スクリプト モジュールの使用

この章は、次の項で構成されています。

- スクリプト モジュールの使用，1 ページ
- スクリプト モジュールの追加，2 ページ
- ライブラリの追加，2 ページ
- JAR ファイル，4 ページ
- 値のリスト，4 ページ
- 表形式のレポート，7 ページ
- コンテキスト マッピング，12 ページ
- スクリプト モジュールのインポートおよびエクスポート，17 ページ

スクリプト モジュールの使用

基本的に、スクリプト モジュールは、カスタム スクリプト、JAR ファイル、および入力コントロールを収容するコンテナです。スクリプト モジュールにこれらの要素を含めることにより、カスタム ワークフロー タスクに統合可能なライブラリ スクリプトを追加するなどといった、カスタマイズされたアクションを実行できます。スクリプト モジュールをエクスポートして、別のアプライアンスにインポートすることができます。インポートされたモジュールに含まれる登録済みスクリプトは、新しいアプライアンスで使用可能になります。

次の表に、スクリプト モジュールを使用して実行できるアクションの説明を記載します。

タスク	参照先
スクリプト モジュールの追加	スクリプト モジュールの追加，（2 ページ）
ライブラリの追加	ライブラリの追加，（2 ページ）
jar の追加	JAR ファイルの追加，（4 ページ）

タスク	参照先
値のリスト（LOV）の追加	値のリスト，（4 ページ）
表形式のレポートの追加	表形式のレポート，（7 ページ）
コンテキスト マッピングの追加	コンテキスト マッピング，（12 ページ）
スクリプト モジュールのエクスポート	ワークフロー、カスタム タスク、スクリプト モジュール、およびアクティビティのエクスポート
スクリプト モジュールのインポート	ワークフロー、カスタム タスク、スクリプト モジュール、およびアクティビティのインポート

スクリプト モジュールの追加

新しいスクリプト モジュールを作成するには、次の手順を実行します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプト モジュール（Script Modules）] タブをクリックします。

ステップ 3 [追加（Add）] をクリックします。

ステップ 4 [モジュール情報（Modules Information）] 画面で、次の情報を入力します。

名前	説明
[モジュール名（Module Name）] フィールド	スクリプト モジュールの名前。
[モジュールの説明（Module Description）] フィールド（オプションですが、入力することを推奨）	スクリプト モジュールの説明。

ステップ 5 [送信（Submit）] をクリックします。

ライブラリの追加

タスク ライブラリを追加するには、次の手順に従います。

はじめる前に

ライブラリを追加するには、その前に、スクリプトモジュールが必要です。スクリプトモジュールの追加, （2 ページ）を参照してください。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプト モジュール（Script Modules）] タブをクリックします。

ステップ 3 [スクリプト モジュール（Script Modules）] ペインで、ライブラリを追加するスクリプト モジュールをダブル クリックします。

ステップ 4 [ライブラリ（Library）] タブをクリックします。

ステップ 5 [追加（Add）] をクリックします。

ステップ 6 [ライブラリ情報（Library Information）] 画面で、次の情報を入力します。

名前	説明
[名前（Name）] フィールド	スクリプト モジュールの名前。
[説明（Description）] フィールド	スクリプト モジュールの説明。
[スクリプト（Script）] テキスト ボックス	この領域にライブラリ スクリプトを追加します。

ステップ 7 [送信（Submit）] をクリックします。

ライブラリへのアクセス

ライブラリには次の方法でアクセスできます。

別のライブラリからライブラリを呼び出す

loadLibrary("Module Name>/<Library Name>") という構文を使用して別のライブラリからライブラリを呼び出します。

次に例を示します。

```
ImportPackage(java.lang);

function test1(){
 logger.addInfo("test1");
 loadLibrary("Test_Module/testlib1");
}
test1();
```

カスタム タスクからライブラリを呼び出す

カスタムワークフロータスクを作成するときに、上記の例と同じ構文を使用してライブラリを呼び出すことができます。カスタムタスクからライブラリを呼び出す方法を含め、カスタムタスクの使用方法については、『Cisco UCS Director Custom Task Getting Started Guide』を参照してください。

JAR ファイル

スクリプト モジュールには、.jar ファイルを登録できます。ファイルを登録すると、スクリプト モジュールを使用して、ライブラリまたはカスタム タスクから .jar の内容を呼び出せるようになります。

