



クイック検索

検索

目次

Copyright © NTT DATA INTRAMART
CORPORATION

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. BPM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. ルール実行履歴レポート機能を利用する
 - 4.10. 一括処理対象者変更を利用する
- 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

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。 「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。 「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。 「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。 「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」

変更年月日	変更内容
2014-09-01	第6版 下記を追加・変更・削除しました。
	「 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。
	「 インポート・エクスポートを行う 」 「 テナントDBクエリ 」
2014-12-24	第8版 下記を追加・変更しました。
	「 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	第9版 下記を削除しました。
	「 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	第10版 下記を追加・変更しました。
	「 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	第11版 下記を追加・変更しました。
	「 JAVA 」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	第12版 下記を追加・変更しました。
	「 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。

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	- 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)	IM-BIS 業務管理者 - 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。 IM-BIS 監査者 - 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	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 に対応する機能・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

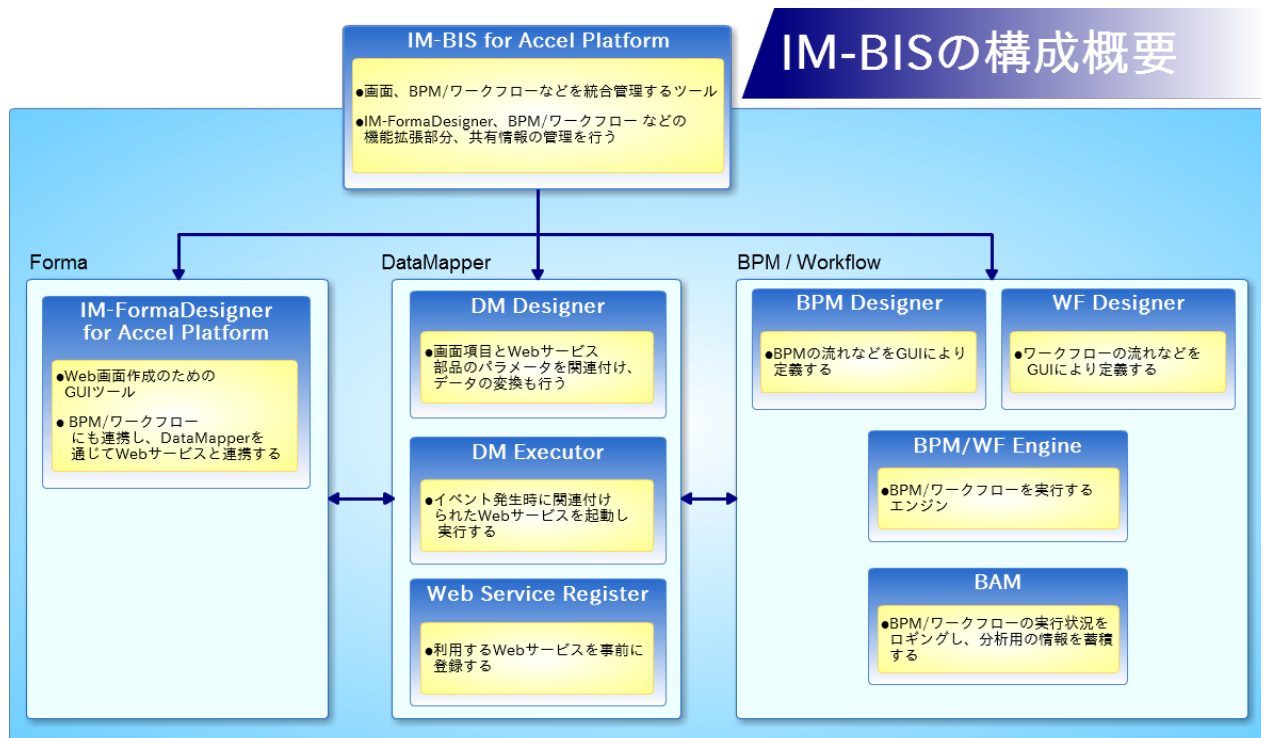
IM-BIS(Business Integration Suite)とは、intra-mart Accel Platform 上で動作する統合開発環境です。

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、BPM、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。
- BPM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。

BPMとワークフロー

IM-BIS でフローを作成する場合には、「BPM」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
 - 「BPM」をご利用ください。

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](#)

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BPM／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
BIS定義	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。 ■ 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。 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。
外部連携	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。

種別	説明
採番ルール定義	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。

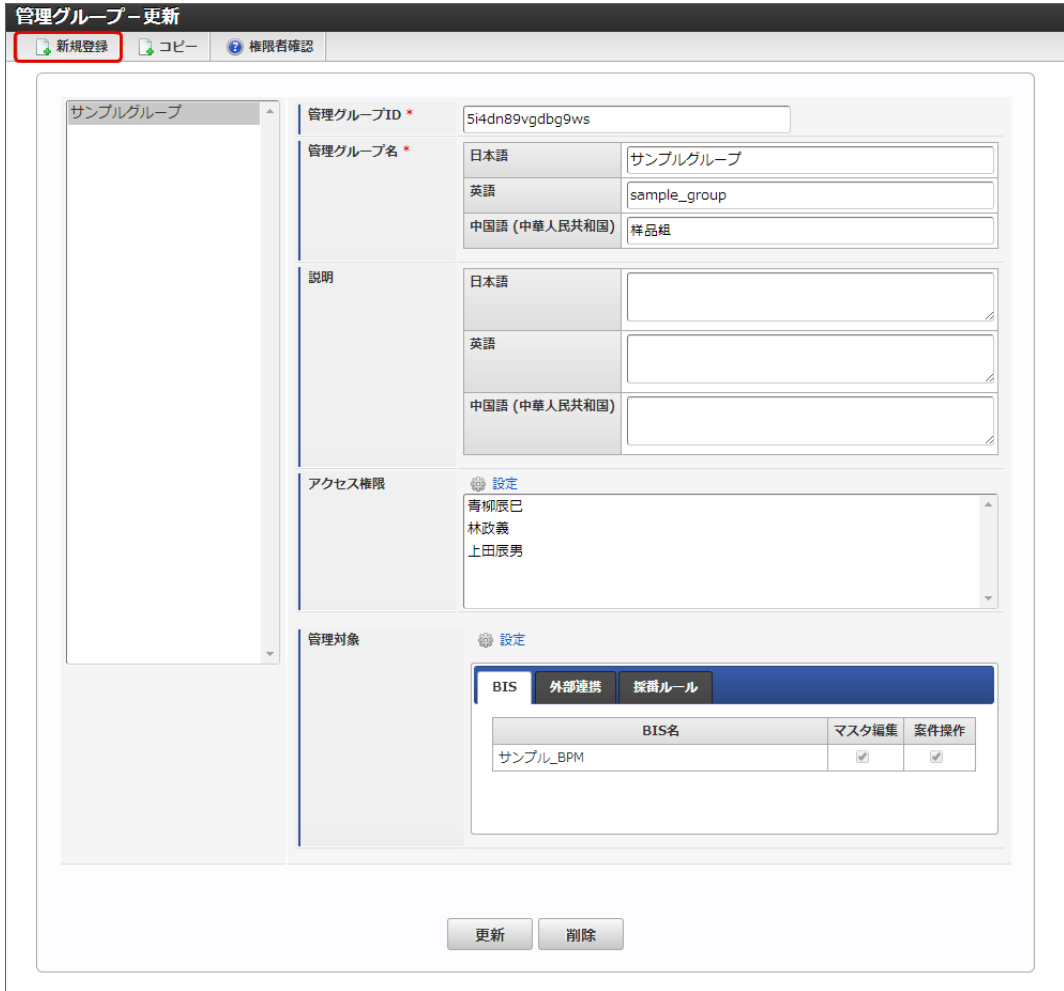
コラム

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。



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 * 5i4dn89vgdbg9ws

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	☒	☒

更新 削除

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 * 5i4dn8allnwayws

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

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

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 * 5i4dn8allnwayws

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

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

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。

管理グループアクセス権限設定

設定

+ 追加 - 削除

	権限種類	権限名
☐	ユーザ	上田辰男
☐	ユーザ	青柳辰巳

8. 「管理対象」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 * 5i4dn8allnwayws

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 マスタ編集 案件操作

登録

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。



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となります。
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
フォームの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 - 新規登録

←

サンプルグループ

管理グループID * 5i4dn8allnwayws

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	マスタ編集	案件操作
サンプル_BPM	☒	☒

登録

i 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、ご注意ください。

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ご注意ください。

■ 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(BPM)

監査者＞フロー＞参照(ワークフロー)

監査者＞フロー＞参照(BPM)

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(BPM)



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、ご注意ください。

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BPM／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. 設定する管理グループの編集画面を表示します。

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	☒	☒

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。

管理グループ管理対象設定

設定

BIS 外部連携 採番ルール

+ 追加 - 削除

	BIS名	マスタ編集	案件操作
☐	サンプル_BPM	☒	☒

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。

管理グループ-管理対象設定

設定

BIS 外部連携 採番ルール

+ 追加 - 削除

	外部連携名
☐	備品検索
☐	乱数生成

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。

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 * 5i4dn89vgdbg9ws

管理グループ名 *

言語	名前
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

言語	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

青柳限巳
林政義
上田展男

管理対象

設定

BIS 外部連携 採番ルール

外部連携名
備品検索
乱数生成

更新 削除

「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」もしくは「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

最近使用した定義 新規登録

BISシステム管理者

☒ BPM ☒ WF
BIS名 検索

選択した定義を削除

編集	BIS作成種類	BIS名 ↑	説明	BIS ID	フローID	アプリ
☐	BPM	サンプル_BPM		5i7j4m8tgrw2w6	5i7j4m8tr1ocqv6	
☐	BPM	サンプルBPM		5i7j4lnisvaenv6	5i7j4lnj1vjouv6	
☐	WF	サンプルワークフロー		5i7j5z0zum68iv6	5i7j5z10jdaxav6	

1 ページ中 1 ページ目 15 3 件中 1 - 3 を表示

IM-BIS - 一覧

最近使用した定義 新規登録

BIS業務管理者

☒ BPM ☒ WF
BIS名 検索

選択した定義を削除

編集	BIS作成種類	BIS名 ↑	説明	BIS ID
☐	BPM	サンプルBPM		5i7j4lnisvaenv6
☐	WF	サンプルワークフロー		5i7j5z0zum68iv6

1 ページ中 1 ページ目 15 2 件中 1 - 2 を表示

「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。

IM-BIS - 一覧

(1) 最近使用した定義 (2) 新規登録 (3) 検索

(4) BPM (5) WF

(6) BIS名 (7) サンプル (8) 検索

(9) 選択した定義を削除

(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
	編集	BIS作成種類	BIS名 ↑	説明	BIS ID	フローID	アプリ
☒		BPM	サンプル_BPM		5i7j4m8tgrw2w6	5i7j4m8tr1ocqv6	
☐		BPM	サンプルBPM		5i7j4lnisvaenv6	5i7j4lnj1vjou6	
☐		WF	サンプルワークフロー		5i7j5zum68iv6	5i7j5z10jdaxav6	

(11)

1 ページ中 1 ページ目 15 3 件中 1 - 3 を表示

- 最近使用した定義
「IM-BIS - 更新履歴」画面を表示します。
- 新規登録
「IM-BIS - 新規登録」画面を表示します。
- 最新情報
表示している内容を最新の状態に更新します。
- BPM(チェックボックス)
チェックをオンにすると、BPMを検索対象にします。
- 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。
- 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。
- 入力フィールド
検索するキーワードを入力します。
- 検索ボタン
押下すると検索を実行します。
- 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。
- 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。
- 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、ご注意ください。



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

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。
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りますので、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](#)

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)	5i7j6ud2img93y
	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
	ショートキー(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		

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)	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 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	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 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 検索 クリア		

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 検索 クリア		

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)	5i7j6ud6o8tts3y	
	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

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	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

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

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	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 検索 クリア		

更新 削除

- 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。
- 「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)	5i7j6udd0ch6m3y		
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

- 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: 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

検索

クリア

更新 削除

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

1. 「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。
2. 「検索」をクリックします。

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
- 役職
- 役割
- 組織+役職
- 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	i	x

更新

5. 代理管理者設定を登録することができました。

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

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

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ーフェースを抽象化し、Webサービスでも、テーブルでも、同じインター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。

データソース - 新規登録

←

(1) データソース種別 * テナントDBクエリ

(2) データソース名 *

(3) 備考

(4) 他 の ロ ケ ー ル

登録

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: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: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: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。



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モジュール / 簡易操作ガイド」を参照してください。

「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。

データソース - 新規登録

データソース種別 * テナントDBクエリ

データソース名 *

備考

他のロケール

登録

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 - 新規登録

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

備考 テナントDBクエリのサンプルです。

他のロケール

登録

「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
  column2 = ?
AND
  column3 IN ( ?, ? )
```

(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: 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	-

(4) 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	-

(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ではエラーになります。
SQLServer では、ORDER BY 句 を指定しないようにしてください。
ただし、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るようになります。
- 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、以下の点にご注意ください。

- 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をチェックします。

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。



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
  imfr_ut_product_master_regist.imfr_ud_product_name,
  imfr_ut_product_master_regist.imfr_ud_cost
FROM
  imfr_ut_product_master_regist
WHERE
  imfr_ut_product_master_regist.imfr_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_regist.imfr_ud_product_name,
  imfr_ut_product_master_regist.imfr_ud_cost
FROM
  imfr_ut_product_master_regist
WHERE
  imfr_ut_product_master_regist.imfr_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	-

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

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_regist.imfr_ud_product_name,
  imfr_ut_product_master_regist.imfr_ud_cost
FROM
  imfr_ut_product_master_regist
WHERE
  imfr_ut_product_master_regist.imfr_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	-

出力値

	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_regist.imfr_ud_product_name,
  imfr_ut_product_master_regist.imfr_ud_cost
FROM
  imfr_ut_product_master_regist
WHERE
  imfr_ut_product_master_regist.imfr_ud_locale = ?
```

