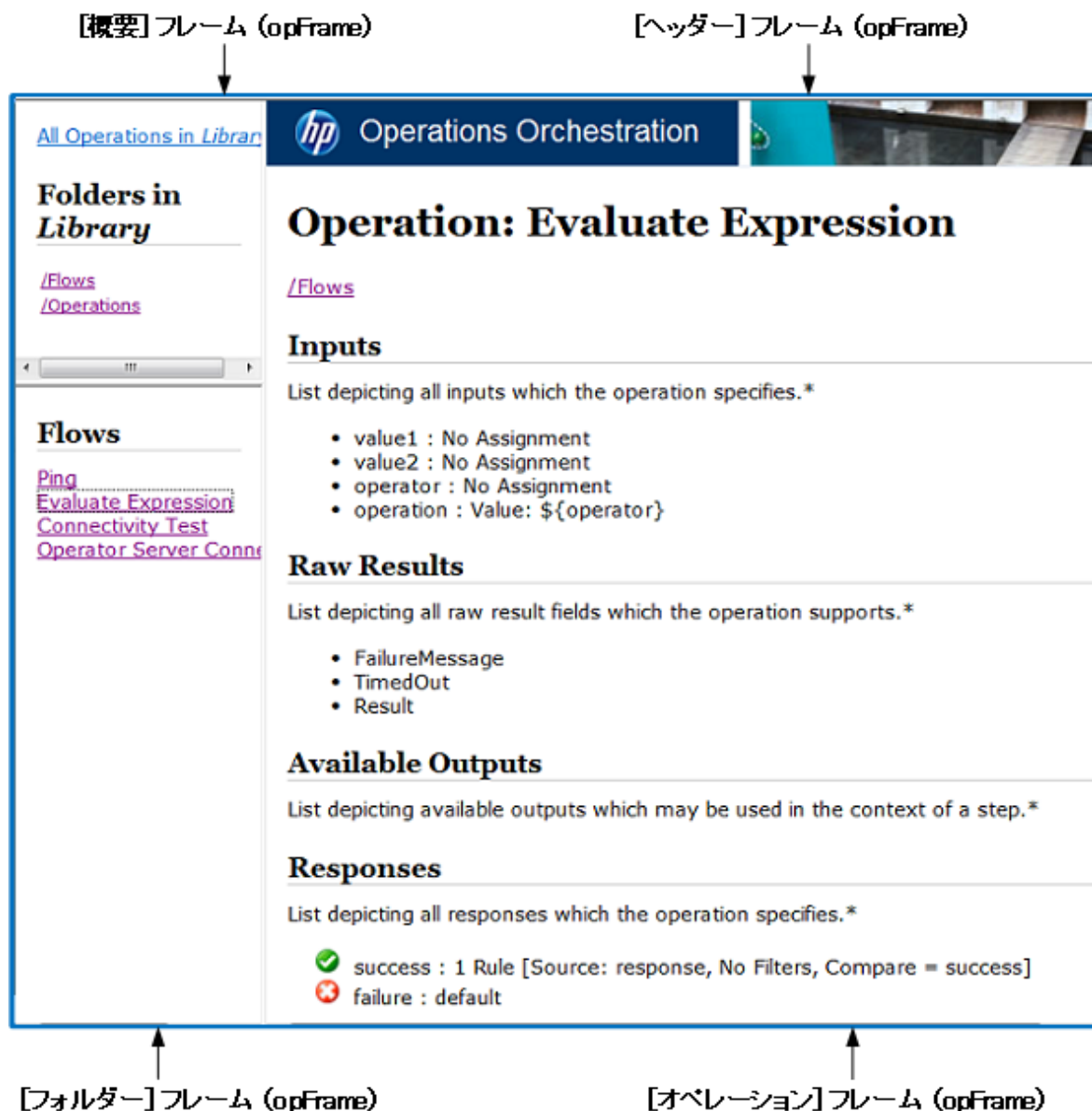
フローやオペレーションをドキュメントにし、ほかのユーザーに詳細情報を提供することができます。

- 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、OOバナーが表示されます。
- **[オペレーション]** フレーム – 右下の**[オペレーション]** フレームには、フローまたはオペレーションの説明が表示されます。**[プロパティ]** シートの**[説明]** タブで入力した情報です。



実行する操作

フローのPNG画像でのエクスポート

1. フローを作成ペインで開きます。
2. 作成ペインを右クリックし、[PNGにエクスポート]を選択します。
3. 画像の保存先を参照し、[保存]をクリックします。

標準形式でのドキュメントの生成

特定のフローやオペレーションを含むフォルダーのドキュメントや、**Library**フォルダー全体のドキュメントを生成することができます。

1. ドキュメントを作成するフォルダーを右クリックします。

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て、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を呼び出してそのフレームセットに組み込みます。

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

All_folders_template.vm

フォルダーのコンテンツの表を生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **All_ops_template.vm** – すべてのオペレーションのリストを生成し、リンクを作成して**folderFrame** (左下)に表示します。
- **Folder_contents.vm** – 選択したフォルダーのコンテンツのリストを生成し、リンクを作成して**folderFrame** (左下)に表示します。