JAR ファイルの追加

スクリプトに .jar ファイルを追加するには、次の手順を実行します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプト モジュール（Script Modules）] タブをクリックします。

ステップ 3 [スクリプト モジュール（Script Modules）] ペインで、jar ファイルを追加するスクリプト モジュールをダブル クリックします。

ステップ 4 [jar] タブをクリックします。

ステップ 5 [追加（Add）] をクリックします。

ステップ 6 [Jar の追加（Add Jar）] 画面で、[参照（Browse）] をクリックします。

ステップ 7 ローカル フォルダから、アップグレードする .jar ファイルを選択します。

ステップ 8 [送信（Submit）] をクリックします。

値のリスト

値のリスト（LOV）とは、Cisco UCS Director GUI から呼び出したり、タスクまたはワークフローの入力として呼び出したりできる、検索可能なリストのことです。たとえば、タスクまたはワークフローの入力のデータ タイプを選択する際に、LOV を呼び出して選択肢を表示することができます。

独自の値のセットを表示する LOV をタスクやワークフローの入力として作成し、作成した LOV をスクリプト モジュールに保存できます。

値のリストの追加

値のリスト（LOV）を作成するには、次の手順を実行します。

はじめる前に

スクリプト モジュールを作成します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3 [スクリプトモジュール（Script Module）] ペインで、使用するスクリプト モジュールをダブルクリックします。

ステップ 4 [LOV（LOVs）] タブをクリックします。

ステップ 5 [追加（Add）] をクリックします。

ステップ 6 [LOV 情報（LOV Information）] ダイアログボックスで、次の手順を実行します。

名前	説明
[名前（Name）] フィールド	LOV の名前。
[説明（Description）] フィールド	LOV の説明（オプション）。
[スクリプト（Script）] テキスト ボックス	LOV プロバイダー登録用のテンプレート コードが示されます。次の手順の説明に従って `getDataProvider()` メソッドに変更を加えることで、独自の実装を提供します。 **（注）** 次の手順で説明されているスクリプトの行だけを編集します。`createLOV()`、`registerLOV()`、または `registerGlobalInputs()` メソッドは編集しないでください。編集すると、LOV が機能しなくなる可能性があります。

ステップ 7 [スクリプト（Script）] テキスト ボックスで、次のようにして名前と値のペアをリストに追加します。

a) `getDataProvider()` 関数を見つけます。この関数は以下のように示されます。

```
function getDataProvider(){
var lovRegistry = LOVProviderRegistry.getInstance();
var lovProvider = new com.cloupia.service.cIM.inframgr.forms.wizard.LOVProviderIf({
getLOVs : function(session) {
//provide your own implementation for Lovprovider
var formlovs = [];
var formlov = new FormLOVPair("Flex","1");
formlovs[0] = formlov;
```

```
var formlov = new FormLOVPair("Generic","2");
formlovs[1] = formlov;
return formlovs;
//End of implementation for Lovprovider
}
});
return lovProvider;
}
```

b) 強調表示されているテキストを変更して、独自の名前と値のペアを定義します。

ステップ 8 [送信（Submit）] をクリックします。

値のリストの編集

既存の値のリスト（LOV）を編集するには、次の手順を実行します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3 [スクリプトモジュール（Script Module）] ペインで、編集する LOV を含むスクリプト モジュールをダブルクリックします。

ステップ 4 [LOV（LOVs）] タブを選択します。

ステップ 5 編集する LOV を選択します。

ステップ 6 [編集（Edit）] をクリックします。

ステップ 7 変更が許可されている LOV フィールドを編集します。

名前	説明
[名前（Name）] フィールド	LOV の名前が表示されます。 （注） 既存の LOV の名前を変更することはできません。
[説明（Description）] フィールド	LOV の説明。
[スクリプト（Script）] テキスト ボックス	LOV プロバイダー登録用のテンプレート コードを編集します。`getDataProvider()` メソッドに変更を加えて、独自の実装を提供します（値のリストの追加, （5 ページ）を参照）。 （注） `createLOV()`、`registerLOV()`、または `registerGlobalInputs()` メソッドは編集しないでください。編集すると、LOV が機能しなくなる可能性があります。

ステップ 8 [送信（Submit）] をクリックします。

値のリストの削除

はじめる前に

すべてのカスタム タスクとワークフローから値のリスト（LOV）への参照を削除します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3 [スクリプトモジュール（Script Module）] ペインで、削除する LOV を含むスクリプト モジュールをダブルクリックします。

