



目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. 本書の目的
 - 2.2. 対象読者
 - 2.3. 本書の構成
- 3. IM-BIS について
 - 3.1. IM-BIS の構成
 - 3.2. IM-BIS の特徴
 - 3.3. BISフローとワークフロー
- 4. システム管理者が使用する機能
 - 4.1. 採番ルール定義を設定する
 - 4.2. 各種定義の権限管理を設定する
 - 4.3. 「IM-BIS 一覧」画面の説明（システム管理者向け）
 - 4.4. 一覧表示パターン定義を設定する
 - 4.5. フローをグループ分類する
 - 4.6. 代理管理者を設定する
 - 4.7. データソース定義を設定する
 - 4.8. インポート・エクスポートを行う
 - 4.9. 一括インポート・エクスポートを行う
 - 4.10. ルール実行履歴レポート機能を利用する
 - 4.11. 一括処理対象者変更を利用する
 - 4.12. フロー設計書を出力する
- 5. IM-BIS の詳細カスタマイズ機能
 - 5.1. 印影表示を利用する
 - 5.2. プロセス管理の参照を利用する
 - 5.3. 参照画面や特殊なタスク（ノード）で表示する画面を設定する
 - 5.4. メールや IMBox を設定する
 - 5.5. 申請者承認防止機能を利用する
 - 5.6. BIS作成種類「BISフロー」の利用可否を切り替えるための設定をする
- 6. サンプル
 - 6.1. 外部連携サンプルプログラム（JAVA）
 - 6.2. 動的処理対象者設定 外部連携（JAVA）サンプル
 - 6.3. 動的処理対象者設定 外部連携（LogicDesigner）サンプル
 - 6.4. 外部連携 処理結果メッセージ（LogicDesigner）サンプル

改訂情報

変更年月日	変更内容
2013-02-01	初版
2013-04-01	第2版 下記を追加・変更しました。 <ul style="list-style-type: none"> 「データソース定義の操作方法」 「データソース定義を設定する」
2013-08-20	第3版 下記を追加・変更しました。 <ul style="list-style-type: none"> 「データソース定義の操作方法」 「データソース定義を設定する」 「IM-BIS で作成したフローに縦配置・横配置ノードを設定する」 「IM-BIS で作成したフローへの動的承認を設定する」 「IM-BIS で外部連携として利用できるJavaプログラムの仕様」 「データマッパーでの関数向けJavaデータソース定義の復旧方法」 「IM-BIS で外部連携として利用できるWebサービスの仕様」
2014-01-01	第4版 下記を追加・変更・削除しました。 <ul style="list-style-type: none"> 「データソース定義の操作方法」 「データソース定義を設定する」 「IM-BIS で外部連携として利用できるクエリの仕様」 「IM-BIS で外部連携として利用できるJavaプログラムの仕様」 「IM-FormaDesigner アプリケーションから IM-BIS で登録したデータソース定義を利用するための権限を設定する」 「データマッパーでCSVインポートとCSVエクスポートを部分的にマッピングする場合の仕様」 「ボタン（イベント）の外部連携の仕様」 「IM-BIS で外部連携として利用できるWebサービスの仕様」 「データマッパーでのマッピングの仕様」 「データマッパーでの「通常イベント」と「特殊イベント」のリクエストとレスポンスの仕様」
2014-04-01	第5版 下記を追加・変更・削除しました。 <ul style="list-style-type: none"> 「ルール」 「IM-BIS、IM-FormaDesigner での権限の考え方」 「サブフローに関する仕様」 「BAMのテーブル情報」
2014-09-01	第6版 下記を追加・変更・削除しました。 <ul style="list-style-type: none"> 「Office365_GoogleApps」 「リクエストパラメータ、レスポンスフィールドの機能と各部の説明」 「Office 365、Google Apps 連携の接続情報を設定する」 「動的処理対象者設定に関する仕様」 「Office 365、Google Apps 連携の仕様」 「ルール」 「各種定義の権限管理を設定する」 「インポート・エクスポートを行う」 「ルール実行履歴レポート機能を利用する」
2014-12-01	第7版 下記を追加・変更・削除しました。 <ul style="list-style-type: none"> 「インポート・エクスポートを行う」 「テナントDBクエリ」

変更年月日	変更内容
2014-12-24	<p>第8版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> 「 テナントDB更新系クエリ 」 「 シェアードDB更新系クエリ 」 「 データソース定義を設定する 」 「 IM-BIS で外部連携として利用できるクエリの仕様」
2015-03-01	<p>第9版 下記を削除しました。</p> <ul style="list-style-type: none"> 「 Office365_GoogleApps 」 「 Office 365 、 Google Apps 連携の仕様 」 「 Office 365 、 Google Apps 連携の接続情報を設定する 」
2015-04-01	<p>第10版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> 「 参照画面や特殊なタスク（ノード）で表示する画面を設定する 」 「 IM-BIS 個別対応表 」
2015-08-01	<p>第11版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> 「 JAVA（非推奨） 」にJAVAプログラムの配置先についての補足を追加しました。 「付録」を「 IM-BIS 仕様書 」に移動しました。
2015-12-01	<p>第12版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> 「 参照画面や特殊なタスク（ノード）で表示する画面を設定する 」に特殊なタスク（ノード）についての説明を追加しました。 「 LogicDesigner 」を追加しました。 「 一括処理対象者変更を利用する 」を追加しました。
2016-04-01	<p>第13版 下記を追加しました。</p> <ul style="list-style-type: none"> 「 外部連携サンプルプログラム（JAVA） 」を追加しました。 「 動的処理対象者設定 外部連携（JAVA）サンプル 」を追加しました。 「 動的処理対象者設定 外部連携（LogicDesigner）サンプル 」を追加しました。
2016-08-01	<p>第14版 下記を変更しました。</p> <ul style="list-style-type: none"> 「 インポート・エクスポートを行う 」に採番ルール定義の移行に関する補足説明を追加しました。 「 申請者承認防止機能を利用する 」を追加しました。 「 BIS作成種類「BISフロー」の利用可否を切り替えるための設定をする 」を追加しました。 「 一括インポート・エクスポートを行う 」を追加しました。 「 メールやIMBoxを設定する 」にショートカットURLから遷移時の「戻る」の遷移先に関する説明を追加しました。 IM-BPM のリリースに伴い、BIS作成種類「BPM」を「BISフロー」に変更しました。
2017-04-01	<p>第15版 下記を変更しました。</p> <ul style="list-style-type: none"> 「 データソース定義を設定する 」にリクエストパラメータへのシステムパラメータの設定方法に関する説明を追加しました。 「 データソース定義を設定する 」にレスポンスパラメータへの処理結果メッセージの設定方法に関する説明を追加しました。 「 サンプル 」に処理結果メッセージのサンプルを追加しました。 「 フロー設計書を出力する 」を追加しました。 以下のページにインポート時のワークフローパラメータの設定に関する注意事項を追加しました。 <ul style="list-style-type: none"> 「 インポート・エクスポートを行う 」 - 「 IM-Workflow に関する定義ファイルをインポートする 」 「 一括インポート・エクスポートを行う 」 - 「 一括インポートを実行する 」

変更年月日	変更内容
2017-08-01	第16版 下記を変更しました。 <ul style="list-style-type: none">■ 機械翻訳時の翻訳精度と可読性向上のため、以下のページの記号等の表現を変更しました。<ul style="list-style-type: none">■ 「一括処理対象者変更を利用する」■ 「アプリケーション情報をインポートする」にインポート時のテーブル作成に関する補足説明を追加しました。
2017-12-01	第17版 下記を変更しました。 <ul style="list-style-type: none">■ 「データソース定義を設定する」で紹介されている OpenRules のドキュメントを変更しました。

はじめに

本書の目的

本書は IM-BIS for Accel Platform（以下 IM-BIS）でのシステム管理者が行う IM-BIS のシステム管理機能について説明したドキュメントです。

対象読者

次の利用者を対象としています。

- IM-BIS のシステム管理を実施する方
 - IM-BIS のシステム管理を実施するユーザは、IM-Workflow、IM-FormaDesigner、IM-BIS の機能について理解していることを前提としています。

IM-BIS では、利用する機能に応じて、ユーザを以下のように分類しています。
ユーザの役割に合わせて必要なドキュメントを参照してください。

	IM-BIS システム管理者 本書の対象	<ul style="list-style-type: none"> ・ IM-BIS for Accel Platformで作成したフローに関する詳細設定を行います。 ・ IM-Workflow、IM-FormaDesignerの機能の利用、外部連携を利用するための設定を行います。
	IM-BIS 業務管理者 (IM-BIS 監査者)	<p>IM-BIS 業務管理者</p> <ul style="list-style-type: none"> ・ IM-BIS for Accel PlatformでBPM、ワークフローの定義を行います。 ・ 作成したIM-BISで履歴やBAMの情報の収集の設定を行います。 <p>IM-BIS 監査者</p> <ul style="list-style-type: none"> ・ IM-BIS業務管理者と同等ですが、実行したプロセスの履歴を参照するなど監査目的で利用します。
	IM-BIS 利用者	<ul style="list-style-type: none"> ・ IM-BISで定義されたBPM、ワークフローを実行します。

本書の構成

本書を初めてご覧になる方は以下の順に読み進めていただくことをお勧めいたします。

- [IM-BIS について](#)
IM-BIS の概要について説明しています。
- [システム管理者が使用する機能](#)
IM-BIS でシステム管理者のみが使用する機能について説明しています。
システム管理者は、上記の機能の他にも業務管理者が使用する機能を使用できます。
詳細については、「[IM-BIS 業務管理者操作ガイド](#)」を参照してください。
- [IM-BIS の詳細カスタマイズ機能](#)
IM-BIS でBIS定義の詳細カスタマイズを行うための方法を説明しています。

本書では、IM-BIS の機能と他の関連する IM-Workflow や IM-FormaDesigner で対応する機能・ドキュメントについてご確認いただけます。
その他に、IM-BIS のシステム管理者が外部連携を登録するための手順を説明しています。

IM-BIS について

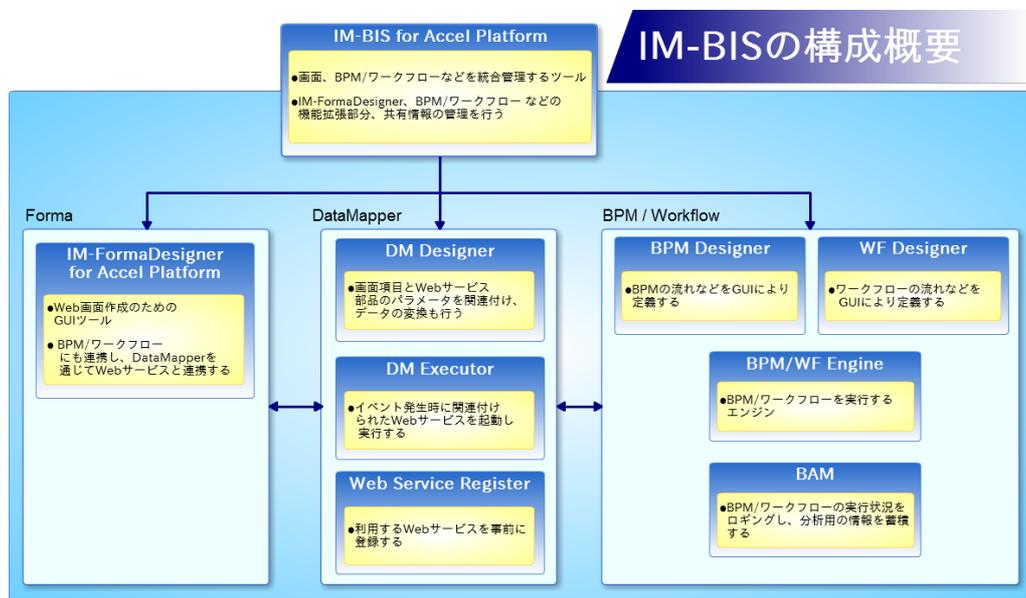
IM-BIS(Business Integration Suite)とは、intra-mart Accel Platform 上で動作する統合開発環境です。

専門的なアプリケーション開発スキルを必要とせず、視覚的で直感的な開発ツールにより、業務に携わる利用者自身が業務プロセスの設計、開発を行うことができます。

外部連携機能や業務プロセスの分析機能を標準で用意しており、既存の資産を有効利用しながら業務改善の基盤としてお使いいただけます。

また、IM-BIS はクラウド環境でも利用することができるため、初期投資を抑えつつ段階的な業務統合を推進することができます。

IM-BIS の構成



IM-BIS の特徴

開発環境を統合

- 画面、外部連携、BISフロー、ワークフローを全て統合化した開発環境です。
- 業務分析のためのBAMを標準装備しており、ビジネスプロセスのボトルネックの改善を図ることができます。
- BISフローとワークフローを一体化するサブフロー機能により柔軟なフローの開発ができます。

画面と処理の完全な分離

- マッピングツールにより外部システムのWebサービスなどと容易に連携ができます。
- 画面と処理をマッピングツールにより連携できるため開発を分業化することができます。

可視化された新たな開発メソッドロジー

- タスクの流れをフローで定義し、フローを参照しながらタスクで使用する画面を開発することができます。
- タスク上に設定されたコンテンツ（画面等）はタスクのアイコンにより可視化され、開発状況が確認できます。
- 複雑な処理は画面とは分離し、SOA化してマッピングツールで連携することができます。

BISフローとワークフロー

IM-BIS でフローを作成する場合には、「BISフロー」と「ワークフロー」の2つから選択できますが、以下のポイントで使い分けていただくと、利用しやすくなります。

- 申請・承認を伴う 業務プロセスの管理を行う場合
 - 「ワークフロー」をご利用ください。
- 申請・承認を伴わない 業務プロセスの管理を行う場合
 - 「BISフロー」をご利用ください。

 コラム

IM-BIS 2016 Summer (8.0.11) 以降、IM-BPM のリリースに伴い、簡易BPMである作成種類「BPM」の名称を「BISフロー」に変更します。

各ドキュメント内にて、画像等旧名称のままのものが存在しますが、「BPM」と「BISフロー」を読み替えて参照をお願いします。

採番ルール定義を設定する

ここでは、採番ルール定義の設定方法と権限の設定方法について説明します。

Contents

- [採番ルール定義とは](#)
- [採番ルール定義を作成する](#)
- [採番ルール定義の権限を設定する](#)

採番ルール定義とは

採番体系を管理し、一意の番号を生成するための定義です。
採番ルール定義を設定すると、画面アイテム「採番」で利用することができます。

採番ルール定義を作成する

詳細は「[IM-FormaDesigner 作成者操作ガイド](#)」を参照してください。

採番ルール定義の権限を設定する

登録した採番ルール定義をBIS業務管理者が利用するためには、管理グループに設定する必要があります。
以下のリンク先の手順に従って、登録した採番ルール定義を管理グループの管理対象に設定してください。

[管理グループに「管理対象」を設定する](#)

各種定義の権限管理を設定する

ここでは、各種定義の権限管理をしている管理グループの設定方法について説明します。

Contents

- [管理グループとは](#)
- [管理グループ設定によってBIS業務管理者が管理・利用できる機能](#)
- [管理グループを設定する](#)
- [管理グループに「管理対象」を設定する](#)

管理グループとは

管理グループは、各種定義（BIS定義、外部連携、採番ルール定義）の権限管理を行います。
管理グループを設定することにより、「BIS業務管理者」は、IM-BIS で作成したBISフロー／ワークフローの作成、更新などを行うことができます。

管理グループ設定によってBIS業務管理者が管理・利用できる機能

「管理グループ」機能では、BIS業務管理者が管理・利用できる対象が設定できます。
「BIS業務管理者」として IM-BISの管理を行う場合には、管理対象となる各種定義（BIS定義、外部連携、採番ルール定義）を設定する必要があります。
対象の定義を設定していない場合、「BIS業務管理者」は新規に業務管理者が作成したフロー（BIS定義）しか管理することができません。

コラム

BIS業務管理者がフロー（BIS定義）を新規作成した場合、BIS業務管理者自身が所属する管理グループの管理対象として自動的に追加されます。

権限管理対象の定義

種別	説明
----	----

種別	説明
BIS定義	<p>作成したフロー（BIS定義）の定義に関する管理権限および実行プロセスに関する管理権限を管理します。マスタ編集の設定により定義の権限範囲が決まり、案件操作の設定により運用上の権限範囲が決まります。</p> <ul style="list-style-type: none"> マスタ編集 ON：対象のBIS定義の参照、編集が行えます。 OFF：対象のBIS定義の参照のみ行えます。 案件操作 ON：対象のBIS定義の全実行プロセスの参照、案件操作が行えます。（プロセスの参照一覧画面） OFF：対象のBIS定義の全実行プロセスの参照のみ行えます。 <p>管理対象に含まれていないBIS定義は、定義の参照および管理者権限での実行プロセスの参照は行えません。</p>
外部連携	<p>BIS定義の作成時に利用できるデータソース定義を管理します。 管理対象に含まれていないデータソースは、外部連携機能では利用できません。</p>
採番ルール定義	<p>BIS定義の作成時に利用できる採番ルール定義を管理します。 採番ルール定義は、フォーム作成時に画面アイテム「採番」で利用します。 管理対象に含まれていない採番ルール定義は、画面アイテムの「採番」では利用できません。</p>

コラム

内部統制の監査目的などで、実行プロセスの履歴を参照させたい場合は、監査者用の管理グループを作成し対象BIS定義にマスタ編集「OFF」、案件操作「OFF」を設定してください。
さらにBIS定義を参照させたくない場合は、認可機能により実行プロセス参照用のメニューのみに限定することができます。

管理グループを設定する

管理グループを設定するには、以下の手順を行います。

1. 「サイトマップ」→「IM-BIS」→「フロー運用管理」→「権限者設定」→「管理グループ設定」をクリックします。
2. 「新規登録」をクリックします。

3. 「グループ名」に任意の名称を入力します。

管理グループ-新規登録

サンプルグループ

管理グループID * S4dn8allnwayws

管理グループ名 *

日本語	サンプルグループ_2
英語	sample_group_2
中国語 (中華人民共和國)	样品组_2

説明

日本語	
英語	
中国語 (中華人民共和國)	

アクセス権限

設定

管理対象

設定

BIS 外部連携 探検ルール

BIS名 マスク編集 案件操作

登録

4. 「アクセス権限」の「設定」をクリックします。

管理グループ-新規登録

サンプルグループ

管理グループID * S4dn8allnwayws

管理グループ名 *

日本語	サンプルグループ_2
英語	sample_group_2
中国語 (中華人民共和國)	样品组_2

説明

日本語	
英語	
中国語 (中華人民共和國)	

アクセス権限

設定

管理対象

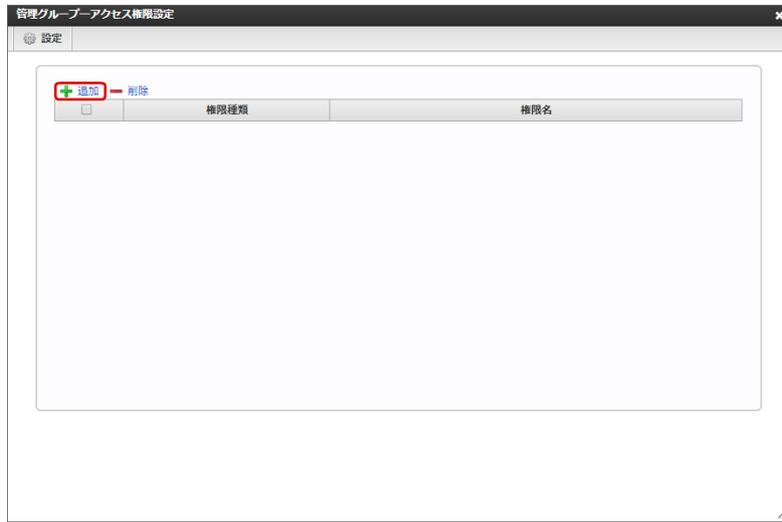
設定

BIS 外部連携 探検ルール

BIS名 マスク編集 案件操作

登録

5. 「追加」をクリックします。



6. 「BIS業務管理者」として設定するユーザなどを選択します。



7. 「設定」をクリックします。



8. 「管理対象」の「設定」をクリックします。

9. 管理できるBIS定義・外部連携・採番ルール定義をそれぞれ検索して設定します。

	BIS名	マスタ編集	案件操作
<input type="checkbox"/>	サンプル_BPM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

コラム

- BIS定義についてはマスタ編集のチェックがオンの場合は更新・削除可能、オフの場合は参照のみが可能です。この設定により、「フロー編集」でノードをダブルクリックした際の動作が変わります。
 - チェックがオンの場合
「フォーム・デザイナー」画面が表示されます。
 - チェックがオフの場合
「フォームプレビュー」画面が表示されます。

10. 「登録」をクリックします。

管理グループID * S14dn8allnwayws

管理グループ名 *

日本語 サンプルグループ_2

英語 sample_group_2

中国語 (中華人民共和国) 样品组_2

説明

日本語

英語

中国語 (中華人民共和国)

アクセス権限

設定

上田 辰男

青柳 辰巳

管理対象

設定

BIS 外部連携 採番ルール

BIS名	マスク編集	案件操作
サンプル_BPM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

登録

コラム

管理グループ設定で設定した管理の権限はメニューの「業務管理者」の「IM-BIS」の一覧からの編集時に適用されます。「システム管理者」の「IM-BIS」の一覧からの編集時には適用されませんので、注意してください。

管理グループのBIS定義に関する権限設定は、使用するメニューに応じて動作が異なりますので注意してください。

■ マスタ編集設定

権限が有効となるメニュー

業務管理者>IM-BIS作成>IM-BIS

権限が無効となるメニュー

システム管理者>IM-BIS作成>IM-BIS

■ 案件操作設定

権限が有効となるメニュー

業務管理者>フロー>参照 (ワークフロー)

業務管理者>フロー>参照 (BISフロー)

監査者>フロー>参照 (ワークフロー)

監査者>フロー>参照 (BISフロー)

権限が無効となるメニュー

システム管理者>フロー運用管理>参照 (ワークフロー)

システム管理者>フロー運用管理>参照 (BISフロー)

注意

IM-BIS で設定した管理グループを IM-Workflow で変更した場合、正しく動作しない可能性がありますので、注意してください。

管理グループに「管理対象」を設定する

IM-FormaDesigner で登録したデータソース定義、採番ルール定義をBIS業務管理者がBISフロー／ワークフローの作成時に利用するためには、管理グループの管理対象として登録する必要があります。

管理グループに作成済みのデータソース定義、採番ルール定義を追加する方法は、以下の手順で行います。

1. 「サイトマップ」→「IM-BIS」→「フロー運用管理」→「権限者設定」→「管理グループ設定」をクリックします。
2. 設定対象の「管理グループ設定 - 編集」画面を表示します。

管理グループ-更新

新規登録 コピー 権限者確認

サンプルグループ

管理グループID * Si4dn89vgdbg9ws

管理グループ名 *

日本語	サンプルグループ
英語	sample_group
中国語 (中華人民共和国)	样品组

説明

日本語	
英語	
中国語 (中華人民共和国)	

アクセス権限

設定

青柳辰巳
林政義
上田辰男

管理対象

設定

BIS 外部連携 採番ルール

BIS名	マスタ編集	案件操作
サンプル_BPM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

更新 削除

3. 「管理対象」の「設定」をクリックします。

管理グループ-更新

新規登録 コピー 権限者確認

サンプルグループ

管理グループID * Si4dn89vgdbg9ws

管理グループ名 *

日本語	サンプルグループ
英語	sample_group
中国語 (中華人民共和国)	样品组

説明

日本語	
英語	
中国語 (中華人民共和国)	

アクセス権限

設定

青柳辰巳
林政義
上田辰男

管理対象

設定

BIS 外部連携 採番ルール

BIS名	マスタ編集	案件操作
サンプル_BPM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

更新 削除

4. データソース定義を追加する場合は「外部連携」、採番ルール定義を追加する場合は「採番ルール」をクリックします。



5. 「追加」をクリックします。



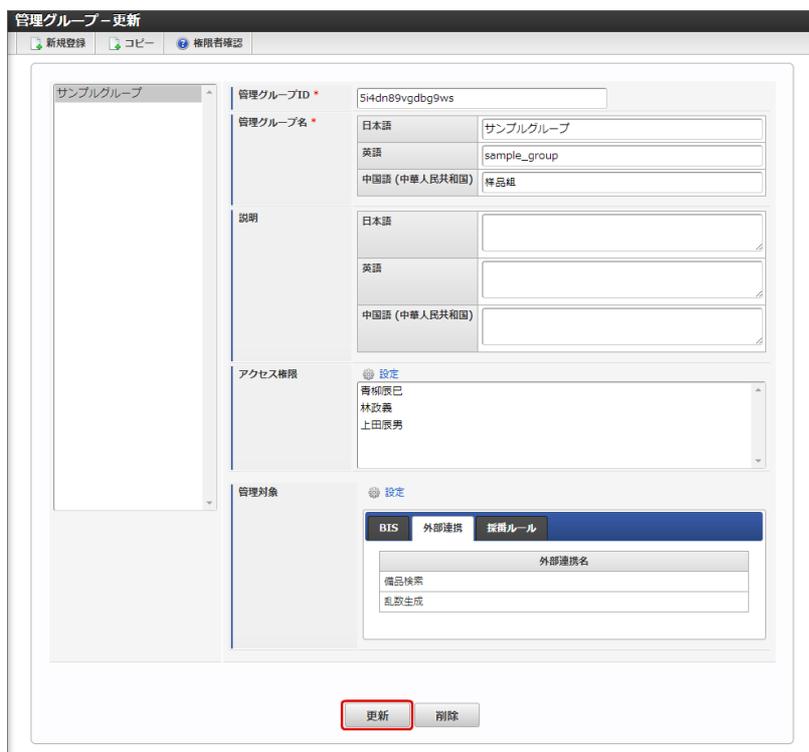
6. 対象のデータソース定義（採番ルール定義）のチェックをオンにし、「決定」をクリックします。



7. 「設定」をクリックします。



8. 管理対象に追加した外部連携（採番ルール定義）が表示されていることを確認し、「更新」をクリックして完了です。



「IM-BIS 一覧」画面の説明（システム管理者向け）

ここでは、システム管理者が使用する場合の「IM-BIS 一覧」画面について説明します。

「BISシステム管理者」は、「IM-BIS - 一覧」からBIS定義の管理や詳細カスタマイズ作業を行うことができます。業務管理者向けの「IM-BIS 一覧」画面については、「[IM-BIS 業務管理者操作ガイド](#)」を参照してください。

Contents

- [BISシステム管理者／BIS業務管理者の一覧の違い](#)
- [「IM-BIS - 一覧」画面の機能と各部の説明](#)

BISシステム管理者／BIS業務管理者の一覧の違い

「IM-BIS - 一覧」画面は、実行しているユーザのロールが「BISシステム管理者」または「BIS業務管理者」であるかの違いにより、表示項目・利用できる機能の一部が異なります。

以下のポイントがユーザのロールによる機能の違いです。

- 表示項目
 - 「BISシステム管理者」の場合は、「フローID」「アプリ」の項目が表示されます。
 - 「BIS業務管理者」の場合、「フローID」「アプリ」の項目は表示されません。
- 利用できる機能

「BISシステム管理者」の場合は、「アプリ」のアイコンから IM-FormaDesigner の機能に基づくカスタマイズを行うことができます。

「BIS業務管理者」の場合、「アプリ」の項目が表示されないため、カスタマイズを行うことはできません。



「IM-BIS - 一覧」画面の機能と各部の説明

「IM-BIS - 一覧」画面の内容は以下の通りです。



1. 最近使用した定義
「IM-BIS - 更新履歴」画面を表示します。
2. 新規登録
「IM-BIS - 新規登録」画面を表示します。

3. 最新情報

表示している内容を最新の状態に更新します。

4. BISフロー（チェックボックス）

チェックをオンにすると、BISフローを検索対象にします。

5. WF（チェックボックス）

チェックをオンにすると、ワークフローを検索対象にします。

6. 検索条件

検索する対象をセレクトボックスから選択します。

7. 入力フィールド

検索するキーワードを入力します。

8. 検索ボタン

押下すると検索を実行します。

9. 選択した定義を削除

一覧から定義を選択して押下すると対象の定義を削除します。

10. 選択（チェックボックス）

チェックをオンにすると、対象の行の定義を選択します

項目欄のチェックボックスをオンにすると、一覧に表示された定義をすべて選択します。

11. 編集

 をクリックすると、対象の定義の「IM-BIS - フロー編集」画面を表示します。

12. BIS作成種類

対象の定義のBIS作成種類を表示します。

13. BIS名

対象の定義のBIS名を表示します。

14. 説明

対象のBIS定義の説明文を表示します。

15. BIS ID

対象の定義のBIS IDを表示します。

16. フローID

対象の定義のフローIDを表示します。

17. アプリ

 をクリックすると対象の定義の「フォーム設定」を表示します。

このメニューから変更を行った場合、BIS定義のデータの整合性がとれなくなる可能性がありますので、注意してください。

コラム

「IM-BIS - 一覧」画面では、以下の項目をクリックすると、昇順・降順でソートすることができます。

- BIS作成種類
- BIS名
- 説明
- BIS ID

一覧表示パターン定義を設定する

ここでは、IM-BIS の一覧表示パターン定義の設定方法について説明します。

Contents

- [一覧表示パターン定義とは](#)
- [IM-BIS で、一覧表示パターン定義を設定する](#)

一覧表示パターン定義とは

一覧表示パターン定義では、ワークフローの処理、案件管理するための各種一覧項目を編集し、任意の表示パターンを定義することができます。

IM-BIS で、一覧表示パターン定義を設定する

IM-BIS では、IM-Workflow と同様に「申請」、「処理開始」の一覧に表示する項目を追加・削除することができます。

「サイトマップ」→「IM-BIS」→「システム管理者」→「マスタ管理」→「フロー」→「一覧表示パターン定義」から、設定できます。

基本的な使い方については、「IM-Workflow 管理者操作ガイド」を参照してください。

コラム

IM-BIS で設定済みの一覧に「詳細」を追加する場合には、別途フローへの詳細画面の設定が必要となるため、以下のページの手順に従って設定してください。

[参照画面や特殊なタスク（ノード）で表示する画面を設定する](#)

フローをグループ分類する

ここでは、フローグループ定義の設定方法について説明します。

Contents

- [フローグループ定義とは](#)
- [フローグループ定義を登録する](#)
- [フローグループ定義を追加する](#)
- [フローグループ定義を階層設定する](#)

フローグループ定義とは

フローグループ定義は、フロー定義を検索目的で分類するための定義です。

ツリー状の親子関係を持ち、1つのフロー定義が複数のフローグループに属することもできます。

フローグループ定義を登録する

1. 「サイトマップ」→「IM-BIS」→「システム管理者」→「マスタ定義」→「フロー」→「フローグループ定義」をクリックします。
2. フローグループが一つも登録されていない場合、「フローグループ定義 - 新規作成」画面が表示されます。既にフローグループが登録されている場合は「フローグループ定義を追加する」を参照してください。

フローグループ定義情報を入力します。



フローグループ定義 - 新規作成	
フローグループID(必須)	<input type="text" value="517f6ud2img93y"/>
親階層(必須)	<input type="text" value="/"/>
フローグループ名(必須)	英語 <input type="text" value="Sample Flow Group"/>
	日本語 <input type="text" value="サンプルフローグループ"/>
	中国語 <input type="text" value="Sample Flow Group"/>
備考	英語 <input type="text" value="Sample flow group remarks"/>
	日本語 <input type="text" value="サンプルフローグループ備考"/>
	中国語 <input type="text" value="Sample flow group remarks"/>
ソートキー(必須)	<input type="text" value="0"/>
フロー	<input type="text" value="サンプルBPM"/> <input type="text" value="サンプルワークフロー"/> <input type="button" value="検索"/> <input type="button" value="クリア"/>
<input type="button" value="登録"/>	

3. 「登録」をクリックします。

フローグループ定義 - 新規作成

フローグループID (必須)	5i7j6ud2imig93y	
親階層 (必須)	/	
フローグループ名 (必須)	英語	Sample Flow Group
	日本語	サンプルフローグループ
	中国語	Sample Flow Group
備考	英語	Sample flow group remarks
	日本語	サンプルフローグループ備考
	中国語	Sample flow group remarks
ソートキー (必須)	0	
フロー	サンプルBPM サンプルワークフロー	検索 クリア

登録

4. フローグループ定義を登録することができました。

フローグループ定義 - 編集

新規作成

サンプルフローグループ	フローグループID	5i7j6ud2imig93y	
	親階層 (必須)	/	
	フローグループ名 (必須)	英語	Sample Flow Group
		日本語	サンプルフローグループ
		中国語	Sample Flow Group
	備考	英語	Sample flow group remarks
		日本語	サンプルフローグループ備考
		中国語	Sample flow group remarks
	ソートキー (必須)	0	
	フロー	サンプルBPM サンプルワークフロー	検索 クリア

更新 削除

フローグループ定義を追加する

1. 「サイトマップ」→「IM-BIS」→「システム管理者」→「マスタ定義」→「フロー」→「フローグループ定義」をクリックします。
2. 「フローグループ定義 - 編集」画面の「新規作成」をクリックします。

フローグループ定義 - 編集

新規作成

サンプルフローグループ

フローグループID	5i7j6ud2imjg93y
親階層 (必須)	/
フローグループ名 (必須)	英語 Sample Flow Group 日本語 サンプルフローグループ 中国語 Sample Flow Group
備考	英語 Sample flow group remarks 日本語 サンプルフローグループ備考 中国語 Sample flow group remarks
ソートキー (必須)	0
フロー	サンプルBPM サンプルワークフロー

更新 削除

3. 「フローグループ定義 - 新規作成」画面でフローグループ定義情報を入力します。

フローグループ定義 - 新規作成

戻る

サンプルフローグループ

フローグループID (必須)	5i7j6ud6o8tzs3y
親階層 (必須)	/
フローグループ名 (必須)	英語 Sample Flow Group 2 日本語 サンプルフローグループ2 中国語 Sample Flow Group 2
備考	英語 Sample flow group 2 remarks 日本語 サンプルフローグループ2備考 中国語 Sample flow group 2 remarks
ソートキー (必須)	0
フロー	BPMサンプル

登録

4. 「登録」をクリックします。

フローグループ定義 - 新規作成

戻る

サンプルフローグループ

フローグループID (必須)	5i7j6ud6o8tzs3y
親階層 (必須)	/
フローグループ名 (必須)	英語 Sample Flow Group 2 日本語 サンプルフローグループ2 中国語 Sample Flow Group 2
備考	英語 Sample flow group 2 remarks 日本語 サンプルフローグループ2備考 中国語 Sample flow group 2 remarks
ソートキー (必須)	0
フロー	BPMサンプル

登録

5. フローグループ定義を追加することができました。

フローグループ定義 - 編集	
新規作成	
<ul style="list-style-type: none"> サンプルフローグループ サンプルフローグループ2 	フローグループID: 517j6ud2imjg93y 親階層 (必須): / フローグループ名 (必須): 英語: Sample Flow Group 日本語: サンプルフローグループ 中国語: Sample Flow Group 備考: 英語: Sample flow group remarks 日本語: サンプルフローグループ備考 中国語: Sample flow group remarks ソートキー (必須): 0 フロー: サンプルBPM サンプルワークフロー
<input type="button" value="更新"/> <input type="button" value="削除"/>	

フローグループ定義を階層設定する

1. 「サイトマップ」→「IM-BIS」→「システム管理者」→「マスタ定義」→「フロー」→「フローグループ定義」をクリックします。
2. 「フローグループ定義 - 編集」画面の「新規作成」をクリックします。

フローグループ定義 - 編集	
新規作成	
<ul style="list-style-type: none"> サンプルフローグループ サンプルフローグループ2 	フローグループID: 517j6ud2imjg93y 親階層 (必須): / フローグループ名 (必須): 英語: Sample Flow Group 日本語: サンプルフローグループ 中国語: Sample Flow Group 備考: 英語: Sample flow group remarks 日本語: サンプルフローグループ備考 中国語: Sample flow group remarks ソートキー (必須): 0 フロー: サンプルBPM サンプルワークフロー
<input type="button" value="更新"/> <input type="button" value="削除"/>	

3. フローグループ定義情報を入力します。
親階層にしたいフローグループ定義を選択します。
4. 「登録」をクリックします。

フローグループ定義 - 新規作成

戻る

サンプルフローグループ
サンプルフローグループ2

フローグループID(必須) 517j6udd0ch6m3y

親階層(必須) /サンプルフローグループ

フローグループ名(必須)
英語 Sample Flow Group 3
日本語 サンプルフローグループ3
中国語 Sample Flow Group 3

備考
英語 Sample flow group 3 remarks
日本語 サンプルフローグループ3備考
中国語 Sample flow group 3 remarks

ソートキー(必須) 0

フロー サンプルワークフロー

検索 クリア

登録

5. フローグループ定義を階層で登録することができました。

左にあるツリーの アイコンをクリックし、階層になっていることを確認してください。

フローグループ定義 - 編集

新規作成

サンプルフローグループ
サンプルフローグループ3
サンプルフローグループ2

フローグループID 517j6ud2imig93y

親階層(必須) /

フローグループ名(必須)
英語 Sample Flow Group
日本語 サンプルフローグループ
中国語 Sample Flow Group

備考
英語 Sample flow group remarks
日本語 サンプルフローグループ備考
中国語 Sample flow group remarks

ソートキー(必須) 0

フロー サンプルBPM
サンプルワークフロー

検索 クリア

更新 削除

代理管理者を設定する

ここでは、代理管理者の設定方法について説明します。

Contents

- 代理管理者設定とは
- 代理管理者を設定する

代理管理者設定とは

代理管理者設定は、代理設定の権限を、第三者に付与することができるユーザを設定する機能です。設定すると、ワークフロー利用者に代理設定の権限を付与します。

代理管理者を設定する

- 「サイトマップ」→「IM-BIS」→「システム管理者」→「フロー運用管理」→「権限者設定」→「代理管理者設定」をクリックします。
- 「検索」をクリックします。

代理管理者設定

検索

対象種別	対象名	権限者確認	クリア
更新			

3. 代理管理者を選択します。

代理管理者設定

代理管理者

- ユーザ
- 組織
- ロール
- パブリックグループ
- 役割
- 組織+役割
- パブリックグループ+役割
- 組織+ロール
- パブリックグループ+ロール

対象種別	対象名	権限者確認	クリア
更新			

4. 「更新」をクリックします。

代理管理者設定

検索

対象種別	対象名	権限者確認	クリア
組織	サンプル課12		
更新			

5. 代理管理者設定を登録することができました。

対象種別	対象名	権限者確認	クリア
組織	サンプル課12		

更新

データソース定義を設定する

ここでは、データソース定義の設定方法について説明します。

Contents

- データソース定義とは
- データソース定義の設定方法
- 階層化したリクエストパラメータ、レスポンスフィールドの設定方法
- BIS業務管理者が、各データソース定義を利用するための設定をする
- リクエストパラメータへのシステムパラメータ設定方法
- レスポンスパラメータへの処理結果メッセージ設定方法

データソース定義とは

データソース定義は、外部連携する場合のインタフェースを抽象化し、Webサービスでも、テーブルでも、同じインタフェースで連携する抽象化層の定義です。

データソース定義では、外部連携機能を設定します。

データソース定義の設定方法

「データソース - 新規登録」画面の機能と各部の説明

「データソース - 新規登録」画面の内容は以下の通りです。

1. データソース種別

データソースの種別を選択します。

データソースの種別は、以下の通りです。

- テナントDBクエリ：テナントDBからデータを取得します。
- シェアードDBクエリ：シェアードDBからデータを取得します。

- REST（非推奨）：REST APIを使用して、Webサービスを利用できるように設定します。
- SOAP（非推奨）：WSDLを使用して、Webサービスを利用できるように設定します。
- JAVA（非推奨）：Javaプログラムを連携できるように設定します。
- LogicDesigner：IM-LogicDesignerを連携できるように設定します。
- CSVインポート：CSVのインポートができるように設定します。
- CSVエクスポート：CSVのエクスポートができるように設定します。
- テナントDB更新系クエリ：テナントDBのデータの登録・更新・削除を実行します。
- シェアードDB更新系クエリ：シェアードDBのデータの登録・更新・削除を実行します。

2. データソース名

データソース定義の名称を入力します。

ここで設定した名称がデータマッパーなどで表示されます。

3. 備考

登録するデータソースの説明を入力します。

4. 他のロケール

クリックすると表示しているロケール以外のロケールに個別にデータソース名や備考の入力欄を表示します。

i コラム

データソース種別「ルール」は、OpenRules モジュールを導入した場合のみ利用できます。
 ルールの設定などの詳細については「[OpenRules for IM-BIS 連携ガイド](#)」を参照してください。

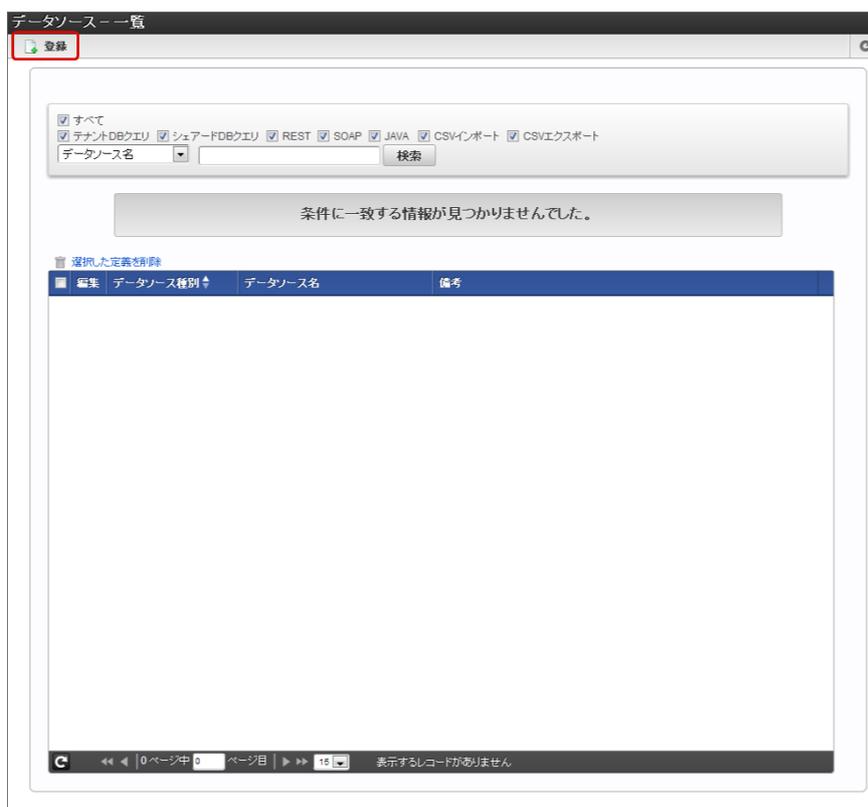
i コラム

IM-BIS 2015 Winterより、データソース種別「LogicDesigner」を追加しました。
 データソース種別「LogicDesigner」は、データソース種別「REST」「SOAP」「JAVA」の機能を補完したもので、今後新規機能の追加は「LogicDesigner」に対してのみ実施されます。
 そのため、「REST」「SOAP」「JAVA」の利用については非推奨とし、「LogicDesigner」の利用を推奨しています。

「データソース - 新規登録」画面の操作手順

「テナントDBクエリ」を設定する場合を例に、「データソース - 新規登録」画面の操作手順について説明します。

1. 「サイトマップ」→「IM-BIS」→「システム管理者」→「マスタ管理」→「外部連携」→「データソース定義」をクリックします。
2. 「データソース - 一覧」画面で「登録」をクリックします。