All_ops_template.vm

すべてのオペレーションのコンテンツおよびすべての子オペレーションのドキュメントの表を生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Op_template.vm** – リンクを作成して**opFrame** (右下)に表示します。

Folder_overview_template.vm

フォルダーのコンテンツについての概要を表形式で生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Folder_contents.vm** – リンクを作成して**folderFrame** (左下)に表示します。

Op_template.vm

単独のオペレーションのドキュメントを生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Folder_template.vm** – リンクを作成して同じフレームに表示します (親フォルダー以下)。
- **Folder_contents.vm** – フォルダーのコンテンツを**folderFrame**に表示します。

Flow_template.vm

単独のフローのドキュメントを生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Flow_template.vm** – リンクを作成して同じフレームに表示します (親フォルダー以下)。
- **Folder_contents.vm** – フォルダーのコンテンツのリストを生成し、リンクを作成して**folderFrame** (左下)に表示します。
- **Op_template.vm** – リンクを作成して**opFrame** (右下)に表示します。

Folder_contents.vm

単独のフォルダーのコンテンツの表を生成します。

- **Header.css** – 全体的なフォント、色などを定義しているスタイルシート。
- **Op_template.vm** – リンクを作成して**opFrame** (右下)に表示します。

Header.html

ヒューレット・パッカートのバナー。

Hp_rockwell.css

ヒューレット・パッカートのバナー用のスタイルシート。

Hp_steps_307x39.jpg

ヒューレット・パッカートのバナー用の画像。

Logo_hp_smallmasthead.gif

ヒューレット・パッカートのバナー用のロゴ。

フローとオペレーションのバージョン履歴管理

構成アイテム、フロー、またはオペレーションが保存されるたびに、そのアイテムの新しいバージョンが作成されます。[バージョン履歴] ダイアログボックスにこれらのバージョンが一覧表示され、以下の操作を実行できます。

- アイテムの前のバージョンを表示する
- 前のバージョンを別の名前で保存する
- アイテムを前のバージョンへ復元する
- 2つのバージョンの差異を表示する

注: バージョン履歴機能は、Studioでのローカルな作成作業にのみ適用され、ソースコントロール管理リポジトリに保存されたバージョンには適用されません。バージョン履歴は、作業内容をソースコントロールリポジトリにコミットする前に使用します。リポジトリで共有するためにフローをコミットした場合、OOのローカルのバージョン履歴機能を前のバージョンに戻す作業に使用することは推奨されません。


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](#)」(95ページ)を参照してください。

バージョン	日時	コメント
4*	13/09/18 11:25	変更されました
3	13/09/18 11:24	変更されました
2	13/09/17 14:15	変更されました
1	13/09/13 15:23	/Base [1.0.121]/

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



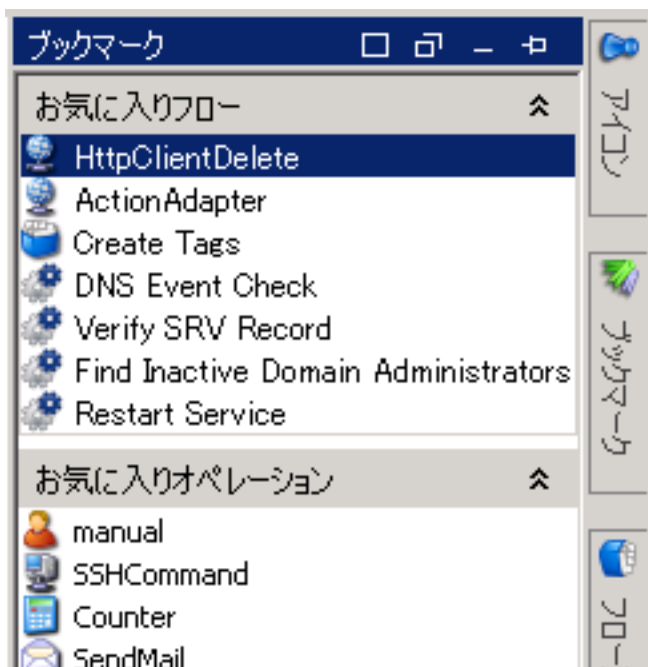
メニューアイテム	説明
バージョン	自動的に生成された、プロジェクトのバージョン番号。
日時	バージョンが変更された日付と時刻。
コメント	加えられた変更の種類。

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. 確認ウインドウで**[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。