入力値 - + 追加

	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 item1 中国語 条件項目1	-

出力値 - + 追加

	カラム名	データ型	論理名	削除
OUT 1	imfr_ud_product_nam	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

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。

データソース編集[テナントDBクエリ]

←

テスト実行に成功しました。取得件数: 23

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_regist.imfr_ud_product_name,
  imfr_ut_product_master_regist.imfr_ud_cost
FROM
  imfr_ut_product_master_regist
WHERE
  imfr_ut_product_master_regist.imfr_ud_locale = ?
```

入力値

	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 item1 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_nam	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

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_regist, imfr_ud_product_name,
  imfr_ut_product_master_regist, imfr_ud_cost
FROM
  imfr_ut_product_master_regist
WHERE
  imfr_ut_product_master_regist, imfr_ud_locale = ?
```

入力値

	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 item1 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	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

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
☒	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 15 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: [Dropdown]

(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	[Input Field]	VARCHAR	日本語: 条件項目 1 英語: Condition item1 中国語: 条件項目 1	-
IN 2	[Input Field]	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 Field]	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	[Input Field]	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

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. 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。

3. 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ではエラーになります。

SQLServer では、ORDER BY句 を指定しないようにしてください。

ただし、IM-BISの外部連携から実行された場合は該当しません。

4. 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)を選択します。

- 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。

5. 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るようになります。
- 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、以下の点にご注意ください。

- 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をチェックします。

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

	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

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

	テスト実行値	データ型	論理名	削除
IN 1		VARCHAR	日本語 条件1 英語 Condition1 中国語 条件1	-

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

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

	テスト実行値	データ型	論理名	削除
IN 1		VARCHAR	日本語 条件1 英語 Condition1 中国語 条件1	-

出力値

	カラム名	データ型	論理名	削除
OUT 1	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

	テスト実行値	データ型	論理名	削除
IN 1	1	VARCHAR	日本語: 条件1 英語: Condition1 中国語: 条件1	-

出力値

	カラム名	データ型	論理名	削除
OUT 1	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。

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

テスト実行に成功しました。取得件数: 1

接続ID: bis_test_db

SQL:

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

入力値

	テスト実行値	データ型	論理名	削除
IN 1	1	VARCHAR	日本語: 条件1 英語: Condition1 中国語: 条件1	-

出力値

	カラム名	データ型	論理名	削除
OUT 1	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

7. 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

	テスト実行値	データ型	論理名	削除
IN 1	1	VARCHAR	日本語: 条件1 英語: Condition1 中国語: 条件1	-

出力値

	カラム名	データ型	論理名	削除
OUT 1	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

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。

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
☒	シェアードDBクエリ	シェアードDBクエリサンプル	シェアードDBクエリのサンプルです。

1件中 1 - 1 を表示

REST

「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 - 編集[REST]

データソース種別 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  http://sample.com

(4) URI  /getCompanyDetail

(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  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	☐	companyCd	string		なし	-

(9) 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	Company	object		なし	-
2	companyCd	string		1	-
3	companyName	string		1	-
4	companyAddress	string		1	-
5	companyBusiness	string		1	-

(10) 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. 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。

3. RESTのURL

REST APIのエントリ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。

4. 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	
1	☐	appid	string		なし	—
2	☐	output	string		なし	—
3	☐	callback	string		なし	—
4	☐	area	string		なし	—
5	☐	datetime	date	yyyy-MM-dd	なし	—
6	☐	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」、もしくは「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」、もしくは「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

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. 「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: /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

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: /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

	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	☐		string		なし	-

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
--	-------	------	--------	---------	----

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

	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	☐	param_plus	number		なし	-
2	☐	param_array	array		なし	-
3	☐		object		2	-
4	☐	param_str	string		3	-
5	☐	param_num	number		3	-
6	☐	param_date	date	y/M/d	3	-
7	☐	param_bin	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	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

	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	☐	param_plus	number		なし	-
2	☐	param_array	array		なし	-
3	☐		object		2	-
4	☐	param_str	string		3	-
5	☐	param_num	number		3	-
6	☐	param_date	date	y/M/d	3	-
7	☐	param_bin	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	削除
1		string		なし	-

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: /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

	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	☐	param_plus	number		なし	-
2	☐	param_array	array		なし	-
3	☐		object		2	-
4	☐	param_str	string		3	-
5	☐	param_num	number		3	-
6	☐	param_date	date	y/M/d	3	-
7	☐	param_bin	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	削除
1		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

/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	☐	param_plus	number		なし	-
2	☐	param_array	array		なし	-
3	☐		object		2	-
4	☐	param_str	string		3	-
5	☐	param_num	number		3	-
6	☐	param_date	date	y/M/d	3	-
7	☐	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
1	param_array	array		なし	-
2		object		1	-
3	param_str	string		2	-
4	param_num	number		2	-
5	param_date	date	y/M/d	2	-
6	param_bln	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: /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

	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	☐	param_plus	number		なし	-
2	☐	param_array	array		なし	-
3	☐		object		2	-
4	☐	param_str	string		3	-
5	☐	param_num	number		3	-
6	☐	param_date	date	y/M/d	3	-
7	☐	param_bln	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	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	y/M/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
☒ デナント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
☐	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件中 1 - 1 を表示

SOAP

「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

(3) WSDLのURL: http://192.168.111.184:8080/mart802/services/WSMasterTest?wsdl (4) 読み込み

(5) サービス名: WSMasterTest

(6) ポート: WSMasterTestHttpSoap11Endpoint

(7) オペレーション: getValueArray

(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
(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	getValueArrayResponse	object			なし
2	return	array			1
3		string			2

(10) 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. 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。

3. 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。

4. 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。

5. 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。

6. 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。

7. 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。

8. 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。)

9. 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。)

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	
1	☐	appid	string		なし	—
2	☐	output	string		なし	—
3	☐	callback	string		なし	—
4	☐	area	string		なし	—
5	☐	datetime	date	yyyy-MM-dd	なし	—
6	☐	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」、もしくは「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」、もしくは「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。

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:// /imart/services/WSTest10?wsdl

読み込み

サービス名

ポート

オペレーション

2. 「読み込み」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:// /imart/services/WSTest10?wsdl 読み込み
サービス名
ポート
オペレーション

3. 正しく読み込めると、「WSDLの読み込みに成功しました。」という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  http:// /imart/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	☐	wsTest10	object		なし
2	☐	param	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

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

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	☐	wsTest10	object		なし
2	☐	param	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

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:// /imart/services/WSTest10?wsdl 

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	☐	wsTest10	object		なし
2	☐	param	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

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
☐ 	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 15 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るようになります。

JAVA

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

「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	mbImail	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	☐	appid	string		なし	—
2	☐	output	string		なし	—
3	☐	callback	string		なし	—
4	☐	area	string		なし	—
5	☐	datetime	date	yyyy-MM-dd	なし	—
6	☐	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」、もしくは「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」、もしくは「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。

実行ファイル一覧

保存

選択	ファイル名	説明
☐	SystemFunction.jar	

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。

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: 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
--	-------	------	--------	---------	----

4. 「読み込み」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:

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
--	-------	------	--------	---------	----

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

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

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

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

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

	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

	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
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

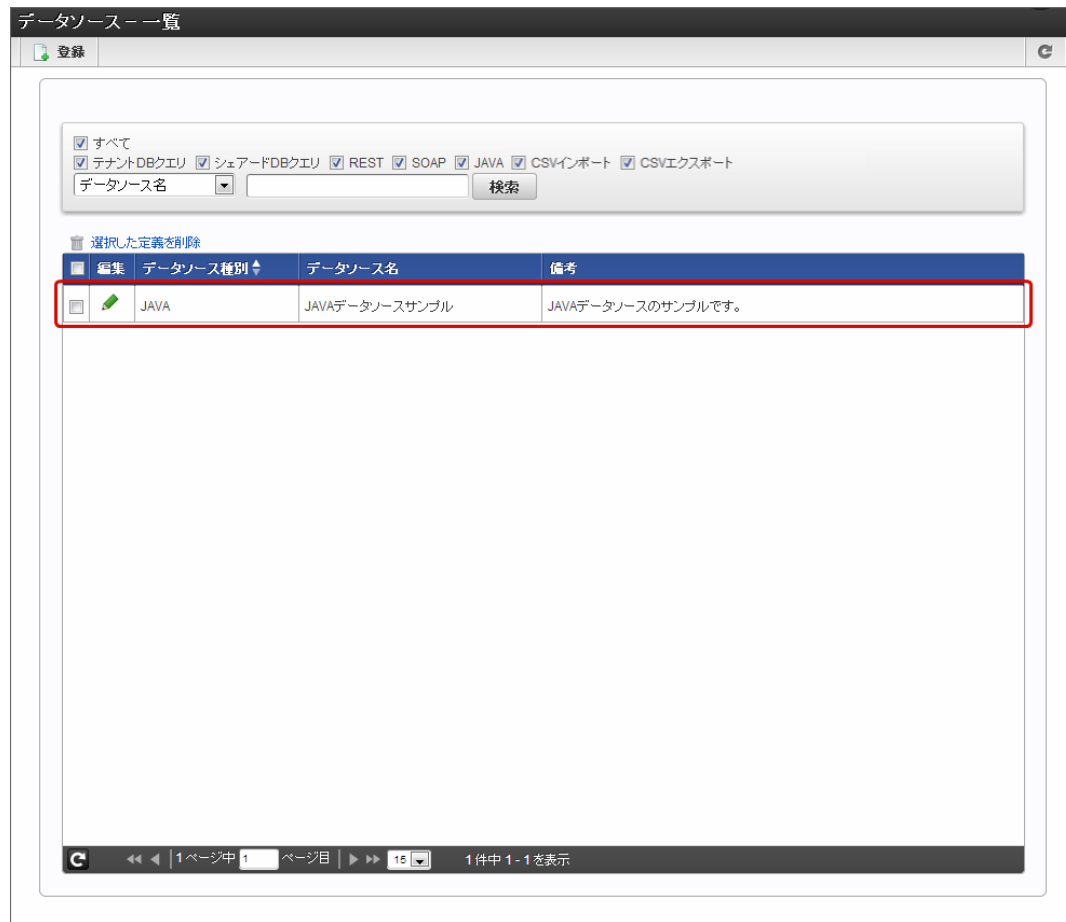
	パラメータ	データ型	フォーマット	親オブジェクト	削除
1	root	object		なし	-
2	parameter1	string		1	-
3	parameter2	number		1	-

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。



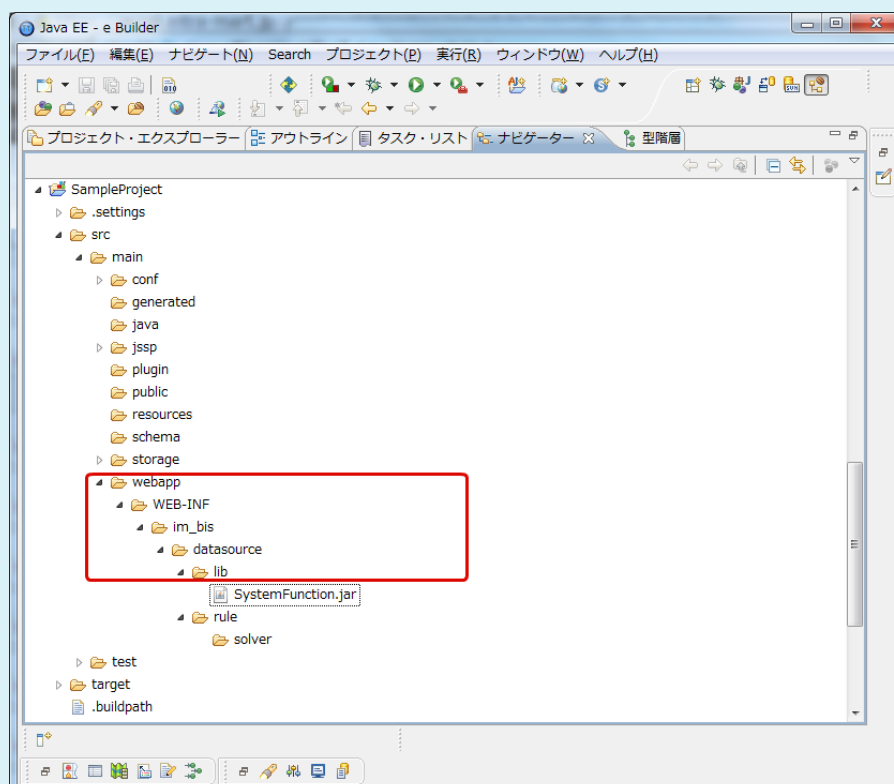
コラム

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」となります。

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。



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

(3) フロー定義: document_test (4) フロー選択

(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) 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. バージョン
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[JAVA]」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。

データソース編集[LogicDesigner]

データソース種類 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 フロー選択

バージョン番号

登録

- バージョン指定なし
常に最新の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、変更しない場合に有効となります。
- バージョンを指定する
特定のバージョンを指定して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。