3. 「データソース種別」で任意のデータソース種別を選択します。

4. 任意の「データソース名」、「備考」を入力して「登録」をクリックします。

「データソース - 編集」画面の機能と各部の説明

「データソース - 編集」画面の内容は以下の通りです。
データソース種別により操作手順が異なりますので、操作手順についても以下を参照してください。

テナントDBクエリ

「データソース - 編集[テナントDBクエリ]」画面の機能と各部の説明

「データソース - 編集[テナントDBクエリ]」画面の内容は以下の通りです。

データソース 編集 [テナントDBクエリ]

データソース種別 テナントDBクエリ
データソース名 (1) tenant_sample

クエリ設定 管理会社設定

(2) SQL

```
SELECT
column1 as col1
FROM
sample_table
WHERE
AND column2 = ?
column3 IN ( ?, ? )
```

(3) 入力値

	テスト実行値	データ型	論理名	削除
IN 1	<input type="text"/>	VARCHAR	日本語 条件項目1 英語 Condition item1 中国語 条件項目1	-
IN 2	<input type="text"/>	VARCHAR	日本語 条件項目2-1 英語 Condition item2-1 中国語 条件項目2-1	-
IN 3	<input type="text"/>	VARCHAR	日本語 条件項目2-2 英語 Condition item2-2 中国語 条件項目2-2	-

(4) 出力値

	カラム名	データ型	論理名	削除
OUT 1	<input type="text"/>	VARCHAR	日本語 取得項目1 英語 Acquiring item1 中国語 取得項目1	-

(5) テスト実行 (6) 登録

1. データソース名
「データソース - 新規登録」画面で設定したデータソース名を表示します。

2. SQL
データソースとして実行するSQLを入力します。
記載するSQLはSELECT文のみとしてください。



注意

ここで指定するクエリは、サブクエリとして実行されます。
そのため、クエリ内で ORDER BY句 を指定すると、SQLServerではエラーが発生します。
SQLServer では、ORDER BY句 を指定しないようにしてください。
ただし、SQLServer 2005以降の場合には、TOP句と組み合わせることにより、ORDER BY句を指定することができます。
この制限について、IM-BISの外部連携から実行された場合は該当しません。

3. 入力値

- テスト実行値：「SQL」に記述したデータ処理の条件（WHERE句）に記述したカラム（テーブルの項目）に代入する値を入力します。アプリケーションの画面アイテムから入力されることが想定される値を入力します。
- データ型：項目に対応するデータの形式を文字型（VARCHAR）、数値型（NUMBER）、日付型（DATE）、タイムスタンプ型（TIMESTAMP）の中から選択します。
- 論理名：IM-FormaDesigner の画面アイテムでデータソースを利用する際の「パラメータ設定」で表示する項目名として利用されます。20文字まで設定することができます。
- 追加・削除：入力値の追加・削除ができます。

4. 出力値

- カラム名：「SQL」に記述したデータ処理がSELECT（データ抽出）の場合に出力する項目名を入力します。アプリケーションの画面アイテムに表示する項目を入力します。複数設定している場合には、画面アイテム「一覧選択」以外の項目にも検索結果を反映することができます。
- データ型：項目に対応するデータの形式を文字型（VARCHAR）、数値型（NUMBER）、日付型（DATE）、タイムスタンプ型（TIMESTAMP）の中から選択します。
- 論理名：IM-FormaDesigner の画面アイテムでデータソースを利用する際の「取得値設定」で表示する項目名として利用されます。20文字まで設定することができます。
- 追加・削除：出力値の追加・削除ができます。

! 注意

出力値を設定する場合には、以下の点に注意してください。

- 出力値のカラム名の設定で、以下のいずれかに該当する設定を行った場合には、アプリケーションの実行時にデータソース定義によるデータ取得が正しく動作しません。
 - 出力値のカラム名に大文字を含んでいる
 - 同一のデータソース定義内で2つ以上の出力値に同じカラム名を設定する
- クエリ詳細設定の出力値の設定を更新した場合には、該当のデータソース定義を呼び出している外部連携設定のデータマッパーの設定を再度実施するようにしてください。
データマッパーの設定を行わないと、データソース定義で行った更新内容が反映されません。

5. テスト実行

設定した情報を用いてデータが取得できるかが確認できます。
取得できなかった場合にはエラーメッセージが表示されます。

6. 登録

設定した内容をデータソース定義として登録します。

i コラム

入力値、出力値の論理名は、IM-BISを導入している環境で、互換用アイテムに分類される画面アイテムを使用する場合に必要な設定です。

「データソース - 編集[テナントDBクエリ]」画面の操作手順

「データソース - 編集[テナントDBクエリ]」画面の操作手順について説明します。

1. 「SQL」に実行するSQLを入力します。

データソース - 編集[テナントDBクエリ]

データソース種別: テナントDBクエリ
データソース名: テナントDBクエリサンプル

クエリ設定 | 管理会社設定

SQL

```
SELECT
  iifr_ut_product_master_regist.iifr_ud_product_name,
  iifr_ut_product_master_regist.iifr_ud_cost
FROM
  iifr_ut_product_master_regist
WHERE
  iifr_ut_product_master_regist.iifr_ud_locale = ?
```

入力値

テスト実行値	データ型	論理名	削除
IN 1	VARCHAR	日本語 条件項目 1 英語 Condition Item1 中国語 条件項目 1	-
IN 2	VARCHAR	日本語 条件項目 2-1 英語 Condition Item2-1 中国語 条件項目 2-1	-
IN 3	VARCHAR	日本語 条件項目 2-2 英語 Condition Item2-2 中国語 条件項目 2-2	-

出力値

カラム名	データ型	論理名	削除
OUT 1	VARCHAR	日本語 取得項目 1 英語 Acquiring Item1 中国語 取得項目 1	-

テスト実行 登録

2. 「SQL」のWHERE句で"?"を利用している場合には、"?"に合わせて「入力値」を設定します。

データベース - 編集 [テナントDBクエリ]

データベース種別: テナントDBクエリ
データベース名: テナントDBクエリサンプル

クエリ設定 管理会社設定

```
SQL
SELECT
  imfr_ut_product_master_resist.imfr_ud_product_name,
  imfr_ut_product_master_resist.imfr_ud_cost
FROM
  imfr_ut_product_master_resist
WHERE
  imfr_ut_product_master_resist.imfr_ud_locale = ?
```

入力値 + 追加

IN	テスト実行値	データ型	論理名	削除
IN 1	<input type="text"/>	VARCHAR	日本語 条件項目 1 英語 Condition item1 中国語 条件項目 1	-

出力値 + 追加

OUT	カラム名	データ型	論理名	削除
OUT 1	<input type="text"/>	VARCHAR	日本語 取得項目 1 英語 Acquiring item1 中国語 取得項目 1	-

テスト実行 登録

3. 「SQL」のSELECT句で設定している列に合わせて「出力値」を設定します。

データベース - 編集 [テナントDBクエリ]

データベース種別: テナントDBクエリ
データベース名: テナントDBクエリサンプル

クエリ設定 管理会社設定

```
SQL
SELECT
  imfr_ut_product_master_resist.imfr_ud_product_name,
  imfr_ut_product_master_resist.imfr_ud_cost
FROM
  imfr_ut_product_master_resist
WHERE
  imfr_ut_product_master_resist.imfr_ud_locale = ?
```

入力値 + 追加

IN	テスト実行値	データ型	論理名	削除
IN 1	<input type="text"/>	VARCHAR	日本語 条件項目 1 英語 Condition item1 中国語 条件項目 1	-

出力値 + 追加

OUT	カラム名	データ型	論理名	削除
OUT 1	imfr_ud_product_name	VARCHAR	日本語 取得項目 1 英語 Acquiring item1 中国語 取得項目 1	-
OUT 2	imfr_ud_cost	VARCHAR	日本語 取得項目 2 英語 Acquiring item 2 中国語 取得項目 2	-

テスト実行 登録

4. 「テスト実行値」にテスト用の値を入力し、「テスト実行」をクリックします。

データベース - 編集 [テナントDBクエリ]

データベース種別: テナントDBクエリ
データベース名: テナントDBクエリサンプル

クエリ設定 管理会社設定

```
SQL
SELECT
  imfr_ut_product_master_resist, imfr_ud_product_name,
  imfr_ut_product_master_resist, imfr_ud_cost
FROM
  imfr_ut_product_master_resist
WHERE
  imfr_ut_product_master_resist.imfr_ud_locale = ?
```

入力値

IN	テスト実行値	データ型	論理名	削除
IN 1	ja	VARCHAR	日本語 条件項目 1 英語 Condition item 1 中国語 条件項目 1	-

出力値

OUT	カラム名	データ型	論理名	削除
OUT 1	imfr_ud_product_namr	VARCHAR	日本語 取得項目 1 英語 Acquiring item 1 中国語 取得項目 1	-
OUT 2	imfr_ud_cost	VARCHAR	日本語 取得項目 2 英語 Acquiring item 2 中国語 取得項目 2	-

テスト実行 登録

5. テストが成功した場合、以下のメッセージが表示されます。

データベース - 編集 [テナントDBクエリ]

データベース種別: テナントDBクエリ
データベース名: テナントDBクエリサンプル

クエリ設定 管理会社設定

テスト実行に成功しました。取得件数: 23

```
SQL
SELECT
  imfr_ut_product_master_resist, imfr_ud_product_name,
  imfr_ut_product_master_resist, imfr_ud_cost
FROM
  imfr_ut_product_master_resist
WHERE
  imfr_ut_product_master_resist.imfr_ud_locale = ?
```

入力値

IN	テスト実行値	データ型	論理名	削除
IN 1	ja	VARCHAR	日本語 条件項目 1 英語 Condition item 1 中国語 条件項目 1	-

出力値

OUT	カラム名	データ型	論理名	削除
OUT 1	imfr_ud_product_namr	VARCHAR	日本語 取得項目 1 英語 Acquiring item 1 中国語 取得項目 1	-
OUT 2	imfr_ud_cost	VARCHAR	日本語 取得項目 2 英語 Acquiring item 2 中国語 取得項目 2	-

テスト実行 登録

6. 内容を確認して、「登録」をクリックします。

データソース - 編集[テナントDBクエリ]

データソース種別: テナントDBクエリ
データソース名: テナントDBクエリサンプル

クエリ設定 管理会社設定

```
SQL
SELECT
  imfr_ut_product_master_resist, imfr_ud_product_name,
  imfr_ut_product_master_resist, imfr_ud_cost
FROM
  imfr_ut_product_master_resist
WHERE
  imfr_ut_product_master_resist.imfr_ud_locale = ?
```

入力値

テスト実行値	データ型	論理名	削除
IN 1	ja	日本語 条件項目 1 英語 Condition item 1 中国語 条件項目 1	-

出力値

カラム名	データ型	論理名	削除
OUT 1	imfr_ud_product_namr	日本語 取得項目 1 英語 Acquiring item 1 中国語 取得項目 1	-
OUT 2	imfr_ud_cost	日本語 取得項目 2 英語 Acquiring item 2 中国語 取得項目 2	-

テスト実行 登録

7. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。

データソース - 一覧

登録

すべて
 テナントDBクエリ シェアードDBクエリ REST SOAP JAVA CSVインポート CSVエクスポート
 データソース名: 検索

選択した定義を削除

編集	データソース種別	データソース名	備考
<input checked="" type="checkbox"/>	テナントDBクエリ	テナントDBクエリサンプル	テナントDBクエリのサンプルです。

1ページ中 1 ページ目 12 1件中 1-1を表示

シェアードDBクエリ

「データソース - 編集[シェアードDBクエリ]」画面の機能と各部の説明

「データソース - 編集[シェアードDBクエリ]」画面の内容は以下の通りです。

データソース編集 [シェアードDBクエリ]

データソース種別: シェアードDBクエリ
データソース名 (1): shared_sample

クエリ設定 | 管理会社設定

(2) 接続ID: [選択]

(3) SQL:

```
SELECT
column1 as col1
FROM
sample_table
WHERE
column2 = ?
AND
column3 IN ( ?, ? )
```

(4) 入力値 [+追加]

テスト実行値	データ型	論理名	削除
IN 1	VARCHAR	日本語: 条件項目 1 英語: Condition item1 中国語: 条件項目 1	-
IN 2	VARCHAR	日本語: 条件項目 2-1 英語: Condition item2-1 中国語: 条件項目 2-1	-
IN 3	VARCHAR	日本語: 条件項目 2-2 英語: Condition item2-2 中国語: 条件項目 2-2	-

(5) 出力値 [+追加]

カラム名	データ型	論理名	削除
OUT 1	VARCHAR	日本語: 取得項目 1 英語: Acquiring item1 中国語: 取得項目 1	-

(6) テスト実行 (7) 登録

- データソース名
「データソース - 新規登録」画面で設定したデータソース名を表示します。
- 接続ID
シェアードDBとして設定されている接続情報を選択します。
- SQL
データソースとして実行するSQLを入力します。
記載するSQLはSELECT文のみとしてください。

注意

ここで指定するクエリは、サブクエリとして実行されます。
そのため、クエリ内で ORDER BY句 を指定すると、SQLServerではエラーが発生します。
SQLServer では、ORDER BY句 を指定しないようにしてください。
ただし、SQLServer 2005以降の場合には、TOP句と組み合わせることにより、ORDER BY句を指定することができます。
この制限について、IM-BISの外部連携から実行された場合は該当しません。

- 入力値
 - テスト実行値: 「SQL」に記述したデータ処理の条件 (WHERE句) に記述したカラム (テーブルの項目) に代入する値を入力します。アプリケーションの画面アイテムから入力されることが想定される値を入力します。
 - データ型: 項目に対応するデータの形式を文字型 (VARCHAR)、数値型 (NUMBER)、日付型 (DATE)、タイムスタンプ型 (TIMESTAMP) の中から選択します。
 - 論理名: 画面アイテムでデータソースを利用する際の「パラメータ設定」で表示する項目名として利用されます。20文字まで利用することができます。
 - 追加・削除: 入力値の追加・削除ができます。
- 出力値
 - カラム名: 「SQL」に記述したデータ処理がSELECT (データ抽出) の場合に出力する項目名を入力します。アプリケーションの画面アイテムに表示する項目を入力します。複数設定している場合には、画面アイテム「一覧選択」以外の項目にも検索結果を反映することができます。
 - データ型: 項目に対応するデータの形式を文字型 (VARCHAR)、数値型 (NUMBER)、日付型 (DATE)、タイムスタンプ型 (TIMESTAMP) の中から選択します。
 - 論理名: 画面アイテムでデータソースを利用する際の「取得値設定」で表示する項目名として利用されます。20文字まで利用することができます。
 - 追加・削除: 出力値の追加・削除ができます。



注意

出力値を設定する場合には、以下の点に注意してください。

- 出力値のカラム名の設定で、以下のいずれかに該当する設定を行った場合には、アプリケーションの実行時にデータソース定義によるデータ取得が正しく動作しません。
 - 出力値のカラム名に大文字を含んでいる
 - 同一のデータソース定義内で2つ以上の出力値に同じカラム名を設定する
- クエリ詳細設定の出力値の設定を更新した場合には、該当のデータソース定義を呼び出している外部連携設定のデータマッパーの設定を再度実施するようにしてください。
データマッパーの設定を行わないと、データソース定義で行った更新内容が反映されません。

6. テスト実行

設定した情報を用いてデータが取得できるかが確認できます。
取得できなかった場合にはエラーメッセージが表示されます。

7. 登録

設定した内容をデータソース定義として登録します。

「データソース - 編集[シェアードDBクエリ]」画面の操作手順

「データソース - 編集[シェアードDBクエリ]」画面の操作手順について説明します。

1. 「接続ID」を選択します。

データソース - 編集[シェアードDBクエリ]

データソース種別: シェアードDBクエリ
データソース名: シェアードDBクエリサンプル

クエリ設定 管理会社設定

接続ID: bis_test_db

SQL:

```
SELECT column1 as col1
FROM sample_table
WHERE column2 = ?
AND column3 IN ( ?, ? )
```

入力値

テスト実行値	データ型	論理名	削除
IN 1	VARCHAR	日本語: 条件項目 1 英語: Condition item1 中国語: 条件項目 1	-
IN 2	VARCHAR	日本語: 条件項目 2-1 英語: Condition item2-1 中国語: 条件項目 2-1	-
IN 3	VARCHAR	日本語: 条件項目 2-2 英語: Condition item2-2 中国語: 条件項目 2-2	-

出力値

カラム名	データ型	論理名	削除
OUT 1	VARCHAR	日本語: 取得項目 1 英語: Acquiring item1 中国語: 取得項目 1	-

テスト実行 登録

2. 「SQL」に実行するSQLを入力します。

データソース 編集[シェアードDBクエリ]

データソース種別 シェアードDBクエリ
データソース名 シェアードDBクエリサンプル

クエリ設定 管理会社設定

接続ID bis_test_db

SQL

```
SELECT column1 as col1
FROM sample_table
WHERE column2 = ?
AND column3 IN ( ?, ? )
```

入力値

IN	テスト実行値	データ型	論理名	削除
IN 1		VARCHAR	条件項目 1 Condition item1 条件項目 1	-
IN 2		VARCHAR	条件項目 2-1 Condition item2-1 条件項目 2-1	-
IN 3		VARCHAR	条件項目 2-2 Condition item2-2 条件項目 2-2	-

出力値

OUT	カラム名	データ型	論理名	削除
OUT 1		VARCHAR	取得項目 1 Acquiring item1 取得項目 1	-

テスト実行 登録

3. 「SQL」のWHERE句で"?"を利用している場合には、"?"に合わせて「入力値」を設定します。

データソース 編集[シェアードDBクエリ]

データソース種別 シェアードDBクエリ
データソース名 シェアードDBクエリサンプル

クエリ設定 管理会社設定

接続ID bis_test_db

SQL

```
SELECT emp_name as EmployeeName
FROM sample_table_for_testing_query
WHERE input_num = ?
```

入力値

IN	テスト実行値	データ型	論理名	削除
IN 1		VARCHAR	条件1 Condition1 条件1	-

出力値

OUT	カラム名	データ型	論理名	削除
OUT 1		VARCHAR	取得項目 1 Acquiring item1 取得項目 1	-

テスト実行 登録

4. 「SQL」のSELECT句で設定している列に合わせて「出力値」を設定します。

データベース編集[シェアードDBクエリ]

データベース種別: シェアードDBクエリ
データベース名: シェアードDBクエリサンプル

クエリ設定 管理会社設定

接続ID: bis_test_db

SQL:

```
SELECT emp_name as EmployeeName
FROM sample_table_for_test_ins_query
WHERE input_num = ?
```

入力値

IN 1	テスト実行値	データ型	論理名	削除
	<input type="text"/>	VARCHAR	日本語: 条件1 英語: Condition1 中国語: 条件1	-

出力値

OUT 1	カラム名	データ型	論理名	削除
	emp_name	VARCHAR	日本語: 項目1 英語: item1 中国語: 項目1	-

テスト実行 登録

5. 「テスト実行値」にテスト用の値を入力し、「テスト実行」をクリックします。

データベース編集[シェアードDBクエリ]

データベース種別: シェアードDBクエリ
データベース名: シェアードDBクエリサンプル

クエリ設定 管理会社設定

接続ID: bis_test_db

SQL:

```
SELECT emp_name as EmployeeName
FROM sample_table_for_test_ins_query
WHERE input_num = ?
```

入力値

IN 1	テスト実行値	データ型	論理名	削除
	<input type="text" value="1"/>	VARCHAR	日本語: 条件1 英語: Condition1 中国語: 条件1	-

出力値

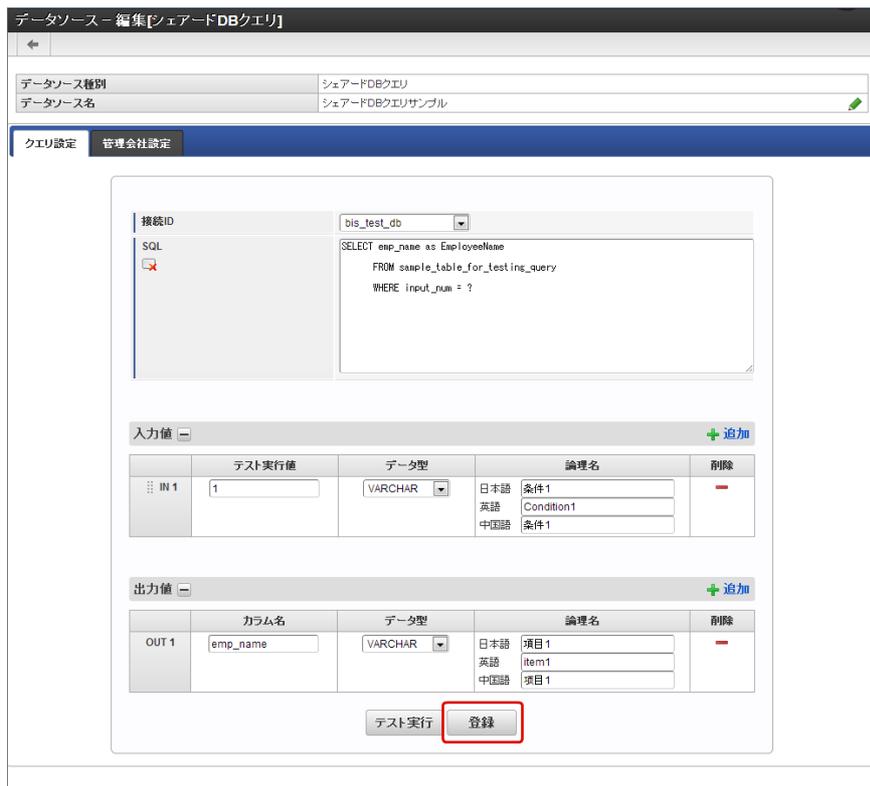
OUT 1	カラム名	データ型	論理名	削除
	emp_name	VARCHAR	日本語: 項目1 英語: item1 中国語: 項目1	-

テスト実行 登録

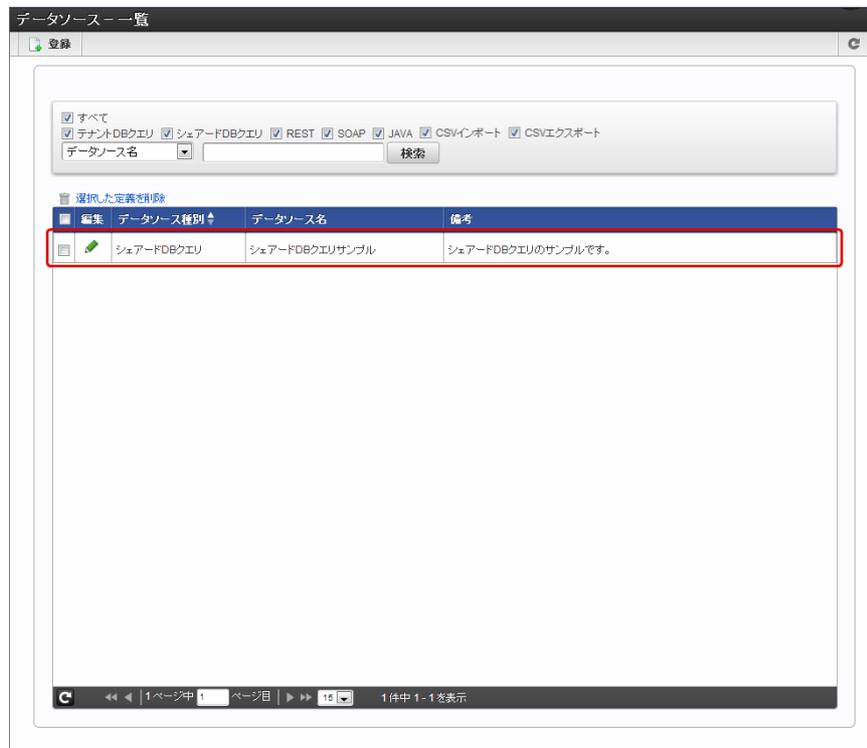
6. テストが成功した場合、以下のメッセージが表示されます。



7. 内容を確認して、「登録」をクリックします。



8. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。



REST（非推奨）

「データソース - 編集[REST]」画面の機能と各部の説明

「データソース - 編集[REST]」画面の内容は以下の通りです。

データソース種類 REST
データソース名 (1) rest_sample

REST設定 管理会社設定

(2) サービスタイプ REST
(3) RESTのURL
(4) URI
(5) メソッド GET
(6) 返却形式 JSON
(7) 返却文字コード UTF-8

(8) リクエスト

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	<input type="checkbox"/>	companyCd	string		なし	-

(9) レスポンス

	フィールド	データ型	フォーマット	親オブジェクト	削除
1	Company	object		なし	-
2	companyCd	string		1	-
3	companyName	string		1	-
4	companyAddress	string		1	-
5	companyBusiness	string		1	-

(10)

- データソース名
「データソース - 新規登録」画面で設定したデータソース名を表示します。
- サービスタイプ
データソース種別として選択した「REST」が表示されます。
- RESTのURL
REST APIのエントリーポイントとなるアドレスを記述します。
右の をクリックすると、入力されているURLをクリアします。
- URI
RESTのURLに続くリソースURIを記述します。

先頭に「/」を記述する必要があります。

5. メソッド
RESTのメソッドを設定します。
「GET」または「POST」が選択できます。
6. 返却形式
レスポンスの形式を設定します。
「JSON」または「XML」が選択できます。
7. 返却文字コード
レスポンスの文字コードを設定します。
8. リクエスト
リクエストパラメータを設定します。
9. レスポンス
レスポンスフィールドを設定します。
10. 登録
設定した内容をデータソース定義として登録します。

リクエストパラメータ、レスポンスフィールドの機能と各部の説明

リクエストパラメータ、レスポンスフィールドの設定は以下の通りです。

リクエストの設定項目

「リクエスト」の設定の内容は以下の通りです。
左の列番号をドラッグすることで、並び順を入れ替えることができます。

リクエスト (7) +追加						
	(1) HEADER	(2) パラメータ	(3) データ型	(4) フォーマット	(5) 親オブジェクト	(6) 削除
...	<input type="checkbox"/>	appid	string		なし	-
...	<input type="checkbox"/>	output	string		なし	-
...	<input type="checkbox"/>	callback	string		なし	-
...	<input type="checkbox"/>	area	string		なし	-
...	<input type="checkbox"/>	datetime	date	yyyy-MM-dd	なし	-
...	<input type="checkbox"/>	latest	number		なし	-

1. HEADER
「パラメータ」をリクエストヘッダに含めるかどうかを設定します。
チェックをオンに設定したパラメータはリクエストヘッダとして扱います。
(JAVAの場合には、この項目は表示されません。)
 2. パラメータ
リクエストパラメータ名を設定します。
- 注意**

マルチバイト文字を入力可能にしておりますが、連携システムのインタフェース仕様やデータソース仕様により、使用できない場合があります。
そのため、シングルバイト文字の使用を推奨します。
3. データ型
リクエストパラメータのデータ型を設定します。
設定できるデータ型は、object / array / string / number / date / boolean です。
 4. フォーマット
「データ型」に「date」を設定した場合、日付の書式を設定します。
 5. 親オブジェクト
リクエストオブジェクトが階層化されている場合に、上位階層に相当するパラメータの列番号を指定します。
親オブジェクトに指定可能なパラメータは、「データ型」が「object」、または「array」となっているものが対象です。
 6. 削除
クリックするとリクエストパラメータを1行削除します。
 7. 追加
クリックするとリクエストパラメータの入力欄を1行追加します。

レスポンスの設定項目

「レスポンス」の設定の内容は以下の通りです。

左の列番号をドラッグすることで、並び順を入れ替えることができます。

レスポンス (6) +追加					
	(1)フィールド	(2)データ型	(3)フォーマット	(4)親オブジェクト	(5)削除
1	responseObject	object		なし	-
2	Area	string		1	-
3	Number	number		1	-
4	Population	number		1	-
5	Date	date	yyyy-MM-dd	1	-
6	Hour	number		1	-
7	Min	number		1	-

1. フィールド

レスポンスフィールド名を設定します。



注意

マルチバイト文字を入力可能にしておりますが、連携システムのインターフェース仕様やデータソース仕様により、使用できない場合があります。

そのため、シングルバイト文字の使用を推奨します。

2. データ型

レスポンスフィールドのデータ型を設定します。

設定できるデータ型は、object / array / string / number / date / boolean です。

3. フォーマット

「データ型」に「date」を設定した場合、日付の書式を設定します。

4. 親オブジェクト

レスポンスオブジェクトが階層化されている場合に、上位階層に相当するフィールドの列番号を指定します。

親オブジェクトに指定可能なパラメータは、「データ型」が「object」、または「array」となっているものが対象です。

5. 削除

クリックするとレスポンスフィールドを1行削除します。

6. 追加

クリックするとレスポンスフィールドの入力欄を1行追加します。

「データソース - 編集[REST]」画面の操作手順

「データソース - 編集[REST]」画面の操作手順について説明します。

1. 「RESTのURL」、「URI」、「メソッド」、「返却形式」、「返却文字コード」を設定します。

データソース - 編集[REST]

データソース種別: REST
データソース名: RESTデータソースサンプル

REST設定 管理会社設定

サービスタイプ: REST

RESTのURL: http:// /RESTA1

URI: /RestRetPt10.php

メソッド: GET

返却形式: JSON

返却文字コード: UTF-8

リクエスト +追加

HEADER	パラメータ	データ型	フォーマット	親オブジェクト	削除
+追加					

レスポンス +追加

フィールド	データ型	フォーマット	親オブジェクト	削除
+追加				

登録

2. 「リクエスト」の「追加」をクリックします。

データソース 編集[REST]

データソース種別 REST
データソース名 RESTデータソースサンプル

REST設定 管理会社設定

サービスタイプ REST
RESTのURL http:// /RESTA1
URI /RestRelPt10.php
メソッド GET
返却形式 JSON
返却文字コード UTF-8

リクエスト +追加

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト	削除

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除

登録

3. 「HEADER」、「パラメータ」、「データ型」、「親オブジェクト」を設定します。「データ型」を「Date」とする場合には、フォーマットも指定してください。

データソース 編集[REST]

データソース種別 REST
データソース名 RESTデータソースサンプル

REST設定 管理会社設定

サービスタイプ REST
RESTのURL http:// /RESTA1
URI /RestRelPt10.php
メソッド GET
返却形式 JSON
返却文字コード UTF-8

リクエスト +追加

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	<input type="checkbox"/>	<input type="text"/>	string	<input type="text"/>	なし	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除

登録

4. 6~7の手順を繰り返し、必要なリクエストパラメータを設定します。

データソース - 編集[REST]

データソース種別 REST
データソース名 RESTデータソースサンプル

REST設定 管理会社設定

サービスタイプ REST
RESTのURL http:// /RESTA1
URI /RestRetPt10.php
メソッド GET
返却形式 JSON
返却文字コード UTF-8

リクエスト +追加

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	<input type="checkbox"/>	param_plus	number		なし	-
2	<input type="checkbox"/>	param_array	array		なし	-
3	<input type="checkbox"/>		object		2	-
4	<input type="checkbox"/>	param_str	string		3	-
5	<input type="checkbox"/>	parame_num	number		3	-
6	<input type="checkbox"/>	param_date	date	y/M/d	3	-
7	<input type="checkbox"/>	param_bln	boolean		3	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除

登録

5. 「レスポンス」の「追加」をクリックします。

データソース - 編集[REST]

データソース種別 REST
データソース名 RESTデータソースサンプル

REST設定 管理会社設定

サービスタイプ REST
RESTのURL http:// /RESTA1
URI /RestRetPt10.php
メソッド GET
返却形式 JSON
返却文字コード UTF-8

リクエスト +追加

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	<input type="checkbox"/>	param_plus	number		なし	-
2	<input type="checkbox"/>	param_array	array		なし	-
3	<input type="checkbox"/>		object		2	-
4	<input type="checkbox"/>	param_str	string		3	-
5	<input type="checkbox"/>	parame_num	number		3	-
6	<input type="checkbox"/>	param_date	date	y/M/d	3	-
7	<input type="checkbox"/>	param_bln	boolean		3	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除

登録

6. 「フィールド」、「データ型」、「親オブジェクト」を設定します。
データ型を「Date」とする場合には、フォーマットも指定してください。

データソース 編集[REST]

データソース種別 REST
データソース名 RESTデータソースサンプル

REST設定 管理会社設定

サービスタイプ REST
RESTのURL http:// /RESTA1
URI /RestRetIP10.php
メソッド GET
返却形式 JSON
返却文字コード UTF-8

リクエスト +追加

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	<input type="checkbox"/>	param_plus	number		なし	-
2	<input type="checkbox"/>	param_array	array		なし	-
3	<input type="checkbox"/>		object		2	-
4	<input type="checkbox"/>	param_str	string		3	-
5	<input type="checkbox"/>	param_num	number		3	-
6	<input type="checkbox"/>	param_date	date	yM/d	3	-
7	<input type="checkbox"/>	param_bin	boolean		3	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除
1	<input type="text"/>	string		なし	-

登録

7. 9～10の手順を繰り返し、必要なレスポンスフィールドを設定します。

データソース 編集[REST]

データソース種別 REST
データソース名 RESTデータソースサンプル

REST設定 管理会社設定

サービスタイプ REST
RESTのURL http:// /RESTA1
URI /RestRetIP10.php
メソッド GET
返却形式 JSON
返却文字コード UTF-8

リクエスト +追加

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	<input type="checkbox"/>	param_plus	number		なし	-
2	<input type="checkbox"/>	param_array	array		なし	-
3	<input type="checkbox"/>		object		2	-
4	<input type="checkbox"/>	param_str	string		3	-
5	<input type="checkbox"/>	param_num	number		3	-
6	<input type="checkbox"/>	param_date	date	yM/d	3	-
7	<input type="checkbox"/>	param_bin	boolean		3	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除
1	param_array	array		なし	-
2		object		1	-
3	param_str	string		2	-
4	param_num	number		2	-
5	param_date	date	yM/d	2	-
6	param_bin	boolean		2	-

登録

8. 「登録」をクリックします。

データソース 編集[REST]

データソース種別 REST
データソース名 RESTデータソースサンプル

REST設定 管理会社設定

サービスタイプ REST
RESTのURL http:// /RESTA1
URI /RestRePlt10.php
メソッド GET
返却形式 JSON
返却文字コード UTF-8

リクエスト +追加

	HEADER	パラメータ	データ型	フォーマット	観オブジェクト	削除
1	<input type="checkbox"/>	param_plus	number		なし	-
2	<input type="checkbox"/>	param_array	array		なし	-
3	<input type="checkbox"/>		object		2	-
4	<input type="checkbox"/>	param_str	string		3	-
5	<input type="checkbox"/>	param_num	number		3	-
6	<input type="checkbox"/>	param_date	date	yM/d	3	-
7	<input type="checkbox"/>	param_bln	boolean		3	-

レスポンス +追加

	フィールド	データ型	フォーマット	観オブジェクト	削除
1	param_array	array		なし	-
2		object		1	-
3	param_str	string		2	-
4	param_num	number		2	-
5	param_date	date	yM/d	2	-
6	param_bln	boolean		2	-

登録

9. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。

データソース 一覧

登録

すべて
 テナントDBクエリ シェアードDBクエリ REST SOAP JAVA CSVインポート CSVエクスポート
 データソース名 検索

選択した定義を削除

編集	データソース種別	データソース名	備考
<input type="checkbox"/>	REST	RESTデータソースサンプル	RESTデータソースのサンプルです。

1ページ中 1 ページ目 15 1件中 1-1を表示

SOAP (非推奨)

「データソース - 編集[SOAP]」画面の機能と各部の説明

「データソース - 編集[SOAP]」画面の内容は以下の通りです。

データソース 編集[SOAP]

データソース種別 SOAP
データソース名 (1) soap_sample

SOAP設定 管理会社設定

(2) サービスタイプ SOAP (4)

(3) WSDLのURL (読み込み)

(5) サービス名 WSMasterTest

(6) ポート WSMasterTestHttpSoap11Endpoint

(7) オペレーション get/toArray

(8) リクエスト

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト
リクエスト					

(9) レスポンス

	フィールド	データ型	フォーマット	親オブジェクト
1	get/toArray/Response	object		なし
2	return	array		1
3		string		2

(10) 登録

- データソース名
「データソース - 新規登録」画面で設定したデータソース名を表示します。
- サービスタイプ
データソース種別として選択した「SOAP」が表示されます。
- WSDLのURL
WSDLの記述されている場所(URL)を指定します。
右の  をクリックすると、入力されているURLをクリアします。
- 読み込み
クリックするとWSDLの内容を読み込み、「サービス名」、「ポート」、「オペレーション」を表示します。
- サービス名
読み込んだWSDLに設定されているサービス名を表示します。
- ポート
読み込んだWSDLに設定されているポートを表示します。
複数のポートがある場合は、利用するポートを選択します。
- オペレーション
読み込んだWSDLに設定されているオペレーションを表示します。
複数のオペレーションがある場合は、利用するオペレーションを選択します。
- リクエスト
リクエストパラメータを設定します。
(SOAPの場合、「WSDLのURL」、「サービス名」、「ポート」、「オペレーション」を設定すると表示されます。)
- レスポンス
レスポンスフィールドを設定します。
(SOAPの場合、「WSDLのURL」、「サービス名」、「ポート」、「オペレーション」を設定すると表示されます。)
- 登録
設定した内容をデータソース定義として登録します。

リクエストパラメータ、レスポンスフィールドの機能と各部の説明

リクエストパラメータ、レスポンスフィールドの設定は以下の通りです。

リクエストの設定項目

「リクエスト」の設定の内容は以下の通りです。

左の列番号をドラッグすることで、並び順を入れ替えることができます。

リクエスト (7) +追加						
	(1) HEADER	(2) パラメータ	(3) データ型	(4) フォーマット	(5) 親オブジェクト	(6) 削除
...	1	<input type="checkbox"/> appid	string		なし	-
...	2	<input type="checkbox"/> output	string		なし	-
...	3	<input type="checkbox"/> callback	string		なし	-
...	4	<input type="checkbox"/> area	string		なし	-
...	5	<input type="checkbox"/> datetime	date	yyyy-MM-dd	なし	-
...	6	<input type="checkbox"/> latest	number		なし	-

1. HEADER

「パラメータ」をリクエストヘッダに含めるかどうかを設定します。
 チェックをオンに設定したパラメータはリクエストヘッダとして扱います。
 (JAVAの場合には、この項目は表示されません。)

2. パラメータ

リクエストパラメータ名を設定します。



注意

マルチバイト文字を入力可能にしておりますが、連携システムのインタフェース仕様やデータソース仕様により、使用できない場合があります。
 そのため、シングルバイト文字の使用を推奨します。

3. データ型

リクエストパラメータのデータ型を設定します。
 設定できるデータ型は、object / array / string / number / date / boolean です。

4. フォーマット

「データ型」に「date」を設定した場合、日付の書式を設定します。

5. 親オブジェクト

リクエストオブジェクトが階層化されている場合に、上位階層に相当するパラメータの列番号を指定します。
 親オブジェクトに指定可能なパラメータは、「データ型」が「object」、または「array」となっているものが対象です。

6. 削除

クリックするとリクエストパラメータを1行削除します。

7. 追加

クリックするとリクエストパラメータの入力欄を1行追加します。

レスポンスの設定項目

「レスポンス」の設定の内容は以下の通りです。
 左の列番号をドラッグすることで、並び順を入れ替えることができます。

レスポンス (6) +追加						
	(1) フィールド	(2) データ型	(3) フォーマット	(4) 親オブジェクト	(5) 削除	
...	1	responseObject	object		なし	-
...	2	Area	string		1	-
...	3	Number	number		1	-
...	4	Population	number		1	-
...	5	Date	date	yyyy-MM-dd	1	-
...	6	Hour	number		1	-
...	7	Min	number		1	-

1. フィールド

レスポンスフィールド名を設定します。



注意

マルチバイト文字を入力可能にしておりますが、連携システムのインタフェース仕様やデータソース仕様により、使用できない場合があります。
 そのため、シングルバイト文字の使用を推奨します。

2. データ型

レスポンスフィールドのデータ型を設定します。
 設定できるデータ型は、object / array / string / number / date / boolean です。

3. フォーマット

「データ型」に「date」を設定した場合、日付の書式を設定します。

4. 親オブジェクト

レスポンスオブジェクトが階層化されている場合に、上位階層に相当するフィールドの列番号を指定します。

親オブジェクトに指定可能なパラメータは、「データ型」が「object」、または「array」となっているものが対象です。

5. 削除

クリックするとレスポンスフィールドを1行削除します。

6. 追加

クリックするとレスポンスフィールドの入力欄を1行追加します。

「データソース - 編集[SOAP]」画面の操作手順

「データソース - 編集[SOAP]」画面の操作手順について説明します。

1. 「WSDLのURL」に利用するWebサービスのアドレスを入力します。

The screenshot shows the 'データソース - 編集[SOAP]' interface. At the top, there are fields for 'データソース種別' (SOAP) and 'データソース名' (SOAPデータソースサンプル). Below this is a 'SOAP設定' section with a '管理会社設定' tab. The main area contains a table with the following fields: 'サービスタイプ' (SOAP), 'WSDLのURL' (http://.../mart/services/WSTest10?wsdl), 'サービス名', 'ポート', and 'オペレーション'. The 'WSDLのURL' field and the '読み込み' button are highlighted with a red box.

2. 「読み込み」をクリックします。

This screenshot is identical to the previous one, but the '読み込み' button next to the 'WSDLのURL' field is highlighted with a red box, indicating the next step in the process.