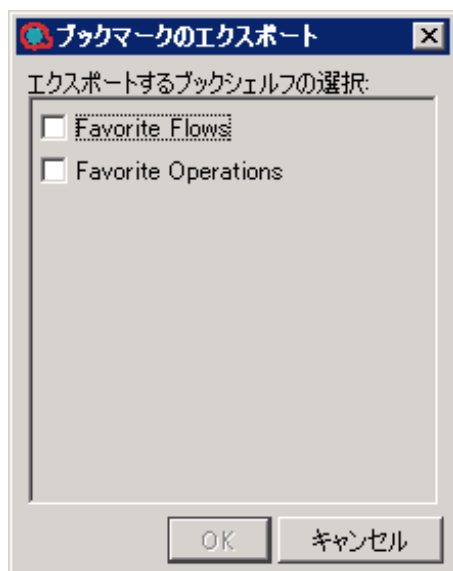
シェルフの折りたたみ/展開

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。

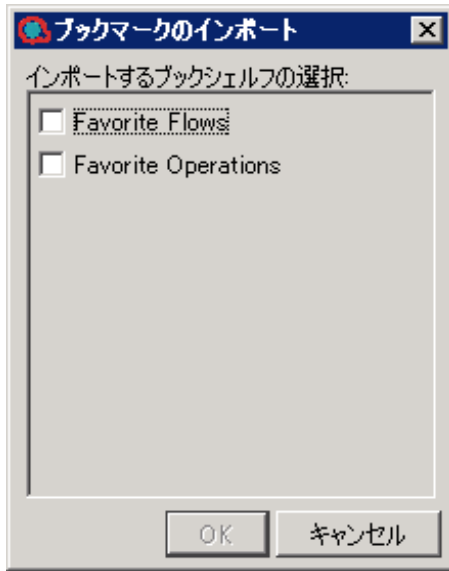
1. **[ブックマーク]**ペイン内を右クリックし、**[エクスポート]**をクリックします。
2. **[ブックマークのエクスポート]**ダイアログボックスで、エクスポートするシェルフを選択します。



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、[参考資料](#)を参照してください。

バージョン10.51以降では、複数のワークスペースで作業を行うことができます。したがって、各ワークスペースにはそれぞれ固有の**studio.properties**ファイルがあり、これらは`.oo`フォルダーの下に置かれます。

- 新しいワークスペースすべてに同じプロパティを使用する場合、**Studio\conf**の下にある**studio.properties.template**ファイルを編集することができます。
- ワークスペースごとに異なる設定を構成するために、各**studio.properties**ファイルを個別に編集できます。これらのファイルは、各ワークスペース内の`.oo`フォルダーの下にあります。

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。

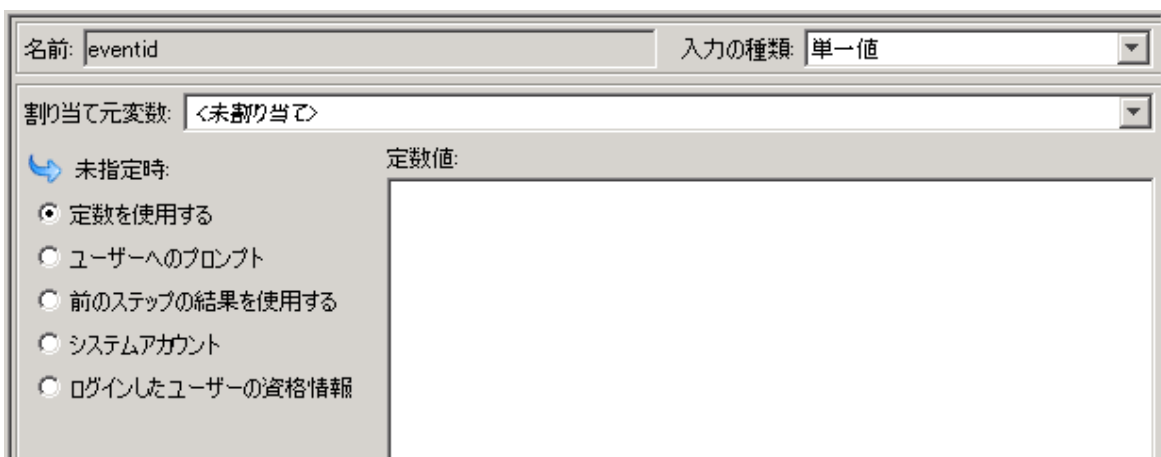
入力値のどちらかまたは両方の入力ソースと同じ名前前のフロー変数を割り当てるようにStudioのデフォルトの動作を構成できます。この操作は、**Studio.properties**ファイルで行います。

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かリストの (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。