1. 「フロー選択」をクリックします。

データソース編集[LogicDesigner]

データソース種類 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

バージョン番号

登録

2. 「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
☒	document_test	document_test
☐	LDTEST2	LDTEST2

11ページ中 1 ページ目 10 2件中 1-2を表示



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、LogicDesignerのフロー定義の選択のみとなります。

「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、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。

1. 「リクエスト」「レスポンス」は、選択した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	map	なし



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

1. 「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

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



コラム

LogicDesignerのフロー定義は、事前に登録する必要があります。
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 - +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

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-31J、UTF-8など)を記述します。

3. 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。

4. 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

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	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。

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
☐	CSVインポート	CSVインポートサンプル	CSVインポートのサンプルです。

1 ページ中 1 ページ目 15 1件中 1 - 1を表示

CSVエクスポート

「データソース - 編集[CSVエクスポート]」画面の機能と各部の説明

「データソース - 編集[CSVエクスポート]」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)をから指定します。

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]

データソース種別: 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]

データソース種別 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。

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
☐		CSVエクスポート	CSVエクスポートサンプル	CSVエクスポートのサンプルです。

1件中 1 - 1 を表示

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 *: DeterminedIfToBuyStock

(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になります。

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となっています。

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

- 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。

Jar名	説明
DecisionXYZ.jar	

「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 ⊖					
	(1)	(2)	(3)	(4)	(5) + 追加
	パラメータ	データ型	フォーマット	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」、もしくは「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」、もしくは「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。



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](#)」-「[バ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
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

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
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	☒	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
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. 「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	☒	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
+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
登録					

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 ☐ 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	☒	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		string		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					

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 ☐ 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	☒	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
--	-------	------	--------	---------	----

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	☒	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
--	-------	------	--------	---------	----

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	☒	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	☒	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	☒	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。

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 ☒ 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
☐		ルール	RuleEngineサンプル	

1 ページ中 1 ページ目 15 1 件中 1-1 を表示

■ 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。

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

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. 「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	☒	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	☒	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。

Java Bean アーカイブ

保存

Jar名: 検索 クリア

Jar名	説明
☐ DecisionXYZ.jar	RuleSolverのサンプルで使用するjarです。

OK

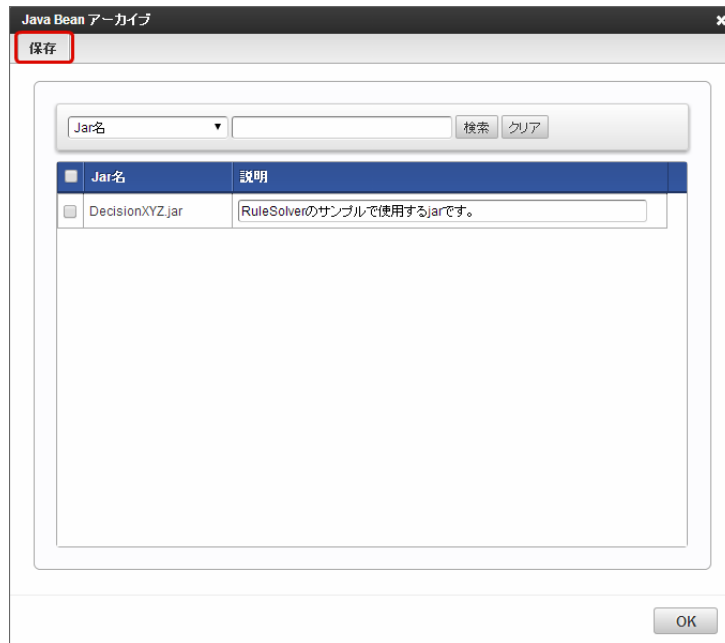


注意

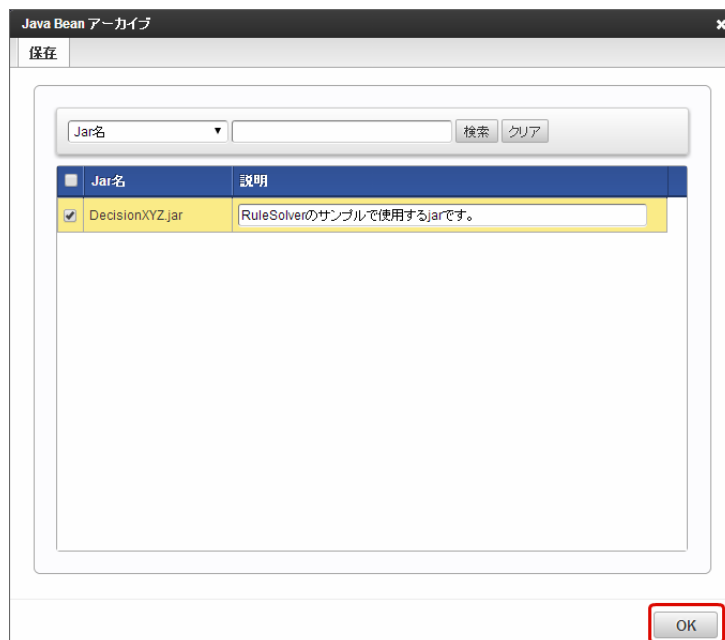
説明の入力は任意です。

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、ご注意ください。

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	☒	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
--	-------	------	--------	---------

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
☒ 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	☒	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 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	☒	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
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DB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
☐	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: テナントDB-DMLサンプル (1)

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

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

(4) 出力値

	カラム名	データ型	論理名	削除
OUT 1	count	NUMBER	処理件数 英語 中国語 (中華人民共和国) Number 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、もしくはテスト実行後に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。



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、以下の通りとなります。

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



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		VARCHAR	更新項目 1-1 Update item1-1 更新項目 1-1	-
IN 2		VARCHAR	更新項目 1-2 Update item1-2 更新項目 1-2	-
IN 3		VARCHAR	更新項目 1-3 Update item1-3 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	名前 名前 名前	-
IN 2	lemon	VARCHAR	コード コード 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

テスト実行に成功しました。取得件数:1

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 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
☒	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 15 1件中 1 - 1 を表示

「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

接続ID: [Dropdown]

(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	[Input Field]	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	[Input Field]	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	[Input Field]	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、もしくはテスト実行後に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。



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、以下の通りとなります。

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



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

1. 接続IDを選択し、「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		VARCHAR	日本語 英語 中国語 (中華人民共和国) 更新項目 1-1 Update item1-1 更新項目 1-1	-
IN 2		VARCHAR	日本語 英語 中国語 (中華人民共和国) 更新項目 1-2 Update item1-2 更新項目 1-2	-
IN 3		VARCHAR	日本語 英語 中国語 (中華人民共和国) 更新項目 1-3 Update item1-3 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		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		VARCHAR	日本語 英語 中国語 (中華人民共和国) 名前 名前 名前	-
IN 3		VARCHAR	日本語 英語 中国語 (中華人民共和国) 単価 単価 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

	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	メロン	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。

データソース編集[シェアードDB更新系クエリ]

←

テスト実行に成功しました。取得件数:1

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

	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	メロン	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

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

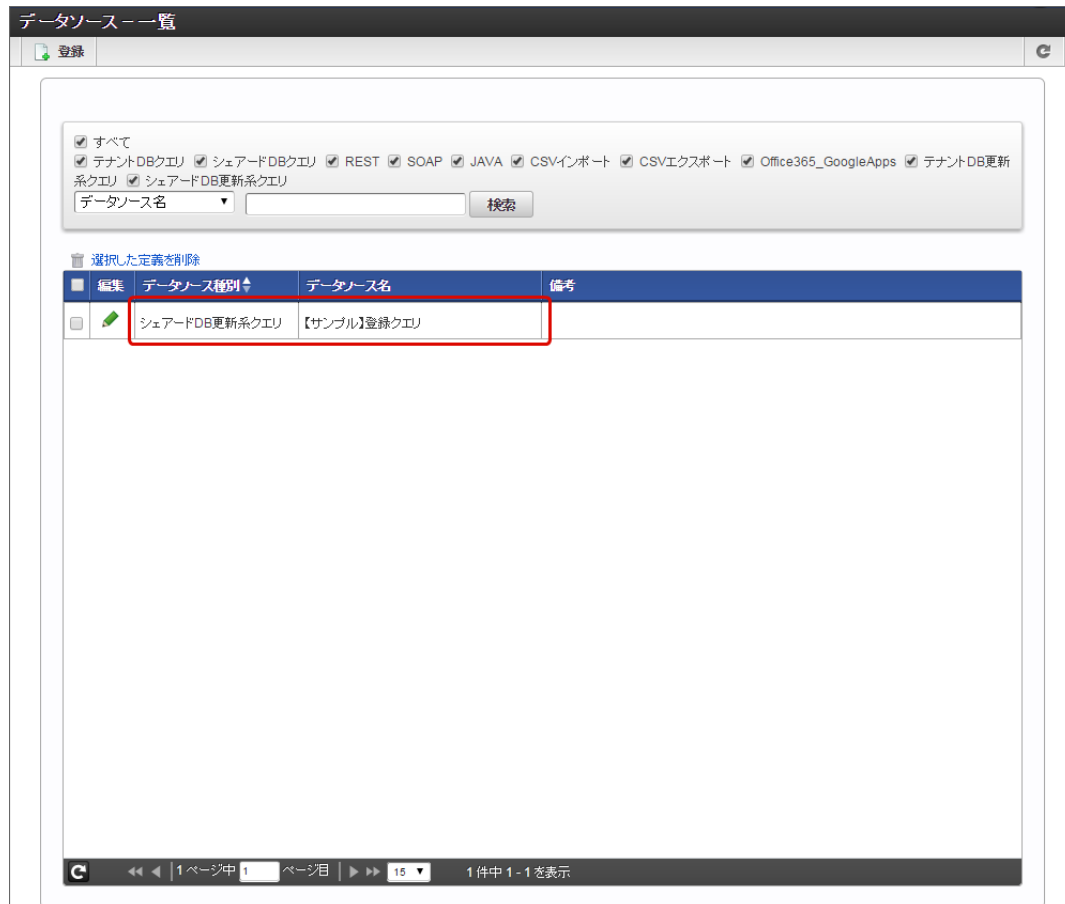
	テスト実行値	データ型	論理名	削除
IN 1	melon	VARCHAR	日本語: コード 英語: コード 中国語 (中華人民共和国): コード	-
IN 2	メロン	VARCHAR	日本語: 名前 英語: 名前 中国語 (中華人民共和国): 名前	-
IN 3	500	NUMBER	日本語: 単価 英語: 単価 中国語 (中華人民共和国): 単価	-

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	

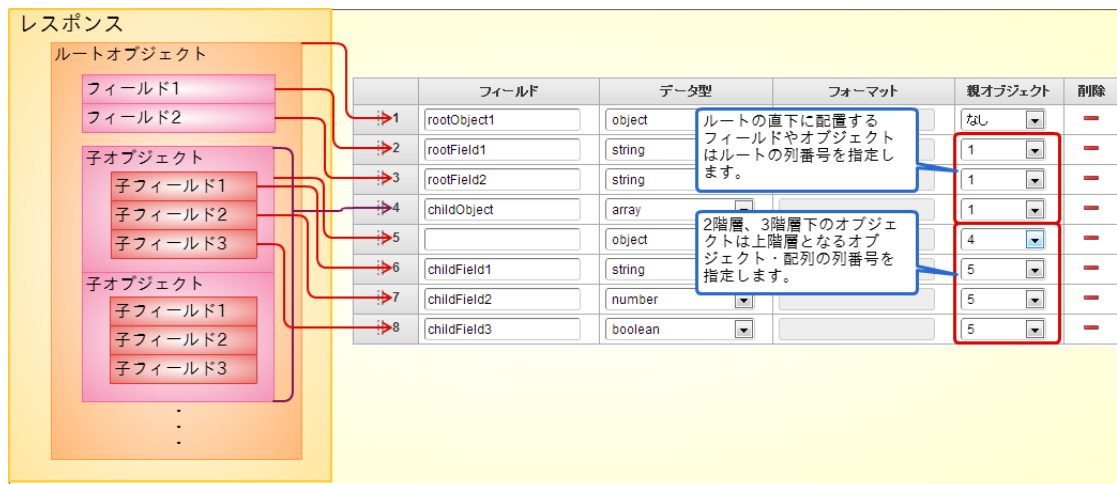
テスト実行 登録

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。

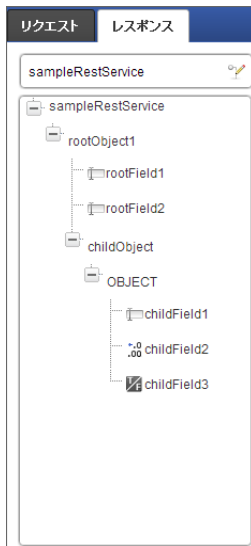


階層化したリクエストパラメータ、レスポンスフィールドの設定方法

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](#)

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
- インポートを実行する

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BPM／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BPM／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名となります。
- ルート定義
必ず作成され、ルート名はBIS名となります。
- フロー定義
必ず作成され、フロー名はBIS名となります。
- 案件プロパティ定義
BPM／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
BPM／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

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BPM／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)」にチェックします)

