



目次

- 1. 改訂情報
- 2. はじめに
 - 2.1. 本書の目的
 - 2.2. 前提条件
 - 2.3. 対象読者
 - 2.4. 本書の構成
- 3. IM-BPM Designerのインストール
 - 3.1. インストールと実行
- 4. 基本機能
 - 4.1. プロジェクト、およびプロセス定義ファイルの作成
 - 4.2. チュートリアル
 - 4.3. エディタ上で配置可能なフローエレメント
 - 4.4. 共通の項目
 - 4.5. プロセス内の変数
 - 4.6. 設計書出力機能
 - 4.7. デプロイ資材の作成

改訂情報

変更年月日	変更内容
2016-08-01	初版
2016-12-01	第2版 ■ Skip expressionの実装に伴い、下記の項に説明を追加しました。 ■ コネクション ■ タスク ■ イベント のTerminateEndEventに対して、プロセスの強制終了の範囲の説明を追加しました。 ■ 「基本機能」のページに日付入力形式についての説明を追加しました。 ■ 「intra-mart」のページ、申請基準日の説明に日付入力形式を加筆しました。
2017-04-01	第3版 ■ 「IM-BPM Designerのインストール」のIM-BPM Designerのダウンロード方法の説明を修正しました。 ■ 「タスク」のUserTaskに対して、Categoryの説明を追加しました。 ■ 「タスク」のScriptTaskに対して、Script fileの説明を追加しました。 ■ 「intra-mart」に対して、各タスクのメインコンフィグの説明を更新しました。 ■ 「イベント」に対して、SignalStartEventの説明を追加しました。 ■ 「プロジェクト、およびプロセス定義ファイルの作成」の「プロセス定義の設定」の項に「ユーザに業務キーの設定を許可しない」の設定に関する説明を追加しました。 ■ 「共通の項目」に対して、「Relation document (関連ドキュメント)」を追加しました。

はじめに

項目

- 本書の目的
- 前提条件
- 対象読者
- 本書の構成

本書の目的

本書は、IM-BPM Designerを利用してBPMN（Business Process Model and Notation）2.0「国際標準（ISO19510）」のビジネスプロセスの作成を始める開発者の皆様の支援を目的とした操作ガイドです。

IM-BPM Designerのインストールから、プロセス定義の作成、intra-mart Accel Platformのデプロイまでの基本的な操作を習得することを目的としています。

intra-mart Accel Platformにデプロイしたプロセス定義をシステム画面上で利用する方法や操作する方法については「[IM-BPM ユーザ操作ガイド](#)」を参照してください

前提条件

- IM-BPM Designerを使用するクライアント環境の要件はeclipse 4.2 (Juno)のリリースノートに準拠します。
OSはWindows 64bit版のみの環境をサポートします。
 - eclipse 4.2 (Juno) rerease notes : http://www.eclipse.org/eclipse/development/readme_eclipse_4.2.html

対象読者

本書では以下の目的で IM-BPM Designerを利用する開発者を対象としています。

- IM-BPM Designerを利用してBPMNによる業務を可視化し、BPMのプロセス定義を作成したい開発者
- IM-BPM for Accel Platformで業務を管理するためのプロセス定義を作成したい開発者

なお、本書では次の内容を理解していることが必須となります。

- eclipseの利用に関する基本的な知識
- BPMN2.0を利用したプロセス定義の基本的な知識

また必要に応じて下記の内容を理解していることで、より幅広く機能を利用することができます。

- Java、またはJavaScriptを利用したプログラム開発の基本的な知識
- intra-mart Accel Platformを利用した開発の基礎知識
- 他のアプリケーションを利用したプロセス定義を作成する場合、対象となるアプリケーションの基礎知識
 - IM-LogicDesigner
 - IM-Workflow
 - IM-FormaDesigner for Accel Platform
 - IM-BIS for Accel Platform

本書の構成

- [IM-BPM Designerのインストール](#)
IM-BPM Designerのインストール方法について解説します。
- [基本機能](#)
IM-BPM Designerを利用したプロセス定義ファイルの作成、フローの設計、システムへのデプロイの方法について解説を行います。

IM-BPM Designerのインストール方法について説明します。

IM-BPM Designerをセットアップについては「[プロジェクト、およびプロセス定義ファイルの作成](#)」を参照してください。

項目

- [インストールと実行](#)

インストールと実行

1. IM-BPM Designerは「[プロダクトファイルダウンロード](#)」より、zipファイルで入手できます。
IM-BPM Designerのzipファイルをダウンロードし、任意のフォルダにて解凍してください。
ダウンロードには intra-mart Accel Platform、または、IM-BPM for Accel Platform のライセンスが必要です。
2. %IM-BPM Designerのzipファイルを解凍したフォルダ %eclipse/eclipse.exeを実行し起動します。

コラム

IM-BPM Designerはeclipse 4.2 (Juno) for Win_64bit をベースにしております。
そのため、IM-BPM DesignerのクライアントOSの要件に関しては、eclipse 4.2 (Juno)とWindows 64bit版の要件と同等です。
クライアント要件の詳細については、「[eclipse 4.2 \(Juno\) リリースノート](#)」を参照してください。

注意

2016 Summer現在、IM-BPM Designerは Windows 64bit のみ対応です。

基本機能

IM-BPM Designerで利用できる機能や設定、BPMのプロセス定義を作成する流れについて説明します。

初めてIM-BPM Designerでのプロセス定義を作成される読者は、チュートリアルを実施してプロセス定義作成の流れを把握してください。

プロジェクト、およびプロセス定義ファイルの作成

本項ではIM-BPM Designerを利用してBPMのプロセス定義を作成するためのプロジェクトとファイルを作成する方法について説明します。

専用のパースペクティブの切り替え、プロジェクトの作成、プロセス定義ファイルの作成について説明します。

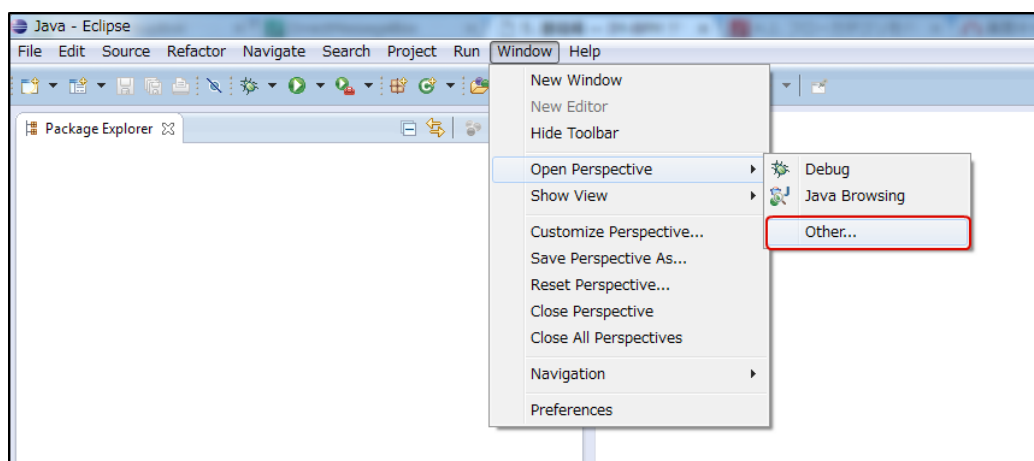
BPM パースペクティブ

プロセス定義の作成、編集をするための専用のパースペクティブが用意されています。

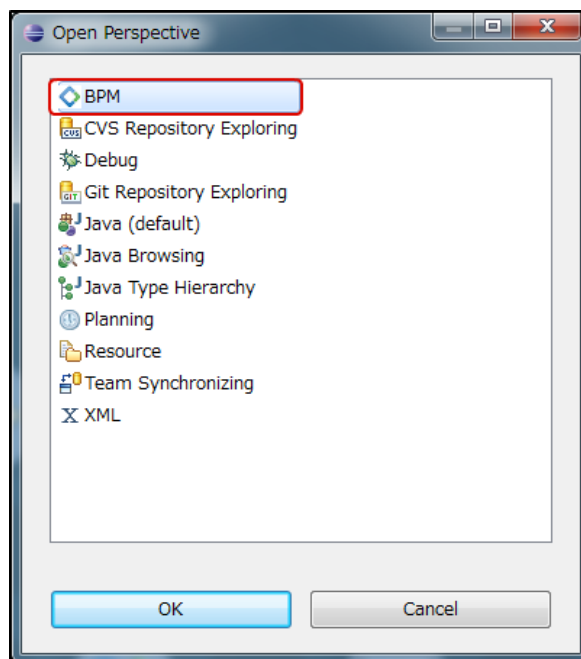
ここで記述されている方法でなくても、BPMプロジェクトを新規作成した際、パースペクティブを切り替えるかどうかの確認ダイアログが表示されますので、そこで切り替えることもできます。

パースペクティブの切り替え

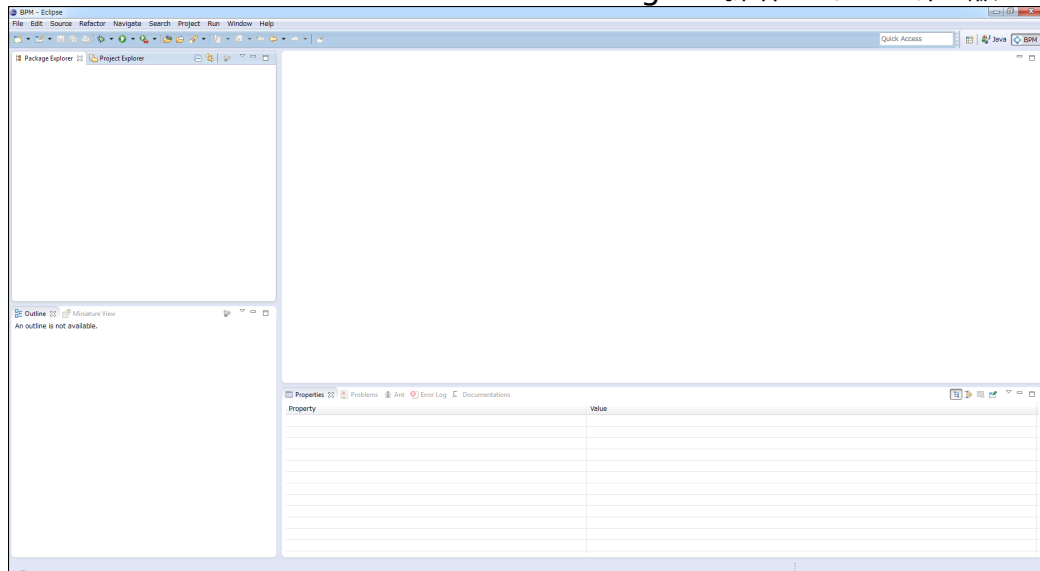
1. EclipseのツールバーにあるWindow>Open Perspective>Otherをクリックします



2. BPMを選択し、OKをクリックします。



3. これでパースペクティブがBPMに切り替わりました。



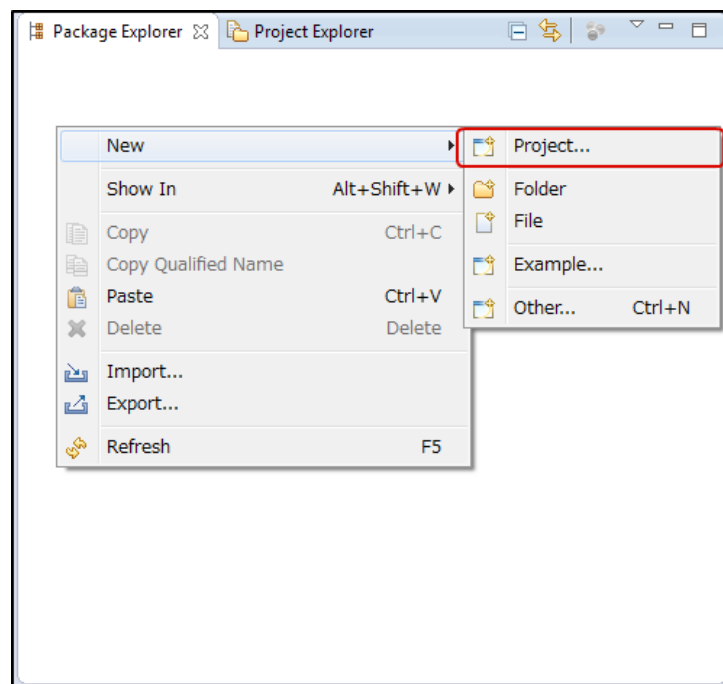
BPMプロジェクト

BPMプロジェクトはプロセス定義、およびプロセスフローの実行時に利用されるJavaクラスを管理するためのプロジェクトです。

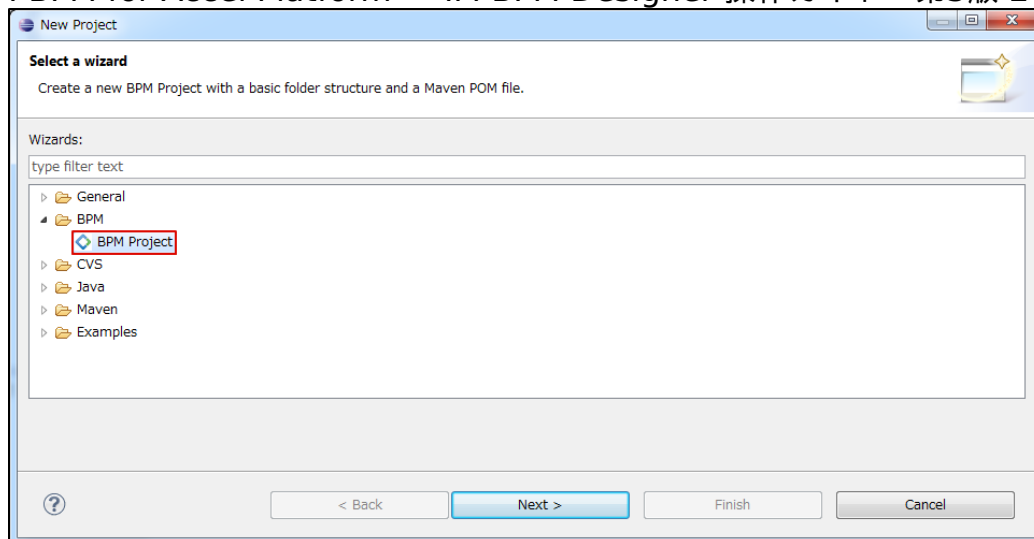
BPMプロジェクトの作成

BPMプロジェクトの作成方法について説明します。

1. Package Explorerで右クリックし、New>Projectをクリックします。



2. BPM>BPM Projectを選択してNextをクリックします。

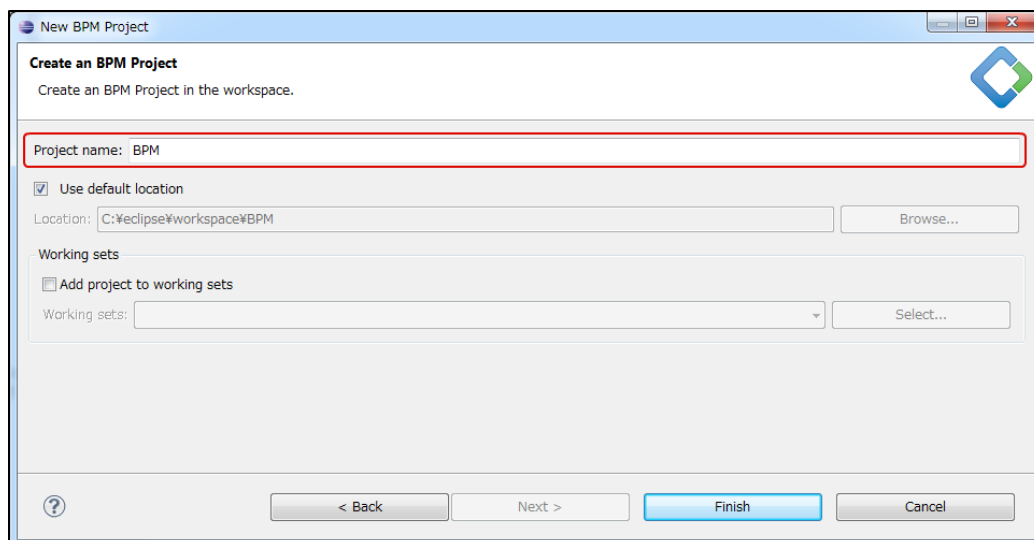


3. プロジェクトの名前を入力します。

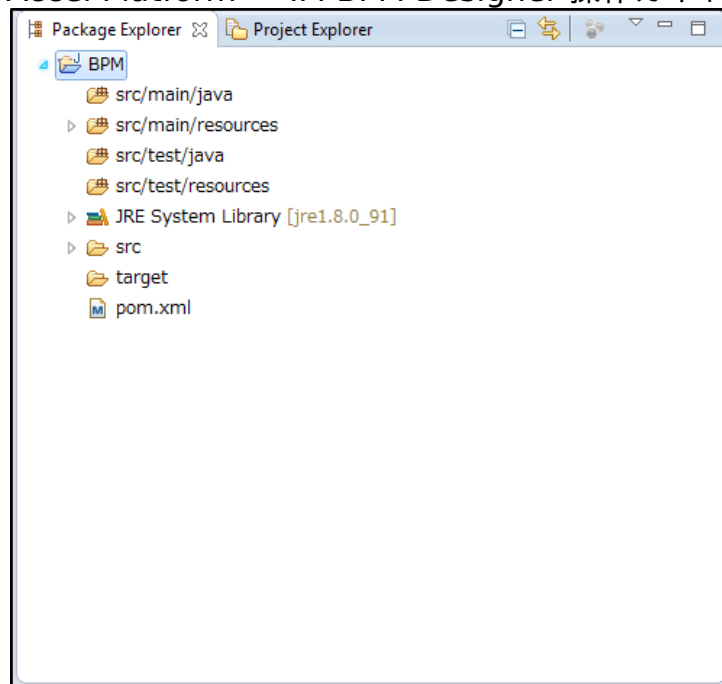
Project nameに任意のプロジェクト名を入力してください。

本書では「BPM」としています。

入力を終わったらFinishをクリックします。



4. 以上でBPMプロジェクトの作成が完了しました。



BPMプロジェクトの構成

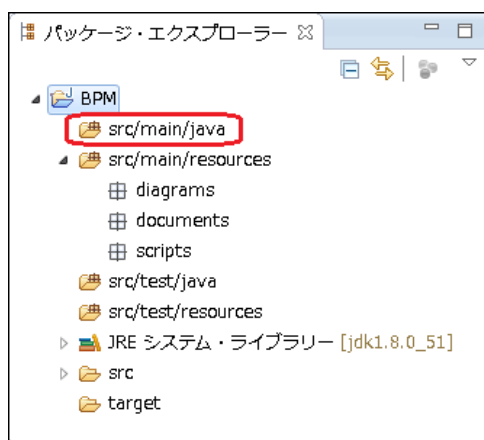
BPMプロジェクトでは、主に以下の2つのソース・フォルダーでファイルを作成・編集していきます。

- src/main/java

javaファイルを作成するためのソース・フォルダーです。

BPMプロジェクトはeclipseのJavaプロジェクトと同様の機能が利用できます。

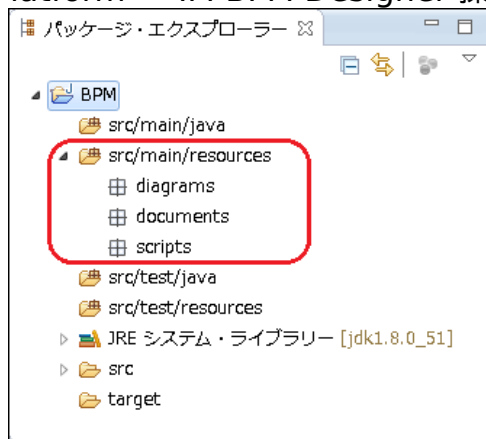
必要に応じてライブラリの追加などを行い、開発してください。



- src/main/resources

このソース・フォルダーは、3つのパッケージを管理しています。

- diagrams - 後述のプロセス定義ファイルを管理するパッケージです
- scripts - ScriptTaskに設定するJavaScriptファイル（.jsファイル）を配置するパッケージです。
- documents - BPMのプロセスに関連するドキュメントを配置するパッケージです。



プロセス定義ファイル

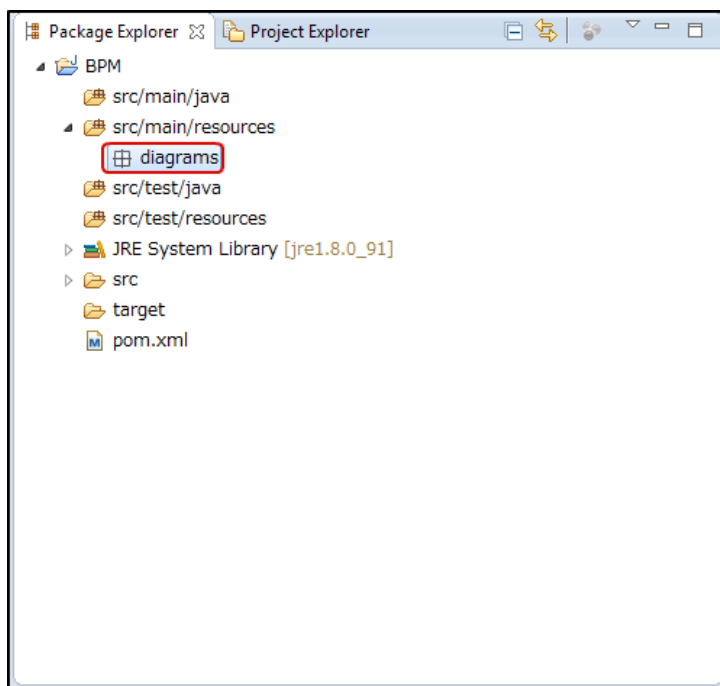
プロセス定義ファイルはBPMのプロセス定義を設計するファイルです。

拡張子は「.bpmn」となり、BPM Diagram EditorというGUIツールでプロセス定義を設計します。

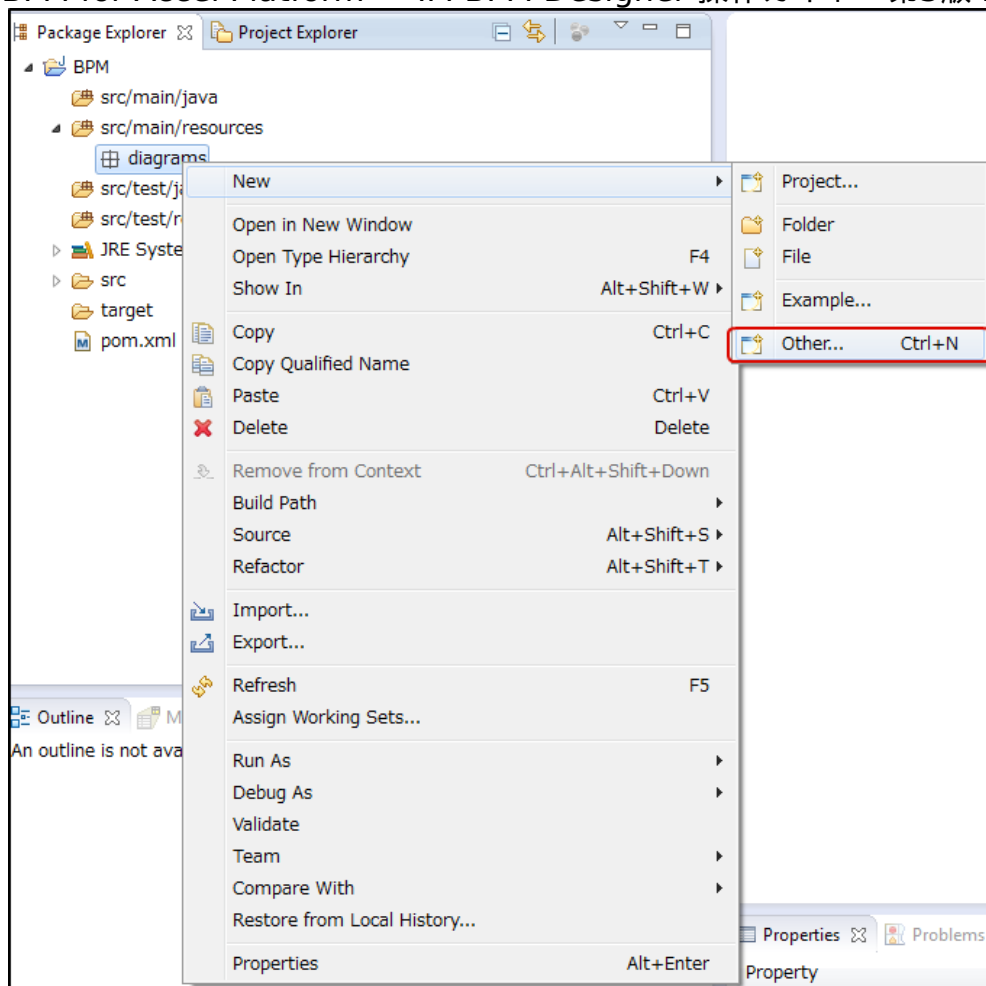
プロセス定義ファイルの作成

プロセス定義ファイルを作成する方法について説明します。

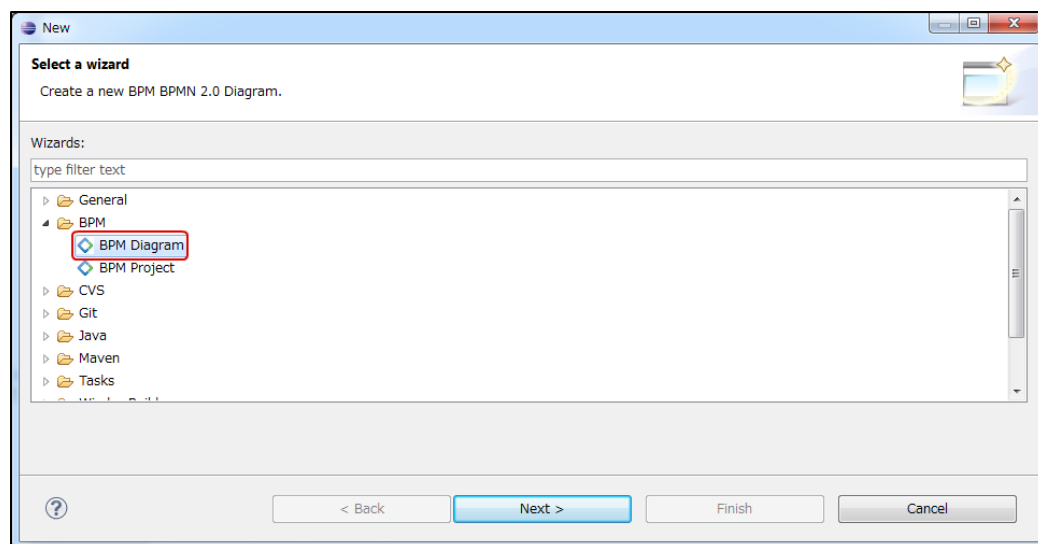
1. Package Explorerでプロセス定義ファイルを作成したいプロジェクト、またはその中の階層にあるファイルを選択し、右クリックします。
下の例では作成したいプロジェクトのsrc/main/resourcesのソース・フォルダーを選択しています。



