



目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. intra-mart WebPlatform / AppFramework と intra-mart Accel Platform の併用運用について
 - 2.2. IM-FormaDesigner への移行について
 - 2.3. 移行手順概要
- 3. 移行手順
 - 3.1. 作業内容概要
 - 3.2. intra-mart WebPlatform / AppFramework と intra-mart Accel Platform の併用運用環境構築
 - 3.3. VisualDesigner のドキュメントから FormaDesigner のアプリケーションのひな形を生成
 - 3.4. IM-FormaDesigner アプリケーションの開発および検証する手順
 - 3.5. IM-VisualDesigner → IM-FormaDesigner 切替手順
- 4. 付録
 - 4.1. 移行支援ツールの内部仕様
 - 4.2. フォーム変換ルール
 - 4.3. IM-VisualDesigner と IM-FormaDesignerの機能比較

変更年月日	変更内容
2014-09-01	初版
2016-08-01	第2版 下記を変更・追加しました。 <ul style="list-style-type: none">■ 「はじめに」にトランザクションデータやファイルアップロードのデータの保存先に関する説明を追加しました。

本書では intra-mart WebPlatform / AppFramework と intra-mart Accel Platform の併用運用環境において、IM-VisualDesigner のドキュメントから IM-FormaDesigner for Accel Platform のアプリケーションへ移行する際の移行手順および開発手順について説明します。登録データの移行については対象外です。

intra-mart WebPlatform / AppFramework と intra-mart Accel Platform の併用運用について

intra-mart WebPlatform / AppFramework と intra-mart Accel Platform を併用運用します。

- intra-mart WebPlatform / AppFramework 上では、IM-VisualDesigner から作成した申請書の運用を引き続き行います。intra-mart Accel Platform 上で IM-FormaDesigner for Accel Platform へ移行した申請書については過去の案件の参照のみ行います。
- intra-mart Accel Platform 上では IM-VisualDesigner からエクスポートした情報を基に IM-FormaDesigner for Accel Platform へ申請書を移行します。移行が完了した申請書については IM-FormaDesigner for Accel Platform 上で運用します。

また、併用運用環境の構築については intra-mart Accel Platform で用意されているドキュメントの手順に従って実施してください。

IM-FormaDesigner への移行について

本書の移行手順に従い、移行支援ツールを用いることで、intra-mart WebPlatform 上で運用している IM-VisualDesigner Ver.7.2（以下、IM-VisualDesigner）からエクスポートした既存ドキュメントを IM-FormaDesigner for Accel Platform（以下、IM-FormaDesigner）へ移行することができます。

移行の対象は以下の通りです。

- IM-VisualDesigner 標準ドキュメント

IM-VisualDesigner のドキュメントでワークフロー連携していないドキュメントです。

- IM-VisualDesigner ワークフロー連携ドキュメント(IM-Workflow)

IM-Workflow と連携しているIM-VisualDesigner のドキュメントです。



コラム

移行後は、過去に申請・処理された案件については intra-mart WebPlatform 上の IM-VisualDesigner で閲覧し、新規の申請を intra-mart Accel Platform 上の IM-FormaDesigner の画面から行えます。

- IM-VisualDesigner ワークフロー連携ドキュメント(BPW)

ドキュメントワークフロー（BPW）と連携しているIM-VisualDesigner のドキュメントです。

IM-FormaDesigner はドキュメントワークフロー（BPW）との連携に対応していないため、ドキュメントワークフロー(BPW)のフロー定義は、新規に IM-Workflow のフロー定義として定義しなおします。



コラム

移行後は、過去に申請・処理された案件については intra-mart WebPlatform 上の IM-VisualDesigner で閲覧し、新規の申請を IM-Workflow のフロー定義として intra-mart Accel Platform 上の IM-FormaDesigner の画面から行えます。

- IM-VisualDesigner BPM連携ドキュメント

im-BPMと連携しているIM-VisualDesigner のドキュメントです。

intra-mart Accel Platform 上でim-BPMが非サポートのため、intra-mart Accel Platform 上で利用することはできません。

移行前 => 移行後

移行前	=> 移行後
IM-VisualDesigner 標準ドキュメント	=> IM-FormaDesigner - 標準
IM-VisualDesigner ワークフロー連携ドキュメント(IM-Workflow)	=> IM-FormaDesigner - IM-Workflow
IM-VisualDesigner ワークフロー連携ドキュメント(BPW)	=> IM-FormaDesigner - IM-Workflow