🏠 エクスポート								
コンテンツ定義	☐ エクスポート(全件) ☒ 検索	<table border="1"> <thead> <tr> <th>コンテンツID</th> <th>コンテンツ名</th> <th>クリア</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> </tbody> </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	☐ エクスポート(全件) ☒ 検索	<table border="1"> <thead> <tr> <th>ルートID</th> <th>ルート名</th> <th>クリア</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> </tbody> </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	☐ エクスポート(全件) ☒ 検索	<table border="1"> <thead> <tr> <th>フローID</th> <th>フロー名</th> <th>クリア</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> </tbody> </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	☐ エクスポート(全件) ☒ 検索	<table border="1"> <thead> <tr> <th>キー</th> <th>キー名</th> <th>クリア</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> </tbody> </table>	キー	キー名	クリア			
キー	キー名	クリア						
ルール定義	☐ エクスポート(全件) ☒ 検索	<table border="1"> <thead> <tr> <th>ルールID</th> <th>ルール名</th> <th>クリア</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> </tbody> </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	☐ エクスポート(全件) ☒ 検索	<table border="1"> <thead> <tr> <th>メールID</th> <th>メール名</th> <th>クリア</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td></tr> </tbody> </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	☐ エクスポート(全件) ☒ 検索							

3. エクスポートしたい定義を選択します。

🏠 コンテンツ定義 - 検索			
🔄 最新情報 🗑️ 閉じる			
コンテンツ名 ▼		検索	
☐	コンテンツID	コンテンツ名	備考
☒	contents_javaee	JavaEE開発モデル	
☐	contents_script	スクリプト開発モデル	
決定			

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	☐ エクスポート(全件) 検索 <table border="1"> <thead> <tr> <th>メールID</th> <th>メール名</th> <th>クリア</th> </tr> </thead> <tbody> <tr> <td>processing</td> <td>処理依頼</td> <td> ✕ </td> </tr> </tbody> </table>	メールID	メール名	クリア	processing	処理依頼	✕
メールID	メール名	クリア					
processing	処理依頼	✕					
IMBox定義	☐ エクスポート(全件) 検索 <table border="1"> <thead> <tr> <th>IMBoxID</th> <th>IMBox名</th> <th>クリア</th> </tr> </thead> <tbody> <tr> <td>act</td> <td>代理通知</td> <td> ✕ </td> </tr> </tbody> </table>	IMBoxID	IMBox名	クリア	act	代理通知	✕
IMBoxID	IMBox名	クリア					
act	代理通知	✕					
一覧表示パターン定義	☐ エクスポート(全件) 検索 <table border="1"> <thead> <tr> <th>対象一覧</th> <th>パターン名</th> <th>クリア</th> </tr> </thead> <tbody> <tr> <td>申請一覧</td> <td>申請標準</td> <td> ✕ </td> </tr> </tbody> </table>	対象一覧	パターン名	クリア	申請一覧	申請標準	✕
対象一覧	パターン名	クリア					
申請一覧	申請標準	✕					
フローグループ定義	☐ エクスポート(全件) ☒ フローとの関連付け情報を含める 検索 <table border="1"> <thead> <tr> <th>フローグループID</th> <th>フローグループ名</th> <th>クリア</th> </tr> </thead> <tbody> <tr> <td>imw_audit_group</td> <td>ワークフロー 監査グループ</td> <td> ✕ </td> </tr> </tbody> </table>	フローグループID	フローグループ名	クリア	imw_audit_group	ワークフロー 監査グループ	✕
フローグループID	フローグループ名	クリア					
imw_audit_group	ワークフロー 監査グループ	✕					
管理グループ定義	☐ エクスポート(全件) ☒ 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 検索 <table border="1"> <thead> <tr> <th>管理グループID</th> <th>管理グループ名</th> <th>クリア</th> </tr> </thead> <tbody> <tr> <td>imw_audit_group</td> <td>ワークフロー 監査グループ</td> <td> ✕ </td> </tr> </tbody> </table>	管理グループID	管理グループ名	クリア	imw_audit_group	ワークフロー 監査グループ	✕
管理グループID	管理グループ名	クリア					
imw_audit_group	ワークフロー 監査グループ	✕					
代理管理者設定	☒ 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。

📁 エクスポート

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。

コラム

エクスポートしたファイルは、ストレージサービスの以下のパス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BPM／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。

エクスポート	アプリケーションID	最新バージョン	アプリケーション名	備考
	5i4dn7tw9tz0ws	1	サンプル_BPM	
	sample_address_chg	1	【サンプル】住所等変更届	IM-Workflow連携用の申請書サンプル
	sample_app_equipment	1	【サンプル】備品管理 (v8)	購入申請で利用できる備品を管理します。
	sample_bihin_mochidashi	1	【サンプル】備品持出申請書	一覧選択アイテムを利用した一般的な申請書サンプル
	sample_enquete	1	【サンプル】アンケート	コピー欄についてのアンケートサンプル
	sample_enquete1	1	【サンプル】受講アンケート	研修受講者向けアンケートのサンプルです。
	sample_keireki	1	【サンプル】業務経歴書	明細アイテムの複数個配置したサンプル
	sample_mailmagazine	1	【サンプル】メールマガジン会員登録	メールマガジン購読会員登録するためのアプリケーションです。
	sample_purchase_application	1	【サンプル】備品購入申請 (IMW)	IM-Workflowで利用できる備品購入申請のサンプルです。
	sample_ringi	1	【サンプル】稟議書	ファイルアップロードアイテムがある、シンプルな申請書サンプル

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。

アプリケーション情報エクスポート

出力先 ☒ ダウンロード ☐ ストレージ storage/

▼ 検索

管理会社名

アプリケーション名

エクスポート	アプリケーションID ↑	最新バージョン	アプリケーション名	備考
	5i4dn7tw9tz0ws	1	サンプル_BPM	

1 ページ中 1 ページ目 15 1 件中 1 - 1 を表示

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。

アプリケーション情報エクスポート

出力先 ☒ ダウンロード ☐ ストレージ storage/

▼ 検索

管理会社名

アプリケーション名

エクスポート	アプリケーションID ↑	最新バージョン	アプリケーション名	備考
	5i4dn7tw9tz0ws	1	サンプル_BPM	

1 ページ中 1 ページ目 15 1 件中 1 - 1 を表示

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BPM／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。

1. 「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。
2. 「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。

データソース情報エクスポート

エクスポートタイプ: ☒ 一括 ☐ 選択

出力先: ☒ ダウンロード ☐ ストレージ storage/

エクスポート実行

- 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。

データソース情報エクスポート

エクスポートタイプ: ☐ 一括 ☒ 選択

出力先: ☒ ダウンロード ☐ ストレージ storage/

☒ すべて
☒ REST ☒ SOAP ☒ JAVA ☒ CSVインポート ☒ CSVエクスポート ☒ Office365_GoogleApps ☒
☒ デナントDBクエリ ☒ シェアードDBクエリ ☒ ルール
 データソース名: 検索

データソースタイプ	データソース名	備考
☐ デナントDBクエリ	備品検索	購入申請の備品情報を取得するためのクエリです。

エクスポート実行

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。

データソース情報エクスポート

エクスポートタイプ: ☒ 一括 ☐ 選択

出力先: ☒ ダウンロード ☐ ストレージ storage/

エクスポート実行

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。

BISエクスポート

☒ BPM ☒ WF

BIS名

エクスポート	BIS作成種類	BIS名 ↑	説明	BIS ID	アプリケーションID	ルートID
	BPM	サンプル_BPM		5i7j4m8tgnw2w6	5i7j4m8tr1ockv6	5i7j4m8tr1ocov6
	BPM	サンプルBPM		5i7j4lnisvaenv6	5i7j4lnj1vjooov6	5i7j4lnj1vjosv6
	WF	サンプルワークフロー		5i7j5z0zum68iv6	5i7j5z10jdax4v6	5i7j5z10jdax8v6

1 ページ中 1 ページ目 15 3 件中 1 - 3 を表示

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。

BISエクスポート

☒ BPM ☒ WF

BIS名

エクスポート	BIS作成種類	BIS名 ↑	説明	BIS ID	アプリケーションID	ルートID
	BPM	サンプル_BPM		5i7j4m8tgnw2w6	5i7j4m8tr1ockv6	5i7j4m8tr1ocov6

1 ページ中 1 ページ目 15 1 件中 1 - 1 を表示

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。

BISエクスポート

☒ BPM ☒ WF

BIS名

エクスポート	BIS作成種類	BIS名	説明	BIS ID	アプリケーションID	ルートID
	BPM	サンプル_BPM		5i7j4m8tgrw2w6	5i7j4m8tr1ockv6	5i7j4m8tr1ocov6

1件中 1 - 1 を表示



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ご注意ください。

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BPM／ワークフローに必要な下記の定義をインポートします。

- インポート対象は、ストレージサービスの以下のパス配下します。
「%PUBLIC_STORAGE_PATH%/im_workflow/data/<テナントID>/import_export/」
- 「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。
- 「インポート」画面が表示されます。

インポート

 ueda_20121219(1).xml
 ueda_20121219(2).xml
 ueda_20121219.xml
 ueda_20121225.xml

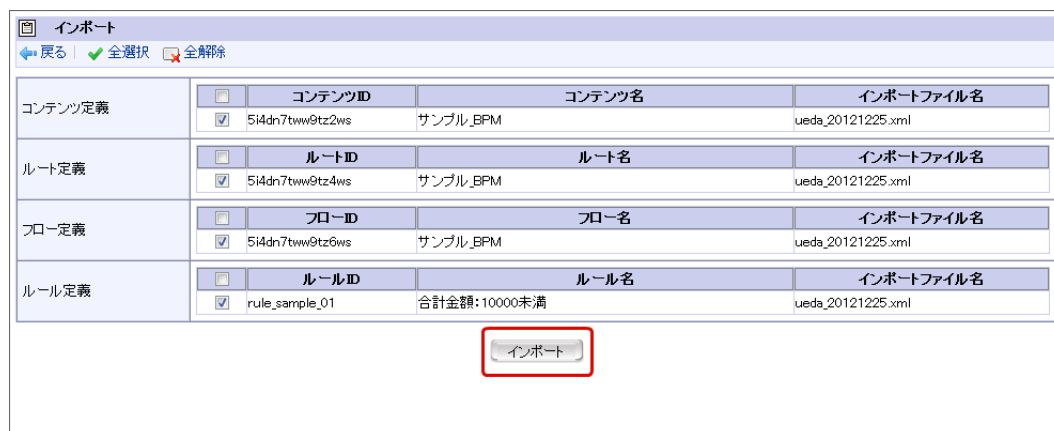
- インポート対象のファイルのチェックボックスをオンにし、「データ選択へ」をクリックします。



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。



6. 「インポート」をクリックしてインポートを実行します。



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BPM／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。

1. 「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。
2. 「アプリケーション情報インポート」画面が表示されます。

アプリケーション情報インポート

インポートファイル*

☒ ローカル
 ファイルを選択 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。

アプリケーション情報インポート

インポートファイル*

☒ ローカル
 ファイルを選択 5i4dn7tw9...5-1707.zip

☐ ストレージ
 storage/

インポート実行

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。

アプリケーション情報インポート

インポートファイル*

☒ ローカル
 ファイルを選択 5i4dn7tw9...5-1707.zip

☐ ストレージ
 storage/

インポート実行

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BPM／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。

データソース情報インポート

☒ ローカル
 ファイルを選択 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。

データソース情報インポート

☒ ローカル
 ファイルを選択 datasource...5-1714.zip

☐ ストレージ
 storage/

インポート実行

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

☒ ローカル
 ファイルを選択 datasource...5-1714.zip

☐ ストレージ
 storage/

インポート実行



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](#)」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
 ファイルを選択 | 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

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。

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
 ファイルを選択 | 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

インポート実行

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。

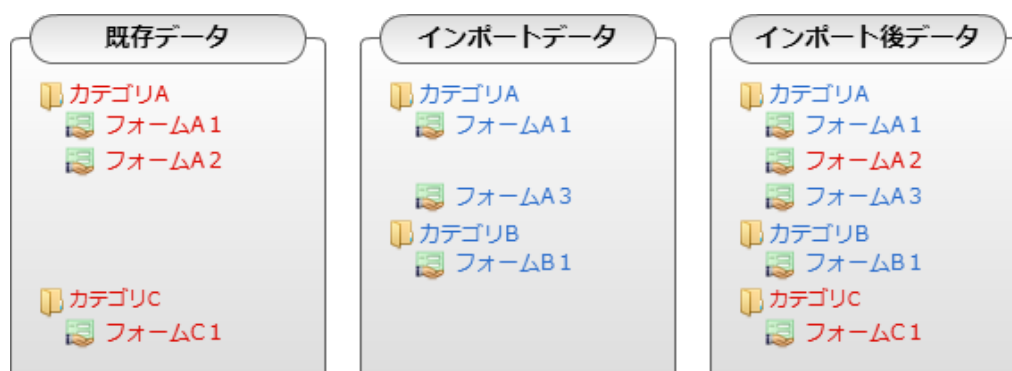
- 上書き

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。

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

ファイルを選択

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

インポート実行

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。

BISインポート

インポートファイル *

ファイルを選択

選択されていません

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。

BISインポート

インポートファイル *

ファイルを選択

514b0lhprny...3-1553.zip

インポート実行

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。

BISインポート

インポートファイル*

ファイルを選択 5idb0lhprmy...3-1553.zip

インポート実行

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。

ルール実行履歴レポート

(1) アプリケーション名

(2) 作成ユーザ名

(3) 作成日

(4) 検索

(5) クリア

(6) 選択したレポートを削除

(7) ダウンロード	(8) アプリケーション名	(9) 作成ユーザ名	(10) 作成日時	(11) データソース名	(12) レポート名
(13)	Flow using RULE	青柳 康巳	2014/09/01 11:31:56	rule report	RuleReport_5ibcq8k47jke2gg.html

1 ページ中 1 ページ目 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BPM／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 BPM の場合は「一括処理対象者変更(BPM)」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。



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。

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				

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
☐	●	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			
☐	●	0000000024	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		2015/11/2			
☐	●	0000000023	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		2015/11/2			
☐	●	0000000022	サンプルフロー (一括処理対象者変更) 2	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			

「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。

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、もしく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
☒		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			
☐		0000000024	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		2015/11/2			
☒		0000000023	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		2015/11/2			
☒		0000000022	サンプルフロー（一括処理対象者変更）2	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			
☐		0000000021	サンプルフロー（一括処理対象者変更）1	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			

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 * 検索

変更先 ×

- 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

申請日	申請者	フロー名	ノード名	状態
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	林政義	サンプルフロー（一括処理対象者変更）	承認	

「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

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。

☒ 名前 ☒ コード ☒ フリガナ

☒ 前方一致 ☐ 完全一致 ☐ 部分一致

検索

上田辰男

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

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」をチェックしてください。

変更元 * 青柳辰巳 検索 変更先 * 上田辰男 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
☒	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。

intra-mart® Top Workflow IM-BIS サンプル サイトマップ 上田辰男 ?