2. New>Otherをクリックします。



3. BPM>BPM Diagramを選択してNextをクリックします。

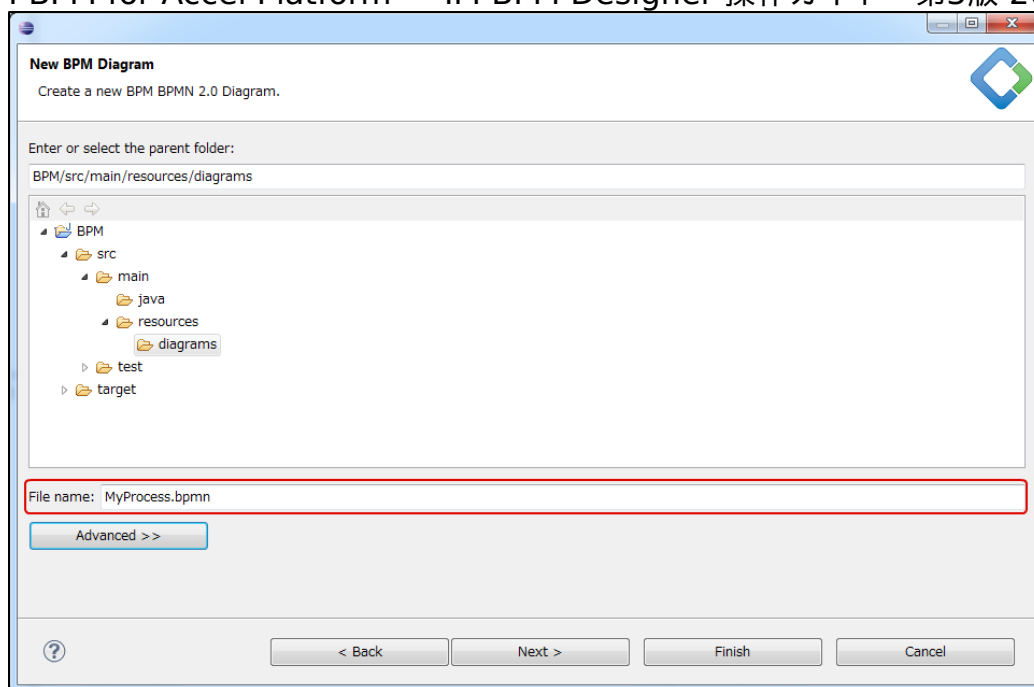


4. ファイルの名前を入力します。

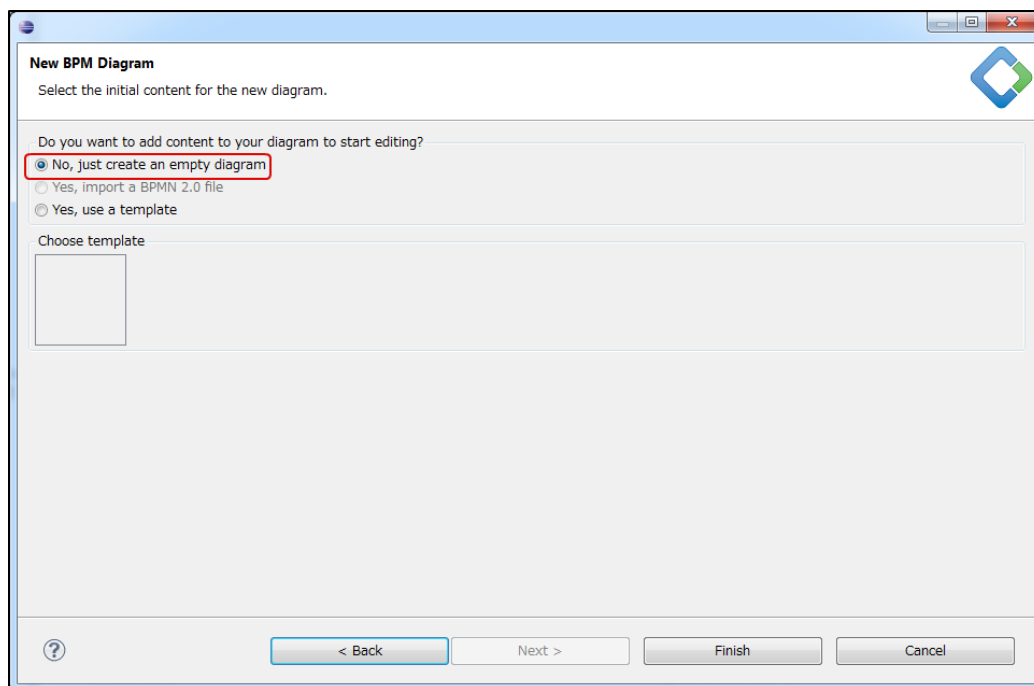
File nameに任意のファイル名を入力してください。

本書では「MyProcess」としています。

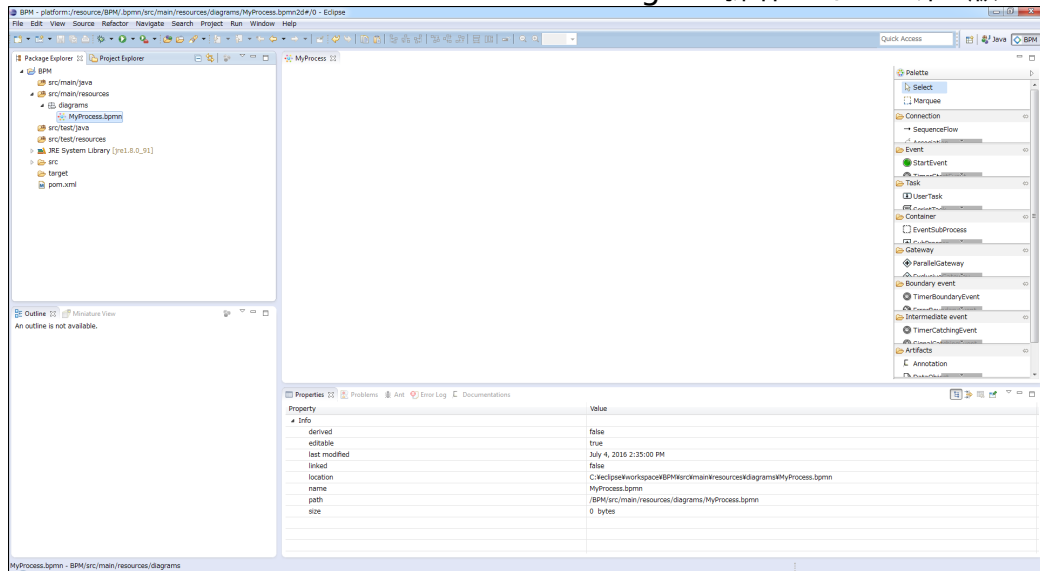
入力を終えたらNextをクリックします。



5. No, just create an empty diagramを選択して、Finishをクリックします。



6. 以上でプロセス定義ファイルの準備ができました。

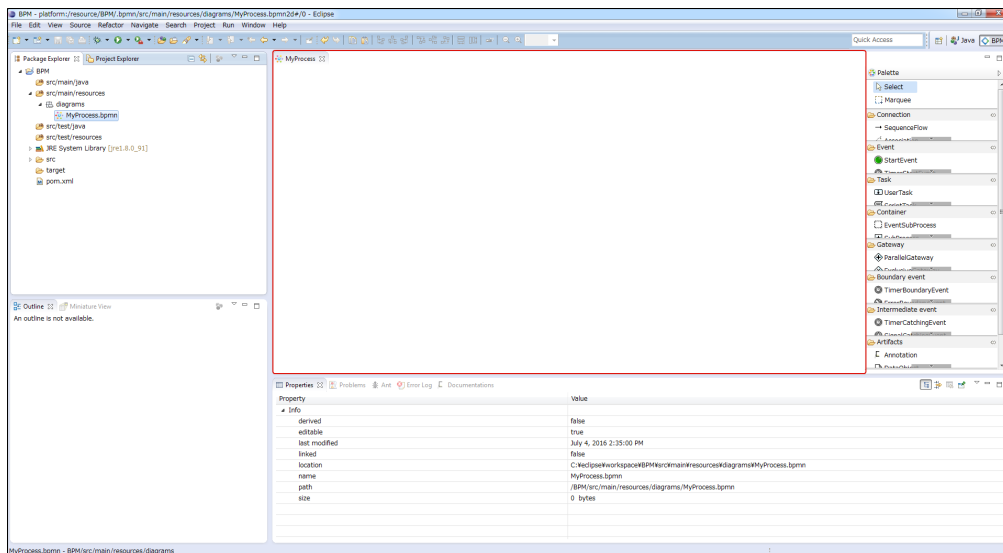


BPM Diagram Editor

BPM Diagram Editorは以下の3つのViewで構成されています。

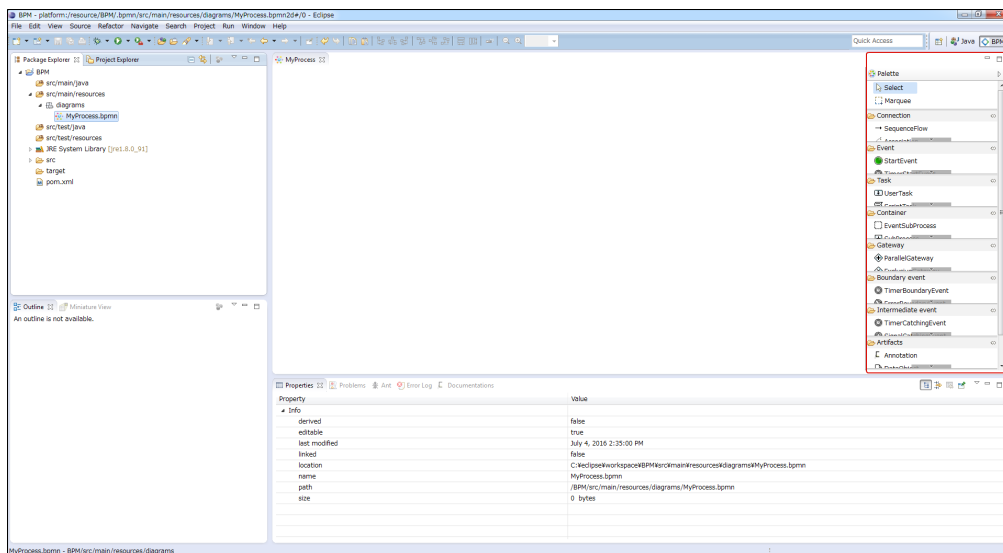
1. Diagram Editor

- 画面中央部に表示されたBPM Diagram Editorは、フローエレメントを配置してプロセス定義を作成します。



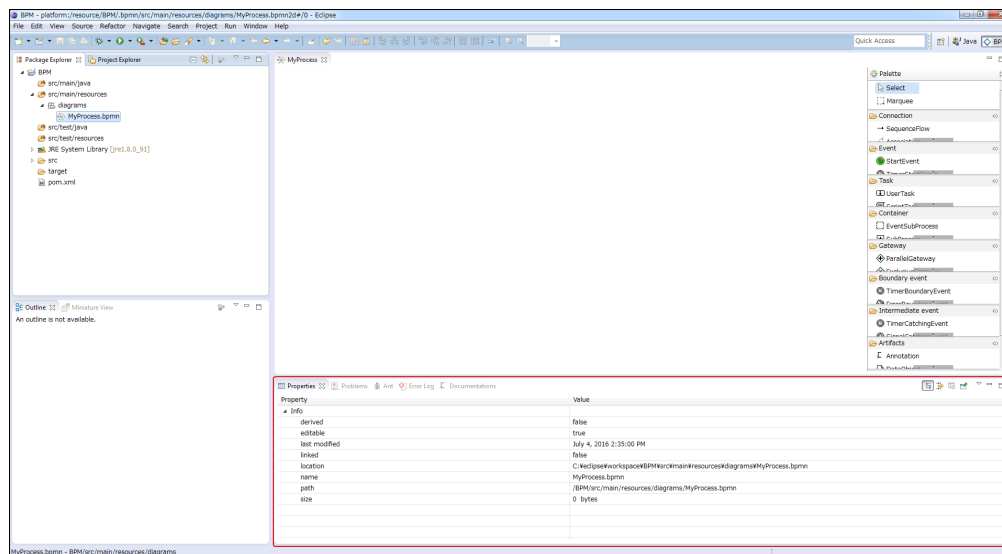
2. Palette

- 画面右側に表示されたPaletteは、BPM Diagram Editorに配置できるフローエレメントの一覧です。



3. Properties ・ ビュー

- 画面下部に表示されたPropertiesは、プロセス定義や各オブジェクトの情報を設定することができるビューです。

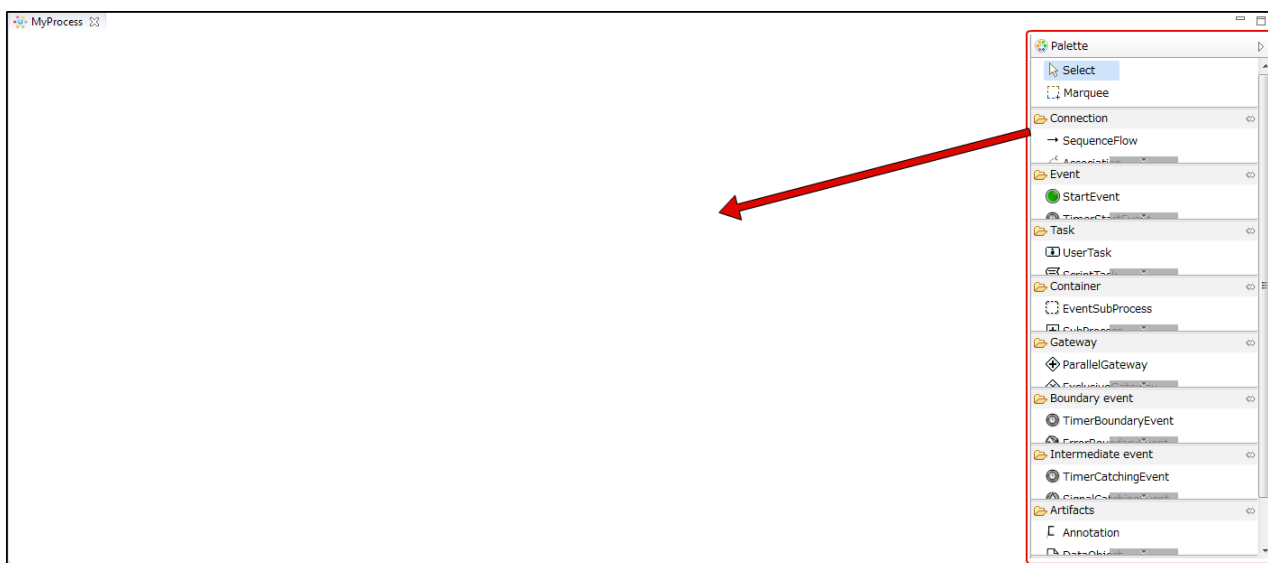


基本的な操作と設定

Paletteの操作

BPMダイアグラムはPaletteでフローエレメントを配置し、Properties ・ ビューで設定を定義します。

画面右側のPaletteから配置したいフローエレメントを画面中央部のBPM Diagram Editorにドラッグ&ドロップ等で配置します。



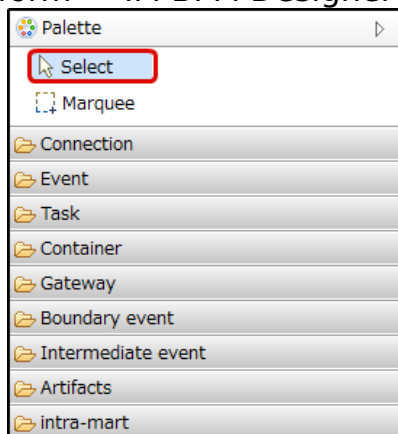
配置されたフローエレメントは、Selectを使う事で選択して設定の編集や配置の移動ができます。

Marqueeを使う事で一度に複数のフローエレメントを選択できます。

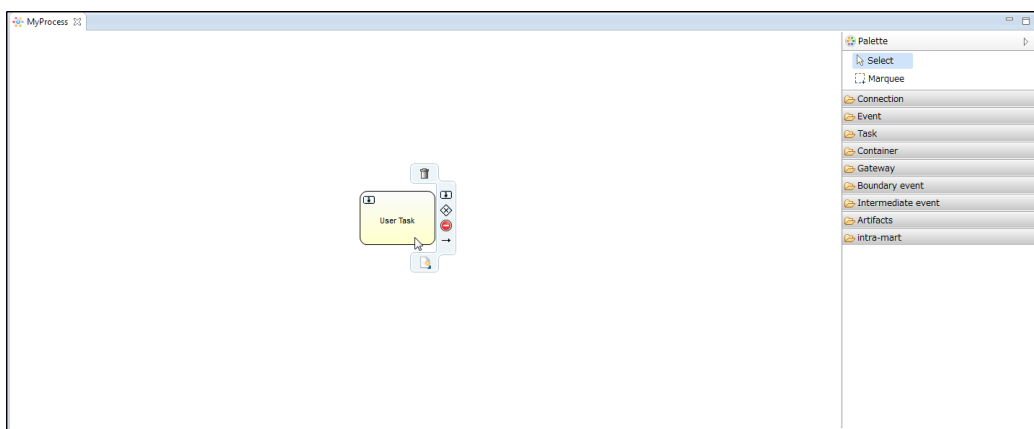
配置したフローエレメントの削除

- 配置したフローエレメントを削除します。
- ここでは、後述する **ユーザタスク** が配置されているものとして説明します。

1. PaletteからSelectをクリックします。

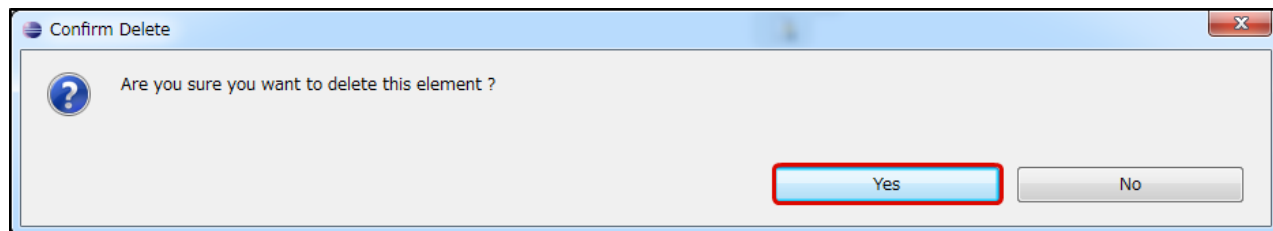


2. BPM Diagram Editorに配置されたユーザタスクにマウスポインタを合わせるとコンテキストボタンが表示されます。

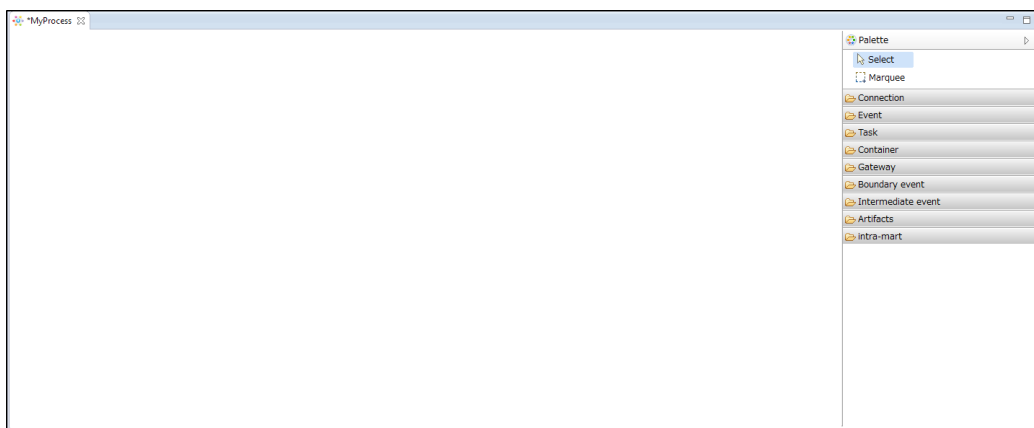


3. 表示されたメニューから  (ゴミ箱のアイコン) をクリックします。

4. Yesをクリックします。



5. これで配置したフローエレメントの削除が完了しました。



プロセス定義の設定

画面下部に表示されたProperties・ビューでプロセス定義の設定ができます。

項目名	説明
Id	プロセス定義を一意に特定するために設定します。 英数字を推奨します。
Name(名前)	プロセス定義の名前を設定します。
Namespace	名前空間を設定します。 ここで設定した名前空間がプロセス定義のカテゴリに指定されます。 通常はURL形式の名前空間を設定します。
Candidate start users(処理対象ユーザ)	定義したプロセスインスタンスを開始できるユーザです。 この項目に記述するユーザはユーザIDで記載します。 複数ユーザの指定はカンマで行います。
Candidate start groups(処理対象グループ)	定義したプロセスインスタンスを開始できる対象グループです。 この項目に記述するグループはロールIDとなり、対象となるユーザは指定したロールを保持するユーザです。 複数グループの指定はカンマで行います。
Do not allow users to set up business keys. (ユーザに業務キーの設定を許可しない)	プロセス開始時に業務キーの入力を許可するかを設定します。 この項目をtrueに設定した場合、プロセス開始時に業務キーの入力を許可しません。
Documentation	プロセス定義について詳細や備考を記載する項目です。

チュートリアル

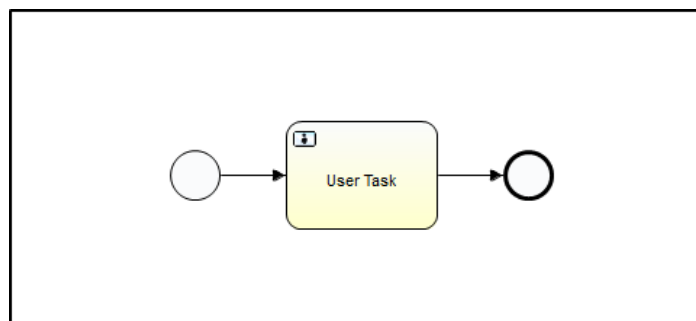
本項ではIM-BPM Designerを用いて、簡単な**ユーザタスク**を実行するBPMのプロセス定義を作成し、デプロイするまでの流れを説明します。

！ 注意

このチュートリアルを実行する前提として、動作させるintra-mart Accel Platformの環境が下記の条件を満たしていることとします。

- intra-mart Accel PlatformにIM-BPM for Accel Platformがインストールされている。
- サンプルデータがインストールされている(サンプルユーザ「aoyagi」でユーザタスクを実行するため)。

- チュートリアルで作成するBPMプロセスの完成イメージは以下のようになります。



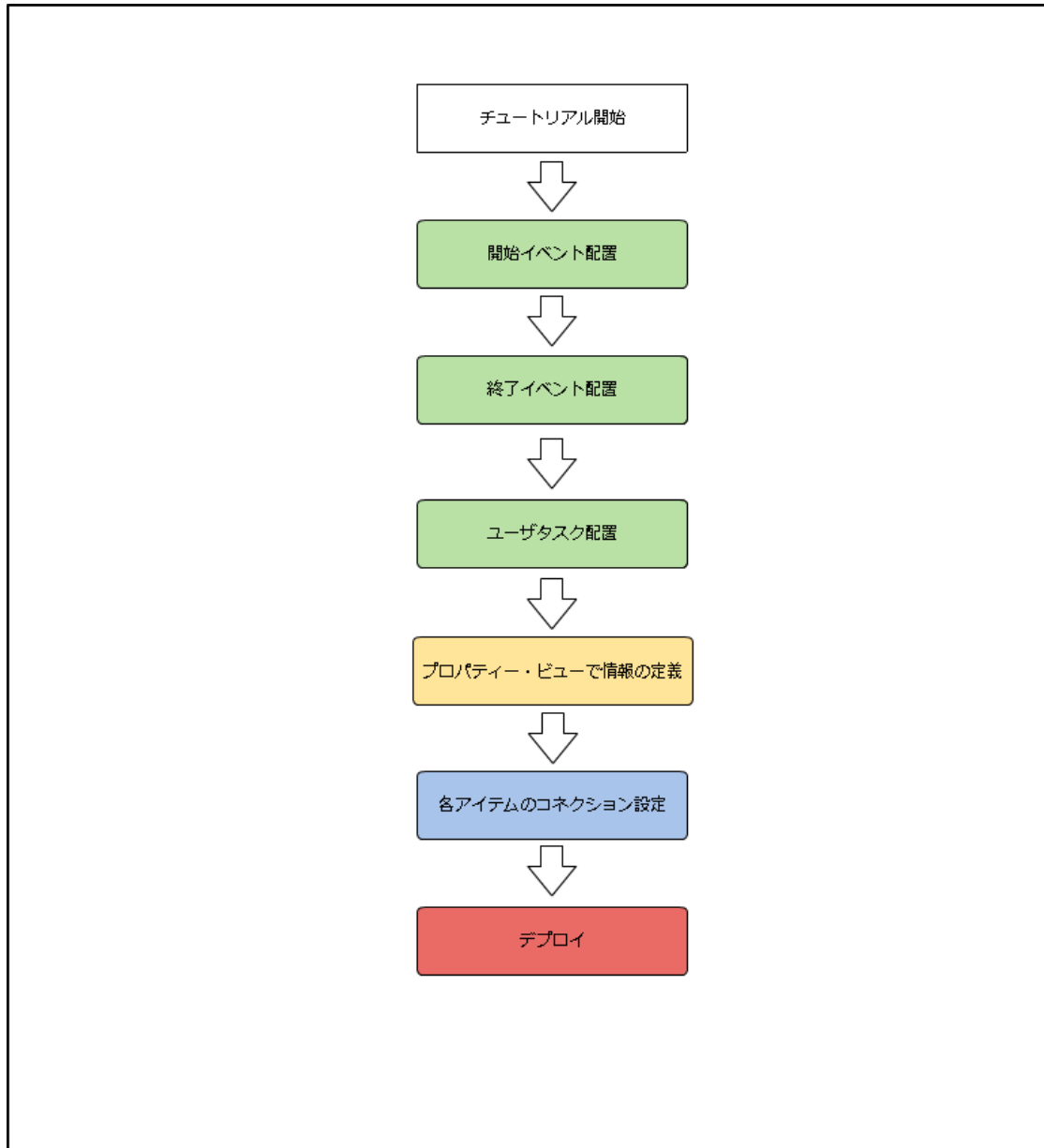
BPMのプロセス定義作成の流れ

このチュートリアルでは、以下の流れでBPMのプロセス定義を作成します。

1. [開始イベント](#)、[終了イベント](#)、[ユーザタスク](#)を配置する。
2. 各アイテムに対して実行する上で必要な情報をProperties・ビューで定義する
3. 各アイテムに対して[コネクション](#)を引いて、開始から終了までの流れを作成する。
4. 作成したファイルをintra-mart上にデプロイする

実際の作業において複雑なプロセス定義を作成する場合は、1.のアイテムを配置する作業から3.のコネクションを引くまでの作業を繰り返し実施し、BPMのプロセス定義を作成、改善していくことになります。

チュートリアル全体の流れは次の図の通りです。



i コラム

本チュートリアルでは開始イベント、終了イベント、ユーザタスクの3つのアイテムを用いて説明していますが、業務ケースやタスクの種類、条件などに応じて様々なパターンを表現できます。

どのような表現が可能かについて「[4.2. エディタ上で配置可能なアイテム](#)」を参照してください。

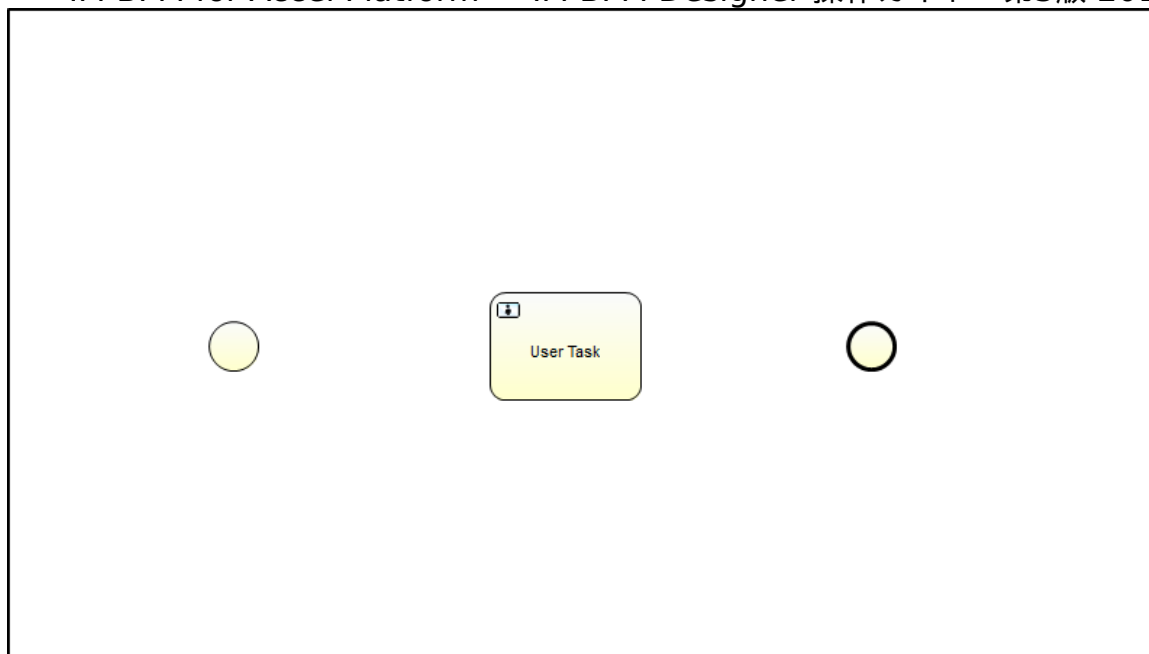
チュートリアルの実施

開始イベント、終了イベント、ユーザタスクを配置する

BPMエディタ上に [開始イベント](#)、[終了イベント](#)、[ユーザタスク](#)を配置します。

[開始イベント](#) と [終了イベント](#) はPaletteの「イベント」の中に、[ユーザタスク](#)はPaletteの「タスク」の中にあります。

配置の方法については「[4.1.5. 基本的な操作と設定](#)」を参照してください。

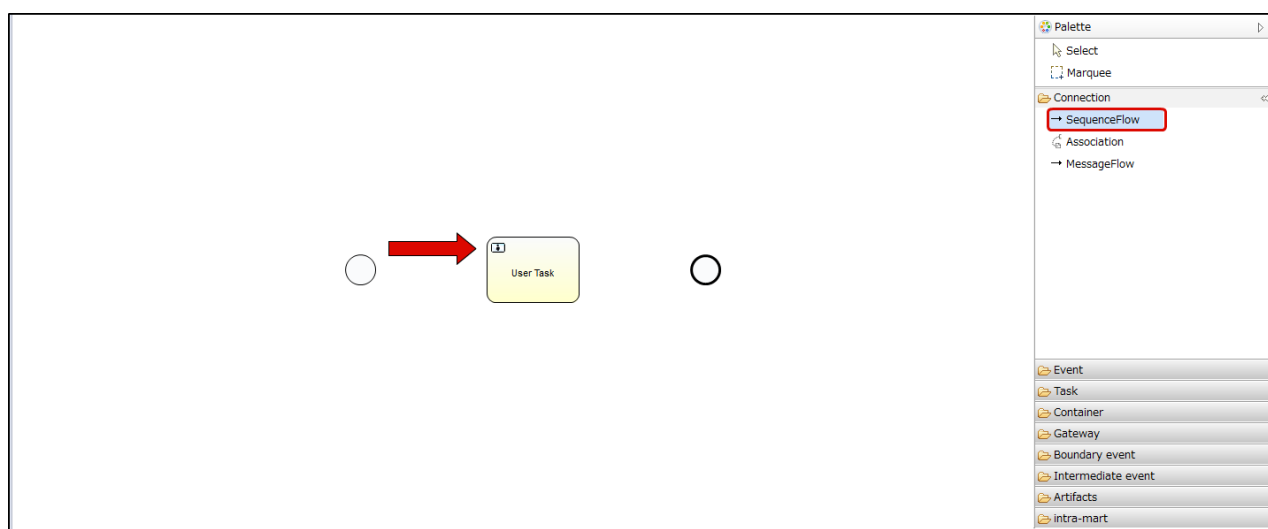


各アイテムに対してコネクションを引いて、開始から終了までの流れを作成する

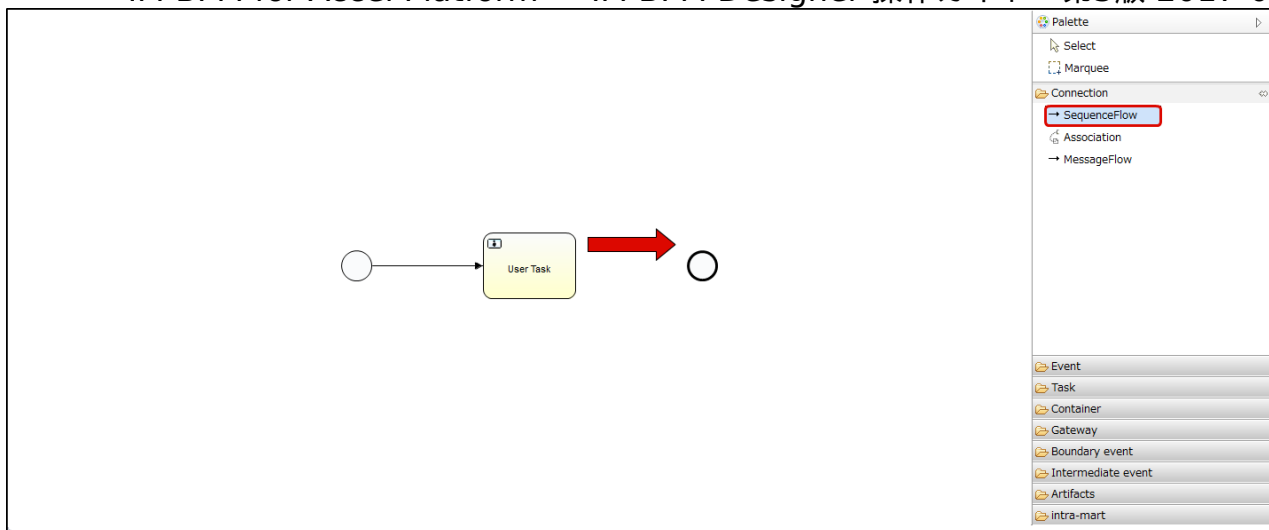
1. で配置したアイテムに対してシーケンスフローを引いて、開始イベントからユーザタスクを介して終了イベントにたどり着くまで、一連のフローになるように設定します。

PaletteのConnectionからSequenceFlowを選択し、以下のようにフローを設定してください。

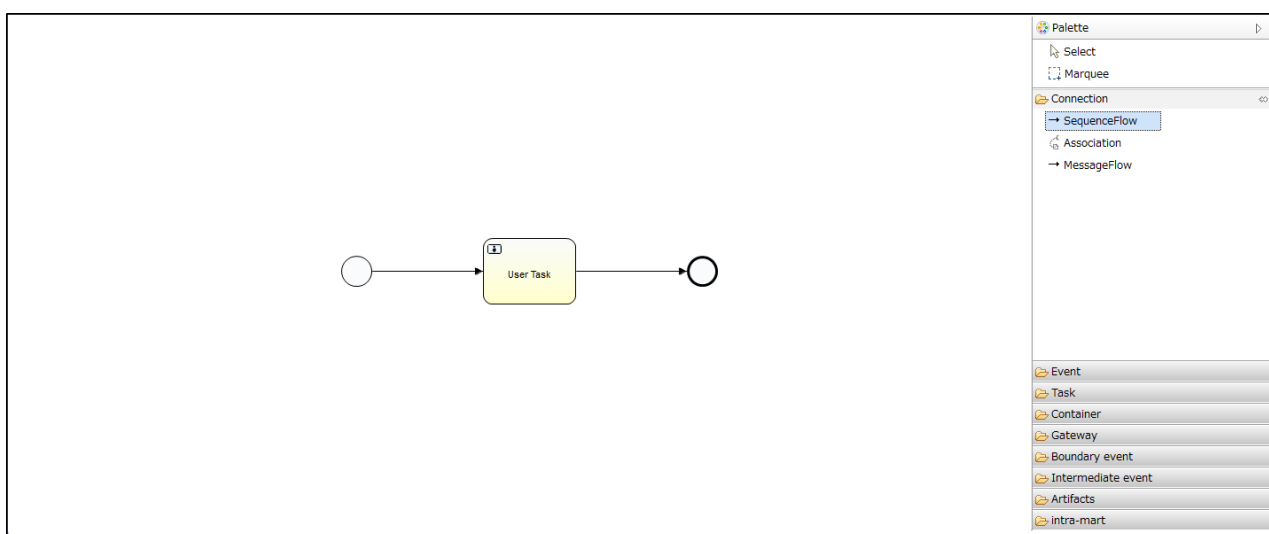
1. 開始イベントからユーザタスクまでシーケンスフローを引きます。
開始イベントをクリックしてから、ユーザタスクをクリックします。