i コラム

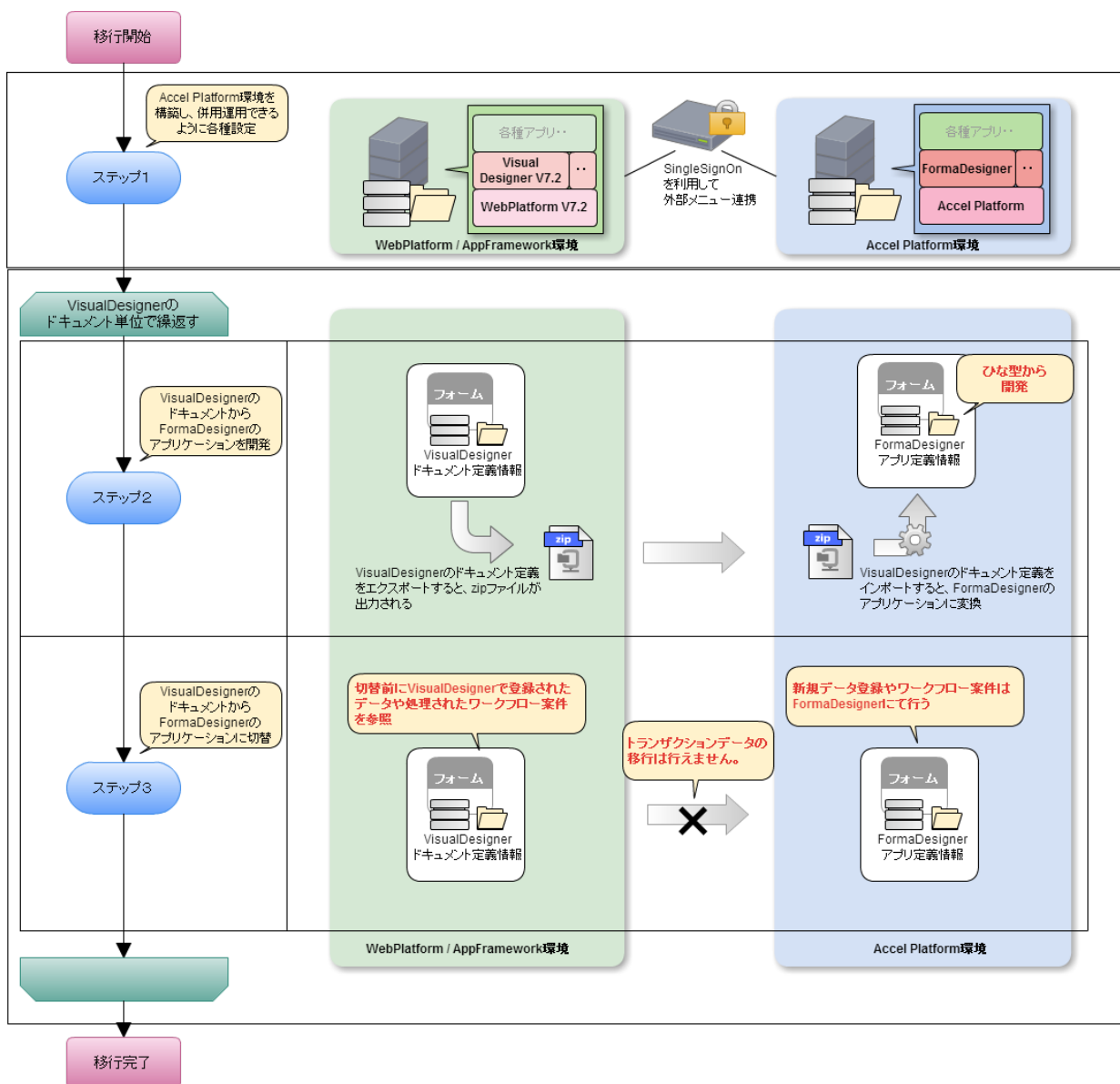
intra-mart Accel Platform 2016 Summer(8.0.14)、IM-FormaDesigner 2016 Summer(8.0.13)以降を併用運用する場合には、本書の移行作業がすべて完了したらワークフローのトランザクションデータやファイルアップロードのデータの保存先を切り替えることができます。

切替が可能な対象は、intra-mart Accel Platform 上で IM-FormaDesigner と IM-Workflow を連携して申請された案件です。手順の詳細については、以下のドキュメントを参照してください。

- 「IM-Workflow 管理者操作ガイド」- 「トランザクションデータ、添付ファイルの保存先を変更する」
- 「IM-FormaDesigner 作成者操作ガイド」- 「IM-FormaDesigner の高度な設定を行う」- 「ファイルアップロードのファイルの保存先（transaction-file-location / fileupload）」

移行手順概要

移行手順は以下の3つのステップで構成されています。



ステップ 1

- intra-mart Accel Platform 移行ツールを用いて intra-mart WebPlatform / AppFramework からの移行環境を作成します。
- intra-mart WebPlatform / AppFramework と intra-mart Accel Platform を連携する設定を適宜行います。

ステップ 2

- intra-mart WebPlatform / AppFramework 環境から各IM-VisualDesigner のドキュメントをエクスポートし、intra-mart Accel

Platform 環境上の移行支援ツールにインポートし、IM-FormaDesigner のアプリケーションに変換します。生成されたアプリケーションのひな形を基に開発を行います。

- ステップ2の移行対象のドキュメントに関連した業務は、IM-FormaDesigner のアプリケーションを開発しステップ3を実行するまでは、移行対象のドキュメントで業務を継続します。

ステップ3

- ステップ2で開発した IM-FormaDesigner のアプリケーションに運用を切替えます。
- ※切替時に IM-VisualDesigner のドキュメントは利用されないように設定します。



コラム

ステップ2、3はシステムを停止せずアプリケーション単位で順次移行します。

作業内容概要

1. intra-mart WebPlatform / AppFramework と intra-mart Accel Platform の併用運用環境構築

intra-mart Accel Platform 移行ツールを用いて intra-mart WebPlatform / AppFramework からの移行環境を作成します。
intra-mart WebPlatform / AppFramework と intra-mart Accel Platform を連携する設定を適宜行います。

コラム

intra-mart WebPlatform / AppFramework 上で引き続き扱う情報については、intra-mart Accel Platform へ移行しません。

2. IM-VisualDesigner のドキュメントから IM-FormaDesigner のアプリケーションのひな形を生成

移行支援ツールを利用すると、インポートした IM-VisualDesigner のドキュメントから IM-FormaDesigner のアプリケーションのひな形が作成されます。

- IM-VisualDesigner のフォーム上のラベルアイテムと入力系アイテムを抽出し、IM-FormaDesigner のフォームを生成します。
- 入力した情報を基に IM-FormaDesigner のアプリケーションが作成され、変換されたフォームがひも付けられます。
- IM-VisualDesigner のクエリ設定を利用していた場合、IM-VisualDesigner のクエリ設定からSQL等を抽出し、IM-FormaDesigner のデータソース定義を生成します。

3. IM-FormaDesigner アプリケーションの開発

生成されたひな形から、IM-FormaDesigner アプリケーションを開発し検証を行います。

アプリケーションの開発と検証作業は、IM-VisualDesigner の下記ドキュメント種別により異なります。以下その概要を説明します。

- IM-VisualDesigner 標準ドキュメント
- IM-VisualDesigner ワークフロー連携ドキュメント(IM-Workflow)
- IM-VisualDesigner ワークフロー連携ドキュメント(BPW)

IM-VisualDesigner 標準ドキュメント の開発と検証

開発作業は主にフォームの開発です。

レイアウトの設定、画面アイテム「スクリプト」を用いたアクション設定を行います。

各フォームの開発後に下記の設定を行います。

- 画面遷移の設定
- 権限設定
- 一覧表示設定
- メニュー設定

コラム

権限設定は、IM-VisualDesigner と IM-FormaDesigner で設定できる内容が異なります。

詳細については、「[IM-FormaDesigner 作成者操作ガイド](#)」の「[アプリケーション利用の権限を設定する](#)」を参照してください。

IM-VisualDesigner ワークフロー連携ドキュメント(IM-Workflow) の開発と検証

開発作業は主にフォームの開発とWF連携設定です。

はじめに、レイアウトの設定、画面アイテム「スクリプト」を用いたアクション設定を行います。

各フォームの開発後に、下記 IM-FormaDesigner のWF連携設定を適宜行います。

- 承認画面設定
- 案件プロパティ設定
- 追記設定

IM-VisualDesigner ワークフロー連携ドキュメント(BPW) の開発と検証

開発作業は主にフォームの開発、IM-Workflow の定義情報作成（フロー定義/ルート定義）とWF連携設定です。
はじめに、レイアウトの設定、画面アイテム「スクリプト」を用いたアクション設定を行います。
次に IM-Workflow の定義情報作成（フロー定義/ルート定義）を、既に運用しているBPWのフローをベースに、新規に作成します。
最後に、IM-FormaDesigner からコンテンツ連携情報を作成し、下記WF連携設定を適宜行います。