未処理 (ワークフロー)

BPM 連続処理 一括処理 表示条件

本人

申請 承認

代理

申請 承認

処理	振替	優先度	案件番	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
		●	00000 00025	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		▶		
		●	00000 00023	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		▶		
		●	00000 00022	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		▶		

IM-BIS の詳細カスタマイズ機能

ここでは、IM-BIS のBPM／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、BPM／ワークフローで印影を表示するためのアイテムです。

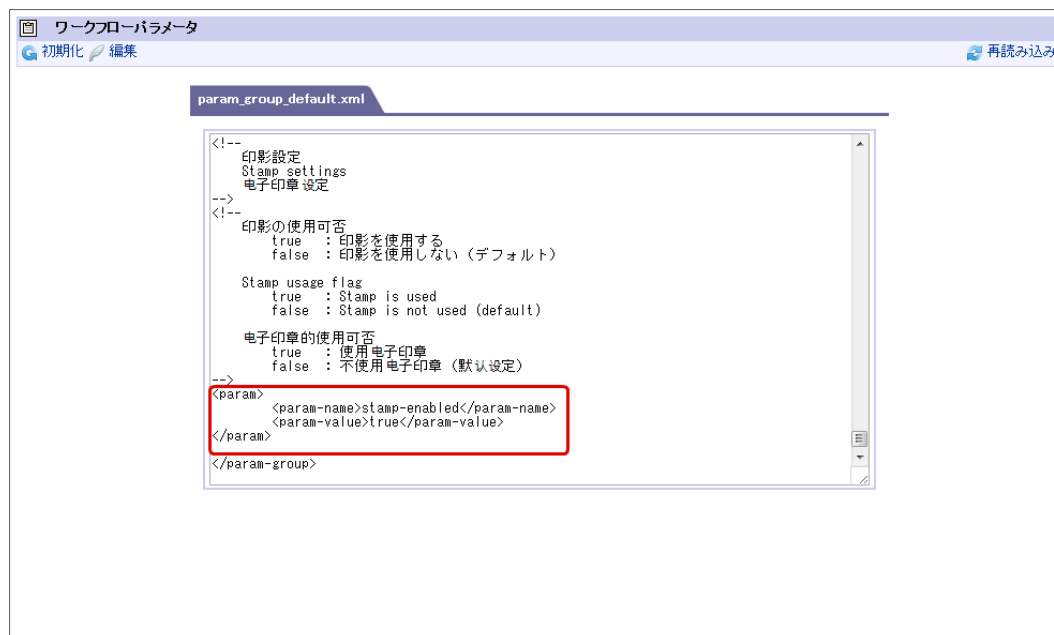
申請・承認処理などで、印影を利用できます。

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。

ワークフローパラメータ

初期化 編集 再読み込み

param_group_default.xml

```

<!--
  印影設定
  Stamp settings
  電子印章設定
-->
<!--
  印影の使用可否
  true  : 印影を使用する
  false : 印影を使用しない (デフォルト)

  Stamp usage flag
  true  : Stamp is used
  false : Stamp is not used (default)

  電子印章の使用可否
  true  : 使用電子印章
  false : 不使用電子印章 (默认設定)
-->
<param>
  <param-name>stamp-enabled</param-name>
  <param-value>true</param-value>
</param>
</param-group>

```

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。

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
	5i4dn7tw9tz2ws	サンプル_BPM	
	contents_javaee	JavaEE開発モデル	
	contents_script	スクリプト開発モデル	

1-3/3

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	5i4dn7tw9tz2ws
コンテンツ名 (必須)	英語 sample_BPM 日本語 サンプル_BPM 中国語 样品_BPM
備考	英語 日本語 中国語

更新 削除

4. 対象のバージョンの「編集」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		
	2012/12/25	2999/12/31		

5. 「ユーザプログラム」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: 5i4dn7tw9tz2ws

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): 2012/12/25 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」をクリックして、「印影処理」のプログラムを設定します。

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: 5i4dn7tw9tz2ws

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	プログラム名	プラグイン種類	対象ノード	備考	初期使用
	BAMアクション処理	【アクション処理】スクリプト開発モデル	承認/処理ノード		
	BAMアクション処理	【アクション処理】スクリプト開発モデル	申請/処理開始ノード		
	BAM案件終了処理	【案件終了処理】スクリプト開発モデル			
	案件プロパティ設定処理	【アクション処理】スクリプト開発モデル	承認/処理ノード		
	案件プロパティ設定処理	【アクション処理】スクリプト開発モデル	申請/処理開始ノード		
	承認アクション処理	【アクション処理】スクリプト開発モデル	承認/処理ノード		
	申請アクション処理	【アクション処理】スクリプト開発モデル	申請/処理開始ノード		

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BPM/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。

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】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	☒ 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	☒ 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BPM/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。

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、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	☒ 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	☒ 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

ユーザープログラム定義 - 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	☒ 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

ユーザープログラム定義 - 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	☒ 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	☒ 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	☒ 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

BPM／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 紐込みキーワード ▼ 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。

印影設定 - 作成画面

←

印影設定 - 作成画面

印影名	常用印
	☒ 常用印として利用する
紐込みキーワード	
	キーワード一覧
印影タイプ	☒ 3段表示 ☐ 縦表示 ☐ 縦表示
上段	青柳辰巳 ☐ 日付
中段	12/12/26 ☒ 日付
下段	☐ 日付
印影プレビュー	

以上で、BIS担当者が利用する印影が設定できました。

ここでは、BPM／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、BPM／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BPM／ワークフローの申請・処理が対象となります。
(設定前に申請・処理が開始されたBPM／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。



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となります。

参照者を設定する

「参照(BPM)」「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BPM／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

編集	フローID	フロー名	備考
	5i4dn7tw9tz6ws	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**

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

[更新](#) [削除](#)

4. 設定するBPM／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。

フロー定義 - バージョン - 編集	
戻る フローコピー バージョンコピー	
フローID	5i4dn7tww9tz6ws
フロー名	サンプル_BPM
基本情報 参照者 コンテンツ詳細 ルート詳細	
バージョン期間 (必須)	2012/12/25 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
コンテンツ (必須)	サンプル_BPM 検索
ルート (必須)	サンプル_BPM 検索
カレンダー	検索 クリア
機能設定	ファイルの添付 ☒ 有効 ☐ 無効 一括処理 ☐ 有効 ☒ 無効 ☐ 全てのノードに設定 一括確認 ☐ 有効 ☒ 無効 完了した案件の確認 ☐ 有効 ☒ 無効 自動処理 ☐ 有効 ☒ 無効 ☐ 全てのノードに設定 自動催促 ☐ 有効 ☒ 無効 ☐ 全てのノードに設定
	更新 削除

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。

フロー定義 - バージョン - 編集	
戻る フローコピー バージョンコピー	
フローID	5i4dn7tww9tz6ws
フロー名	サンプル_BPM
基本情報 参照者 コンテンツ詳細 ルート詳細	
参照者	検索
権限設定	対象種別 対象名 クリア
更新	

7. 更新をクリックします。

フロー定義 - バージョン - 編集	
戻る フローコピー バージョンコピー	
フローID	5i4dn7tww9tz6ws
フロー名	サンプル_BPM
基本情報 参照者 コンテンツ詳細 ルート詳細	
参照者	検索
権限設定	対象種別 対象名 クリア ☒ ユーザ 上田辰男 ☐
更新	

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のノードに対して一括で設定されます。



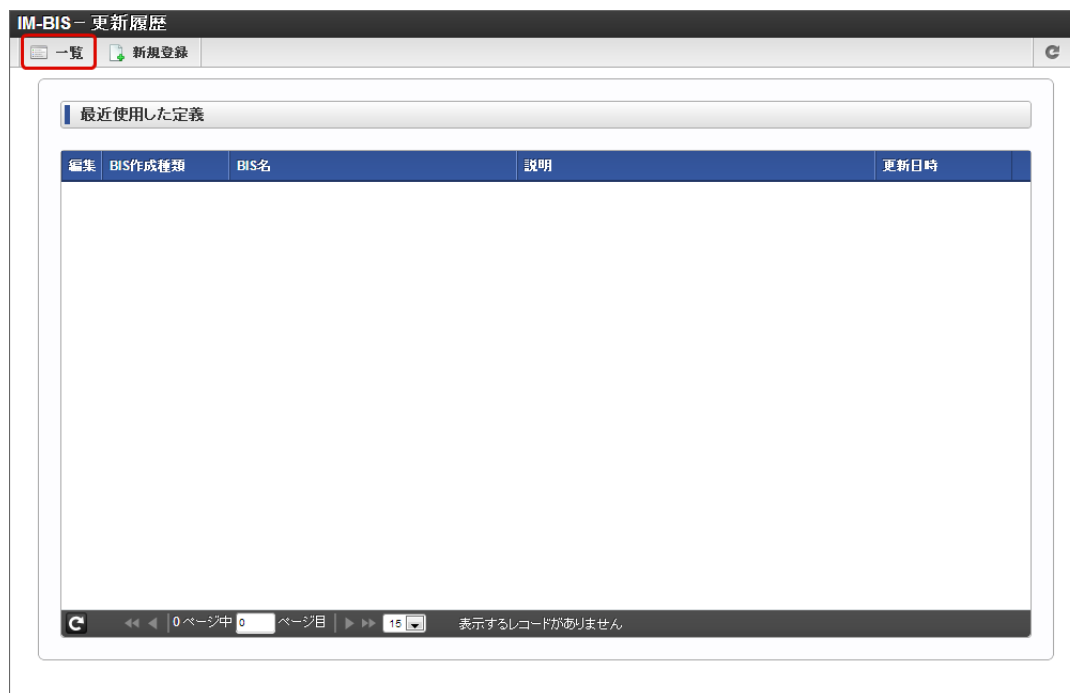
コラム

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。

フォーム設定

←

アプリケーションID 5i7j1ej3ogxyiv6
アプリケーション名 サンプルBPM

アプリケーション情報 フォーム設定 テーブル設定 WF連携設定

登録

アプリケーション履歴一覧

編集	アプリケーション履歴番号 ↑	有効日付(開始)	有効日付(終了)	備考	更新日
	1	2013/01/11	2999/12/31		2013/01/15

1 ページ中 1 ページ目 1 件中 1 - 1 を表示

5. 「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 5i7j1ej3ogxyiv6
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

編集	フォーム編集	プレビュー	フォーム名	フォーム備考
			メインフォーム	
			申請/処理開始	

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: 5i7j1ej3ogxyiv6
 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
			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？

決定 取り消し



注意

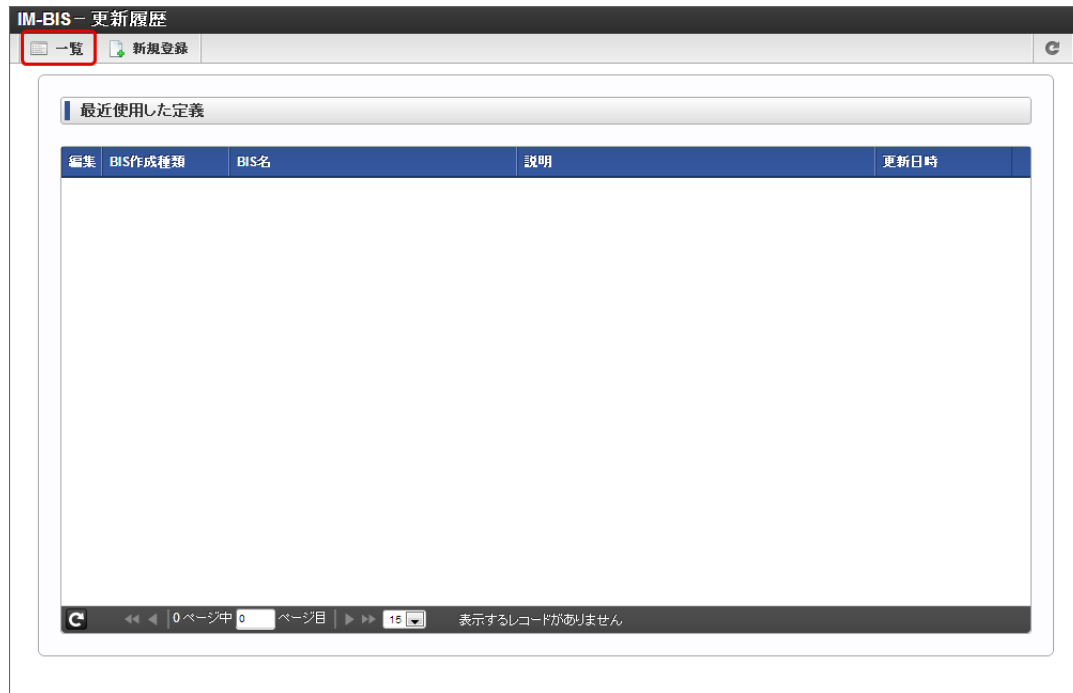
上記の設定後に、「フロー編集」で共有設定等の変更を行うと、設定内容が変更されてしまう場合があります。