ステップ 4 [LOV（LOVs）] タブをクリックします。

ステップ 5 削除する LOV を選択します。

ステップ 6 [削除（Delete）] をクリックします。

ステップ 7 [送信（Submit）] をクリックします。

表形式のレポート

表形式のレポートとは、Cisco UCS Director GUI から呼び出したり、タスクまたはワークフローの入力として呼び出したりできる、カラム型リストのことです。たとえば、[ワークフロー（Workflows）] タブでワークフローを選択する際に、表形式のレポートを呼び出して選択肢を表示できます。

独自の定義済み値のセットをタスクまたはワークフローの入力として表示する表形式のレポートを作成できます。作成した表形式のレポートは、スクリプト モジュールに保存できます。

表形式のレポートの追加

はじめる前に

スクリプト モジュールを作成します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3 [スクリプトモジュール（Script Module）] ペインで、使用するスクリプト モジュールをダブルクリックします。

ステップ 4 [表形式のレポート（Tabular Reports）] タブをクリックします。

ステップ 5 追加（+）ボタンをクリックします。

ステップ 6 [表形式のレポート情報（Tabular Report Information）] ダイアログボックスで、次の手順を実行します。

名前	説明
[表形式レポートの名前（Tabular Report Name）] フィールド	表形式レポートの名前。
[説明（Description）] フィールド	LOV の説明（オプション）。
[列のエントリ（Column Entries）] リスト	追加（+）ボタンをクリックして、次のフィールドに値を入力します。
[列名（Column Name）] テキスト ボックス	列の名前。
[列タイプ（Column Type）] ドロップダウン リスト	列のエントリのデータ型。次のオプションがあります。 • テキスト（Text） • 整数（Integer） • 長整数型（Long） • 倍精度浮動小数点型（Double）

名前	説明
[列のエントリ（Column Entries）] テキスト ボックス	列にオプションを追加するには、このチェックボックスをオンにします。次のオプションがあります。 • [管理列（Management column）] チェックボックス：列のエントリを 1 つだけにする場合はオンにする必要があります。 • [列を表示（Display column）] チェックボックス：列のエントリを 1 つだけにする場合はオンにする必要があります。 • [フィールドを非表示（Hide the Field）] チェックボックス：列のエントリを任意にする場合は、このオプションをオンにします。

ステップ 7 [送信（Submit）] をクリックします。

ステップ 8 作成する列のすべてに対して、上記の 2 つのステップを繰り返します。

ステップ 9 [スクリプトを表示（Show Script）] チェックボックス：[スクリプト（Script）] テキスト エリアを表示して編集できるようにするには、このチェックボックスをオンにします。
列を作成するときに、[スクリプト（Script）] テキスト エリアの列の作成スクリプトが自動的に生成されます。

レコードは 1 つの行からなり、その行には列ごとに 1 つの値が格納されます。レコードを作成するには、[スクリプト（Script）] テキスト エリアを編集します。

（注） [スクリプト（Script）] テキスト エリア内の `//START OF YOUR IMPLEMENTATION` の前および `//END OF YOUR IMPLEMENTATION` の後を編集または削除しないでください。.

a) レコード エントリを作成するには、次の手順を実行します。
表形式のレポートで定義した各列エントリに対する値を作成します。スクリプトの行 `//START OF YOUR IMPLEMENTATION.` と行 `//END OF YOUR IMPLEMENTATION.` の間のセクションに以下の値を作成します。次のリストに示すように、[列エントリ（Column Entry）] の [列タイプ（Column Type）] に応じて正しい関数を使用します。

- テキスト：`model.addTextValue("value");`
- 整数：`model.addNumberValue(42);`
- 長整数型：`model.addLongNumberValue（1000）;`
- 倍精度浮動小数点型：`model.addDoubleValue（8.6）;`

`model.completedRow();` を使用してレコード エントリの関数呼び出しを区別します。

ステップ 10 [送信（Submit）] をクリックします。

例：列のエントリとしてのレコード エントリの作成

表形式レポートに 2 つの列のエントリを作成するとします。最初の列のエントリの列名は `Name`、列タイプは `Text` です。2 番目の列のエントリの列名は `Department`、列タイプは `Long` です。表形式レポートに列のエントリを作成すると、システムによって、それらの列を作成するスクリプト内に以下の関数呼び出しが生成されます。

```
function implementationForTabularReport(report)
{
var model = new TabularReportInternalModel();
model.addTextColumn("Name","Name");
model.addNumberColumn("Department","Department");
model.completedHeader();
//START OF YOUR IMPLEMENTATION.



//END OF YOUR IMPLEMENTATION.
model.updateReport(report);
}
function getSelectionColumnId(){
return "1";
}
function getDisplayColumnId(){
return "0";
}
```