3. 正しく読み込めると、WSDLの読み込みに成功した内容のメッセージが表示されます。

データソース 編集【SOAP】

WSDLの読み込みに成功しました。

データソース種別: SOAP
データソース名: SOAPデータソースサンプル

SOAP設定 管理会社設定

サービスタイプ: SOAP

WSDLのURL:

サービス名: WSTest10

ポート:

オペレーション:

リクエスト

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト
1	<input type="checkbox"/>	wsTest10	object		なし
2	<input type="checkbox"/>	param	array		1
3	<input type="checkbox"/>		object		2
4	<input type="checkbox"/>	res01Str	string		3
5	<input type="checkbox"/>	res02Int	number		3
6	<input type="checkbox"/>	res03Date	date	yyyy-MM-ddTHH:mm:ss	3
7	<input type="checkbox"/>	res04Boolean	boolean		3

レスポンス

	フィールド	データ型	フォーマット	親オブジェクト
1	wsTest10Response	object		なし
2	return	array		1
3		object		2
4	res01Str	string		3
5	res02Int	number		3
6	res03Date	date	yyyy-MM-ddTHH:mm:ss	3
7	res04Boolean	boolean		3

4. 「ポート」と「オペレーション」を設定します。

データソース 編集【SOAP】

データソース種別: SOAP
データソース名: SOAPデータソースサンプル

SOAP設定 管理会社設定

サービスタイプ: SOAP

WSDLのURL:

サービス名: WSTest10

ポート:

オペレーション:

リクエスト

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト
1	<input type="checkbox"/>	wsTest10	object		なし
2	<input type="checkbox"/>	param	array		1
3	<input type="checkbox"/>		object		2
4	<input type="checkbox"/>	res01Str	string		3
5	<input type="checkbox"/>	res02Int	number		3
6	<input type="checkbox"/>	res03Date	date	yyyy-MM-ddTHH:mm:ss	3
7	<input type="checkbox"/>	res04Boolean	boolean		3

レスポンス

	フィールド	データ型	フォーマット	親オブジェクト
1	wsTest10Response	object		なし
2	return	array		1
3		object		2
4	res01Str	string		3
5	res02Int	number		3
6	res03Date	date	yyyy-MM-ddTHH:mm:ss	3
7	res04Boolean	boolean		3

5. 選択したポートとオペレーションに合わせた「リクエスト」と「レスポンス」が表示されることを確認し「登録」をクリックします。

データソース 編集[SOAP]

データソース種別 SOAP
データソース名 SOAPデータソースサンプル

SOAP設定 管理会社設定

サービスタイプ SOAP
WSDLのURL http:// /mart/services/WSTest10?wsdl 読み込み
サービス名 WSTest10
ポート WSTest10HttpSoap11Endpoint
オペレーション wsTest10

リクエスト

	HEADER	パラメータ	データ型	フォーマット	親オブジェクト
1	<input type="checkbox"/>	wsTest10	object		なし
2	<input type="checkbox"/>	param	array		1
3	<input type="checkbox"/>		object		2
4	<input type="checkbox"/>	res01Str	string		3
5	<input type="checkbox"/>	res02Int	number		3
6	<input type="checkbox"/>	res03Date	date	yyyy-MM-ddTHH:mm:ss	3
7	<input type="checkbox"/>	res04Boolean	boolean		3

レスポンス

	フィールド	データ型	フォーマット	親オブジェクト
1	wsTest10Response	object		なし
2	return	array		1
3		object		2
4	res01Str	string		3
5	res02Int	number		3
6	res03Date	date	yyyy-MM-ddTHH:mm:ss	3
7	res04Boolean	boolean		3

登録

6. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。

データソース 一覧

登録

すべて
 テナントDBクエリ シェアードBクエリ REST SOAP JAVA CSVインポート CSVエクスポート
 データソース名 検索

選択した定義を削除

編集	データソース種別	データソース名	備考
<input checked="" type="checkbox"/>	SOAP	SOAPデータソースサンプル	SOAPデータソースのサンプルです。

1ページ中 1 ページ目 10 1件中 1-1を表示

i コラム

2013 Spring以前で作成したSOAP定義を2013 Summer以降で参照した場合、仕様によりパラメータが初期表示されないことがあります。

そのような場合でパラメータの確認が必要な場合には、以下の手順でパラメータを表示することができます。

- 「データソース - 編集[SOAP]」画面の「WSDLのURL」に参照したいWebサービスのURLを入力します。
- 「読み込み」をクリックします。

パラメータ表示後、「更新」をクリックするとパラメータが初期表示されます。

Contents

- 「データソース - 編集[JAVA]」画面の機能と各部の説明
- リクエストパラメータ、レスポンスフィールドの機能と各部の説明
- 「データソース - 編集[JAVA]」画面の操作手順
- 設定例

「データソース - 編集[JAVA]」画面の機能と各部の説明

「データソース - 編集[JAVA]」画面の内容は以下の通りです。

データソース - 編集[JAVA]

データソース種別: JAVA
データソース名 (1): ユーザ情報取得関数

JAVA設定 管理会社設定

(2) サービスタイプ: JAVA

(3) 実行ファイル: クラスファイル一覧
SystemFunction.jar (5)

(4) 実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction (読み込み)

(6) メソッド: getUserInfo

(7) リクエスト

	パラメータ	データ型	フォーマット	親オブジェクト	削除
...					

(8) レスポンス

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	root	object		なし	-
2	user	string		1	-
3	username	string		1	-
4	locale	string		1	-
5	encoding	string		1	-
6	mail	string		1	-
7	mbtmail	string		1	-

(9) 登録

- データソース名
「データソース - 新規登録」画面で設定したデータソース名を表示します。
- サービスタイプ
データソース種別として選択した「JAVA」が表示されます。
- 実行ファイル
JAVA連携として利用するクラスを含むJARファイルを指定します。
- 実行クラス
JARファイルのクラス名を完全修飾クラス名で指定します。
- 読み込み
クリックすると実行クラスに含まれるメソッドが「メソッド」に表示されます。
- メソッド
読み込んだクラスに含まれるメソッドを表示します。
複数のメソッドがある場合は、利用するメソッドを選択します。
- リクエスト
リクエストパラメータを設定します。
- レスポンス
レスポンスパラメータを設定します。
- 登録
設定した内容をデータソース定義として登録します。

リクエストパラメータ、レスポンスフィールドの機能と各部の説明

リクエストパラメータ、レスポンスフィールドの設定は以下の通りです。

リクエストの設定項目

「リクエスト」の設定の内容は以下の通りです。
 左の列番号をドラッグすることで、並び順を入れ替えることができます。

リクエスト (7) +追加						
	(1) HEADER	(2) パラメータ	(3) データ型	(4) フォーマット	(5) 親オブジェクト	(6) 削除
⋮ 1	<input type="checkbox"/>	appid	string		なし	—
⋮ 2	<input type="checkbox"/>	output	string		なし	—
⋮ 3	<input type="checkbox"/>	callback	string		なし	—
⋮ 4	<input type="checkbox"/>	area	string		なし	—
⋮ 5	<input type="checkbox"/>	datetime	date	yyyy-MM-dd	なし	—
⋮ 6	<input type="checkbox"/>	latest	number		なし	—

1. HEADER

「パラメータ」をリクエストヘッダに含めるかどうかを設定します。
 チェックをオンに設定したパラメータはリクエストヘッダとして扱います。
 (JAVAの場合には、この項目は表示されません。)

2. パラメータ

リクエストパラメータ名を設定します。



注意

マルチバイト文字を入力可能にしておりますが、連携システムのインタフェース仕様やデータソース仕様により、使用できない場合があります。
 そのため、シングルバイト文字の使用を推奨します。

3. データ型

リクエストパラメータのデータ型を設定します。
 設定できるデータ型は、object / array / string / number / date / boolean です。



コラム

各データ型の仕様については、「IM-BIS 仕様書」 - 「IM-BIS で外部連携として利用できるJavaプログラムの仕様」を参照してください。

4. フォーマット

「データ型」に「date」を設定した場合、日付の書式を設定します。



注意

「フォーマット」に設定した日付の書式は、現在適用されません。
 「フォーマット」は入力必須のため、仮設定を入力してください。

5. 親オブジェクト

リクエストオブジェクトが階層化されている場合に、上位階層に相当するパラメータの列番号を指定します。
 親オブジェクトに指定可能なパラメータは、「データ型」が「object」、または「array」となっているものが対象です。

6. 削除

クリックするとリクエストパラメータを1行削除します。

7. 追加

クリックするとリクエストパラメータの入力欄を1行追加します。

レスポンスの設定項目

「レスポンス」の設定の内容は以下の通りです。
 左の列番号をドラッグすることで、並び順を入れ替えることができます。

レスポンス (6) +追加					
	(1) フィールド	(2) データ型	(3) フォーマット	(4) 親オブジェクト	(5) 削除
⋮ 1	ResponseObject	object		なし	—
⋮ 2	Area	string		1	—
⋮ 3	Number	number		1	—
⋮ 4	Population	number		1	—
⋮ 5	Date	date	yyyy-MM-dd	1	—
⋮ 6	Hour	number		1	—
⋮ 7	Min	number		1	—

1. フィールド

レスポンスフィールド名を設定します。



注意

マルチバイト文字を入力可能にしておりますが、連携システムのインターフェース仕様やデータソース仕様により、使用できない場合があります。

そのため、シングルバイト文字の使用を推奨します。

2. データ型

レスポンスフィールドのデータ型を設定します。

設定できるデータ型は、object / array / string / number / date / boolean です。



コラム

各データ型の仕様については、「IM-BIS 仕様書」 - 「IM-BIS で外部連携として利用できるJavaプログラムの仕様」を参照してください。

3. フォーマット

「データ型」に「date」を設定した場合、日付の書式を設定します。



注意

「フォーマット」に設定した日付の書式は、現在適用されません。

「フォーマット」は入力必須のため、仮設定を入力してください。

4. 親オブジェクト

レスポンスオブジェクトが階層化されている場合に、上位階層に相当するフィールドの列番号を指定します。

親オブジェクトに指定可能なパラメータは、「データ型」が「object」、または「array」となっているものが対象です。

5. 削除

クリックするとレスポンスフィールドを1行削除します。

6. 追加

クリックするとレスポンスフィールドの入力欄を1行追加します。

「データソース - 編集[JAVA]」画面の操作手順

「データソース - 編集[JAVA]」画面の操作手順について説明します。

1. 「クラスファイル一覧」をクリックします。

データソース - 編集[JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 | 管理会社設定

サービスタイプ: JAVA
実行ファイル: **クラスファイル一覧**
実行クラス: [] [読み込み]
メソッド: []

リクエスト

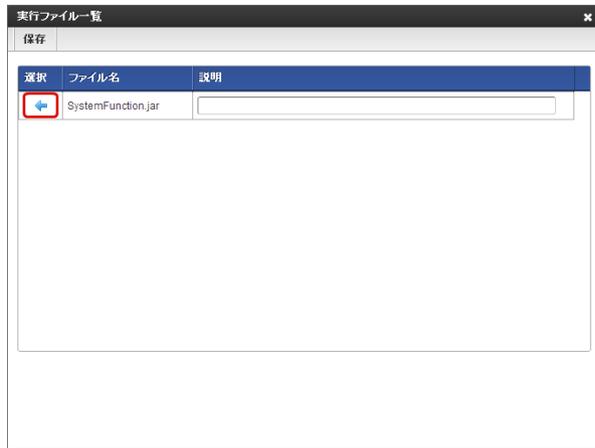
パラメータ	データ型	フォーマット	親オブジェクト	削除

レスポンス

パラメータ	データ型	フォーマット	親オブジェクト	削除

登録

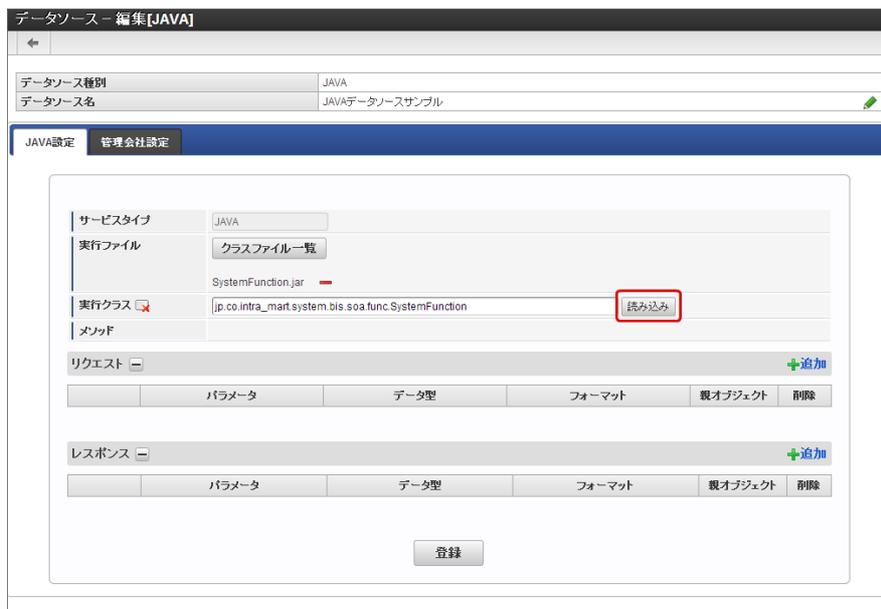
2. 「実行ファイル一覧」画面から設定するファイルの「選択」をクリックします。



3. 「実行クラス」に、設定したファイルに含まれるクラスの完全修飾クラス名を入力します。



4. 「読み込み」をクリックします。



5. 読み込みに成功すると「メソッド」が表示されますので、利用するメソッドを選択します。

データソース 編集 [JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 管理会社設定

サービスタイプ: JAVA
実行ファイル: クラスファイル一覧
SystemFunction.jar

実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction 読み込み

メソッド: **getUserInfo**

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除

レスポンス +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除

登録

6. 「リクエスト」の「追加」をクリックします。

データソース 編集 [JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 管理会社設定

サービスタイプ: JAVA
実行ファイル: クラスファイル一覧
SystemFunction.jar

実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction 読み込み

メソッド: getUserInfo

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除

レスポンス +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除

登録

7. 「パラメータ」、「データ型」、「親オブジェクト」を設定します。
「データ型」を「Date」とする場合には、フォーマットも指定してください。

データソース 編集 [JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 管理会社設定

サービスタイプ: JAVA
実行ファイル: クラスファイル一覧
SystemFunction.jar

実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction 読み込み

メソッド: getUserInfo

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1		string		なし	-

レスポンス +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除

登録

8. 6~7の手順を繰り返し、必要なリクエストパラメータを設定します。

データソース 編集 [JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 管理会社設定

サービスタイプ: JAVA
実行ファイル: クラスファイル一覧
SystemFunction.jar

実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction 読み込み

メソッド: getUserInfo

リクエスト

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	root	object		なし	-
2	parameter1	string		1	-
3	parameter2	number		1	-

レスポンス

	パラメータ	データ型	フォーマット	親オブジェクト	削除

登録

9. 「レスポンス」の「追加」をクリックします。

データソース 編集 [JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 管理会社設定

サービスタイプ: JAVA
実行ファイル: クラスファイル一覧
SystemFunction.jar

実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction 読み込み

メソッド: getUserInfo

リクエスト

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	root	object		なし	-
2	parameter1	string		1	-
3	parameter2	number		1	-

レスポンス

	パラメータ	データ型	フォーマット	親オブジェクト	削除

登録

10. 「パラメータ」、「データ型」、「親オブジェクト」を設定します。
「データ型」を「Date」とする場合には、フォーマットも指定してください。

データソース 編集 [JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 管理会社設定

サービスタイプ: JAVA
実行ファイル: クラスファイル一覧
SystemFunction.jar
実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction
メソッド: getUserInfo

リクエスト

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	root	object		なし	-
2	parameter1	string		1	-
3	parameter2	number		1	-

レスポンス

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1		string		なし	-

登録

11. 9~10の手順を繰り返し、必要なレスポンスパラメータを設定します。

データソース 編集 [JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 管理会社設定

サービスタイプ: JAVA
実行ファイル: クラスファイル一覧
SystemFunction.jar
実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction
メソッド: getUserInfo

リクエスト

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	root	object		なし	-
2	parameter1	string		1	-
3	parameter2	number		1	-

レスポンス

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	responseRoot	object		なし	-
2	responseP1	string		1	-
3	responseP2	number		1	-
4	responseP3	boolean		1	-

登録

12. 「登録」をクリックします。

データソース - 編集 [JAVA]

データソース種別: JAVA
データソース名: JAVAデータソースサンプル

JAVA設定 | 管理会社設定

サービスタイプ: JAVA
実行ファイル: クラスファイル一覧
SystemFunction.jar
実行クラス: jp.co.intra_mart.system.bis.soa.func.SystemFunction
メソッド: getUserInfo

リクエスト

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	root	object		なし	-
2	parameter1	string		1	-
3	parameter2	number		1	-

レスポンス

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	responseRoot	object		なし	-
2	responseP1	string		1	-
3	responseP2	number		1	-
4	responseP3	boolean		1	-

登録

13. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。

データソース - 一覧

登録

すべて
 テナントDBクエリ
 シェアードBクエリ
 REST
 SOAP
 JAVA
 CSVインポート
 CSVエクスポート

データソース名: 検索

選択した定義を削除

編集	データソース種別 ↑	データソース名	備考
<input checked="" type="checkbox"/>	JAVA	JAVAデータソースサンプル	JAVAデータソースのサンプルです。

1ページ中 1 ページ目 15 1件中 1-1を表示

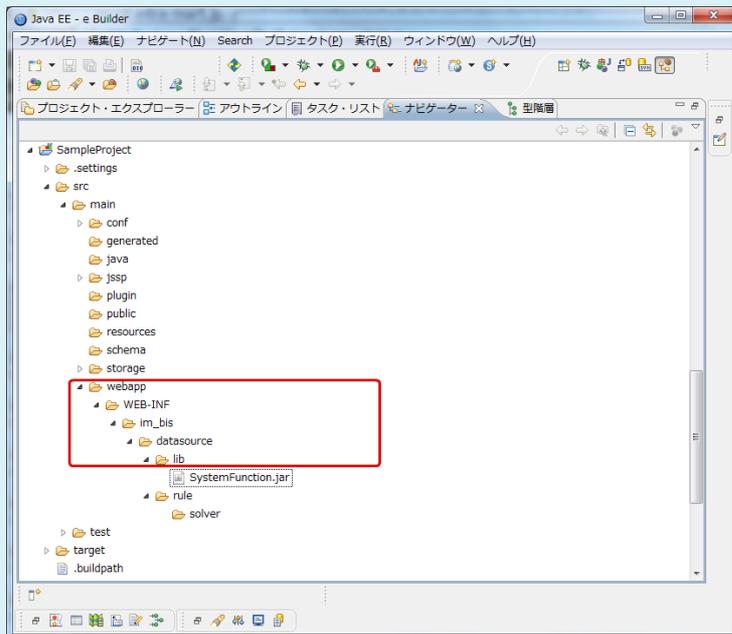
i コラム

JAVAプログラムは、ユーザモジュールとして、事前に配置する必要があります。

JAVAプログラムの配置先は、「IM-BIS セットアップガイド」 - 「バーチャルテナントに関する設定」で設定できます。

デフォルトの設定では、「WEB-INF/im_bis/datasource/lib」です。

ユーザモジュールとしては、以下のようにJAVAプログラムを配置します。



設定例

外部連携 (JAVA) の設定例については、「IM-BIS システム管理者操作ガイド」 「外部連携サンプルプログラム (JAVA)」を参照してください。

LogicDesigner

Contents

- 「データソース - 編集[LogicDesigner]」画面の機能と各部の説明
- 「データソース - 編集[LogicDesigner]」画面の操作手順

「データソース - 編集[LogicDesigner]」画面の機能と各部の説明

「データソース - 編集[LogicDesigner]」画面の内容は以下の通りです。

データソース - 編集[LogicDesigner]

データソース種別: LogicDesigner
 データソース名 (1): LDTEST01

LogicDesigner設定 管理会社設定

(2) バージョン: バージョン指定なし バージョンを指定する (4)

(3) フロー定義: document_test

(5) バージョン番号: 1

(6) リクエスト

	パラメータ	データ型	親オブジェクト
1	string1	string	なし
2	integer1	integer	なし
3	date1	date	なし
4	character1	character	なし

(7) レスポンス

	パラメータ	データ型	親オブジェクト
1	result	string	なし

(8)

- データソース名
「データソース - 新規登録」画面で設定したデータソース名を表示します。
- バージョン
IM-LogicDesigner のフロー実行時のフロー定義のバージョンの扱いを指定します。

3. フロー定義

実行する IM-LogicDesigner のフロー定義を指定します。

4. フロー選択

クリックすると IM-LogicDesigner のフロー定義の一覧が表示されます。

5. バージョン番号

「バージョン」で、「バージョン指定なし」を選択した場合は、フロー選択時点での最新のバージョンが表示されます。
「バージョン」で、「バージョンを指定する」を選択した場合は、指定したバージョンが表示されます。

6. リクエスト

リクエストパラメータを表示します。

7. レスポンス

レスポンスパラメータを表示します。

8. 登録

設定した内容をデータソース定義として登録します。

「データソース - 編集[LogicDesigner]」画面の操作手順

「データソース - 編集[LogicDesigner]」画面の操作手順について説明します。

1. 「バージョン」を選択します。

The screenshot shows the 'データソース - 編集[LogicDesigner]' interface. At the top, there are fields for 'データソース種別' (LogicDesigner) and 'データソース名' (DocSample). Below this, there are two tabs: 'LogicDesigner設定' and '管理会社設定'. The 'LogicDesigner設定' tab is active. In the main content area, there are three rows: 'バージョン' with two radio buttons ('バージョン指定なし' selected and 'バージョンを指定する'), 'フロー定義' with a text input field and a 'フロー選択' button, and 'バージョン番号' with a text input field. A '登録' button is at the bottom. A red box highlights the radio buttons in the 'バージョン' row.

- バージョン指定なし
常に最新の IM-LogicDesigner のフロー定義のバージョンで処理を実行します。
フローの入出力設定が、データソース定義作成時から変更されていない場合に有効です。
- バージョンを指定する
特定のバージョンを指定して IM-LogicDesigner のフローを実行します。

コラム

データソースを登録後に IM-LogicDesigner のフローの入出力設定を変更した場合は、再度フロー選択を行い、リクエストとレスポンスの情報を更新してください。

2. 「フロー選択」をクリックします。

This screenshot is identical to the previous one, but the 'フロー選択' button in the 'フロー定義' row is highlighted with a red box.

3. 「フロー定義検索」画面から設定するフローを選択し、「決定」をクリックします。

ロジックフロー定義検索

フロー名を入力してください。

選択	フローID	フロー名
<input checked="" type="checkbox"/>	document_test	document_test
<input type="checkbox"/>	LDTEST2	LDTEST2

1ページ中 1 ページ目 10 2件中 1-2 を表示

i コラム

「バージョン」の選択で、「バージョン指定なし」を選んだ場合は、IM-LogicDesigner のフロー定義のみ選択してください。
「バージョン」の選択で、「バージョンを指定する」を選んだ場合は、IM-LogicDesigner のフロー定義の選択後、任意のバージョンを選択します。

4. 「リクエスト」「レスポンス」は、選択した IM-LogicDesigner のフローの定義から読み込み、自動で設定されます。

データソース編集[LogicDesigner]

データソース種別 LogicDesigner
データソース名 DocSample

LogicDesigner設定 管理会社設定

バージョン バージョン指定なし バージョンを指定する
フロー定義 document_test
バージョン番号 2

リクエスト			
	パラメータ	データ型	親オブジェクト
1	string1	string	なし
2	integer1	integer	なし
3	date1	date	なし
4	character1	character	なし

レスポンス			
	パラメータ	データ型	親オブジェクト
1	result	string	なし
2	map1	<input type="checkbox"/> map	なし

i コラム

IM-LogicDesigner で入出力に設定したデータ型の中に、外部連携に対応していないデータ型が含まれる場合は、フロー選択時に警告メッセージが表示されます。
その場合、データソース定義としては利用が可能ですが、外部連携では該当のデータ型へのマッピングはできません。
入出力データ型の仕様については、「[IM-LogicDesigner との連携の仕様](#)」を参照してください

5. 「登録」をクリックします。

データソース - 編集[LogicDesigner]

データソース種別: LogicDesigner
データソース名: DocSample

LogicDesigner設定 管理会社設定

バージョン: バージョン指定なし バージョンを指定する
フロー定義: document_test フロー選択
バージョン番号: 2

リクエスト

	パラメータ	データ型	親オブジェクト
1	string1	string	なし
2	integer1	integer	なし
3	date1	date	なし
4	character1	character	なし

レスポンス

	パラメータ	データ型	親オブジェクト
1	result	string	なし
2	map1	map	なし

登録

i コラム

IM-LogicDesigner のフロー定義は、事前に登録する必要があります。
IM-LogicDesigner のフロー定義については、「IM-LogicDesigner仕様書」を参照してください。

CSVインポート

「データソース - 編集[CSVインポート]」画面の機能と各部の説明

「データソース - 編集[CSVインポート]」画面の内容は以下の通りです。

データソース - 編集[CSVインポート]

データソース種別: CSVインポート
データソース名 (1): CSV_import_sample

CSVインポート 管理会社設定

(2) 文字コード*: SJIS
(3) 改行コード: CR+LF
(4) 区切り文字: カンマ
(5) スキップ行数(先頭)*: 0
(6) スキップ行数(末尾)*: 0

(7) データフォーマット

	列名	データ型	フォーマット	削除
1	列1	string		-
2	列2	string		-
3	列3	string		-
4	列4	string		-
5	列5	string		-

(8) 登録

- データソース名
「データソース - 新規登録」画面で設定したデータソース名を表示します。
- 文字コード
インポート対象ファイルの文字コードを指定します。
対象ファイルに合わせて、Javaでサポートされているエンコーディングセット (SJIS、Windows-31)、UTF-8などを指定します。
- 改行コード
インポート対象ファイルの改行コードを指定します。
「CR+LF」、「CR」、「LF」の中から設定します。
- 区切り文字
インポート対象ファイルの区切り文字を指定します。

「タブ」、「セミコロン」、「カンマ」、「スペース」の中から設定します。

5. スキップ行数（先頭）

インポート時に、先頭の読み込まない行数を指定します。

初期値に「0」が記述されています。

見出し行をインポートしたくないときなどは「1」を指定します。



注意

改行コードが正しく設定されていない場合、正常にスキップされない可能性があります。

6. スキップ行数（末尾）

インポート時に、末尾の読み込まない行数を指定します。

初期値に「0」が記述されています。



注意

改行コードが正しく設定されていない場合、正常にスキップされない可能性があります。

7. データフォーマット

- 列名：取り込むCSVの項目名を記述します。
- データ型：取り込むCSVの項目に対応するデータ型を設定します。
設定できるデータ型は、string / number / date / boolean です。
- フォーマット：データ型が date の場合に、フォーマットを設定します。
- 追加・削除：項目の追加・削除ができます。

8. 登録

設定した内容をデータソース定義として登録します。

「データソース - 編集[CSVインポート]」画面の操作手順

「データソース - 編集[CSVインポート]」画面の操作手順について説明します。

1. インポートファイルに合わせて、「文字コード」、「改行コード」、「区切り文字」、「スキップ行数（先頭）」、「スキップ行数（末尾）」を入力します。

データソース - 編集[CSVインポート]

データソース種別: CSVインポート
データソース名: CSVインポートサンプル

CSVインポート 管理会社設定

文字コード*: SJIS
改行コード: CR+LF
区切り文字: カンマ
スキップ行数(先頭)*: 1
スキップ行数(末尾)*: 0

データフォーマット [+追加]

	列名	データ型	フォーマット	削除
1	列1	string		-
2	列2	string		-
3	列3	string		-
4	列4	string		-
5	列5	string		-

登録

2. インポートファイルに合わせて、「データフォーマット」を入力し、「登録」ボタンをクリックします。

データソース 編集[CSVエクスポート]

データソース種別 CSVエクスポート
データソース名 (1) CSV_export_sample

CSVエクスポート 管理会社設定

(2) 文字コード SJIS
(3) 改行コード CR+LF
(4) 区切り文字 カンマ
(5) 見出し行 あり なし
(6) ダブルクォート 全データ型 stringのみ なし
(7) データフォーマット +追加

	列名	データ型	フォーマット	削除
1	列1	string		-
2	列2	string		-
3	列3	string		-
4	列4	string		-
5	列5	string		-

(8) 登録

1. データソース名

「データソース - 新規登録」画面で設定したデータソース名を表示します。

2. 文字コード

エクスポート対象ファイルの文字コードを指定します。

出力したい文字コードを、Javaでサポートされているエンコーディングセット（SJIS、Windows-31J、UTF-8など）を指定します。

3. 改行コード

エクスポート対象ファイルの改行コードを指定します。

「CR+LF」、「CR」、「LF」の中から設定します。

4. 区切り文字

エクスポート対象ファイルの区切り文字を指定します。

「タブ」、「セミコロン」、「カンマ」、「スペース」の中から設定します。

5. 見出し行

見出し行の有無を設定します。

- 「あり」の場合
データフォーマットの「列名」を、見出しとしてCSVファイルの1行目に出力します。
- 「なし」の場合
見出しを表示しません。
1行目にはエクスポート対象のデータが出力されます。

6. ダブルクォート

CSV出力時に、項目を「"（ダブルクォート）」で囲うかを設定します。

- 「全データ型」の場合
見出し行から対象データまで、すべての項目をダブルクォートで囲みます。
- 「Stringのみ」の場合
データフォーマットのデータ型が「String」の項目のみ、ダブルクォートで囲みます。
- 「なし」の場合
すべての項目をダブルクォートで囲みません。

コラム

「なし」の場合、ダブルクォートを使用しませんが、「カンマ」や「タブ」などの特殊な文字列を含む場合は、「"（ダブルクォート）」で囲むことがあります。

7. データフォーマット

- 列名：CSVに出力する項目名を記述します。
- データ型：取り込むCSVの項目に対応するデータ型を設定します。
設定できるデータ型は、string / number / date / boolean です。
- フォーマット：データ型が date の場合に、フォーマットを設定します。
- 追加・削除：項目の追加・削除ができます。

8. 登録

設定した内容をデータソース定義として登録します。

「データソース - 編集[CSVエクスポート]」画面の操作手順

「データソース - 編集[CSVエクスポート]」画面の操作手順について説明します。

1. エクスポートしたい設定で「文字コード」、「改行コード」、「区切り文字」、「見出し行」、「ダブルクォート」を入力します。

データソース - 編集[CSVエクスポート]

データソース種類: CSVエクスポート
データソース名: CSVエクスポートサンプル

CSVエクスポート 管理会社設定

文字コード*: SJIS
改行コード: CR+LF
区切り文字: カンマ
見出し行: あり なし
ダブルクォート: 全データ型 stringのみ なし

データフォーマット +追加

列名	データ型	フォーマット	削除
1 列1	string		-
2 列2	string		-
3 列3	string		-
4 列4	string		-
5 列5	string		-

登録

2. エクスポートしたいデータに合わせて、「データフォーマット」を入力し、「登録」ボタンをクリックします。

データソース - 編集[CSVエクスポート]

データソース種類: CSVエクスポート
データソース名: CSVエクスポートサンプル

CSVエクスポート 管理会社設定

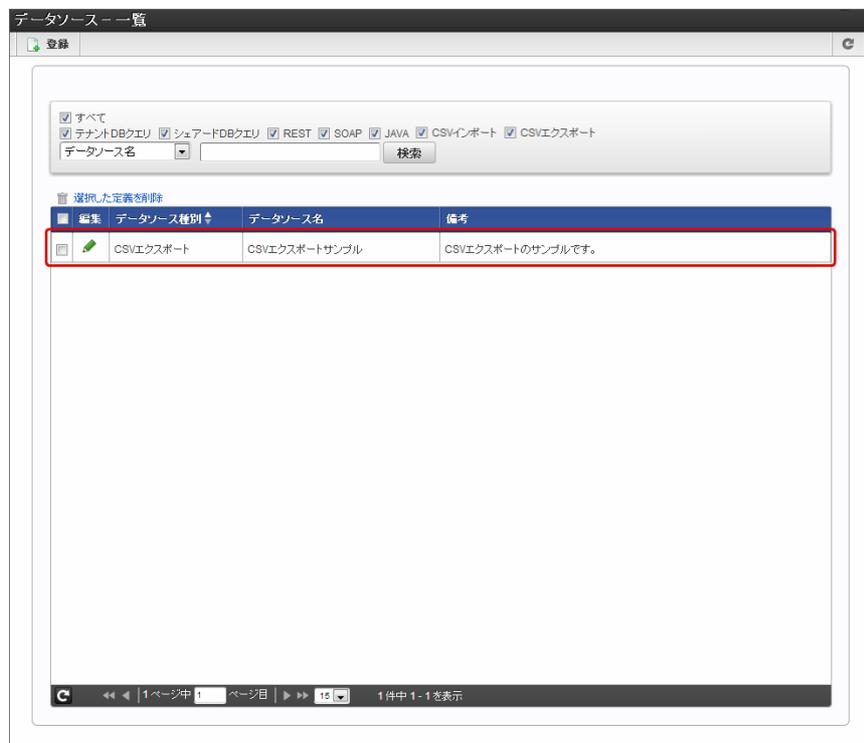
文字コード*: SJIS
改行コード: CR+LF
区切り文字: カンマ
見出し行: あり なし
ダブルクォート: 全データ型 stringのみ なし

データフォーマット +追加

列名	データ型	フォーマット	削除
1 備品名	string		-
2 メーカー	string		-
3 価格	number		-
4 会社	string		-

登録

3. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。



ルール

「データソース編集[ルール]」画面の機能と各部の説明

「データソース編集[ルール]」画面の内容は以下の通りです。

データソース 編集[ルール]

データソース種別: ルール
データソース名 (1): RuleSolverシナリオ用

ルール設定 | 管理会社設定

(2) サービスタイプ: RULE

(3) Decision名: DeterminellToBuyStock

(4) 実行モード: シーケンシャル 推論型

(5) 解のタイプ: 単一解 複数解

(6) 詳細設定:

(7) 解の返却範囲: 小

(8) コスト変数: 最大化

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... 開始 中断

(9)

Decisionファイル	ファイル名	ダウンロード	削除
1	DecisionRiskyStock.xls		

(10) Java Beanアーカイブリスト +追加

Jar名	削除
1 RiskyStock.jar	

(11) リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト
1	Stock	object		なし
2	inDebt	string		1
3	inFusion	string		1
4	inFusionWithStrong	string		1
5	hasGoodPrice	string		1
6	risky	string		1
7	buyShares	string		1

(12) レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト
1	Stock	object		なし
2	inDebt	string		1
3	inFusion	string		1
4	inFusionWithStrong	string		1
5	hasGoodPrice	string		1
6	risky	string		1
7	buyShares	string		1
8		object		なし
9	reportParam	string		8

更新

1. データソース名

「データソース-新規登録」画面で設定したデータソース名を表示します。

2. サービスタイプ

データソース種別として選択した「RULE」が表示されます。

3. Decision名

設定ファイルのDecision名を入力します。

4. 実行モード

「Rule」の実行モードを選択します。

「シーケンシャル」モードを選択すると「Rule Engine」を実行します。

「推論型」モードを選択すると「Rule Solver」を実行します。

5. 解のタイプ

「推論型」モード利用時に、返却する解の個数を設定できます。

■ 単一解

解を1つのみ返却します。この解のタイプの場合は、詳細設定が選択可能です。
OpenRulesからのレスポンスはオブジェクト、htmlで取得可能です。

■ 複数解

解を複数返却します。レスポンスはhtmlでのみ取得可能です。

6. 詳細設定

解のタイプが単一解の時に選択可能です。

選択することで、解の返却範囲の設定とコスト変数の設定が可能です。

7. 解の返却範囲

「推論型」モード利用時に、最大いくつの解を考慮するかの設定を行います。

各選択可能項目と、考慮する解の個数は以下の通りです。

選択可能項目	解の個数
小	10
中	25
大	50

コスト変数の設定を行う場合は、この解の返却範囲で設定された解の中から最適解を選択します。

8. コスト変数

最適解を返却させたい時に設定します。

1. 条件

最適解の条件を指定します。指定可能な条件は以下です。

- 最大化
コスト変数が最大となる解を最適解と判断します。
- 最少化
コスト変数が最少となる解を最適解と判断します。

2. 変数名

コスト変数として指定する変数名を記入します。

指定する変数は、ルール設定ファイル (.xls) 内の「Glossary」テーブルで設定した「Decision Variable」の中から指定してください。
ただし、指定可能な変数の型はint型のみです。

9. 設定ファイルのアップロード (decisionファイル設定)

設定ファイルをアップロードします。

- Decisionファイル
Decisionを含むExcelファイルのラジオボタンをオンにします。
- ファイル名
アップロードしたファイル名が表示されます。
- ダウンロード
アップロードしたファイルをローカルにダウンロードします。
- 削除
アップロードしたファイルをサーバのStorageから削除します。

10. Java Beanアーカイブリスト

「推論型」モードを実行する際に使用するクラスファイル名を設定します。

11. リクエスト

「シーケンシャルモード」を実行する際に手動でリクエストパラメータを設定します。

12. レスポンス

「シーケンシャルモード」を実行する際に手動でレスポンスフィールドを設定します。



注意

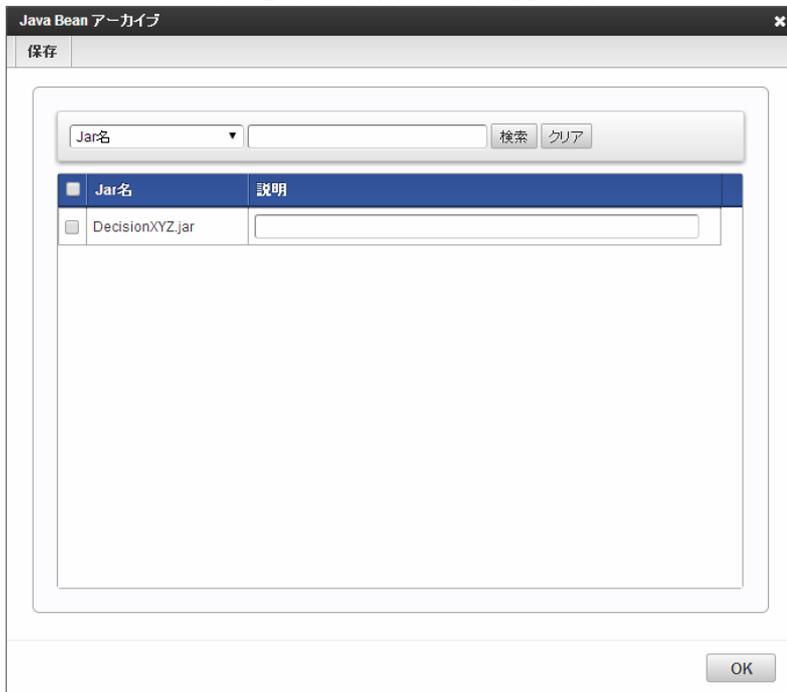
1つのデータソース定義に対して、同じ名称の設定ファイルを複数アップロードすることはできません。
同じ名称の設定ファイルをアップロードする場合には、一度登録済みの設定ファイルを削除してからアップロードしてください。

Java Beanアーカイブリストの機能と各部の説明

Java Beanアーカイブリストの設定は以下の通りです。

Java Beanアーカイブリスト

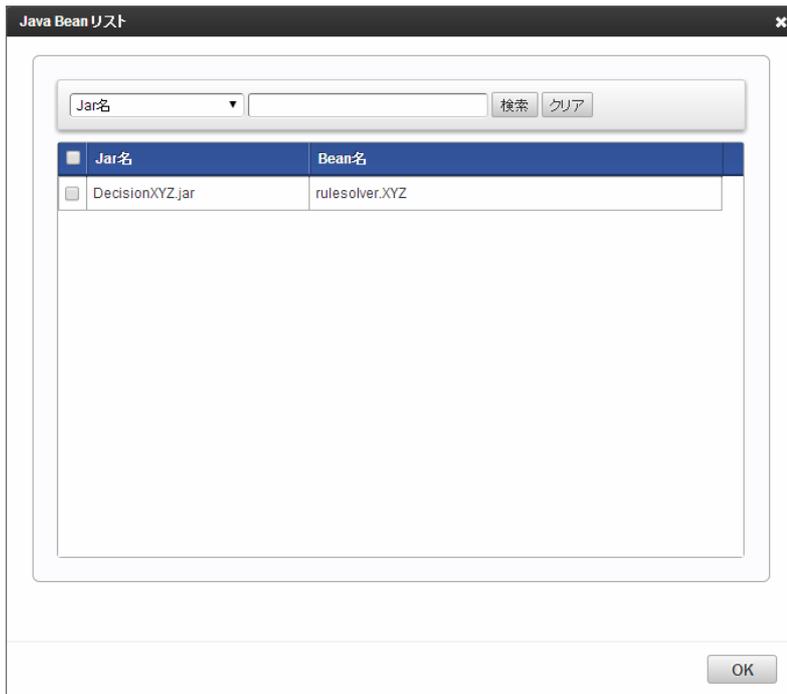
Java Beanアーカイブリストの設定の内容は以下の通りです。



「推論型」モード実行時に使用するjarファイルを選択します。

Java Beanリスト

Java Beanリストの設定の内容は以下の通りです。



Java Beanアーカイブリストで選択したjarファイル内使用するクラスを選択します。
使用するクラスを選択すると、自動でリクエストまたはレスポンスに値が設定されます。

リクエストパラメータ、レスポンスフィールドの機能と各部の説明

リクエストパラメータ、レスポンスフィールドの設定は以下の通りです。

リクエストの設定項目

リクエストの設定の内容は以下の通りです。

「実行モード」が「シーケンシャル」の場合、左の列番号をドラッグすることで、並び順を入れ替えることができます。

リクエスト (5) +追加					
	(1) パラメータ	(2) データ型	(3) フォーマット	(4) 親オブジェクト	(6) 削除
1	RequestObject	object		なし	-
2	String	string		1	-
3	Number	number		1	-
4	Date	date	yyyy/MM/dd	1	-
5	Boolean	boolean		1	-

1. パラメータ

リクエストパラメータ名を設定します。

外部連携設定（データマッパー）でリクエストとして設定したい値を、「Glossary」のBusiness ConceptとAttributeを参照して記述します。



注意

全角文字などの、マルチバイト文字を設定することはできません。

2. データ型

リクエストパラメータのデータ型を設定します。

設定できるデータ型は、string / number / date / boolean / object / array です。

3. フォーマット

型に「date」を設定した場合、日付の書式を設定します。

4. 親オブジェクト

リクエストオブジェクトが階層化されている場合に、上位階層に相当するパラメータの列番号を指定します。

親オブジェクトに指定可能なパラメータは、「型」が「object」、または「array」となっているものが対象です。

5. 追加

クリックするとリクエストパラメータの入力欄を1行追加します。

6. 削除

クリックするとリクエストパラメータを1行削除します。

レスポンスの設定項目

レスポンスの設定の内容は以下の通りです。

左の列番号をドラッグすることで、並び順を入れ替えることができます。

レスポンス (5) +追加					
	(1) フィールド	(2) データ型	(3) フォーマット	(4) 親オブジェクト	(6) 削除
1	ResponseObject	object		なし	-
2	Date	date	yyyy/MM/dd	1	-

1. パラメータ

レスポンスフィールド名を設定します。

外部連携設定（データマッパー）でレスポンスとして設定したい値を「Glossary」のBusiness ConceptとAttributeを参照して記述します。



注意

全角文字などの、マルチバイト文字を設定することはできません。

2. データ型

レスポンスフィールドのデータ型を設定します。

設定できるデータ型は、object / array / string / number / date / boolean です。

3. フォーマット

型に「date」を設定した場合、日付の書式を設定します。

4. 親オブジェクト

レスポンスオブジェクトが階層化されている場合に、上位階層に相当するフィールドの列番号を指定します。

親オブジェクトに指定可能なパラメータは、「型」が「object」、または「array」となっているものが対象です。

5. 追加

クリックするとレスポンスフィールドの入力欄を1行追加します。

6. 削除

クリックするとレスポンスフィールドを1行削除します。

i コラム

「推論型」モードを利用する場合はリクエスト・レスポンス用にJavaBeanを作成し、jarファイルとして配置する必要があります。デフォルトの配置先は以下です。

```
%RESIN_HOME% /webapps/ %コンテキストパス% /WEB-INF/im_bis/datasource/rule/solver
```

また、jarファイルの配置先は IM-BIS の設定ファイルによって変更することができます。設定ファイルについては「IM-BIS セットアップガイド」-「[パーチャルテナントに関する設定](#)」を参照してください。

「データソース - 編集[ルール]」画面の操作手順

「データソース - 編集[ルール]」画面の操作手順について説明します。

■ RuleEngineの場合

1. 設定ファイルに記述した「Decision名」を入力します。

データソース - 編集[ルール]

データソース種類: ルール
データソース名: RuleEngineサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンシャル 推論型

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... | 開始 | 中断

Decisionファイル	ファイル名	ダウンロード	削除
リクエスト +追加			
	パラメータ	データ型	フォーマット 親オブジェクト 削除
レスポンス +追加			
	フィールド	データ型	フォーマット 親オブジェクト 削除

登録

2. 「実行モード」が「シーケンシャル」となっていることを確認し、「ファイル追加...」をクリックし、ファイルを選択します。

データソース - 編集[ルール]

データソース種類: ルール
データソース名: RuleEngineサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンシャル 推論型

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... | 開始 | 中断

Decisionファイル	ファイル名	ダウンロード	削除
リクエスト +追加			
	パラメータ	データ型	フォーマット 親オブジェクト 削除
レスポンス +追加			
	フィールド	データ型	フォーマット 親オブジェクト 削除