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。



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。

フォーム設定

←

アプリケーションID 5i7j1ej3ogxyiv6
アプリケーション名 サンプルBPM

アプリケーション情報 フォーム設定 テーブル設定 WF連携設定

登録

アプリケーション履歴一覧

編集	アプリケーション履歴番号	有効日付(開始)	有効日付(終了)	備考	更新日
	1	2013/01/11	2999/12/31		2013/01/15

1 ページ中 1 ページ目 1 件中 1 - 1 を表示

5. 「登録」をクリックします。

フォーム一覧

←

アプリケーションID 5i7j1ej3ogxyiv6
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

編集	フォーム編集	プレビュー	フォーム名	フォーム備考
			メインフォーム	
			申請/処理開始	

6. フォーム名を入力し、「登録」をクリックします。

フォーム登録

日本語

フォーム名 * 特殊ノード用フォーム

フォーム備考 特殊ノード用フォーム 備考

英語

フォーム名 * Special Node Form

フォーム備考 Special Node Form Remarks

中国語

フォーム名 * Special Node Form

フォーム備考 Special Node Form Remarks

登録

7. アイテムコピーをクリックします。

フォーム編集

更新 画像アップロード ラベル一覧 フィールド一覧 グリッド 枠線 再利用 H ヘッダーとフッター アクション設定

ツールキット アイテムコピー 日本語

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。

フォーム一覧

←

アプリケーションID 5i7j1ej3ogxyiv6

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

編集	フォーム編集	プレビュー	フォーム名	フォーム備考
			メインフォーム	
			申請/処理開始	
			特殊ノード用フォーム	特殊ノード用フォーム 備考

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。

フォーム 遷移一覧

←

アプリケーションID 5i7j1ej3ogxyiv6

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
			申請/処理開始	1

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。

フォーム遷移詳細編集

←

アプリケーションID	5i7j1ej3ogxyiv6
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
メインフォーム		

+ 追加

更新

13. 「追加」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	5i7j1ej3ogxyiv6
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
+ 追加		

更新

14. 先の手順で作成したフォームを選択します。

フォーム検索

フォーム名 検索 クリア

選択	プレビュー	フォーム名	アプリケーションID ↑	アプリケーション名	アプリケーション履歴番号
☐		メインフォーム	5i7j1ej3ogxyiv6	サンプルBPM	1
☐		申請/処理開始	5i7j1ej3ogxyiv6	サンプルBPM	1
☐		特殊ノード用フォーム	5i7j1ej3ogxyiv6	サンプルBPM	1

1 ページ中 1 ページ目 15 3 件中 1 - 3 を表示

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	5i7j1ej3ogxyiv6
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	5i7j1ej3ogxyiv6
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作成種類	BIS名	説明	BIS ID	フローID	アプリ
	WF	子画面サンプル		5ieovgkb6stx0pd	5ieovgkbjqb1lpd	

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。

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作成種類	BIS名	説明	BIS ID	フローID	アプリ
	WF	子画面サンプル		5ieovgkb6stx0pd	5ieovgkbjqb1lpd	

3. 「アプリケーション履歴一覧」の「編集」をクリックします。

フォーム設定

←

アプリケーションID 5ieovgkbjgb1fpd
アプリケーション名 子画面サンプル

アプリケーション情報 フォーム設定 テーブル設定 WFL連携設定

登録

アプリケーション履歴一覧

編集	アプリケーション履歴番号	有効日付(開始)	有効日付(終了)	備考	更新日
	1	2015/03/16	2999/12/31		2015/03/16

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。

フォーム一覧

←

アプリケーションID 5ieovgkbjgb1fpd
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	

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。

フォーム登録

←

フォーム登録

日本語

フォーム名* 子画面サンプル
フォーム備考

英語

フォーム名* 子画面サンプル
フォーム備考

中国語 (中華人民共和国)

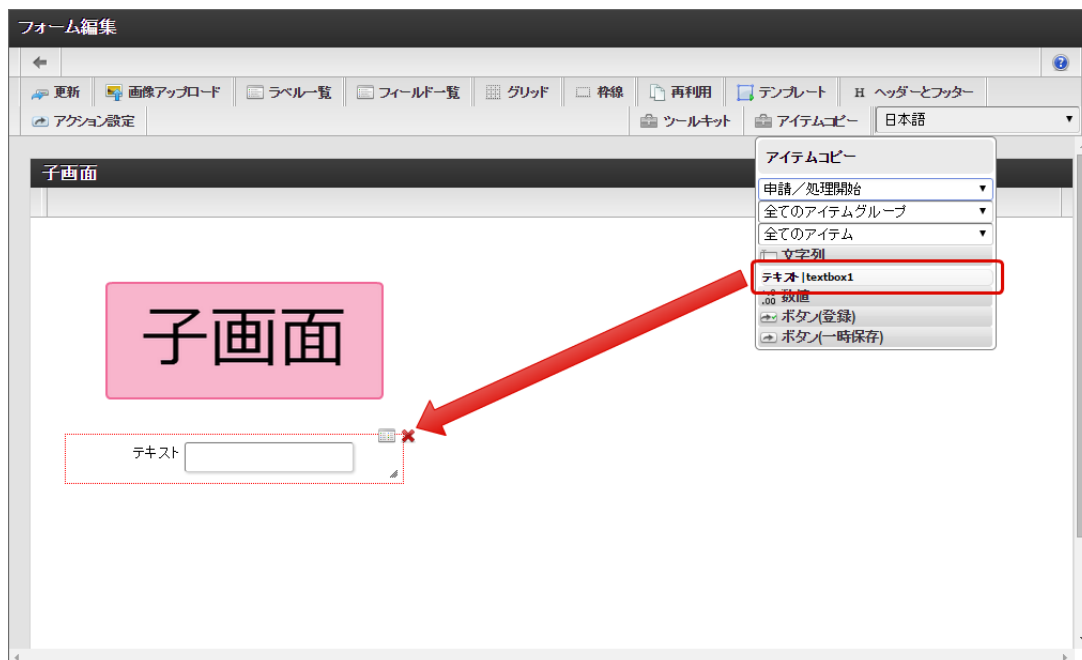
フォーム名* 子画面サンプル
フォーム備考

登録

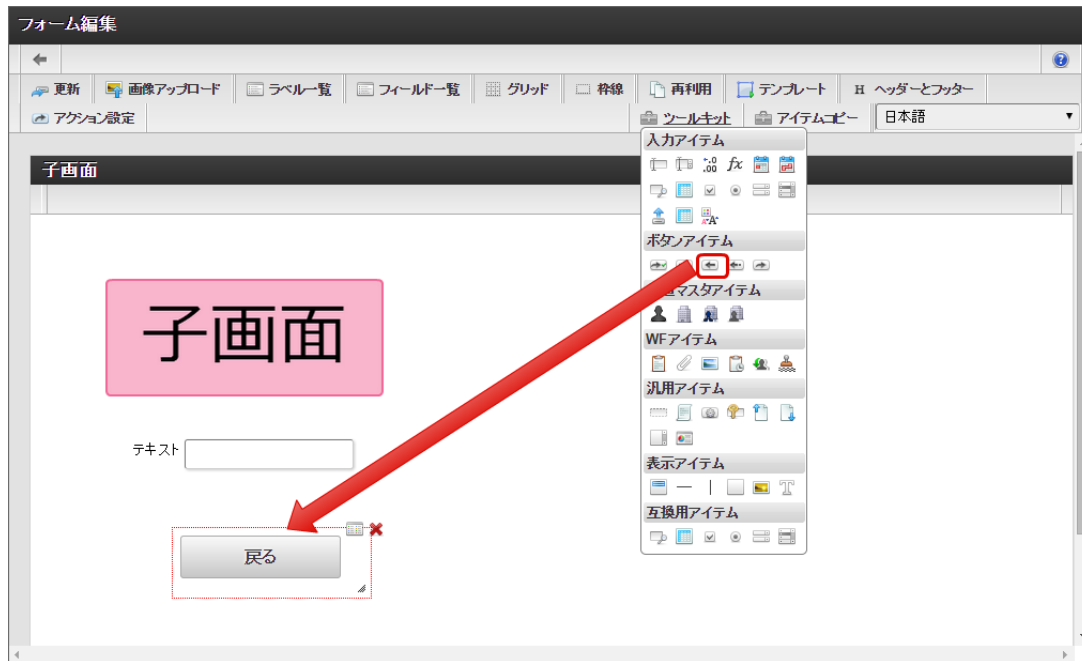
6. 子画面のレイアウトを設定し、「更新」をクリックしてフォームを保存します。



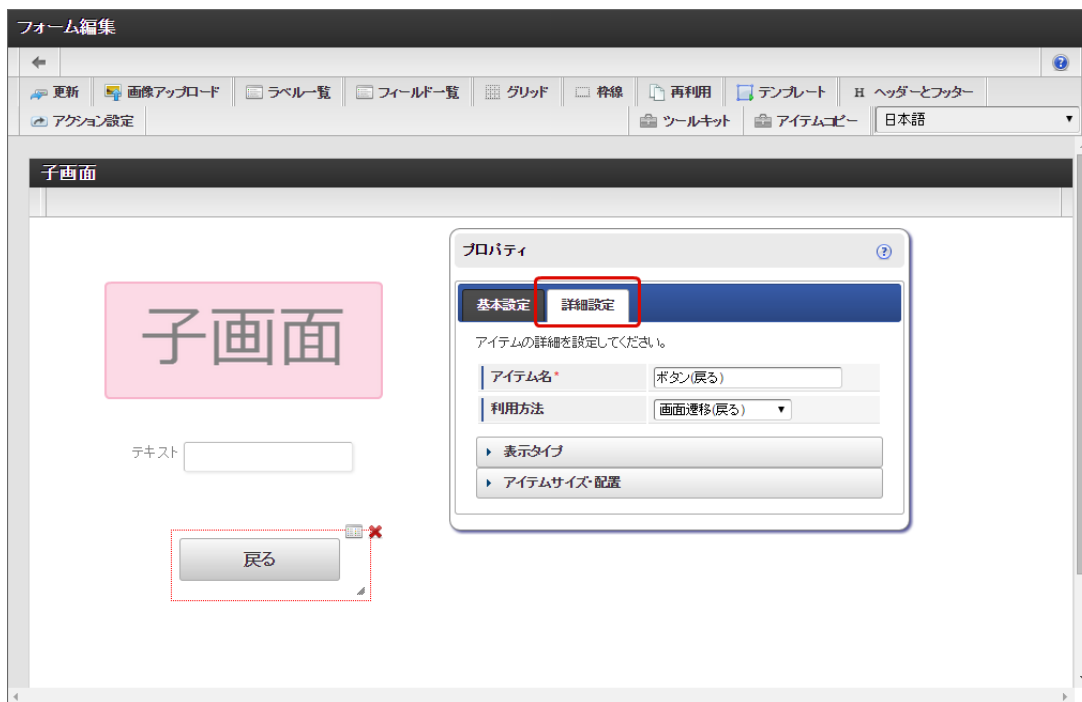
7. 呼び出し元画面と値をやり取りしたい場合には、アイテムコピーを利用して、フィールド識別IDが同じアイテムを配置します。