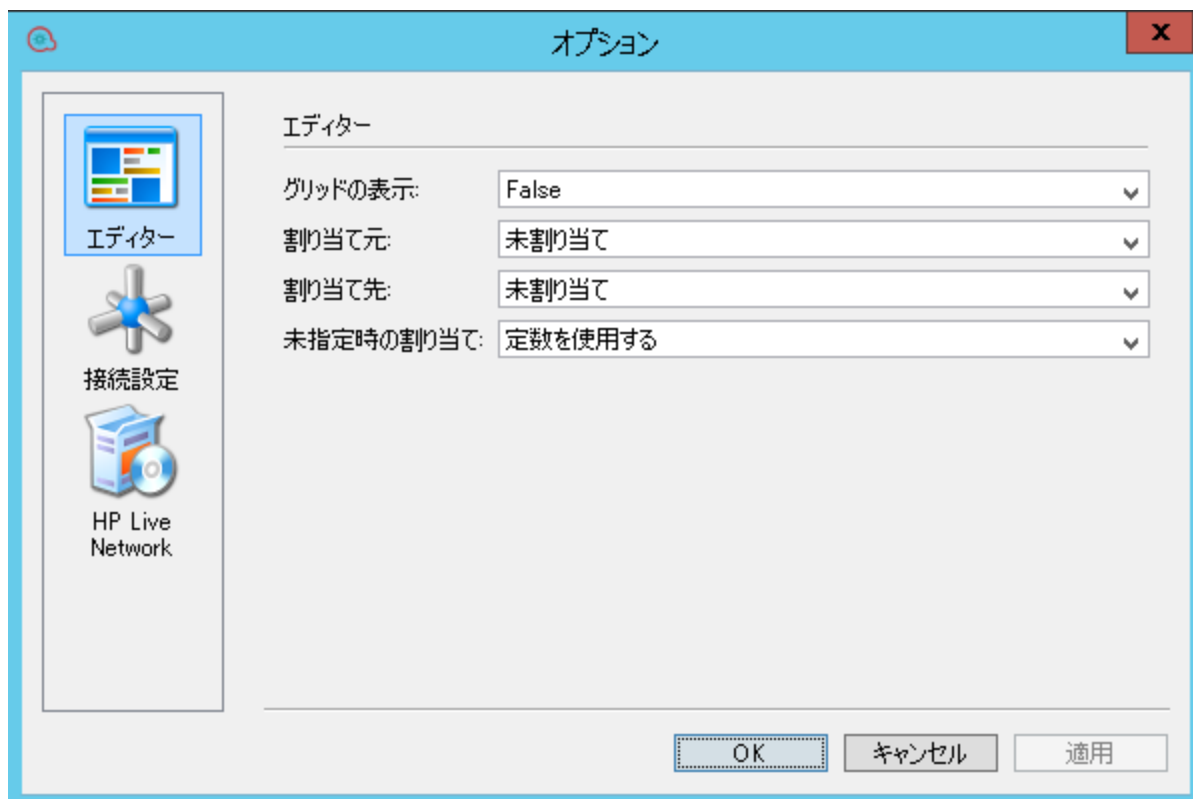
実行する操作

Studioのプロパティを編集します。

1. Windowsエクスプローラーで、<OO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](#)」(408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]に設定されます。

[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、<OO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。

dharma.studio.ui.inputinspector.assignfrom.selected=not-assigned

dharma.studio.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 10.5x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10.5x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、<OO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 10.5x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10.5x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、<OO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、<OO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、<OO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=&#x2022;
```

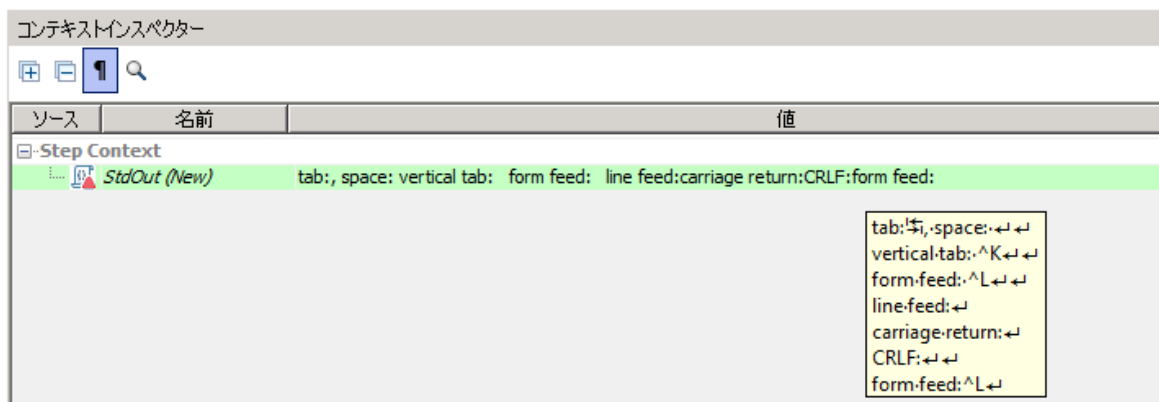
```
dharmastudio.ui.tooltip.showwhitespacecharacters.tab=&#x2022;
```