登録

3. 対象のファイルが選択されていることを確認し、「Decisionファイル」をクリックします。

データソース 編集[ルール]

データソース種別: ルール
データソース名: RuleEngineサンプル

ルール設定 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンス型 推論型

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	Pattern1.xls		

リクエスト

	パラメータ	データ型	フォーマット	親オブジェクト	削除

レスポンス

	フィールド	データ型	フォーマット	親オブジェクト	削除

4. 「リクエスト」の「追加」をクリックします。

データソース 編集[ルール]

データソース種別: ルール
データソース名: RuleEngineサンプル

ルール設定 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンス型 推論型

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	Pattern1.xls		

リクエスト

	パラメータ	データ型	フォーマット	親オブジェクト	削除

レスポンス

	フィールド	データ型	フォーマット	親オブジェクト	削除

5. 「パラメータ」、「データ型」、「親オブジェクト」を設定します。
「データ型」を「Date」とする場合には、フォーマットも指定してください。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleEngineサンプル

ルール設定 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンシャル (選択)

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	Pattern1.xls		

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	<input type="text"/>	string	<input type="text"/>	なし	

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除
	<input type="text"/>				

登録

6. 4~5の手順を繰り返し、必要なリクエストパラメータを設定します。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleEngineサンプル

ルール設定 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンシャル (選択)

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	Pattern1.xls		

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	RequestObject	object	<input type="text"/>	なし	
2	requestString	string	<input type="text"/>	1	
3	requestNumber	number	<input type="text"/>	1	
4	requestDate	date	yyyy/MM/dd	1	
5	requestBoolean	boolean	<input type="text"/>	1	

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除
	<input type="text"/>				

登録

7. 「レスポンス」の「追加」をクリックします。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleEngineサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンシャル 推論型

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	Pattern1.xls		-

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	RequestObject	object		なし	-
2	requestString	string		1	-
3	requestNumber	number		1	-
4	requestDate	date	yyyy/MM/dd	1	-
5	requestBoolean	boolean		1	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除
1					

登録

8. 「パラメータ」、「データ型」、「親オブジェクト」を設定します。
「データ型」を「Date」とする場合には、フォーマットも指定してください。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleEngineサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンシャル 推論型

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	Pattern1.xls		-

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	RequestObject	object		なし	-
2	requestString	string		1	-
3	requestNumber	number		1	-
4	requestDate	date	yyyy/MM/dd	1	-
5	requestBoolean	boolean		1	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除
1		string		なし	-

登録

9. 7~8の手順を繰り返し、必要なレスポンスパラメータを設定します。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleEngineサンプル

ルール設定 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンシャル 推論型

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	Pattern1.xls		-

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	RequestObject	object		なし	-
2	requestString	string		1	-
3	requestNumber	number		1	-
4	requestDate	date	yyyy/MM/dd	1	-
5	requestBoolean	boolean		1	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除
1	ResponseObject	object		なし	-
2	responseString	string		1	-

登録

10. 「登録」ボタンをクリックします。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleEngineサンプル

ルール設定 管理会社設定

サービスタイプ: RULE
Decision名: ForIntegrationTest
実行モード: シーケンシャル 推論型

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	Pattern1.xls		-

リクエスト +追加

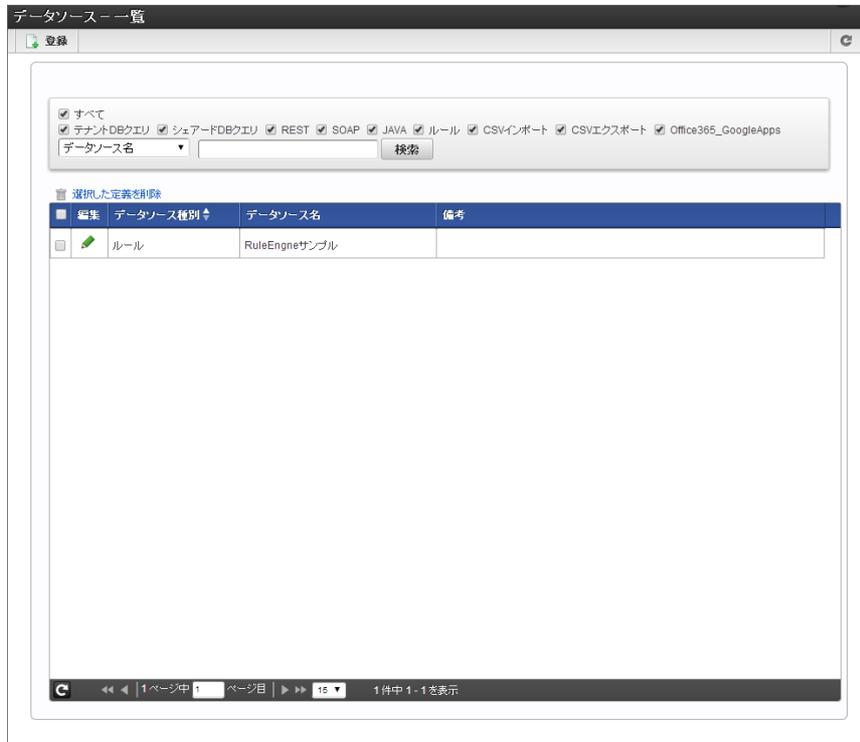
	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	RequestObject	object		なし	-
2	requestString	string		1	-
3	requestNumber	number		1	-
4	requestDate	date	yyyy/MM/dd	1	-
5	requestBoolean	boolean		1	-

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト	削除
1	ResponseObject	object		なし	-
2	responseString	string		なし	-

登録

11. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。



- RuleSolverの場合

1. 以下のフォルダに設定したいjarファイルを配置します。

```
%RESIN_HOME% /webapps/ %コンテキストパス% /WEB-INF/im_bis/datasource/rule/solver
```

2. 設定ファイルに記述した「Decision名」を入力します。



3. 「実行モード」の「推論型」をクリックします。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleSolverサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: FindXYZ
実行モード: シーケンシャル 推論型
解のタイプ: 単一解 複数解
詳細設定:

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... | 開始 | 中断

Decisionファイル	ファイル名	ダウンロード	削除
Java Beanアーカイブリスト +追加			
	Jar名		削除
リクエスト +追加			
	パラメータ	データ型	フォーマット 親オブジェクト
レスポンス +追加			
	フィールド	データ型	フォーマット 親オブジェクト

登録

4. 「ファイル追加...」をクリックし、ファイルを選択します。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleSolverサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: FindXYZ
実行モード: シーケンシャル 推論型
解のタイプ: 単一解 複数解
詳細設定:

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... | 開始 | 中断

Decisionファイル	ファイル名	ダウンロード	削除
Java Beanアーカイブリスト +追加			
	Jar名		削除
リクエスト +追加			
	パラメータ	データ型	フォーマット 親オブジェクト
レスポンス +追加			
	フィールド	データ型	フォーマット 親オブジェクト

登録

5. 対象のファイルが選択されていることを確認し、「Decisionファイル」をクリックします。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleSolverサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: FindXYZ
実行モード: シーケンシャル 推論型
解のタイプ: 単一解 複数解
詳細設定:

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... | 開始 | 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	FindXYZ.xls		-

Java Beanアーカイブリスト +追加

	Jar名	削除

リクエスト +追加

	パラメータ	データ型	フォーマット	親オブジェクト

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト

登録

6. 「Java Beanアーカイブリスト」の「追加」をクリックします。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleSolverサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: FindXYZ
実行モード: シーケンシャル 推論型
解のタイプ: 単一解 複数解
詳細設定:

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... | 開始 | 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	FindXYZ.xls		-

Java Beanアーカイブリスト +追加

	Jar名	削除

リクエスト +追加

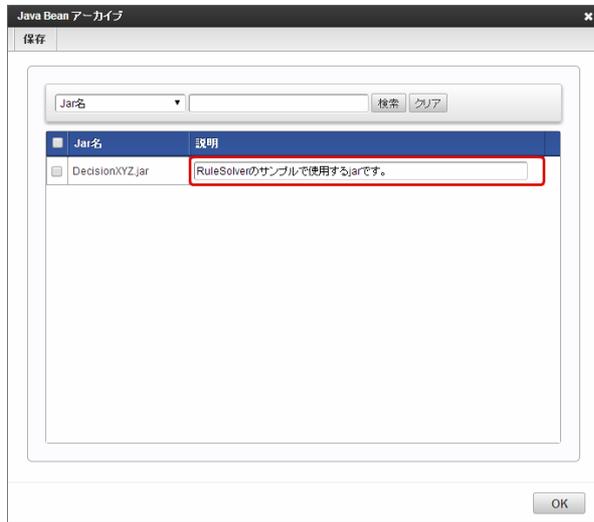
	パラメータ	データ型	フォーマット	親オブジェクト

レスポンス +追加

	フィールド	データ型	フォーマット	親オブジェクト

登録

7. 「Java Beanアーカイブリスト」画面に対象のjarファイルが表示されていることを確認し、説明を入力します。

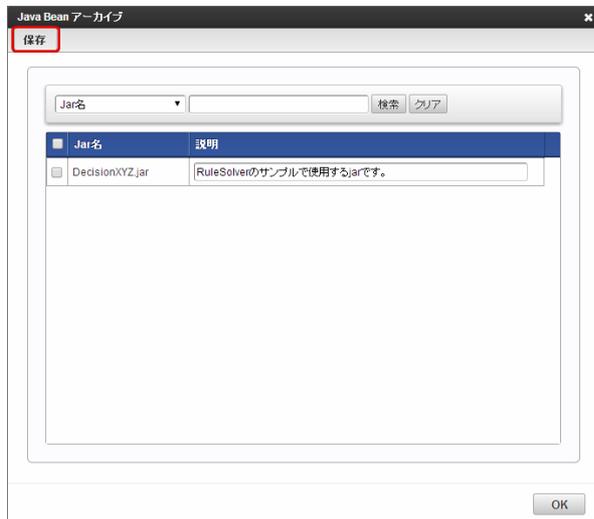


注意

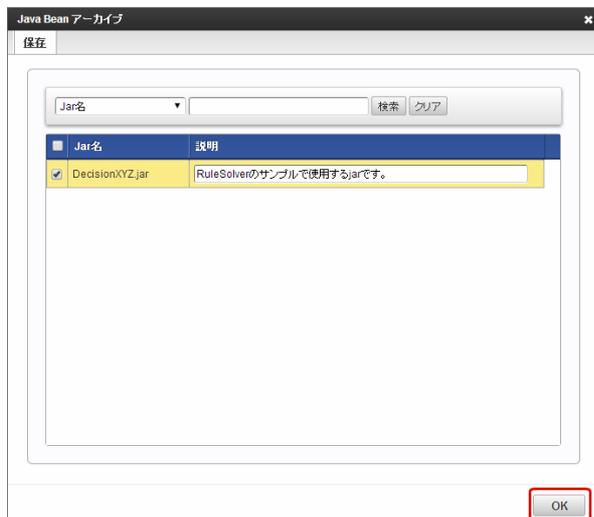
説明の入力は任意です。

説明を入力した場合は、保存ボタンをクリックしないと説明が保存されないため、注意してください。

8. 「保存」ボタンをクリックします。



9. 対象のjarファイルを選択し、「OK」ボタンをクリックします。



10. 「Java Beanアーカイブリスト」に対象のjarが表示されていることを確認し、「リクエスト」の「追加」をクリックします。

データソース 編集[ルール]

データソース種別: ルール
データソース名: RuleSolverサンプル

ルール設定 管理会社設定

サービスタイプ: RULE
Decision名: FindXYZ
実行モード: シーケンシャル (推論型)
解のタイプ: 単一解 (複数解)
詳細設定:

Excelファイルのアップロード (decisionファイル設定)

ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	FindXYZ.xls		-

Java Beanアーカイブリスト +追加

	Jar名	削除
1	DecisionXYZ.jar	-

リクエスト +追加

パラメータ	データ型	フォーマット	親オブジェクト

レスポンス +追加

フィールド	データ型	フォーマット	親オブジェクト

登録

11. 「Java Beanリスト」画面のjarファイルのリクエストに使用するJava Beanを選択し、OKをクリックします。

Java Bean リスト

Jar名: 検索 クリア

Jar名	Bean名
<input checked="" type="checkbox"/> DecisionXYZ.jar	rulesolver.XYZ

OK

12. 「リクエスト」の値を確認し、「レスポンス」の「追加」をクリックします。

データソース 編集[ルール]

データソース種別: ルール
データソース名: RuleSolverサンプル

ルール設定 管理会社設定

サービスタイプ: RULE
Decision名: FindXYZ
実行モード: シーケンシャル (推論型)
解のタイプ: 単一解 (複数解)
詳細設定:

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... 開始 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	FindXYZ.xls		-

Java Beanアーカイブリスト +追加

	Jar名	削除
1	DecisionXYZ2.jar	-

リクエスト - +追加

	パラメータ	データ型	フォーマット	親オブジェクト
1	XYZ	object		なし
2	x	number		1
3	y	number		1
4	z	number		1

レスポンス - +追加

	フィールド	データ型	フォーマット	親オブジェクト
--	-------	------	--------	---------

登録

13. 「Java Beanリスト」画面のjarファイルのレスポンスに使用するJava Beanを選択し、OKをクリックします。

Java Bean リスト

Jar名: 検索 クリア

Jar名	Bean名
DecisionXYZ.jar	rulesolver.XYZ

14. 「レスポンス」の値を確認し、「登録」ボタンをクリックします。

データソース - 編集[ルール]

データソース種別: ルール
データソース名: RuleSolverサンプル

ルール設定 | 管理会社設定

サービスタイプ: RULE
Decision名: FindXYZ
実行モード: シーケンシャル (選択) / 推論型
解のタイプ: 単一解 (選択) / 複数解
詳細設定:

Excelファイルのアップロード (decisionファイル設定)

+ ファイル追加... | 開始 | 中断

	Decisionファイル	ファイル名	ダウンロード	削除
1	<input checked="" type="radio"/>	FindXYZ.xls		-

Java Beanアーカイブリスト +追加

	Jar名	削除
1	DecisionXYZ2.jar	-

リクエスト - +追加

	パラメータ	データ型	フォーマット	観オブジェクト
1	XYZ	object		なし
2	x	number		1
3	y	number		1
4	z	number		1

レスポンス - +追加

	フィールド	データ型	フォーマット	観オブジェクト
1	XYZ	object		なし
2	x	number		1
3	y	number		1
4	z	number		1
5		object		なし
6	reportParam	string		5

登録

15. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。

データソース - 一覧

登録

すべて
 テナントDBクエリ シェアードBクエリ REST SOAP JAVA ルール CSVインポート CSVエクスポート Office365_GoogleApps

データソース名: 検索

選択した定義を削除

編集	データソース種別	データソース名	備考
<input type="checkbox"/>	ルール	RuleSolverサンプル	

1ページ中 1 ページ目 15 1件中 1-1 を表示

テナントDB更新系クエリ

「データソース - 編集[テナントDB更新系クエリ]」画面の機能と各部の説明

「データソース - 編集[テナントDB更新系クエリ]」画面の内容は以下の通りです。

データソース編集[テナントDB更新系クエリ]

データソース種別 テナントDB更新系クエリ
データソース名 (1) テナントDB-DMLサンプル

クエリ設定 管理会社設定

(2) SQL
INSERT INTO sample_table (column1 , column2 , column3) VALUES (? , ? , ?)

(3) 入力値 + 追加

	テスト実行値	データ型	論理名	削除
IN 1		VARCHAR	日本語 更新項目 1-1 英語 Update item1-1 中国語 (中華人民共和国) 更新項目 1-1	-
IN 2		VARCHAR	日本語 更新項目 1-2 英語 Update item1-2 中国語 (中華人民共和国) 更新項目 1-2	-
IN 3		VARCHAR	日本語 更新項目 1-3 英語 Update item1-3 中国語 (中華人民共和国) 更新項目 1-3	-

(4) 出力値

	カラム名	データ型	論理名	削除
OUT 1	count	NUMBER	日本語 処理件数 英語 Number 中国語 (中華人民共和国) 交易数量	

(5) テスト実行 (6) 登録

1. データソース名

「データソース - 新規登録」画面で設定したデータソース名を表示します。

2. SQL

データソースとして実行するSQLを入力します。

記載するSQLは、INSERT、UPDATE、DELETEが利用できます。

3. 入力値

- テスト実行値：
INSERT、UPDATEの場合には、データを登録・更新する対象の列と、条件（WHERE句）に代入する値を入力します。
- データ型：項目に対応するデータの形式を文字型(VARCHAR)、数値型(NUMBER)、日付型(DATE)、タイムスタンプ型(TIMESTAMP)の中から選択します。
- 論理名：IM-FormaDesigner の画面アイテムでデータソースを利用する際の「パラメータ設定」で表示する項目名として利用されます。20文字まで設定することができます。
- 追加・削除：入力値の追加・削除ができます。

注意

テナントDB更新系クエリ・シェアードDB更新系クエリでのテスト実行について

テナントDB更新系クエリ・シェアードDB更新系クエリでは、テスト実行を行うと、実際にテーブルへのデータの登録・更新・削除が行われます。

データソース定義の設定画面でテスト実行を行う場合には、実際に登録・更新・削除となるデータを入力するか、またはテスト実行後にTableMaintenanceやデータベース管理ツールなどを用いて不要なデータの削除などを行うようにしてください。

i コラム**DATE型・TIMESTAMP型のテスト実行値について**

日付を扱うデータ型の書式設定は、以下の通りです。

データ型	書式（デフォルト）	設定
DATE	“yyyy/MM/dd”	IM_DATETIME_FORMAT_TIME_INPUT
TIMESTAMP	“yyyy/MM/dd HH:mm”	IM_DATETIME_FORMAT_DATE_INPUT IM_DATETIME_FORMAT_TIME_INPUT（※）

※ TIMESTAMPの設定は、2つの項目の設定値を半角スペースで結合した値で設定されます。

書式を変更する場合には、以下の設定ファイルを変更してください。

conf/date-time-format-config/im-date-time-format-config.xml

i コラム**更新系クエリにおけるトランザクションの単位**

更新系クエリ（テナントDB更新系クエリ、シェアードDB更新系クエリ）において、トランザクションは実行のタイミングに基づいて、以下の通り設定しています。

- タスク（ノード）の前処理
データソース定義の単位で別のトランザクションを設定します。
- アクション設定（画面でのイベント）
データソース定義の単位で別のトランザクションを設定します。
- タスク（ノード）の後処理
すべてのデータソース定義で1つのトランザクションを設定します。
（シェアードDB更新系クエリを含む場合、XAではありませんが、各アプリケーションサーバの分散トランザクションによって管理されます。）
- 案件終了処理
データソース定義の単位で別のトランザクションを設定します。

4. 出力値

テナントDB更新系クエリでは、固定でデータ処理件数が設定されます。
データマッパーを利用して、アイテムの値に処理件数を反映することもできます。

5. テスト実行

設定した情報に基づいて、データの登録・更新・削除を行います。
実行後のデータ処理については、データベースに即座に反映されます。
データの登録・更新・削除ができなかった場合には、エラーメッセージを表示します。

6. 登録

設定した内容をデータソース定義として登録します。

「データソース - 編集[テナントDB更新系クエリ]」画面の操作手順

「データソース - 編集[テナントDB更新系クエリ]」画面の操作手順について説明します。

1. 「SQL」に実行するSQLを入力します。

データベース編集[テナントDB更新系クエリ]

データベース種別: テナントDB更新系クエリ
データベース名: 【サンプル】更新クエリ

クエリ設定 管理会社設定

SQL

```
UPDATE sample_fruit
SET fruit_name = ?
WHERE fruit_id = ?
```

入力値

	テスト実行値	データ型	論理名	削除	
IN 1		VARCHAR	日本語	更新項目 1-1	-
			英語	Update item1-1	
			中国語 (中華人民共和国)	更新項目 1-1	
IN 2		VARCHAR	日本語	更新項目 1-2	-
			英語	Update item1-2	
			中国語 (中華人民共和国)	更新項目 1-2	
IN 3		VARCHAR	日本語	更新項目 1-3	-
			英語	Update item1-3	
			中国語 (中華人民共和国)	更新項目 1-3	

2. 「SQL」の列名・条件で"?"（プレースホルダ）を利用している場合には、"?"に合わせて「入力値」を設定します。
 なお、列名と条件の両方に"?"を利用している場合には、SQLに登場する順番に設定してください。

データベース編集[テナントDB更新系クエリ]

データベース種別: テナントDB更新系クエリ
データベース名: 【サンプル】更新クエリ

クエリ設定 管理会社設定

SQL

```
UPDATE sample_fruit
SET fruit_name = ?
WHERE fruit_id = ?
```

入力値

	テスト実行値	データ型	論理名	削除	
IN 1	レモン	VARCHAR	日本語	名前	-
			英語	名前	
			中国語 (中華人民共和国)	名前	
IN 2	lemon	VARCHAR	日本語	コード	-
			英語	コード	
			中国語 (中華人民共和国)	コード	

3. 「テスト実行値」には、実際にデータの登録・変更・削除に利用する 値を入力し、「テスト実行」をクリックします。

データソース編集[テナントDB更新系クエリ]

データソース種別: テナントDB更新系クエリ
 データソース名: 【サンプル】更新クエリ

クエリ設定 | 管理会社設定

SQL: UPDATE sample_fruit SET fruit_name = ? WHERE fruit_id = ?

入力値

テスト実行値	データ型	論理名	削除
IN 1 レモン	VARCHAR	日本語 名前 英語 名前 中国語 (中華人民共和国) 名前	-
IN 2 lemon	VARCHAR	日本語 コード 英語 コード 中国語 (中華人民共和国) コード	-

出力値

カラム名	データ型	論理名	削除
OUT 1 count	NUMBER	日本語 処理件数 英語 Number 中国語 (中華人民共和国) 交易数量	

テスト実行 | 登録

4. テストが成功した場合、以下のメッセージが表示されます。

データソース編集[テナントDB更新系クエリ]

データソース種別: テナントDB更新系クエリ
 データソース名: 【サンプル】更新クエリ

クエリ設定 | 管理会社設定

SQL: UPDATE sample_fruit SET fruit_name = ? WHERE fruit_id = ?

テスト実行に成功しました。取得件数:1

入力値

テスト実行値	データ型	論理名	削除
IN 1 レモン	VARCHAR	日本語 名前 英語 名前 中国語 (中華人民共和国) 名前	-
IN 2 lemon	VARCHAR	日本語 コード 英語 コード 中国語 (中華人民共和国) コード	-

出力値

カラム名	データ型	論理名	削除
OUT 1 count	NUMBER	日本語 処理件数 英語 Number 中国語 (中華人民共和国) 交易数量	

テスト実行 | 更新

5. 内容を確認して、「登録」をクリックします。

データソース - 編集[テナントDB更新系クエリ]

データソース種別: テナントDB更新系クエリ
 データソース名: 【サンプル】更新クエリ

クエリ設定 | 管理会社設定

```
SQL
UPDATE sample_fruit
SET fruit_name = ?
WHERE fruit_id = ?
```

入力値

テスト実行値	データ型	論理名	削除
IN 1	レモン	VARCHAR	
		日本語	名前
		英語	名前
		中国語 (中華人民共和国)	名前
IN 2	lemon	VARCHAR	
		日本語	コード
		英語	コード
		中国語 (中華人民共和国)	コード

出力値

カラム名	データ型	論理名	削除
OUT 1	count	NUMBER	
		日本語	処理件数
		英語	Number
		中国語 (中華人民共和国)	交易数量

テスト実行 | **登録**

6. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。

データソース - 一覧

登録

すべて
 テナントDBクエリ シェアードDBクエリ REST SOAP JAVA CSVインポート CSVエクスポート Office365_GoogleApps テナントDB更新系クエリ シェアードDB更新系クエリ

データソース名: 検索

選択した定義を削除

編集	データソース種別	データソース名	備考
<input checked="" type="checkbox"/>	テナントDB更新系クエリ	【サンプル】更新クエリ	

1ページ中 1 ページ目 10 1件中 1-1 を表示

シェアードDB更新系クエリ

「データソース - 編集[シェアードDB更新系クエリ]」画面の機能と各部の説明

「データソース - 編集[シェアードDB更新系クエリ]」画面の内容は以下の通りです。

データソース 編集[シェアードDB更新系クエリ]

データソース種別 シェアードDB更新系クエリ
データソース名 (1) シェアードDB-DMLサンプル

クエリ設定 管理会社設定

(2) 接続ID []
SQL INSERT INTO sample_table (column1 , column2 , column3) VALUES (? , ? , ?)

(3) 入力値 [] + 追加

	テスト実行値	データ型	論理名	削除	
IN 1	[]	VARCHAR	日本語	更新項目 1-1	-
			英語	Update Item1-1	
			中国語 (中華人民共和国)	更新項目 1-1	
IN 2	[]	VARCHAR	日本語	更新項目 1-2	-
			英語	Update Item1-2	
			中国語 (中華人民共和国)	更新項目 1-2	
IN 3	[]	VARCHAR	日本語	更新項目 1-3	-
			英語	Update Item1-3	
			中国語 (中華人民共和国)	更新項目 1-3	

(4) 出力値 []

	カラム名	データ型	論理名	削除	
OUT 1	count	NUMBER	日本語	処理件数	
			英語	Number	
			中国語 (中華人民共和国)	交易数量	

(5) テスト実行 (6) 登録

1. データソース名

「データソース - 新規登録」画面で設定したデータソース名を表示します。

2. SQL

データソースとして実行するSQLを入力します。

記載するSQLは、INSERT、UPDATE、DELETEが利用できます。

3. 入力値

- テスト実行値：
INSERT、UPDATEの場合には、データを登録・更新する対象の列と、条件（WHERE句）に代入する値を入力します。
- データ型：項目に対応するデータの形式を文字型(VARCHAR)、数値型(NUMBER)、日付型(DATE)、タイムスタンプ型(TIMESTAMP)の中から選択します。
- 論理名：IM-FormaDesignerの画面アイテムでデータソースを利用する際の「パラメータ設定」で表示する項目名として利用されます。20文字まで設定することができます。
- 追加・削除：入力値の追加・削除ができます。



注意

テナントDB更新系クエリ・シェアードDB更新系クエリでのテスト実行について

テナントDB更新系クエリ・シェアードDB更新系クエリでは、テスト実行を行うと、実際にテーブルへのデータの登録・更新・削除が行われます。

データソース定義の設定画面でテスト実行を行う場合には、実際に登録・更新・削除となるデータを入力するか、またはテスト実行後にTableMaintenanceやデータベース管理ツールなどを用いて不要なデータの削除などを行うようにしてください。

i コラム**DATE型・TIMESTAMP型のテスト実行値について**

日付を扱うデータ型の書式設定は、以下の通りです。

データ型	書式 (デフォルト)	設定
DATE	“yyyy/MM/dd”	IM_DATETIME_FORMAT_TIME_INPUT
TIMESTAMP	“yyyy/MM/dd HH:mm”	IM_DATETIME_FORMAT_DATE_INPUT IM_DATETIME_FORMAT_TIME_INPUT (*)

※ TIMESTAMPの設定は、2つの項目の設定値を半角スペースで結合した値で設定されます。

書式を変更する場合には、以下の設定ファイルを変更してください。

conf/date-time-format-config/im-date-time-format-config.xml

i コラム**更新系クエリにおけるトランザクションの単位**

更新系クエリ（テナントDB更新系クエリ、シェアードDB更新系クエリ）において、トランザクションは実行のタイミングに基づいて、以下の通り設定しています。

- タスク（ノード）の前処理
データソース定義の単位で別のトランザクションを設定します。
- アクション設定（画面でのイベント）
データソース定義の単位で別のトランザクションを設定します。
- タスク（ノード）の後処理
すべてのデータソース定義で1つのトランザクションを設定します。
（シェアードDB更新系クエリを含む場合、XAではありませんが、各アプリケーションサーバの分散トランザクションによって管理されます。）
- 案件終了処理
データソース定義の単位で別のトランザクションを設定します。

4. 出力値

シェアードDB更新系クエリでは、固定でデータ処理件数が設定されます。
データマッパーを利用して、アイテムの値に処理件数を反映することもできます。

5. テスト実行

設定した情報に基づいて、データの登録・更新・削除を行います。
実行後のデータ処理については、データベースに即座に反映されます。
データの登録・更新・削除ができなかった場合には、エラーメッセージを表示します。

6. 登録

設定した内容をデータソース定義として登録します。

「データソース - 編集[シェアードDB更新系クエリ]」画面の操作手順

「データソース - 編集[シェアードDB更新系クエリ]」画面の操作手順について説明します。

1. 接続IDを選択し、「SQL」に実行するSQLを入力します。

データベース編集[シェアードDB更新系クエリ]

データベース種別 シェアードDB更新系クエリ
データベース名 【サンプル】登録クエリ

クエリ設定 管理会社設定

接続ID sample

SQL
INSERT INTO sample_fruit (fruit_id , fruit_name , price) VALUES (? , ? , ?)

入力値 + 追加

	テスト実行値	データ型	論理名	削除	
IN 1		VARCHAR	日本語	更新項目 1-1	-
			英語	Update item1-1	
			中国語 (中華人民共和国)	更新項目 1-1	
IN 2		VARCHAR	日本語	更新項目 1-2	-
			英語	Update item1-2	
			中国語 (中華人民共和国)	更新項目 1-2	
IN 3		VARCHAR	日本語	更新項目 1-3	-
			英語	Update item1-3	
			中国語 (中華人民共和国)	更新項目 1-3	

2. 「SQL」の列名・条件で"?"（プレースホルダ）を利用している場合には、"?"に合わせて「入力値」を設定します。
 なお、列名と条件の両方に"?"を利用している場合には、SQLに登場する順番に設定してください。

データベース編集[シェアードDB更新系クエリ]

データベース種別 シェアードDB更新系クエリ
データベース名 【サンプル】登録クエリ

クエリ設定 管理会社設定

接続ID sample

SQL
INSERT INTO sample_fruit (fruit_id , fruit_name , price) VALUES (? , ? , ?)

入力値 + 追加

	テスト実行値	データ型	論理名	削除	
IN 1		VARCHAR	日本語	コード	-
			英語	コード	
			中国語 (中華人民共和国)	コード	
IN 2		VARCHAR	日本語	名前	-
			英語	名前	
			中国語 (中華人民共和国)	名前	
IN 3		VARCHAR	日本語	単価	-
			英語	単価	
			中国語 (中華人民共和国)	単価	

3. 「テスト実行値」には、実際にデータの登録・変更・削除に利用する値を入力し、「テスト実行」をクリックします。

データベース編集[シェアードDB更新系クエリ]

データベース種別: シェアードDB更新系クエリ
データベース名: 【サンプル】登録クエリ

クエリ設定 | 管理会社設定

接続ID: sample

SQL: INSERT INTO sample_fruit (fruit_id , fruit_name , price) VALUES (? , ? , ?)

入力値

テスト実行値	データ型	論理名	削除
IN 1 melon	VARCHAR	日本語: コード 英語: コード 中国語 (中華人民共和国): コード	-
IN 2 >りん	VARCHAR	日本語: 名前 英語: 名前 中国語 (中華人民共和国): 名前	-
IN 3 500	NUMBER	日本語: 単価 英語: 単価 中国語 (中華人民共和国): 単価	-

出力値

カラム名	データ型	論理名	削除
OUT 1 count	NUMBER	日本語: 処理件数 英語: Number 中国語 (中華人民共和国): 交易数量	

テスト実行 | 登録

4. テストが成功した場合、以下のメッセージが表示されます。

データベース編集[シェアードDB更新系クエリ]

データベース種別: シェアードDB更新系クエリ
データベース名: 【サンプル】登録クエリ

クエリ設定 | 管理会社設定

接続ID: sample

SQL: INSERT INTO sample_fruit (fruit_id , fruit_name , price) VALUES (? , ? , ?)

入力値

テスト実行値	データ型	論理名	削除
IN 1 melon	VARCHAR	日本語: コード 英語: コード 中国語 (中華人民共和国): コード	-
IN 2 >りん	VARCHAR	日本語: 名前 英語: 名前 中国語 (中華人民共和国): 名前	-
IN 3 500	NUMBER	日本語: 単価 英語: 単価 中国語 (中華人民共和国): 単価	-

出力値

カラム名	データ型	論理名	削除
OUT 1 count	NUMBER	日本語: 処理件数 英語: Number 中国語 (中華人民共和国): 交易数量	

テスト実行 | 登録

テスト実行に成功しました。取得件数:1

5. 内容を確認して、「登録」をクリックします。

データソース編集[シェアードDB更新系クエリ]

データソース種別 シェアードDB更新系クエリ
データソース名 【サンプル】登録クエリ

クエリ設定 管理会社設定

接続ID sample

SQL
INSERT INTO sample_fruit (fruit_id , fruit_name , price) VALUES (? , ? , ?)

入力値 + 追加

	テスト実行値	データ型	論理名	削除	
IN 1	melon	VARCHAR	日本語 英語 中国語 (中華人民共和国)	コード コード コード	-
IN 2	メロン	VARCHAR	日本語 英語 中国語 (中華人民共和国)	名前 名前 名前	-
IN 3	500	NUMBER	日本語 英語 中国語 (中華人民共和国)	単価 単価 単価	-

出力値

	カラム名	データ型	論理名	削除	
OUT 1	count	NUMBER	日本語 英語 中国語 (中華人民共和国)	処理件数 Number 交易数量	

テスト実行 登録

6. 正常に登録できると、次のように「データソース - 一覧」画面に追加されます。

データソース一覧

登録

すべて
 テナントDBクエリ
 シェアードDBクエリ
 REST
 SOAP
 JAVA
 CSVインポート
 CSVエクスポート
 Office365_GoogleApps
 テナントDB更新系クエリ
 シェアードDB更新系クエリ

データソース名 検索

選択した定義を削除

編集	データソース種別	データソース名	備考
<input type="checkbox"/>	シェアードDB更新系クエリ	【サンプル】登録クエリ	

1ページ中 1 ページ目 15 1件中 1-1 を表示

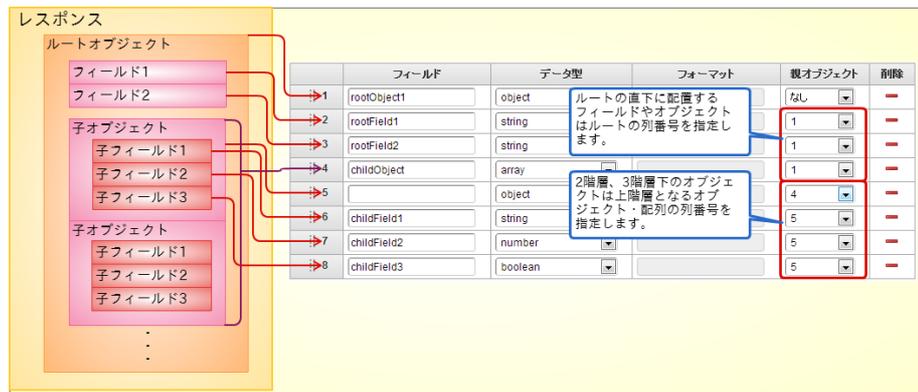
コラム

IM-BIS 2015 Winterより、データソース種別「LogicDesigner」を追加しました。
 データソース種別「LogicDesigner」は、データソース種別「REST」「SOAP」「JAVA」の機能を補完したもので、今後新規機能の追加は「LogicDesigner」に対してのみ実施されます。
 そのため、「REST」「SOAP」「JAVA」の利用については非推奨とし、「LogicDesigner」の利用を推奨しています。

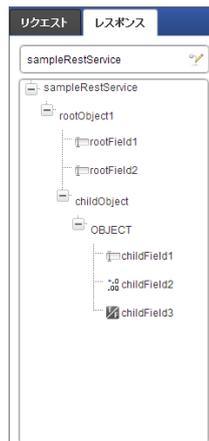
階層化したリクエストパラメータ、レスポンスフィールドの設定方法

呼び出す対象のパラメータ・フィールドの設定が、階層構造となる場合には、呼び出し先で指定した階層に合わせて設定します。

階層構造となるパラメータ・フィールドの定義と、データソース定義の設定例は、以下の図の通りです。



上記の図のように設定すると、データマッパーでは、ツリー形式で階層化して表示します。



コラム

レスポンスで名称を入力せずに設定したパラメータ・フィールドはデータマッパー上で以下のように表示します。

- データ型が「object」の場合、「OBJECT」と表示します。
- データ型が「array」の場合、「ARRAY」と表示します。
- データ型が「string」「number」「date」「boolean」のいずれかの場合、「VALUE」と表示します。

BIS業務管理者が、各データソース定義を利用するための設定をする

登録したデータソース定義をBIS業務管理者が利用するためには、登録後に以下のリンク先の手順に従って、管理グループの管理対象に追加してください。

[管理グループに「管理対象」を設定する](#)

リクエストパラメータへのシステムパラメータ設定方法

外部連携でシステムパラメータを利用する場合には、リクエストのパラメータ・フィールドに特定のシステムパラメータ名を設定してください。

パラメータ	データ型	親オブジェクト
1 message	string	なし
2 imfr_application_id	string	なし
3 imfr_application_no	string	なし
4 imwFlowId	string	なし
5 imwApplyBaseDate	string	なし

i コラム

外部連携で利用できるシステムパラメータについては、「IM-BIS 仕様書」-「暗黙的に連携するリクエストパラメータの仕様」を参照してください。

i コラム

以下のデータソース種別でシステムパラメータを利用する場合、入力値の論理名・フィールドに表示されている全てのロケールに特定のシステムパラメータ名を設定してください。

- テナントDBクエリ
- シェアードDBクエリ
- テナントDB更新系クエリ
- シェアードDB更新系クエリ

i コラム

以下のデータソース種別でシステムパラメータを利用する場合、データフォーマットの列名・フィールドに特定のシステムパラメータ名を設定してください。

- CSVエクスポート

レスポンスパラメータへの処理結果メッセージ設定方法

外部連携で処理結果を実行画面に表示する場合には、レスポンスのパラメータ・フィールドに処理結果メッセージを設定してください。

レスポンス			
	パラメータ	データ型	親オブジェクト
1	imfSoaResult	object	なし
2	imfErrorFlag	boolean	1
3	imfMessage	array	1
4	string	string	3
5	imfMultiMessages	array	1
6		object	5
7	imfMulti_ja	string	6
8	imfMulti_en	string	6
9	imfMulti_zh_CN	string	6
10	imfErrorItems	array	1
11		object	10
12	imfInputId	array	11
13	string	string	12
14	imfErrorMessage	string	11
15	imfErrorMultiMessages	object	11
16	imfErrorMulti_ja	string	15
17	imfErrorMulti_en	string	15
18	imfErrorMulti_zh_CN	string	15
19	imfIndex	double	11

i コラム

処理結果メッセージについては、「IM-BIS 仕様書」-「暗黙的に連携するレスポンスパラメータの仕様」を参照してください。

インポート・エクスポートを行う

ここでは、インポート・エクスポートの実行方法について説明します。

Contents

- インポート・エクスポートとは
- インポート・エクスポートの前提条件
- エクスポートの事前準備
- エクスポートを実行する
 - IM-Workflow に関する定義ファイルをエクスポートする
 - アプリケーション情報をエクスポートする
 - データソース定義をエクスポートする
 - テンプレートカテゴリをエクスポートする
 - IM-BIS の定義をエクスポートする
- インポートを実行する
 - IM-Workflow に関する定義ファイルをインポートする
 - アプリケーション情報をインポートする
 - データソース定義をインポートする
 - テンプレートカテゴリの定義をインポートする
 - IM-BIS の定義をインポートする

インポート・エクスポートとは

インポート・エクスポートは、他の環境のIM-BISへ定義を移すための機能です。

インポート・エクスポートの対象は、「IM-Workflow に関する定義ファイル」や「アプリケーション情報」、「データソース定義」、「BIS定義」です。

インポート・エクスポートの前提条件

- IM-BIS で作成したBISフロー／ワークフローで使用している定義をインポート・エクスポートを実行するためには IM-Workflow、IM-FormaDesigner のインポート・エクスポート機能も利用するため、実行ユーザに必要な権限を設定してください。
- BIS定義のインポート・エクスポート機能は、ローカルからの入出力のみ対応しています。

エクスポートの事前準備

IM-BIS でのエクスポート作業を行う前に事前準備として、対象の確認を行います。

IM-BIS でBISフロー／ワークフローを作成を行うと以下の情報が自動で作成されます。そのため、対象名を事前に確認する必要があります。

- コンテンツ定義
必ず作成され、コンテンツ名はBIS名が設定されます。
- ルート定義
必ず作成され、ルート名はBIS名が設定されます。
- フロー定義
必ず作成され、フロー名はBIS名が設定されます。
- 案件プロパティ定義
BISフロー／ワークフローでルール定義を利用した分岐や、追記設定などで案件プロパティを利用した場合に作成されます。
「IM-BIS - ルール定義」画面で、キー名となる画面項目の名称を確認します。
- ルール定義
BISフロー／ワークフローでルール定義を利用した分岐を含む場合に作成されます。
「IM-BIS - フロー編集」画面の分岐条件を設定する「分岐開始」ノードで、ルール名を確認します。



コラム

以下の情報は IM-BIS では定義されません。
IM-Workflow を利用して定義した場合はインポート/エクスポートをしてください。
詳細は、別紙「[IM-Workflow 管理者操作ガイド](#)」を参照してください。

- IMBox定義
- メール定義

コラム

「管理グループ」のインポート・エクスポートを行う場合には、IM-Workflow、TableMaintenance のインポート・エクスポートをご利用ください。

- IM-Workflow では、インポート・エクスポートで「管理グループ」に IM-BIS で使用している管理グループを含めてください。
- TableMaintenance では、以下のテーブルを対象にインポート・エクスポートを実行してください。
 - imbis_m_administration_group
 - imbis_m_administration_plugin
 - imbis_m_administration_target
 - imbis_m_admin_relation

「一覧表示パターン」、「採番ルール定義」のインポート・エクスポートを行う場合には、TableMaintenance のインポート・エクスポートをご利用ください。

- 「一覧表示パターン」については、以下のテーブルを対象にインポート・エクスポートを実行してください。
 - imbis_m_list_pattern
 - imbis_m_selected_column_list
 - imbis_m_column
- 「採番ルール定義」については、以下のテーブルを対象にインポート・エクスポートを実行してください。
 - imfr_m_auto_no
 - imfr_m_auto_no_company
 - imfr_m_auto_no_locale

上記の「imfr_m_auto_no」、「imfr_m_auto_no_company」は、採番ルール定義を論理削除しているため、テーブルに削除済みのデータが残っています。
不要なデータを移行ファイルから削除するためには、各テーブルに対応したCSVファイルから以下の条件でレコードを削除してください。

- 対象テーブル.delete_flag = 1

エクスポートを実行する

IM-BIS の定義をエクスポートするには、以下の手順で行います。

IM-Workflow に関する定義ファイルをエクスポートする

IM-Workflow のエクスポート機能を利用して、IM-BIS で作成したBISフロー／ワークフローに必要な IM-Workflow の定義をエクスポートします。

1. 「サイトマップ」→「IM-BIS」→「エクスポート」→「IM-BIS」→「IM-Workflow エクスポート」をクリックします。
2. エクスポートするマスタ定義の「検索」をクリックします。
(全てエクスポートしたい場合、「エクスポート (全件)」のチェックをオンにしてください。)