2. ユーザタスクから終了イベントまでシーケンスフローを引きます。
ユーザタスクをクリックしてから、終了イベントをクリックします。



3. これでBPMのプロセス定義の作成が完了しました。

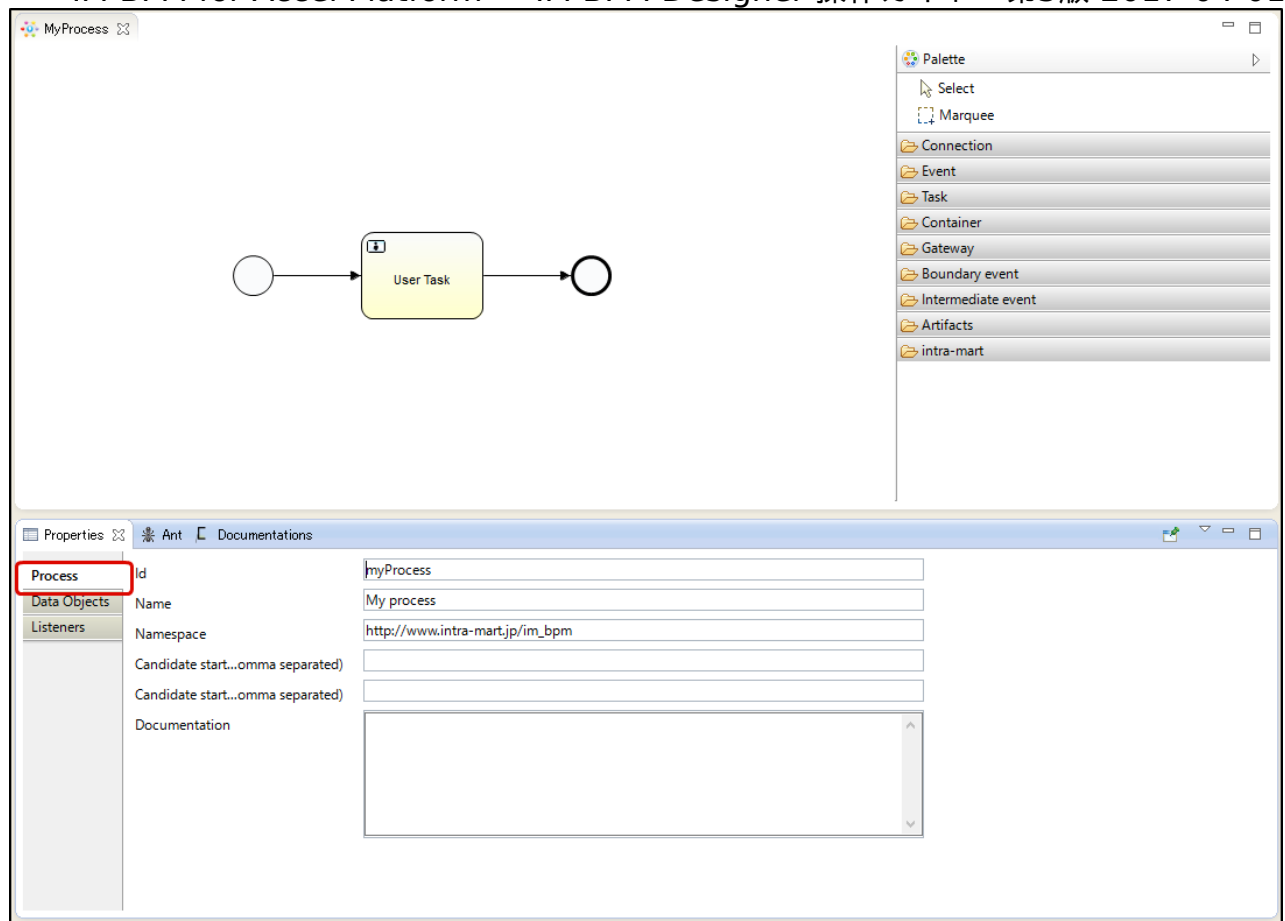


各アイテムに対して実行する上で必要な情報をプロパティー・ビューで定義する

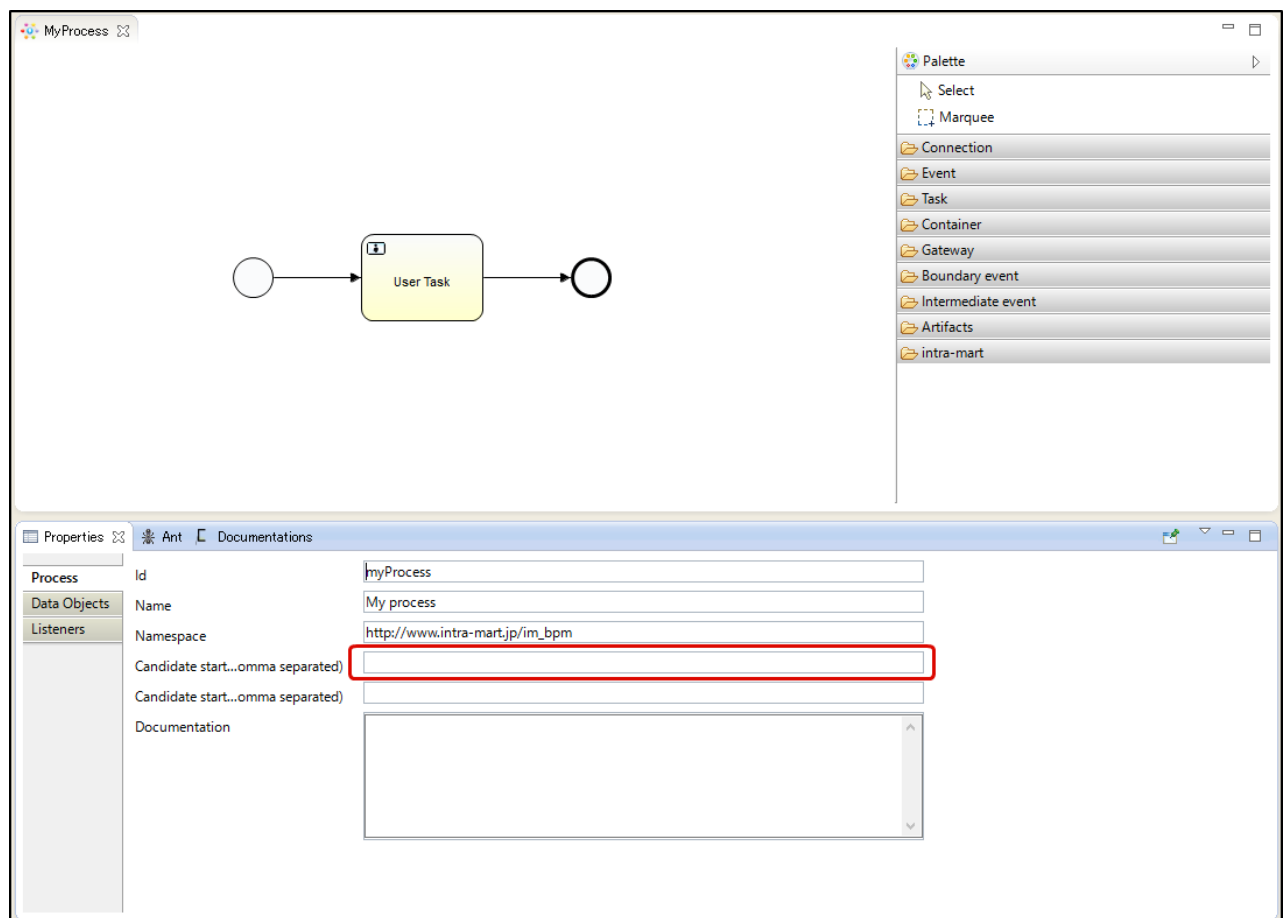
プロセスに対してプロパティー・ビューで以下の情報を設定します。

1. エディタの空白部分をクリックし、画面下部のProperties・ビューを選択します。

- BPMのプロセス定義全体の情報を編集する場合はBPM Diagram Editorの空白をクリックすると選択できます。



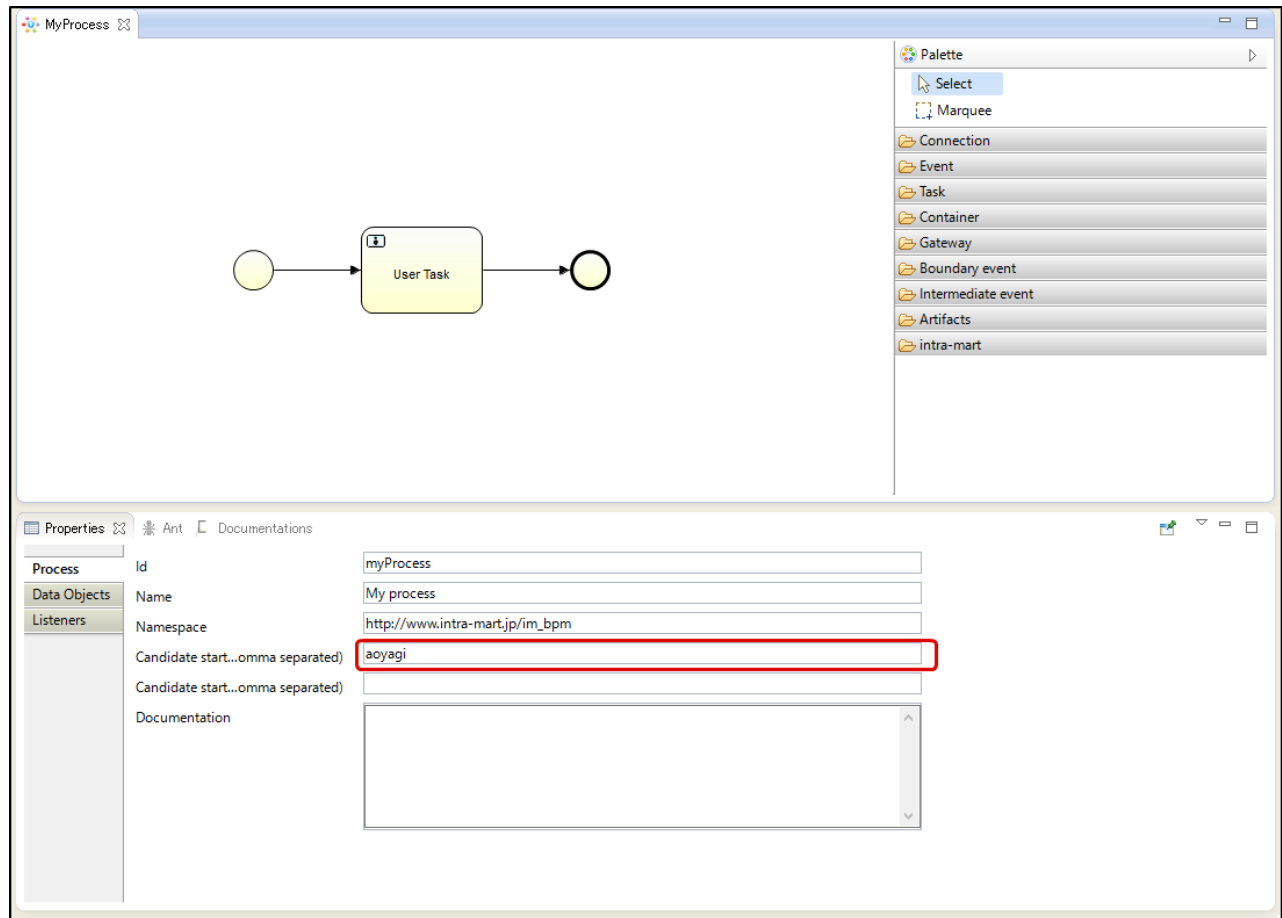
2. 1.で表示したProperties・ビューのCandidate start usersの項目にサンプルユーザのaoyagiさんのユーザIDを設定します。



- BPMのプロセス定義のプロパティに設定する情報は以下の通りです。

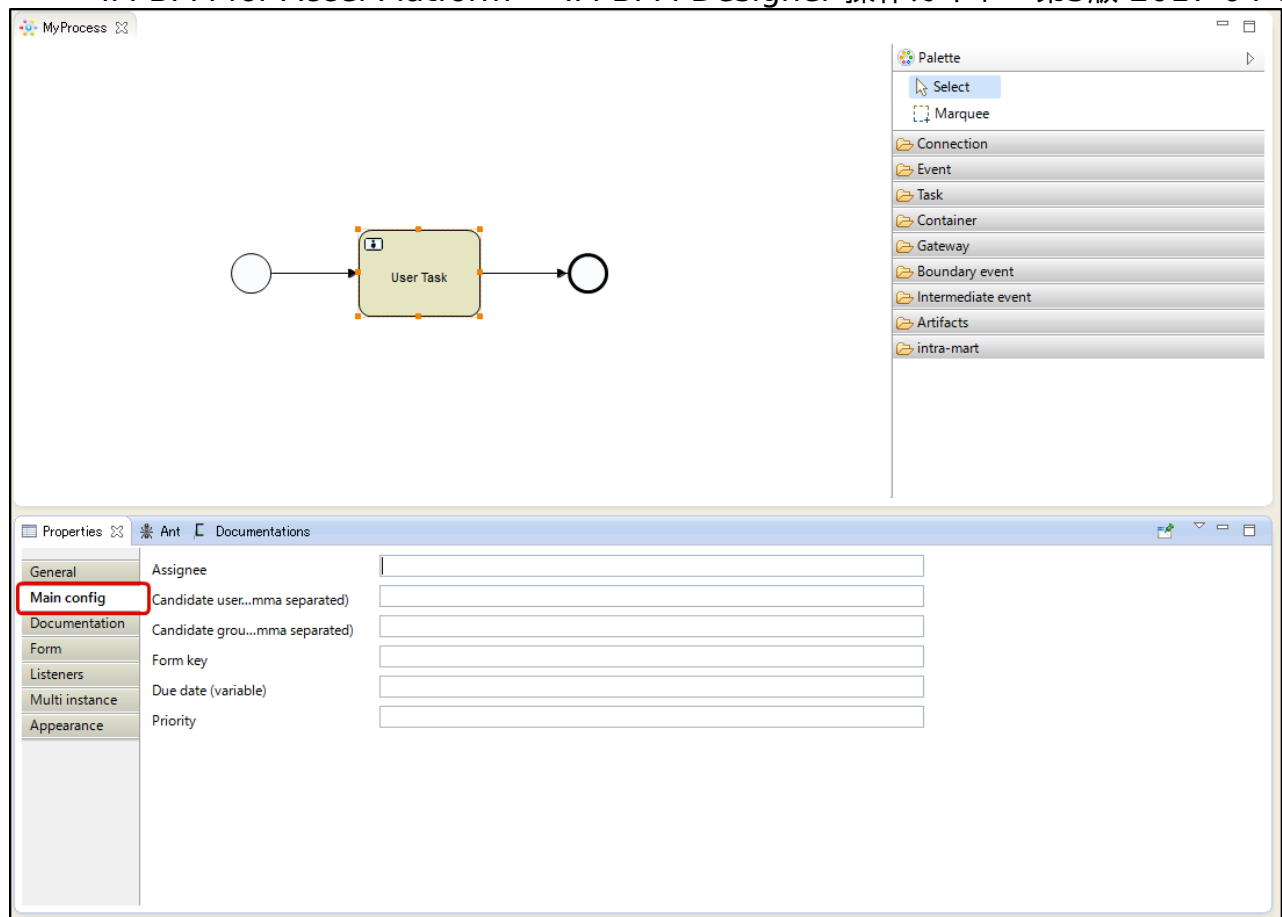
項目名	設定値
Candidate start users(処理対象ユーザ)	aoyagi

3. aoyagiさんがCandidate start usersに設定されている事が確認できれば設定は完了です。

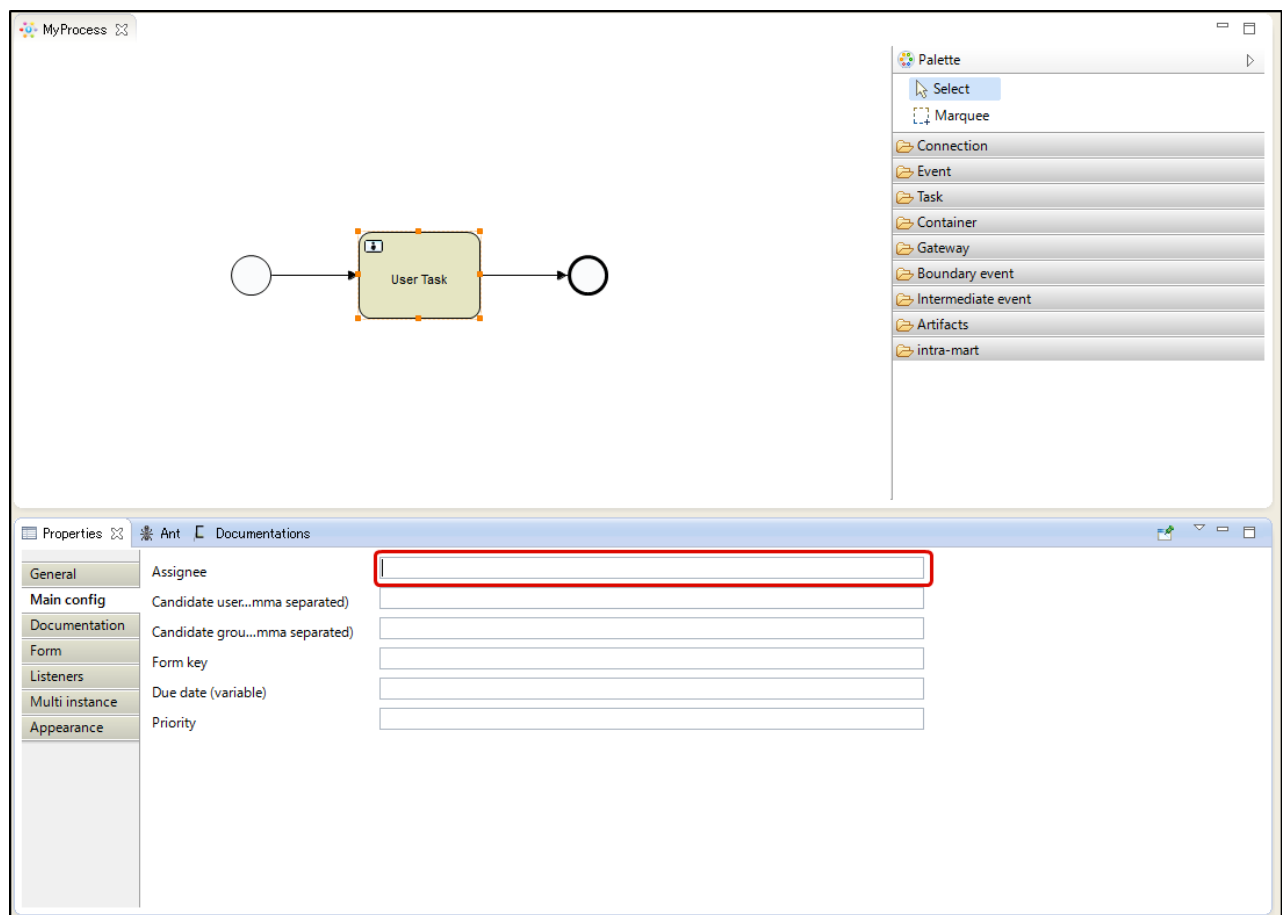


1. で配置したユーザタスクに対してProperties・ビューで以下の情報を設定します。

1. ユーザタスクを選択し、画面下部のProperties・ビューからMain configを選択します。



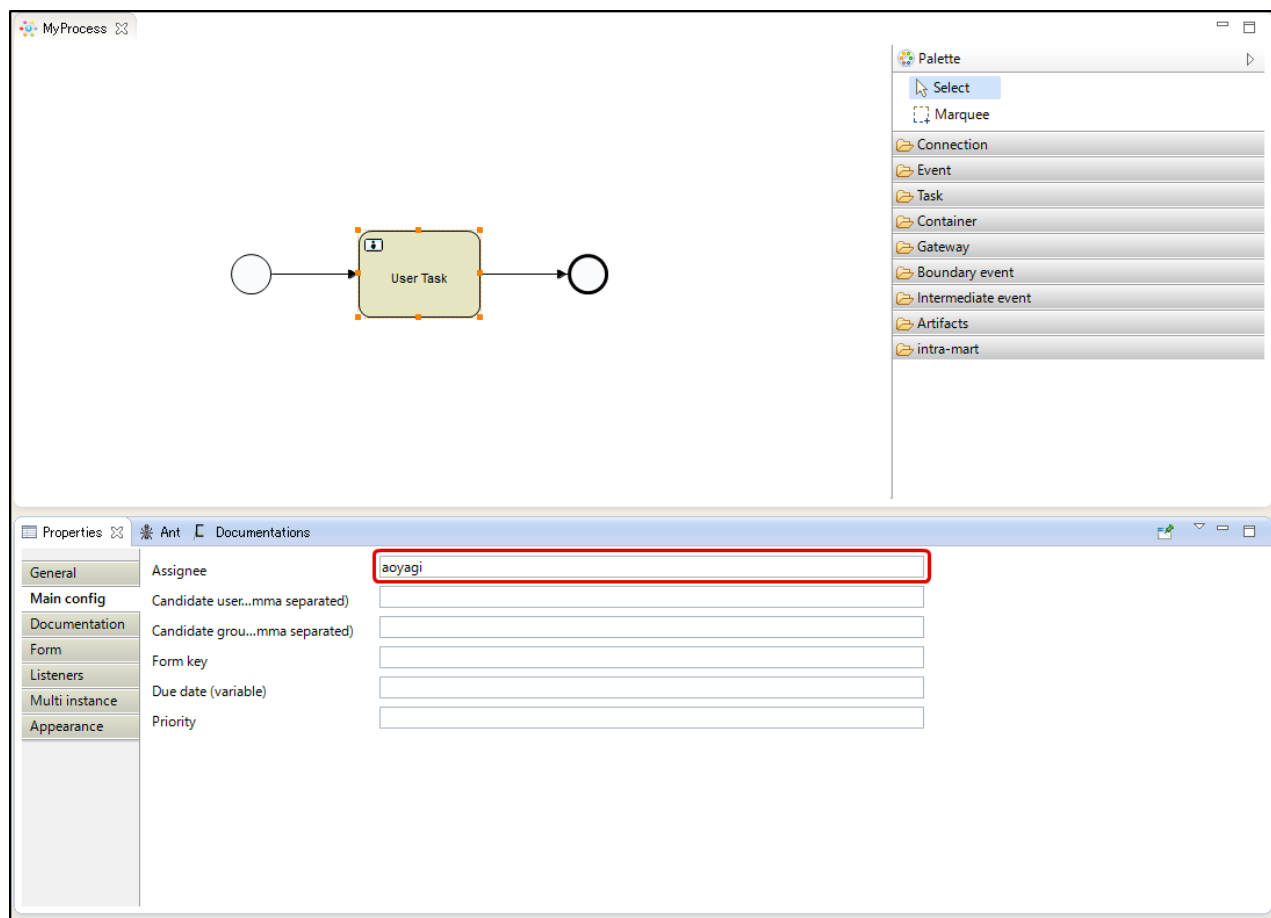
2. ユーザタスクをaoyagiさんが実行できるようにAssigneeを設定します。



- ユーザタスクのプロパティに設定する情報は以下の通りです。

項目名	設定値
Assignee(担当者)	aoyagi

3. aoyagiさんがAssigneeに設定されている事が確認できれば設定は完了です。



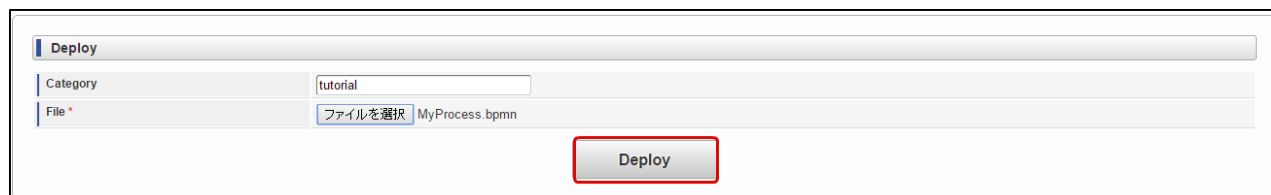
作成したファイルをintra-mart上にデプロイする

4.2.2.3で作成したファイルをintra-mart上にデプロイします。
デプロイの詳細については「[デプロイ 資料の作成](#)」を参照してください。

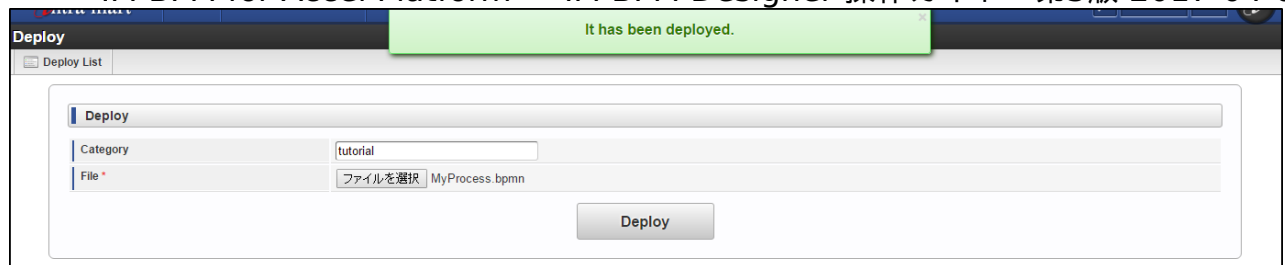
1. デプロイを行うためにカテゴリ名の入力とファイルの選択をします。
 - ここではカテゴリ名は「tutorial」、ファイルは「MyProcess.bpmn」とします。



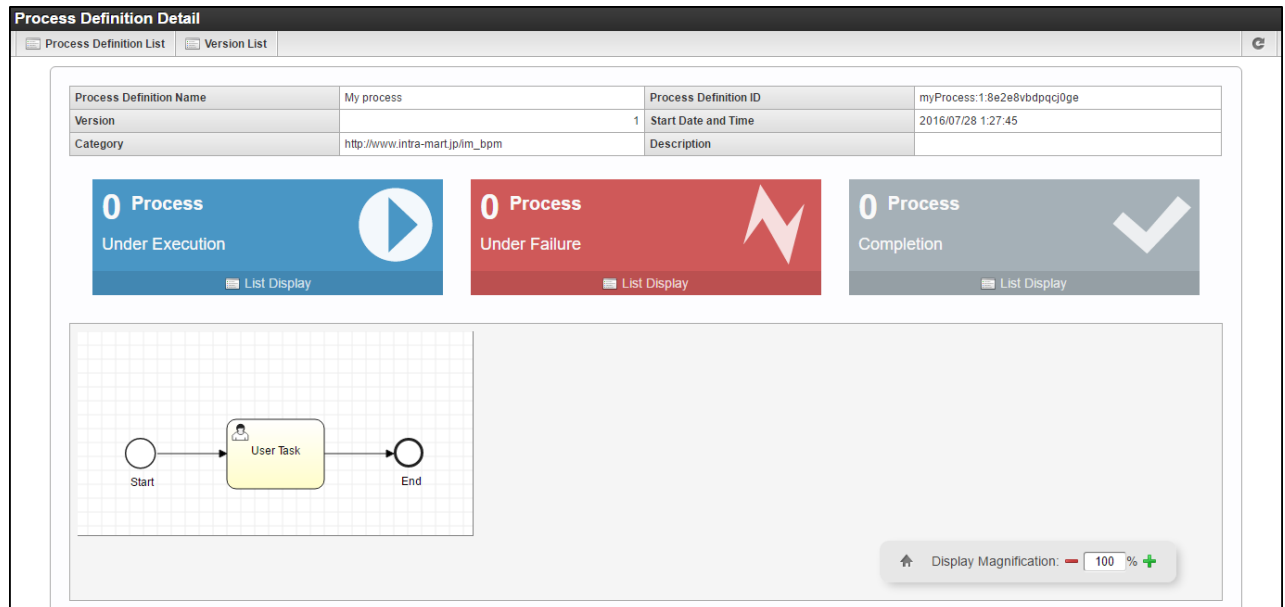
2. カテゴリ名の入力とファイルの選択が完了したら、デプロイをクリックします。



3. デプロイが成功すると「デプロイしました。」と表示されます。



4. プロセス定義詳細画面でデプロイしたBPMプロセス定義の詳細が確認できます。



エディタ上で配置可能なフローエレメント

本項では、BPMダイアグラムにおいてフローを表現するのに利用可能なフローエレメントについて説明します。
BPMダイアグラムのPaletteでは大きく分けて以下のメニューが用意されています。

コネクション

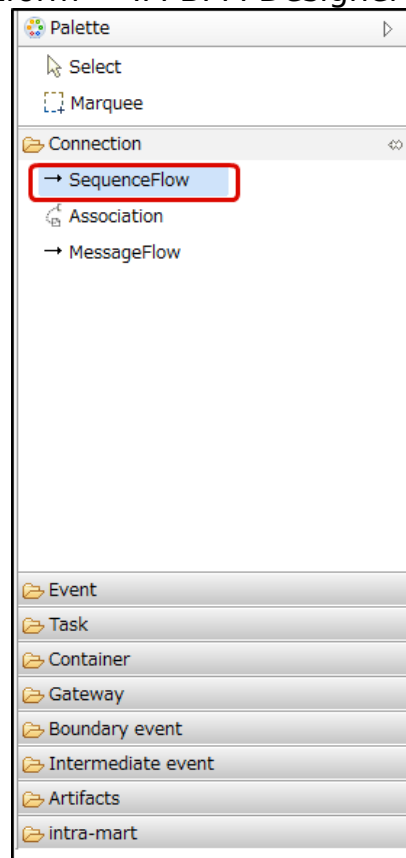
本項ではIM-BPM DesignerのPaletteにある「Connection」で定義されているコンポーネントの説明と定義方法について説明します。
コネクションはフローの流れや補足情報の関連付けを表現するために用いられます。

SequenceFlow(シーケンスフロー)

シーケンスフローとはアクティビティやイベント、ゲートウェイを接続して、プロセス定義の処理の流れを矢印付きの実線によって示したものです。

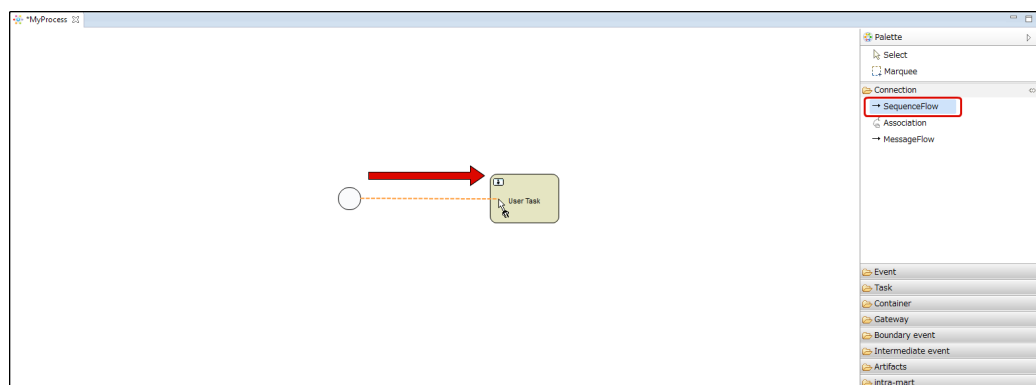
シーケンスフローの配置

1. IM-BPM Designerの画面右側PaletteのConnectionにあるSequenceFlowをクリックします。

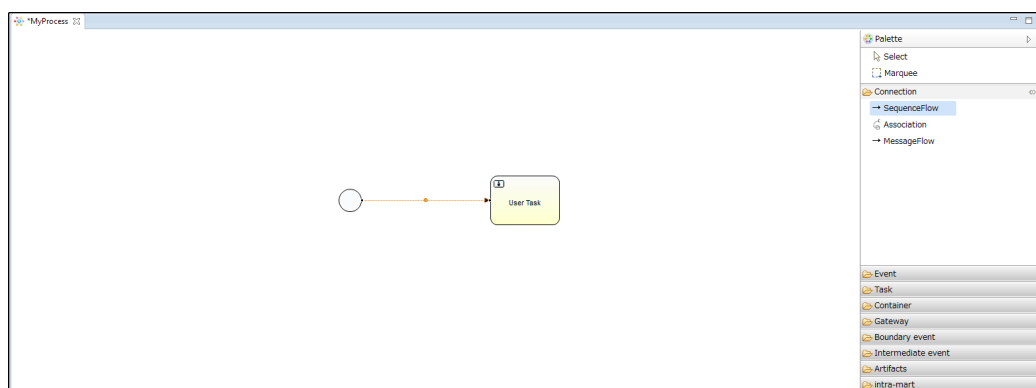


2. エディタに配置したシーケンスフローの接続を定義したいフローエレメントをクリックし、接続先のフローエレメントをクリックします。

- ここでは後述する開始イベントからユーザタスクへのシーケンスフローを定義します。



3. これで2つのフローエレメントのシーケンスの接続が定義されました。



プロパティの設定

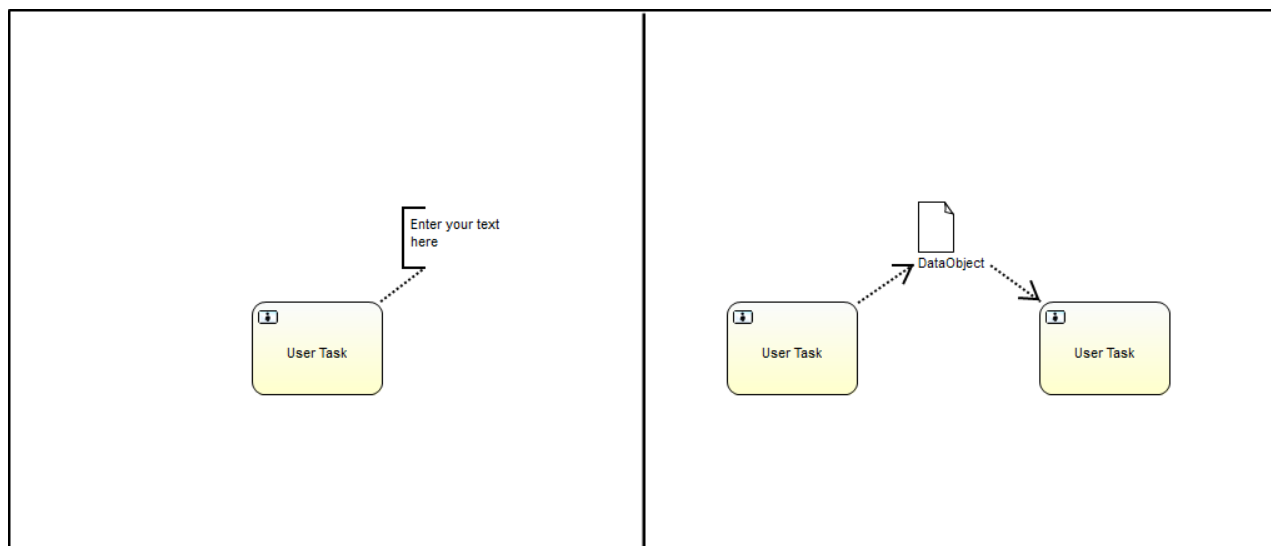
シーケンスフローのメインコンフィグの設定について解説します。

項目名	説明
Label width(ラベル幅)	ラベル幅を指定できます。
Skip expression(スキップ条件)	スキップ条件のEL式を記述します。 スキップ条件がtrueと評価された場合、シーケンスフローは進みます。 スキップ条件が評価された場合、条件は評価されません。 スキップ条件がfalseの場合、条件がtrueの場合もシーケンスフローは進みません。 また、スキップ条件がtrueの場合、条件がfalseの場合でもシーケンスフローは進みます。
Condition (条件)	シーケンスフローの進行条件を定義できます。 条件が真となる場合のみ、シーケンスフローは進みます。 この条件は、分岐が行われる場合にのみ評価されます。

Association(アソシエーション)

アソシエーションはアクティビティに対して、注釈やデータオブジェクトを関連付ける際に引かれる破線のことを指します。
アソシエーションは各アクティビティに対して、説明の補足や関連成果物があることを表し、システム上のフローの動作には影響はありません。
接続先が注釈の場合は単なる破線で、データオブジェクトの場合は矢印の破線で表現されます

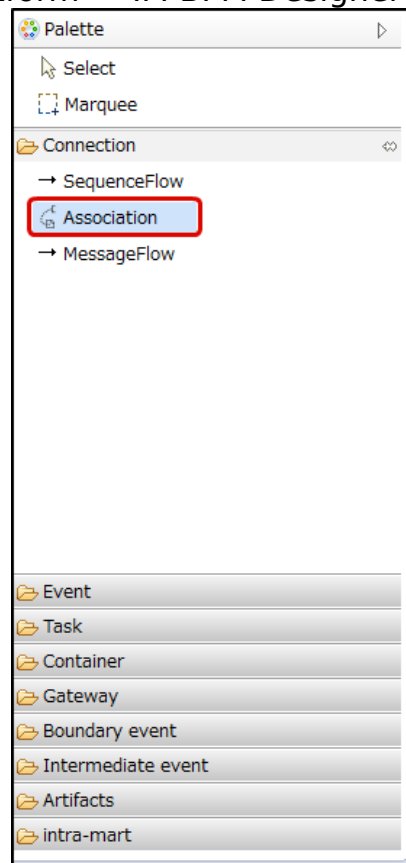
- 左：破線、右：矢印の破線



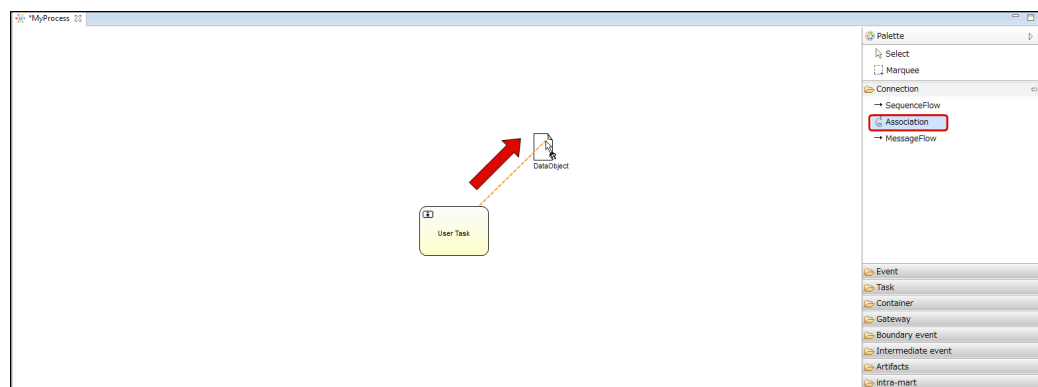
アソシエーションの配置

ここではユーザタスクに対して、データオブジェクトで定義された出力成果物を、アソシエーションを用いて表現する方法について説明します。
なお、ユーザタスクとデータオブジェクトは予めエディタ上に表示されているものとします。

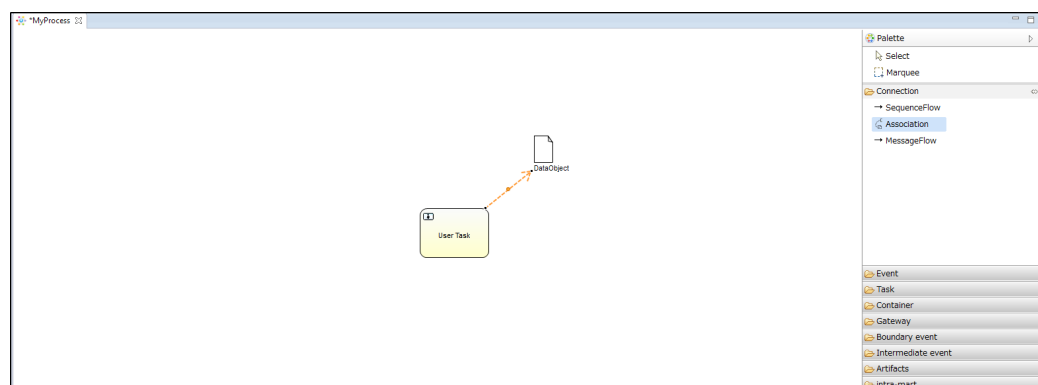
1. IM-BPM Designerの画面右側PaletteのConnectionにあるAssociationをクリックします。