```
dharmastudio.ui.tooltip.showwhitespacecharacters.enter=&#x2022;
```

```
dharmastudio.ui.tooltip.showwhitespacecharacters.otherwhitespace=&#x2022;
```

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HPE 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、<OO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、<OO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、HPE 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、<OO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。
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	
dharmarepo.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。
dharmastudio.ui.lookandfeel	Windows	Studioアプリケーションの外観。
dharmastudio.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。
dharmastudio.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。
dharmastudio.ui.tooltip.showwhitespacecharacters.space	•	次のいずれかの値です。 •、•
dharmastudio.ui.tooltip.showwhitespacecharacters.tab	↹	次のいずれかの値です。 ↹、⇥、→、¬
dharmastudio.ui.tooltip.showwhitespacecharacters.enter	↵	次のいずれかの値です。 ↵、↲、¶
dharmastudio.ui.tooltip.showwhitespacecharacters.otherwhitespace	•	次のいずれかの値です。 •、•、☯
dharmascripting.template.Rhino dharmascripting.template.Nashorn	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	

パラメーター	デフォルト値	説明
		<p>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: <code>myData = inputName;</code></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、次の手順を実行します。 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、次の手順を実行します。 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、次の手順を実行します。 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)、次の手順を実行します。 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、次の手順を実行します。 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、次の手順を実行します。 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)、次の手順を実行します。 scriptletResponse = "success";</p> <p>オペレーションの結果を設定するには、次の手順を実行します。 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、次の手順を実行します。 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。 scriptletContext.put("LocalVariable", "LocalValue");</p>