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。

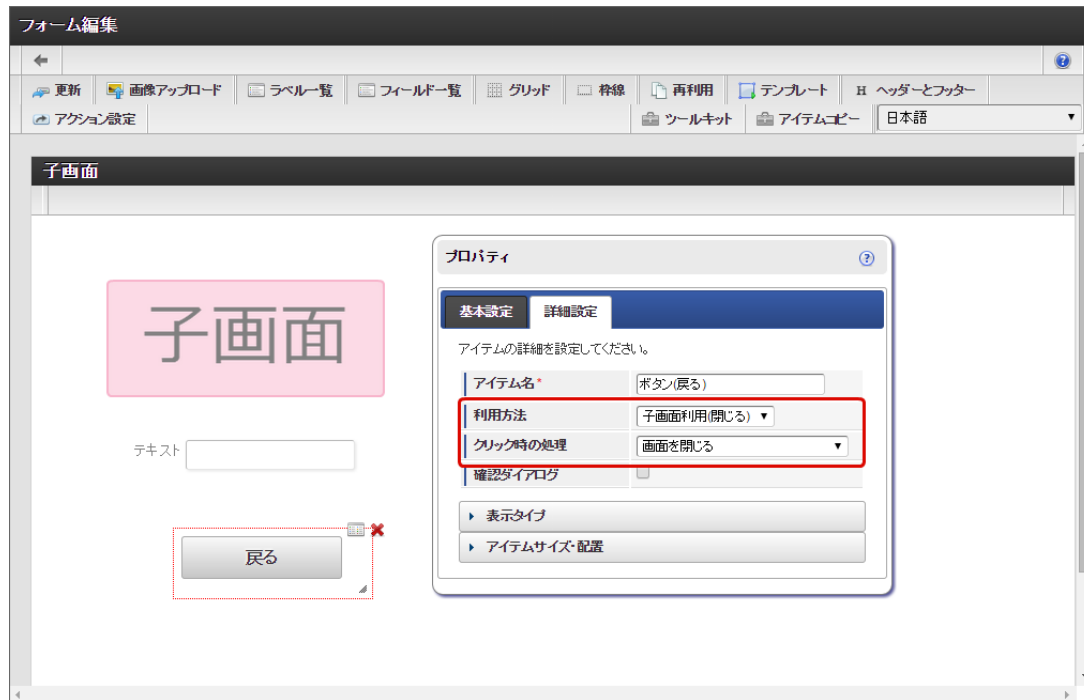


9. 配置した「戻る」ボタンのプロパティの「詳細設定」を表示します。

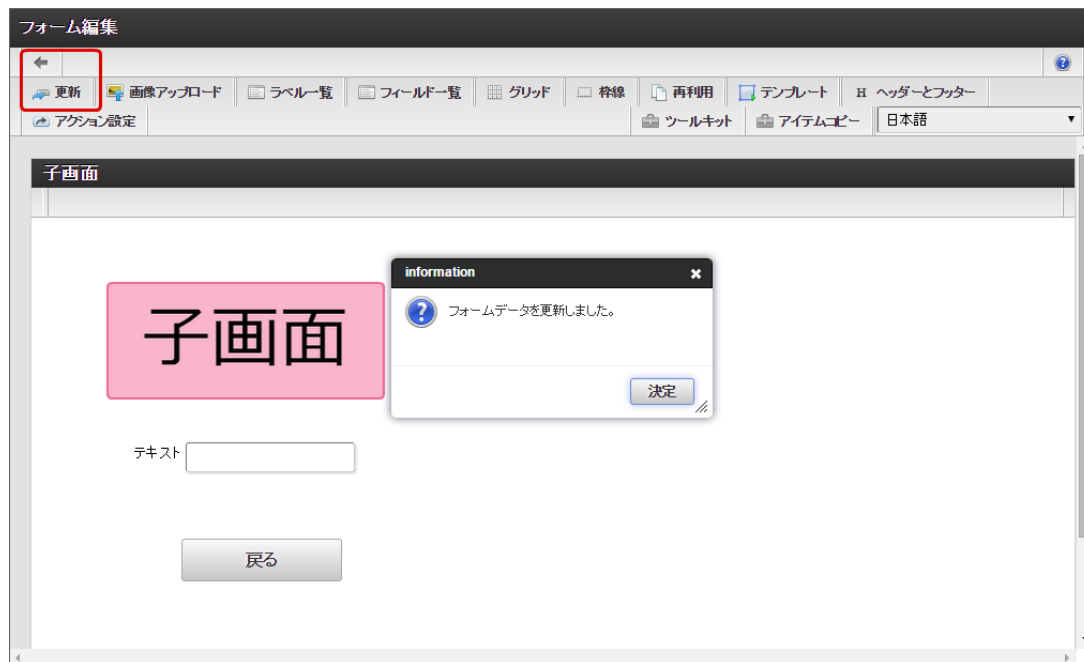


10. 利用方法を「子画面利用（閉じる）」に変更します。

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。



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: 5ieovgkbjgb1fpd
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: 5ieovgkbjgb1fpd
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* 子画面遷移

英語

フォーム遷移名* 子画面遷移

中国語 (中華人民共和国)

フォーム遷移名* 子画面遷移

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。

フォーム遷移詳細編集

←

アプリケーションID	5ieovgkbjqb1fpd
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
フォーム遷移名	子画面遷移

遷移方法 * ☒ 画面遷移 ☐ タブ切替

フォーム名	プレビュー	削除
+ 追加		

更新

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。

フォーム検索

フォーム名 検索 クリア

選択	プレビュー	フォーム名	アプリケーションID	アプリケーション名	アプリケーション履歴番号
☐		子画面サンプルフォ	5ieovgkbjqb1fpd	子画面サンプル	1
☐		申請/処理開始	5ieovgkbjqb1fpd	子画面サンプル	1
☒		子画面サンプル	5ieovgkbjqb1fpd	子画面サンプル	1

1 ページ中 1 ページ目 15 3 件中 1 - 3 を表示

17. 「更新」をクリックして、フォーム遷移を保存します。

フォーム遷移詳細編集

←

アプリケーションID	5ieovgkbjqb1fpd
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
フォーム遷移名	子画面遷移

遷移方法 * ☒ 画面遷移 ☐ タブ切替

フォーム名	プレビュー	削除
子画面サンプル		-

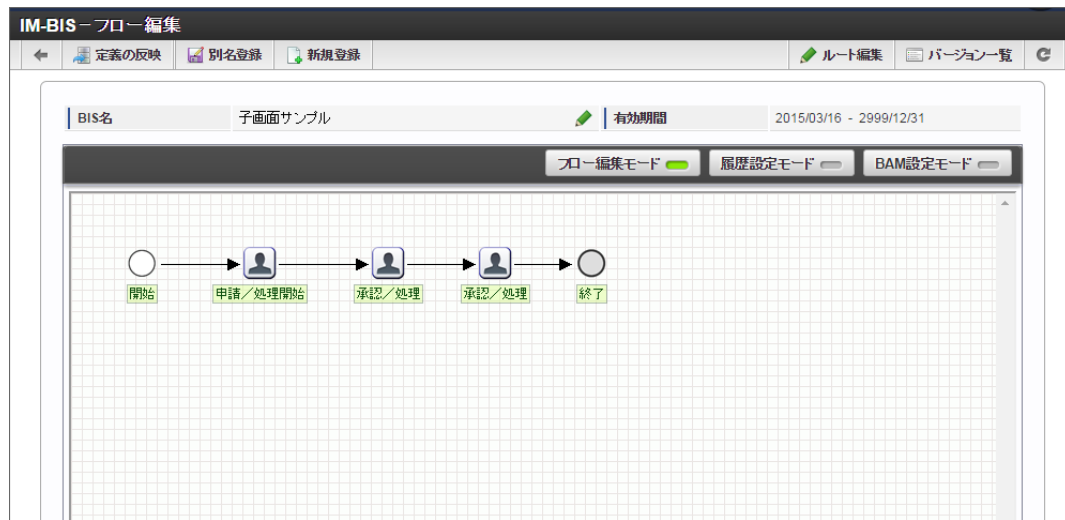
+ 追加

更新

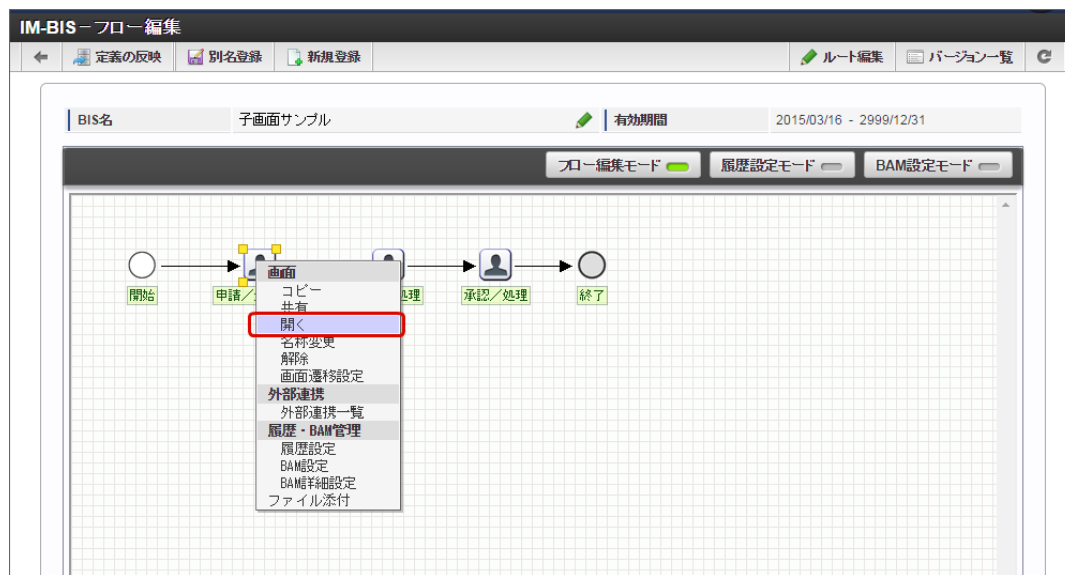
18. 以上で、呼び出す対象の子画面とフォーム遷移を設定できました。
続いて、呼び出し元の画面の設定を行います。

呼び出し元画面の設定を行う

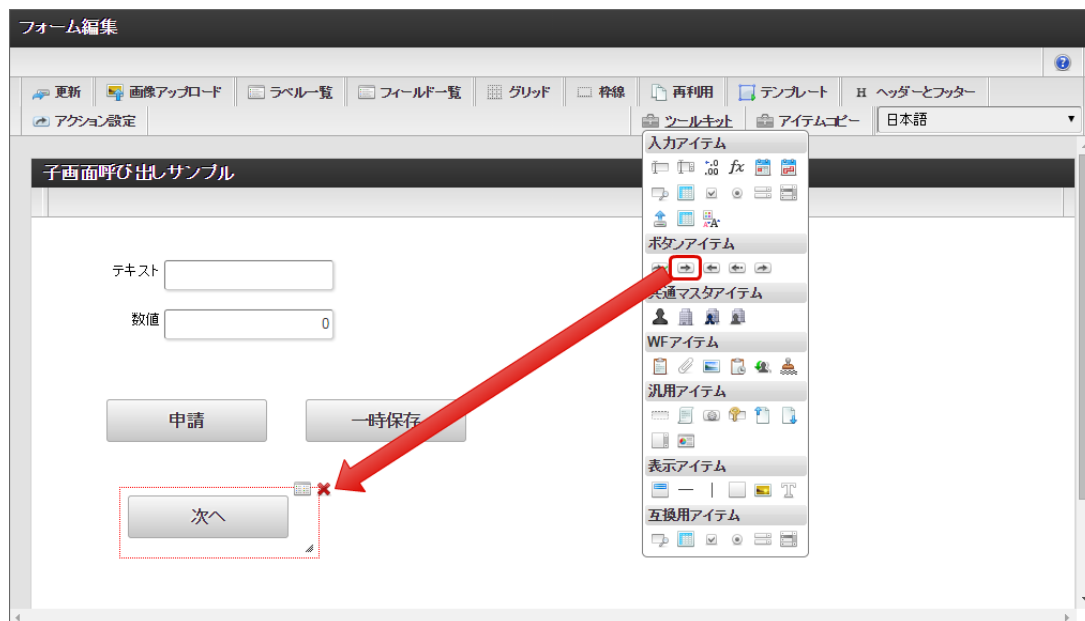
1. 編集中的フローの「フロー編集」画面を表示します。



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。



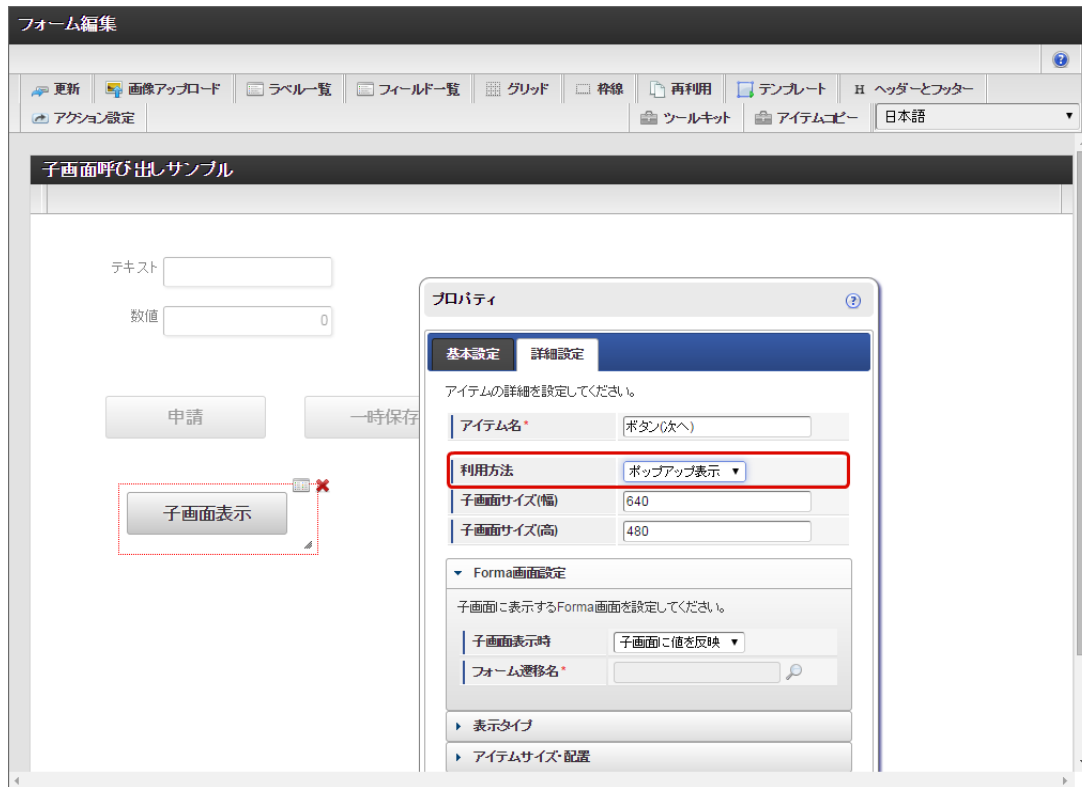
4. 配置した「次へ」ボタンのプロパティを表示し、ラベルを「子画面表示」に変更します。