2. エディタ上にあるユーザタスクをクリックした後、データオブジェクトをクリックします。



3. これでデータオブジェクトを出力成果物として表現するアソシエーションを定義することができました。



コラム

タスクに対する入力成果物を表現したい場合はアソシエーションをデータオブジェクトからタスクへと引くように設定してください。注釈に対するアソシエーションに関しては特に起点の制限はありません。

MessageFlow(メッセージフロー)

プロセス定義における関係者(プール)間で情報が送受信される事を破線の矢印で表現します。

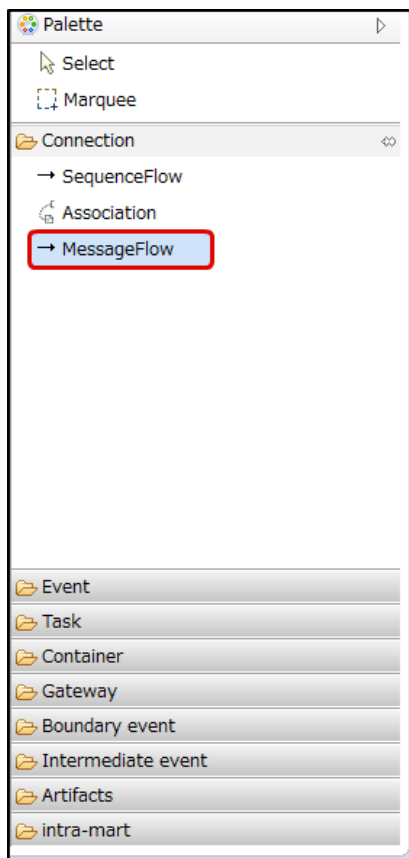
同じプール内のフローエレメント同士で、メッセージフローを接続する事はできません。

IM-BPM for Accel Platformではプロセス定義の図上の表現であり、実際のプロセスフローの動作に対して影響しません。

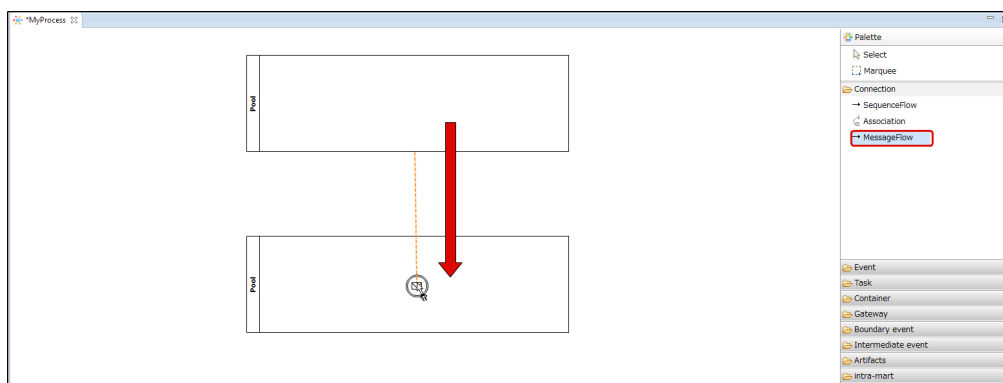
メッセージフローの配置

- ここではプールから別のプールにあるメッセージキャッチイベントへのメッセージフローを定義します。

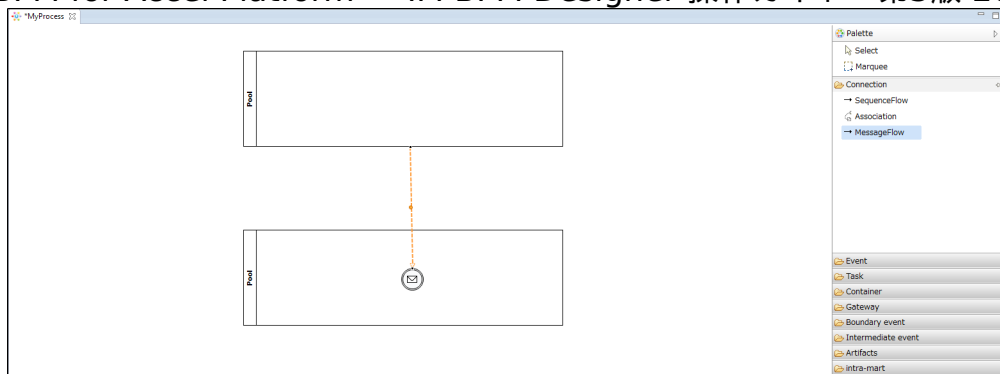
1. IM-BPM Designerの画面右側PaletteのConnectionにあるMessageFlowをクリックします。



2. エディタ上のメッセージフローの接続を定義したいプールをクリックし、接続先のメッセージキャッチイベントをクリックします。



3. これでメッセージフローが定義されました。



イベント

本項ではIM-BPM DesignerのPaletteにある「Event」で設定可能なコンポーネントの説明と定義方法について説明します。
イベントは、一連のフローの起点や終点を表現するために用いられます。

開始イベント

開始イベントはプロセス定義の実行開始箇所と条件を表します。
プロセス定義には、開始イベントと後述する終了イベントが存在するように設定する必要があります。
開始イベントは、開始条件によって、4つに分類されます。

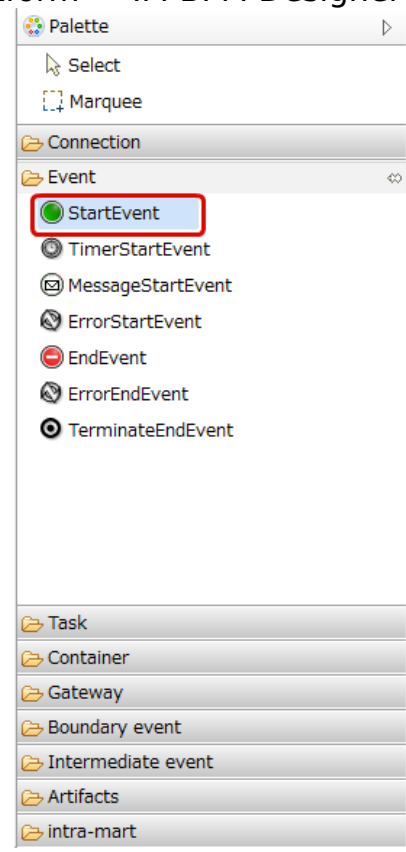
- [StartEvent\(開始イベント\)](#)
- [TimerEvent\(タイマー開始イベント\)](#)
- [SignalEvent\(シグナル開始イベント\)](#)
- [MessagesStartEvent\(メッセージ開始イベント\)](#)
- [ErrorStartEvent\(エラー開始イベント\)](#)

StartEvent(開始イベント)

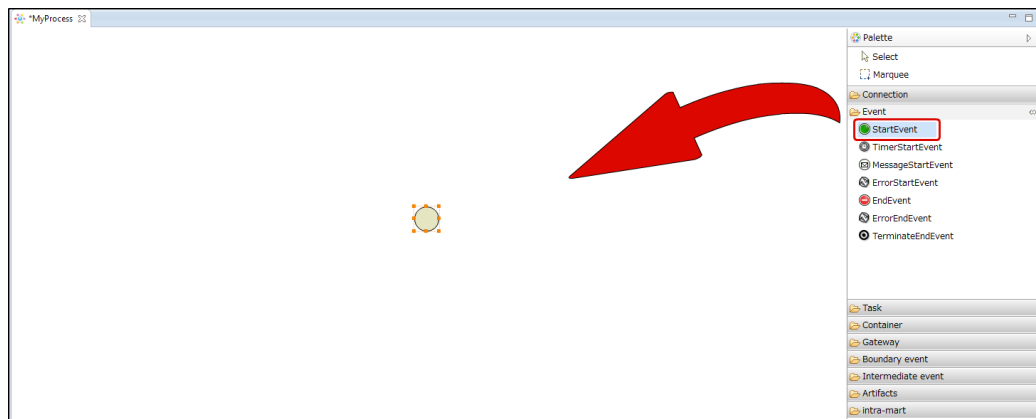
開始する条件を持たないプロセスフローの起点を表します。

開始イベントの配置

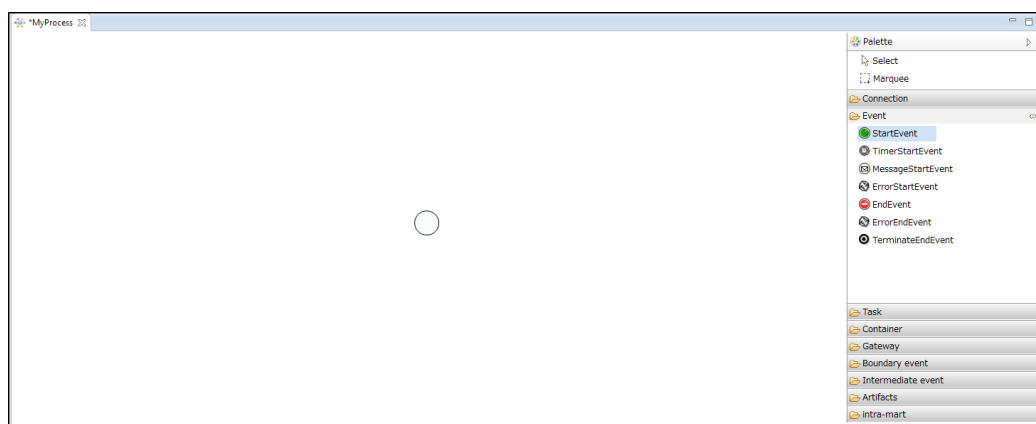
1. IM-BPM Designerの画面右側PaletteのEventにあるStartEventをクリックします。



2. エディタ上のプロセス定義の実行を開始したい場所をクリックします。

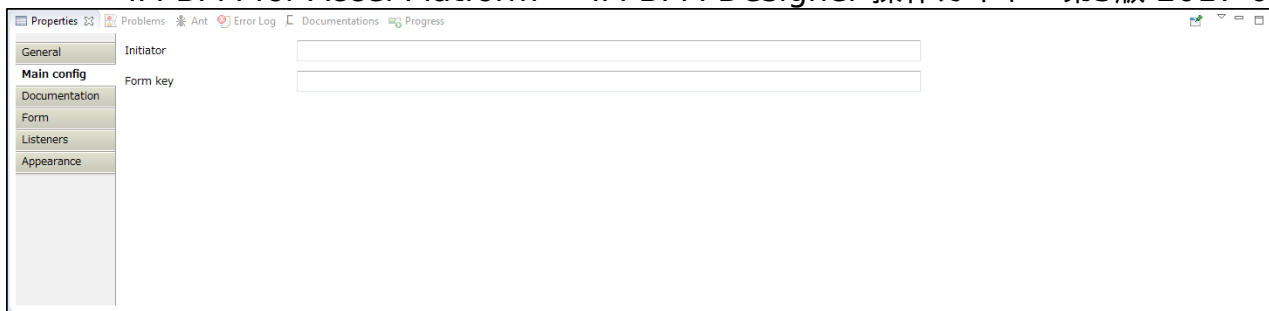


3. これで開始イベントが定義されました。



プロパティの設定

開始イベントのメインコンフィグの設定について解説します。



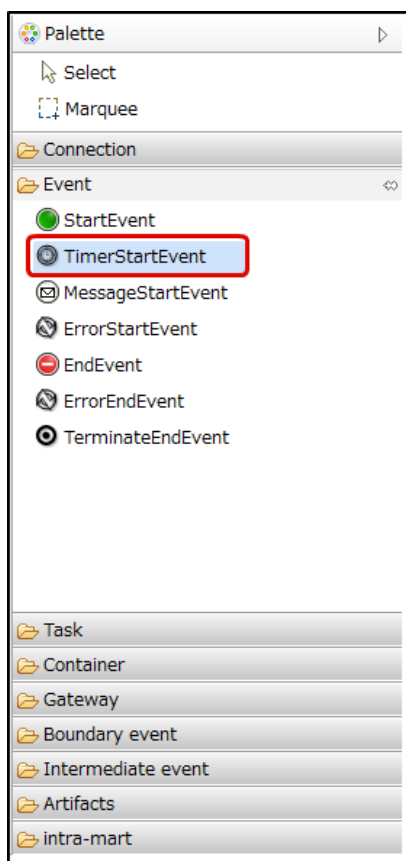
項目名	説明
Initiator(イニシエータ)	プロセス定義を実行する権限を持つユーザのユーザIDを設定します。
Form key(フォームキー)	呼び出すアプリケーションと画面IDを設定します。 例1: formaのアプリケーション「application_id」を呼び出す場合 → forma:application_id 例2: スクラッチ開発した画面「http://screen/url」を呼び出す場合 → forward:http://screen/url

TimerStartEvent(タイマー開始イベント)

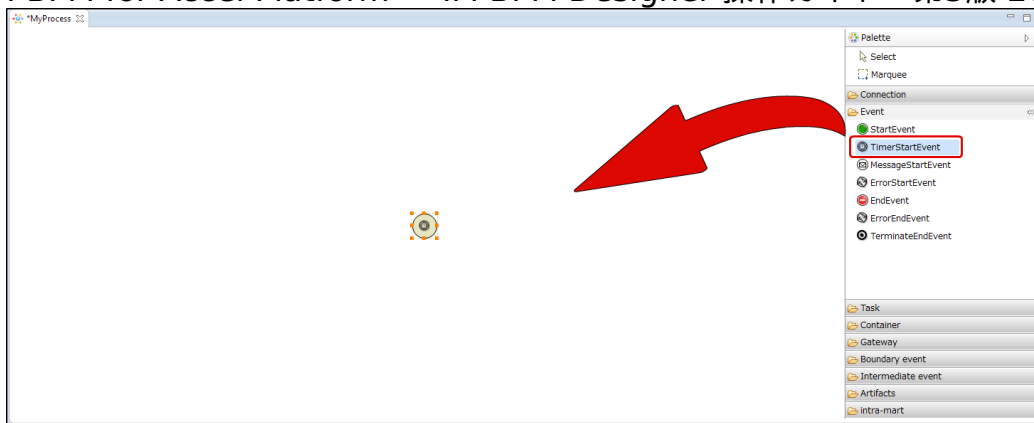
特定の日時や繰り返しなどのスケジュールによって開始されるプロセスフローの起点を表します。

タイマー開始イベントの配置

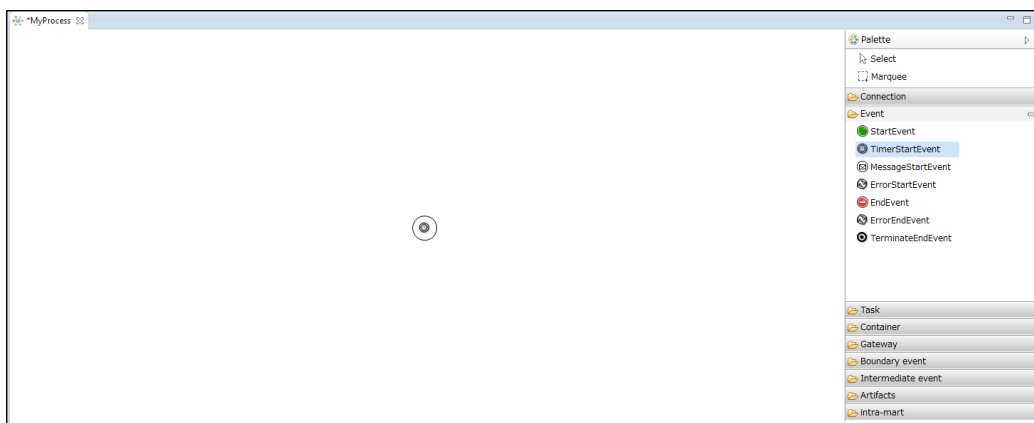
1. IM-BPM Designerの画面右側PaletteのEventにあるTimerStartEventをクリックします。



2. エディタ上のプロセス定義の実行を開始したい場所をクリックします。



3. これでタイマー開始イベントが定義されました。



プロパティの設定

タイマー開始イベントのメインコンフィグの設定について解説します。

入力方式は日付と時刻の表記に関する国際規格であるISO 8601の形式に則って記述する必要があります。

具体的な記述方法については説明欄を参照してください。

項目名	説明
Time duration(期間)	開始するまでの期間を指定します。 デプロイしたときから指定した期間が経過したとき1回のみ開始されます。 例:10日後開始→P10D
Time date(日付)	開始する日付を指定します。 指定日に1回のみ開始されます。 例:2016年1月1日12時開始→ 2016-01-01T12:00:00
Time cycle(周期)	開始する周期を指定します。 例:10時間毎に3回開始→R3/PT10H

項目名	説明
Time duration(期間)	開始するまでの期間を指定します。 デプロイしたときから指定した期間が経過したとき1回のみ開始されます。 例:10日後開始→P10D
Time date(日付)	開始する日付を指定します。 指定日に1回のみ開始されます。 例:2016年1月1日12時開始→ 2016-01-01T12:00:00
Time cycle(周期)	開始する周期を指定します。 例:10時間毎に3回開始→R3/PT10H

SignalStartEvent(シグナル開始イベント)

シグナルを受信する事で開始されるプロセスフローの起点を表します。

シグナル開始イベントの配置

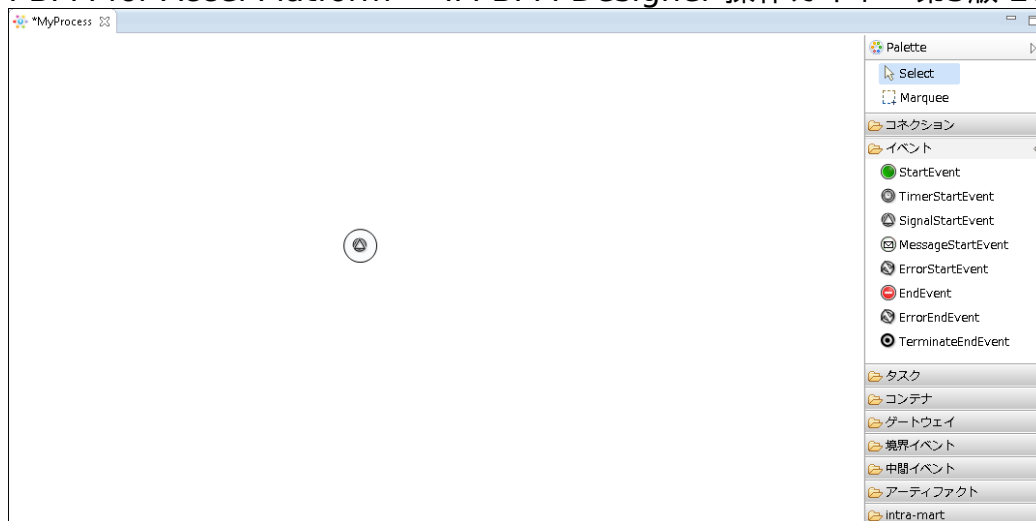
1. IM-BPM Designerの画面右側PaletteのEventにあるSignalStartEventをクリックします。



2. エディタ上のプロセス定義の実行を開始したい場所をクリックします。

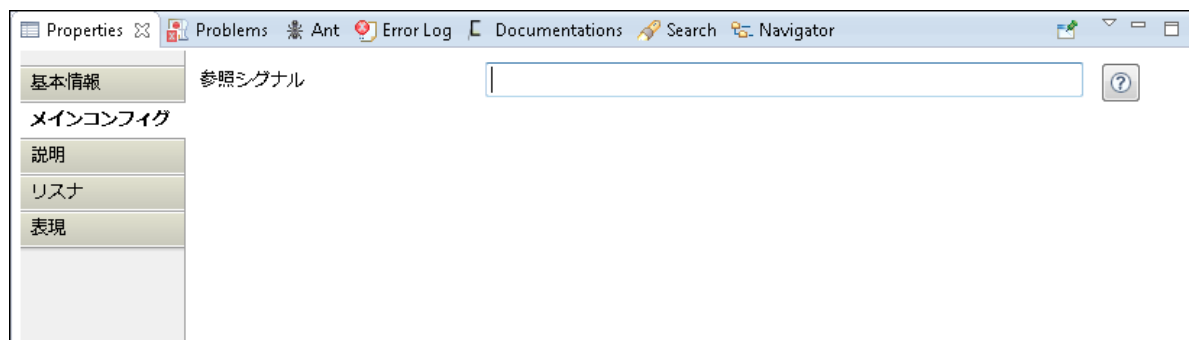


3. これでシグナル開始イベントが定義されました。



プロパティの設定

シグナル開始イベントのメインコンフィグの設定について解説します。



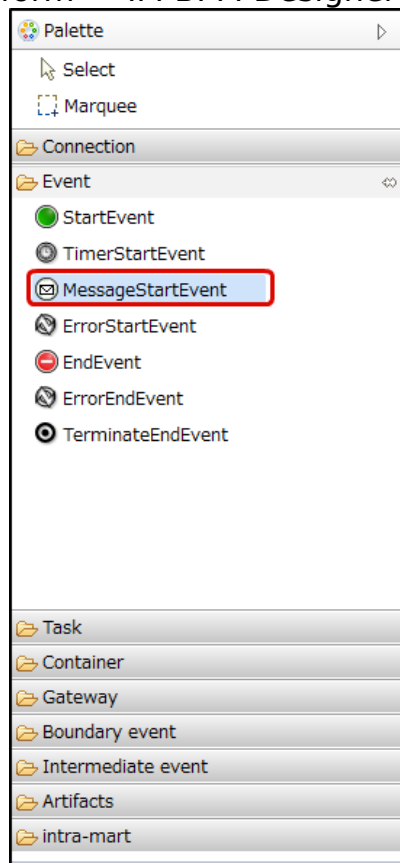
項目名	説明
Signal ref(参照シグナル)	受信するシグナルを文字列で定義します。 ここで定義したシグナルを受信すると、参照シグナルを起点としたプロセスフローが実行されます。

MessageStartEvent(メッセージ開始イベント)

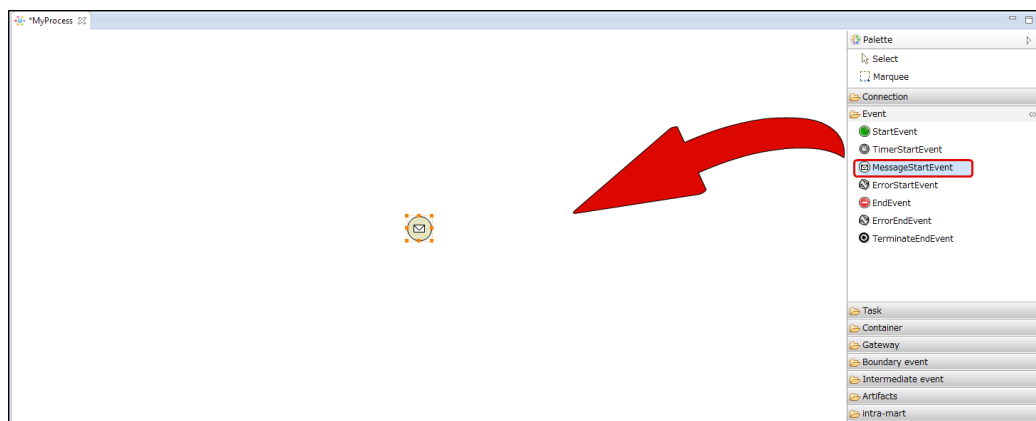
メッセージを受信する事で開始されるプロセスフローの起点を表します。

メッセージ開始イベントの配置

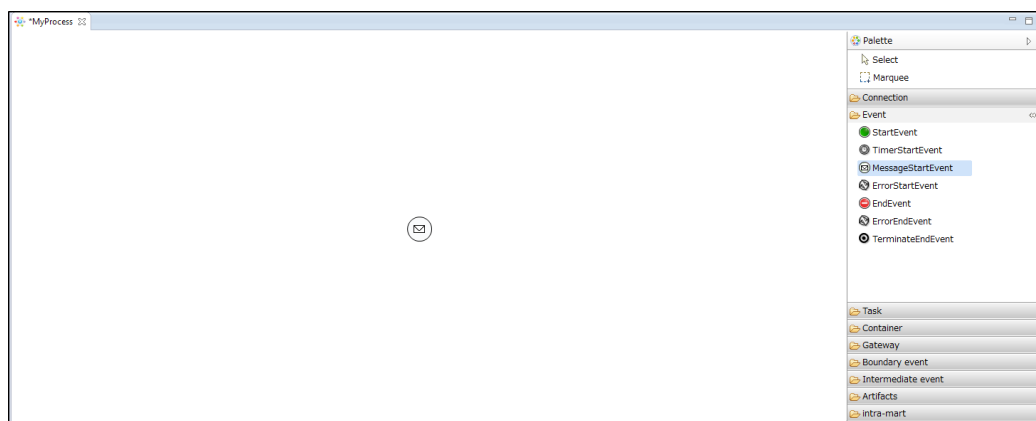
1. IM-BPM Designerの画面右側PaletteのEventにあるMessageStartEventをクリックします。



2. エディタ上のプロセス定義の実行を開始したい場所をクリックします。

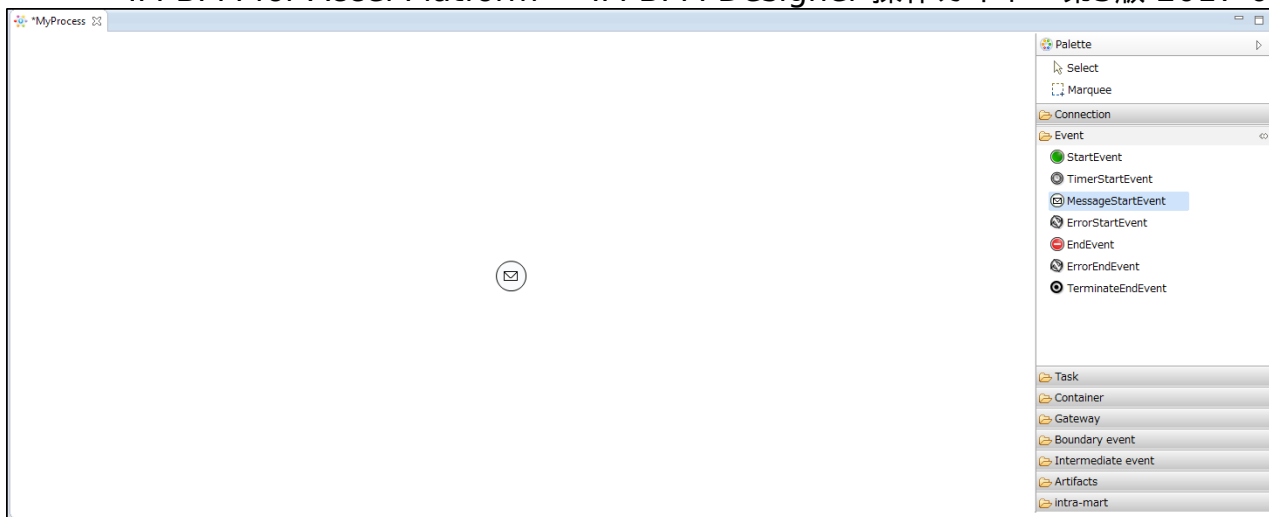


3. これでメッセージ開始イベントが定義されました。



プロパティの設定

メッセージ開始イベントのメインコンフィグの設定について解説します。



項目名	説明
Message ref(参照メッセージ)	受信するメッセージを文字列で定義します。 ここで定義したメッセージを受信すると、メッセージ開始イベントを起点としたプロセスフローが実行されます。

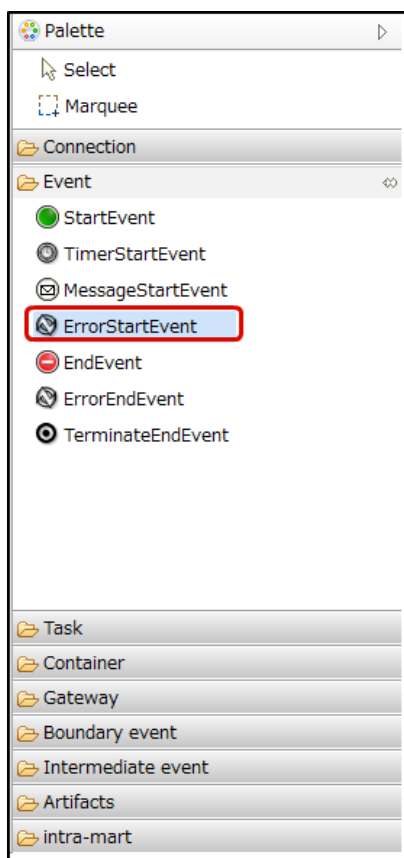
ErrorStartEvent(エラー開始イベント)

イベントサブプロセスでのみ使用できるイベントで、親のプロセス定義からのエラーを受信して開始されるプロセスフローの起点を表します。

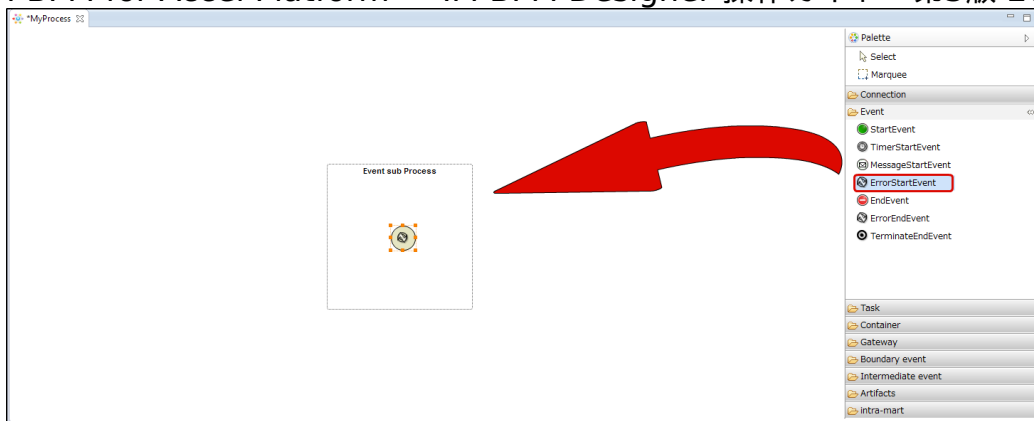
エラー開始イベントの配置

エラー開始イベントの配置には、事前に [イベントサブプロセス](#) を配置している必要があります。

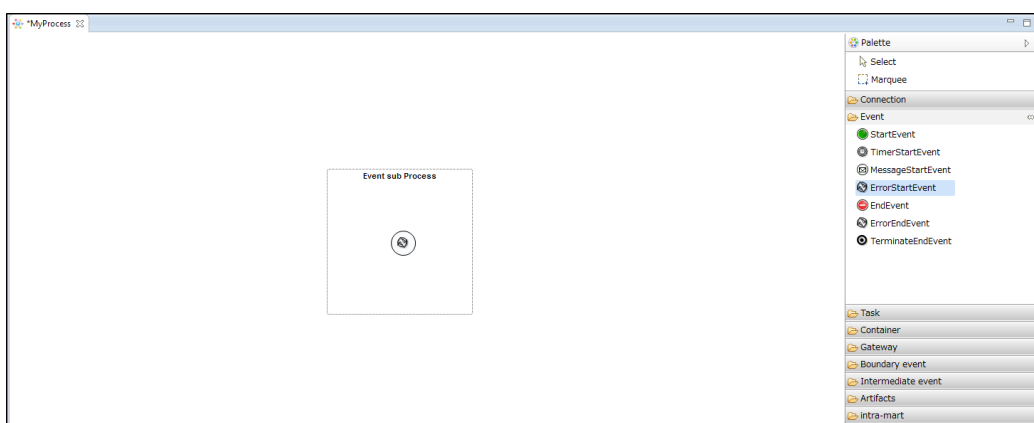
1. IM-BPM Designerの画面右側EventにあるErrorStartEventをクリックします。



2. エディタ上のイベントサブプロセス内でプロセス定義の実行を開始したい場所をクリックします。

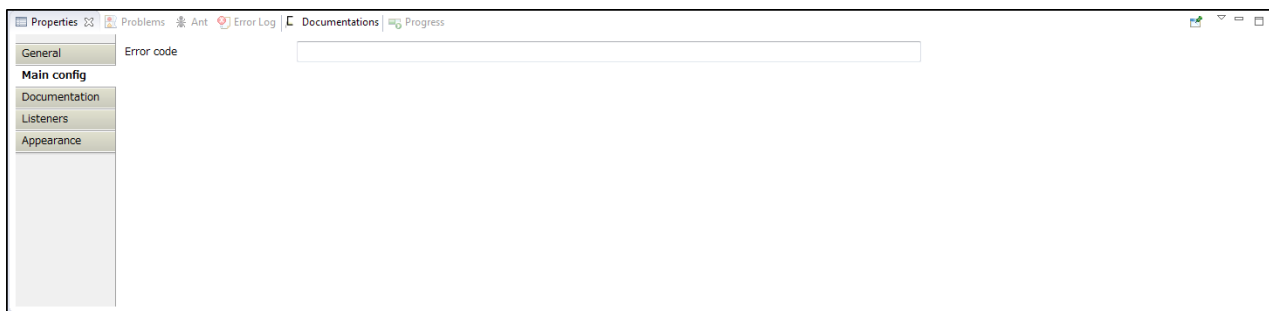


3. これでメッセージ開始イベントが定義されました。



プロパティの設定

エラー開始イベントのメインコンフィグの設定について解説します。



項目名	説明
Error code(エラーコード)	受信するエラーコードを文字列で定義します。 このエラーコードのエラーが発生するとエラー開始イベントを起点としたプロセスフローが実行されます。

終了イベント

終了イベントはプロセス定義の終了箇所や実行結果を表します。

開始イベント1つに対して終了イベントが複数存在する場合もあります。

複数の分岐したプロセスフローにおいて、どれか1つでも終了イベントに到達した場合でも、並行で動作しているタスク等に関しては継続して処理が進みます。

終了イベントは、終了した結果として何を行うかで、3つに分類されます。

- [EndEvent\(終了イベント\)](#)
- [ErrorEndEvent\(エラー終了イベント\)](#)
- [TerminateEndEvent\(強制終了イベント\)](#)

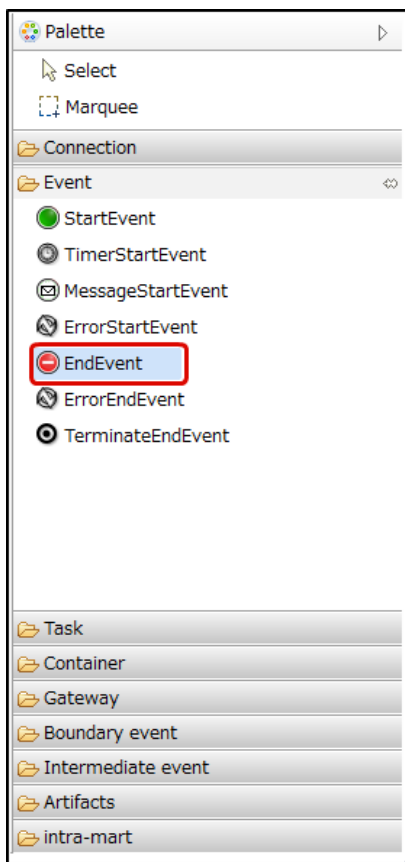
EndEvent(終了イベント)