- 承認画面設定
- 案件プロパティ設定
- 追記設定

4. IM-VisualDesigner → IM-FormaDesigner 切替手順

開発が完了した IM-FormaDesigner アプリケーションを有効化し新規の申請/登録が行えるようにします。
intra-mart WebPlatform / AppFramework 上の IM-VisualDesigner ドキュメントを凍結し、新規の申請/登録が行えないようにします。

コラム

上記の手順作業は、ドキュメント種別に応じて異なります。

intra-mart WebPlatform / AppFramework と intra-mart Accel Platform の併用運用環境構築

1. intra-mart WebPlatform / AppFramework からの移行環境構築

intra-mart Accel Platform 移行ツールを用いて intra-mart WebPlatform / AppFramework からの移行環境を作成します。
移行環境構築の詳細な手順については、「[移行ガイド](#)」を参照してください。

また、注意点として以下のアプリケーションを含めて Juggling プロジェクト を作成してください。

- IM-VisualDesigner Migration Tool
 1. アプリケーションの選択より、IM-VisualDesigner Migration Tool を選択します。
 2. アプリケーションより、IM-VisualDesigner Migration Tool（IM-VisualDesigner Migration Tool（V72併用運用環境用）、ライブラリ）を選択します。

intra-mart WebPlatform / AppFramework 上で利用していた IM-VisualDesigner と IM-Workflow のデータについては移行しないので、「[2. データベースのコピー](#)」でコピーしたデータベースに対してアンインストールを実施します。

- IM-VisualDesigner

intra-mart WebPlatform / AppFramework からコピーしたデータベースから IM-VisualDesigner のテーブルを全て削除します。コピーしたデータベースに対してStorageService上の下記SQLを実行します。

- `%StorageService%/straform/basic/visualDesignerDropTable.sql`

- IM-Workflow

intra-mart WebPlatform / AppFramework からコピーしたデータベースから IM-Workflow のテーブルを全て削除します。
テーブル削除の詳細な手順については、「[IM-Workflow Ver.7.2 セットアップガイド](#)」を参照してください。

IM-Workflow については intra-mart WebPlatform / AppFramework から移行せず新規インストールを行います。「[移行ガイド](#)」 - 「[3. IM-Jugglingによるwarの作成](#)」のステップにおいて、IM-Workflow移行モジュールをwarの構成に含めないでください。

コラム

IM-VisualDesigner と IM-Workflow 以外にも intra-mart Accel Platform 上で不要となるデータ、または利用しないアプリケーションに関しては、intra-mart WebPlatform / AppFramework からの移行用のデータに対してアンインストールを実施します。

また、intra-mart Accel Platform へ移行しないアプリケーションについては、該当する移行モジュールをwarの構成に含めないようにしてください。

2. intra-mart WebPlatform / AppFramework と intra-mart Accel Platform との連携設定

intra-mart WebPlatform / AppFramework と intra-mart Accel Platform を連携する設定を適宜行います。

- intra-mart WebPlatform / AppFramework と intra-mart Accel Platform の間でシングルサインオン連携（IM-HybridSSO）させる場合
「[intra-mart Accel Platform セットアップガイド](#)」-「[iAP-iWP間SSO連携（IM-HybridSSO）](#)」を参照してください。
- intra-mart WebPlatform / AppFramework と intra-mart Accel Platform のマスタ情報を同期させる場合
「[BackwardSync（version 7.2へのマスタ同期）操作ガイド](#)」を参照してください。
- intra-mart WebPlatform / AppFramework と intra-mart Accel Platform のメニューを連携させる場合
「[intra-mart Accel Platform セットアップガイド](#)」-「[外部メニュー連携](#)」を参照してください。

VisualDesigner のドキュメントから FormaDesigner のアプリケーションのひな形を生成

1. IM-VisualDesigner ドキュメントのエクスポート

1. intra-mart WebPlatform / AppFramework にアクセスし、移行対象である IM-VisualDesigner のドキュメント 最新バージョンをzipファイルとしてエクスポートします。
2. ドキュメントをエクスポートするための詳細な手順については、「[IM-VisualDesigner Ver.7.2 サーバモジュール 操作ガイド](#)」-「[5.2.2 エクスポートを行う](#)」を参照してください。



注意

複数のドキュメントに対する一括のエクスポートは利用しないでください。必ず1つずつエクスポートしてください。

2. IM-VisualDesigner ドキュメント→ IM-FormaDesigner アプリケーション 変換

1. エクスポートしたzipファイルをアップロードした上で、アプリケーション情報を入力し、「登録」をクリックします。

VisualDesignerからの変換機能

VisualDesignerからの変換機能	
インポートファイル *	<input type="button" value="ファイルを選択"/> 選択されていません
アプリケーションID *	<input type="text"/>
アプリケーション種別 *	IM-Workflow ▼
有効日付(開始) *	<input type="text"/> <input type="button" value="📅"/>
有効日付(終了)	<input type="text"/> <input type="button" value="📅"/>
アプリケーション名 *	<input type="text"/>
備考	<input type="text"/>
<input type="button" value="実行"/>	

2. 正常に完了していることを確認します。

VisualDesignerからFormaDesignerへの変換結果

VisualDesignerからFormaDesignerへの変換に成功しました。

アプリケーション情報

アプリケーションID	アプリケーション名	処理結果
ringi	稟議書	アプリケーションの登録に成功しました。

ドキュメント情報

処理結果

インポートされたドキュメントの解析に成功しました。

フォーム情報

フォームID	フォーム名	処理結果
ringi	稟議書	フォームのインポートに成功しました。
ringi_start	稟議登録	フォームのインポートに成功しました。
ringi_start	稟議登録	フォームのインポートに成功しました。