エクスポート				
コンテンツ定義	<input type="checkbox"/> エクスポート(全件) <input checked="" type="checkbox"/> 検索 <table border="1"> <tr> <td>コンテンツID</td> <td>コンテンツ名</td> <td>クリア</td> </tr> </table>	コンテンツID	コンテンツ名	クリア
コンテンツID	コンテンツ名	クリア		
ルート定義	<input type="checkbox"/> エクスポート(全件) <input type="checkbox"/> 検索 <table border="1"> <tr> <td>ルートID</td> <td>ルート名</td> <td>クリア</td> </tr> </table>	ルートID	ルート名	クリア
ルートID	ルート名	クリア		
フロー定義	<input type="checkbox"/> エクスポート(全件) <input type="checkbox"/> 検索 <table border="1"> <tr> <td>フローID</td> <td>フロー名</td> <td>クリア</td> </tr> </table>	フローID	フロー名	クリア
フローID	フロー名	クリア		
案件プロパティ定義	<input type="checkbox"/> エクスポート(全件) <input type="checkbox"/> 検索 <table border="1"> <tr> <td>キー</td> <td>キー名</td> <td>クリア</td> </tr> </table>	キー	キー名	クリア
キー	キー名	クリア		
ルール定義	<input type="checkbox"/> エクスポート(全件) <input type="checkbox"/> 検索 <table border="1"> <tr> <td>ルールID</td> <td>ルール名</td> <td>クリア</td> </tr> </table>	ルールID	ルール名	クリア
ルールID	ルール名	クリア		
メール定義	<input type="checkbox"/> エクスポート(全件) <input type="checkbox"/> 検索 <table border="1"> <tr> <td>メールID</td> <td>メール名</td> <td>クリア</td> </tr> </table>	メールID	メール名	クリア
メールID	メール名	クリア		
IMBox定義	<input type="checkbox"/> エクスポート(全件) <input type="checkbox"/> 検索			

3. エクスポートしたい定義を選択します。

コンテンツID	コンテンツ名	備考
<input checked="" type="checkbox"/> contents_javaee	JavaEE開発モデル	
<input type="checkbox"/> contents_script	スクリプト開発モデル	

4. 「エクスポート」をクリックします。

確認ダイアログが表示されたら「OK」をクリックします。

メール定義	<input type="checkbox"/> エクスポート(全件) <input checked="" type="checkbox"/> 検索
IMBox定義	<input type="checkbox"/> エクスポート(全件) <input checked="" type="checkbox"/> 検索
一覧表示パターン定義	<input type="checkbox"/> エクスポート(全件) <input checked="" type="checkbox"/> 検索
フローグループ定義	<input type="checkbox"/> エクスポート(全件) <input checked="" type="checkbox"/> フローとの関連付け情報を含める <input checked="" type="checkbox"/> 検索
管理グループ定義	<input type="checkbox"/> エクスポート(全件) <input checked="" type="checkbox"/> フロー、ルート、コンテンツ、メール、IMBoxとの関連付け情報を含める <input checked="" type="checkbox"/> 検索
代理管理者設定	<input checked="" type="checkbox"/> エクスポート(全件)

フローグループ定義で「フローとの関連付け情報を含める」のチェックを外すと、フローグループ定義に関連付けられたフローのリンク情報をファイルに含めずに出力することができます。

コラム

エクスポート処理中に「閉じる」をクリックすると、途中でエクスポート処理を中止することができます。エクスポート処理を中止すると、それまで出力されていたファイルは削除されます。

5. エクスポートすることができました。

10件完了しました。(正常:10件、異常:0件)

出力ファイル名: tenant_20140710(1).xml

エクスポート実行結果
正常終了

全てのマスタ定義が問題なくエクスポートされた場合は、エクスポート実行結果に「正常終了」が表示されます。エクスポート時に問題があったマスタ定義は、赤字で問題の内容が表示されます。赤字で表示された定義情報については出力されていません。また、出力されたファイルの名称が「出力ファイル名」として表示されます。

i コラム

エクスポートしたファイルは、パブリックストレージの以下のディレクトリ配下に出力されます。

「%PUBLIC_STORAGE_PATH%/im_workflow/data/<テナントID>/import_export/」

エクスポートしたファイルをローカルにダウンロードしたい場合には、エクスポート結果のファイル名をクリックします。

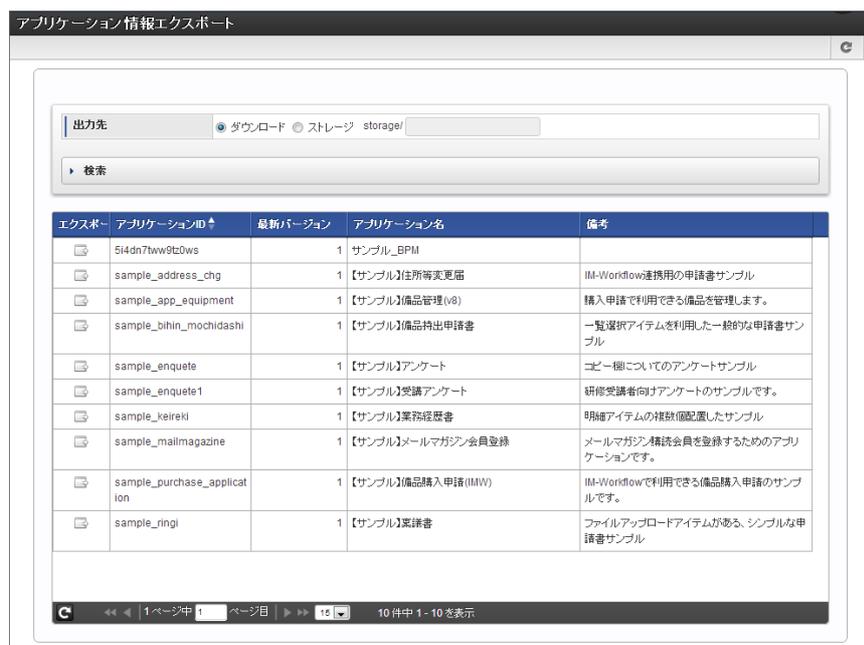
(ローカルへのダウンロードを有効にするためには、ワークフローパラメータの設定が必要です。ワークフローパラメータの詳細は、別紙「IM-Workflow 仕様書」を参照してください。)



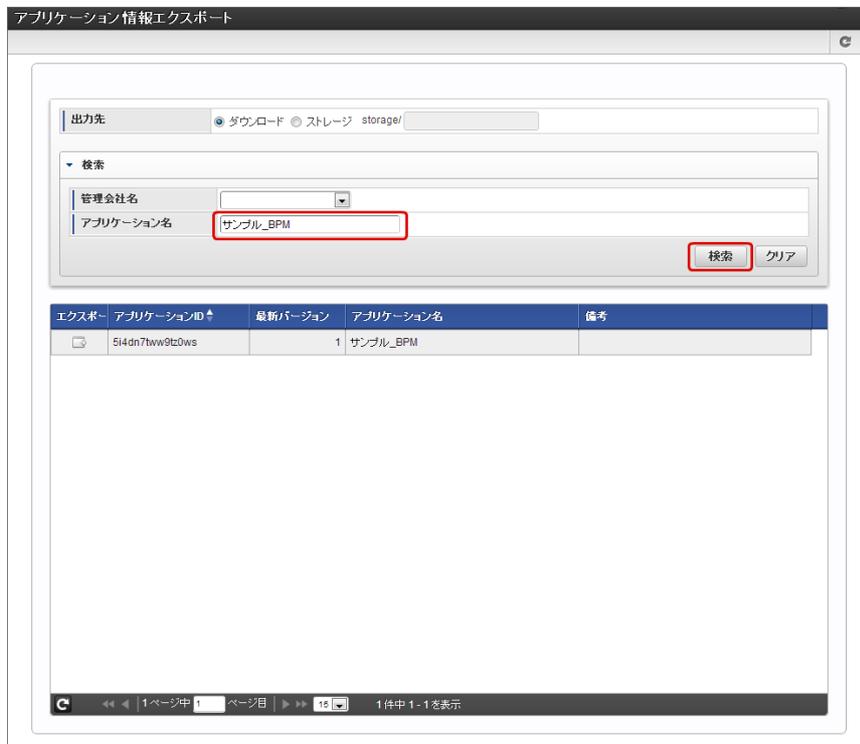
アプリケーション情報をエクスポートする

IM-FormaDesigner のエクスポート機能を利用して、IM-BIS で作成したBISフロー／ワークフローに必要なアプリケーション情報をエクスポートします。

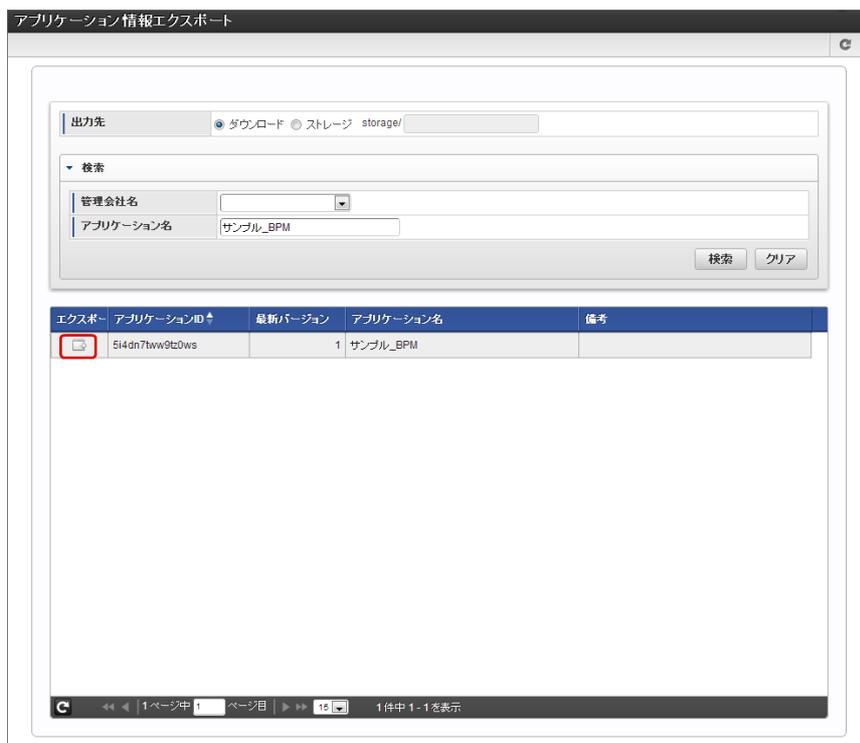
1. 「サイトマップ」→「IM-BIS」→「システム管理者」→「エクスポート」→「IM-BIS」→「IM-Formaアプリエクスポート」をクリックします。
2. 「アプリケーション情報エクスポート」画面が表示されます。



3. 検索条件をクリックして開き、「アプリケーション名」にエクスポート対象のBIS名を入力して「検索」をクリックします。



4. 「エクスポート」をクリックすると、エクスポートファイルを出力します。



データソース定義をエクスポートする

IM-FormaDesigner のエクスポート機能を利用して、IM-BIS で作成したBISフロー／ワークフローに必要なデータソース定義をエクスポートします。

- 「サイトマップ」→「IM-BIS」→「エクスポート」→「データソース定義」→「データソース定義エクスポート」をクリックします。
- 「データソース情報エクスポート」画面が表示されます。
 - エクスポートタイプが「一括」の場合
データソース定義をすべてエクスポートします。

- エクスポートタイプが「選択」の場合
一覧で選択したデータソース定義をエクスポートします。

データソースタイプ	データソース名	備考
<input type="checkbox"/>	テナントDBクエリ	権限検索 購入申請の備品情報を取得するためのクエリです。

3. 「エクスポート実行」をクリックすると、エクスポートファイルを出力します。

テンプレートカテゴリをエクスポートする

フォームのテンプレート機能のテンプレートカテゴリをエクスポートします。

1. 「サイトマップ」→「IM-BIS」→「エクスポート」→「テンプレートカテゴリ定義」→「テンプレートカテゴリ定義エクスポート」をクリックします。
2. 「テンプレートカテゴリ定義エクスポート」画面が表示されます。

3. 「エクスポート実行」をクリックすると、エクスポートファイルを出力します。



IM-BIS の定義をエクスポートする

IM-BIS の定義をエクスポートします。

1. 「サイトマップ」→「IM-BIS」→「エクスポート」→「IM-BIS」→「IM-BIS エクスポート」をクリックします。
2. 「BISエクスポート」画面が表示されます。



3. 一覧の表示を絞り込む場合には、「BIS作成種類」、「BIS名」、「BIS ID」、説明のいずれかを検索条件に指定し、「検索」をクリックします。



4. 「エクスポート」をクリックすると、エクスポートファイルをローカルに出力します。



注意

BIS定義のサイズが合計4GB以上の状態でエクスポートを行った場合に、システムエラーが発生する可能性があります。また、「設計文書管理」を利用してサイズの大きいファイルを添付している場合には、ダウンロードに時間がかかりますので注意してください。

インポートを実行する

IM-BIS の定義をインポートするには、以下の手順で行います。

IM-Workflow に関する定義ファイルをインポートする

IM-Workflow のインポート機能を利用して、IM-BIS で作成したBISフロー／ワークフローに必要な下記の定義をインポートします。

1. インポート対象は、パブリックストレージの以下のディレクトリ配下に配置します。
「%PUBLIC_STORAGE_PATH%/im_workflow/data/<テナントID>/import_export/」
2. 「サイトマップ」→「IM-BIS」→「インポート」→「IM-BIS」→「IM-Workflow インポート」をクリックします。
3. 「インポート」画面が表示されます。



4. インポート対象のファイルのチェックボックスをオンにし、「データ選択へ」をクリックします。



5. インポート対象のチェックボックスをオンにして選択します。

コンテンツID	コンテンツ名	インポートファイル名
<input checked="" type="checkbox"/> 514dn7tw9tz2ws	サンプル_BPM	ueda_20121225.xml

ルートID	ルート名	インポートファイル名
<input checked="" type="checkbox"/> 514dn7tw9tz4ws	サンプル_BPM	ueda_20121225.xml

フローID	フロー名	インポートファイル名
<input checked="" type="checkbox"/> 514dn7tw9tz6ws	サンプル_BPM	ueda_20121225.xml

ルールID	ルール名	インポートファイル名
<input checked="" type="checkbox"/> rule_sample_01	合計金額+10000未満	ueda_20121225.xml

6. 「インポート」をクリックしてインポートを実行します。



注意

異なる環境からの IM-Workflow に関する定義ファイルのインポート時に、インポート元の環境およびインポート先の環境のワークフローパラメータの設定に関する注意事項があります。

詳細は以下のドキュメントを参照してください。

- 「IM-Workflow 仕様書」 - 「バージョンの設定」

アプリケーション情報をインポートする

IM-FormaDesigner のインポート機能を利用して、IM-BIS で作成したBISフロー／ワークフローに必要なアプリケーション情報をインポートします。

- 「サイトマップ」→「IM-BIS」→「インポート」→「IM-BIS」→「IM-Formaインポート」をクリックします。
- 「アプリケーション情報インポート」画面が表示されます。

- 「インポートファイル」にインポートするファイルの情報を設定します。

4. 「インポート実行」をクリックすると、インポートを実行します。



i コラム

アプリケーション情報インポートに伴い、テーブルの作成や更新が必要な場合には、インポート後に以下の作業が必要です。

- a. インポート先環境に対象のテーブルが存在しない（新規にアプリケーションを作成した等）
この場合には、エクスポート時に作成されたテーブル作成のDDLをインポート処理時に実行するため、追加の作業は必要ありません。
- b. インポート先環境に対象のテーブルが存在する
この場合には、インポート処理時にはテーブルの更新・再作成等の処理は一切行われません。
カラムの追加等のテーブル操作が必要な場合、インポート後にBISの編集画面から「定義の反映」を実行してください。
定義の反映の処理により、テーブルに必要な処理が実行されます。
該当のBIS定義に複数のバージョンが含まれる場合には、バージョン単位に「定義の反映」を実行してください。

IM-FormaDesigner のアプリケーションのインポートの詳細については「[IM-FormaDesigner 作成者操作ガイド](#)」-「[アプリケーションの個別対応一覧](#)」を参照してください。

データソース定義をインポートする

IM-FormaDesigner のインポート機能を利用して、IM-BIS で作成したBISフロー／ワークフローに必要なデータソース定義をインポートします。

1. 「サイトマップ」→「IM-BIS」→「インポート」→「データソース定義」→「データソース定義インポート」をクリックします。
2. 「データソース情報インポート」画面が表示されます。



3. 「インポートファイル」にインポートするファイルの情報を設定します。



4. 「インポート実行」をクリックすると、インポートを実行します。

データソース情報インポート

インポートファイル *

ローカル datasource...5-1714.zip

ストレージ storage/



コラム

データソース種別「JAVA」をインポートする場合は、JAVAプログラムを別途配置する必要があります。
詳細は、「[JAVA \(非推奨\)](#)」を参照してください。

テンプレートカテゴリの定義をインポートする

フォームのテンプレート機能のテンプレートカテゴリをインポートします。

1. 「サイトマップ」→「IM-BIS」→「インポート」→「テンプレートカテゴリ定義」→「テンプレートカテゴリ定義インポート」をクリックします。
2. 「テンプレートカテゴリ定義インポート」画面が表示されます。

テンプレートカテゴリインポート

インポートファイル *

ローカル 選択されていません

ストレージ storage/

インポートオプション *

上書き マージ

3. 「インポートファイル」にインポートするファイルの情報を設定します。

テンプレートカテゴリインポート

インポートファイル *

ローカル 選択されていません

ストレージ storage/

インポートオプション *

上書き マージ

4. 「インポートオプション」にインポートの方法を指定します。

- 上書き

テンプレートカテゴリ定義のデータをすべて削除してから、インポートファイルのデータを登録します。



■ マージ

テンプレートカテゴリ定義のデータにインポートファイルのデータを追加します。



5. 「インポート実行」をクリックすると、インポートを実行します。



IM-BIS の定義をインポートする

IM-BIS の定義をインポートします。

1. 「サイトマップ」→「IM-BIS」→「インポート」→「IM-BIS」→「IM-BIS インポート」をクリックします。
2. 「BISインポート」画面が表示されます。



3. 「インポートファイル」にインポートするファイルの情報を設定します。



4. 「インポート実行」をクリックすると、インポートを実行します。



一括インポート・エクスポートを行う

ここでは、一括インポート・エクスポートの実行方法について説明します。

Contents

- 一括インポート・エクスポートとは
- 一括エクスポートを実行する
- 一括インポートを実行する

一括インポート・エクスポートとは

特定のBIS定義とその定義に関連する定義のインポート・エクスポートをまとめて実行できる機能です。詳細については、以下のドキュメントを参照してください。

- 「IM-BIS 仕様書」 - 「一括インポート・エクスポートの仕様」

i コラム

一括インポート・エクスポートを利用した場合、以下の定義については個別インポートで行っていた TableMaintenance によるテーブル・インポート、テーブル・エクスポートは実行不要です。

- 採番ルール定義
- 管理グループ定義

! 注意

一括インポート・エクスポートは、対象の定義ファイルの数、含まれる添付ファイル（設計文書管理や、画面アイテム「イメージ」で指定したファイル等）のサイズに比例して所要時間が長くなります。
多くの定義を対象に一括インポート・エクスポートを行う場合には、上記の観点 considering して実行するようにしてください。

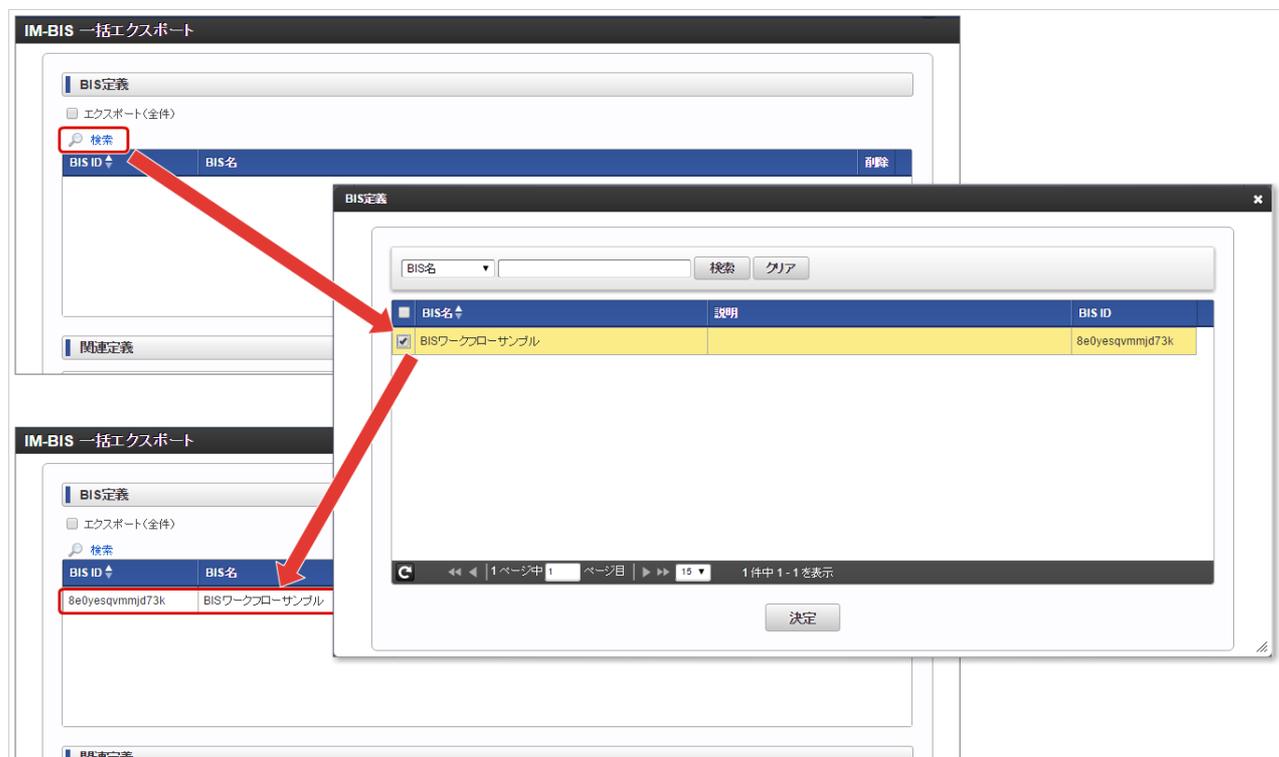
一括エクスポートを実行する

一括エクスポートを実行するには、以下の手順で行います。

1. 「サイトマップ」→「IM-BIS」→「エクスポート」→「IM-BIS 一括エクスポート」をクリックします。
2. エクスポート対象のマスタ定義を選択します。
 - 対象のマスタ定義すべてをエクスポートの対象としたい場合、「エクスポート（全件）」のチェックボックスをクリックしてください。



- 特定のマスタ定義をエクスポートの対象としたい場合、「検索」をクリック後、検索画面から対象のマスタ定義を追加してください。



3. 「エクスポート実行」をクリックします。
確認ダイアログが表示されたら「決定」をクリックします。

IM-BIS 一括エクスポート

BIS定義

エクスポート(全件)

検索

BIS ID ↑	BIS名	削除
8e0yesqvmjmd73k	BISワークフロー-サンプル	✕

関連定義

ルール定義 案件フローバリエーション定義 管理グループ定義 IMBox定義 メール定義 データソース定義 振替ルール定義

エクスポート(全件)

検索

ルールID ↑	ルール名	削除
---------	------	----

エクスポート実行

4. 一括エクスポートを実行することができました。
ファイル名をクリックすると、ローカルにエクスポートしたファイルをダウンロードできます。

IM-BIS 一括エクスポート

① 選択した定義をエクスポートしました。
8e0zvh9ontyn03k/bis_lump_20160622-1424.zip

BIS定義

エクスポート(全件)

検索

BIS ID ↑	BIS名	削除
8e0yesqvmjmd73k	BISワークフロー-サンプル	✕

エクスポートでエラーが発生した場合には、メッセージを確認し、必要な対応を行ってください。

i コラム

エクスポートしたファイルは、パブリックストレージの以下のパス配下に出力されます。
「%PUBLIC_STORAGE_PATH% / im_bis / lump_export / %一意キー%」

一括インポートを実行する

一括インポートを実行するには、以下の手順で行います。

- 「サイトマップ」→「IM-BIS」→「インポート」→「IM-BIS 一括インポート」をクリックします。
- 「IM-BIS 一括インポート」画面が表示されます。

IM-BIS 一括インポート

インポートファイル*

ローカル 選択されていません

ストレージ storage/

インポート実行

3. 「インポートファイル」にインポートするファイルの情報を設定します。

パブリックストレージにインポート対象のファイルが配置されている場合、ストレージを選択後に `%PUBLIC_STORAGE_PATH%` 配下からファイル名までを入力してください。

4. 「インポート実行」をクリックします。

確認ダイアログが表示されたら「決定」をクリックします。

5. 一括インポートを実行することができました。

関連定義	
ルール定義	ルール定義のインポートに成功しました。
案件プロファイル定義	案件プロファイル定義のインポートに成功しました。
管理グループ定義	管理グループ定義のインポートに成功しました。
IMBox定義	IMBox定義のインポートに成功しました。
メール定義	メール定義のインポートに成功しました。
データソース定義	データソース定義のインポートに成功しました。
採番ルール定義	採番ルール定義のインポートに成功しました。

インポート結果にエラーや警告が表示されている場合には、メッセージを確認し、必要な対応を行ってください。

注意

異なる環境からの IM-Workflow に関する定義ファイルのインポート時に、インポート元の環境およびインポート先の環境のワークフローパラメータの設定に関する注意事項があります。
詳細は以下のドキュメントを参照してください。

- 「IM-Workflow 仕様書」 - 「バージョンの設定」

ルール実行履歴レポート機能を利用する

ここではルール実行履歴レポート機能の使い方を説明します。

Contents

- ルール実行履歴レポートとは
- 「ルール実行履歴レポート」画面の機能と各部の説明

ルール実行履歴レポートとは

ルール実行履歴レポートとは、外部連携：ルールを実行した際の実行履歴・実行結果のレポートを、htmlファイルにてダウンロードできる機能です。

本機能を利用することで、今まで IM-BIS から取得することができなかった「実行モード：推論型で複数解が返ってくる」場合でも、解を取得することができます。

実行履歴・実行結果の取得対象は、実行モード：推論型で登録されたすべてのデータソース定義です。

「ルール実行履歴レポート」画面の機能と各部の説明

「ルール実行履歴レポート」画面の内容は以下の通りです。

The screenshot shows the 'ルール実行履歴レポート' (Rule Execution History Report) screen. At the top, there are search filters: (1) Application Name, (2) User Name, and (3) Creation Date. There are (4) Search and (5) Clear buttons. Below the filters is a table with columns: (6) Delete selected reports, (7) Download, (8) Application Name, (9) User Name, (10) Creation Time, (11) Data Source Name, and (12) Report Name. A single row is shown with (13) Report Name 'RuleReport_Sibcq8k47jke2gq.html'. At the bottom, there is a pagination bar showing '1件中 1 - 1 を表示'.

1. アプリケーション名
検索するアプリケーション名を入力します。
2. 作成ユーザ名
検索する作成ユーザ名を入力します。
3. 作成日
検索する作成日の範囲を入力します。
4. 検索ボタン
押下すると検索を実行します。
5. クリアボタン
押下すると検索条件をクリアします。
6. 選択したレポートを削除
一覧からレポートを選択して押下すると対象のレポートを削除します。
7. 選択（チェックボックス）
チェックをオンにすると、対象の行のレポートを選択します
ヘッダーのチェックボックスをオンにすると、一覧に表示されたレポートをすべて選択します。
8. ダウンロード
ダウンロードアイコンを押下すると、対象のレポートをダウンロードします。
9. アプリケーション名
レポートが作成されたアプリケーション名です。
10. 作成ユーザ名
レポートを作成したユーザ名です。
11. 作成日時
レポートが作成された日時です。

12. データソース名
レポートを作成した際に使用されたデータソース名です。
13. レポート名
レポートのファイル名です。

**注意**

ルール実行履歴レポートは、現在英語でのレポート出力のみしか対応しておりません。

一括処理対象者変更を利用する

一括処理対象者メニューでは、未完了案件の処理対象者を一括で変更できます。
退職等の理由でBISフロー／ワークフローを処理中のまま不在になったユーザから、在籍中のユーザに処理対象者を変更する場合等に利用できます。

BISシステム管理者は、全ての案件の処理対象者を変更できます。

BIS業務管理者は、管理対象となっているフロー（BIS定義）の案件の処理対象者を変更できます。

目次

- [変更したい案件を表示する](#)
- [処理対象者を変更する](#)

変更したい案件を表示する

1. BISシステム管理者の場合は、「サイトマップ」→「IM-BIS」→「システム管理者」→「フロー運用管理」→「一括処理対象者変更（ワークフロー）[*1]」をクリックします。
BIS業務管理者の場合は、「サイトマップ」→「IM-BIS」→「業務管理者」→「フロー」→「一括処理対象者変更（ワークフロー）[*1]」をクリックします。
*1 - 変更したい案件が BISフロー の場合は「一括処理対象者変更（BISフロー）」をクリックします。
2. 「一括処理対象者変更」画面が表示されます。初期表示の場合は、一覧には何も表示されません。
「変更元」を選択して、処理対象者の変更を行いたい案件を検索します。

3. 変更したい案件を検索します。
「変更元」の右にある「検索」リンクをクリックします。
「変更元」を選択するフローティングウィンドウが表示されます。

「変更元」は以下を選択できます。

- ユーザ

検索条件として指定したユーザの案件を検索します。

- 処理対象者なし
検索条件として処理対象者が存在しない案件を検索します。

i コラム

以下のケースが処理対象者が存在しない場合に該当します。

- 「ルート定義」 - 「バージョン」 - 「編集」で、処理対象者に「指定なし」を設定した場合
- 「ルート定義」 - 「バージョン」 - 「編集」で、処理対象者に指定したユーザや組織が無効、または有効期限切れで、展開された処理対象者がゼロ人の場合

4. 例として、「変更元」をユーザにして「青柳辰巳」が処理対象者になっている案件を検索します。
フローティングウィンドウの「ユーザ」を選択→「ユーザ検索」で「青柳辰巳」を選択します。

ユーザ検索

検索基準日: 2015/11/24 ロケール: 日本語

キーワード検索 | 会社組織(キーワード) | 会社組織(ツリー) | パブリックグループ(キーワード)

パブリックグループ(ツリー)

検索キーワードを入力してください。 **青柳辰巳**

名前 コード フリガナ
 前方一致 完全一致 部分一致

検索

あ行	あ	い	う	え	お
か行	か	き	く	け	こ
さ行	さ	し	す	せ	そ
た行	た	ち	つ	て	と
な行	な	に	ぬ	ね	の
は行	は	ひ	ふ	へ	ほ
ま行	ま	み	む	め	も
や行	や		ゆ	よ	
ら行	ら	り	る	れ	ろ
わ行	わ				

決定

「一括処理対象者変更」画面の一覧に検索結果が表示されました。

一括処理対象者変更 (ワークフロー)

BPM 表示条件

変更元 * 青柳辰巳 検索 変更先 * 検索

変更通知を送信する 変更

優先度	案件番号	案件名	申請基準	申請日	申請者	フロー名	ノード名	状態	到達日	処理期限	フロー	履歴
	000000025	サンプルフロー - (一括処理対象者変更) 5	2015/11/2	2015/11/2	林政義	サンプルフロー - (一括処理対象者変更)	承認	承認	2015/11/2			
	000000024	サンプルフロー - (一括処理対象者変更) 4	2015/11/2	2015/11/2	林政義	サンプルフロー - (一括処理対象者変更)	承認	承認	2015/11/2			
	000000023	サンプルフロー - (一括処理対象者変更) 3	2015/11/2	2015/11/2	林政義	サンプルフロー - (一括処理対象者変更)	承認	承認	2015/11/2			
	000000022	サンプルフロー - (一括処理)	2015/11/2	2015/11/2	林政義	サンプルフロー - (一括処理)	承認	承認	2015/11/2			

「青柳辰巳」が処理対象者になっているタスク（ノード）単位で一覧が表示されます。

- フロー

を押下すると、案件のフロー参照画面がポップアップ表示されます。案件のフロー図、現在処理中のタスク（ノード）、各タスク

(ノード) の処理者や処理日時等を確認できます。

- 履歴 

 を押下すると、案件の履歴参照画面がポップアップ表示されます。案件のフロー図、処理が終了したタスク（ノード）の処理日時や、処理の際に処理者が付けたコメント等を確認できます。

i コラム

一括処理対象者変更の検索・表示条件指定

表示条件を指定することで、現在表示されている一括処理対象者変更対象の案件を、さらに絞り込んで表示させることができます。

- ＜表示条件＞を押下します。
「＜表示条件＞」画面（タブ：検索条件）が表示されます。



- 絞込み条件 - 申請
申請／処理開始タスク（ノード）の対象／対象外を検索条件として指定します。初期値は対象外です。
対象にすると、申請／処理開始タスク（ノード）で処理中の案件を一覧に表示します。
- 優先度
案件で指定されている優先度を検索条件として指定します。プルダウンから、「通常」「高」「低」を選択します。
- 案件番号
案件の案件番号を検索条件として指定します。入力値の部分一致で検索します。
- 案件名
案件の案件名を検索条件として指定します。入力値の部分一致で検索します。
- 申請基準日
案件の申請基準日を検索条件として範囲指定します。  を押下して、表示されるカレンダーから日付を選択します。
一度指定した値を解除したい場合は、  を押下します。
- 申請日
案件の申請日を検索条件として範囲指定します。  を押下して、表示されるカレンダーから日付を選択します。
一度指定した値を解除したい場合は、  を押下します。
- 申請者
案件の申請者を検索条件として指定します。
 ＜検索＞を押下すると、申請者検索画面がポップアップ表示されますので、検索条件としたい申請者を選択します。
一度指定した値を解除したい場合は、  ＜解除＞を押下します。
- フロー名
案件のフロー名を検索条件として指定します。入力値の部分一致で検索します。
- ノード名
ノード名を検索条件として指定します。入力値の部分一致で検索します。
- 状態

案件の状態を検索条件として指定します。「申請待ち」「承認待ち」「再申請待ち」「保留状態」のいずれかを選択します。

- 到達日

案件の到達日を検索条件として範囲指定します。  を押下して、表示されるカレンダーから日付を選択します。

指定した値を解除したい場合は、  を押下します。

- 処理期限

案件の処理期限日付を検索条件として範囲指定します。  を押下して、表示されるカレンダーから日付を選択します。

指定した値を解除したい場合は、  を押下します。

- 案件プロパティで設定した項目

案件プロパティに設定した項目で、「一覧表示項目の項目として使用する」が有効、かつ一覧表示パターンで表示項目に設定しているとき、案件プロパティの値に基づいて検索できます。

検索範囲は、案件プロパティ定義の設定で「単一検索」、「範囲検索」のどちらが設定されているかによって異なります。

単一値の場合、入力値の部分一致で検索します。範囲検索の場合、入力値以上、または入力値以下で検索します。

一覧表示パターンを選択することで、現在表示されている参照対象の案件一覧の表示項目やソートを変更することができます。

- <表示条件>を押下します。

タブを「表示設定」に切り替えます。



- 一覧表示パターン

一覧表示パターン定義で、対象一覧を「一括処理対象者変更」として定義したパターン名が、プルダウンで表示されます。設定したい表示パターンを選択します。

「プレビュー」には、選択した一覧表示パターンのプレビューが表示されます。

初期ソートが設定されている場合は、表示の際に、指定されている項目に対して、指定されている順（昇順/降順）で、ソートをかけてから表示します。

処理対象者を変更する

1. チェックボックスをクリックして、処理対象者を変更したいレコードを選択します。

一括処理対象者変更 (ワークフロー)

BPM 表示条件

変更元: 青柳辰巳 検索 変更先: 検索

変更通知を送信する 変更

	優先度	案件番号	案件名	申請基準	申請日	申請者	フロー名	ノード名	状態	到達日	処理期限	フロー	履歴
<input checked="" type="checkbox"/>		0000000025	サンプルフロー (一括処理対象者変更) 5		2015/11/2	林政義	サンプルフロー (一括処理対象者変更)	承認	▶	2015/11/2			
<input type="checkbox"/>		0000000024	サンプルフロー (一括処理対象者変更) 4		2015/11/2	林政義	サンプルフロー (一括処理対象者変更)	承認	▶	2015/11/2			
<input checked="" type="checkbox"/>		0000000023	サンプルフロー (一括処理対象者変更) 3		2015/11/2	林政義	サンプルフロー (一括処理対象者変更)	承認	▶	2015/11/2			
<input checked="" type="checkbox"/>		0000000022	サンプルフロー (一括処理対象者変更) 2		2015/11/2	林政義	サンプルフロー (一括処理対象者変更)	承認	▶	2015/11/2			
<input type="checkbox"/>		0000000021	サンプルフロー (一括処理対象者変更) 1		2015/11/2	林政義	サンプルフロー (一括処理対象者変更)	承認	▶	2015/11/2			

2. 変更先の処理対象者を選択します。

「変更先」の右にある「検索」リンクをクリックします。

「変更先」を選択するフローティングウィンドウが表示されます。



「変更先」は以下を選択できます。

- ユーザ
- 組織
- パブリックグループ
- 役職
- 役割
- 組織+役職
- パブリックグループ+役割

3. 例として、「変更先」をユーザにして「上田辰男」に変更します。

フローティングウィンドウの「ユーザ」を選択→「ユーザ検索」で「上田辰男」を選択します。

ユーザ検索

検索基準日: 2015/11/24 ロケール: 日本語

検索キーワードを入力してください。 **上田辰男**

名前
 コード
 フリガナ
 前方一致
 完全一致
 部分一致

あ行	あ	い	う	え	お
か行	か	き	く	け	こ
さ行	さ	し	す	せ	そ
た行	た	ち	つ	て	と
な行	な	に	ぬ	ね	の
は行	は	ひ	ふ	へ	ほ
ま行	ま	み	む	め	も
や行	や		ゆ	よ	
ら行	ら	り	る	れ	ろ
わ行	わ				

4. 変更先に変更の通知を行いたい場合は、「変更通知を送信する」のチェックをオンにしてください。

変更元: 青柳辰巳 変更先: 上田辰男

変更通知を送信する

優先度	案件番号	案件名	申請基準	申請日	申請者	フロー名	ノート名	状態	到達日	処理期限	フロー	履歴
<input checked="" type="checkbox"/>	0000000025	サンプルフロー (一括処理対象者変更) 5	2015/11/2	2015/11/2	林政義	サンプルフロー (一括処理対象者変更)	承認		2015/11/2			

5. 「変更」ボタンを押下します。
確認メッセージボックスが表示されますので、「決定」を押下します。



コラム

システム日時点で有効なユーザがタスク（ノード）の処理対象者として展開されます。

6. 選択した案件の処理対象者が「変更元」から「変更先」に変更されました。
以下は、「上田辰男」の未処理一覧画面です。変更した案件が表示されました。

処理	振替	優先度	案件番号	案件名	申請基	申請日	申請者	フロー	ノード	状態	到達日	処理期	処理権	フロー	履歴
		●	0000000025	サンプルフロー（一括処理対象者変更）5		2015/11	2015/11	林政義	サンプルフロー（一括処理対象者変更）	承認	2015/11				
		●	0000000023	サンプルフロー（一括処理対象者変更）3		2015/11	2015/11	林政義	サンプルフロー（一括処理対象者変更）	承認	2015/11				
		●	0000000022	サンプルフロー（一括処理対象者）		2015/11	2015/11	林政義	サンプルフロー（一括処理対象者）	承認	2015/11				

フロー設計書を出力する

「フロー設計書出力」では、IM-BISやIM-Workflowで作成したフローに紐づく設定内容をExcelファイルに出力することができます。

目次

- [フロー設計書出力を利用する](#)
- [フロー設計書を出力する](#)

フロー設計書出力を利用する

フロー設計書出力は、フローの設定内容をExcelファイルに出力する機能です。

フロー設計書出力機能は、IM-BISの機能拡張モジュールとして提供されています。

IM-BISのデフォルト構成には含まれないため、以下のモジュールを構成に含めてwar作成を行ってください。

モジュールID	モジュール名
jp.co.intra_mart.bis_design_document	IM-BIS 設計書出力

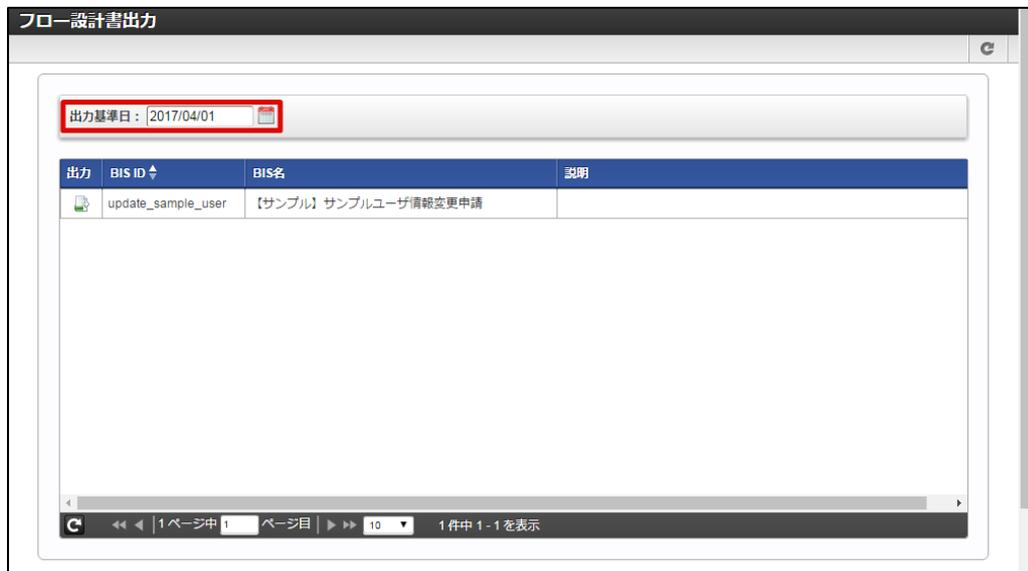
フロー設計書を出力する

フロー設計書を出力する手順は以下の通りです。

- 「フロー設計書出力」の画面を表示します。
 - IM-BIS フロー設計書を出力する場合

「サイトマップ」→「IM-BIS/IM-Workflow設計書出力」→「IM-BIS」→「システム管理者」→「フロー設計書出力」をクリックします。
 - IM-Workflow フロー設計書を出力する場合

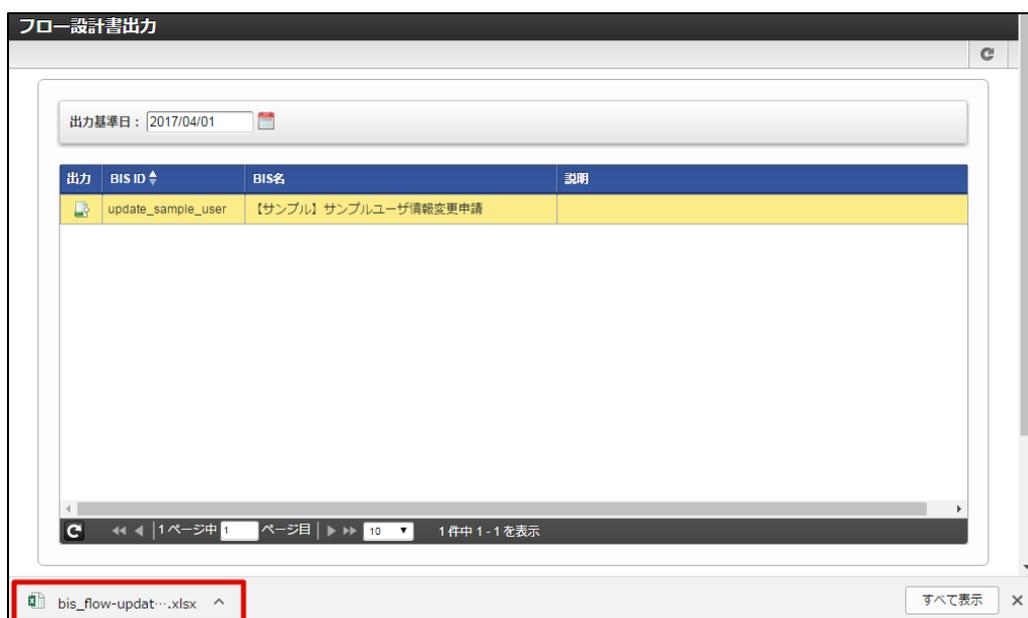
「サイトマップ」→「IM-BIS/IM-Workflow設計書出力」→「IM-Workflow」→「ワークフロー管理者」→「フロー設計書出力」をクリックします。
- 出力基準日を選択します。