パラメーター	デフォルト値	説明
		<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>
dharmastudio.ui.inputinspector.assignfrom.selected	説明を参照してください	<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>
dharmastudio.ui.inputinspector.assignto.selected	説明を参照してください	<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>
default.behavior.define.input	説明を参照してください	<p>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に使用するデフォルトの動作を設定できます。</p> <p>有効な値:</p> <p>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</p> <p>use_constant - Studio 10.20をインストールした場合のデフォルト</p>
dharmastudio.ui.activegrid	false	<p>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。</p>
engine.assigner.trigger.repeatInterval	10.	

パラメーター	デフォルト値	説明
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。
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コンテンツオファリングサービスの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.

パラメーター	デフォルト値	説明
		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

OO 9.xからアップグレードする際のStudioのトラブルシューティング

Studioのユーザーインターフェイスアイテムの場所

OO 9.xでの作業に慣れているユーザーが、Studioのユーザーインターフェイ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？

OOではリポジトリは使用されなくなりました。ファイルはユーザーのファイルシステムにローカルで保存されるため、共同作業ではソースコントロールアプリケーションを使用することをお勧めします。

[チェックイン] ボタン、[チェックアウト] ボタン、[自分の変更/チェックアウト] ペインはどこにありますか？

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OO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OOが含まれます。プロジェクト内でこれらのオブジェクトを使用できますが、編集はできません。これらのオブジェクトを編集する場合は、[プロジェクト] ペインにコピーします。

オペレーションを作成できないのはなぜですか？

この機能は存在し、オペレーションを作成することは可能です。削除されたのは、組み込みオペレーション(HTTP、SSH、コマンドラインなど)を作成するオプションです。これらは既存のテンプレートオペレーションからコピーする必要があります。