データソース情報

読み取りファイル	変換後のデータソース名	処理結果
----------	-------------	------

3. 変換された FormaDesigner のフォーム修正を行います。詳細は次の「[IM-FormaDesigner アプリケーションの開発および検証する手順](#)」へ。

i コラム

ドキュメント設定の使用フォーム一覧 に設定されているフォームのみ IM-FormaDesigner アプリケーション のフォーム に変換されます。

i コラム

上記手順では、IM-VisualDesigner のラベルアイテムや入力系アイテムを IM-FormaDesigner の画面アイテムに変換しただけの状態のため、フォームの位置調整や装飾、スクリプトやアクション処理は、IM-FormaDesigner を使って再度設定する必要があります。

IM-FormaDesigner アプリケーションの開発および検証する手順

開発および検証手順

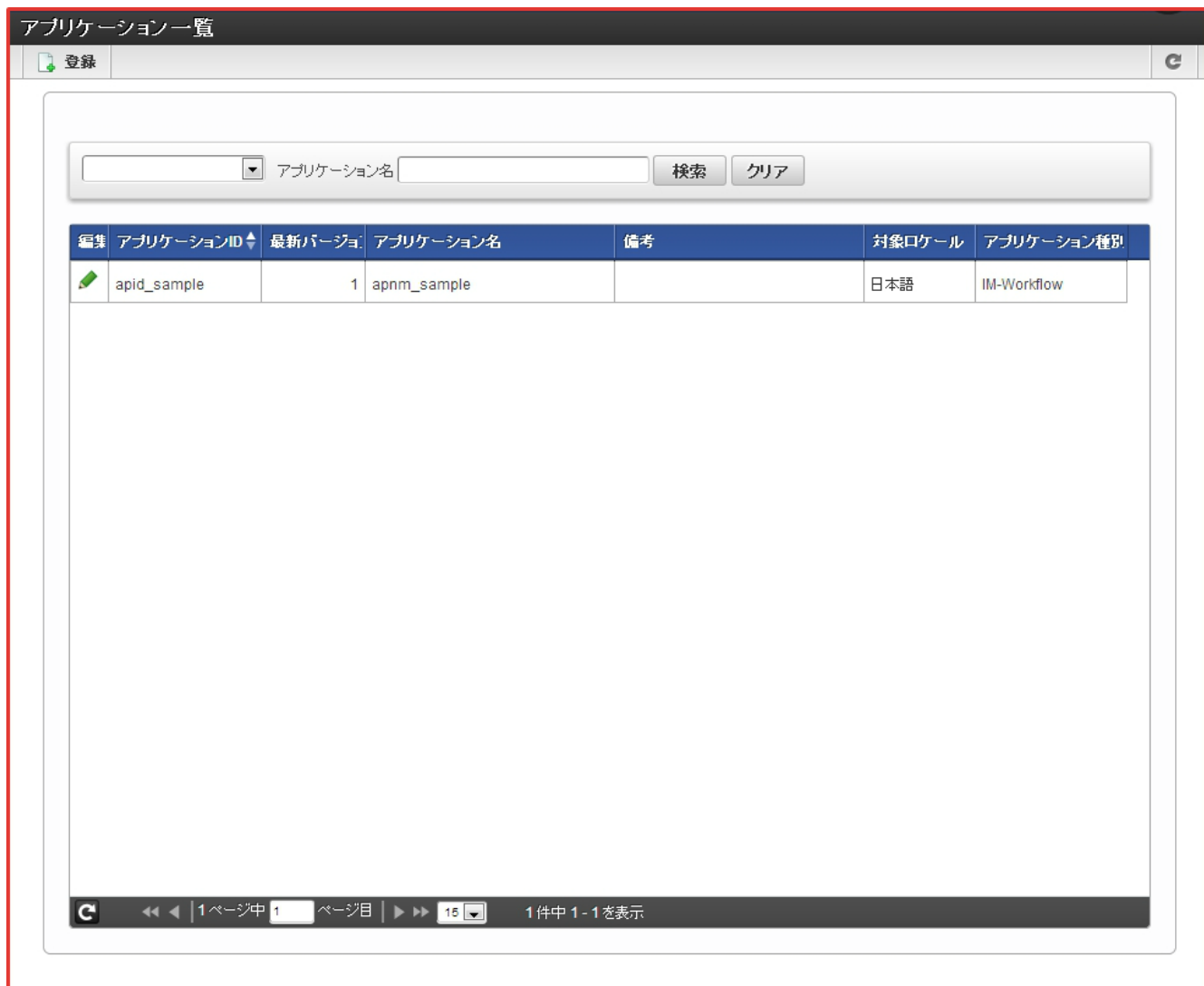
- 1. IM-FormaDesigner フォーム修正
- 2. 各種設定

i コラム

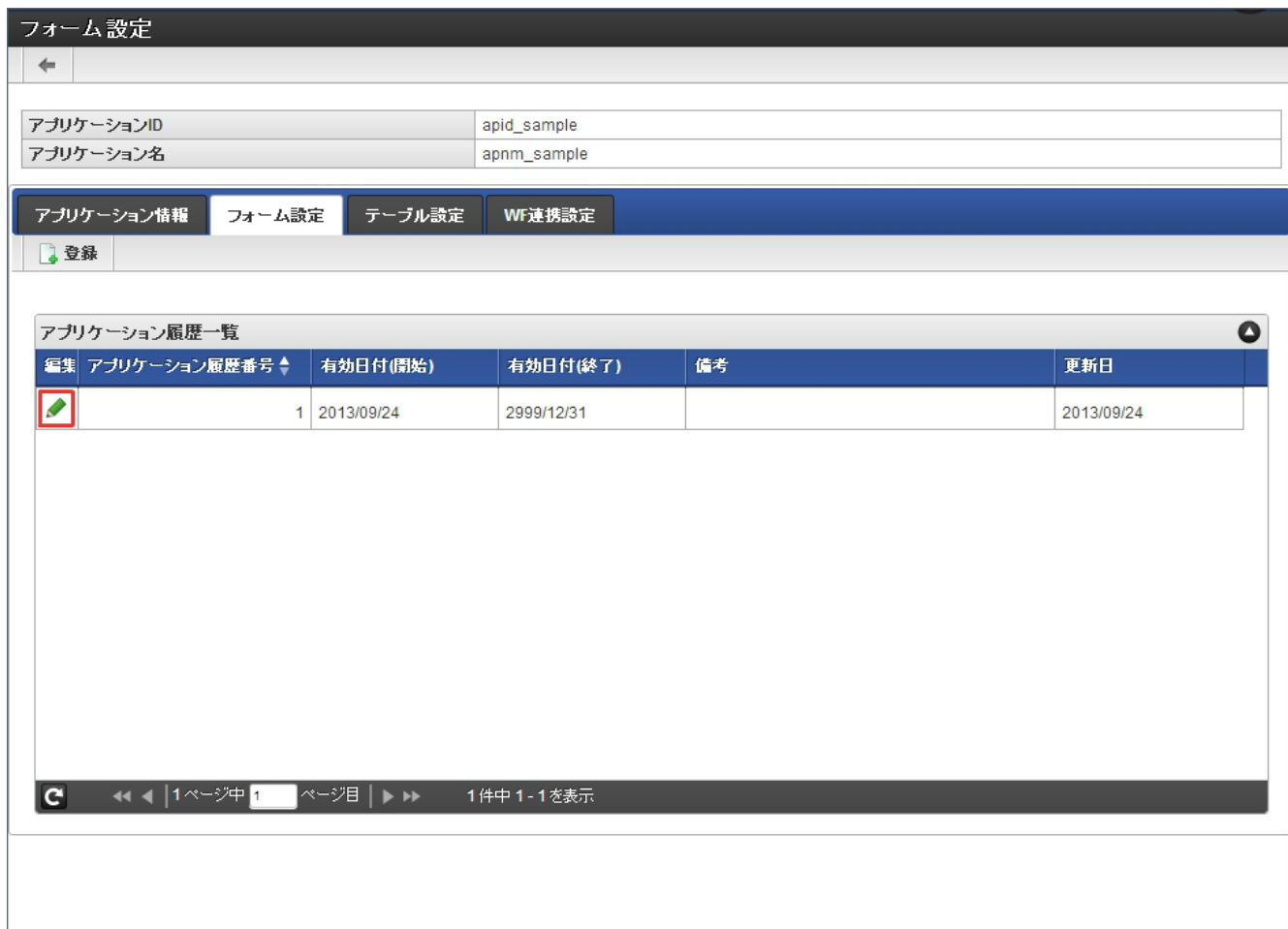