スクリプト中央に `//START OF YOUR IMPLEMENTATION.` で始まり `//END OF YOUR IMPLEMENTATION.` で終わるセクションがあります。この 2 つの行の間にレコード エントリを作成します。 レコード エントリにより、値が列のエントリに割り当てられます。

以下の例では、テキスト値 `"Smith"` が列名 `Name` の列に割り当てられ、長整数型の数値 `40` が列名 `Department` の列に割り当てられます。関数呼び出し `model.completedRow();` は、このレコード エントリの終了を示します。

```
function implementationForTabularReport(report)
{
var model = new TabularReportInternalModel();
model.addTextColumn("Name","Name");
model.addLongNumberColumn("Department","Department");
model.completedHeader();
//START OF YOUR IMPLEMENTATION.

model.addTextValue("Smith");
model.addLongNumberValue(40);
model.completedRow();

//END OF YOUR IMPLEMENTATION.
model.updateReport(report);
}
function getSelectionColumnId(){
return "1";
}
function getDisplayColumnId(){
return "0";
}
```

表形式のレポートの編集

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3 [スクリプト モジュール（Script Module）] ペインで、表形式のレポートを編集するスクリプト モジュールをダブルクリックします。

ステップ 4 [表形式のレポート（Tabular Report）] タブを選択します。

ステップ 5 編集する表形式のレポートの名前を選択し、[編集（Edit）] をクリックします。

ステップ 6 [編集（Edit）] ダイアログボックスで、変更するフィールドを編集します。

（注） [管理（Management）] 列または [表示（Display）] 列の [列タイプ（Column Type）] ドロップダウン リストの値を変更できます。[管理（Management）] 列または [表示（Display）] 列の [列タイプ（Column Type）] チェックボックスをオフにする場合は、既存の列エントリを削除し、新しく作成する必要があります。

ステップ 7 [送信（Submit）] をクリックします。

表形式のレポートの削除

はじめる前に

すべてのカスタム ワークフロー タスクから表形式のレポートへの参照を削除します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3 [スクリプトモジュール（Script Module）] ペインで、削除する表形式のレポートが含まれているスクリプト モジュールをダブルクリックします。

ステップ 4 [表形式のレポート（Tabular Report）] タブを選択します。

ステップ 5 削除する表形式のレポートの名前を選択します。

ステップ 6 [削除（Delete）] をクリックします。

ステップ 7 [送信（Submit）] をクリックします。

コンテキスト マッピング

コンテキストワークフローマッピングは、ページ上のワークフローにマッピングされたアクションラベルからなります。このアクションラベルをクリックすると、ワークフローがトリガーされます。

（注） 厳密には、コンテキスト マッピングはオーケストレーションに含まれません。これは、Cisco UCS Director ユーザ インターフェイスに対する変更です。コンテキスト マッピングがここに含まれているわけは、スクリプト モジュールの有用なアプリケーションであるためです。

コンテキスト マッピング モジュールを使用すると、コンテキスト ワークフロー マッピングを動的に追加することができます。この場合、ページのアクション ラベルを作成し、そのアクション ラベルにワークフローを割り当てることで、Cisco UCS Director UI をカスタマイズできます。

コンテキストマッピングを作成するには、アクションラベルを追加するページの名前を把握している必要があります。ページの名前およびその他のメタデータは、Cisco UCS Director で開発者メニュー を有効にすると、[情報（Information）] ダイアログに表示されます。

コンテキスト マッピングを作成する前に、次のセクションの説明に従って、メタデータを有効にします。

メタデータの有効化

レポート メタデータにアクセスするには、開発者メニューを有効にします。

（注） レポート メタデータでのレポートという言葉は、Cisco UCS Director ユーザ インターフェイス内のページを指します。

開発者メニューを有効にするには、次の手順を実行します。

ステップ 1 Cisco UCS Director で右上にあるログイン名をクリックします。
たとえば、admin としてログインしている場合、Cisco UCS Director の右上に admin と表示されます。