HTTP、SSH、コマンドラインなどの組み込みオペレーションを作成することはできません。既存のテンプレートオペレーションからこれらをコピーし、プラグイン内のアクションから新しいオペレーションを作成する必要があります。詳細については、「[オペレーションの作成](#)」(366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。OO 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フォルダーが追加され、カテゴリを保存してフローを分類することができるようになりました。これは、カテゴリドメインタームの代わりに使用します。

OO 9.xと10.xのバージョン比較

タスク	OO 9.xでの処理	OO 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」 (366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。 「OO Studioのソースコントロールの使用」 (42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OOにデプロイします。 『OOインストールガイド』および『OO Application Program Interface (API) Guide』を参照してください。
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	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

タスク	OO 9.xでの処理	OO 10.xでの処理方法
	ターゲットについて複数のループを作成します。	シジ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」(297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 StudioI 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、StudioI 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」(366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。

HPE Live Networkのトラブルシューティング

HPE 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か。

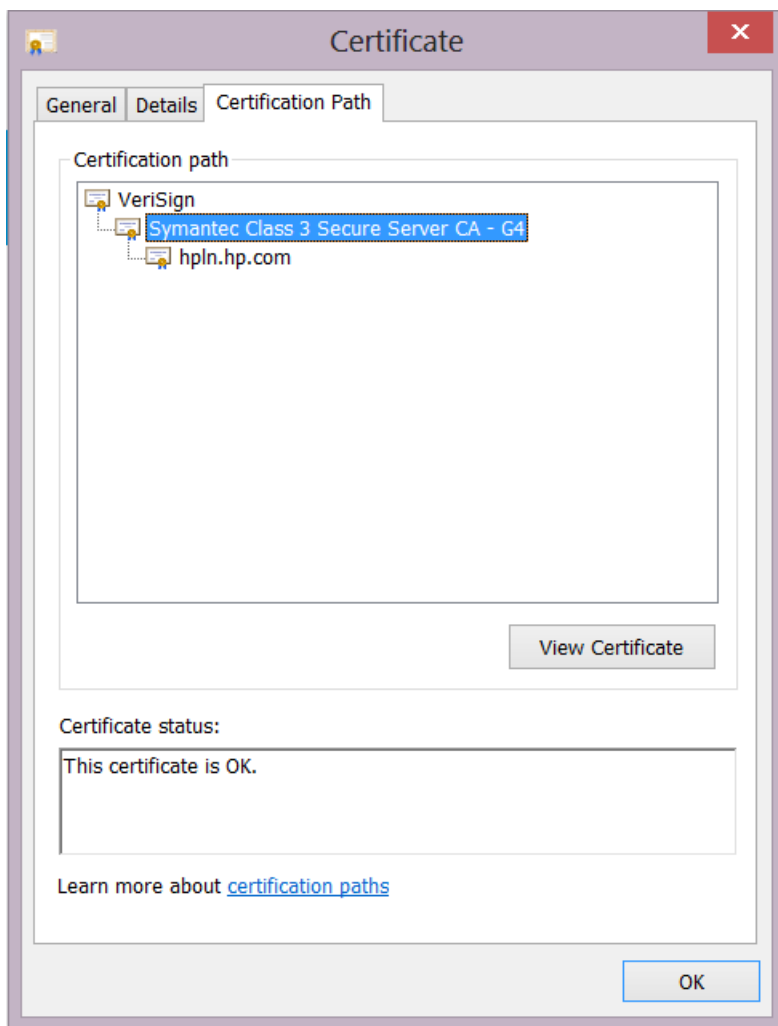
OOではリポジトリは使用されなくなりました。ファイルはユーザーのファイルシステムにローカルで保存されるため、共同作業ではソースコントロールアプリケーションを使用することをお勧めします。

HPE 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。

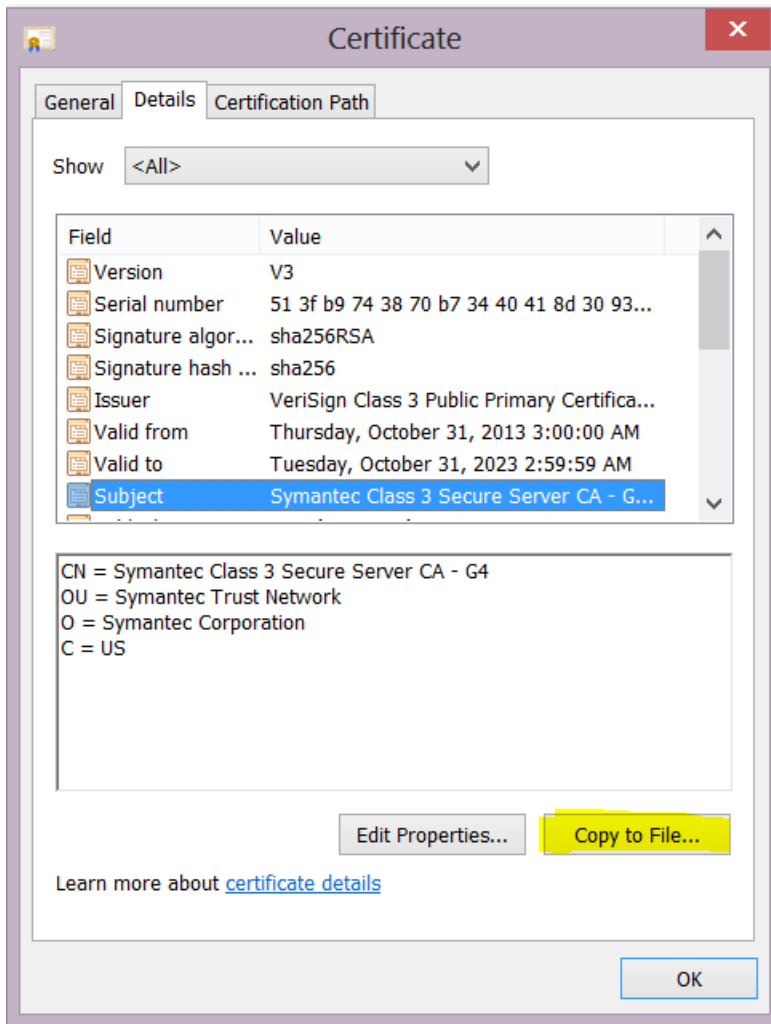
```
sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to
find valid certification path to requested target
```

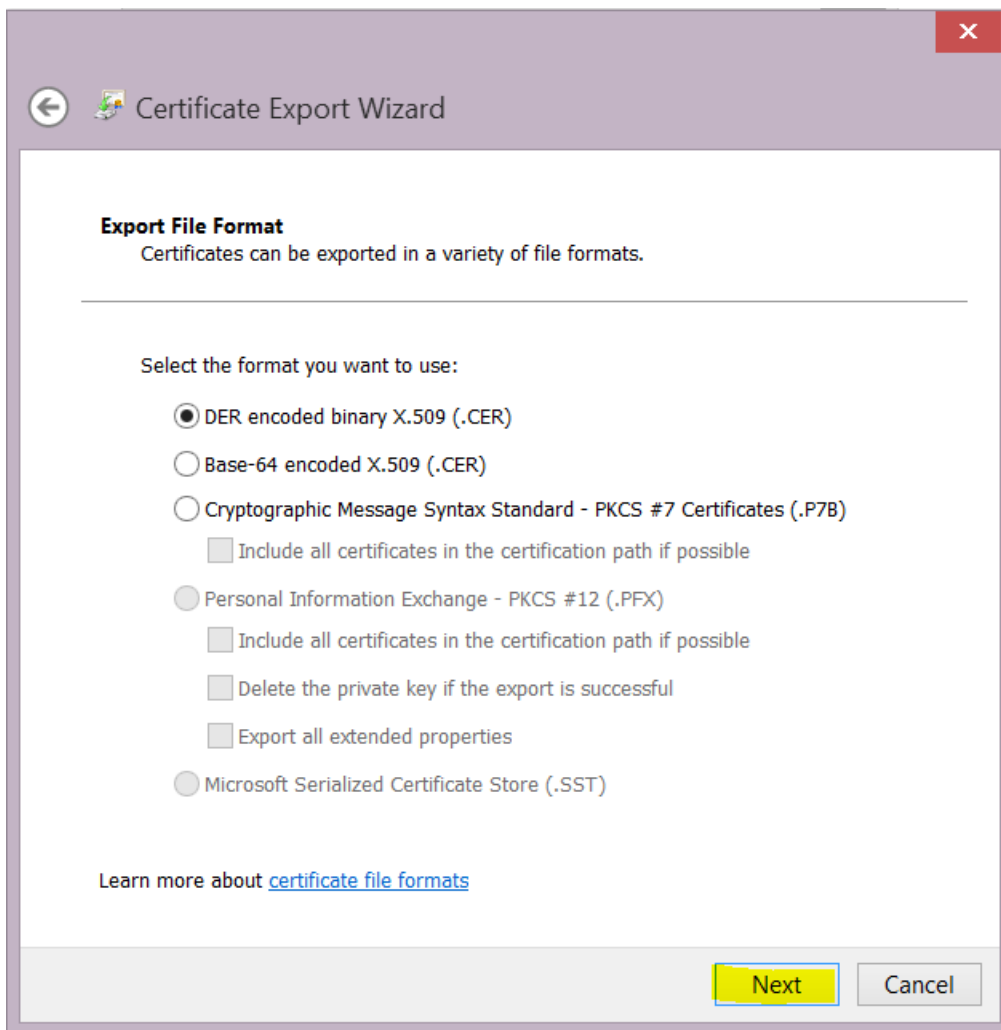
この問題を解決するには、次の手順を実行します。

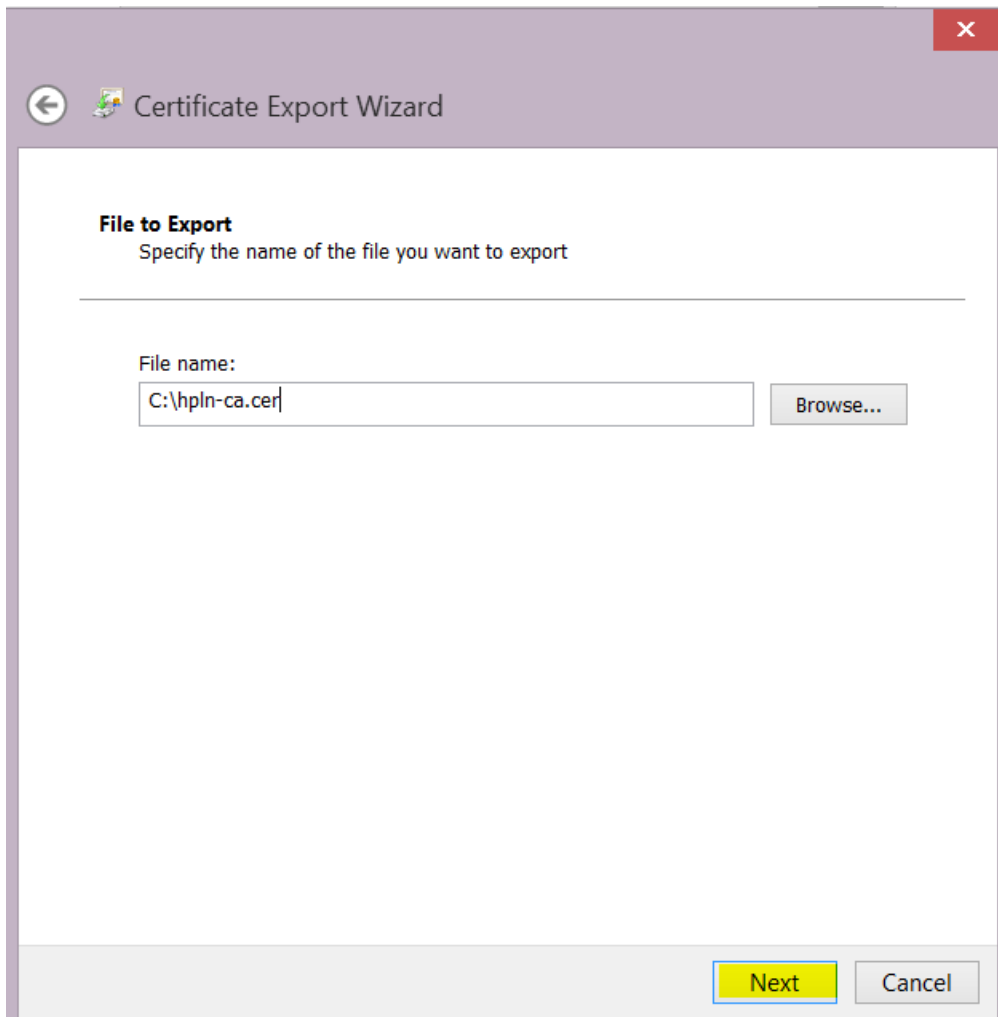
1. Studioを閉じます。
2. HPE Live Networkルート証明書 (Symantec Class 3 Secure Server CA - G4という名前のチェーンの2番目の証明書) をDERエンコードされたバイナリX.509の形式でhpln-ca.cerという名前のファイルにエクスポートします。