IM-FormaDesigner の操作については、「[IM-FormaDesigner 作成者操作ガイド](#)」を参照してください。

1. IM-FormaDesigner フォーム修正

1. Forma管理者でログインし、「アプリケーション一覧」画面を開きます。



- 修正対象のアプリケーションの「編集」アイコンをクリックします。



3. 「フォーム一覧」タブより、修正対象フォームの「フォーム編集」アイコンをクリックします。



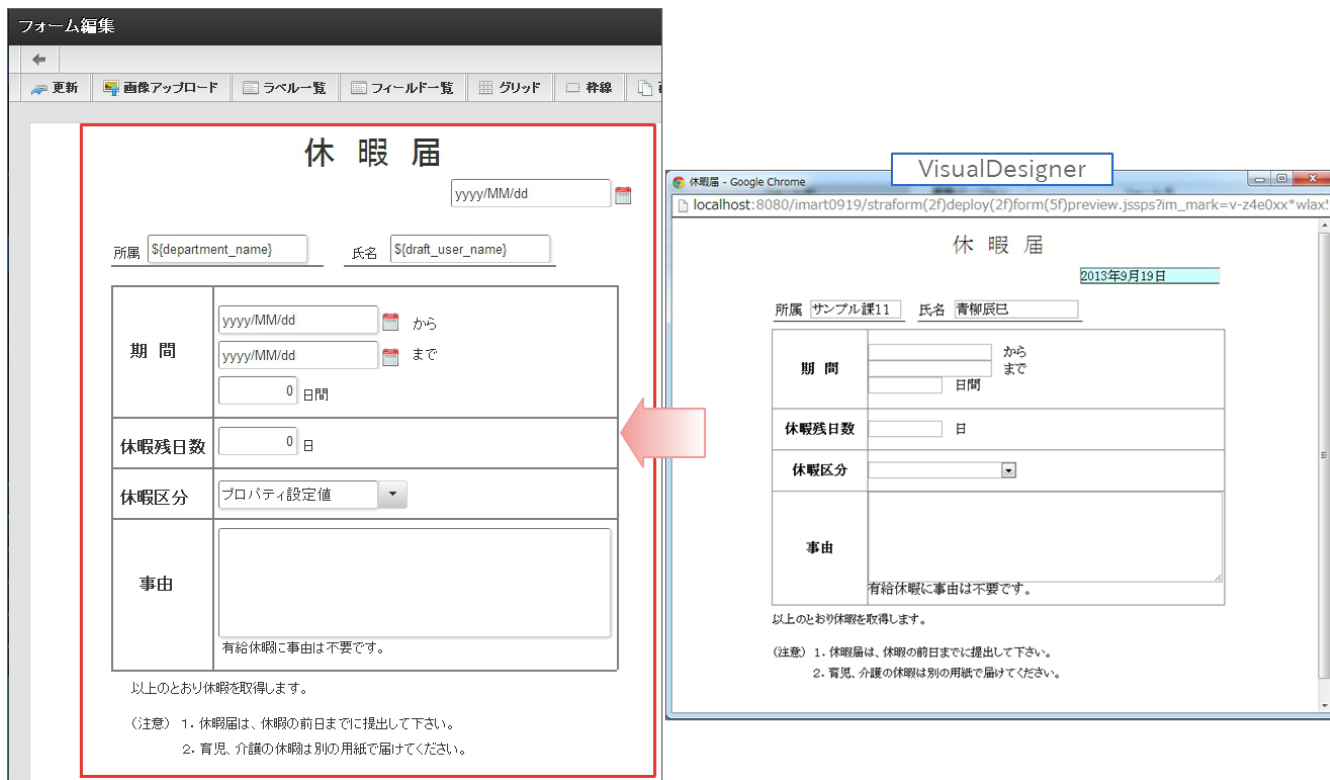
4. IM-VisualDesigner フォームのラベルアイテムと入力系アイテムが IM-FormaDesigner の画面アイテムとして存在することを確認します。

i コラム

ユーザ名 \$(draft_user_name)

IM-FormaDesigner のテキストボックスの左側はラベルアイテムではありません。IM-FormaDesigner の場合、入力系の画面アイテムにラベルと同等のプロパティが付いており、移行時に IM-VisualDesigner の入力系アイテムのアイテム名がそのプロパティに付与されます。