プロセス定義の完了を表します。

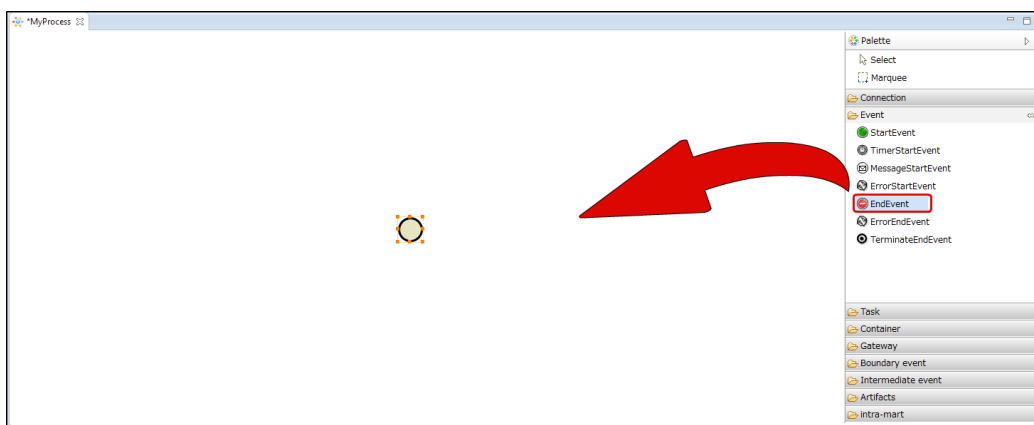
サブプロセスで使用された場合は、親のプロセス定義に処理が戻る事を表します。

終了イベントの配置

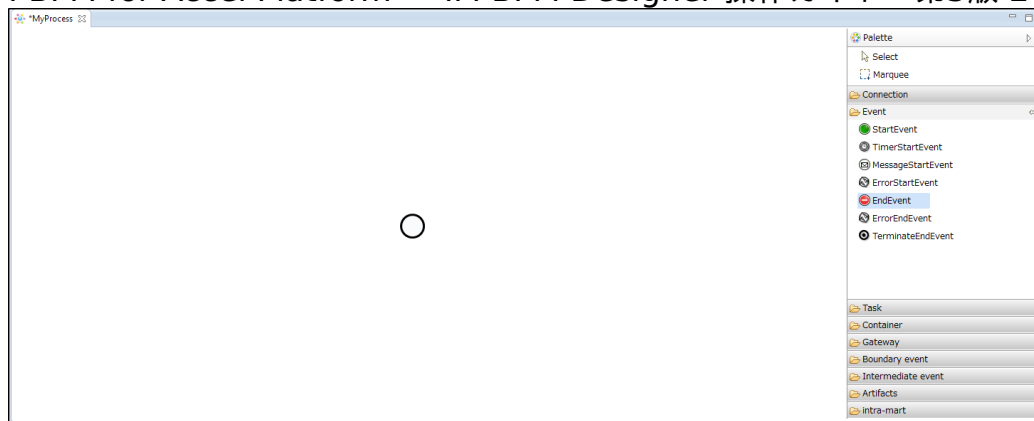
1. IM-BPM Designerの画面右側PaletteのEventにあるEndEventをクリックします。



2. エディタ上でプロセス定義の終了を定義したい場所をクリックします。



3. これで終了イベントが定義されました。



ErrorEndEvent(エラー終了イベント)

エラーコードを送信する事を表します。

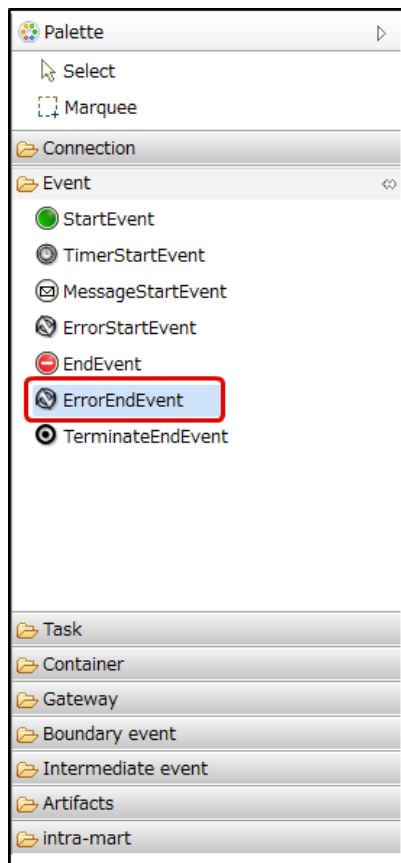
このエラーコードはエラー開始イベント、またはエラー境界イベントで受信する事ができます。

エラー終了イベントを配置した場合には必ずそのエラーコードを受信するエラー開始イベントを配置する必要があります。

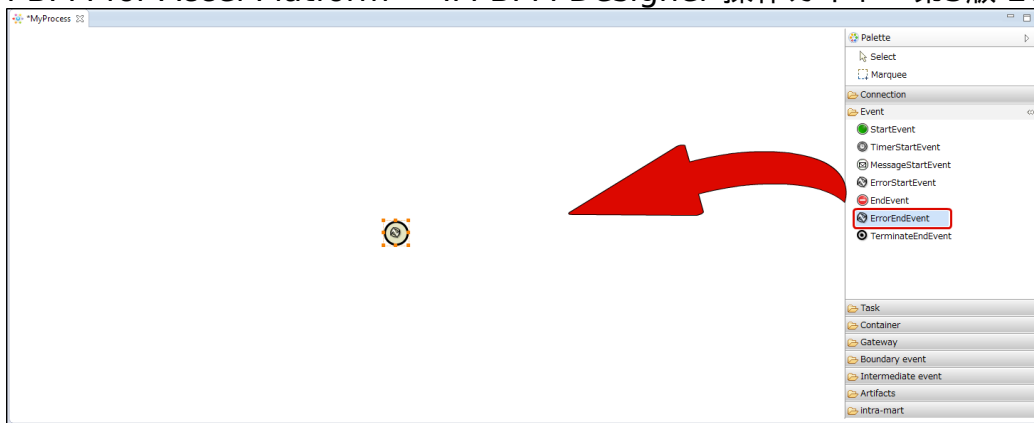
コールアクティビティを利用している場合には呼び出し元の親プロセス定義側で受信する事もできます。

エラー終了イベントの配置

1. IM-BPM Designerの画面右側PaletteのEventにあるErrorEndEventをクリックします。



2. エディタ上のプロセス定義を終了したい場所でクリックします。

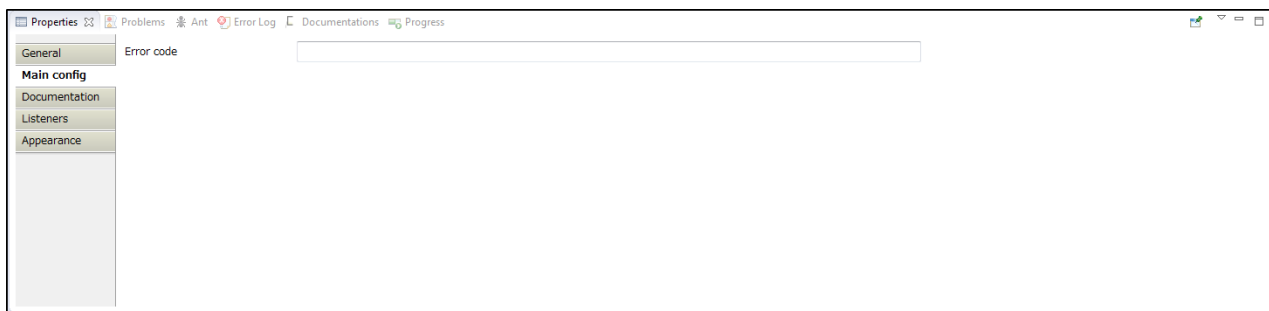


3. これでエラー終了イベントが定義されました。



プロパティの設定

エラー終了イベントのメインコンフィグの設定について解説します。



項目名	説明
Error code(エラーコード)	送信するエラーコードを文字列で定義します。 エラー終了イベントに到達すると、設定したエラーコードが設定されているエラー開始イベント,またはエラー境界イベントが開始します。 エラーコードを受信するエラー開始イベント、またはエラー境界イベントがない場合、Runtime側でエラーが発生します。

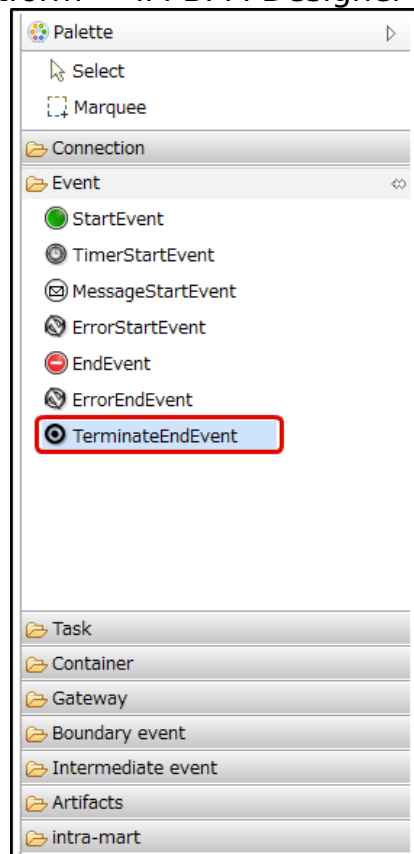
TerminateEndEvent(強制終了イベント)

特殊なイベントで、該当のプロセス定義フローを直ちに停止して終了する事を表します。

サブプロセス内に配置した場合も、サブプロセス内のプロセスフローが並行しているものも含めて完了とみなして、親プロセスの次のタスクやイベントへと続きます。

強制終了イベントの配置

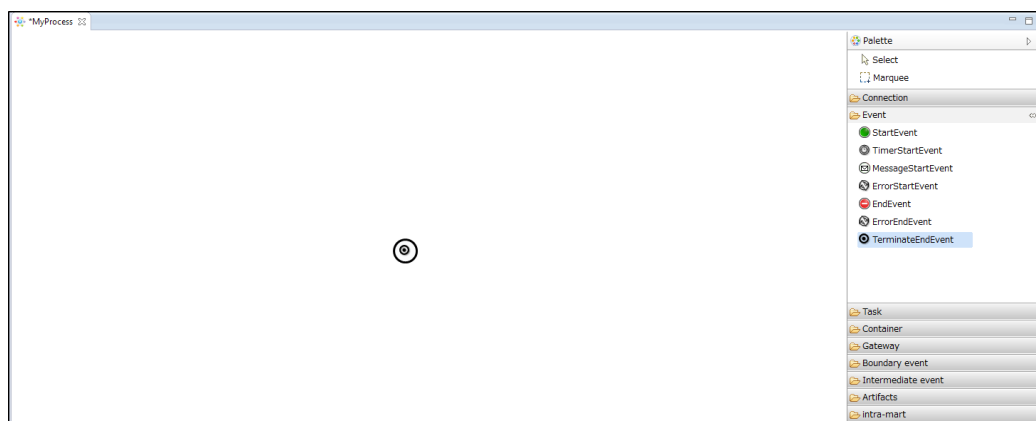
1. IM-BPM Designerの画面右側PaletteのEventにあるTerminateEndEventをクリックします。



2. エディタ上のプロセス定義を終了したい場所でクリックします。

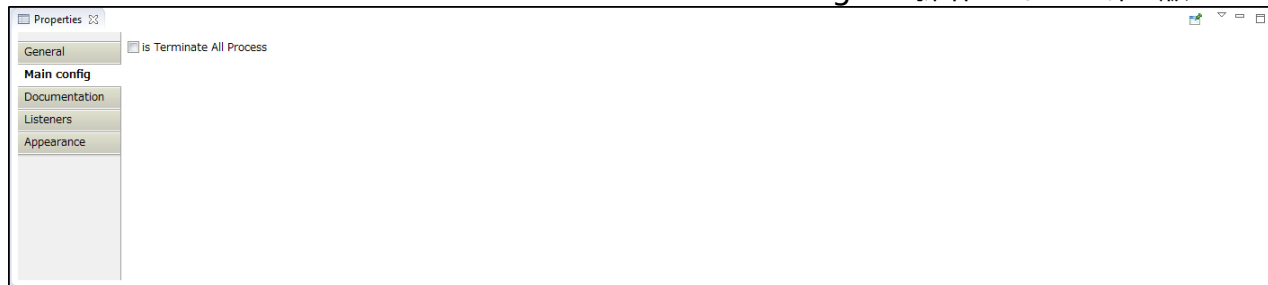


3. これで強制終了イベントが定義されました。



プロパティの設定

強制終了イベントのメインコンフィグの設定について解説します。



項目名	説明
Terminate all(親プロセスを強制終了する)	強制終了イベントに到達した場合、関連するプロセスのインスタンスも強制終了させるかどうかを選択します。 trueの場合、強制終了イベント到達時に関連したプロセスのインスタンスも自動的に終了させます。 falseの場合、強制終了イベントが所属しているプロセスインスタンスのみを終了します。

タスク

プロセス定義の中での最小単位の業務・アクションを表します。

その作業内容によって、8つに分類されます。

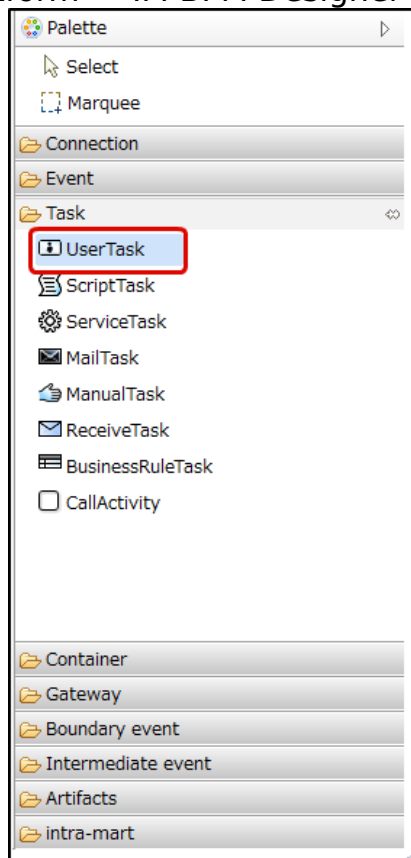
- [UserTask](#)
- [ScriptTask](#)
- [ServiceTask](#)
- [MailTask](#)
- [ManualTask](#)
- [ReceiveTask](#)
- [BusinessRuleTask](#)
- [CallActivity](#)

UserTask(ユーザタスク)

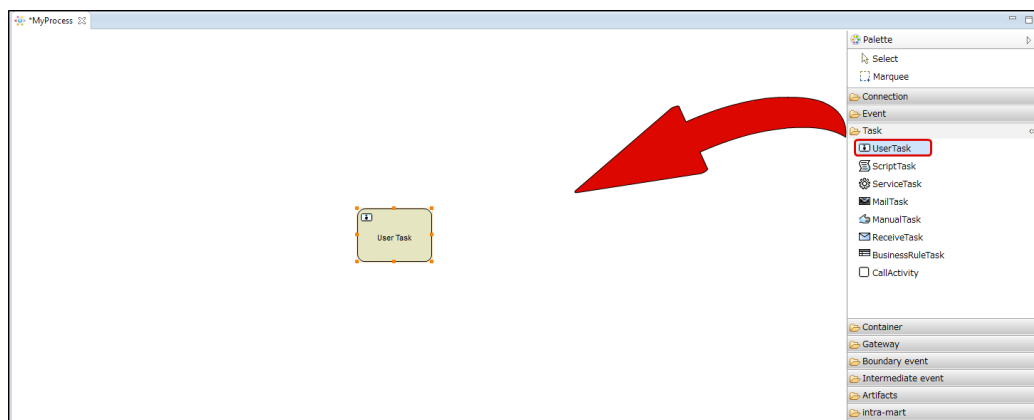
人によって実行されなければならないタスクを表します。
 このタスクは担当者となったユーザが完了させるまで、プロセスが進むことはありません。
 IM-BPM for Accel PlatformのRuntimeでは、「タスク一覧」画面にタスクが表示されます。

ユーザタスクの配置

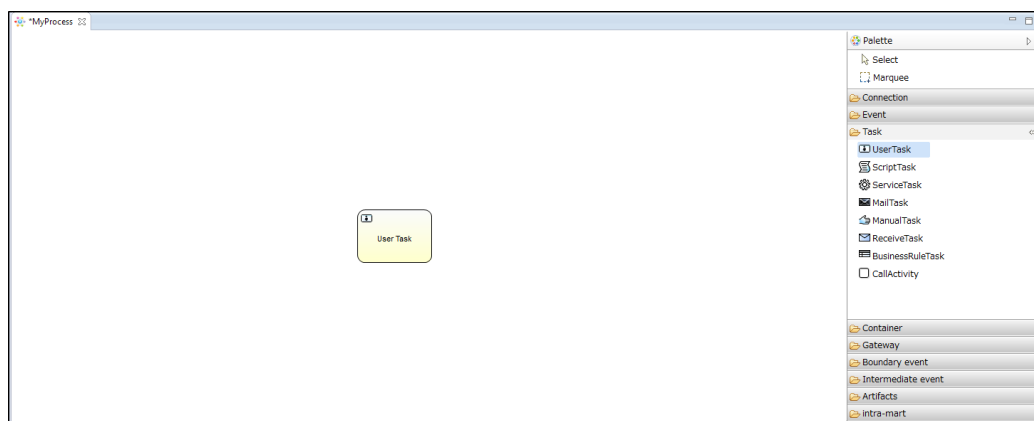
1. IM-BPM Designerの画面右側PaletteのTaskにあるUserTaskをクリックします。



2. エディタ上のユーザタスクを定義したい場所をクリックします。



3. これでユーザタスクが定義されました。



ユーザタスクのメインコンフィグの設定について解説します。

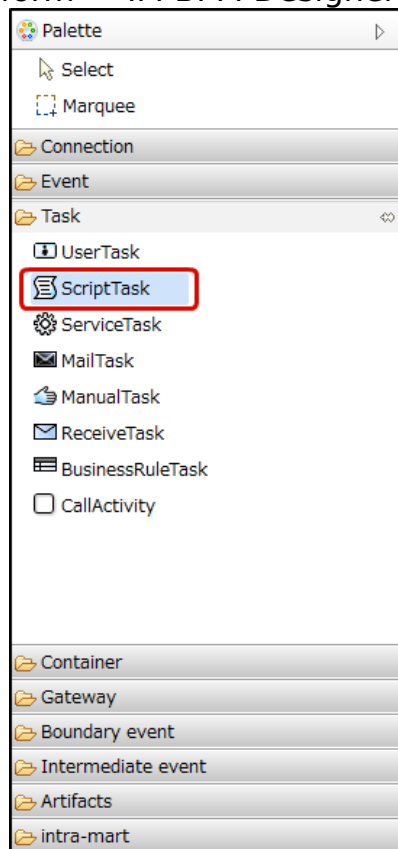
項目名	説明
Assignee(担当者)	タスクの担当者になるユーザを設定します。 ここで設定されたユーザはタスクに到達したときに自動的に担当者として割り振られます。
Candidate start users(処理対象ユーザ)	タスクを振り分けの権限を持つユーザを設定します。 権限をもつユーザを複数指定する場合はカンマで行います。
Candidate start groups(処理対象グループ)	定義したプロセスインスタンスを開始できる対象グループです。 この項目に記述するグループはロールIDとなり、対象となるユーザは指定したロールを保持するユーザです。 複数グループの指定はカンマで行います。
Form key(フォームキー)	利用するアプリケーションと画面IDをEL式で設定できます。 例1: formaのアプリケーション「application_id」を呼び出す場合 → forma:application_id 例2: スクラッチ開発した画面「http://screen/url」を呼び出す場合 → forward:http://screen/url
Due date(期限)	ユーザタスクの期限を設定します。
Priority(優先度)	タスクの優先度を設定します。 ソートや検索条件としても使用できます。
Category(カテゴリ)	ユーザタスクのカテゴリを指定します。
Skip expression(スキップ条件)	タスクを自動的に終了する条件のEL式を記述します。 記入した条件がtrueと評価された場合、このタスクは完了した状態となり次のタスク、またはイベントへと自動的に遷移します。 falseの場合、従来どおり担当者となったユーザがタスクを完了する必要があります。

ScriptTask(スクリプトタスク)

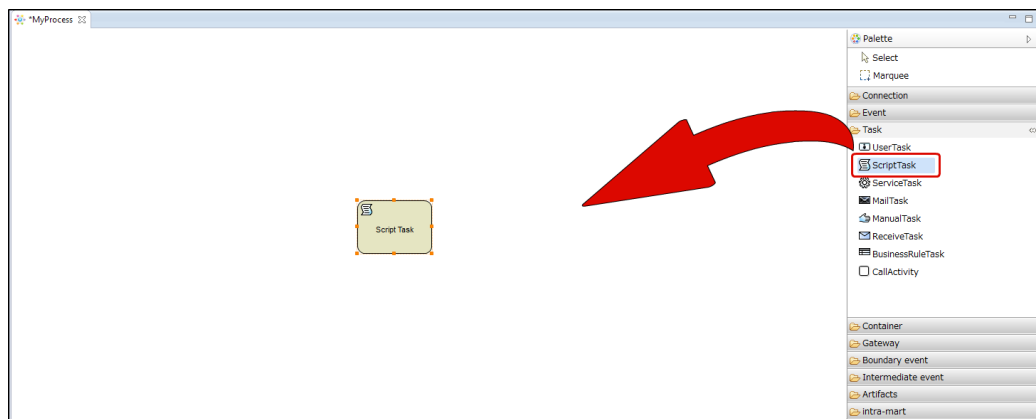
- タスクに到達すると自動的に設定されたスクリプトを実行する事を表します。
- intra-martのスクリプト開発モデルを利用した実装が可能です。

スクリプトタスクの配置

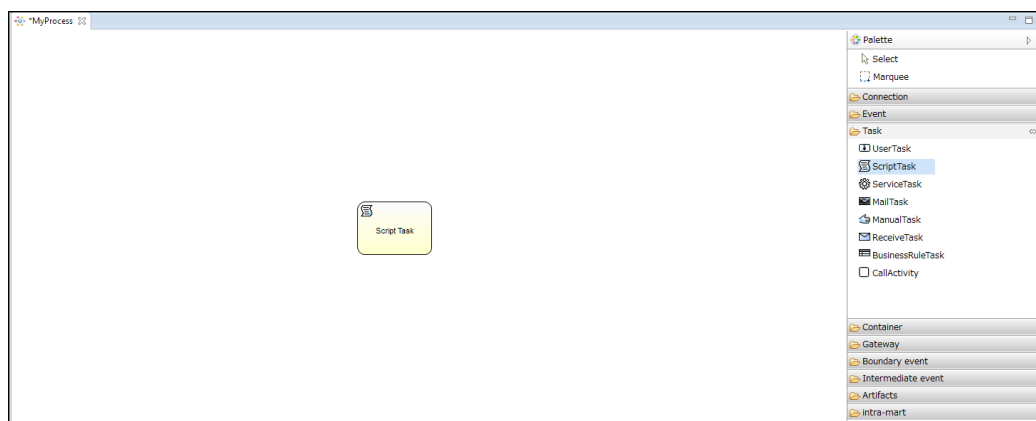
1. IM-BPM Designerの画面右側PaletteのTaskにあるScriptTaskをクリックします。



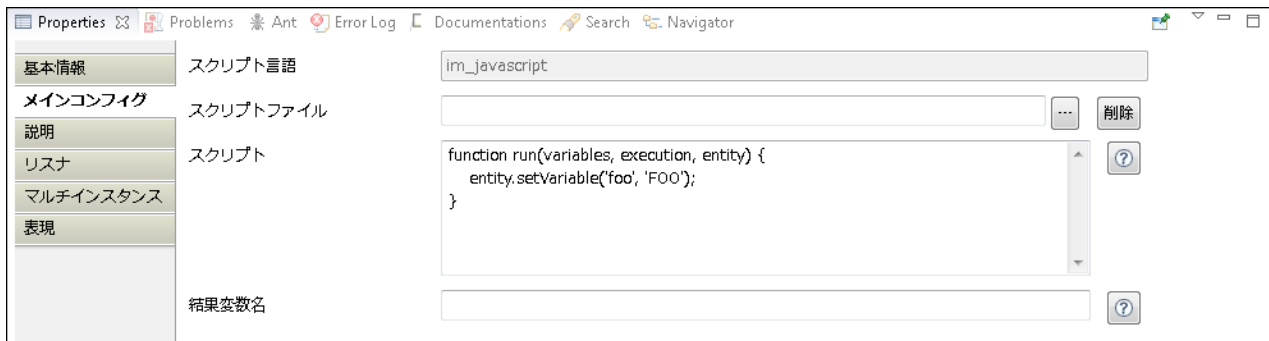
2. エディタ上のスクリプトタスクを定義したい場所をクリックします。



3. これでスクリプトタスクが定義されました。



スクリプトタスクのメインコンフィグの設定について解説します。



項目名	説明
Script language(スクリプト言語)	現在変更はできません。
Script file(スクリプトファイル)	実行するスクリプトファイルを指定します。 「...」 ボタンを押下すると表示されるダイアログでjsファイルを選択します。 ダイアログには同一プロジェクトの「src/main/resources」ソース・フォルダー配下の「scripts」パッケージ内にあるjsファイルが表示されます。 jsファイルのパスを直接入力することはできません。 スクリプトで記述されたプログラムを実行する場合は、スクリプトファイルの値を削除する必要があります。 スクリプトファイルの値を削除する場合は、「削除」ボタンを押下します。
Script(スクリプト)	実行するスクリプトを記述します。 実行はサーバ側で行われます。 スクリプトの編集を行う場合は、スクリプトファイルの値を削除する必要があります。 スクリプトファイルの値が入力されている場合は、スクリプトで記述したプログラムは実行されません。
Result variable(結果変数名)	実行結果を保存する変数を設定します。 例:変数「result」に保存してある値で設定したい場合 → \${result}

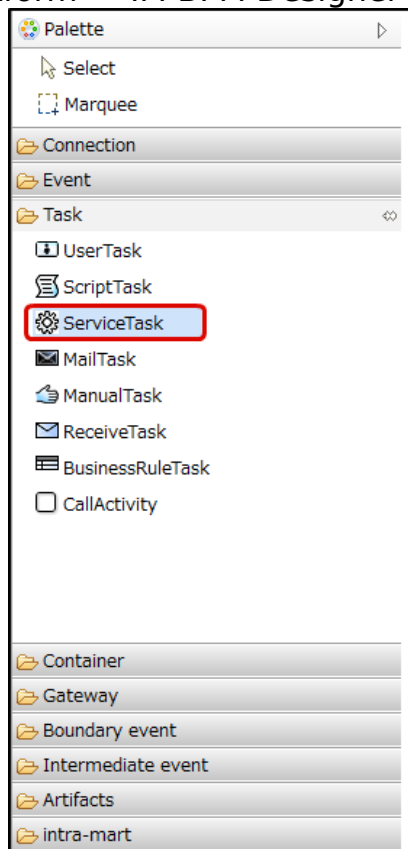
ServiceTask(サービスタスク)

任意の処理を呼び出す際に利用します。

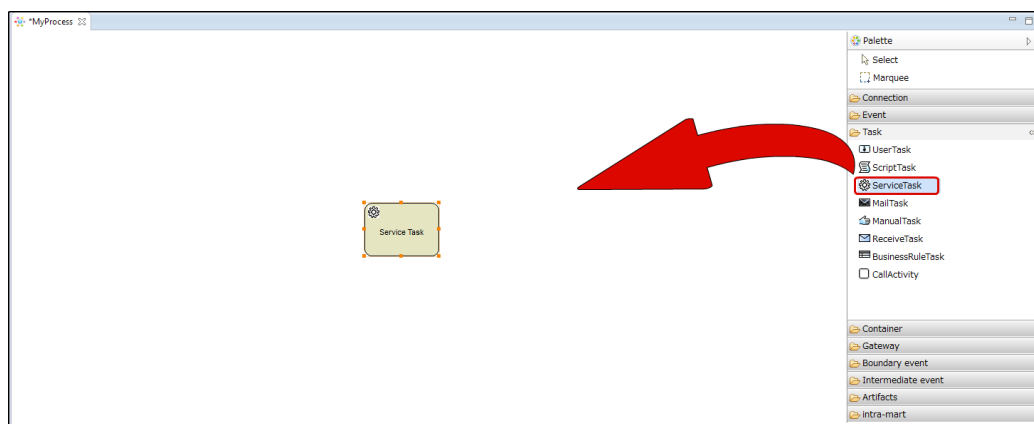
このタスクは処理の完了と同時にタスクも完了します。

ServiceTaskの配置

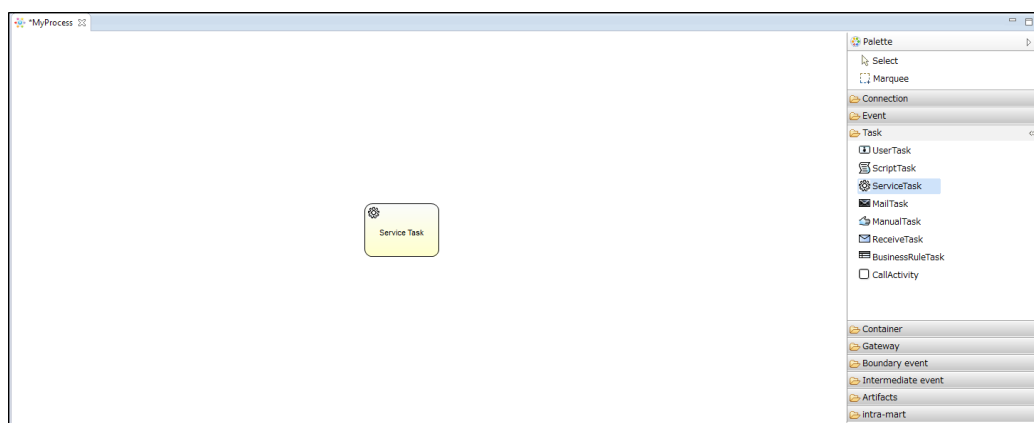
1. IM-BPM Designerの画面右側PaletteのTaskにあるServiceTaskをクリックします。



2. エディタ上のサービスタスクを定義したい場所をクリックします。



3. これでサービスタスクが定義されました。



サービスタスクのメインコンフィグの設定について解説します。

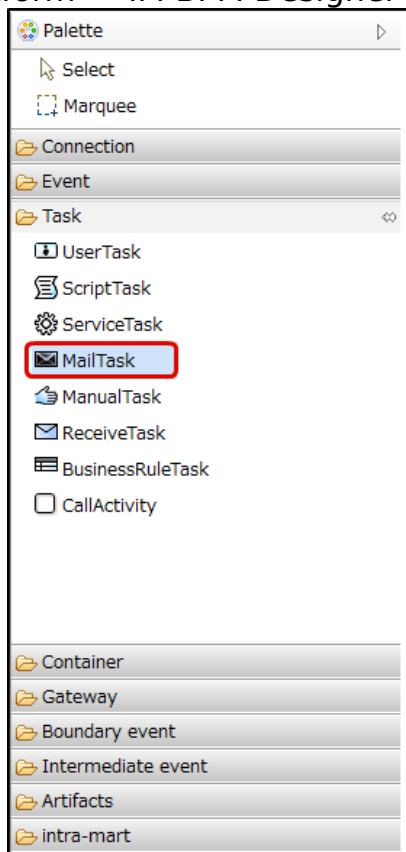
項目名	説明
Task type(タスクタイプ)	呼び出すタスクのタイプを選択します。 Java class・Expression・Delegate expressionの3種類があります。
Class name(クラス名)	呼び出すJava class名を選択します。 下記のInterfaceのいずれかを実装する必要があります。 - org.activiti.engine.delegate.JavaDelegate - jp.co.intra_mart.activiti.engine.impl.pvm.delegate.ActivityBehavior タスクタイプでjava classを選択した場合に入力します。
Expression(式)	呼び出すメソッドをEL式で指定します。 タスクタイプでExpressionを選択した場合に入力します。
Delegate Expression(Delegate Expression)	呼び出すクラスをEL式で指定します。 タスクタイプでDelegate expressionを選択した場合に入力します。
Skip expression(スキップ条件)	タスクを自動的に終了する条件のEL式を記述します。 記入した条件がtrueと評価された場合、このタスクは設定されたサービスを実行せずに次のタスク、またはイベントへと遷移します。 falseの場合、従来どおり設定されたサービスを実行してタスクを完了します。
Result variable(結果変数名)	実行結果を保存する変数を設定します。
Fields(フィールド)	java classのフィールドに値をセットする設定です。 「New」ボタンで新しいフィールド変数の登録が行えます。 「Edit」ボタンで、既にあるフィールド変数の編集が行えます。 「Remove」ボタンで既にあるフィールド変数の削除が行えます。 「Up」ボタンと「Down」ボタンでフィールド変数の並び替えが行えます。

MailTask(メールタスク)

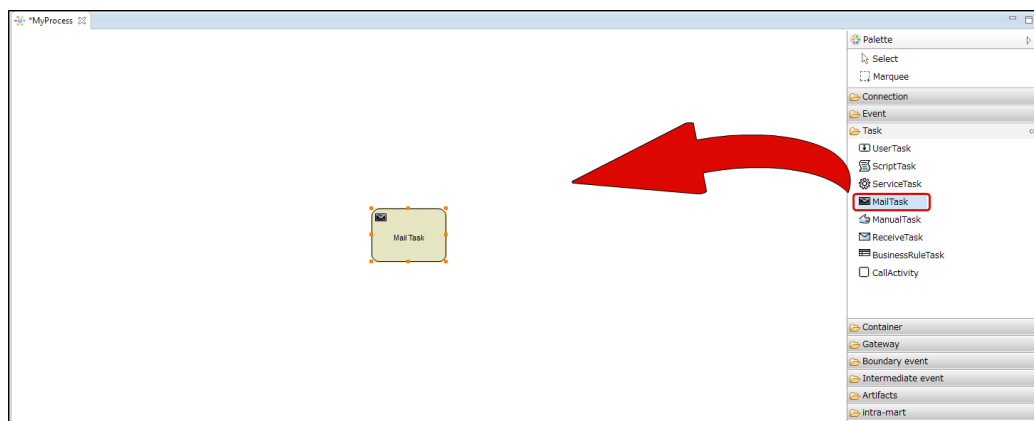
このタスクに到達すると設定した内容でメールの送信を行います。
このタスクはサービスタスク同様、自動的にタスクを完了します。