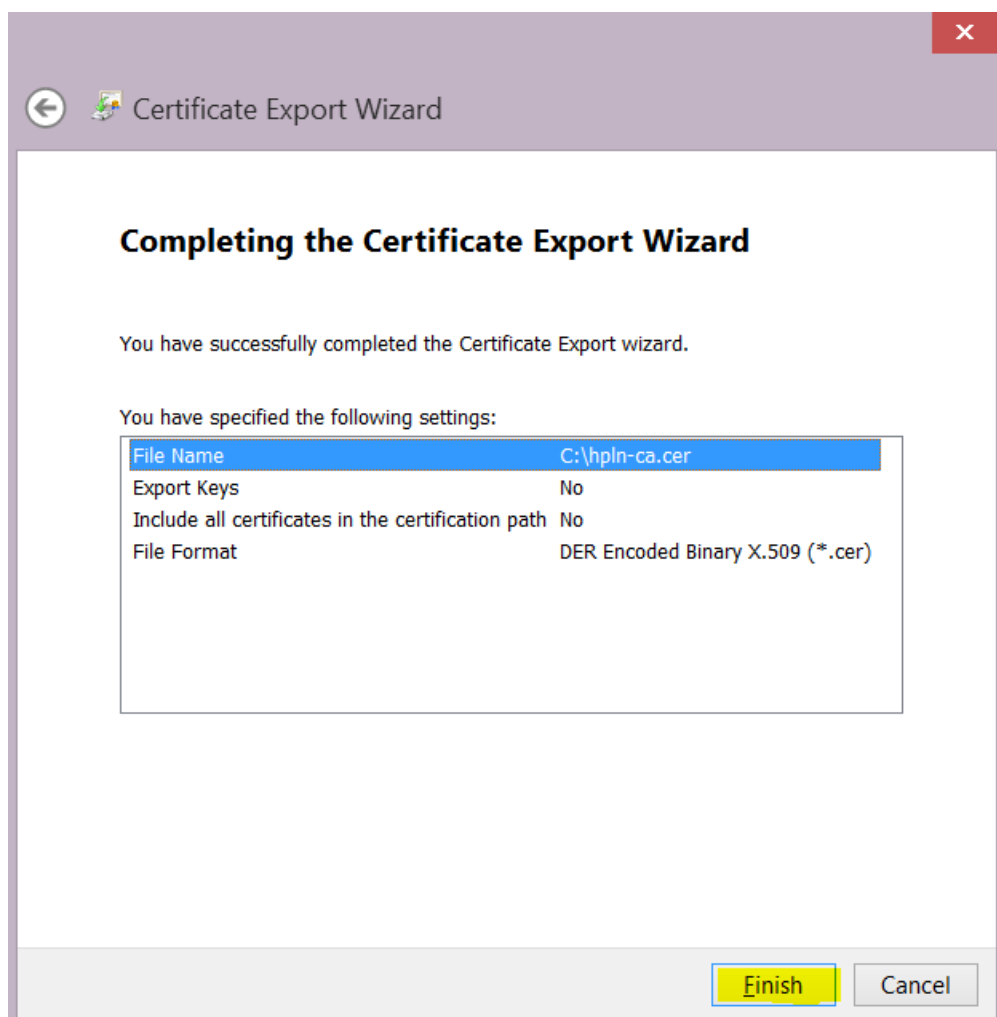


以下に示すように、ブラウザーを使用して<https://api.hpln.hp.com/hpln>にアクセスして、ルート証明書をエクスポートすることができます。









- 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フォルダーにコピーします。
- 次のコマンドを使用して証明書をStudioのTrustStore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:

- KeyStore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 **Djavax.net.ssl.trustStorePassword**パラメーターの値を使用します。
- Studio TrustStoreのパスワードを変更するには、“oo”フォルダーの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、TrustStore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か。

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、次の手順を実行して、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。パスの区切り記号として **/** 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。

注: この2番目の方法を選択した場合、次の点に注意する必要があります。

複数のワークスペースを使用している場合に、Gitの場所プロパティを新規ワークスペースのそれぞれに追加するには、**Studio\conf\studio.properties.template**にあるテンプレートプロパティファイルを編集する必要があります。そうしないと、新規ワークスペースに切り替えるたびに、**.ool\Studio.properties**ファイルで新規ワークスペースのGitの場所を設定する必要があります。

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。これはStudioで検証されているバージョンです。その他のバージョンも正常に動作する可能性はありますが、正式にはサポートされていません。

SVNのトラブルシューティング

自分のプロジェクトを更新したときにほかのユーザーの変更を削除してしまった場合はどうすればよいのでしょうか？

- 次の状況では、更新中に競合が発生することがあります。
 - a. User1がFlow1を編集してコミットします。
 - b. User2がFlow1を編集して更新します。

User2はローカルに変更されたファイルに対する更新を受け取ることになり、競合が発生します。

Studioはこのような競合を、更新側のユーザーのコピーで解決します。つまり、このような場合、User1がFlow1に対して行ったすべての変更は削除されます。

このような状況を回避するには、アイテムを編集する前にアイテムを必ず更新し、アイテムをロックすることをお勧めします。

- **ファイルシステム内で直接変更を行ったら、何らかの問題が発生した**

ファイルシステムで直接ディレクトリを作成してファイル名を変更することはできますが、これはお勧めできません。Studio内からすべてのタスクを実行することをお勧めします。

ファイルシステムで行った変更が原因で問題が発生した場合、SlickSVNやTortoiseSVNなどの外側にあるSVNツールを使用してワークスペースのクリーンアップを実行して、これらの変更を元に戻すことをお勧めします。

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。