5. IM-VisualDesigner フォームをもとに、IM-FormaDesigner フォームの各アイテムの位置調整や装飾を修正します。以下は、サンプルとして「休暇届」のフォームの修正を行っています。



6. IM-VisualDesigner のフォームで各アイテムに設定していた内容を適宜 IM-FormaDesigner の各アイテムのプロパティに設定します。
 IM-FormaDesigner の各アイテムのプロパティの設定方法については、「[IM-FormaDesigner デザイナヘルプ](#)」を参照してください。

コラム

IM-VisualDesigner で各フォームに対して設定していたクエリーについては、IM-FormaDesigner のデータソース定義に移行されます。データソース定義と各アイテムとの紐付け設定を適宜行ってください。

注意

IM-VisualDesigner のアクション設定に該当する機能は IM-FormaDesigner では提供されていません。IM-VisualDesigner でアクション設定を利用していた場合は、同様のイベント・アクション処理を記述したJavaScriptのソースコードを用意し、画面アイテム「スクリプト」のプロパティに設定してください。

7. IM-FormaDesigner フォーム修正が完了したら、次の各種設定を行います。

2. 各種設定

- データソース定義設定
 - IM-VisualDesigner のスキーマ名表示設定を有効にした状態で作成したクエリーについては、データソース定義の移行の際に移行されたSQLの中にスキーマ情報が含まれますので、適宜SQLを修正してください。
- 承認画面設定
 - アプリケーション種別が「IM-Workflow」の場合、IM-FormaDesigner 側で承認画面の設定を適宜行います。
 1. 「Formaアプリ作成管理」のアプリ一覧画面より、対象アプリケーションの編集アイコンをクリックします。
 2. WF連携設定画面の「WF連携設定」タブを選択し、対象コンテンツの編集アイコンをクリックします。
 3. コンテンツ連携情報画面の「承認画面設定」タブを選択し、承認画面を登録します。

コラム

手順の詳細については「[IM-FormaDesigner 作成者操作ガイド](#)」の「[WF連携設定を行う](#)」-「承認画面設定登録画面の機能と各部の説明」を参照してください。

- 案件プロパティ設定
 - アプリケーション種別が「IM-Workflow」、かつ IM-VisualDesigner で案件プロパティ設定を行っている場合、IM-FormaDesigner 側で再設定が必要です。

1. 「Formaアプリ作成管理」のアプリ一覧画面より、対象アプリケーションの編集アイコンをクリックします。
2. WF連携設定画面の「WF連携設定」タブを選択し、対象コンテンツの編集アイコンをクリックします。
3. コンテンツ連携情報画面の「案件プロパティ設定」タブを選択し、IM-VisualDesigner で設定していた案件プロパティを登録します。

i コラム

手順の詳細については「[IM-FormaDesigner 作成者操作ガイド](#)」の「[WF連携設定を行う](#)」-「[案件プロパティ登録画面の機能と各部の説明](#)」を参照してください。

■ 追記設定

- アプリケーション種別が「IM-Workflow」、かつ IM-VisualDesigner で追記設定を行っている場合、IM-FormaDesigner 側で再設定が必要です。
 1. 「Formaアプリ作成管理」のアプリ一覧画面より、対象アプリケーションの編集アイコンをクリックします。
 2. WF連携設定画面の「WF連携設定」タブを選択し、対象コンテンツの編集アイコンをクリックします。
 3. コンテンツ連携情報画面の「追記設定」タブを選択し、IM-VisualDesigner で設定していた追記設定を登録します。

i コラム

手順の詳細については「[IM-FormaDesigner 作成者操作ガイド](#)」の「[WF連携設定を行う](#)」-「[追記設定登録画面の機能と各部の説明](#)」を参照してください。

IM-VisualDesigner → IM-FormaDesigner 切替手順

既存の IM-VisualDesigner ドキュメントから新規に作成した IM-FormaDesigner アプリケーションへ運用を切り替えます。

IM-VisualDesigner ドキュメントの無効化

intra-mart WebPlatform / AppFramework 上の IM-VisualDesigner ドキュメントに対して新規の申請/登録を行えないように設定します。

i コラム

■ IM-VisualDesigner 標準ドキュメント

ドキュメントに対するメニューのアクセス権限設定を変更し、管理者のみドキュメントの登録画面/一覧画面を閲覧できるようにしてください。

メニュー設定の手順の詳細については、「[IM-VisualDesigner Ver.7.2 サーバモジュール 操作ガイド](#)」-「[第3章ドキュメントの設定](#)」を参照してください。

■ IM-VisualDesigner ワークフロー連携ドキュメント(IM-Workflow)

ドキュメントに紐づく IM-Workflow のフロー定義の設定を変更し、最新バージョンのバージョン期間の終了日を設定します。

IM-Workflow 期間化仕様の詳細については、「[IM-Workflow Ver.7.2 仕様書](#)」-「[2.7.3 案件と申請基準日](#)」を参照してください。

■ IM-VisualDesigner ワークフロー連携ドキュメント(BPW)

ドキュメントに紐づくドキュメントワークフロー(BPW)のプロセス定義の設定を変更し、最新バージョンのバージョン期間の終了日を設定します。

ドキュメントワークフロー(BPW) 期間化仕様の詳細については、「[intra-mart WebPlatform/AppFramework Ver.7.2 ワークフロー 仕様書](#)」-「[1.2.6 起票](#)」を参照してください。

移行支援ツールの内部仕様

IM-VisualDesigner のドキュメントから IM-FormaDesigner のアプリケーションのひな形生成

コラム

入力された情報を基に IM-FormaDesigner のアプリケーションを生成します。IM-VisualDesigner のドキュメント情報からフォーム・クエリーに関する情報を抽出し、IM-FormaDesigner 用にデータの変換を行い、生成したアプリケーションに設定します。

- IM-FormaDesigner に移行される情報

IM-VisualDesigner	=> IM-FormaDesigner
フォーム	=> フォーム（注1）
DB参照機能（クエリー）	=> データソース定義（注3）

（注1） フォームの変換については、「[フォーム変換ルール](#)」を参照してください。

（注2） データソース定義については、DB参照機能（クエリー）設定のSQLと抽出条件が移行されます。

フォーム変換ルール

ステップ2の IM-VisualDesigner → IM-FormaDesigner 変換により、フォームやフォーム内の各アイテムがどのように変換されるかの一覧です。