ステップ 2 [ユーザ情報（User Information）] ダイアログボックスで、[詳細設定（Advanced）] タブをクリックします。

ステップ 3 [[開発者]メニューの有効化(このセッション)（Enable Developer Menu (for this session)）] チェックボックスをオンにします。
現在のセッションの間中、[REST API ブラウザ（REST API Browser）] が [オーケストレーション（Orchestration）] メニューでアクティブになり、このセッションで開かれるレポート ビューで [レポート メタデータ（Report Metadata）] オプションが利用可能になります。

ヒント　[詳細（Advanced）] タブにはアカウントの REST API アクセス キー コードも表示されます。

ステップ 4　[ユーザ情報（User Information）] ダイアログボックスを選択します。

次の作業

任意のレポートで [レポート メタデータ（Report Metadata）] アクションをクリックして、レポート名を含むレポート メタデータを表示します。

コンテキスト マッピングの追加

はじめる前に

- スクリプト モジュールを作成します。
- アクション ラベルを使用するページを識別します。レポート メタデータを使用して、そのページの名前を見つけます。メタデータの有効化, （12 ページ）を参照してください。コンテキスト マッピングを作成する際は、そのページの名前を使用します。

ステップ 1　メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2　[スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3　[スクリプトモジュール（Script Module）] ペインで、使用するスクリプト モジュールをダブルクリックします。

ステップ 4　[コンテキストマッピング（Context Mapping）] タブをクリックします。

ステップ 5　[追加（Add）] をクリックします。

ステップ 6　[コンテキストマッパーの追加（Add Context Mapper）] ダイアログボックスで、次の手順を実行します。

名前	説明
[名前（Name）] テキスト フィールド	コンテキスト マッピングの一意の名前を入力します。システムによって同じ名前のコンテキスト ワークフロー マッピングが作成されます。
[レポート名（Report Name）] テキスト フィールド	アクション ラベルを追加するレポート（ページ）の名前を入力します。この名前は、レポートの [レポート メタデータ（Report Metadata）] の *reportName* フィールドの値です。

名前	説明
[表形式フィールドを選択（Select Tabular Field）] ドロップダウン リスト	[選択（Select）] をクリックして [選択（Select）] ダイアログボックスを表示します。使用する表形式フィールドの横にあるボックスにチェックマークを入れます。
[説明（Description）] テキスト フィールド	コンテキスト マッピングの説明。
[スクリプト（Script）] テキスト エリア	コンテキスト マッピングを作成して、そのコンテキスト マッピングを既存のレポートに関連付けるスクリプト。 （注） システムによって、フィールドに入力した情報でスクリプトが更新されます。スクリプトを編集する必要はありません。

ステップ 7 [送信（Submit）] をクリックします。
[ステータス（Status）] 列に [成功（Success）] または [失敗（Fail）] と表示されます。

ステップ 8 結果が [失敗（Fail）] の場合は、コンテキスト マッピングを編集し、正しい**レポート名**と**表形式フィールド**が入力されていることを確認してから、再度 [送信（Submit）] をクリックします。

ステップ 9 結果が [成功（Success）] の場合は、メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 10 [コンテキストワークフローマッピング（Context Workflow Mapping）] タブをクリックします。
作成したコンテキスト マッピングが [マッピング名（Mapping Name）] 列にリストされます。

ステップ 11 作成したコンテキスト ワークフロー マッピングをクリックします。

ステップ 12 [編集（Edit）] をクリックします。

ステップ 13 [ワークフローマッピングの編集（Edit Workflow Mappings）] ダイアログボックスで、[ワークフローの追加（Add Workflow）] をクリックします。

ステップ 14 [ワークフロー 1（Workflow 1）] ダイアログボックスで、次の手順を実行します。

名前	説明
[選択が必要です（Selection Required）] チェックボックス	このボックスをオンにすると、[ワークフロー（Workflow）] ドロップダウンリストに使用可能な値が表示されなくなります。ワークフロー ドロップダウン リストにワークフローが表示されるようにするには、[ワークフロー デザイナ（Workflow Designer）] ダイアログボックスで [コンテキスト ワークフロー マッピング（Context Workflow Mapping）] を実行します。 このボックスをオンにしていない場合は、残りのフィールドに情報を入力します。