メールタスクの配置

1. IM-BPM Designerの画面右側PaletteのTaskにあるMailTaskをクリックします。



2. エディタ上のメールタスクを定義したい場所をクリックします。



3. これでメールタスクが定義されました。



メールタスクのメインコンフィグの設定について解説します。

項目名	説明
To	送信先のメールアドレスを設定します。 必須の項目です。 複数の送信先はカンマ区切りで設定します。
Form	送信者のメールアドレスを設定します。 設定されていない場合、デフォルト設定のメールアドレスが使用されます。
Subject(件名)	メールの件名を設定します。
Cc	Ccで送信するメールアドレスを設定します。 複数の送信先はカンマ区切りで設定します。
Bcc	Bccで送信するメールアドレスを設定します。 複数の送信先はカンマ区切りで設定します。
Text(本文)	メールの本文を記述します。
Ignore exception(エラーを無視する)	このタスクで発生したエラーを送信するかの有無を設定します。
Exception variable name(エラー変数名)	エラーが発生した際にエラーメッセージを保存する変数を設定します。 例:変数「errmsg」に保存してある値で設定したい場合 → \${errmsg}

ManualTask(マニュアルタスク)

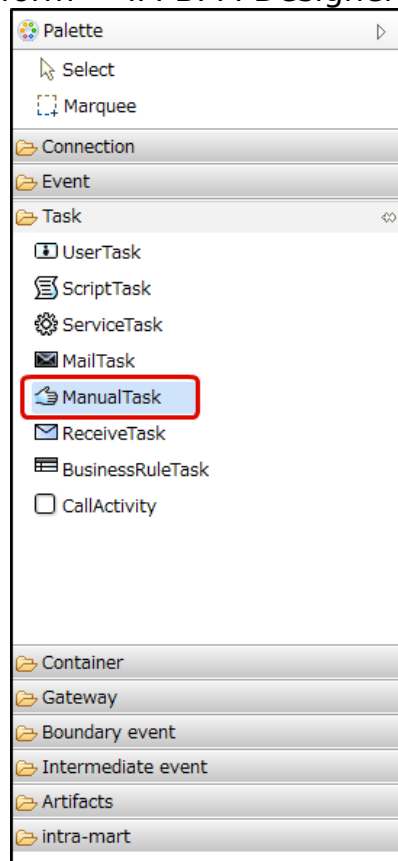
人によって一部の処理が実行されなければならない事を表します。

ユーザタスクとの違いは担当者の振り分けの必要がなく、到達後すぐにタスクを完了します。

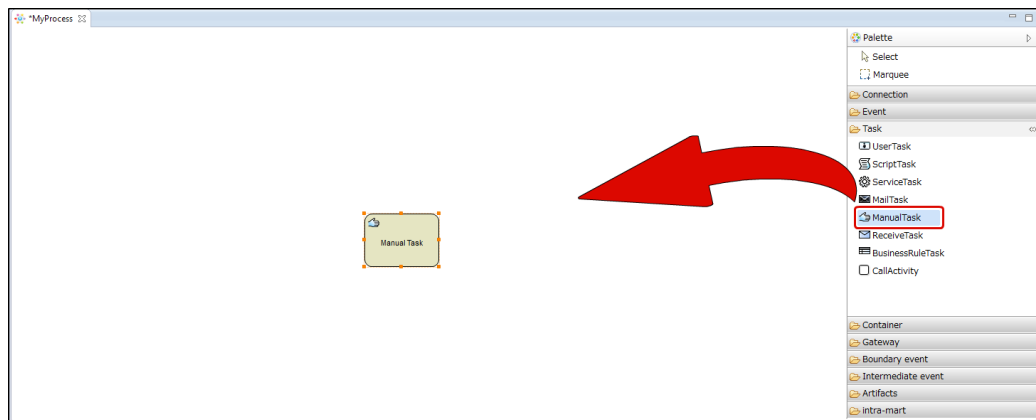
マニュアルタスクは、業務フローとして表現が必要ですが、実際にシステムとして実行する必要のないタスクとして利用されます。

マニュアルタスクの配置

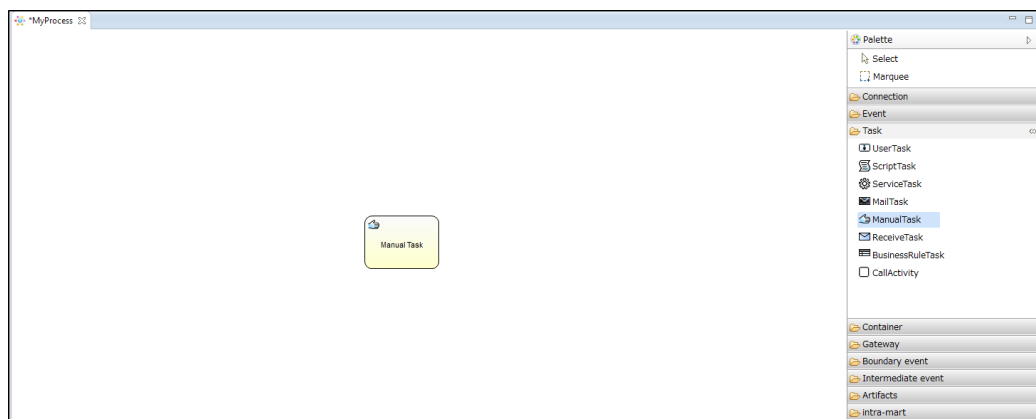
1. IM-BPM Designerの画面右側PaletteのTaskにあるManualTaskをクリックします。



2. エディタ上のマニュアルタスクを定義したい場所をクリックします。



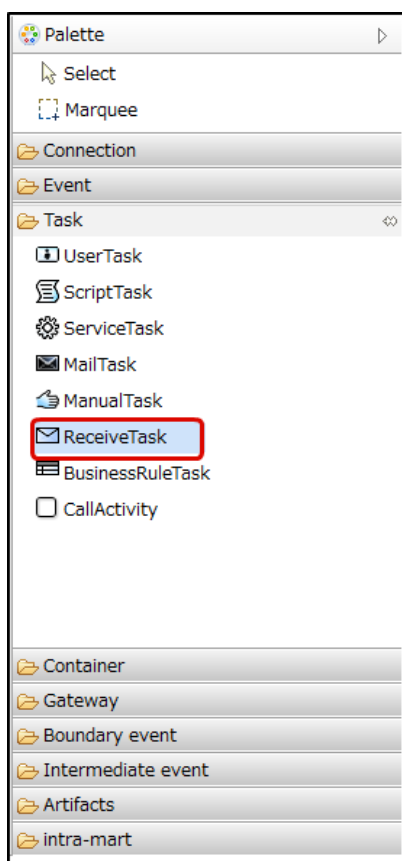
3. これでマニュアルタスクが定義されました。



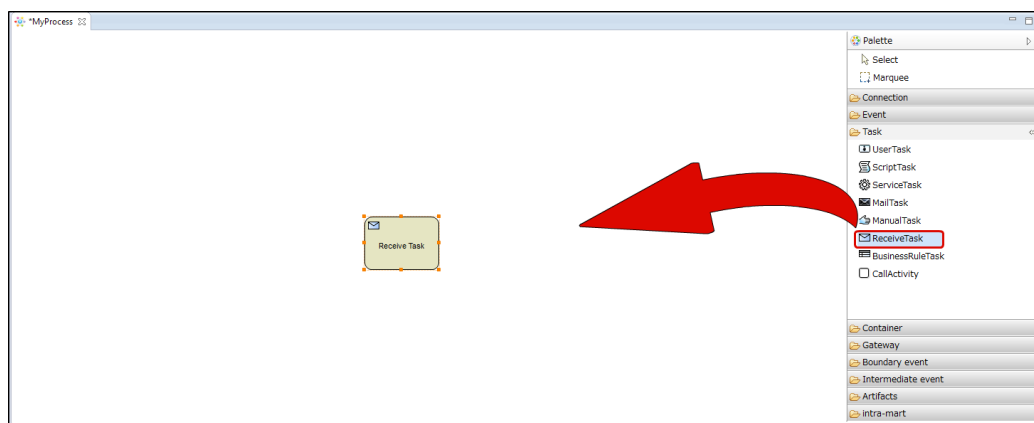
メッセージの受信を待つ実行されるタスクを表します。

受信タスクの配置

1. IM-BPM Designerの画面右側PaletteのTaskにあるReciveTaskをクリックします。



2. エディタ上の受信タスクを定義したい場所をクリックします。



3. これで受信タスクが定義されました。



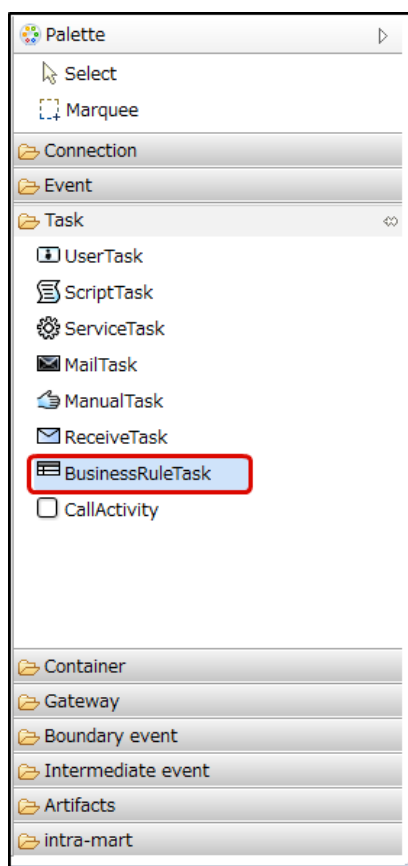
BusinessRuleTask(ビジネスルールタスク)

ルールエンジンを実行します。

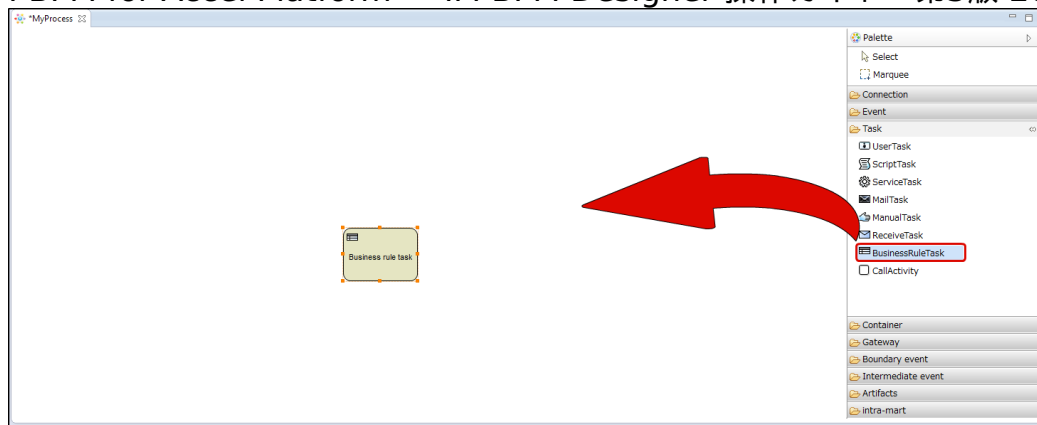
IM-BPM for Accel Platformでは OpenRulesを利用することができます。

ビジネスルールタスクの配置

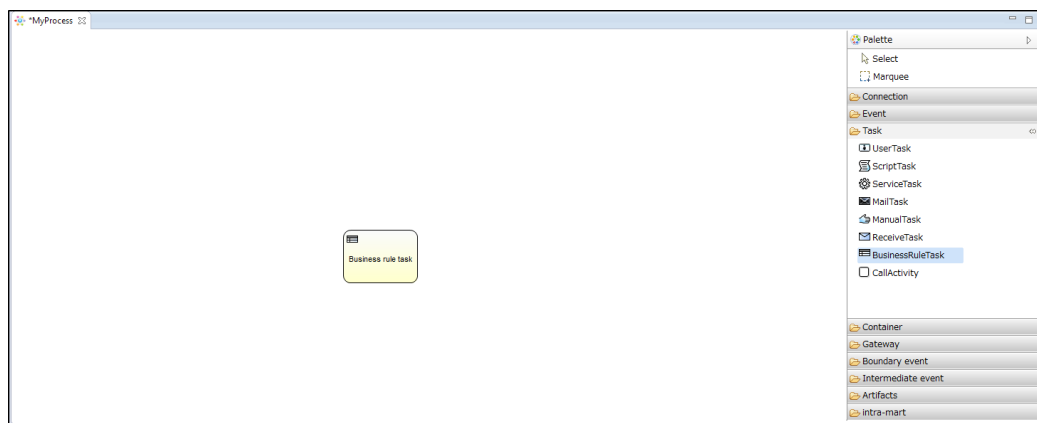
1. IM-BPM Designerの画面右側PaletteのTaskにあるBusinessRuleTaskをクリックします。



2. エディタ上のビジネスルールタスクを定義したい場所をクリックします。



3. これでビジネスルールタスクが定義されました。



プロパティの設定

ビジネスルールタスクのメインコンフィグの設定について解説します。

項目名	説明
Rule File(ルールファイル)	OpenRulesのルール定義ファイルを設定します。
Decision(デシジョン)	ルール定義ファイル内の処理順を指定します。
Input Concept(入力コンセプト)	ルールファイル内の入力コンセプトを指定します。
Output Concept(出力コンセプト)	ルールファイル内の出力コンセプトを指定します。
Result variable(結果変数名)	実行結果を保存する変数を設定します。 例:変数「result」に保存してある値で設定したい場合 → \${result}

CallActivity(コールアクティビティ)

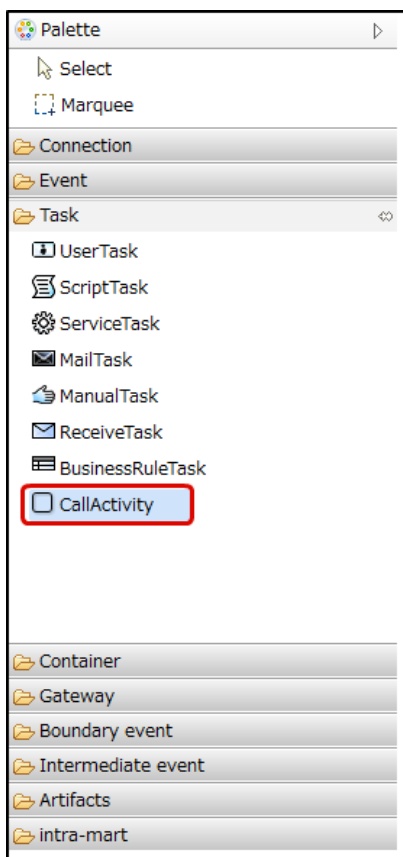
外部に定義されたプロセス定義を参照実行する事を表します。

サブプロセスとの違いは、異なるプロセス定義ファイルに定義されたプロセスフローを呼び出すことができます。

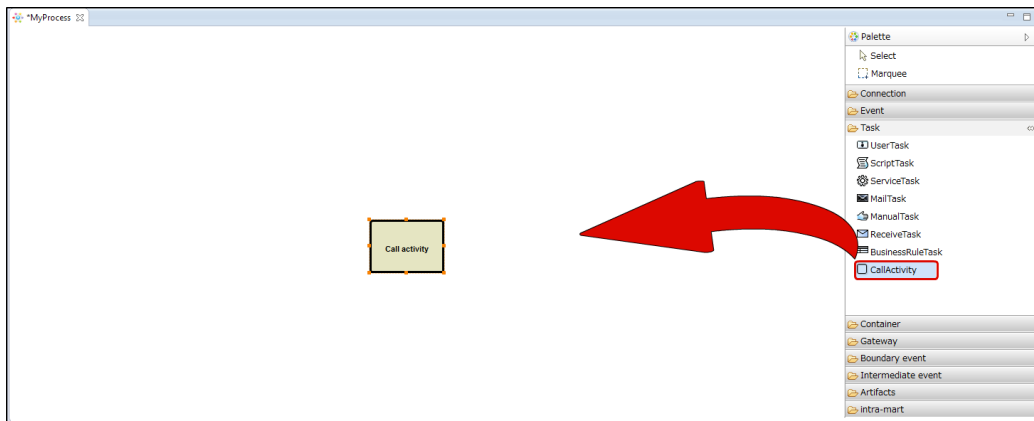
そのため、複数のプロセスの共通する処理がある場合、共通処理のフローを記述したプロセス定義ファイルを作成し、他のプロセスからコールアクティビティを使って呼び出すことができます。

コールアクティビティの配置

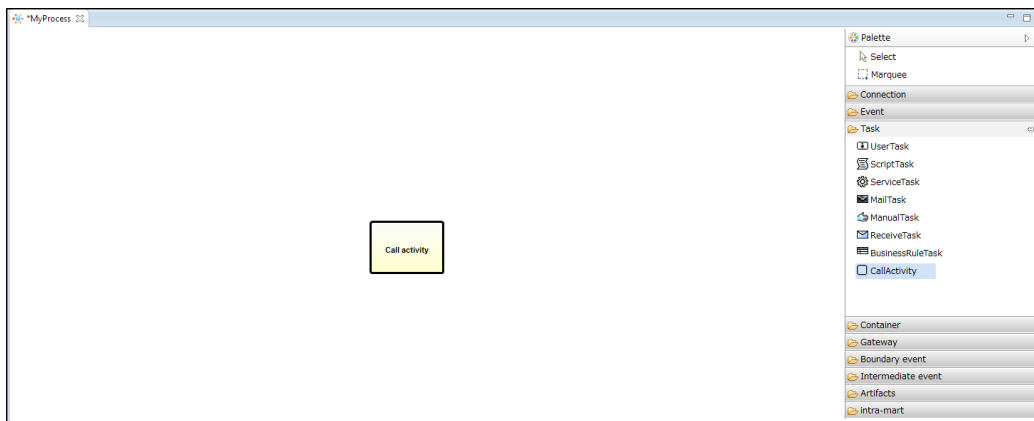
1. IM-BPM Designerの画面右側PaletteのTaskにあるCallActivityをクリックします。



2. エディタ上のコールアクティビティを定義したい場所をクリックします。

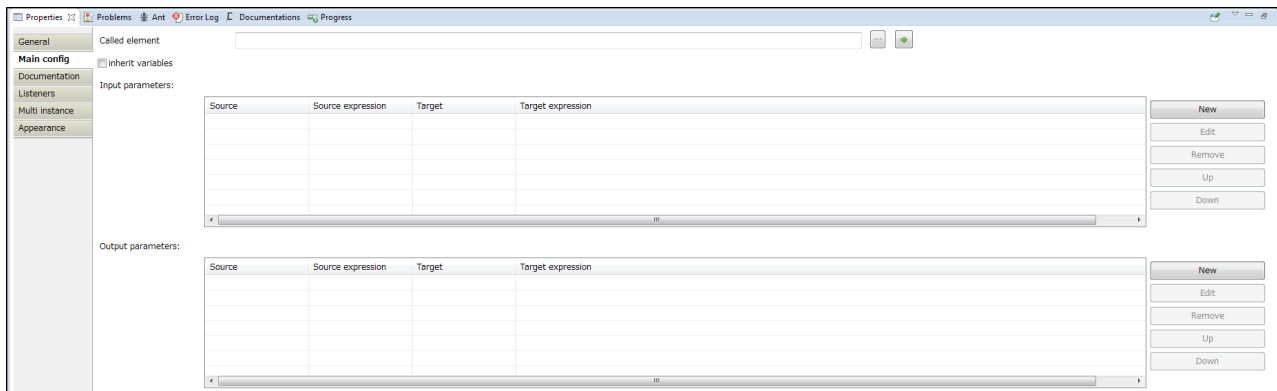


3. これでコールアクティビティが定義されました。



プロパティの設定

コールアクティビティのメインコンフィグの設定について解説します。



項目名	説明
Called element(呼び出し対象)	呼び出す対象のプロセスを指定します。
inherit variables(参照可能な変数を継承する)	呼び出し元プロセスの変数を呼び出し先のプロセスにすべて引き継ぐかを選択します。
Input parameters(入力パラメータ)	呼び出し先プロセスに渡すパラメータを設定します。
Output parameters(出力パラメータ)	呼び出し先プロセスから取得するパラメータを設定します。

コンテナ

本項ではIM-BPM DesignerのPaletteにある「コンテナ」で設定可能なコンポーネントの説明と定義方法について説明します。

「コンテナ」に配置されているコンポーネントはフロー中のタスクに対する特定のグループの割り当ての表現や、特定の条件が発生した場合に実施されるサブフローを表現するために利用されます。

EventSubProcess(イベントサブプロセス)

イベントサブプロセスはイベントを受信した場合のみ実行されるでプロセス定義であり、破線の長方形によって表現されます。

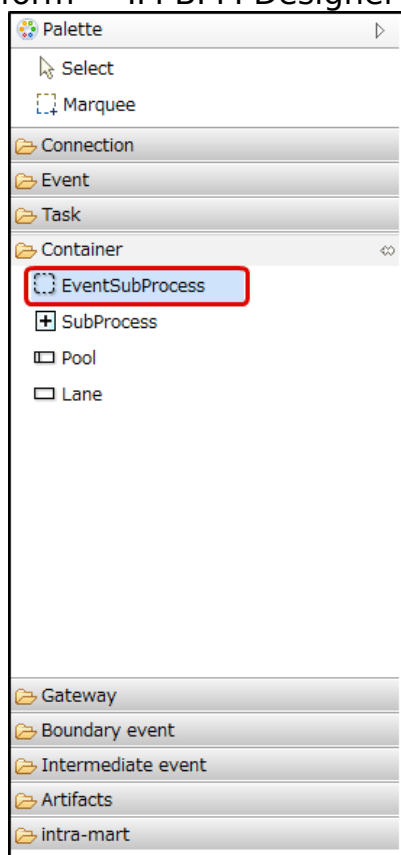
イベントサブプロセスの内部には、イベントを受信した際に実行されるプロセスフローを定義します。

イベントサブプロセスには、いくつかの制限があります。

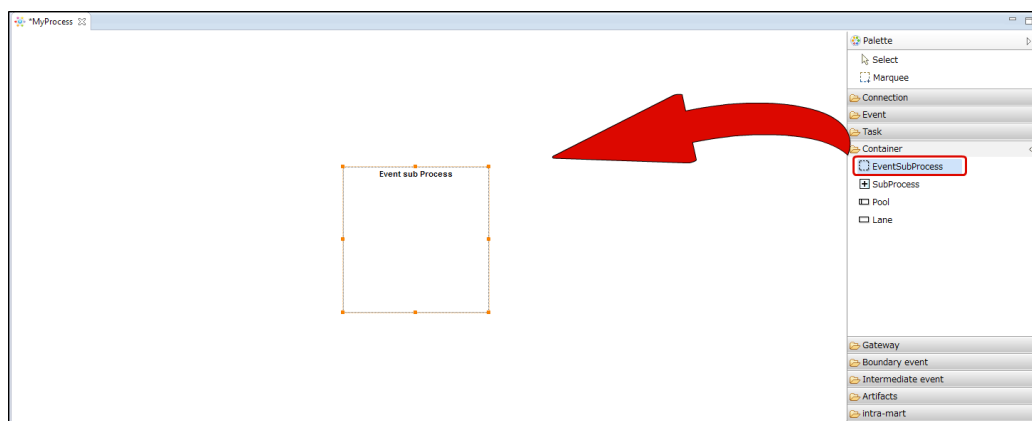
1. イベントを受信して実行されるため、開始イベントは定義できません。
2. 開始イベントとして、エラー開始イベントまたはメッセージ開始イベントを定義しなければなりません。

イベントサブプロセスの配置

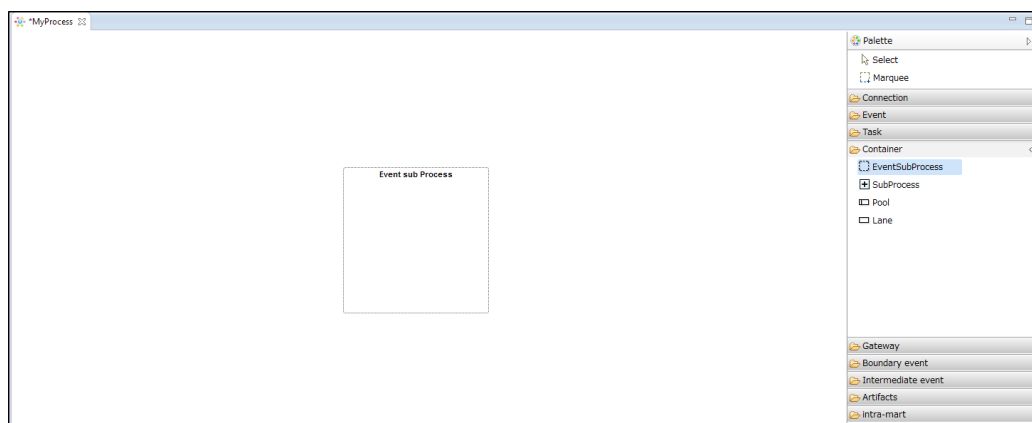
1. IM-BPM Designerの画面右側PaletteのContainerにあるEventSubProcessをクリックします。



2. エディタ上のイベントサブプロセスを定義したい範囲をドラッグします。



3. これでイベントサブプロセスが定義されました。



サブプロセスは、特定のプロセス定義内で利用されるプロセス定義であり、実線の長方形によって表現されます。

サブプロセスの主な用途としてサブプロセスに対する境界イベントの配置が挙げられます。

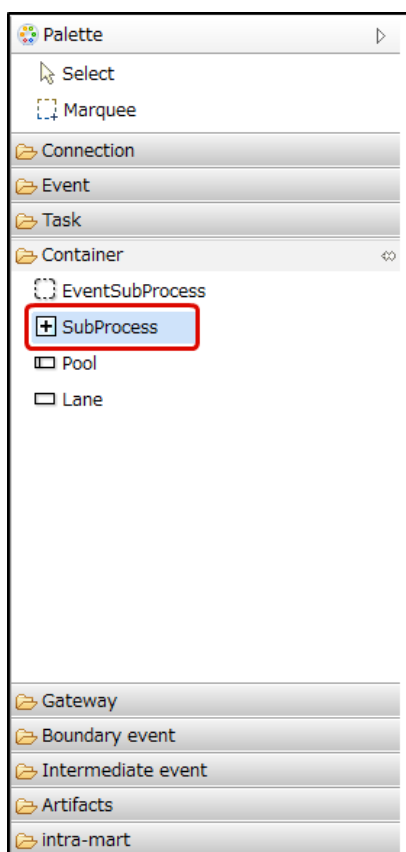
また、サブプロセスに対してマルチインスタンスの指定を行う事も可能です。

サブプロセスには、いくつかの制約があります。

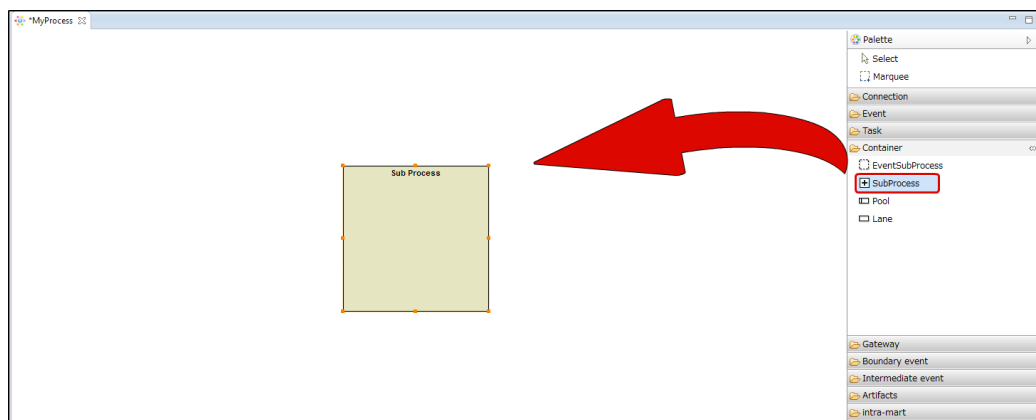
1. 開始イベントは通常の開始イベントのみを設定できます。
タイマー開始イベント、メッセージ開始イベント、エラー開始イベントは利用できません。
2. 複数の開始イベントを持つ事はできません。
3. サブプロセスの境界を越えてシーケンスフローを定義する事はできません。

サブプロセスの配置

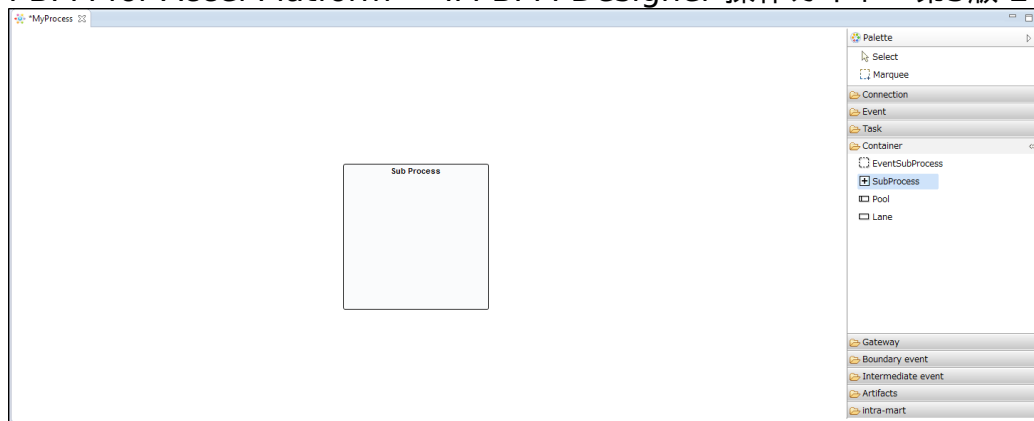
1. IM-BPM Designerの画面右側PaletteのContainerにあるSubProcessをクリックします。



2. エディタ上のサブプロセスを定義したい場所をクリックします。



3. これでサブプロセスイベントが定義されました。



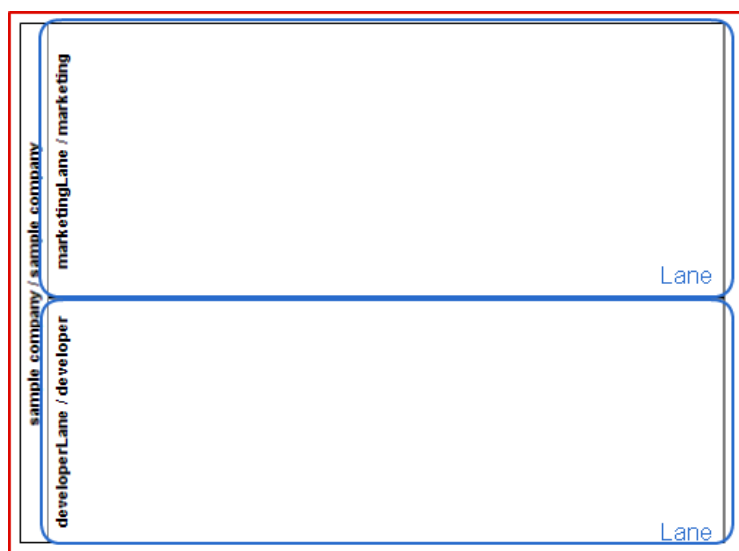
プール・レーン

プールはプロセス定義における関係者の境界線を表現したフローエレメントです。

レーンはプールの中でさらに役割を区分したい場合に使用します。

プロセス定義を設計する時に、プロセスフロー中に複数の関係者が登場するときにプールとレーンを利用して表現します。

たとえば1つのプロセス定義に対して、同じ会社の2つの部門が登場するとき、設計者は下記の図のように「会社」というプールを作成し、レーンを2つ配置して部門を表現することになります。



Pool

プロセス定義において単一の関係者しか登場しない場合は、プールやレーンは配置せずにプロセス定義の設計を行うことができます。

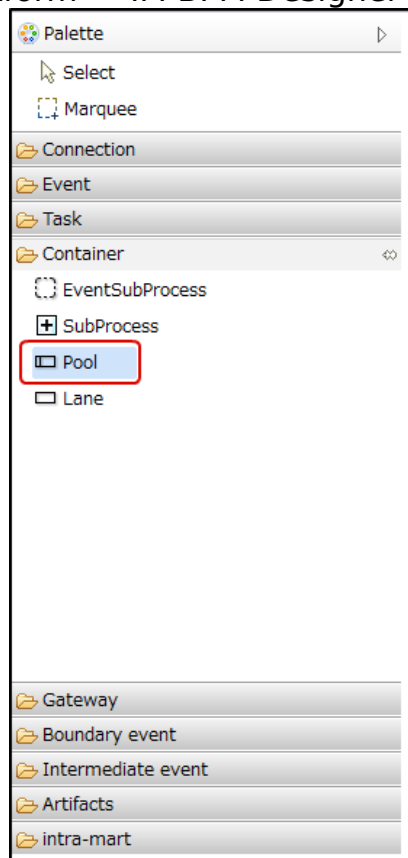
Pool(プール)

プールはプロセス定義における関係者を表すための境界線です。

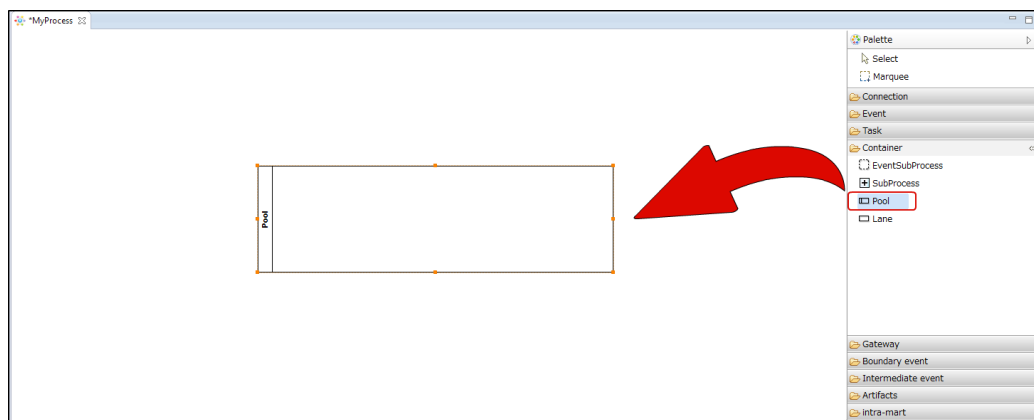
この関係者の範囲は、会社、組織、役職、人等を設定することができます。

プールの配置

1. IM-BPM Designerの画面右側PaletteのContainerにあるPoolをクリックします。



2. エディタ上でドラッグをしてプールの範囲を指定します。



3. これでプロセス定義中へのプールの配置が完了しました。



プールを配置すると自動的に1つのレーンがプール内に配置されます。

プロパティの設定

プールのプロパティの設定について解説します。

項目名	説明
Id	プールを一意に特定するために設定します。 英数字を推奨します。
Name(名前)	プールの名前を設定します。
Namespace	名前空間を設定します。 通常はURL形式の名前空間を設定します。
Candidate start users(処理対象ユーザ)	このタスクを担当することができる処理対象ユーザ（複数） ここで設定されているユーザはタスクを担当できる。 複数ユーザの指定はカンマで行います。
Candidate start groups(処理対象グループ)	このタスクを担当することができる処理対象グループ(複数) ここで設定されたグループに所属するユーザはタスクを担当できる。 複数グループの指定はカンマで行います。
Documentation	このプロセス定義についての詳細・備考を記載する項目です。