3. フロー設計書を出力したいフローの出力アイコンをクリックします。



4. フロー設計書が出力されます。
出力される設計書の詳細は「[IM-BIS 仕様書](#)」-「[フロー設計書出力](#)」を参照してください。



画像は Google Chrome の場合の例です。

ご利用のブラウザやバージョンによってダウンロードの表示が異なる場合があります。



コラム

画面設計書出力のファイル形式

IM-BIS の画面設計書は **XLSX** 形式で出力されます。



コラム

IM-Workflowのフロー設計書出力の画面から出力したフロー設計書にはBIS定義の情報が出力されません。

BIS定義の情報を含むフロー設計書を出力する場合は、IM-BISフロー設計書の画面でフロー設計書を出力してください。

印影表示を利用する

ここでは、IM-BIS のBISフロー／ワークフローで、画面アイテム「印影表示」を利用するための方法を説明します。

Contents

- 画面アイテム「印影表示」とは
- ワークフローパラメータを設定する
- IM-BIS で作成したコンテンツへの印影処理を設定する
- BIS担当者が利用する印影を設定する

画面アイテム「印影表示」とは

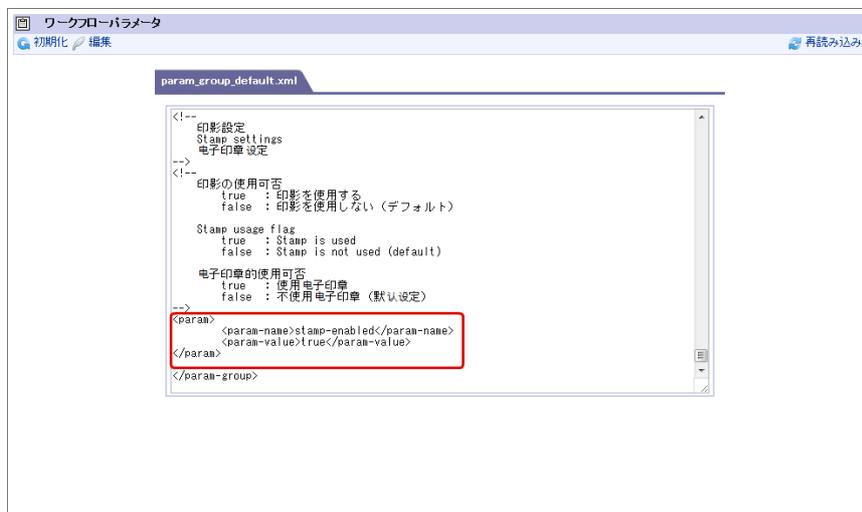
画面アイテム「印影表示」は、BISフロー／ワークフローで印影を表示するためのアイテムです。
申請・承認処理などで、印影を利用できます。

ワークフローパラメータを設定する

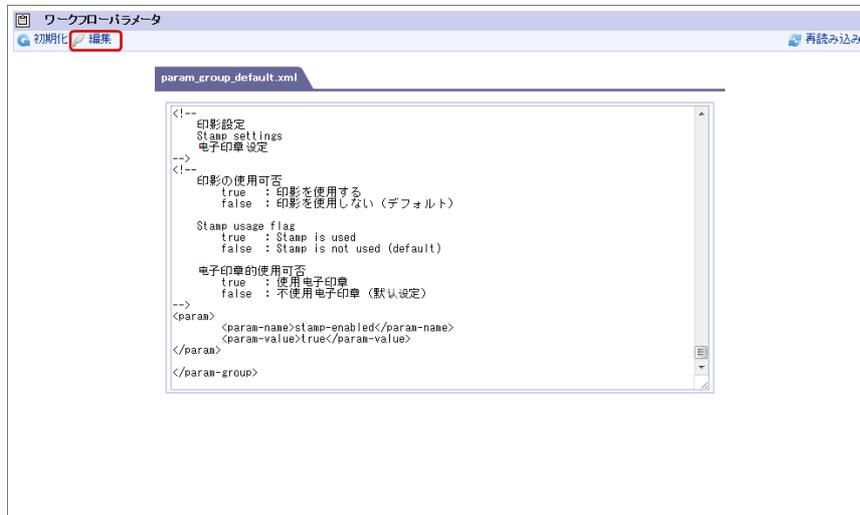
IM-BIS で作成したフォームで画面アイテム「印影表示」を利用するためには、ワークフローパラメータを変更する必要があります。
ワークフローパラメータは、以下の手順で設定してください。

1. 「サイトマップ」→「ワークフロー管理者」→「ワークフローパラメータ」をクリックします。
2. 「param_group_%テナントID%.xml」で下記の通りに修正します。

```
<param>
  <param-name>stamp-enabled</param-name>
  <param-value>true</param-value>
</param>
```



3. 「編集」をクリックします。
確認ダイアログが表示されたら「OK」をクリックします。



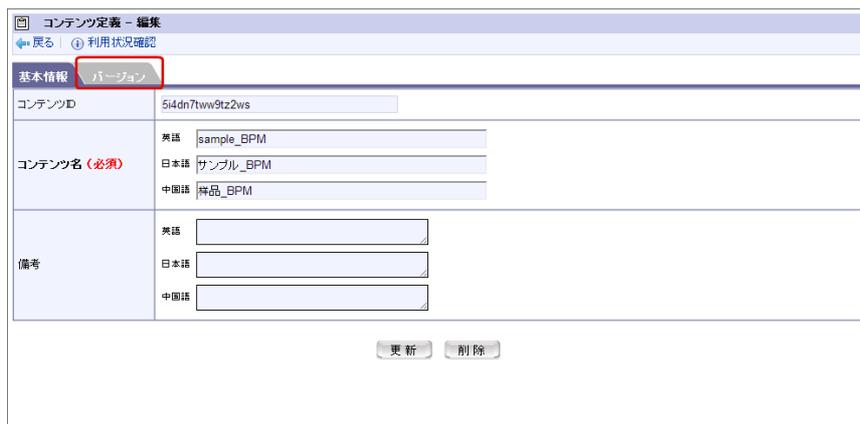
IM-BIS で作成したコンテンツへの印影処理を設定する

IM-BIS で作成したフォームに対応するコンテンツ定義に印影処理を設定するためには、以下の手順で設定してください。

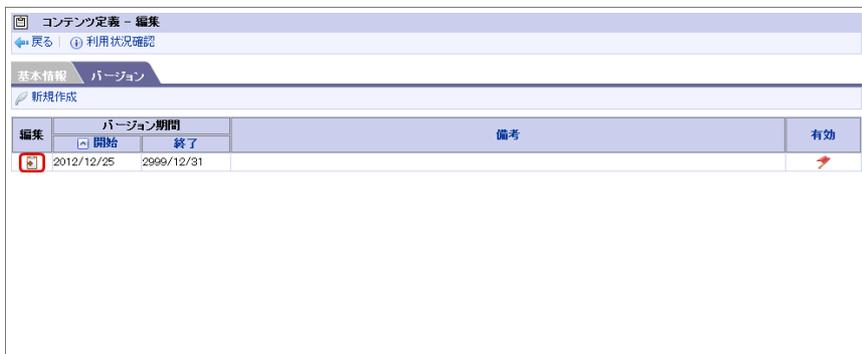
1. 「サイトマップ」→「ワークフロー」→「ワークフロー管理者」→「マスタ定義」→「コンテンツ定義」をクリックします。
2. コンテンツ名が設定対象の「BIS名」となっているコンテンツ定義の「編集」をクリックします。



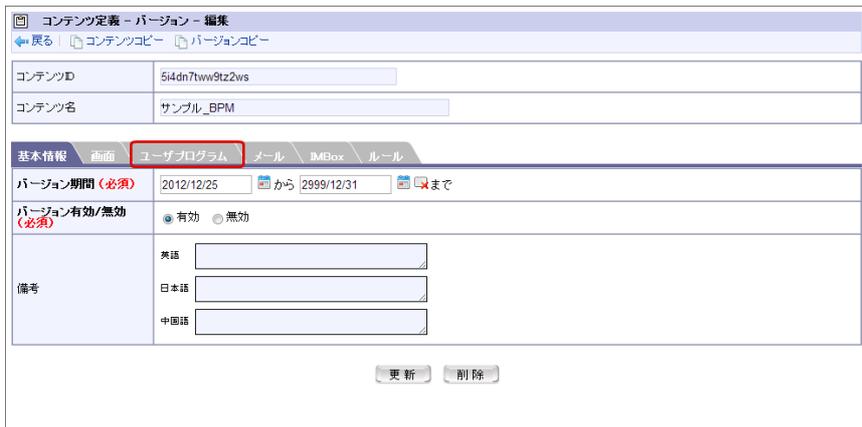
3. 「バージョン」をクリックします。



4. 対象のバージョンの「編集」をクリックします。



5. 「ユーザプログラム」をクリックします。



6. 「新規作成」をクリックして、「印影処理」のプログラムを設定します。



申請/処理開始ノード、承認/処理ノードなどのタスクへ印影処理を設定する

対象のコンテンツ定義を利用するBISフロー/ワークフローのすべてのタスクに以下の手順で印影処理を設定します。

- 「ユーザプログラム定義 - 新規作成」画面で、以下の内容に従って、必要なユーザプログラムを設定します。
 - プログラム名：「【アクション処理】印影処理」
 - プラグイン種別：アクション処理
 - 対象ノード：対象のノード名（申請/処理開始ノード、承認/処理ノード）
 - プラグイン種類：「【アクション処理】印影処理」
 - 実行順番：1

ユーザプログラム定義 - 新規作成	
プログラム名(必須)	英語 [Action process] Stamp Process 日本語 【アクション処理】印影処理 中国語 【动作处理】电子印章处理
プラグイン種別(必須)	アクション処理
対象ノード(必須)	申請/処理開始ノード
プラグイン種類(必須)	【アクション処理】印影処理
備考	英語 日本語 中国語
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する
実行順番(必須)	1
登録	

2. 上記の通りに設定したら、「登録」をクリックします。

ユーザプログラム定義 - 新規作成	
プログラム名(必須)	英語 [Action process] Stamp Process 日本語 【アクション処理】印影処理 中国語 【动作处理】电子印章处理
プラグイン種別(必須)	アクション処理
対象ノード(必須)	申請/処理開始ノード
プラグイン種類(必須)	【アクション処理】印影処理
備考	英語 日本語 中国語
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する
実行順番(必須)	1
登録	

以上で、タスクに対する印影処理が設定できました。

案件に対する各処理へ印影処理を設定する

対象のコンテンツ定義を利用するBISフロー/ワークフローのコンテンツ定義の案件処理に以下の手順で印影処理を設定します。

- 「ユーザプログラム定義 - 新規作成」画面で、以下の内容に従って、処理ごとに必要なユーザプログラムを設定します。
 - 案件終了処理に下記の通り設定します。
 - プログラム名：「【案件終了処理】印影処理」
 - プラグイン種別：案件終了処理
 - プラグイン種類：「【案件終了処理】印影処理」
 - 実行順番：1

ユーザプログラム定義 - 新規作成	
プログラム名(必須)	英語 [Job end processing]Stamp Process 日本語 【案件終了処理】印影処理 中国語 【案件结束处理】电子印章处理
プラグイン種別(必須)	案件終了処理
プラグイン種類(必須)	【案件終了処理】印影処理
備考	英語 日本語 中国語
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する
実行順番(必須)	1
登録	

- 案件退避処理に下記の通り設定します。
 - プログラム名：「【案件退避処理】印影処理」

- プラグイン種別：案件退避処理
- プラグイン種類：「【案件退避処理】印影処理」
- 実行順番：1

ユーザプログラム定義 - 新規作成	
プログラム名(必須)	英語 [Job archive processing]Stamp Process 日本語 【案件退避処理】印影処理 中国語 【案件存档处理】电子印章处理
プラグイン種別(必須)	案件退避処理
プラグイン種類(必須)	【案件退避処理】印影処理
備考	英語 日本語 中国語
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する
実行順番(必須)	1

登録

- 過去案件削除処理に下記の通り設定します。
 - プログラム名：「【過去案件削除処理】印影処理」
 - プラグイン種別：過去案件削除
 - プラグイン種類：「【過去案件削除処理】印影処理」
 - 実行順番：1

ユーザプログラム定義 - 新規作成	
プログラム名(必須)	英語 [Past job deletion processing]Stamp Process 日本語 【過去案件削除処理】印影処理 中国語 【过去案件删除处理】电子印章处理
プラグイン種別(必須)	過去案件削除
プラグイン種類(必須)	【過去案件削除処理】印影処理
備考	英語 日本語 中国語
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する
実行順番(必須)	1

登録

- 完了案件削除処理に下記の通り設定します。
 - プログラム名：「【完了案件削除処理】印影処理」
 - プラグイン種別：完了案件削除
 - プラグイン種類：「【完了案件削除処理】印影処理」
 - 実行順番：1

ユーザプログラム定義 - 新規作成	
プログラム名(必須)	英語 [Complete job deletion processing]Stamp Process 日本語 【完了案件削除処理】印影処理 中国語 【已完成案件删除处理】电子印章处理
プラグイン種別(必須)	完了案件削除
プラグイン種類(必須)	【完了案件削除処理】印影処理
備考	英語 日本語 中国語
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する
実行順番(必須)	1

登録

- 未完了案件削除処理に下記の通り設定します。
 - プログラム名：「【未完了案件削除処理】印影処理」
 - プラグイン種別：未完了案件削除
 - プラグイン種類：「【未完了案件削除処理】印影処理」

- 実行順番 : 1

ユーザプログラム定義 - 新規作成	
プログラム名(必須)	英語 [Incomplete job deletion processing]Stamp Process 日本語 【未完了案件削除処理】印影処理 中国語 【未完成案件删除处理】电子印章处理
プラグイン種別(必須)	未完了案件削除
プラグイン種類(必須)	【未完了案件削除処理】印影処理
備考	英語 日本語 中国語
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する
実行順番(必須)	1

登録

2. 各処理ごとに上記の通りに設定したら、「登録」をクリックします。

ユーザプログラム定義 - 新規作成	
プログラム名(必須)	英語 [Incomplete job deletion processing]Stamp Process 日本語 【未完了案件削除処理】印影処理 中国語 【未完成案件删除处理】电子印章处理
プラグイン種別(必須)	未完了案件削除
プラグイン種類(必須)	【未完了案件削除処理】印影処理
備考	英語 日本語 中国語
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する
実行順番(必須)	1

登録

以上で、案件の処理に対する印影処理が設定できました。

BIS担当者が利用する印影を設定する

BISフロー／ワークフローの「申請／処理開始」、「承認／処理」でBIS担当者が利用する印影を設定します。

1. 「サイトマップ」→「IM-BIS」→「印影設定」をクリックします。
2. 「新規作成」をクリックします。

印影設定

新規作成

印影設定

簽込みキーワード

印影情報がまだ作成されていません。

3. 印影名などを入力して「登録」をクリックします。

以上で、BIS担当者が利用する印影が設定できました。

プロセス管理の参照を利用する

ここでは、BISフロー／ワークフローの処理対象者以外のユーザに、参照権限を設定する方法について説明します。

Contents

- プロセス管理の参照とは
- 前提条件
- 参照者を設定する

プロセス管理の参照とは

プロセス管理の参照を設定すると、BISフロー／ワークフローの申請・処理の内容を、処理対象者以外のユーザが参照できます。

！ 注意

参照権限の設定内容は、設定後に開始されたBISフロー／ワークフローの申請・処理が対象です。
(設定前に申請・処理が開始されたBISフロー／ワークフローの内容は参照画面には表示されません。)

！ 注意

業務管理者に対する参照の設定は、「管理グループ」で設定するようにしてください。

i コラム

参照者への案件操作

- 参照者に設定した場合には、設定したユーザのロールに応じて、案件操作の権限設定内容が異なります。
案件操作の詳細については、「[IM-Workflow 仕様書](#)」を参照してください。
 - BISシステム管理者
案件操作権限があるため、参照画面では、案件操作のアイコンが表示され、すべての案件操作を行うことができます。
 - BIS業務管理者
BISシステム管理者に参照者として設定された上で、必要な案件操作権限を付与された場合のみ、案件操作を行うことができます。
 - BIS担当者
BISシステム管理者に参照者として設定された上で、必要な案件操作権限を付与された場合のみ、案件操作を行うことができます。

前提条件

BISシステム管理者に、対象のフロー定義（BIS定義）の管理権限、IM-Workflowの「ワークフロー管理者」メニューの参照権限が付与されていることが前提です。

参照者を設定する

「参照（BISフロー）」「参照（ワークフロー）」を利用するためには、以下の手順で設定します。

1. 「サイトマップ」→「ワークフロー-管理者」→「マスタ定義」→「フロー定義」をクリックします。
2. 「フロー定義」画面で IM-BIS で作成したBISフロー／ワークフローに該当するフロー定義の「編集」アイコンをクリックします。

フロー定義

新規作成 最新情報

フロー名 検索

1-11/11

編集	フローID	フロー名	備考
	5i4dn7twv9tz6ws	サンプル_BPM	
	flow_javaee_01	直線ルート[JavaEE開発モデル]	
	flow_javaee_02	横配置ルート[JavaEE開発モデル]	
	flow_javaee_03	縦配置ルート[JavaEE開発モデル]	
	flow_javaee_04	分岐ルート[JavaEE開発モデル]	
	flow_javaee_05	複合ルート[JavaEE開発モデル]	
	flow_script_01	直線ルート[スクリプト開発モデル]	
	flow_script_02	横配置ルート[スクリプト開発モデル]	
	flow_script_03	縦配置ルート[スクリプト開発モデル]	
	flow_script_04	分岐ルート[スクリプト開発モデル]	
	flow_script_05	複合ルート[スクリプト開発モデル]	

1-11/11

3. 「バージョン」タブをクリックします。

フロー定義 - 編集

[戻る](#)

基本情報 **バージョン**

新規作成

フローID

フロー名 (必須)

英語

日本語

中国語

備考

英語

日本語

中国語

4. 設定するBISフロー／ワークフローのバージョンの「編集」アイコンをクリックします。

フロー定義 - 編集

[戻る](#)

基本情報 **バージョン**

新規作成

編集	バージョン期間	備考	有効	コンテンツ	ルート
	開始	終了			
	2012/12/25	2999/12/31			

5. 「参照者」をクリックします。

6. 検索をクリックして対象者を選択し、一覧に追加します。

7. 更新をクリックします。

参照画面や特殊なタスク（ノード）で表示する画面を設定する

ここでは、参照画面や特殊なタスク（ノード）で、表示する画面の設定方法について説明します。
表示する画面の設定は、BISシステム管理者が実施する必要があります。

Contents

- BISシステム管理者が、表示する画面を設定する必要があるケース
- 参照画面や特殊なタスク（ノード）で表示する画面に申請／処理開始、承認／処理の画面を設定する
- 参照画面や特殊なタスク（ノード）で表示する画面に別の画面を設定する
- 処理中の画面から別の画面を子画面として表示するための設定を行う

BISシステム管理者が、表示する画面を設定する必要があるケース

IM-BIS のルートで、下記の条件に合致する場合には、BISシステム管理者が表示する画面を設定します。

設定を行わない場合、表示される画面のレイアウトが崩れる場合があります。

- 下記の種類のノードを利用しており、申請／処理開始と承認／処理で「共有」を行っていない
 - 確認
 - テンプレート置換
- 下記の画面を利用しており、申請／処理開始と承認／処理で「共有」を行っていない
 - 参照画面（一覧（処理済、未処理、参照など）の詳細で表示する画面）

注意

BISシステム管理者が「IM-BIS - フロー編集」画面以外からフォームを設定した場合は、BIS定義でフォーム関連の編集を行う都度、BISシステム管理者が設定した内容に影響がないかを確認してください。
画面の設定後に IM-BIS から変更を行った場合、正しく動作しない可能性があります。

コラム

参照画面や特殊なノードで表示する画面はノードの種類単位で設定することはできません。
この項で記載している手順を行って設定した画面は、フローで使用されている上記の種類ノードに対して一括で設定されます。

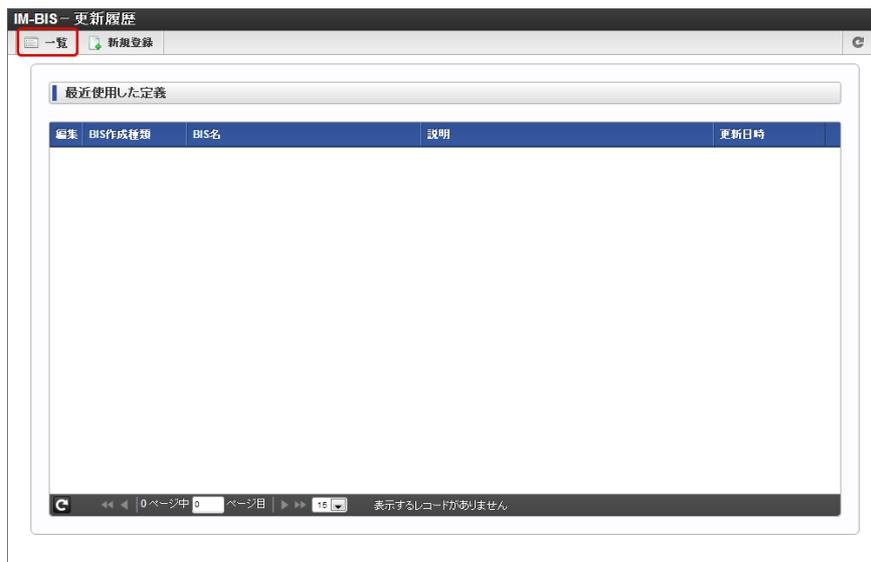
コラム

フローで「共有」を設定している場合には、上記の特殊なノードを実行した際に共有している画面（フォーム）が表示されます。

参照画面や特殊なタスク（ノード）で表示する画面に申請／処理開始、承認／処理の画面を設定する

「IM-BIS - フロー編集」画面からフォームを設定できないノードに、フォームを設定する操作は、以下の手順で行います。

1. 「サイトマップ」→「IM-BIS」→「システム管理者」→「IM-BIS作成」→「IM-BIS」をクリックします。
2. 「IM-BIS - 更新履歴」画面が表示されますので、「一覧」をクリックします。



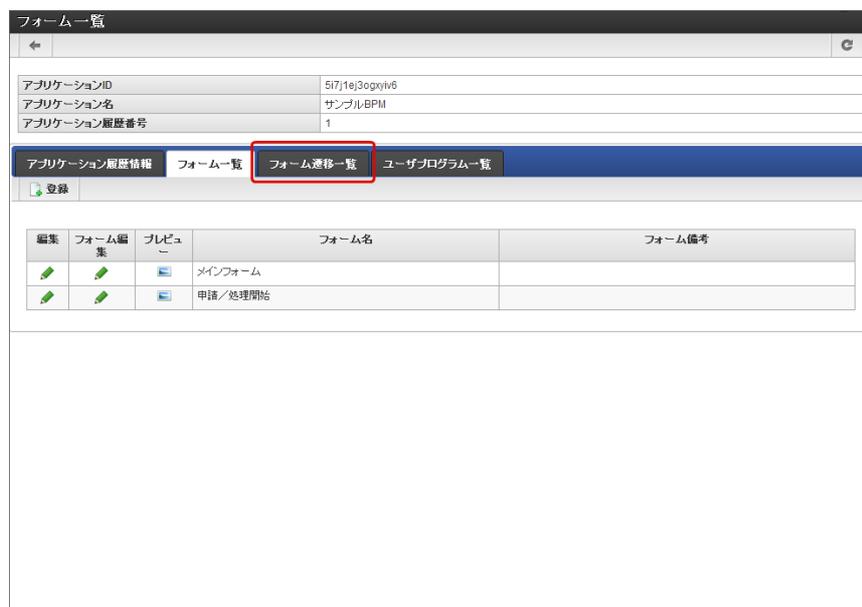
3. 対象のフロー（BIS定義）の  をクリックします。



4. フォーム設定で「アプリケーション履歴一覧」の  をクリックします。



5. 「フォーム遷移一覧」をクリックします。



6. 「フォーム遷移一覧」の「参照画面遷移」で対象のノードの「デフォルト」をクリックします。

フォーム遷移一覧

アプリケーションID: 5f71e30gwxiv6
 アプリケーション名: サンプルBPM
 アプリケーション履歴番号: 1

アプリケーション履歴情報 | フォーム一覧 | フォーム遷移一覧 | ユーザプログラム一覧

登録画面遷移

編集	詳細編集	デフォルト	フォーム遷移名	フォーム件数
			申請/処理開始	1

更新画面遷移

編集	詳細編集	デフォルト	フォーム遷移名	フォーム件数
			申請/処理開始	1

参照画面遷移 + 追加

編集	詳細編集	デフォルト	フォーム遷移名	フォーム件数
			参照画面	1
		<input type="checkbox"/>	申請/処理開始	1

7. 確認メッセージで「決定」をクリックしたら設定完了です。

更新確認

「申請/処理開始」をデフォルトのフォーム遷移に設定してよろしいですか？

注意

上記の設定後に、「フロー編集」で共有設定等の変更を行うと、設定内容が変更されてしまう場合があります。参照画面遷移の設定後に、フロー編集の設定を変更した場合には、設定内容を確認し、必要に応じて再設定するようにしてください。

参照画面や特殊なタスク（ノード）で表示する画面に別の画面を設定する

IM-BIS のフロー編集からフォームを設定できないノードに他のノードと異なる画面を設定するには、以下の手順で行います。

1. 「サイトマップ」→「IM-BIS」→「システム管理者」→「IM-BIS作成」→「IM-BIS」をクリックします。
2. 「IM-BIS - 更新履歴」画面が表示されますので、「一覧」をクリックします。

IM-BIS - 更新履歴

最近使用した定義

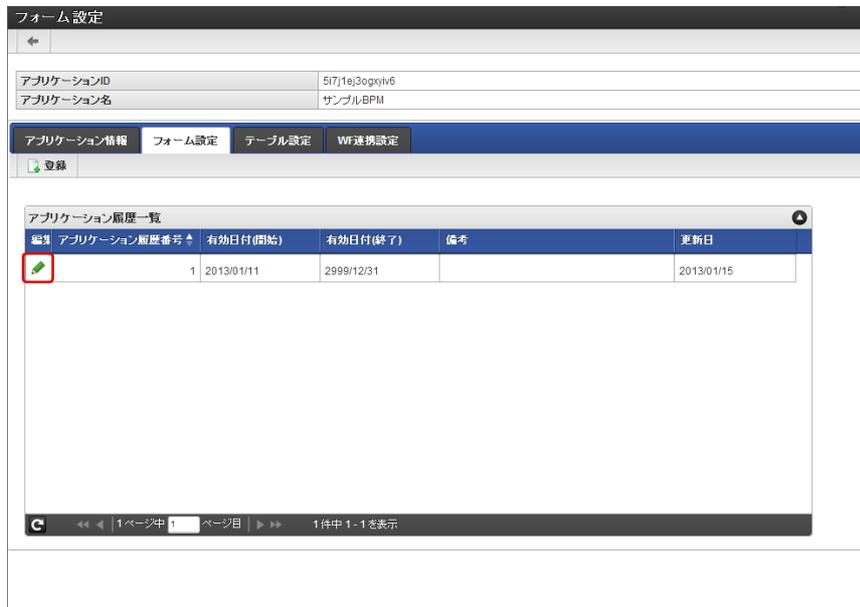
編集	BIS作成種類	BIS名	説明	更新日時
表示するレコードがありません				

0 ページ中 0 ページ目 15

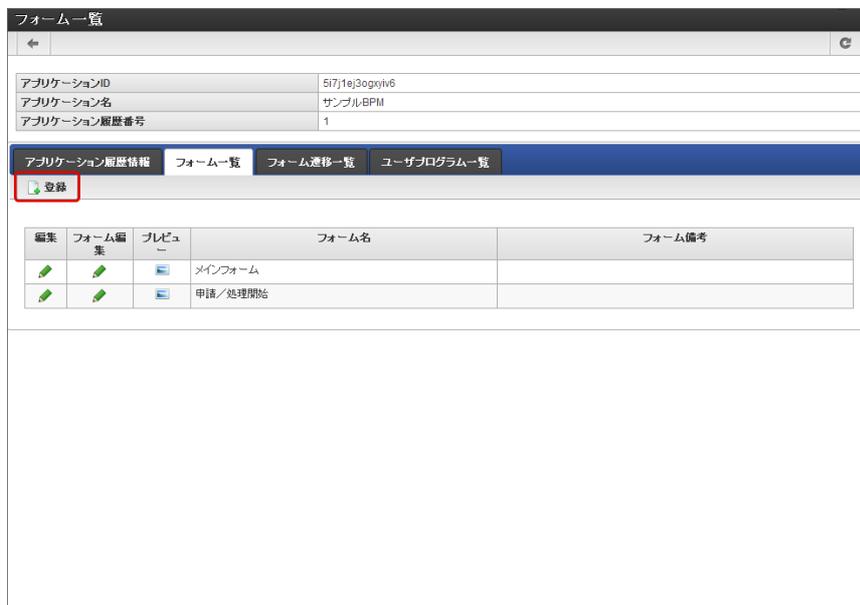
3. 対象のフロー（BIS定義）の をクリックします。



4. フォーム設定で「アプリケーション履歴一覧」の  をクリックします。



5. 「登録」をクリックします。



6. フォーム名を入力し、「登録」をクリックします。

7. アイテムコピーをクリックします。

8. フォームで表示したい画面アイテムをコピーします。

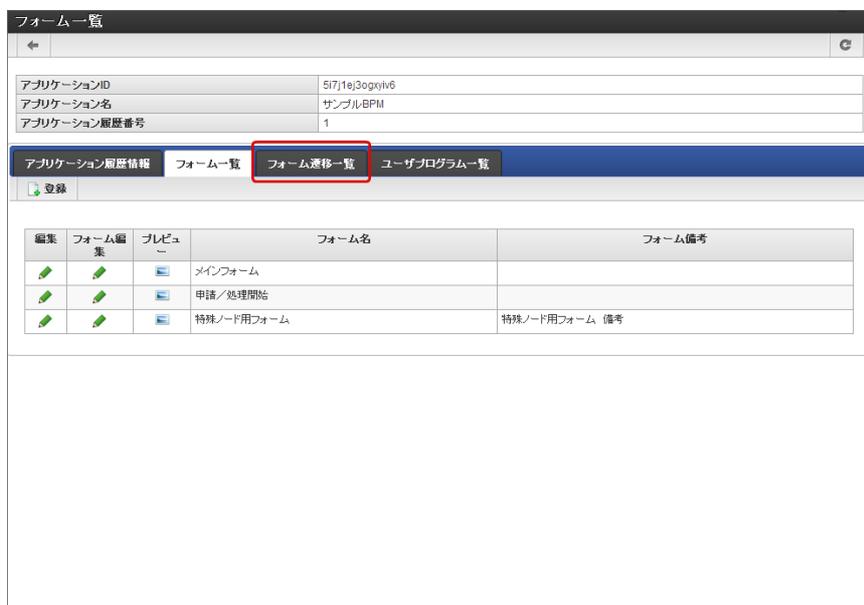
コラム

特殊なノードで入力・選択系の画面アイテム（文字列、日付など）を利用する場合は、同じフロー内の「申請／処理開始」、「承認／処理」の画面に配置されており、「フィールド値DB登録」が有効となっている必要があります。

9. 画面アイテムを配置したら「更新」をクリックします。



10. 「フォーム遷移一覧」をクリックします。



11. 「フォーム遷移一覧」の「参照画面遷移」でフォーム遷移名が「参照画面」の「デフォルト」が設定されていることを確認し、「詳細編集」をクリックします。



12. フォーム名に「メインフォーム」が設定されていますので、「削除」をクリックします。



13. 「追加」をクリックします。



14. 先の手順で作成したフォームを選択します。



15. 「更新」をクリックします。

フォーム遷移詳細編集

アプリケーションID	5171ej3oggyiv6
アプリケーション名	サンプルBPM
アプリケーション履歴番号	1
遷移種別	参照フォーム遷移
フォーム遷移名	参照画面

フォーム名	プレビュー	削除
特殊ノード用フォーム		

+ 追加

更新

16. これで、特殊なノード用に表示する画面が設定できました。

フォーム遷移詳細編集

アプリケーションID	5171ej3oggyiv6
アプリケーション名	サンプルBPM
アプリケーション履歴番号	1
遷移種別	参照フォーム遷移
フォーム遷移名	参照画面

フォーム名	プレビュー	削除
特殊ノード用フォーム		

+ 追加

更新

処理中の画面から別の画面を子画面として表示するための設定を行う

処理中（申請や承認）の画面から別の画面を子画面としてポップアップで表示するための設定は、以下の手順で行います。

子画面として呼び出す画面（フォーム）の追加、フォーム遷移設定を行う

1. 「BIS一覧」を表示します。

IM-BIS 一覧

最近使用した定義 新規登録

BPM WF
 BIS名 検索

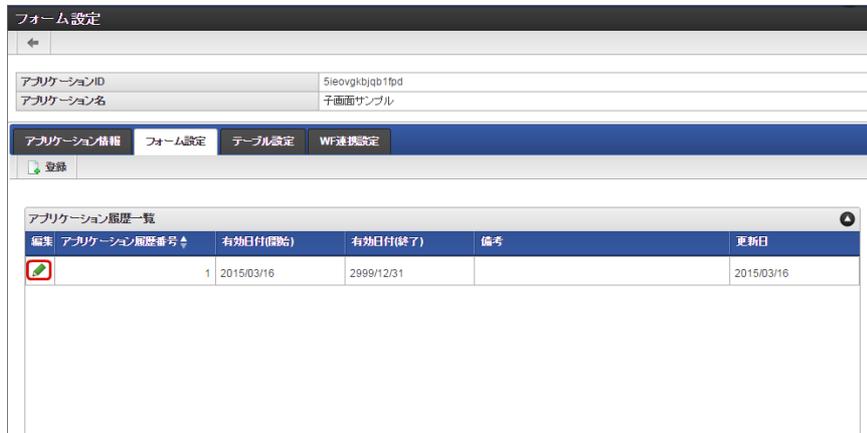
選択した定義を削除

編集	BIS作成種別	BIS名	説明	BIS ID	フローID	アプリ
<input type="checkbox"/>	WF	子画面サンプル		51eovgkb6stx0pd	51eovgkbqb1pd	

2. 子画面を呼び出す対象のフローの「アプリ」をクリックします。



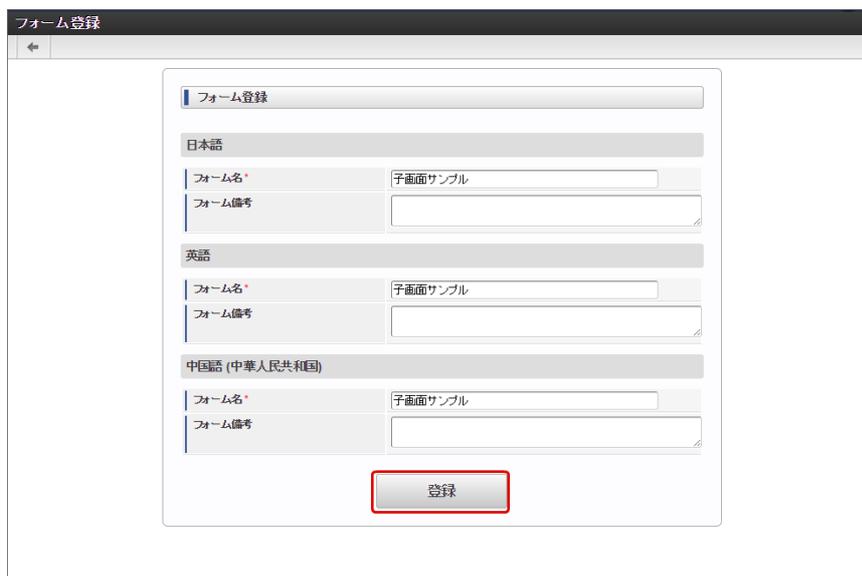
3. 「アプリケーション履歴一覧」の「編集」をクリックします。



4. 「フォーム一覧」の「登録」をクリックして、子画面として呼び出す画面を登録します。



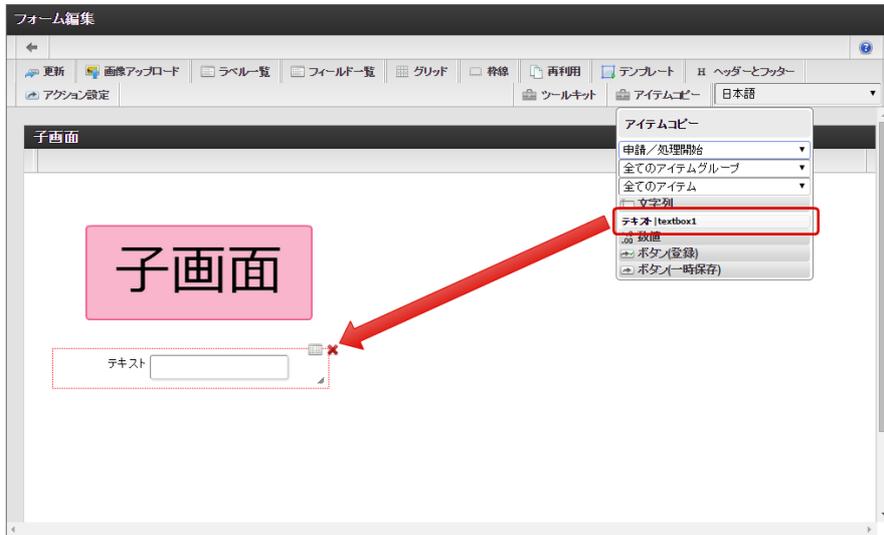
5. 「フォーム登録」画面が表示されますので、必要な項目を入力して「登録」をクリックします。



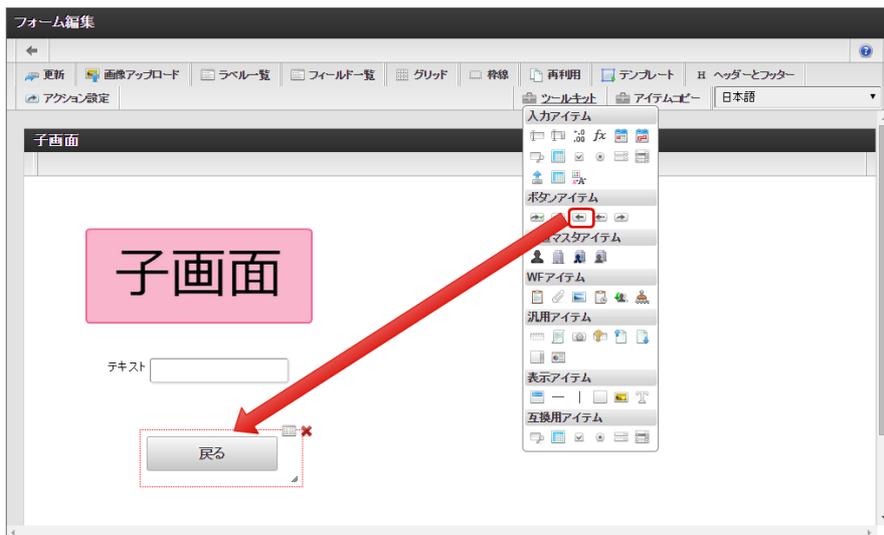
6. 子画面のレイアウトを設定し、「更新」をクリックしてフォームを保存します。



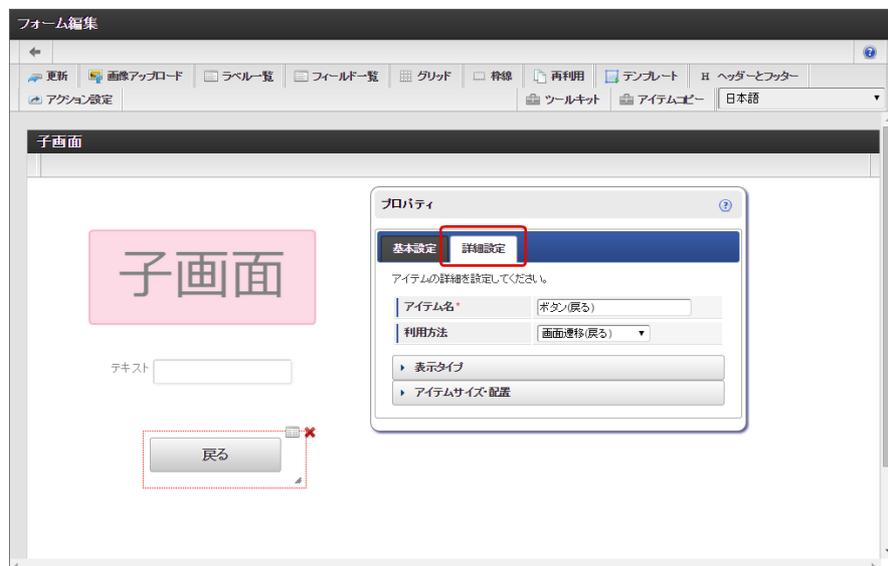
7. 呼び出し元画面と値をやり取りしたい場合には、アイテムコピーを利用して、フィールド識別IDが同じアイテムを配置します。



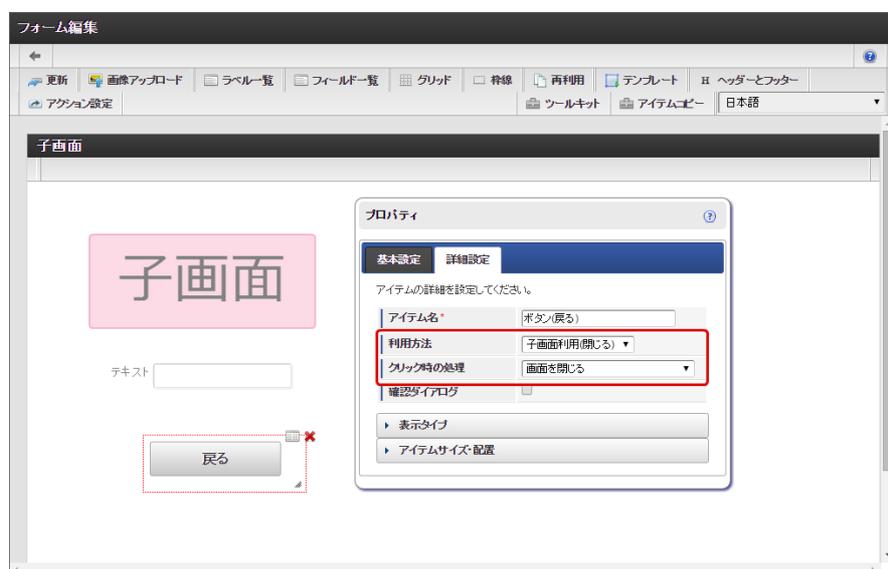
8. 呼び出した子画面は、「戻る」ボタンを利用して閉じることができます。子画面を閉じるためのボタンとして、「戻る」ボタンを配置します。