名前	説明
[レポート名（Report Name）] テキスト フィールド	アクション ラベルを追加するレポート（ページ）の名前を入力します。この名前は、レポートの [レポート メタデータ（Report Metadata）] の *reportName* フィールドの値です。
[アクション ラベル（Action Label）]テキスト フィールド	ワークフローを関連付けるためのアクション ラベルの名前を入力します。一意の名前である必要があります。
[ワークフロー（Workflow）] ドロップダウン リスト	ユーザがレポートのアクション ラベルをクリックしたときに実行されるワークフローを選択します。 [選択が必要です（Selection Required）] チェックボックスをオンにしていると、新しいワークフローが作成されます。[選択が必要です（Selection Required）] チェック ボックスをオンにしていない場合は、このドロップダウンリストに表示されるワークフローを選択します。
[選択が必要です（Selection Required）] チェックボックス	コンテキスト マッピングを作成して、そのコンテキスト マッピングを既存のレポートに関連付けるスクリプト。 （注） システムによって、フィールドに入力した情報でスクリプトが更新されます。スクリプトを編集する必要はありません。
[認定ユーザ タイプ（Authorized User Types）] ダイアログボックス	[認定ユーザ タイプ（Authorized User Types）] ダイアログボックス：[項目の選択（Select Items）] ダイアログボックスで、認定するユーザ タイプごとにチェックボックスをクリックします。選択できるタイプは、次のとおりです。 • サービスエンドユーザ（Service End-User） • グループ管理者（Group Admin） • システム管理者（System Admin） • オペレータ/その他の管理者（Operator/Other Administrator）

このコンテキスト ワークフロー マッピングに追加の**ワークフロー**手順を追加するには、[ワークフローの追加（Add Workflow）] ボタンをクリックし、フィールドに値を入力します。

このコンテキスト ワークフロー マッピングから**ワークフロー**の手順を削除するには、削除するワークフローの下の [フィールドの削除（Delete Field）] をクリックします。

ステップ 15 ワークフロー マッピングが完成したら、[送信（Submit）] をクリックします。
マッピングされたレポートに、定義したアクション ラベルが表示されます。

コンテキスト マッピングの編集

コンテキスト マッピングを編集するには、次の手順を実行します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3 [スクリプト モジュール（Script Module）] ペインで、編集するコンテキスト マッピングが含まれているスクリプト モジュールをダブルクリックします。

ステップ 4 [コンテキスト マッピング（Context Mapping）] タブを選択します。

ステップ 5 編集するコンテキスト マッピングの名前を選択し、[編集（Edit）] をクリックします。

ステップ 6 [コンテキスト マッパーの編集（Edit Context Mapper）] 画面で、変更するフィールドを編集します。
[名前（Name）] フィールドは編集できません。

[表形式フィールド（Tabular Fields）] フィールドの値を選択します。この選択は必須です。前の値は維持されません。

ステップ 7 [送信（Submit）] をクリックします。

コンテキスト マッピングの削除

コンテキスト マッピングを削除するには、次の手順を実行します。

はじめる前に

すべてのワークフローとレポートからコンテキスト マッピングへの参照を削除します。

ステップ 1 メニュー バーで、[ポリシー（Policies）] > [オーケストレーション（Orchestration）] を選択します。

ステップ 2 [スクリプトモジュール（Script Module）] タブをクリックします。

ステップ 3 [スクリプトモジュール（Script Module）] ペインで、削除するコンテキスト マッピングが含まれているスクリプト モジュールをダブルクリックします。

ステップ 4 [コンテキストマッピング（Context Mapping）] タブを選択します。

ステップ 5 削除するコンテキスト マッピングの名前を選択します。

ステップ 6 [削除（Delete）] をクリックします。
削除を確認するポップアップが表示されます。

ステップ 7 [送信（Submit）] をクリックします。

スクリプトモジュールのインポートおよびエクスポート

ワークフロー、カスタム タスク、およびアクティビティをインポートおよびエクスポートするために使用する [インポート（Import）] および [エクスポート（Export）] アクションを使用して、スクリプト モジュールをインポート、エクスポートすることもできます。

スクリプト モジュールをインポートする場合は、ワークフロー、カスタム タスク、スクリプト モジュール、およびアクティビティのインポートを参照してください。

スクリプトモジュールをエクスポートする場合は、ワークフロー、カスタムタスク、スクリプトモジュール、およびアクティビティのエクスポートを参照してください。