Lane(レーン)

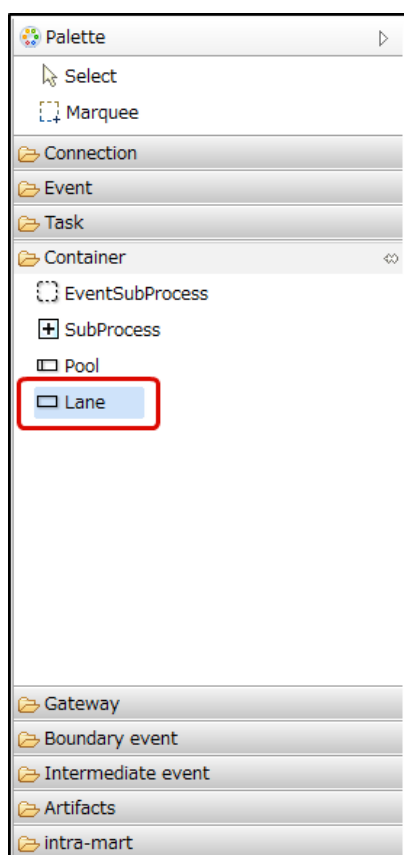
レーンはプール内で任意の区分を定義します。

レーンを配置する場合は、事前にプールが設定されていることが前提となります。

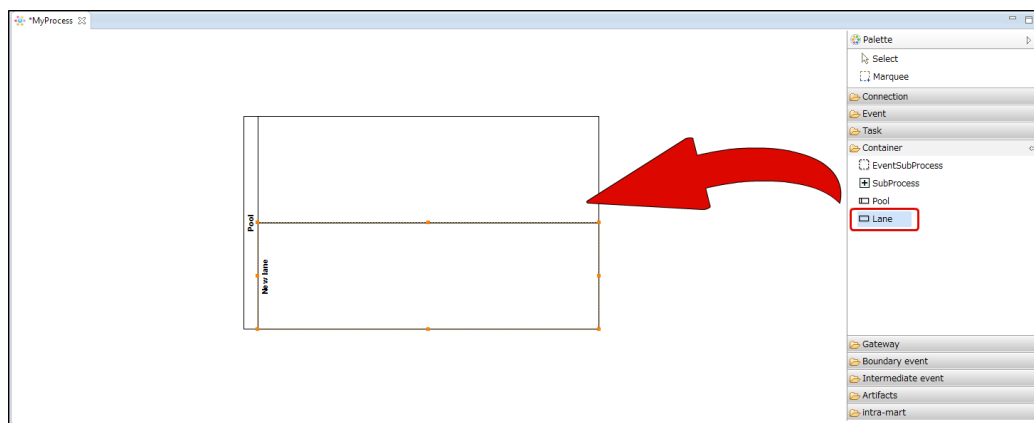
プロセス定義中にプールが配置されていない状態でレーンを配置することはできません。

レーン

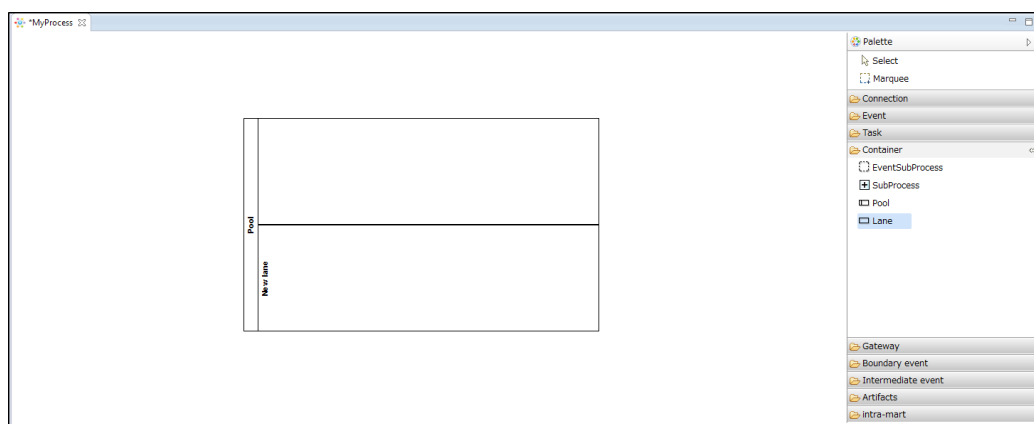
1. IM-BPM Designerの画面右側PaletteのContainerにあるLaneをクリックします。



2. エディタに配置したプールをクリックします。



3. これでプールへのレーンの配置が完了しました。



ゲートウェイ

ゲートウェイはプロセスフローの分岐・結合に利用します。

ゲートウェイを使用することで複数のタスクを同時並行で処理することや、特定の条件を評価することによってプロセスフローを変更することを可能にします。

ゲートウェイは分岐・結合の条件によって、4つに分類されます。

- [ParallelGateway](#)
- [ExclusiveGateway](#)
- [InclusiveGateway](#)
- [EventGateway](#)

ParallelGateway(パラレルゲートウェイ)

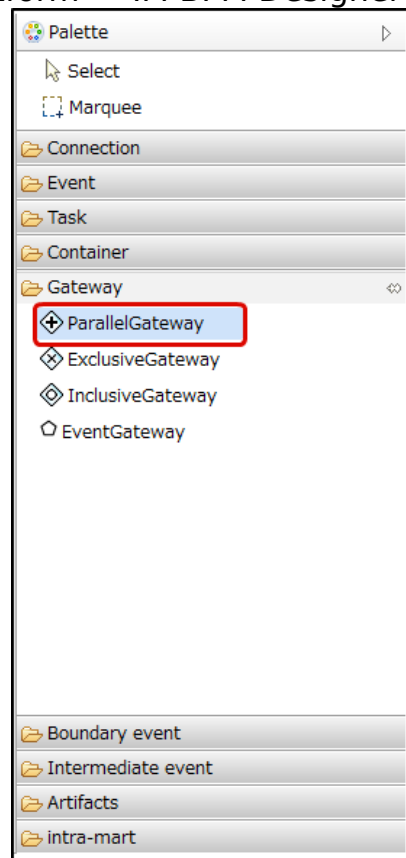
パラレルゲートウェイは、無条件で同時並行に分岐し、シーケンスフローが次の処理に進む事を表します。

結合の際には、全てのシーケンスフローの到着を待ってプロセス定義の実行が継続されます。

- パラレルゲートウェイで分岐を行うためには、パラレルゲートウェイから複数のアクティビティにシーケンスフローを接続する必要があります。
- パラレルゲートウェイで結合を行うためには、複数のアクティビティからパラレルゲートウェイにシーケンスフローを接続する必要があります。

パラレルゲートウェイの配置

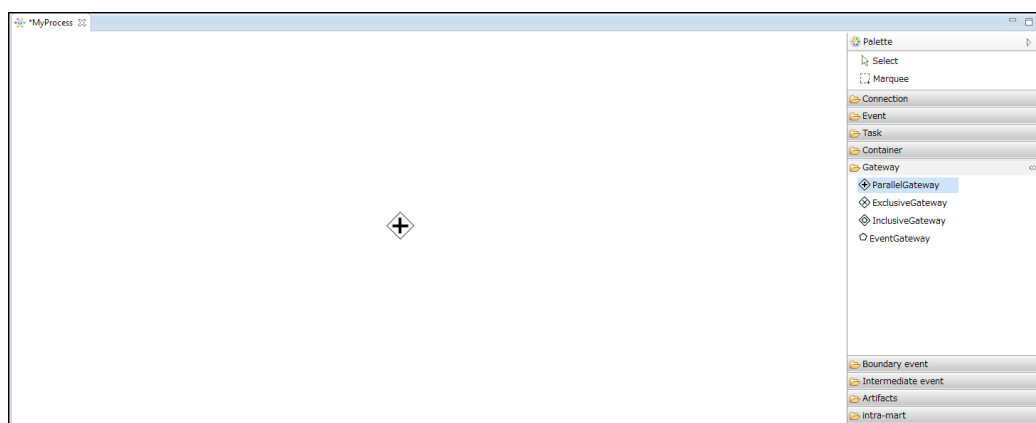
1. IM-BPM Designerの画面右側PaletteのGatewayにあるParallelGatewayをクリックします。



2. エディタ上のパラレルゲートウェイを定義したい場所をクリックします。

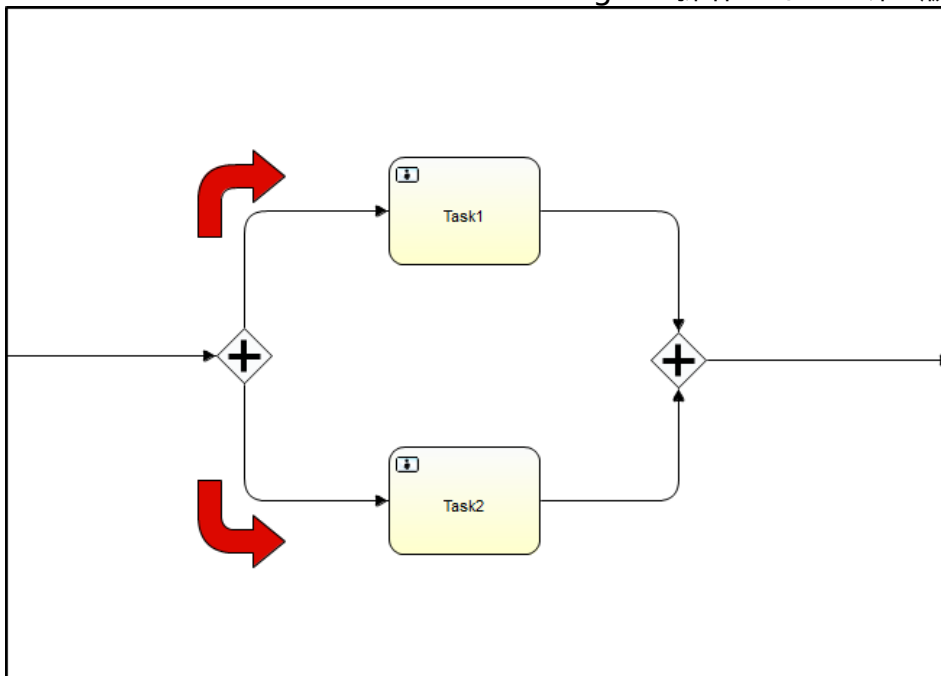


3. これでパラレルゲートウェイが定義されました。

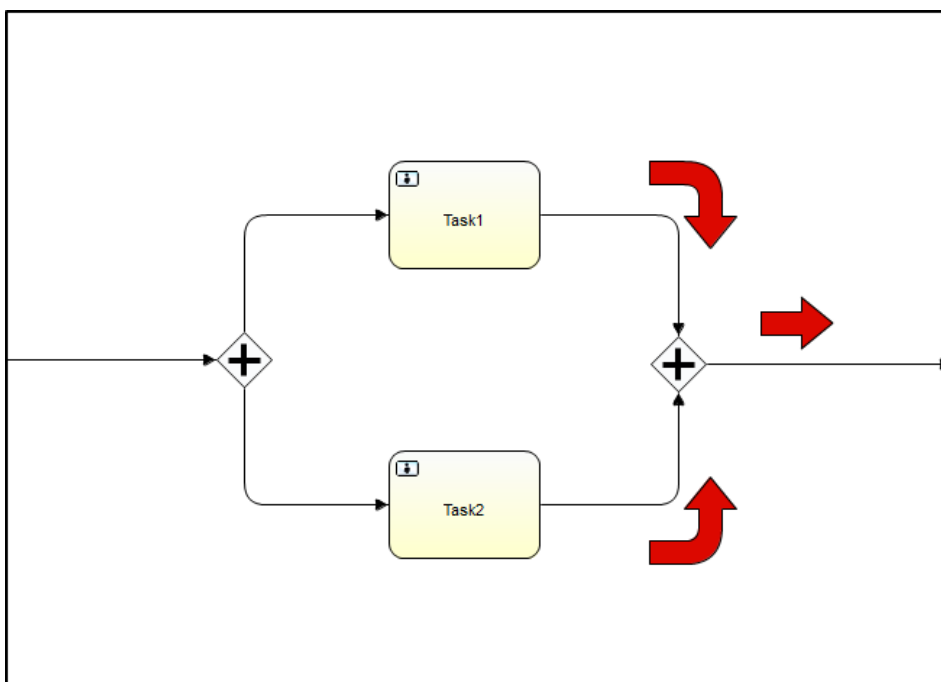


パラレルゲートウェイによる分岐と結合

- 次のようにパラレルゲートウェイで分岐を行った場合、Task1、Task2両方にシーケンスフローが進みます。



- パラレルゲートウェイで結合を行った場合、Task1、Task2両方のシーケンスフローの到着を待って次のシーケンスフローに進みます。



ExclusiveGateway(排他ゲートウェイ)

条件式を評価する事で分岐先を決定することができます。

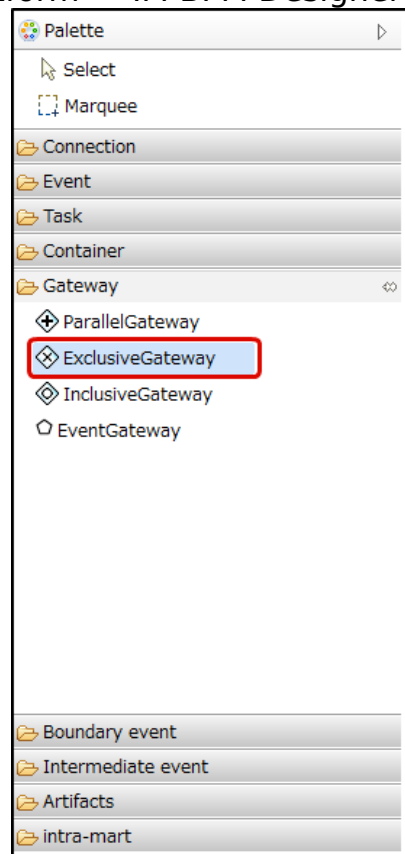
条件式を評価するためには接続されている「[Sequence Flow](#)」に条件式の設定が必要です。

最初に評価が真となる1つのシーケンスフローだけが継続し次の処理に進む事を表します。

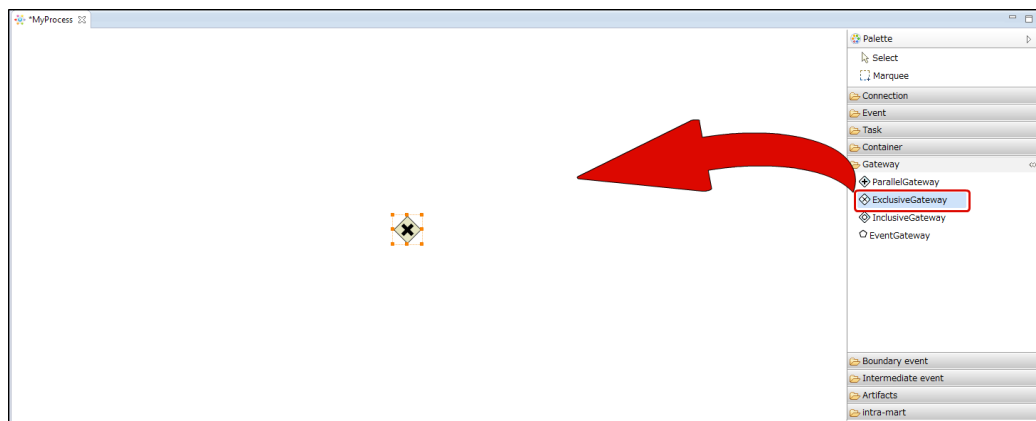
- 排他ゲートウェイで分岐を行うためには、排他ゲートウェイから複数のアクティビティにシーケンスフローを接続する必要があります。
- 排他ゲートウェイで結合を行うためには、複数のアクティビティから排他ゲートウェイにシーケンスフローを接続する必要があります。

排他ゲートウェイの配置

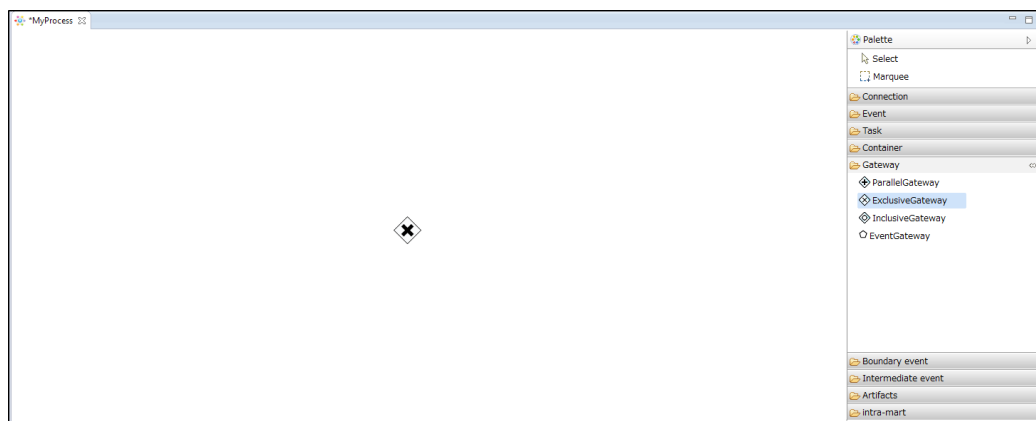
1. IM-BPM Designerの画面右側PaletteのGatewayにあるExclusiveGatewayをクリックします。



2. エディタ上の排他ゲートウェイを定義したい場所をクリックします。



3. これで排他ゲートウェイが定義されました。

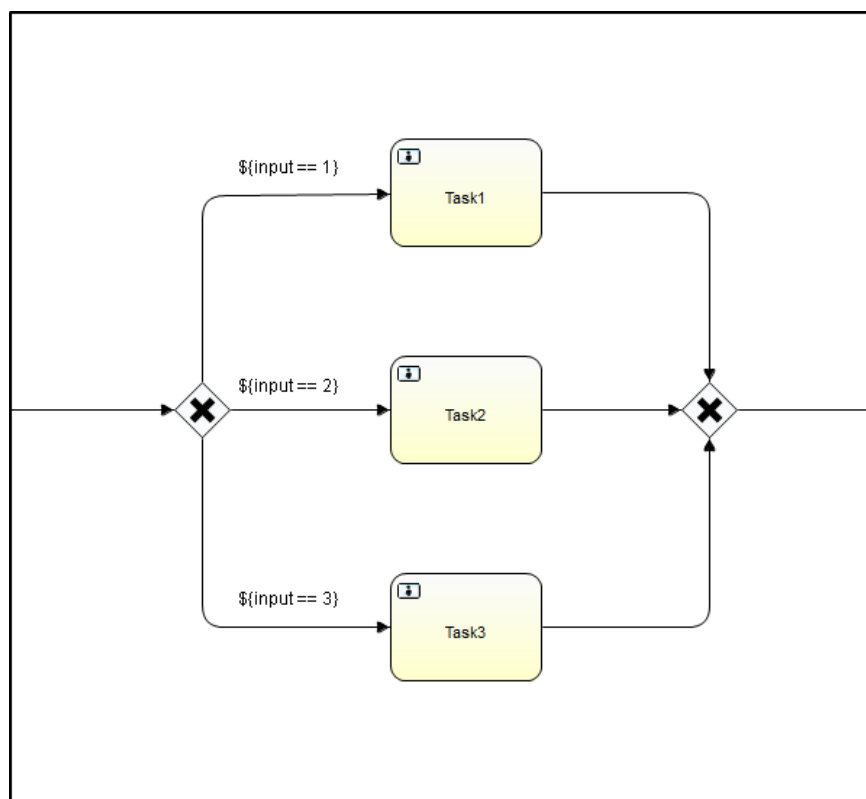


排他ゲートウェイによる分岐と結合

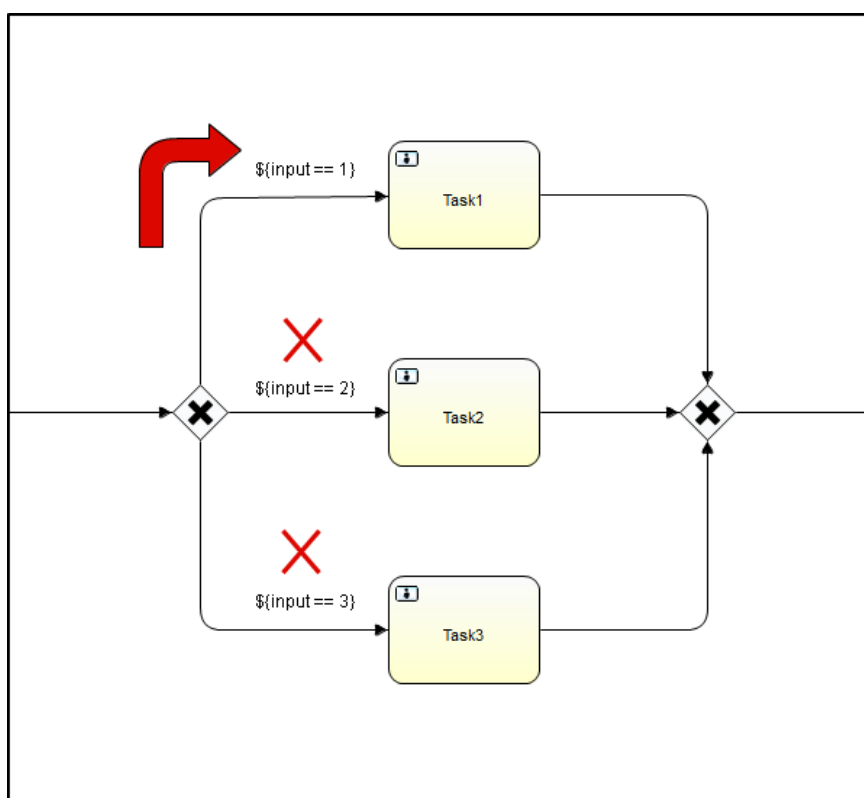
- 次のように排他ゲートウェイから分岐するシーケンスフローそれぞれに条件が定義されているものとします。

変数「input」の値を評価する条件です。

シーケンスフロー	条件
Task1へのシーケンスフロー	<code>\${input == 1}</code>
Task2へのシーケンスフロー	<code>\${input == 2}</code>
Task3へのシーケンスフロー	<code>\${input == 3}</code>



- 排他ゲートウェイでは接続されているシーケンスフローに定義されている条件を評価し、真となる1つのシーケンスフローに進みます。複数の条件が真となる場合、最初に真となったシーケンスフローに進みます。変数「input」の値が1の場合は、次のように進みます。



InclusiveGateway(包括ゲートウェイ)

条件式を評価する事で分岐先を決定することができます。

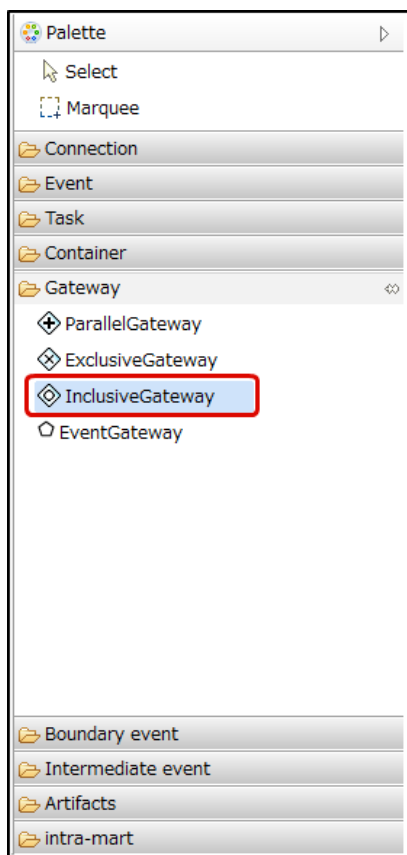
条件式を評価するためには接続されている「[Sequence Flow](#)」に条件式の設定が必要です。

排他ゲートウェイと異なる点は、評価が真となるシーケンスフローに対して1つの分岐先に決定せず、同時並行で次の処理に進むことができます。

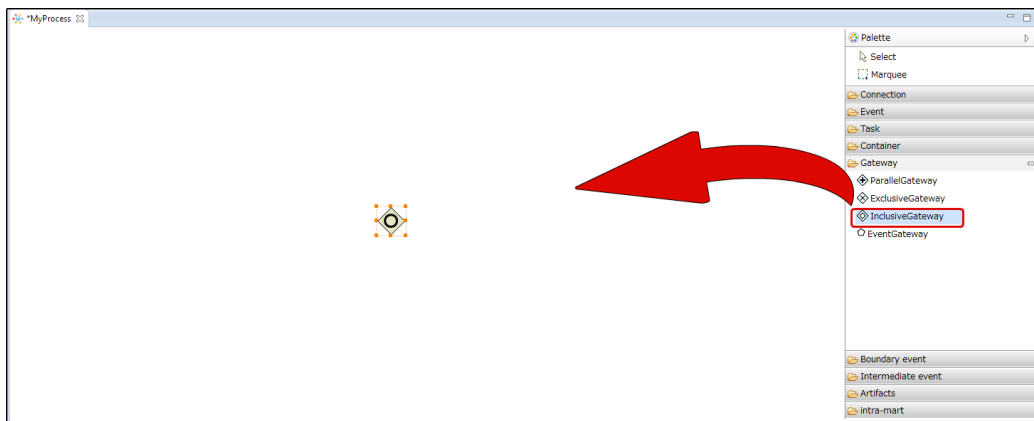
- 包括ゲートウェイで分岐を行うためには、包括ゲートウェイから複数のアクティビティにシーケンスフローを接続する必要があります。
- 包括ゲートウェイで結合を行うためには、複数のアクティビティから包括ゲートウェイにシーケンスフローを接続する必要があります。

包括ゲートウェイの配置

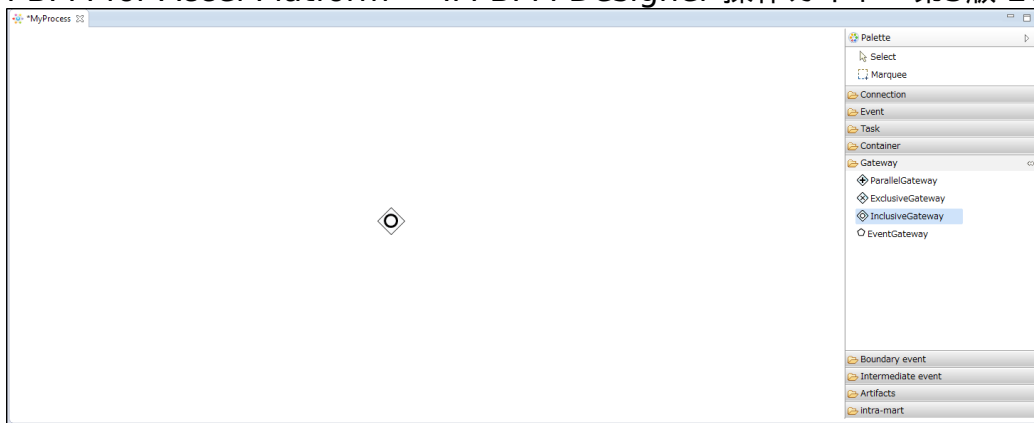
1. IM-BPM Designerの画面右側PaletteのGatewayにあるInclusiveGatewayをクリックします。



2. エディタ上の包括ゲートウェイを定義したい場所をクリックします。



3. これで包括ゲートウェイが定義されました。

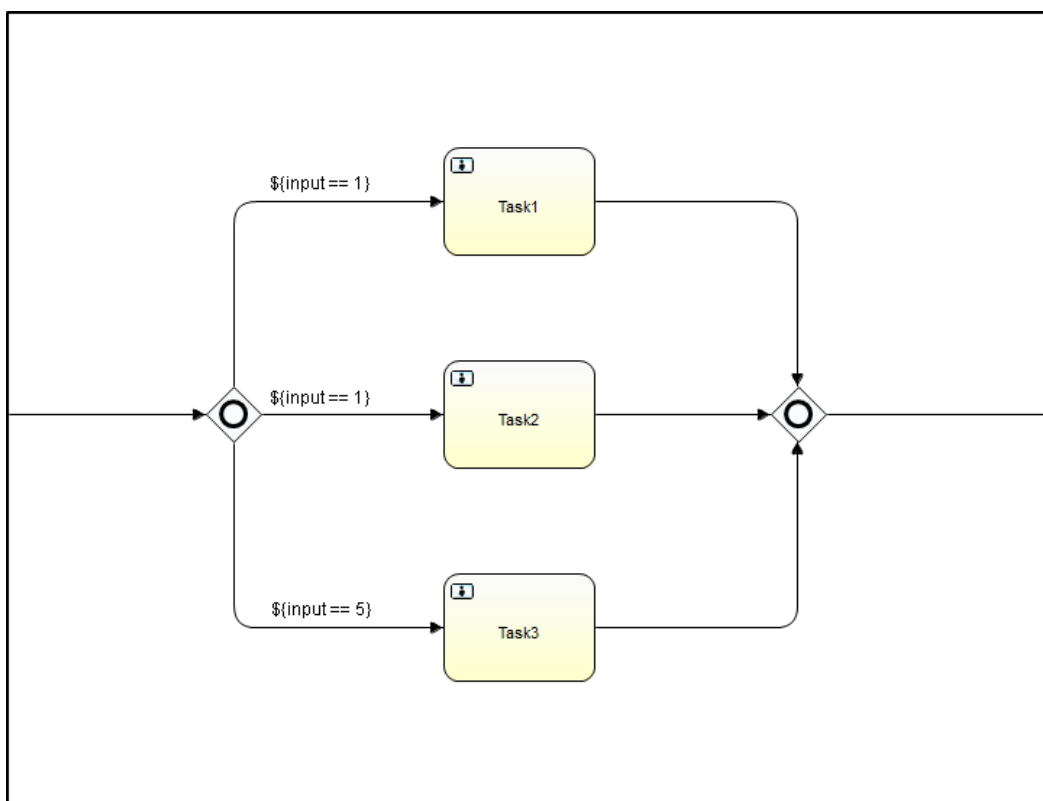


包括ゲートウェイによる分岐と結合

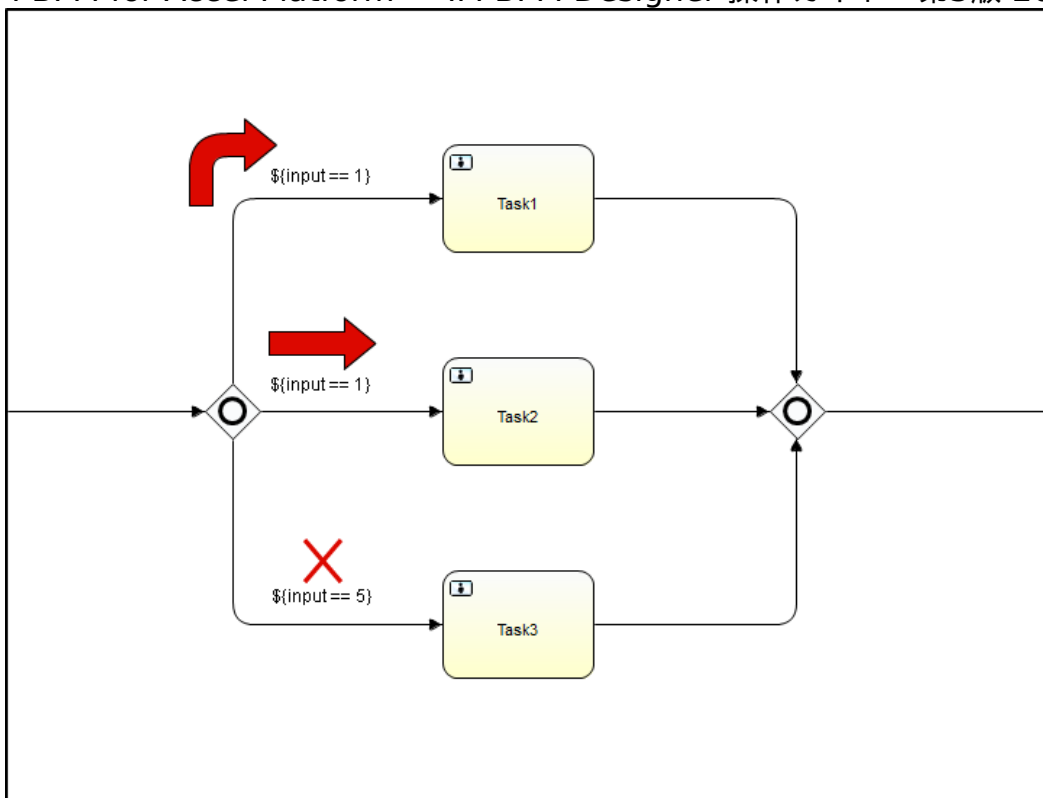
- 次のように包括ゲートウェイから分岐するシーケンスフローそれぞれに条件が定義されているものとします。

変数「input」の値を評価する条件です。

シーケンスフロー	条件
Task1へのシーケンスフロー	<code>\${input == 1}</code>
Task2へのシーケンスフロー	<code>\${input == 1}</code>
Task3へのシーケンスフロー	<code>\${input == 5}</code>



- 包括ゲートウェイでは接続されているシーケンスフローに定義されている条件を評価し、真となるシーケンスフローが並行に進みます。
- 変数「input」の値が1の場合は、次のように進みます。



EventGateway(イベントゲートウェイ)