9. 配置した「戻る」ボタンのプロパティの「詳細設定」を表示します。



10. 利用方法を「子画面利用（閉じる）」に変更します。
子画面で入力した内容呼び出し元の画面に返却しない場合は、「クリック時の処理」を「画面を閉じる」にします。



11. 必要な設定が終わったら「更新」をクリックして、フォームを保存後、「戻る」で前の画面に戻ります。



12. 「フォーム遷移一覧」をクリックします。

フォーム一覧

アプリケーションID: 5ieovgkjbq1fpd
 アプリケーション名: 子画面サンプル
 アプリケーション履歴番号: 1

アプリケーション履歴情報 | フォーム一覧 | **フォーム遷移一覧** | ユーザプログラム一覧

登録

編集	フォーム編集	プレビュー	フォーム名	フォーム備考
			子画面サンプルフォーム	
			申請/処理開始	
			子画面サンプル	

13. 「参照画面遷移」の「追加」をクリックします。

フォーム遷移一覧

アプリケーションID: 5ieovgkjbq1fpd
 アプリケーション名: 子画面サンプル
 アプリケーション履歴番号: 1

アプリケーション履歴情報 | フォーム一覧 | **フォーム遷移一覧** | ユーザプログラム一覧

登録画面遷移

編集	詳細編集	デフォルト	フォーム遷移名	フォーム件数
			申請/処理開始	1

更新画面遷移

編集	詳細編集	デフォルト	フォーム遷移名	フォーム件数
			申請/処理開始	1

参照画面遷移 **+ 追加**

編集	詳細編集	デフォルト	フォーム遷移名	フォーム件数
			参照画面	1
			申請/処理開始	1

14. 「フォーム遷移名」を入力し、「登録」をクリックします。

フォーム遷移登録

アプリケーションID: 5ieovgkjbq1fpd
 アプリケーション名: 子画面サンプル
 アプリケーション履歴番号: 1
 遷移種別: 参照フォーム遷移

日本語

フォーム遷移名*

英語

フォーム遷移名*

中国語 (中華人民共和国)

フォーム遷移名*

登録

15. 「遷移方法」を「画面遷移」であることを確認し、「追加」をクリックします。



16. 「フォーム検索」から先ほど作成した子画面のフォームを選択します。



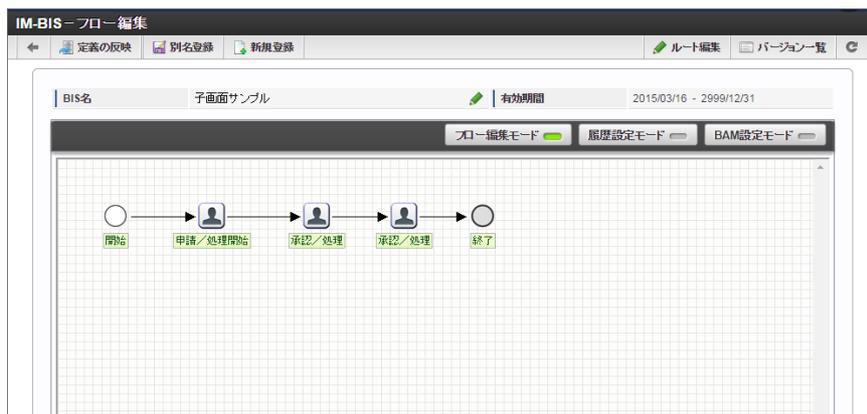
17. 「更新」をクリックして、フォーム遷移を保存します。



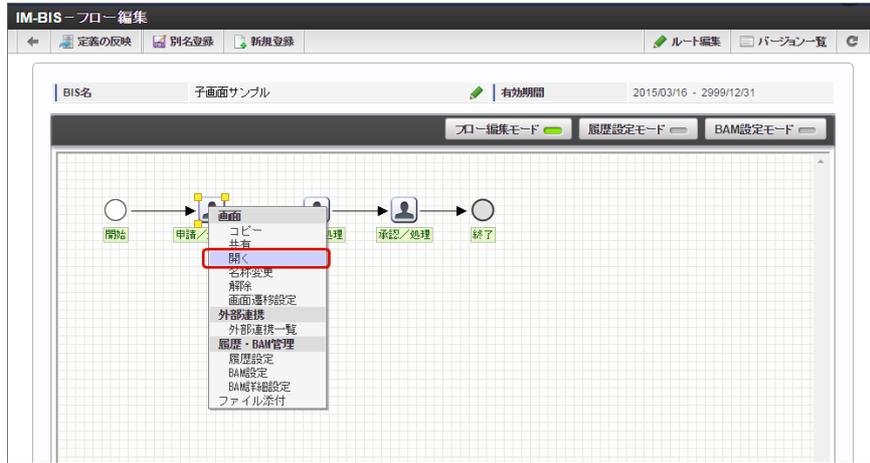
18. 以上で、呼び出す対象の子画面とフォーム遷移を設定できました。
続いて、呼び出し元の画面の設定を行います。

呼び出し元画面の設定を行う

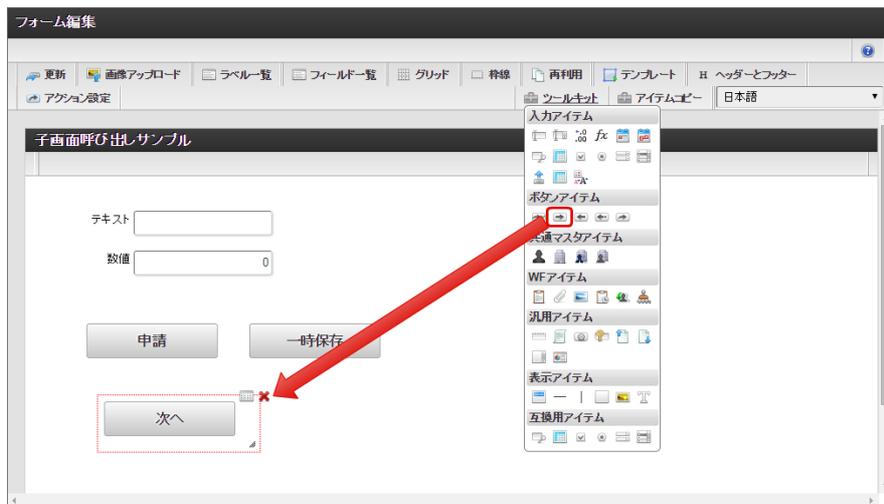
1. 編集中のフローの「フロー編集」画面を表示します。



2. 子画面を呼び出す画面を設定したいタスク（ノード）のフォーム・デザイナーを表示します。



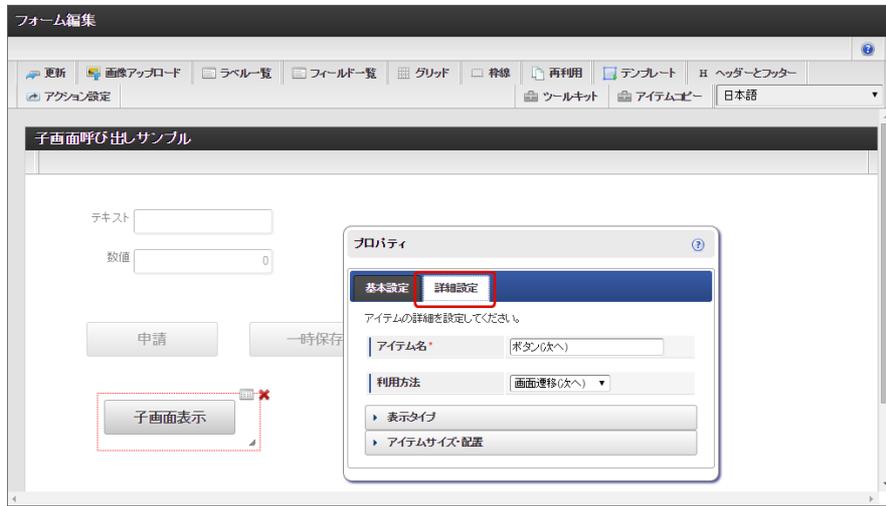
3. 対象の画面（フォーム）に「次へ」ボタンを配置します。



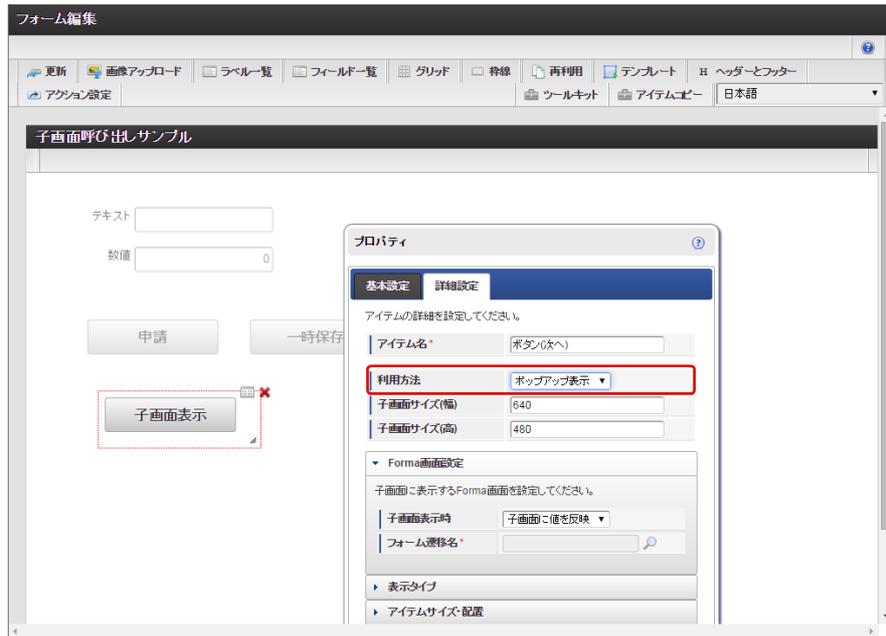
4. 配置した「次へ」ボタンのプロパティを表示し、ラベルを「子画面表示」に変更します。



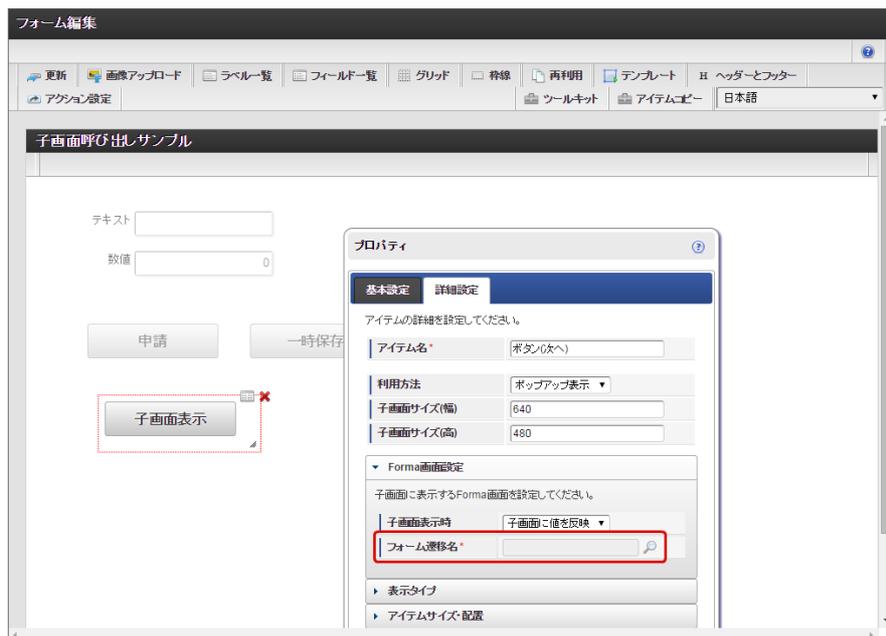
5. 「次へ」ボタンのプロパティの「詳細設定」を表示します。



6. 「利用方法」を「ポップアップ表示」に変更します。



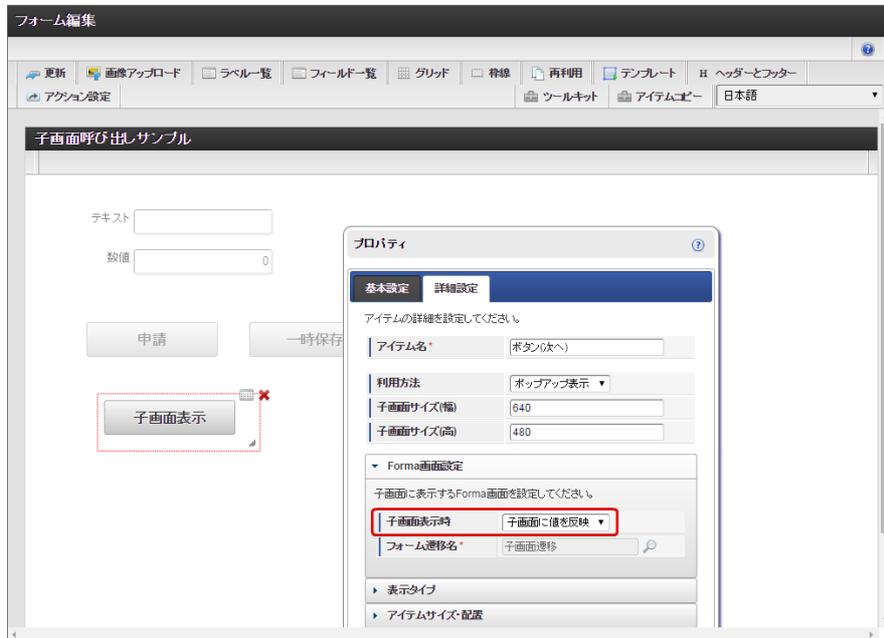
7. 「Form画面設定」の「フォーム遷移名」の 🔍 をクリックします。



8. 先の手順で追加したフォーム遷移をクリックします。



9. 子画面に配置したアイテムに、呼び出し元画面で入力した内容を反映したい場合には、「子画面表示時」を「子画面に値を反映」にします。



10. 「更新」をクリックしてフォームを保存します。



11. 以上で、子画面を呼び出すための設定ができました。
フローの実行に必要な設定を行って、実際に画面からボタンをクリックすると、以下のように子画面を呼び出すことができます。



メールや IMBox を設定する

ここでは、メールや IMBox の設定方法について説明します。

IM-BIS で作成したフローでメールや IMBox による通知を利用する場合には、BISシステム管理者がメールや IMBox を設定する必要があります。

Contents

- メールや IMBox の利用に必要な設定
- IM-BIS のフローでメール通知を設定する
- IM-BIS のフローで IMBox 通知を設定する

メールや IMBox の利用に必要な設定

IM-Workflow、IM-BIS のフローでメールや IMBox の通知を行う場合には、セットアップ時に「ベースURL」を設定しておく必要があります。設定方法については、「[設定ファイルリファレンス](#)」-「[コアモジュール-サーバコンテキスト設定](#)」を参照してください。

IM-BIS のフローでメール通知を設定する

IM-BIS のフローの処理時にメールを送信するための設定を行います。

利用するメール種別の詳細については、「[IM-Workflow 仕様書](#)」を参照してください。

i コラム

IM-BIS が導入されている環境で、メール・IMBox 定義内のショートカットURLから遷移した場合の「戻る（ボタン・リンク）」「一覧に戻る（ボタン）」の遷移先は以下の通りです。

※「戻る」については、ヘッダに設定されるヘッダのアイコンのリンクを含んでいます。

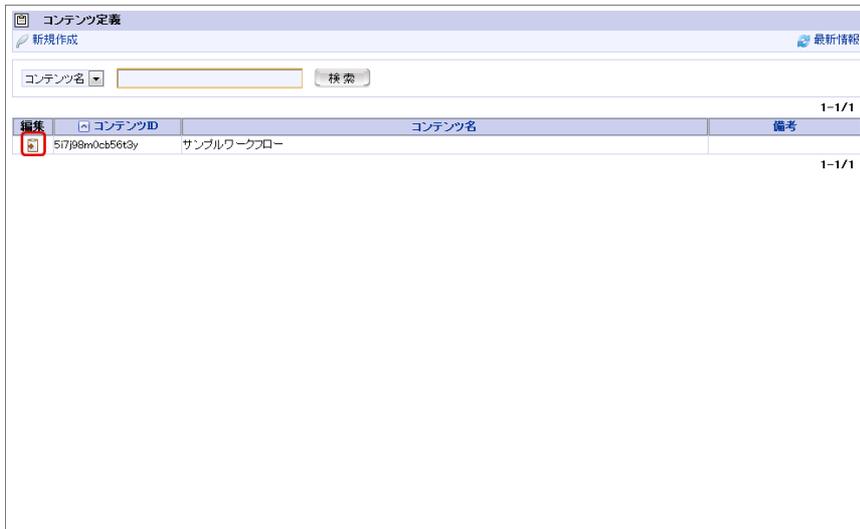
- メール / IMBox 種別「**処理通知依頼**」の場合
「戻る（ボタン・リンク）」「一覧に戻る（ボタン）」のどちらも「未処理（ワークフロー）」、または「未処理（BISフロー）」に遷移します。
遷移先の一覧の種類（ワークフロー・BISフロー）は、BIS作成種類に基づいて決定されます。
また、IM-BIS 以外で作成したワークフローの場合は「未処理（ワークフロー）」に遷移します。
- メール / IMBox 種別「**参照依頼**」の場合
「戻る（ボタン・リンク）」「一覧に戻る（ボタン）」のどちらも「参照（ワークフロー）」、または「参照（BISフロー）」に遷移します。
遷移先の一覧の種類（ワークフロー・BISフロー）は、BIS作成種類に基づいて決定されます。
また、IM-BIS 以外で作成したワークフローの場合は「参照（ワークフロー）」に遷移します。
- メール / IMBox 種別「**処理結果通知**」の場合
「戻る（ボタン・リンク）」「一覧に戻る（ボタン）」は利用できません。

処理依頼、処理結果通知、参照依頼、確認依頼メールを設定する

フローの処理時にメールの通知を利用できるように、コンテンツ定義にメールの設定を行います。

1. 「サイトマップ」→「ワークフロー」→「ワークフロー管理者」→「マスタ定義」→「コンテンツ定義」をクリックします。

2. 設定する対象のBIS名のコンテンツ定義の  をクリックします。



コンテンツ定義

新規作成 最新情報

コンテンツ名 検索

編集	コンテンツID	コンテンツ名	備考
	51798m0cb5613y	サンプルワークフロー	

1-1/1

3. 「バージョン」をクリックします。



コンテンツ定義 - 編集

[戻る](#) | [利用状況確認](#)

基本情報 **バージョン**

コンテンツID

コンテンツ名 (必須)

英語	<input type="text" value="Sample Workflow"/>
日本語	<input type="text" value="サンプルワークフロー"/>
中国語	<input type="text" value="Sample Workflow"/>

備考

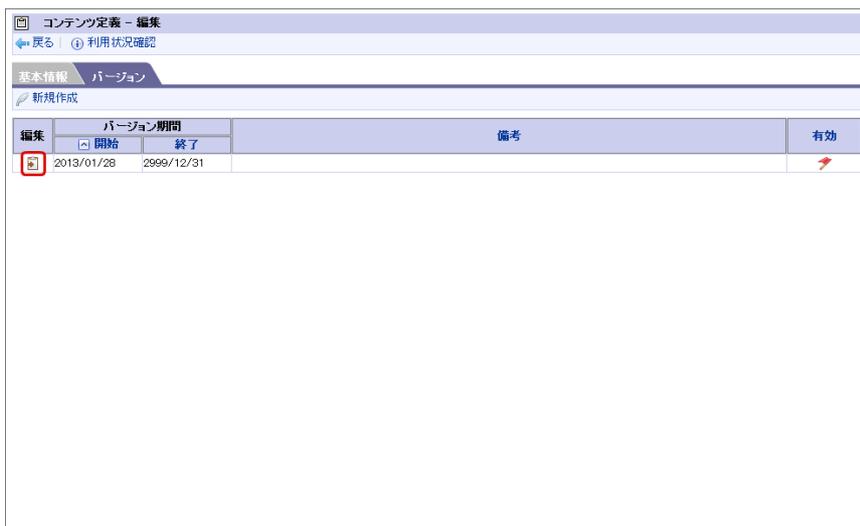
英語	<input type="text"/>
日本語	<input type="text"/>
中国語	<input type="text"/>

管理グループ 検索

管理グループ名 編集権限 クリア

更新 削除

4. 対象のバージョンの  をクリックします。



コンテンツ定義 - 編集

[戻る](#) | [利用状況確認](#)

基本情報 **バージョン**

新規作成

編集	バージョン期間		備考	有効
	開始	終了		
	2013/01/26	2999/12/31		

5. 「メール」をクリックします。

コンテンツ定義 - バージョン - 編集

戻る | コンテンツコピー | バージョンコピー

コンテンツID: 5i7j98m0cb56i3y

コンテンツ名: サンプルワークフロー

基本情報 | 画面 | ユーザプログラム | **メール** | IMBox | ルール

バージョン期間 (必須): 2013/01/28 から 2999/12/31 まで

バージョン有効/無効 (必須): 有効 無効

備考: 英語, 日本語, 中国語

更新 削除

6. 「新規作成」をクリックします。

コンテンツ定義 - バージョン - 編集

戻る | コンテンツコピー | バージョンコピー

コンテンツID: 5i7j98m0cb56i3y

コンテンツ名: サンプルワークフロー

基本情報 | 画面 | ユーザプログラム | **メール** | IMBox | ルール

新規作成

編集	メール名	メール種別	標準	初期使用
----	------	-------	----	------

7. 「メール定義 - 新規作成」画面で「選択」をクリックします。

メール定義 - 新規作成

閉じる

メール名 (必須): **選択**

メール種別:

標準:

初期使用: フローの初期設定で使用する

登録

8. 設定対象のメール定義の  をクリックします。

選択	メールID	メール名	メール種別	備考
<input type="checkbox"/>	confirm	確認依頼	確認依頼	確認依頼用の標準メールテンプレート
<input type="checkbox"/>	processing	処理依頼	処理依頼	処理依頼用の標準メールテンプレート
<input type="checkbox"/>	reference	参照依頼	参照依頼	参照依頼用の標準メールテンプレート
<input type="checkbox"/>	result	処理結果通知	処理結果通知	処理結果通知用の標準メールテンプレート

9. 「登録」をクリックして内容を保存します。

メール名 (必須)	処理依頼	選択
メール種別	処理依頼	
標準	システム標準	
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する	

10. 6～9の手順を繰り返し、必要なメール定義の設定を行います。



コラム

処理依頼メール定義について、特定のノード（タスク）時では送信する・しないなどの詳細設定を行う場合は、フロー定義の「ルート詳細」から設定してください。

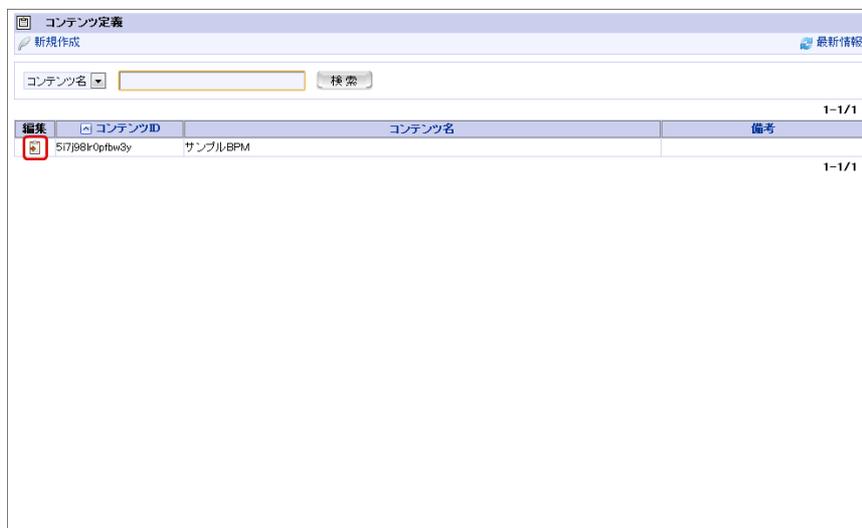
IM-BIS のフローで IMBox 通知を設定する

IM-BIS のフローの処理時に IMBox で通知を送信するための設定を行います。
 利用する IMBox 種別の詳細については、「[IM-Workflow 仕様書](#)」を参照してください。

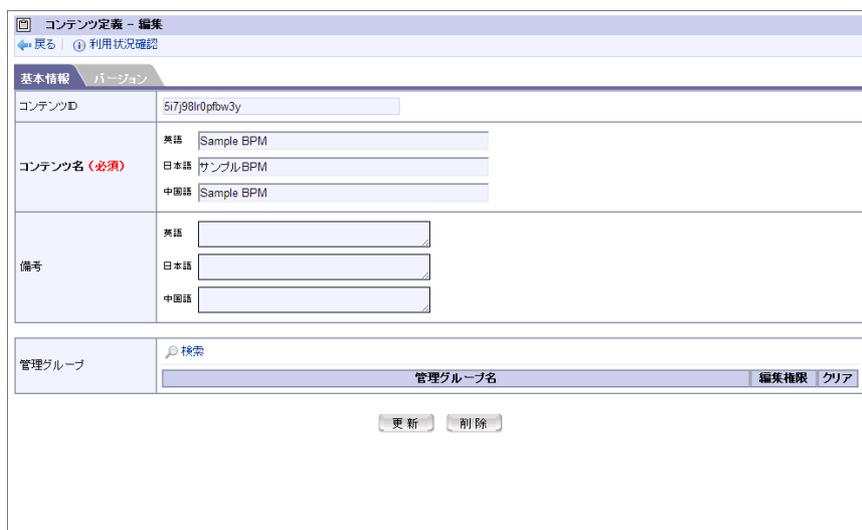
処理依頼、処理結果通知、参照依頼、確認依頼 IMBox を設定する

フローの処理時に IMBox の通知を利用できるように、コンテンツ定義に IMBox の設定を行います。

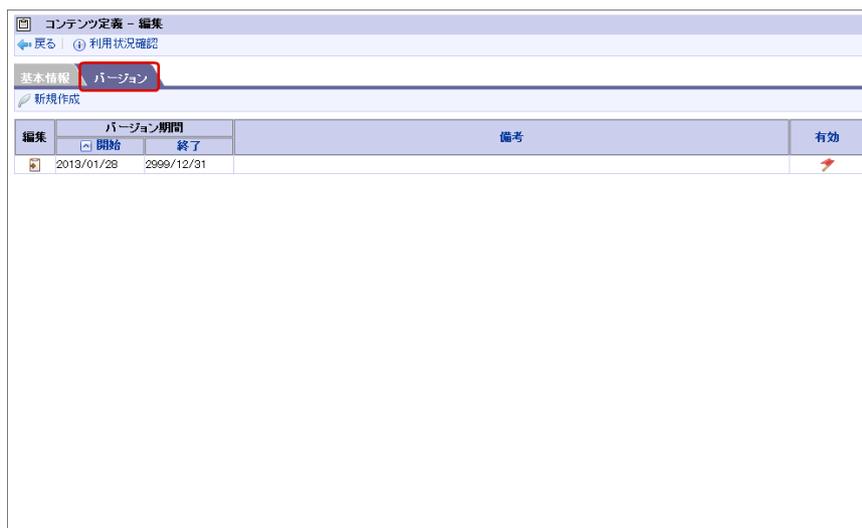
1. 「サイトマップ」→「ワークフロー」→「ワークフロー管理者」→「マスタ定義」→「コンテンツ定義」をクリックします。
2. 設定する対象のBIS名のコンテンツ定義の  をクリックします。



3. 「バージョン」をクリックします。



4. 対象のバージョンの をクリックします。



5. 「IMBox」をクリックします。

コンテンツ定義 - バージョン - 編集

戻る | コンテンツコピー | バージョンコピー

コンテンツID: 5i7j98lr0ptbw3y

コンテンツ名: サンプルBPM

基本情報 画面 ユーザプログラム メール **IMBox** ルール

バージョン期間 (必須) 2013/01/28 から 2999/12/31 まで

バージョン有効/無効 (必須) 有効 無効

備考

英語

日本語

中国語

更新 削除

6. 「新規作成」をクリックします。

コンテンツ定義 - バージョン - 編集

戻る | コンテンツコピー | バージョンコピー

コンテンツID: 5i7j98lr0ptbw3y

コンテンツ名: サンプルBPM

基本情報 画面 ユーザプログラム メール IMBox ルール

新規作成

編集	IMBox名	IMBox種別	標準	初期使用

7. 「IMBox 定義 - 編集」画面で「選択」をクリックします。

IMBox定義 - 編集

閉じる

IMBox名 (必須) **選択**

IMBox種別

標準

初期使用 フローの初期設定で使用する

登録

8. 設定対象の IMBox 定義の  をクリックします。

選択	IMBoxID	IMBox名	IMBox種別	備考
<input checked="" type="checkbox"/>	confirm	確認依頼	確認依頼	確認依頼用の標準IMBoxテンプレート
<input type="checkbox"/>	processing	処理依頼	処理依頼	処理依頼用の標準IMBoxテンプレート
<input type="checkbox"/>	reference	参照依頼	参照依頼	参照依頼用の標準IMBoxテンプレート
<input type="checkbox"/>	result	処理結果通知	処理結果通知	処理結果通知用の標準IMBoxテンプレート

9. 「登録」をクリックして内容を保存します。

IMBox名 (必須)	確認依頼	選択
IMBox種別	確認依頼	
標準	システム標準	
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する	

登録

10. 6～9の手順を繰り返し、必要な IMBox 定義の設定を行います。



コラム

処理依頼 IMBox 定義について、特定のノード（タスク）時では送信する・しないなどの詳細設定を行う場合は、フロー定義の「ルート詳細」から設定してください。

申請者承認防止機能を利用する

ここでは、IM-BIS のワークフローにおいて、ユーザプログラム「申請者承認防止処理」を利用するための方法を説明します。

Contents

- ユーザプログラム「申請者承認防止処理」とは
- IM-BIS で作成したコンテンツへ「申請者承認防止処理」を設定する

ユーザプログラム「申請者承認防止処理」とは

「申請者承認防止処理」は、ワークフローにおいて、案件の申請者が当該案件に対して承認 / 承認終了 / 否認 / 保留を行うことを防止するユーザプログラムです。

詳細は、「IM-Workflow 仕様書」- 「申請者承認防止処理」を参照してください。



コラム

この機能は、intra-mart Accel Platform 2016 Summer(Nirvana) より追加されました。

IM-BIS で作成したコンテンツへ「申請者承認防止処理」を設定する

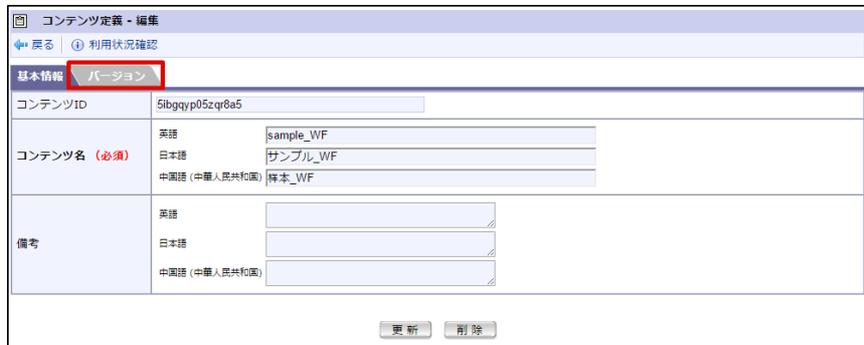
IM-BIS で作成したフォームに対応するコンテンツ定義に申請者承認防止処理を設定するためには、以下の手順で設定してください。

1. 「サイトマップ」→「ワークフロー」→「ワークフロー管理者」→「マスタ定義」→「コンテンツ定義」をクリックします。

2. コンテンツ名が設定対象の「BIS名」となっているコンテンツ定義の「編集」をクリックします。



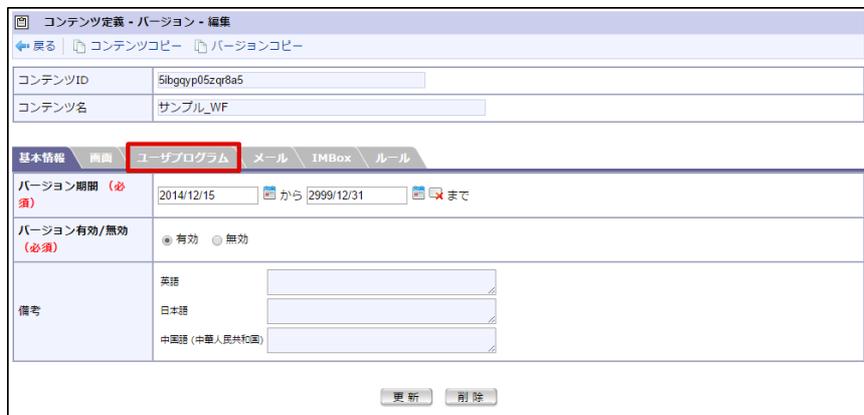
3. 「バージョン」をクリックします。



4. 対象のバージョンの「編集」をクリックします。



5. 「ユーザプログラム」をクリックします。



6. 「新規作成」をクリックします。



承認/処理ノードのタスクへ申請者承認防止処理を設定する

対象のコンテンツ定義を利用するBISフロー/ワークフローのすべてのタスクに以下の手順で申請者承認防止処理を設定します。

1. 「ユーザプログラム定義 - 新規作成」画面で、以下の内容に従って、必要なユーザプログラムを設定します。

- プログラム名：「【アクション処理】申請者承認防止処理」
- プラグイン種別：アクション処理
- 対象ノード：承認／処理ノード
- プラグイン種類：「【アクション処理】申請者承認防止処理」
- 実行順番：0

ユーザプログラム定義 - 新規作成		閉じる
プログラム名(必須)	英語 【Action process】Prevention Of Approval By Applicant Prc 日本語 【アクション処理】申請者承認防止処理 中国語(中華人民共和国) 【动作处理】申请人审批预防处理	
プラグイン種別(必須)	アクション処理	
対象ノード(必須)	承認／処理ノード	
プラグイン種類(必須)	【アクション処理】申請者承認防止処理	
備考	英語 日本語 中国語(中華人民共和国)	
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する	
実行順番(必須)	0	
登録		

コラム

承認／処理ノードにおいて、当ユーザプログラムが最初に行われるよう実行順番を設定することを推奨します。

2. 上記の通りに設定したら、「登録」をクリックします。

ユーザプログラム定義 - 新規作成		閉じる
プログラム名(必須)	英語 【Action process】Prevention Of Approval By Applicant Prc 日本語 【アクション処理】申請者承認防止処理 中国語(中華人民共和国) 【动作处理】申请人审批预防处理	
プラグイン種別(必須)	アクション処理	
対象ノード(必須)	承認／処理ノード	
プラグイン種類(必須)	【アクション処理】申請者承認防止処理	
備考	英語 日本語 中国語(中華人民共和国)	
初期使用	<input checked="" type="checkbox"/> フローの初期設定で使用する	
実行順番(必須)	0	
登録		

以上で、タスクに対する申請者承認防止処理が設定できました。

BIS作成種類「BISフロー」の利用可否を切り替えるための設定をする

IM-BPM のリリースに伴い、簡易BPMであるBIS作成種類「BISフロー」を利用しないケースが想定されます。

そのため、IM-BIS 2016 Summer (8.0.11) 以降、BIS作成種類「BISフロー」を利用しない場合、BPMに関連するリンクや画面のボタンを非表示にすることができます。

また、名称が重複するため、BIS作成種類「BPM」を「BISフロー」に変更しています。

ここでは、BIS作成種類「BISフロー」関連のメニューやボタンの表示を切り替えるための設定方法について説明します。

Contents

- BIS作成種類「BISフロー」の利用可否の設定
- BIS作成種類「BISフロー」を利用不可にする場合の設定
- BIS作成種類「BISフロー」を利用可能にする場合の設定

BIS作成種類「BISフロー」の利用可否の設定では、設定内容に応じて以下の表示が変わります。

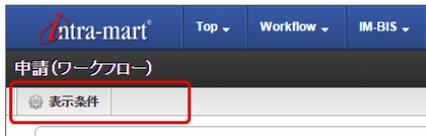
一覧画面の「BISフロー」切替ボタンの表示

- 一覧画面のヘッダにある「BISフロー」の切替リンクの表示・非表示が変わります。利用不可の場合には該当のボタンは表示されなくなります。

- 利用可能に設定している場合



- 利用不可に設定している場合



- 上記の表示が切り替わる一覧画面は以下の通りです。下記の一覧画面は、製品標準で「IM-BIS」のメニューグループカテゴリに含まれるメニューの一覧画面を指します。
 - 申請（ワークフロー）一覧
 - 未処理（ワークフロー）一覧
 - 処理済（ワークフロー）一覧
 - 参照（ワークフロー）一覧

BISの新規登録画面におけるBIS作成種類の選択内容

- IM-BIS でのフローの新規登録画面の「BIS作成種類」に表示される内容が変わります。利用不可の場合には「BISフロー」の選択肢は表示されなくなります。

- 利用可能に設定している場合



- 利用不可に設定している場合



BIS作成種類「BISフロー」を利用不可にする場合の設定

BIS作成種類「BISフロー」を利用不可にするには、以下の手順で設定してください。

注意

BIS作成種類「BISフロー」関連のメニュー・機能を無効化した場合には、一覧画面では「BIS作成種類」を判別しません。そのため、一覧画面やAPIではBIS作成種類を考慮せずにフローや案件を取得します。

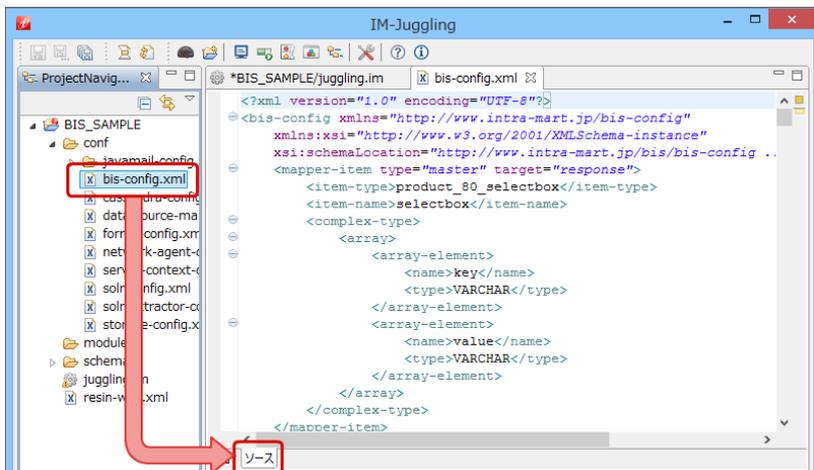
したがって、BISフローの利用不可の設定後に、BISフローのフローや案件が存在する場合には、ワークフローの一覧にも表示されますので、注意してください。

bis-configの設定

IM-BIS のフロー編集画面等で「BISフロー」に関する機能を無効にするには、「bis-config.xml」で設定します。

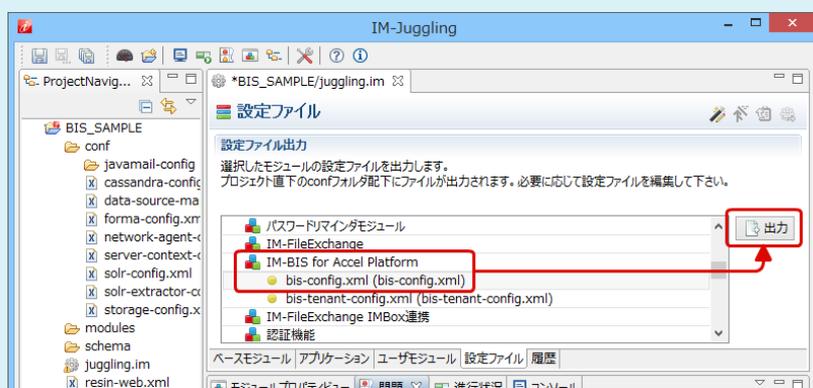
bis-configの設定を変更するには、以下の手順で IM-Juggling から設定ファイルを編集後、WARファイルの再デプロイを実施してください。

- IM-Juggling から対象のJugglingプロジェクトを表示します。
- 「ProjectNavigator」内の < (プロジェクト名) /conf/bis-config.xml> ファイルをダブルクリックで開き、「ソース」タブを選択してください。

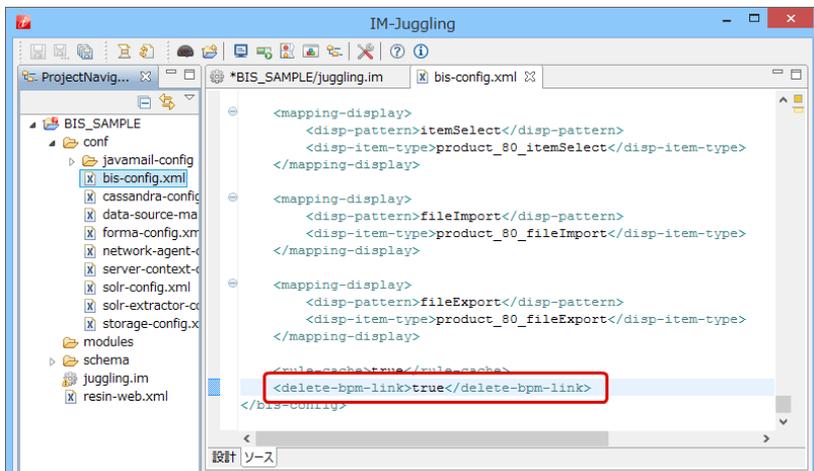


i コラム

対象のJugglingプロジェクトに「bis-config.xml」が表示されていない場合には、juggling.imを開き、「設定ファイル」タブに切り替え、「IM-BIS for Accel Platform」-「bis-config.xml」を出力してください。



3. bis-config.xmlの下に<delete-bpm-link>タグを追加し、値に“true”を設定してください。



```
<?xml version="1.0" encoding="UTF-8"?>
<bis-config xmlns="http://www.intra-mart.jp/bis-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/bis/bis-config ../schema/bis-config.xsd">
... (略) ...
<rule-cache>true</rule-cache>
<delete-bpm-link>true</delete-bpm-link>
</bis-config>
```

4. bis-config.xmlの変更内容を保存後、WARファイルを出力してください。

メニューの設定

bis-configの設定後、テナント管理者がBISフロー関連のメニューを削除します。

1. 「サイトマップ」→「テナント管理」→「メニュー」をクリックします。
2. メニューグループカテゴリごとに関連するメニューを削除します。



コラム

「メニュー設定」の操作は、以下のドキュメントを参照してください。

- 「テナント管理者操作ガイド」- 「メニューを設定する」

■ グローバルナビ (PC用)

削除対象のメニューフォルダ

BISフロー

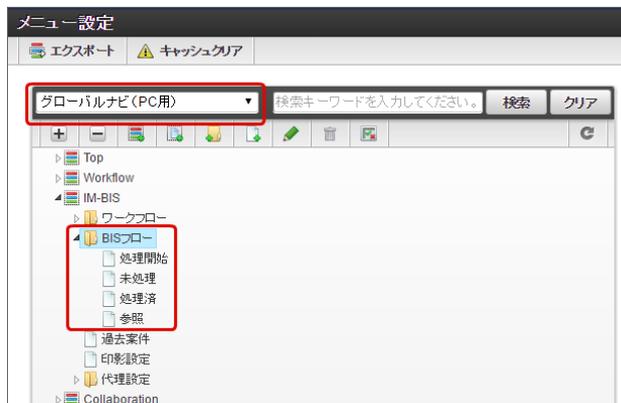
削除対象のメニューアイテム

処理開始

未処理

処理済

参照



■ グローバルナビ (スマートフォン用)