- フォーム名
IM-VisualDesigner のフォーム名が IM-FormaDesigner のフォーム名に変換されます。
- 画面遷移
IM-VisualDesigner 上で設定されていた画面遷移は移行されません。
ただし、IM-VisualDesigner の「登録画面」については、IM-FormaDesigner の「メインフォーム」として設定されます。
- レイアウト
レイアウトは移行されません。
移行後は縦一列にアイテムが整列します。
- ラベル
画面アイテム「ラベル」に変換されます。

- 画面アイテム「ラベル」

移行されるプロパティ

名前	=> アイテム名
文字列	=> ラベル

- テキストフィールド
data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「文字列」
データ型「数値」	=> 画面アイテム「数値」
データ型「日付」	=> 画面アイテム「日付」

- 画面アイテム「文字列」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、ラベル

移行されるプロパティ

文字列	=> フィールド初期値
-----	-------------

- 画面アイテム「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、（前）ラベル
文字列	=> フィールド初期値

- 画面アイテム「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、ラベル

- テキストエリア

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「複数行文字列」
データ型「数値」	=> 画面アイテム「数値」
データ型「日付」	=> 画面アイテム「日付」

- 画面アイテム「複数行文字列」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、ラベル
文字列	=> フィールド初期値

- 画面アイテム「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、（前）ラベル
文字列	=> フィールド初期値

- 画面アイテム「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、ラベル

- ボタン

attach_file の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

attach_file = true	=> 画面アイテム「ファイルアップロード」
attach_file = false	=> 移行されない

- 画面アイテム「ファイルアップロード」

移行されるプロパティ

名前	=> アイテム識別ID
文字列（登録・申請時）	=> ラベル

- チェックボックス

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」 => 画面アイテム「チェックボックス」

データ型「数値」 => 画面アイテム「数値」

データ型「日付」 => 画面アイテム「日付」

■ 画面アイテム「チェックボックス」

移行されるプロパティ

名前 => フィールド識別ID

表示名 => アイテム名、フィールド識別名、ラベル

■ 画面アイテム「数値」

移行されるプロパティ

名前 => フィールド識別ID

表示名 => アイテム名、フィールド識別名、（前）ラベル

文字列 => フィールド初期値

■ 画面アイテム「日付」

移行されるプロパティ

名前 => フィールド識別ID

表示名 => アイテム名、フィールド識別名、ラベル

■ ラジオボタン

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」 => 画面アイテム「ラジオボタン」

データ型「数値」 => 画面アイテム「数値」

データ型「日付」 => 画面アイテム「日付」

■ 画面アイテム「ラジオボタン」

移行されるプロパティ

名前 => フィールド識別ID

表示名 => アイテム名、フィールド識別名、ラベル

項目（項目） => 項目定義（表示値）

項目（送信項目） => 項目定義（送信値）

※ アイテムにクエリ設定を行っている場合、その設定情報は移行されません（クエリは移行されます）。
クエリ設定を行う場合は、IM-FormaDesigner 側で「外部連携」の「データソース設定値」の設定を行ってください。

■ 画面アイテム「数値」

移行されるプロパティ

名前 => フィールド識別ID

表示名 => アイテム名、フィールド識別名、（前）ラベル

文字列 => フィールド初期値

■ 画面アイテム「日付」

移行されるプロパティ

名前 => フィールド識別ID

表示名 => アイテム名、フィールド識別名、ラベル

■ コンボボックス

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「セレクトボックス」
データ型「数値」	=> 画面アイテム「数値」
データ型「日付」	=> 画面アイテム「日付」

■ 画面アイテム「セレクトボックス」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、ラベル
項目（項目）	=> 項目定義（表示値）
項目（送信項目）	=> 項目定義（送信値）

※ アイテムにクエリ設定を行っている場合、その設定情報は移行されません（クエリは移行されます）。
クエリ設定を行う場合は、IM-FormaDesigner 側で「外部連携」の「データソース設定値」の設定を行ってください。

■ 画面アイテム「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、（前）ラベル
文字列	=> フィールド初期値

■ 画面アイテム「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、ラベル

■ リスト

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「リストボックス」
データ型「数値」	=> 画面アイテム「数値」
データ型「日付」	=> 画面アイテム「日付」

■ 画面アイテム「リストボックス」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、ラベル
項目（項目）	=> 項目定義（表示値）
項目（送信項目）	=> 項目定義（送信値）

※ アイテムにクエリ設定を行っている場合、その設定情報は移行されません（クエリは移行されます）。
クエリ設定を行う場合は、IM-FormaDesigner 側で「外部連携」の「データソース設定値」の設定を行ってください。

■ 画面アイテム「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、（前）ラベル
文字列	=> フィールド初期値

■ 画面アイテム「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、ラベル

- レイヤ
移行されません。
- テーブルアイテム
「テーブル行を追加」アクションの有無によって、変換される画面アイテム、およびテーブル内の子アイテムの変換アイテムが異なります。

変換される画面アイテム

「テーブル行を追加」有り	=> 画面アイテム「明細テーブル」
「テーブル行を追加」無し	=> 移行されない

- 「テーブル行を追加」有りのテーブルに紐付く子アイテムの場合
 - ラベル
画面アイテム「ラベル」に変換されます。
※ テーブルとしては変換されず、テーブル外部に通常の画面アイテムとして変換されます。

画面アイテム「ラベル」

移行されるプロパティ

名前	=> アイテム名
文字列	=> ラベル

- テキストフィールド
data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「明細テーブル」の列定義「文字列」
データ型「数値」	=> 画面アイテム「明細テーブル」の列定義「数値」
データ型「日付」	=> 画面アイテム「明細テーブル」の列定義「日付」

- 画面アイテム「明細テーブル」の列定義「文字列」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
文字列	=> フィールド初期値

- 画面アイテム「明細テーブル」の列定義「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
文字列	=> フィールド初期値

- 画面アイテム「明細テーブル」の列定義「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名

- テキストエリア
data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「明細テーブル」の列定義「文字列」
-----------	----------------------------

変換される画面アイテム

データ型「数値」	=> 画面アイテム「明細テーブル」の列定義「数値」
データ型「日付」	=> 画面アイテム「明細テーブル」の列定義「日付」

- 画面アイテム「明細テーブル」の列定義「文字列」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
文字列	=> フィールド初期値