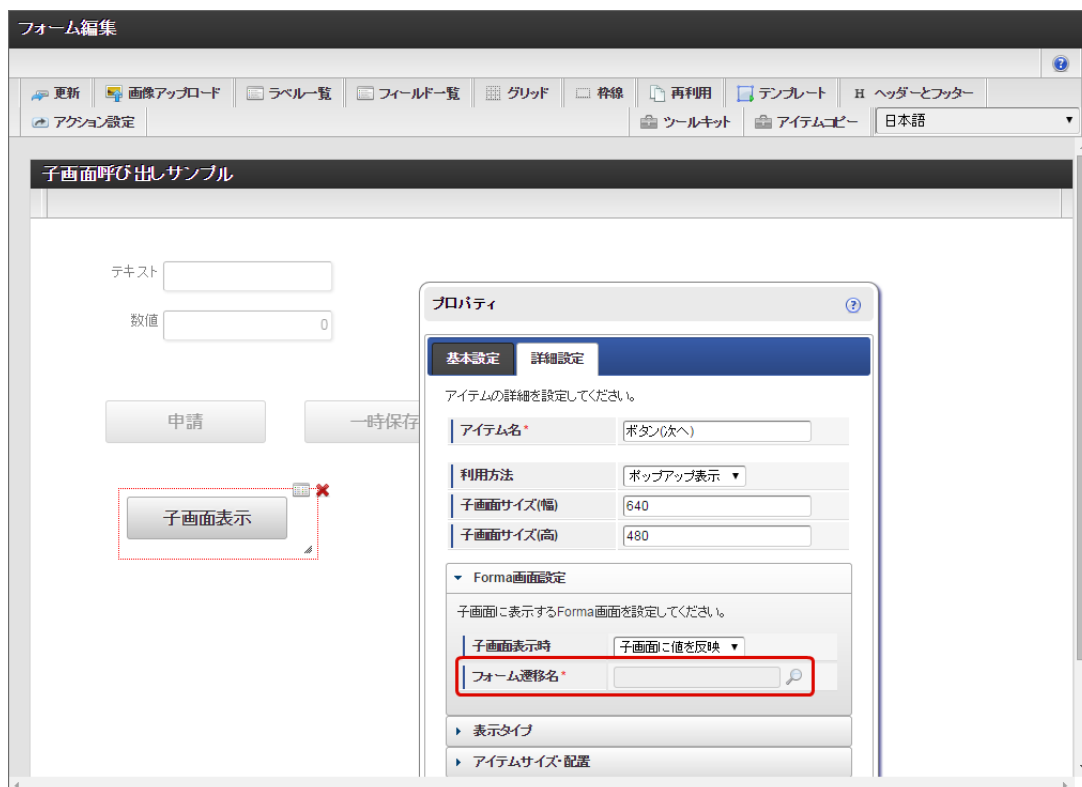
5. 「次へ」ボタンのプロパティの「詳細設定」を表示します。



6. 「利用方法」を「ポップアップ表示」に変更します。



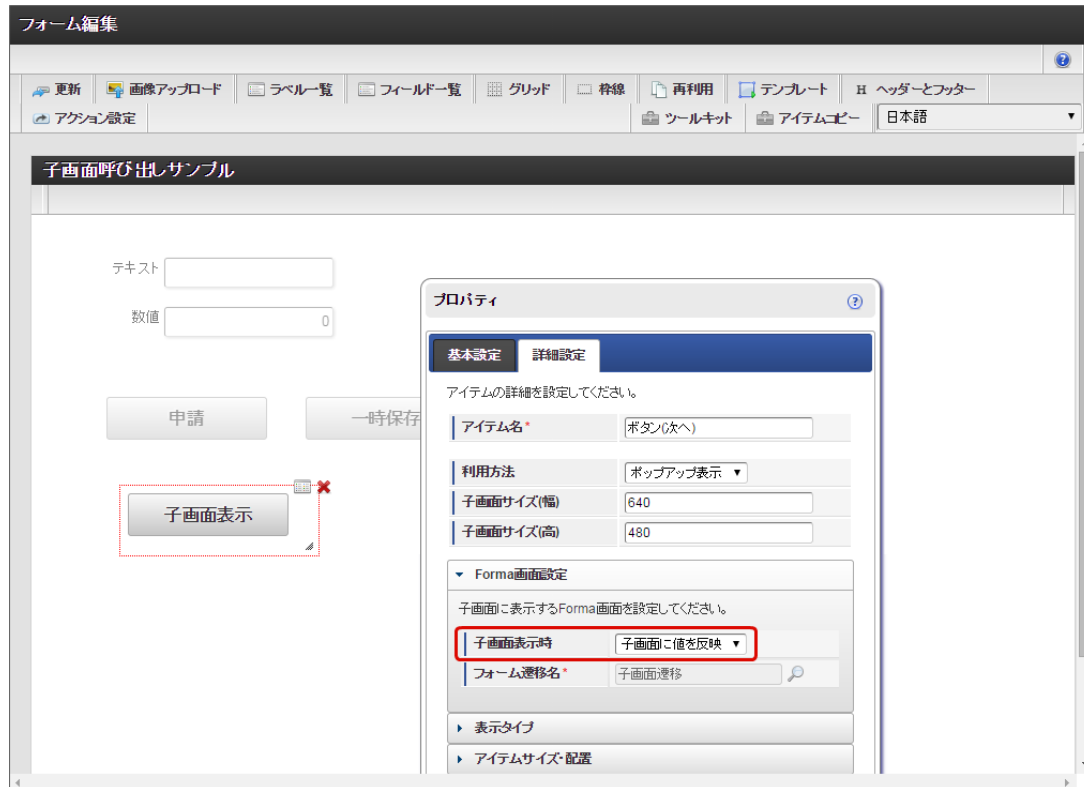
7. 「Forma画面設定」の「フォーム遷移名」の 🔍 をクリックします。



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。

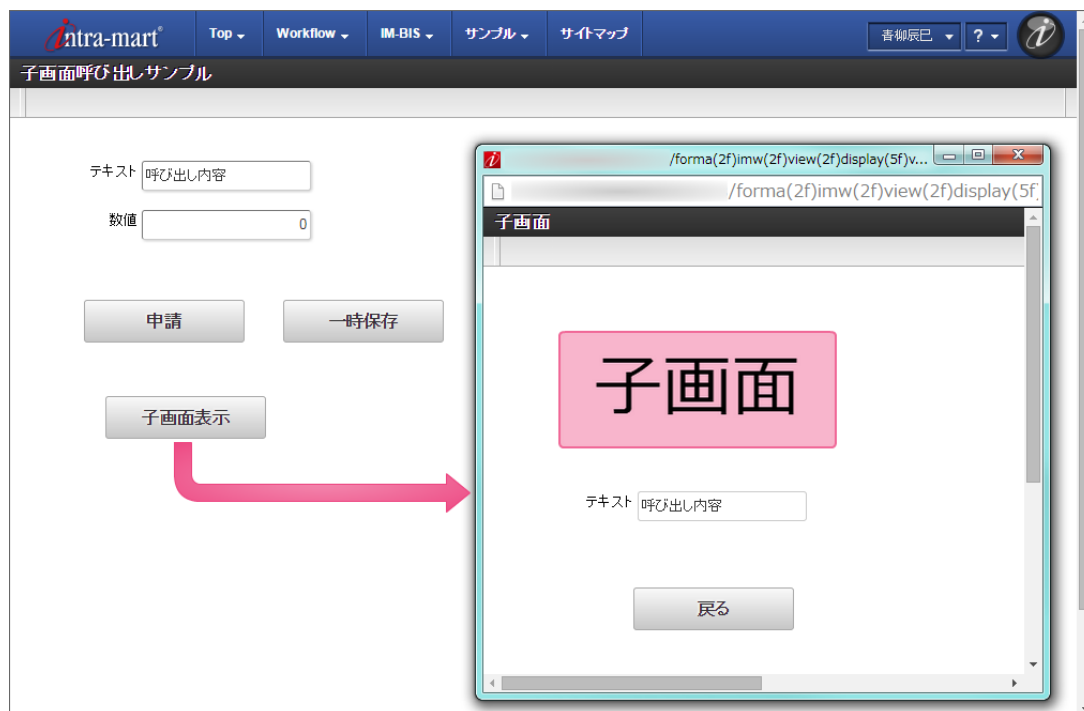


10. 「更新」をクリックしてフォームを保存します。



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](#)

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

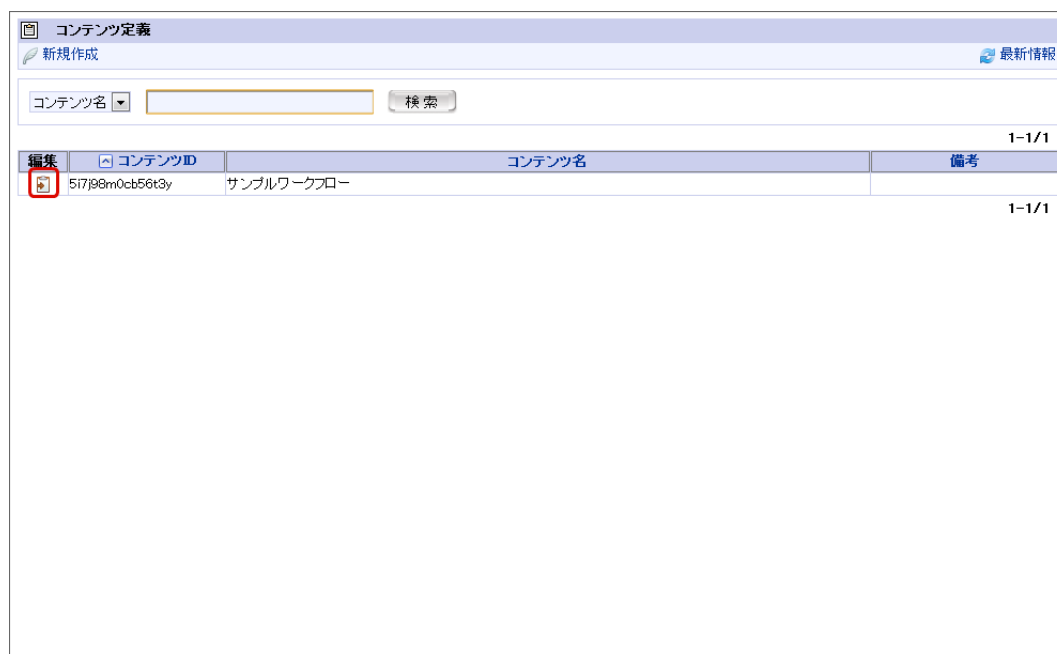
IM-BIS のフローの処理時にメールを送信するための設定を行います。

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。

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。



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: 5i7j98m0cb56t3y

コンテンツ名 (必須)

英語: Sample Workflow

日本語: サンプルワークフロー

中国語: Sample Workflow

備考

英語:

日本語:

中国語:

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		
	2013/01/28	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	5i7j98m0cb56t3y
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	5i7j98m0cb56t3y								
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									
<input checked="" type="button" value="新規作成"/>									
編集	<table border="1"> <thead> <tr> <th>メール名</th> <th>メール種別</th> <th>標準</th> <th>初期使用</th> </tr> </thead> <tbody> <tr> <td colspan="4" style="height: 100px;"> </td> </tr> </tbody> </table>	メール名	メール種別	標準	初期使用				
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。

メール定義 - 検索 最新情報 閉じる

メール種別 | メール名 検索

1-4/4

選択	メールID	メール名	メール種別	備考
	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
	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
	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
	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

1-4/4

9. 「登録」をクリックして内容を保存します。

メール定義 - 新規作成 閉じる

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	☒ 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。

編集	コンテンツID	コンテンツ名	備考
	51798lrOptbw3y	サンプルBPM	

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	5i7j98lr0pfbw3y		
コンテンツ名 (必須)	英語	Sample BPM	
	日本語	サンプルBPM	
	中国語	Sample BPM	
備考	英語		
	日本語		
	中国語		

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		
	2013/01/28	2999/12/31		

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	5i7j98lr0pfw3y
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	5i7j98lr0pfw3y								
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									
<input checked="" type="button" value="新規作成"/>									
編集	<table border="1"> <thead> <tr> <th>IMBox名</th> <th>IMBox種別</th> <th>標準</th> <th>初期使用</th> </tr> </thead> <tbody> <tr> <td colspan="4" style="height: 100px;"></td> </tr> </tbody> </table>	IMBox名	IMBox種別	標準	初期使用				
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。

IMBox定義 - 検索 最新情報 閉じる

IMBox種別 | IMBox名 検索

1-4/4

選択	IMBoxID	IMBox名	IMBox種別	備考
	confirm	確認依頼	確認依頼	確認依頼用の標準IMBoxテンプレート
	processing	処理依頼	処理依頼	処理依頼用の標準IMBoxテンプレート
	reference	参照依頼	参照依頼	参照依頼用の標準IMBoxテンプレート
	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

1-4/4

9. 「登録」をクリックして内容を保存します。

IMBox定義 - 編集 閉じる

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	☒ 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。