削除対象のメニューグループ

IM-BIS BISフロー

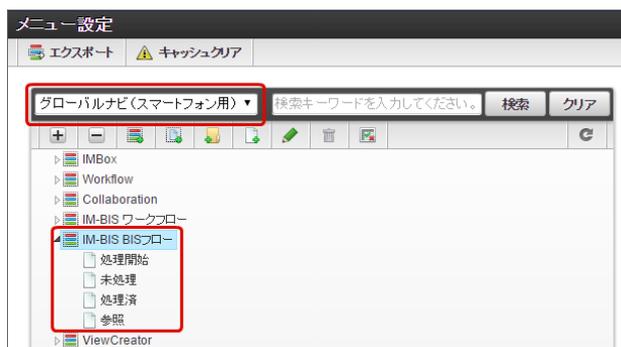
削除対象のメニューアイテム

処理開始

未処理

処理済

参照



■ サイトマップ (PC用)

削除対象のメニューフォルダ

BISフロー

削除対象のメニューアイテム

参照 (BISフロー) (フローフォルダ)

一括処理対象者変更 (BISフロー) (フローフォルダ)

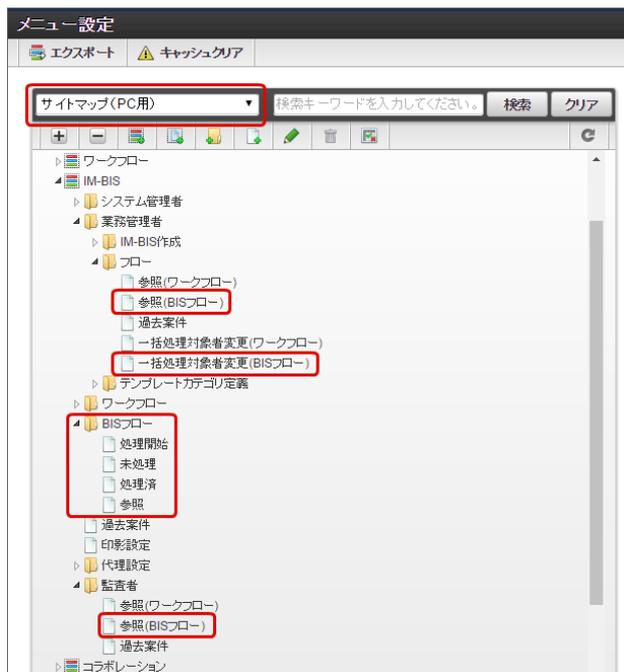
処理開始 (BISフローフォルダ)

未処理 (BISフローフォルダ)

処理済 (BISフローフォルダ)

参照 (BISフローフォルダ)

参照 (BISフロー) (監査者フォルダ)



■ サイトマップ (スマートフォン用)

削除対象のメニューフォルダ

IM-BIS BISフロー

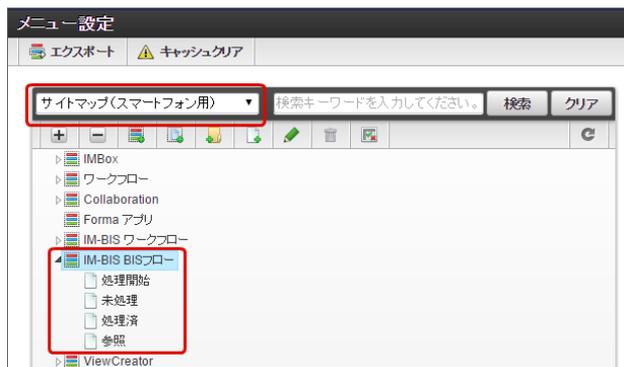
削除対象のメニューアイテム

処理開始

未処理

処理済

参照



3. これでBISフロー関連のメニューを利用不可に設定することができました。

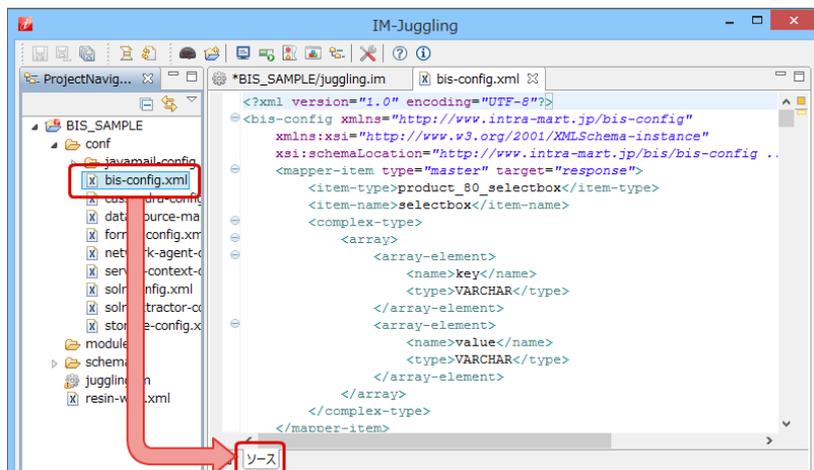
BIS作成種類「BISフロー」を利用可能にするには、以下の手順で設定してください。

bis-configの設定

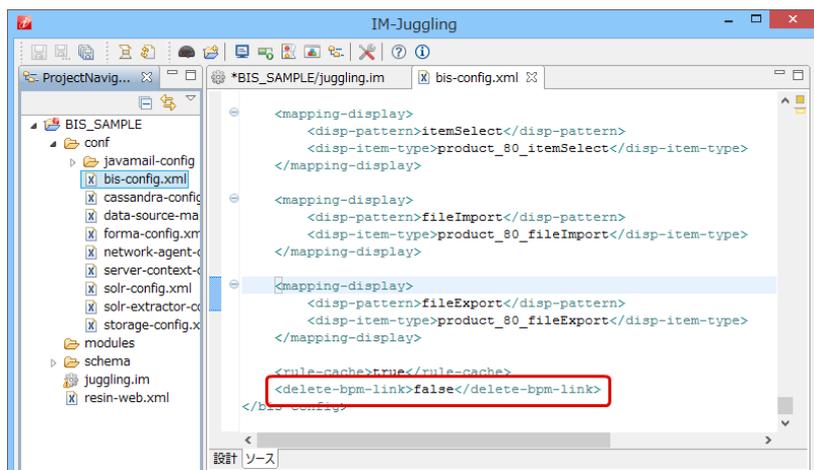
IM-BISのフロー編集画面等で「BISフロー」に関する機能を有効にするには、「bis-config.xml」で設定します。

bis-configの設定を変更するには、以下の手順で IM-Juggling から設定ファイルを編集後、WARファイルの再デプロイを実施してください。

1. IM-Juggling から対象のJugglingプロジェクトを表示します。
2. 「ProjectNavigator」内の < (プロジェクト名) /conf/bis-config.xml > ファイルをダブルクリックで開き、「ソース」タブを選択してください。



3. bis-config.xmlの下に<delete-bpm-link>タグの値に"false"を設定してください。



```
<?xml version="1.0" encoding="UTF-8"?>
<bis-config xmlns="http://www.intra-mart.jp/bis-config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.intra-mart.jp/bis/bis-config ../schema/bis-config.xsd">
... (略) ...
<rule-cache>true</rule-cache>
<delete-bpm-link>false</delete-bpm-link>
</bis-config>
```

4. bis-config.xmlの変更内容を保存後、WARファイルを出力してください。

メニューの設定

bis-configの設定後、テナント管理者がBISフロー関連のメニューを登録します。

1. 「サイトマップ」→「テナント管理」→「メニュー」をクリックします。
2. BISフロー関連のメニューを利用可能とするメニューグループカテゴリに対し、以下の情報を参考にメニューグループやメニューアイテムを登録してください。
製品標準では、以下の情報をメニューに登録しています。

 コラム

「メニュー設定」の操作は、以下のドキュメントを参照してください。

- 「テナント管理者操作ガイド」- 「メニューを設定する」

- グローバルナビ (PC用)

- メニューフォルダ : BISフロー

設定項目	設定値
メニューフォルダID	imbis_bf_gn
メニューフォルダ名 (日本語)	BISフロー
メニューフォルダ名 (英語)	BIS flow
メニューフォルダ名 (中国語 (中華人民共和国))	BIS flow
アイコン画像	ファイルパス : (ブランク)

- メニューアイテム : 処理開始

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / BISフロー
メニューアイテムID	imbis_bf_start_process_gn
メニューアイテム名 (日本語)	処理開始
メニューアイテム名 (英語)	Start Process
メニューアイテム名 (中国語 (中華人民共和国))	开始处理
URL	bis/user/businessflow/apply/apply_list_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス : (ブランク)
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- メニューアイテム : 未処理

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / BISフロー
メニューアイテムID	imbis_bf_process_gn
メニューアイテム名 (日本語)	未処理
メニューアイテム名 (英語)	Unprocessed
メニューアイテム名 (中国語 (中華人民共和国))	未处理
URL	bis/user/businessflow/process/process_list_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス : (ブランク)
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)

補足 (なし)

- メニューアイテム：処理済

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / BISフロー
メニューアイテムID	imbis_bf_cpl_proc_gn
メニューアイテム名 (日本語)	処理済
メニューアイテム名 (英語)	Processed
メニューアイテム名 (中国語 (中華人民共和国))	已理
URL	bis/user/businessflow/cpl_proc/actv_proc_list_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：(ブランク)
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- メニューアイテム：参照

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / BISフロー
メニューアイテムID	imbis_bf_reference_gn
メニューアイテム名 (日本語)	参照
メニューアイテム名 (英語)	Reference
メニューアイテム名 (中国語 (中華人民共和国))	参照
URL	bis/reference/businessflow/reference_user_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：(ブランク)
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- グローバルナビ (スマートフォン用)

- メニューグループ：IM-BIS BISフロー

設定項目	設定値
メニューグループID	bis-sp_bf_gn-sp
メニューグループ名 (日本語)	IM-BIS BISフロー
メニューグループ名 (英語)	IM-BIS BIS flow
メニューグループ名 (中国語 (中華人民共和国))	IM-BIS BIS flow
アイコン画像	ファイルパス：bis_smartphone/images/bpm-48.png

- メニューアイテム：処理開始

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS BISフロー
メニューアイテムID	imbis-sp_bf_apply_gn-sp
メニューアイテム名（日本語）	処理開始
メニューアイテム名（英語）	Start Process
メニューアイテム名 （中国語（中華人民共和国））	开始理
URL	bis_smartphone/user/businessflow/apply/apply_list_bf_pre
呼び出し方法	GET
引数	（なし）
アイコン画像	ファイルパス：bis_smartphone/images/bpm-apply-48.png
IFRAME表示	オフ
ポップアップ表示	オフ
説明	（なし）
補足	（なし）

- メニューアイテム：未処理

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS BISフロー
メニューアイテムID	imbis-sp_bf_process_gn-sp
メニューアイテム名（日本語）	未処理
メニューアイテム名（英語）	Unprocessed
メニューアイテム名 （中国語（中華人民共和国））	未理
URL	bis_smartphone/user/businessflow/process/process_list_bf_pre
呼び出し方法	GET
引数	（なし）
アイコン画像	ファイルパス：bis_smartphone/images/bpm-process-48.png
IFRAME表示	オフ
ポップアップ表示	オフ
説明	（なし）
補足	（なし）

- メニューアイテム：処理済

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS BISフロー
メニューアイテムID	imbis-sp_bf_cpl_proc_gn-sp
メニューアイテム名（日本語）	処理済
メニューアイテム名（英語）	Processed
メニューアイテム名 （中国語（中華人民共和国））	已理
URL	bis_smartphone/user/businessflow/cpl_proc/actv_proc_list_bf_pre
呼び出し方法	GET
引数	（なし）
アイコン画像	ファイルパス：bis_smartphone/images/bpm-approved-48.png

IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- メニューアイテム：参照

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS BISフロー
メニューアイテムID	imbis-sp_bf_reference_gn-sp
メニューアイテム名（日本語）	参照
メニューアイテム名（英語）	Reference
メニューアイテム名 （中国語（中華人民共和国））	参照
URL	bis_smartphone/reference/businessflow/reference_actv_list_bf_pre
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：bis_smartphone/images/bpm-reference-48.png
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- サイトマップ（PC用）

- メニューフォルダ：BISフロー

設定項目	設定値
メニューフォルダID	imbis_bf
メニューフォルダ名（日本語）	BISフロー
メニューフォルダ名（英語）	BIS flow
メニューフォルダ名 （中国語（中華人民共和国））	BIS flow
アイコン画像	ファイルパス：（ブランク）

- メニューアイテム：処理開始

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / BISフロー
メニューアイテムID	imbis_bf_start_process
メニューアイテム名（日本語）	処理開始
メニューアイテム名（英語）	Start Process
メニューアイテム名 （中国語（中華人民共和国））	开始[]理
URL	bis/user/businessflow/apply/apply_list_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：（ブランク）
IFRAME表示	オフ
ポップアップ表示	オフ

説明	(なし)
補足	(なし)

- メニューアイテム：未処理

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / BISフロー
メニューアイテムID	imbis_bf_process
メニューアイテム名（日本語）	未処理
メニューアイテム名（英語）	Unprocessed
メニューアイテム名 （中国語（中華人民共和国））	未处理
URL	bis/user/businessflow/process/process_list_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：（ブランク）
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- メニューアイテム：処理済

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / BISフロー
メニューアイテムID	imbis_bf_cpl_proc
メニューアイテム名（日本語）	処理済
メニューアイテム名（英語）	Processed
メニューアイテム名 （中国語（中華人民共和国））	已处理
URL	bis/user/businessflow/cpl_proc/actv_proc_list_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：（ブランク）
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- メニューアイテム：参照

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / BISフロー
メニューアイテムID	imbis_bf_reference
メニューアイテム名（日本語）	参照
メニューアイテム名（英語）	Reference
メニューアイテム名 （中国語（中華人民共和国））	参照
URL	bis/reference/businessflow/reference_user_bf

呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：(ブランク)
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- メニューアイテム：参照（業務管理者）

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / 業務管理者 / フロー
メニューアイテムID	imbis_bm_reference_bf
メニューアイテム名（日本語）	参照（BISフロー）
メニューアイテム名（英語）	Reference(BIS flow)
メニューアイテム名 （中国語（中華人民共和国））	参BISフローIS flow)
URL	bis/reference/businessflow/reference_operation_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：(ブランク)
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- メニューアイテム：一括処理対象者変更（業務管理者）

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / 業務管理者 / フロー
メニューアイテムID	imbis_bm_change_bf
メニューアイテム名（日本語）	一括処理対象者変更（BISフロー）
メニューアイテム名（英語）	Batch Process target User Change(BIS flow)
メニューアイテム名 （中国語（中華人民共和国））	批量[]理[]象[]更(BIS flow)
URL	bis/change/businessflow/change_operation_bf
呼び出し方法	GET
引数	(なし)
アイコン画像	ファイルパス：(ブランク)
IFRAME表示	オフ
ポップアップ表示	オフ
説明	(なし)
補足	(なし)

- メニューアイテム：参照（監査者）

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS / 監査者 / フロー

メニューアイテムID	imbis_auditor_reference_bf
メニューアイテム名（日本語）	参照（BISフロー）
メニューアイテム名（英語）	Reference(BISフロー flow)
メニューアイテム名 （中国語（中華人民共和国））	参照(BIS flow)
URL	bis/reference/businessflow/reference_audit_bf
呼び出し方法	GET
引数	（なし）
アイコン画像	ファイルパス：（ブランク）
IFRAME表示	オフ
ポップアップ表示	オフ
説明	（なし）
補足	（なし）

- サイトマップ（スマートフォン用）

- メニューグループ：IM-BIS BISフロー

設定項目	設定値
メニューグループID	bis-sp_bf_sm-sp
メニューグループ名（日本語）	IM-BIS BISフロー
メニューグループ名（英語）	IM-BIS BIS flow
メニューグループ名 （中国語（中華人民共和国））	IM-BIS BIS flow
アイコン画像	ファイルパス：（ブランク）

- メニューアイテム：処理開始

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS BISフロー
メニューアイテムID	imbis-sp_bf_apply_sm-sp
メニューアイテム名（日本語）	処理開始
メニューアイテム名（英語）	Start Process
メニューアイテム名 （中国語（中華人民共和国））	开始□理
URL	bis_smartphone/user/businessflow/apply/apply_list_bf_pre
呼び出し方法	GET
引数	（なし）
アイコン画像	ファイルパス：bis_smartphone/images/bpm-apply-16.png
IFRAME表示	オフ
ポップアップ表示	オフ
説明	（なし）
補足	（なし）

- メニューアイテム：未処理

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS BISフロー
メニューアイテムID	imbis-sp_bf_process_sm-sp
メニューアイテム名（日本語）	未処理

メニューアイテム名（英語）	Unprocessed
メニューアイテム名 （中国語（中華人民共和国））	未理
URL	bis_smartphone/user/businessflow/process/process_list_bf_pre
呼び出し方法	GET
引数	（なし）
アイコン画像	ファイルパス：bis_smartphone/images/bpm-process-16.png
IFRAME表示	オフ
ポップアップ表示	オフ
説明	（なし）
補足	（なし）

- メニューアイテム：処理済

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS BISフロー
メニューアイテムID	imbis-sp_bf_cpl_proc_sm-sp
メニューアイテム名（日本語）	処理済
メニューアイテム名（英語）	Processed
メニューアイテム名 （中国語（中華人民共和国））	已理
URL	bis_smartphone/user/businessflow/cpl_proc/actv_proc_list_bf_pre
呼び出し方法	GET
引数	（なし）
アイコン画像	ファイルパス：bis_smartphone/images/bpm-approved-16.png
IFRAME表示	オフ
ポップアップ表示	オフ
説明	（なし）
補足	（なし）

- メニューアイテム：参照

設定項目	設定値
設定先のメニューグループ・メニューフォルダ	IM-BIS BISフロー
メニューアイテムID	imbis-sp_bf_reference_sm-sp
メニューアイテム名（日本語）	参照
メニューアイテム名（英語）	Reference
メニューアイテム名 （中国語（中華人民共和国））	参照
URL	bis_smartphone/reference/businessflow/reference_actv_list_bf_pre
呼び出し方法	GET
引数	（なし）
アイコン画像	ファイルパス：bis_smartphone/images/bpm-reference-16.png
IFRAME表示	オフ
ポップアップ表示	オフ
説明	（なし）
補足	（なし）

3. これでBISフロー関連のメニューを利用可能に設定することができました。

サンプル

外部連携サンプルプログラム (JAVA)

Contents

- このサンプルプログラムについて
- サンプルプログラムのダウンロード
- サンプルプログラムのインポート
- サンプルプログラムのコード

このサンプルプログラムについて

このサンプルプログラムでは、入力したユーザIDから、ユーザ情報の明細を取得する処理を行います。

1. ユーザIDを入力します。複数のユーザIDを入力する場合は、「, (カンマ)」で区切ります。

外部連携サンプル (JAVA)

ユーザIDを入力して、ユーザ情報取得ボタンをクリックしてください。
「, (カンマ)」で区切ることで、複数のユーザIDを入力することも可能です。
例) aoyagi,yoshikawa,maruyama

ユーザID ユーザ情報取得

ユーザ情報

+	ユーザID	ユーザ名	開始日	ソートキー	削除フラグ
1	<input type="text"/>				

申請

2. ユーザ情報取得ボタンをクリックします。

外部連携サンプル (JAVA)

ユーザIDを入力して、ユーザ情報取得ボタンをクリックしてください。
「, (カンマ)」で区切ることで、複数のユーザIDを入力することも可能です。
例) aoyagi,yoshikawa,maruyama

ユーザID ユーザ情報取得

ユーザ情報

+	ユーザID	ユーザ名	開始日	ソートキー	削除フラグ
1	<input type="text"/>				

申請

3. ユーザ情報の明細を取得して表示します。

外部連携サンプル (JAVA)

ユーザIDを入力して、ユーザ情報取得ボタンをクリックしてください。
「, (カンマ)」で区切ることで、複数のユーザIDを入力することも可能です。
例) aoyagi,yoshikawa,maruyama

ユーザID ユーザ情報取得

ユーザ情報

+	ユーザID	ユーザ名	開始日	ソートキー	削除フラグ
1	aoyagi	青柳原巳	2000/01/01	2	false
2	yoshikawa	吉川一哉	2000/01/01	7	false
3	maruyama	円山益男	2000/01/01	4	false

申請

サンプルプログラムのダウンロード

サンプルプログラムは以下のリンクより入手できます。

- サンプルプログラム
[sampleJava-1.0.0.imm](#)
- データソース定義

[datasource-java.zip](#)

- IM-Workflow定義
[IM-Workflow-java.zip](#)
- IM-Forma定義
[IM-Forma-java.zip](#)
- IM-BIS定義
[IM-BIS-java.zip](#)

サンプルプログラムのインポート

ダウンロードしたファイルは、下記手順にてインポートすることができます。

- IM-Jugglingを起動し、Jugglingプロジェクト内にあるjuggling.imをIM-Juggling Editorで開きます。
- 「ユーザモジュール」タブを選択し、右上にある「+」のアイコンをクリックします。
- ダウンロードした「sampleJava-1.0.0.imm」ファイルを選択します。
- Jugglingプロジェクトからwarファイルを作成します。サンプルを含めるにチェックを入れます。



コラム

warファイルの作成については、「[intra-mart Accel Platform セットアップガイド](#)」 - 「[WARファイルの作成](#)」を参照してください。

- 作成したWARファイルを、Web Application Server（Resin）上にデプロイ（展開）します。



注意

「BIS管理者」ロールを付与したユーザでインポートを実施してください。



注意

IM-Workflow定義のインポートでは、zipファイルを解凍し、xmlファイルをインポートしてください。

- ダウンロードしたファイルをインポートします。
インポート方法は、「[IM-BIS 業務管理者操作ガイド](#)」 - 「[インポート・エクスポートを行う](#)」を参照してください。
- 「IM-Workflow ユーザ」ロールを付与したユーザでログインします。
- [サイトマップ] - [ワークフロー] - [申請] - 「外部連携サンプル (JAVA)」の「申請/処理開始」をクリックします。
- テキストボックスにユーザIDを入力し、「ユーザ情報取得ボタン」をクリックします。

ユーザ情報の明細を取得します。

複数のユーザIDを「, (カンマ)」で区切って入力すると、複数のユーザ情報を取得します。

外部連携サンプル (JAVA)

ユーザIDを入力して、ユーザ情報取得ボタンをクリックしてください。
「, (カンマ)」で区切ることで、複数のユーザIDを入力することも可能です。
例) aoyagi,yoshikawa,maruyama

ユーザID

ユーザ情報

+	ユーザID	ユーザ名	開始日	ソートキー	削除フラグ
1	<input type="text" value="aoyagi"/>	<input type="text" value="青柳 宗巳"/>	<input type="text" value="2000/01/01"/>	<input type="text" value="2"/>	<input type="text" value="false"/>
2	<input type="text" value="yoshikawa"/>	<input type="text" value="吉川 一哉"/>	<input type="text" value="2000/01/01"/>	<input type="text" value="7"/>	<input type="text" value="false"/>
3	<input type="text" value="maruyama"/>	<input type="text" value="円山 益男"/>	<input type="text" value="2000/01/01"/>	<input type="text" value="4"/>	<input type="text" value="false"/>

以下の操作をした場合、エラーメッセージを表示します。

- ユーザIDに何も入力しない

サンプルプログラムのコード

サンプルプログラムで使用するコードは以下の通りです。


```

1 package jp.co.intra_mart.system.bis.soa.func;
2
3 import java.math.BigDecimal;
4 import java.util.*;
5
6 import javax.swing.JFrame;
7 import javax.swing.JOptionPane;
8
9 import jp.co.intra_mart.foundation.context.Contexts;
10 import jp.co.intra_mart.foundation.context.model.AccountContext;
11 import jp.co.intra_mart.foundation.exception.BizApiException;
12 import jp.co.intra_mart.foundation.master.user.UserManager;
13 import jp.co.intra_mart.foundation.master.user.model.*;
14
15 public class SampleJava {
16
17     public SampleJava(){
18
19     }
20
21     private AccountContext getAccountContext(){
22         return (AccountContext)Contexts.get(AccountContext.class);
23     }
24
25     //Exceptionにthrowsすることで、Java連携の外部連携としてエラー処理されます
26     public List<Map<String,Object>> sample(Map<String,Object> map) throws Exception{
27
28         //レスポンスフィールド生成
29         List<Map<String,Object>> resUserList = new ArrayList<Map<String,Object>>();
30         Map<String,Object> detail;
31
32         //リクエストパラメータ取得
33         String reqUserList = map.get("userList").toString();
34
35         //リクエストパラメータの文字列を配列に変換
36         String[] userCdArray = reqUserList.split(",");
37
38         //トリミング実施
39         for(int i = 0; i < userCdArray.length; i++){
40             userCdArray[i] = userCdArray[i].trim();
41         }
42
43         for(int i = 0; i < userCdArray.length; i++){
44
45             //ユーザ情報取得
46             IUserBizKey userBizKey = new UserBizKey();
47             userBizKey.setUserCd(userCdArray[i]);
48             UserManager manager = new UserManager();
49             User user = manager.getUser(userBizKey, getAccountContext().getLoginTime());
50
51             if(user != null){
52
53                 //ユーザIDが存在する場合
54                 detail = new HashMap<String,Object>();
55
56                 //レスポンスパラメータにユーザ情報を設定する
57                 detail.put("user", user.getUserCd());
58                 detail.put("userName", user.getUserName());
59                 detail.put("startDate", user.getStartDate());
60                 detail.put("sortKey", BigDecimal.valueOf(user.getSortKey()));
61                 detail.put("deleteFlag", user.isDisable());
62
63                 resUserList.add(detail);
64             }
65             else{
66
67                 //ユーザIDが存在しない場合
68                 detail = new HashMap<String,Object>();
69
70                 //レスポンスパラメータを設定する
71                 detail.put("user", "ユーザIDがありません");
72                 detail.put("userName", null);
73                 detail.put("startDate", null);
74                 detail.put("sortKey", null);
75                 detail.put("deleteFlag", null);
76

```

```

77     resUserList.add(detail);
78     }
79     }
80     return resUserList;
81     }
82     }

```

コラム

外部連携（JAVA）の詳細な仕様については、「IM-BIS 仕様書」 - 「IM-BIS で外部連携として利用できるJavaプログラムの仕様」を参照してください。

コラム

外部連携（JAVA）へのリクエスト、レスポンスの設定方法については、「IM-BIS システム管理者操作ガイド」 - 「リクエストパラメータ、レスポンスフィールドの機能と各部の説明」を参照してください。

階層化したリクエスト、レスポンスを設定する場合は、「IM-BIS システム管理者操作ガイド」 - 「階層化したリクエストパラメータ、レスポンスフィールドの設定方法」を参照してください。

コラム

外部連携の設定方法については、「IM-BIS 業務管理者操作ガイド」 - 「外部連携を設定する」を参照してください。

動的処理対象者設定 外部連携（JAVA）サンプル

Contents

- 動的処理対象者設定 外部連携（JAVA）サンプルについて
- 動的処理対象者設定 外部連携（JAVA）サンプルのダウンロード
- 動的処理対象者設定 外部連携（JAVA）サンプルのインポート
- 動的処理対象者設定 外部連携（JAVA）サンプルのコード

動的処理対象者設定 外部連携（JAVA）サンプルについて

動的処理対象者設定では、動的に処理対象者を設定するノード（動的承認、縦配置、横配置）に対し、処理対象者設定条件を設定することができません。

動的処理対象者設定で外部連携設定を利用した場合は、処理対象者を自動で設定したり、手動で選択できる処理対象者の範囲を限定したりすることができます。

このサンプルでは、外部連携設定で選択可能なデータソース種別のうち、JAVAでの設定方法を使用します。

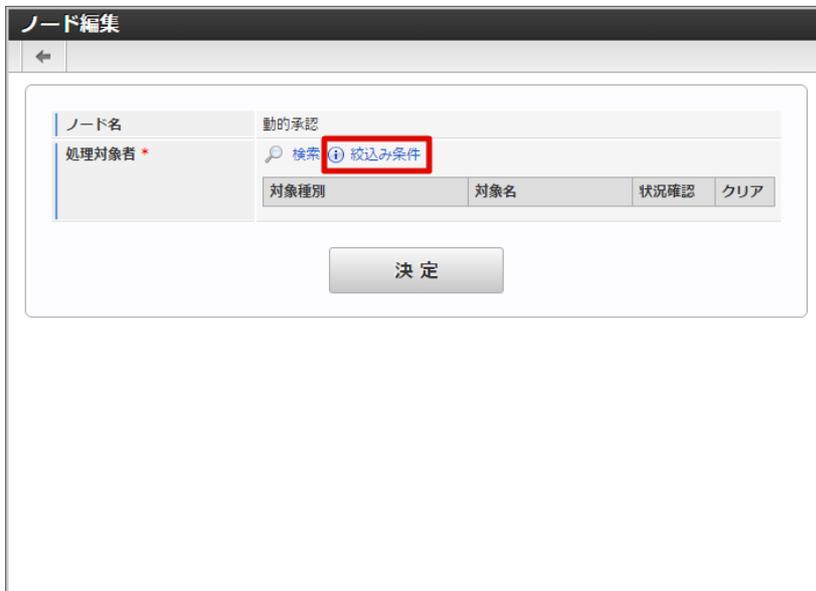
「承認者の初期設定サンプル」は、処理対象者を自動で設定する方法を使用します。

「承認者の絞込条件設定サンプル」は、手動で選択できる処理対象者の範囲を限定する方法を使用します。

- 承認者の初期設定サンプル
 - ノード編集画面に初期値を設定します。



- 承認者の絞込条件設定サンプル
絞込み条件確認画面に値を設定します。
ノード編集画面で検索ボタンをクリック後のユーザ検索画面に暗黙条件として適用されます。



i コラム

処理対象者、動的ノードについては、「IM-Workflow 仕様書」を参照してください。

i コラム

動的ノード（動的承認、縦配置、横配置）の処理対象者条件の設定については、「IM-BIS 業務管理者操作ガイド」 - 「動的ノード（動的承認、縦配置、横配置）の処理対象者条件を設定する」を参照してください。

i コラム

動的処理対象者設定については、「IM-BIS 業務管理者操作ガイド」 - 「動的処理対象者設定」または、「IM-BIS 仕様書」 - 「動的処理対象者設定に関する仕様」を参照してください。

i コラム

外部連携については、「IM-BIS 業務管理者操作ガイド」 - 「外部連携を設定する」または、「IM-BIS 仕様書」 - 「外部連携に関する仕様」を参照してください。

動的処理対象者設定 外部連携 (JAVA) サンプルのダウンロード

サンプルは、以下のリンクより入手できます。

- サンプルプログラム [bis_sample_dynamic-1.0.0.imm](#)
- データソース定義
初期設定サンプル [datasource_DynamicInitialProposal.zip](#)
絞込条件設定サンプル [datasource_DynamicCondition.zip](#)
- IM-Workflow 定義
初期設定サンプル [IM-Workflow_DynamicInitialProposal.zip](#)
絞込条件設定サンプル [IM-Workflow_DynamicCondition.zip](#)
- IM-Forma定義
初期設定サンプル [IM-Forma_DynamicInitialProposal.zip](#)
絞込条件設定サンプル [IM-Forma_DynamicCondition.zip](#)
- IM-BIS定義
初期設定サンプル [IM-BIS_DynamicInitialProposal.zip](#)
絞込条件設定サンプル [IM-BIS_DynamicCondition.zip](#)

動的処理対象者設定 外部連携 (JAVA) サンプルのインポート

ダウンロードしたファイルは、下記手順にてインポートすることができます。

1. IM-Jugglingを起動し、Jugglingプロジェクト内にあるjuggling.imをIM-Juggling Editorで開きます。
2. 「ユーザモジュール」タブを選択し、右上にある「+」のアイコンをクリックします。
3. ダウンロードした「bis_sample_dynamic-1.0.0.imm」ファイルを選択します。
4. Jugglingプロジェクトからwarファイルを作成します。サンプルを含めるにチェックを入れます。

i コラム

warファイルの作成については、「intra-mart Accel Platform セットアップガイド」 - 「WARファイルの作成」を参照してください。

5. 作成したWARファイルを、Web Application Server（Resin）上にデプロイ（展開）します。

! 注意

「BIS管理者」ロールを付与したユーザで、インポートを実施してください。

! 注意

IM-Workflow 定義については、zipファイルを解凍してから、インポートしてください。

6. ダウンロードしたファイルをインポートします。インポート方法は、「IM-BIS 業務管理者操作ガイド」 - 「インポート・エクスポートを行う」を参照してください。

これで、必要な設定作業はすべて完了しましたので、実際にフローで申請・承認を行ってみましょう。

i コラム

このサンプルでは、「BIS担当者」ロールを付与したユーザで、申請を行います。

i コラム

申請については、「[IM-BIS ユーザ 操作ガイド](#)」 - 「[BPMの処理を開始する／ワークフローの申請を行う](#)」を参照してください。

動的処理対象者設定 外部連携 (JAVA) サンプルのコード

このサンプルの外部連携 (JAVA) は、以下のように実装しています。

```

1 package jp.co.intra_mart.system.bis.soa.func;
2
3 import java.util.ArrayList;
4 import java.util.HashMap;
5 import java.util.List;
6 import java.util.Map;
7 import java.util.Map.Entry;
8
9 public class DynamicPersonJavaSample {
10
11     public DynamicPersonJavaSample() {
12     }
13
14     //初期設定
15     public Map<String,Map<String,List<Map<String,String>>>> getDynamicPerson(Map<String,Object> param) {
16
17         System.out.println("\n\n---start DynamicPersonJavaSample#getDynamicPerson----");
18         for (Entry<String, Object> entry : param.entrySet() ) {
19             System.out.println("key[" + entry.getKey()+"] value["+entry.getValue() +"]");
20         }
21         System.out.println("\n\n---start DynamicPersonJavaSample#getDynamicPerson----");
22         List<Map<String,String>> dynamicSettingList = new ArrayList<Map<String,String>>();
23
24         // ユーザ
25         Map<String,String> detail = new HashMap<String,String>();
26         detail.put("code", "1");
27         detail.put("processSetNo", "1");
28         detail.put("userCd", "aoyagi");
29         dynamicSettingList.add(detail);
30
31         // 組織
32         detail = new HashMap<String,String>();
33         detail.put("code", "2");
34         detail.put("processSetNo", "1");
35         detail.put("companyCd", "comp_sample_01");
36         detail.put("departmentSetCd", "comp_sample_01");
37         detail.put("departmentCd", "dept_sample_11");
38         detail.put("compare", "eq");
39         dynamicSettingList.add(detail);
40
41         // パブリックグループ
42         detail = new HashMap<String,String>();
43         detail.put("code", "3");
44         detail.put("processSetNo", "1");
45         detail.put("publicGroupSetCd", "sample_public");
46         detail.put("publicGroupCd", "public_team_a");
47         detail.put("compare", "eq");
48         dynamicSettingList.add(detail);
49
50         // 役職
51         detail = new HashMap<String,String>();
52         detail.put("code", "4");
53         detail.put("processSetNo", "1");
54         detail.put("companyCd", "comp_sample_01");
55         detail.put("departmentSetCd", "comp_sample_01");
56         detail.put("postCd", "ps002");
57         detail.put("compare", "eq");
58         dynamicSettingList.add(detail);
59
60         Map<String,List<Map<String,String>>> returnObject = new HashMap<String,List<Map<String,String>>>();
61         returnObject.put("settings", dynamicSettingList);
62
63         Map<String,Map<String,List<Map<String,String>>>> returnMap = new
64         HashMap<String,Map<String,List<Map<String,String>>>>();
65         returnMap.put("ResponseObject", returnObject);
66         System.out.println("---End DynamicPersonJavaSample#getDynamicPerson----\n\n\n");
67         return returnMap;
68     }
69
70     //絞込条件設定
71     public Map<String,Map<String,List<Map<String,String>>>> getDynamicPersonDepartment(Map<String,Object> param) {
72
73         System.out.println("\n\n---start DynamicPersonJavaSample#getDynamicPersonDepartment----");
74         for (Entry<String, Object> entry : param.entrySet() ) {
75             System.out.println("key[" + entry.getKey()+"] value["+entry.getValue() +"]");
76         }

```

```

77 System.out.println("\n\n---start DynamicPersonJavaSample#getDynamicPersonDepartment----");
78 List<Map<String,String>> dynamicSettingList = new ArrayList<Map<String,String>>();
79
80 Map<String,String> detail = new HashMap<String,String>();
81
82 // 組織+役職 (上位階層検索)
83 detail.put("code", "6");
84 detail.put("processSetNo", "1");
85 detail.put("companyCd", "comp_sample_01");
86 detail.put("departmentSetCd", "comp_sample_01");
87 detail.put("departmentCd", "dept_sample_11");
88 detail.put("compare", "ge");
89 detail.put("postCd", "ps003");
90 dynamicSettingList.add(detail);
91
92 Map<String,List<Map<String,String>>> returnObject = new HashMap<String,List<Map<String,String>>>();
93 returnObject.put("settings", dynamicSettingList);
94
95 Map<String,Map<String,List<Map<String,String>>>> returnMap = new
96 HashMap<String,Map<String,List<Map<String,String>>>>();
97 returnMap.put("ResponseObject", returnObject);
98 System.out.println("---End DynamicPersonJavaSample#getDynamicPersonDepartment----\n\n");
99 return returnMap;
100 }
}

```

動的処理対象者設定 外部連携 (LogicDesigner) サンプル

Contents

- 動的処理対象者設定 外部連携 (LogicDesigner) サンプルについて
- 動的処理対象者設定 外部連携 (LogicDesigner) サンプルのダウンロード
- 動的処理対象者設定 外部連携 (LogicDesigner) サンプルのインポート

動的処理対象者設定 外部連携 (LogicDesigner) サンプルについて

動的処理対象者設定では、動的に処理対象者を設定するノード（動的承認、縦配置、横配置）に対し、処理対象者設定条件を設定することができます。

動的処理対象者設定で外部連携設定を利用した場合は、処理対象者を自動で設定したり、手動で選択できる処理対象者の範囲を限定したりすることができます。

このサンプルでは、外部連携設定で選択可能なデータソース種別のうち、IM-LogicDesigner での設定方法を使用します。

「承認者の初期設定サンプル」は、処理対象者を自動で設定する方法を使用します。

「承認者の絞込条件設定サンプル」は、手動で選択できる処理対象者の範囲を限定する方法を使用します。

- 承認者の初期設定サンプル
ノード編集画面に初期値を設定します。

対象種別	対象名	状況確認	クリア
ユーザ	青柳辰巳	ⓘ	✖
組織	サンプル課 1 1	ⓘ	✖
パブリックグループ	チームA	ⓘ	✖
役職	部長	ⓘ	✖

- 承認者の絞り込み条件設定サンプル

絞り込み条件確認画面に値を設定します。

ノード編集画面で検索ボタンをクリック後のユーザ検索画面に暗黙条件として適用されます。

対象種別 ↑	対象名
組織	サンプル課 1 1 [上位]

i コラム

処理対象者、動的ノードについては、「IM-Workflow 仕様書」を参照してください。

i コラム

動的ノード（動的承認、縦配置、横配置）の処理対象者条件の設定については、「IM-BIS 業務管理者操作ガイド」 - 「動的ノード（動的承認、縦配置、横配置）の処理対象者条件を設定する」を参照してください。

i コラム

動的処理対象者設定については、「IM-BIS 業務管理者操作ガイド」 - 「動的処理対象者設定」または、「IM-BIS 仕様書」 - 「動的処理対象者設定に関する仕様」を参照してください。

i コラム

外部連携については、「IM-BIS 業務管理者操作ガイド」 - 「外部連携を設定する」または、「IM-BIS 仕様書」 - 「IM-LogicDesigner との連携の仕様」を参照してください。

サンプルは、以下のリンクより入手できます。

- IM-LogicDesigner 定義
初期設定サンプル [IM_LogicDesigner-DynamicInitialProposal.zip](#)
絞込条件設定サンプル [IM_LogicDesigner-DynamicCondition.zip](#)
- データソース定義
初期設定サンプル [datasource_DynamicInitialProposal_LD.zip](#)
絞込条件設定サンプル [datasource_DynamicCondition_LD.zip](#)
- IM-Workflow 定義
初期設定サンプル [IM-Workflow_DynamicInitialProposal_LD.zip](#)
絞込条件設定サンプル [IM-Workflow_DynamicCondition_LD.zip](#)
- IM-Forma定義
初期設定サンプル [IM-Forma_DynamicInitialProposal_LD.zip](#)
絞込条件設定サンプル [IM-Forma_DynamicCondition_LD.zip](#)
- IM-BIS定義
初期設定サンプル [IM-BIS_DynamicInitialProposal_LD.zip](#)
絞込条件設定サンプル [IM-BIS_DynamicCondition_LD.zip](#)

動的処理対象者設定 外部連携（LogicDesigner）サンプルのインポート

ダウンロードしたファイルは、下記手順にてインポートすることができます。

1. 「サイトマップ」→「LogicDesigner」→「インポート」から、ダウンロードした IM-LogicDesigner 定義をインポートします。



注意

「LogicDesigner管理者」ロールを付与したユーザで、インポートを実施してください。

2. ダウンロードしたファイルをインポートします。インポート方法は、「[IM-BIS 業務管理者操作ガイド](#)」-「[インポート・エクスポートを行う](#)」を参照してください。



注意

「BIS管理者」ロールを付与したユーザで、インポートを実施してください。



注意

IM-Workflow 定義については、zipファイルを解凍してから、インポートしてください。

これで、必要な設定作業はすべて完了しましたので、実際にフローで申請・承認を行ってみましょう。



コラム

このサンプルでは、「BIS担当者」ロールを付与したユーザで、申請を行います。



コラム

申請については、「[IM-BIS ユーザ 操作ガイド](#)」-「[BPMの処理を開始する／ワークフローの申請を行う](#)」を参照してください。

外部連携 処理結果メッセージ（LogicDesigner）サンプル

外部連携では、レスポンスパラメータとして処理結果メッセージを設定することができます。

以下から IM-LogicDesigner のサンプルがダウンロードできます。

サンプルは、IM-LogicDesigner の出力設定に処理結果メッセージを設定しています。

インポートした定義から複製して、フロー定義を作成することで、入出力設定が作成済みの状態からフロー定義を作成できます。前処理や後処理で利用する場合は、入力チェックエラーメッセージ部分は削除して利用してください。

- IM-LogicDesigner 定義
[sample_process_result_message.zip](#)

 コラム

処理結果メッセージについては、「[IM-BIS 仕様書](#)」 - 「[暗黙的に連携するレスポンスパラメータの仕様](#)」を参照してください。

 コラム

以下のintra-mart製品にて動作を確認しています。

- intra-mart : intra-mart Accel Platform 2015 Winter
- IM-BIS : IM-BIS for Accel Platform - 8.0.9

 注意

サンプルプログラムをインポートする場合、DBはPostgreSQLを使用してください。

 注意

サンプルプログラムを動作させる前に、サンプルデータを投入してください。

サンプルデータの投入方法については、「[intra-mart Accel Platform セットアップガイド](#)」 - 「[サンプルデータの投入](#)」を参照してください。

 注意

本書に掲載されているサンプルコードは可読性を重視しており、性能面や保守性といった観点において必ずしも適切な実装ではありません。

開発においてサンプルコードを参考にされる場合には、上記について十分に注意してください。