- 画面アイテム「明細テーブル」の列定義「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
文字列	=> フィールド初期値

- 画面アイテム「明細テーブル」の列定義「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名

- ボタン

attach_file の値によって、変換される画面アイテムが異なります。

※ テーブルとしては変換されず、テーブル外部に通常の画面アイテムとして変換されます。

変換される画面アイテム

attach_file = true	=> 画面アイテム「ファイルアップロード」
attach_file = false	=> 移行されない

- 画面アイテム「ファイルアップロード」

移行されるプロパティ

名前	=> アイテム識別ID
文字列（登録・申請時）	=> ラベル

- チェックボックス

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「明細テーブル」の列定義「文字列」
データ型「数値」	=> 画面アイテム「明細テーブル」の列定義「数値」
データ型「日付」	=> 画面アイテム「明細テーブル」の列定義「日付」

- 画面アイテム「明細テーブル」の列定義「文字列」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名

- 画面アイテム「明細テーブル」の列定義「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
文字列	=> フィールド初期値

- 画面アイテム「明細テーブル」の列定義「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名

- ラジオボタン

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「明細テーブル」の列定義「ラジオボタン」
データ型「数値」	=> 画面アイテム「明細テーブル」の列定義「数値」
データ型「日付」	=> 画面アイテム「明細テーブル」の列定義「日付」

- 画面アイテム「明細テーブル」の列定義「ラジオボタン」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
項目 (項目)	=> 項目定義 (表示値)
項目 (送信項目)	=> 項目定義 (送信値)

※ アイテムにクエリ設定を行っている場合、その設定情報は移行されません (クエリは移行されます)。クエリ設定を行う場合は、IM-FormaDesigner 側で「外部連携」の「データソース設定値」の設定を行ってください。

- 画面アイテム「明細テーブル」の列定義「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
文字列	=> フィールド初期値

- 画面アイテム「明細テーブル」の列定義「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名

- コンボボックス

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「明細テーブル」の列定義「セレクトボックス」
データ型「数値」	=> 画面アイテム「明細テーブル」の列定義「数値」
データ型「日付」	=> 画面アイテム「明細テーブル」の列定義「日付」

- 画面アイテム「明細テーブル」の列定義「セレクトボックス」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
項目 (項目)	=> 項目定義 (表示値)
項目 (送信項目)	=> 項目定義 (送信値)

※ アイテムにクエリ設定を行っている場合、その設定情報は移行されません (クエリは移行されます)。クエリ設定を行う場合は、IM-FormaDesigner 側で「外部連携」の「データソース設定値」の設定を行ってください。

- 画面アイテム「明細テーブル」の列定義「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
文字列	=> フィールド初期値

- 画面アイテム「明細テーブル」の列定義「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名

- リスト

data_type の値によって、変換される画面アイテムが異なります。

変換される画面アイテム

データ型「文字列」	=> 画面アイテム「明細テーブル」の列定義「セレクトボックス」
データ型「数値」	=> 画面アイテム「明細テーブル」の列定義「数値」
データ型「日付」	=> 画面アイテム「明細テーブル」の列定義「日付」

- 画面アイテム「明細テーブル」の列定義「セレクトボックス」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
項目（項目）	=> 項目定義（表示値）
項目（送信項目）	=> 項目定義（送信値）

※ アイテムにクエリ設定を行っている場合、その設定情報は移行されません（クエリは移行されます）。クエリ設定を行う場合は、IM-FormaDesigner 側で「外部連携」の「データソース設定値」の設定を行ってください。

- 画面アイテム「明細テーブル」の列定義「数値」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名
文字列	=> フィールド初期値

- 画面アイテム「明細テーブル」の列定義「日付」

移行されるプロパティ

名前	=> フィールド識別ID
表示名	=> アイテム名、フィールド識別名、列名

- レイヤ

移行されません。

- 直線

移行されません。

- 矩形

移行されません。

- 「テーブル行を追加」無しのテーブルに紐付く子アイテムの場合

テーブルに紐付かない通常のアイテムとして変換されます。変換ルールは各アイテムの変換ルールを参照してください。

- 直線

移行されません。

- 矩形

移行されません。

- イメージ
移行されません。

i コラム

IM-VisualDesigner 側のボックス系アイテム（チェックボックス、ラジオボタン、コンボボックス、リスト）のプロパティ「送信項目」に2バイト文字を利用していた場合は、IM-FormaDesigner では扱えません。

IM-VisualDesigner とIM-FormaDesignerの機能比較

IM-VisualDesigner と IM-FormaDesignerの機能比較一覧です。

IM-VisualDesigner Ver.7.2	IM-FormaDesigner for Accel Platform
<p>【データソース定義機能】 - 検索クエリの作成方法</p> <p>テーブルから任意のカラムを選択して検索クエリを作成します。</p>	<p>ユーザがSQLを記述し、検索クエリを作成します。 SQLの記述についてはViewCreatorを用いることで容易化できます。</p>
<p>【テーブルアイテム】 - テーブル内に配置できるアイテムの種類</p> <p>すべてのフィールドアイテムに対応しています。</p>	<p>一部のフィールドアイテムのみ対応しており、以下のフィールドアイテムはテーブル内に配置できません。</p> <ul style="list-style-type: none"> ・ 複数行文字列 ・ リストボックス ・ チェックボックス
<p>【ボックス系アイテム】 - 対応するデータ型</p> <p>文字列、数値、日付に対応しています。</p>	<p>文字列のみ対応しています（2バイト文字を含む文字列には対応していません）。</p>
<p>【関数機能】 - 利用方法</p> <p>指定のコードにより、テキストフィールドの初期値として関数を利用します。</p>	<p>画面アイテム「関数」を利用します。</p>
<p>【イベント・アクション設定機能】 - 設定方法</p> <p>イベントの種類とアクションの種類を設定します。</p>	<p>ユーザが画面アイテム「スクリプト」を用いてクライアントJavaScriptのコードを記述します。</p>
<p>【イベント・アクション設定機能】 - リンク機能</p> <p>別画面をポップアップで表示できます。 呼び出し先の画面は下記4つの中から選択可能です。</p> <ul style="list-style-type: none"> ・ スクリプト開発モデル ・ JavaEE開発モデル ・ IM-VisualDesigner で作成したドキュメント ・ 外部URL 	<p>別画面をポップアップで表示できます。 呼び出し先の画面は「IM-FormaDesigner で作成した画面」のみが利用可能です。</p>