イベントに対して条件を評価する事で分岐先を決定することができます。

中間イベントのキャッチイベントに接続されている必要があります(キャッチイベントの詳細については「[中間イベント](#)」を参照してください)。

イベントゲートウェイは他のゲートウェイと分岐の仕方が異なり、シーケンスフローで接続したキャッチイベントの状態と真偽を評価します。

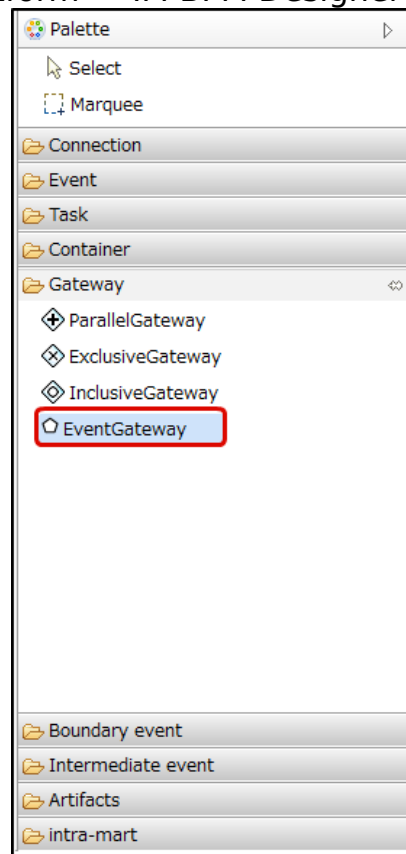
接続先のキャッチイベントが実行状態になった時に、真であると判断します。

複数の評価が真となる場合、最初に評価が真となる1つのシーケンスフローだけを継続して次の処理に進む事を表します。

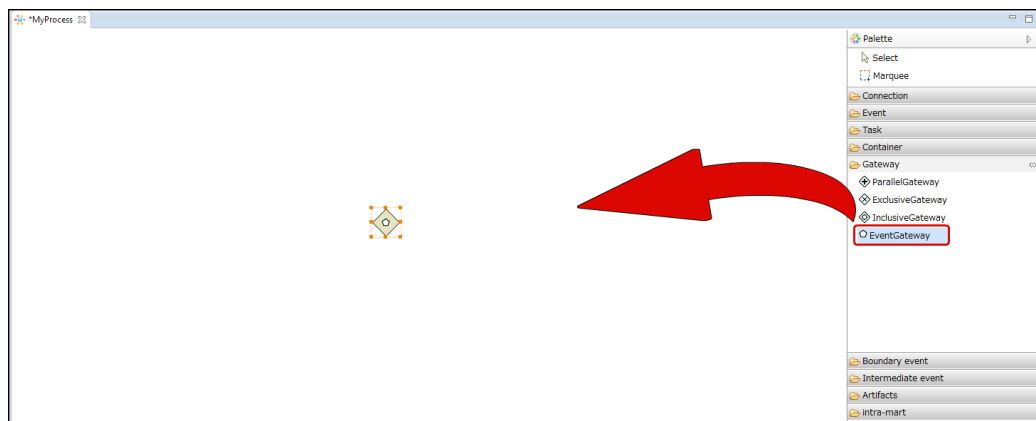
イベントゲートウェイで分岐を行うためには、イベントゲートウェイから複数のキャッチイベントにシーケンスフローを接続する必要があります。

EventGatewayの配置

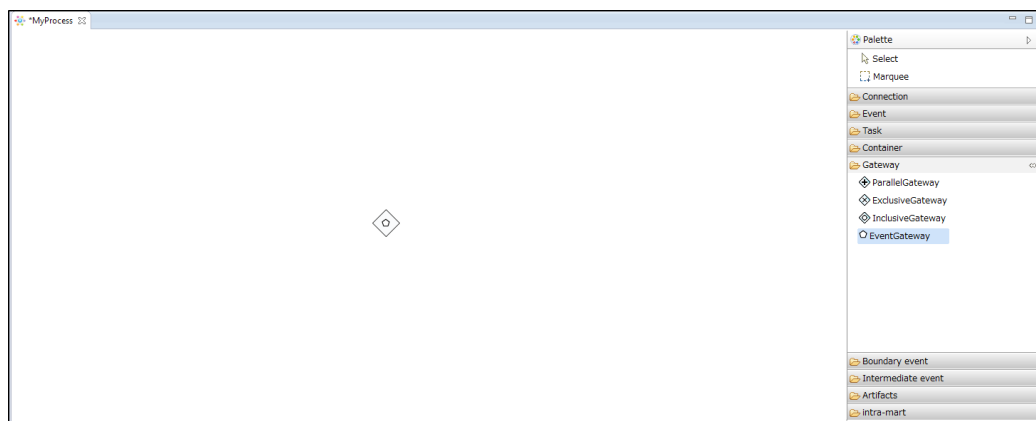
1. IM-BPM Designerの画面右側PaletteのGatewayにあるEventGatewayをクリックします。



2. エディタ上のイベントゲートウェイを定義したい場所をクリックします。

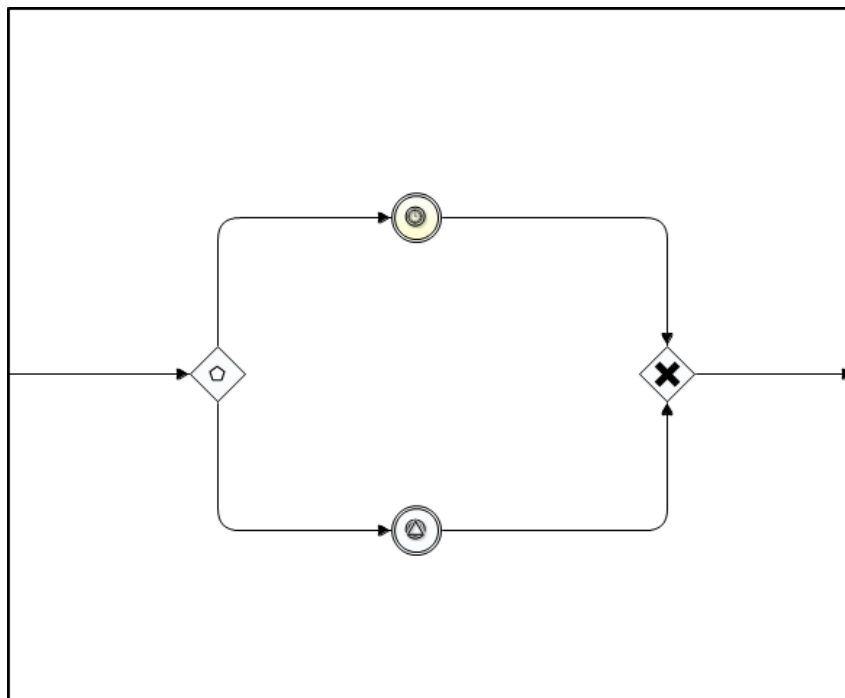


3. これでイベントゲートウェイが定義されました。

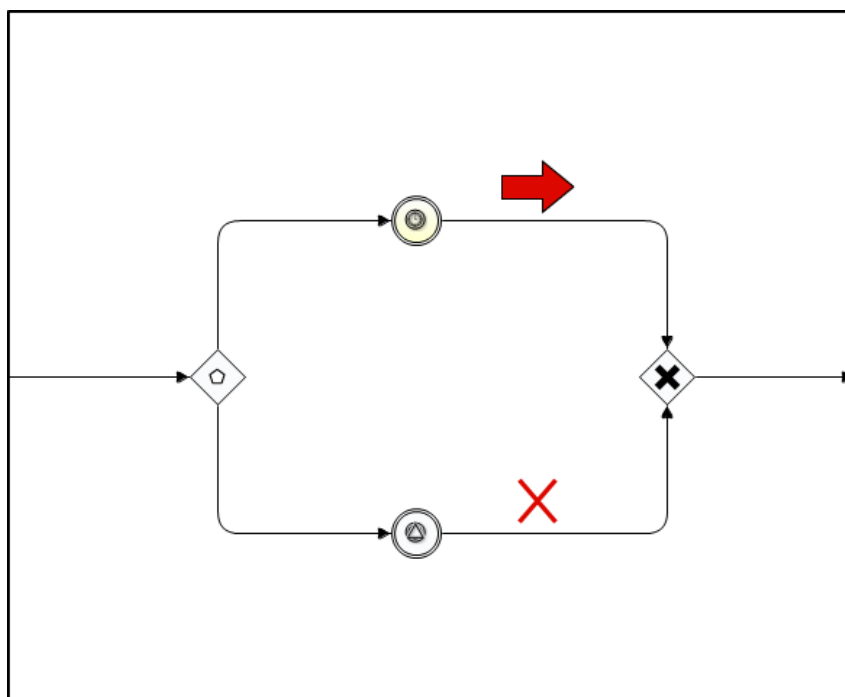


イベントゲートウェイによる分岐

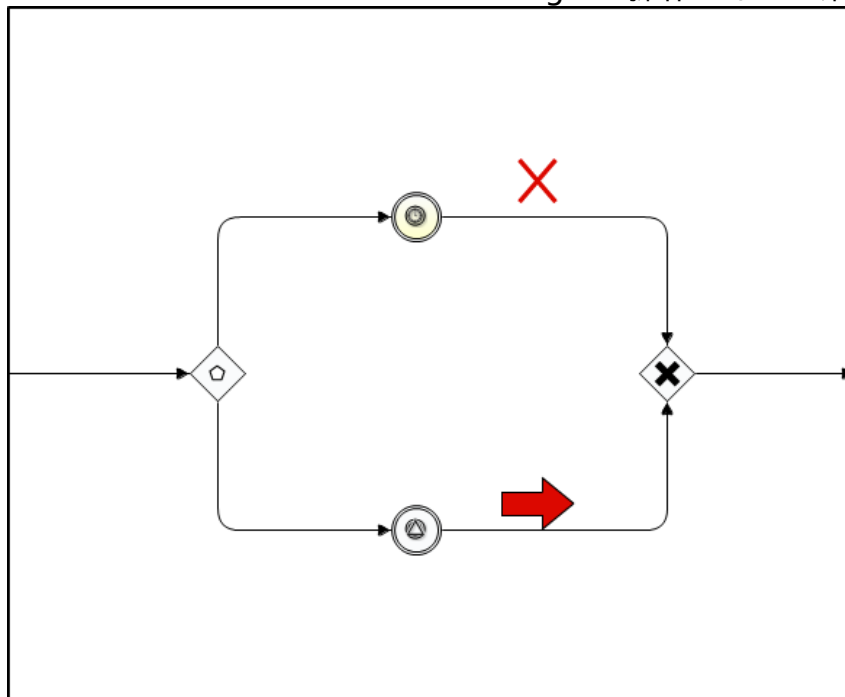
- 次のようにイベントゲートウェイで分岐を行う場合には、キャッチイベントを配置しシーケンスフローで接続している必要があります。
- ここでは[TimerCatchingEvent](#)と[SignalCatchingEvent](#)に接続しています。



- EventGatewayにシーケンスフローが到達すると[TimerCatchingEvent](#)と[SignalCatchingEvent](#)がそれぞれ待機状態となります。
- [TimerCatchingEvent](#)に設定された期間が経過すると[SignalCatchingEvent](#)の処理は中断され[TimerCatchingEvent](#)からのシーケンスフローだけが次に進みます。



- [TimerCatchingEvent](#)に設定された期間が経過するより早く[SignalCatchingEvent](#)がシグナルを受信した場合は、[TimerCatchingEvent](#)の処理が中断され[SignalCatchingEvent](#)からのシーケンスフローだけが次に進みます。



Default flow(デフォルトフロー)

分岐を行う際に、デフォルトフローの設定ができます。

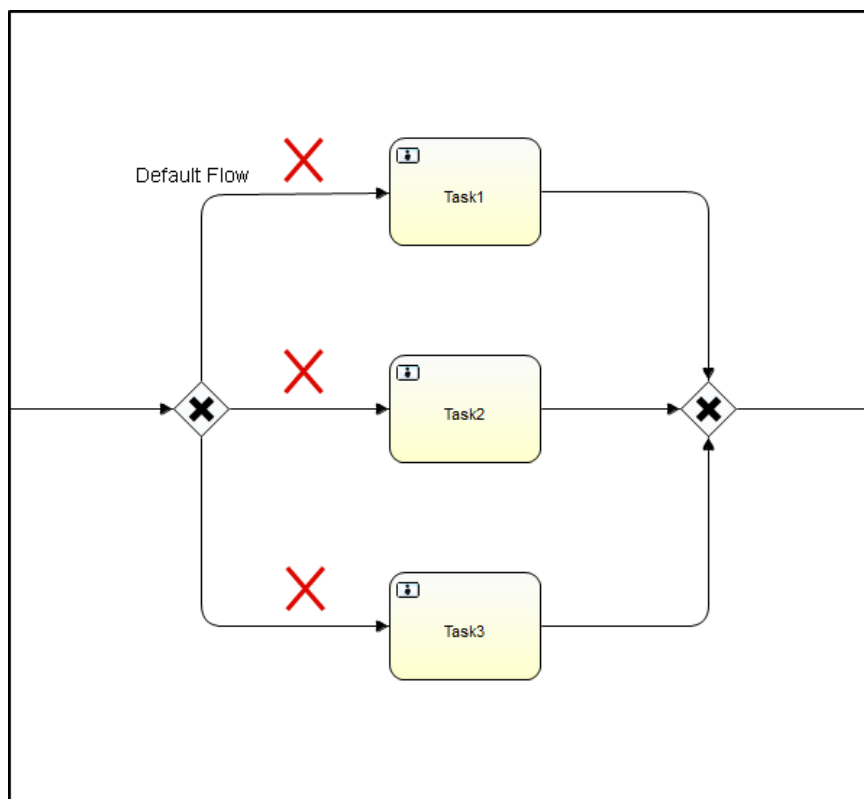
デフォルトフローは分岐を行う際に、接続されているシーケンスフローに設定された条件が全て偽となった場合に進行するシーケンスフローです。

デフォルトフローを設定していない場合に、全てのシーケンスフローで条件が偽になるとプロセスはそれ以上進む事ができなくなりエラーが発生します。

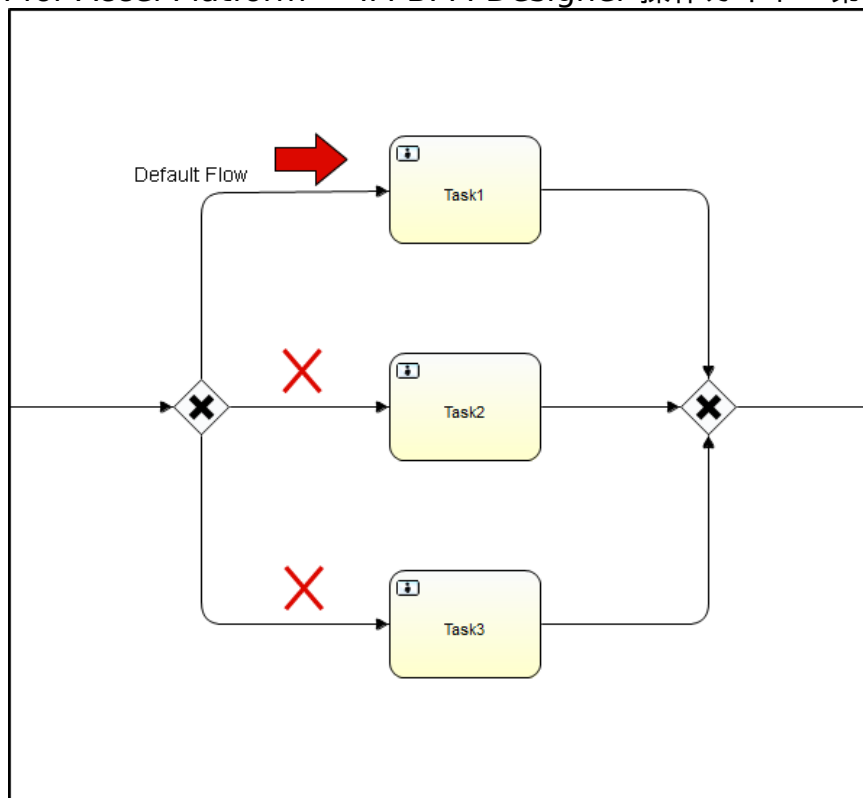
設定方法については「[General](#)」を参照してください。

デフォルトフローによる分岐

- ゲートウェイの分岐に接続された全てのシーケンスフローの条件が偽になるとプロセスはそれ以上進む事ができなくなります。
 - ここではExclusiveGatewayによる分岐を行っています。



- デフォルトフローが設定されているとそのシーケンスフローの条件が偽であっても次の処理に進む事ができます。



境界イベント

境界イベントは、接続しているアクティビティの処理中にイベントを受信した場合に起動します。

境界イベントを設定したアクティビティでイベントを受信した際、アクティビティを中断させて境界イベントから引かれたシーケンスフローにフローを進行させます。

境界イベントは、発生するイベントの種類によって4つに分類されます

- [TimerBoundaryEvent](#)
- [ErrorBoundaryEvent](#)
- [MessageBoundaryEvent](#)
- [SignalBoundaryEvent](#)

TimerBoundaryEvent(タイマー境界イベント)

タイマー境界イベントは設定した時間情報を元にイベントを受信する境界イベントです。

指定時間が経過するか、または特定の日時になった場合にタイマー境界イベントから引かれたシーケンスフローへとフローを進行させます。

また、タイマー境界イベントがイベントを受信した際、通常のプロセスフローの継続が中止を設定できます。

タイマー境界イベントの配置

タイマー境界イベントを配置するためには、事前にタイマー境界イベントが接続可能なアクティビティを配置しておく必要があります。接続可能なアクティビティは以下の通りです。

アクティビティ名 **Palette**

UserTask	Task
----------	------

ReceiveTask	Task
-------------	------

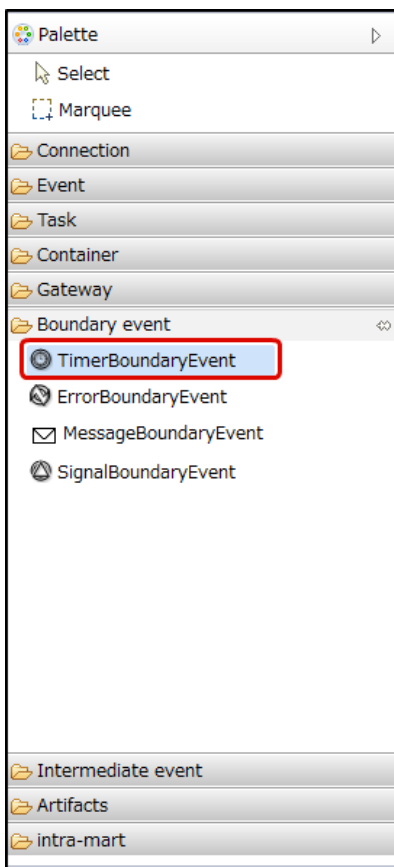
CallActivity	Task
--------------	------

SubProcess	Container
------------	-----------

ApplyTask	intra-mart
-----------	------------

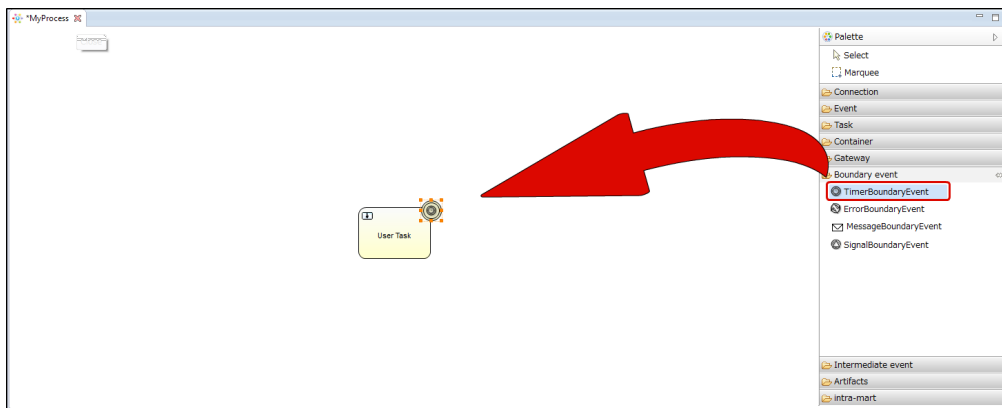
DraftTask	intra-mart
-----------	------------

1. IM-BPM Designerの画面右側PaletteのBoundary eventにあるTimerBoundaryEventをクリックします。

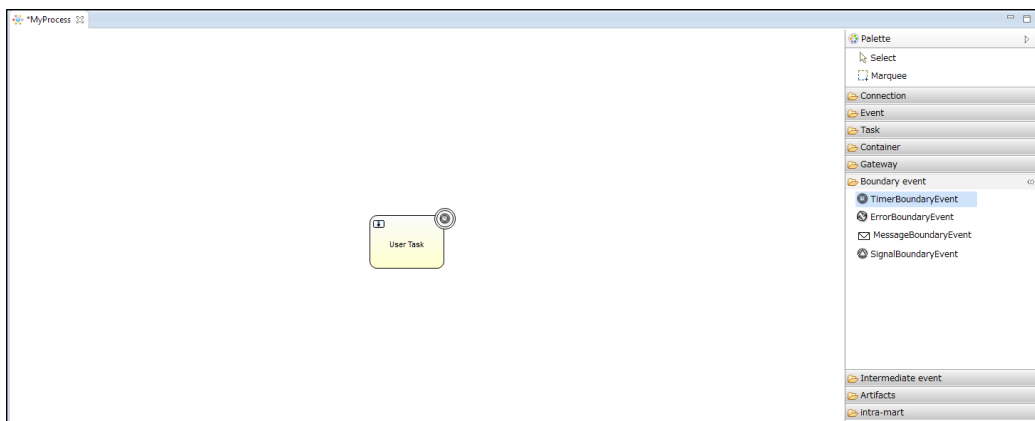


2. エディタ上のタイマー境界イベントを定義したいアクティビティをクリックします。

- ここではユーザタスクに対してタイマー境界イベントを定義しています。



3. これでタイマー境界イベントが定義されました。

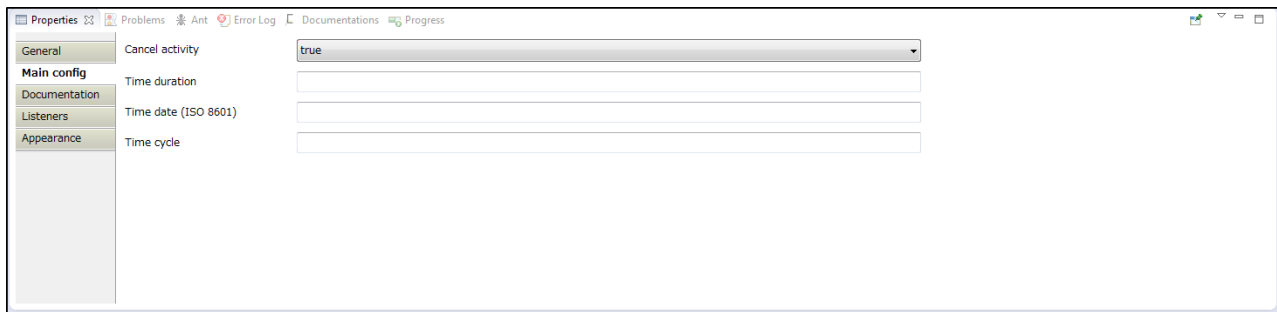


プロパティの設定

タイマー境界イベントのメインコンフィグの設定について解説します。

入力方式は日付と時刻の表記に関する国際規格であるISO 8601の形式に則って記述する必要があります。

具体的な記述方法については説明欄を参照してください。



項目名	説明
Cancel activity(アクティビティの取消)	タイマー境界イベントが開始したときに接続しているアクティビティを終了するか継続するかを設定します。
Time duration(期間)	開始するまでの期間を指定します。 タイマー境界イベントに到達したときから指定した期間が経過したとき1回のみ開始されます。 例:10日後開始→P10D
Time date(日付)	開始する日付を指定します。 指定日に1回のみ開始されます。 例:2016年1月1日12時開始→ 2016-01-01T12:00:00
Time cycle(周期)	開始する周期を指定します。 例:10時間毎に3回開始→R3/PT10H

ErrorBoundaryEvent(エラー境界イベント)

エラー境界イベントは接続されたアクティビティ内でのエラーを受信する境界イベントです。

エラー境界イベントで設定したアクティビティ内で発生したエラーコードを比較して合致した場合、エラー境界イベントから引かれたシーケンスフローへとフローを進行させます。

エラー境界イベントがイベントを受信した際、エラー境界イベントを接続したアクティビティ内のサブプロセスを含む全てのプロセスは中断されます。

エラー境界イベントにエラーコードを指定しなかった場合は、アクティビティ内で発生した全てのエラーコードを受信ようになります。

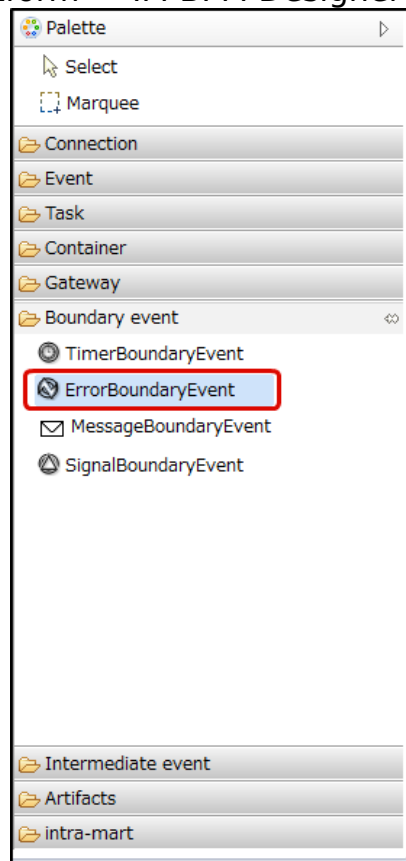
エラー境界イベントの配置

エラー境界イベントを配置するためには、事前にエラー境界イベントが接続可能なアクティビティを配置しておく必要があります。

接続可能なアクティビティは以下の通りです。

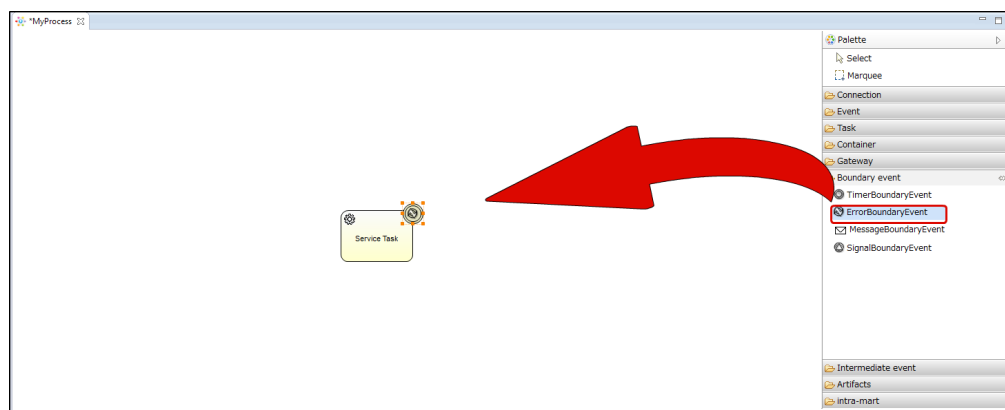
アクティビティ名	Palette
ServiceTask	Task
CallActivity	Task
SubProcess	Container
IMLogicDesignerTask	intra-mart

1. IM-BPM Designerの画面右側PaletteのBoundary eventにあるErrorBoundaryEventをクリックします。

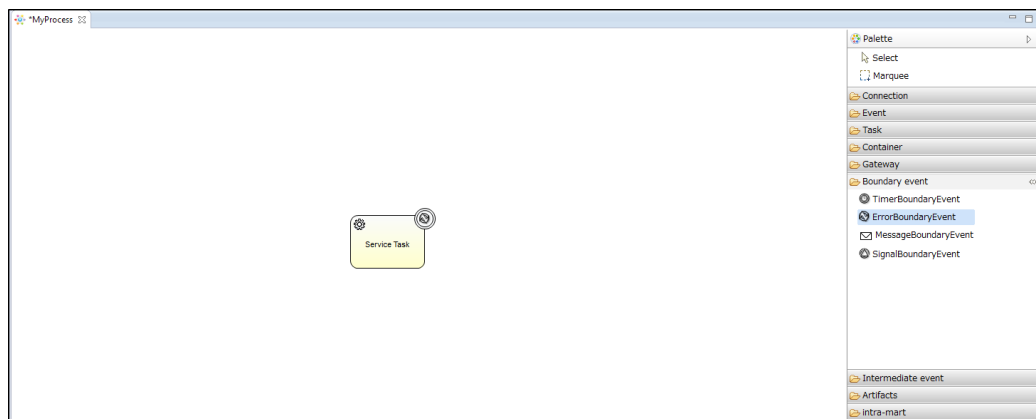


2. エディタ上のエラー境界イベントを定義したいアクティビティをクリックします。

- ここではサービスタスクに対してエラー境界イベントを定義しています。

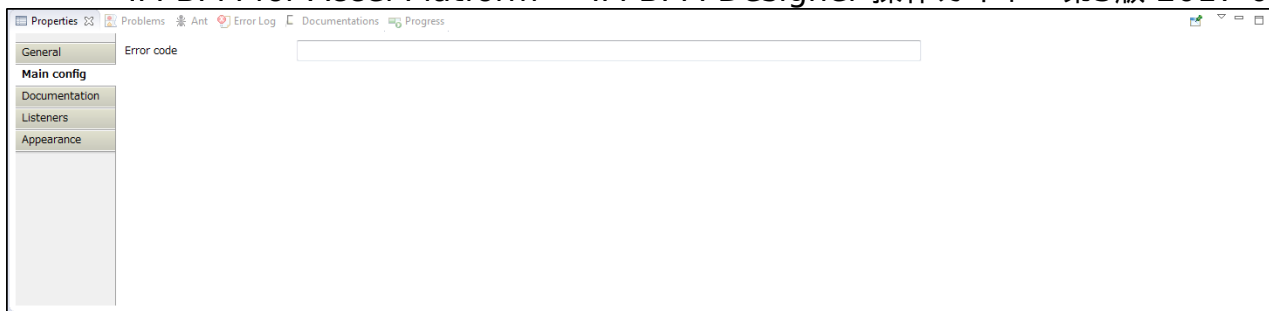


3. これでエラー境界イベントが定義されました。



プロパティの設定

エラー境界イベントのメインコンフィグの設定について解説します。



項目名	説明
Error code(エラーコード)	設定したアクティビティで受信するエラーコードを文字列で定義します。 ここで設定したエラーコードを受信した場合、エラー境界イベントから引かれたシーケンスフローへとフローを進行させます。 エラーコードを指定しなかった場合、設定したアクティビティで発生したすべてのエラーイベントを受信します。

MessageBoundaryEvent(メッセージ境界イベント)

メッセージ境界イベントは接続されたアクティビティ内から送信されたメッセージを受信する境界イベントです。

メッセージ境界イベントで設定したメッセージと同じ名前のメッセージを受信した場合、メッセージ境界イベントから引かれたシーケンスフローへとフローを進行させます。

また、メッセージ境界イベントがメッセージを受信した際、通常のプロセスフローの継続か中止を設定できます。

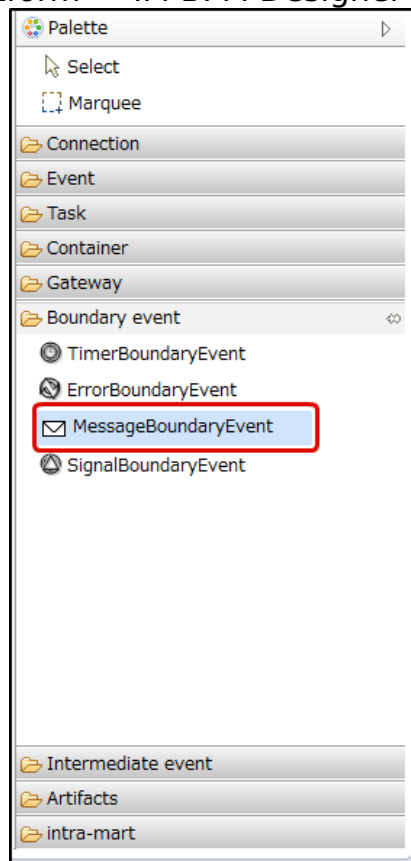
メッセージ境界イベントの配置

メッセージ境界イベントを配置するためには、事前にメッセージ境界イベントが接続可能なアクティビティを配置しておく必要があります。

接続可能なアクティビティは以下の通りです。

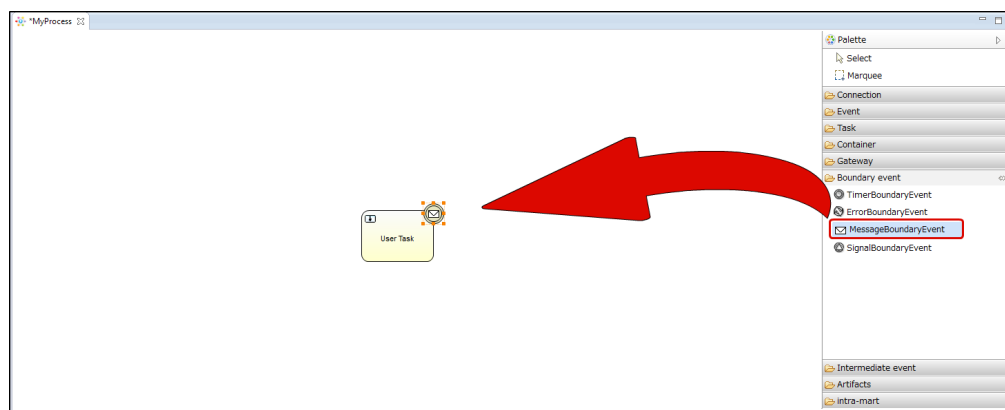
アクティビティ名 Palette	
UserTask	Task
ServiceTask	Task
ReciveTask	Task
CallActivity	Task
SubProcess	Container
ApplyTask	intra-mart
DraftTask	intra-mart

1. IM-BPM Designerの画面右側PaletteのBoundary eventにあるMessageBoundaryEventをクリックします。

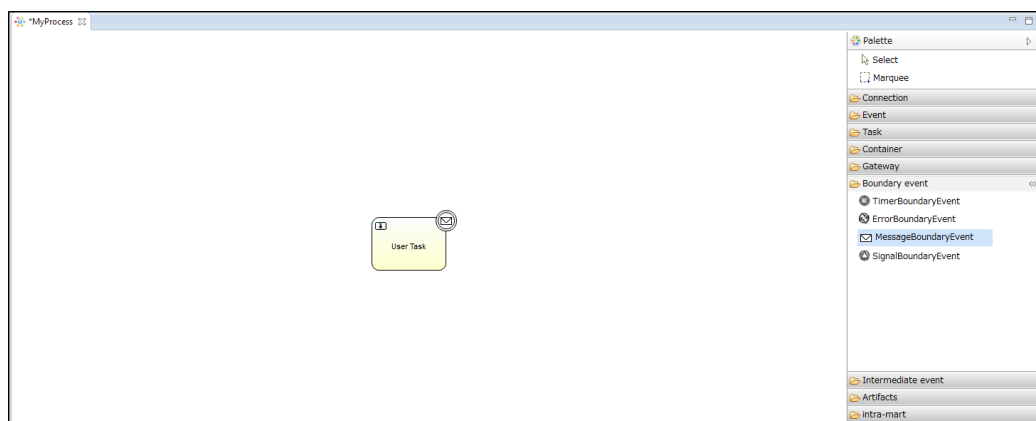


2. エディタ上のメッセージ境界イベントを定義したいアクティビティをクリックします。

- ここではユーザタスクに対してメッセージ境界イベントを定義しています。

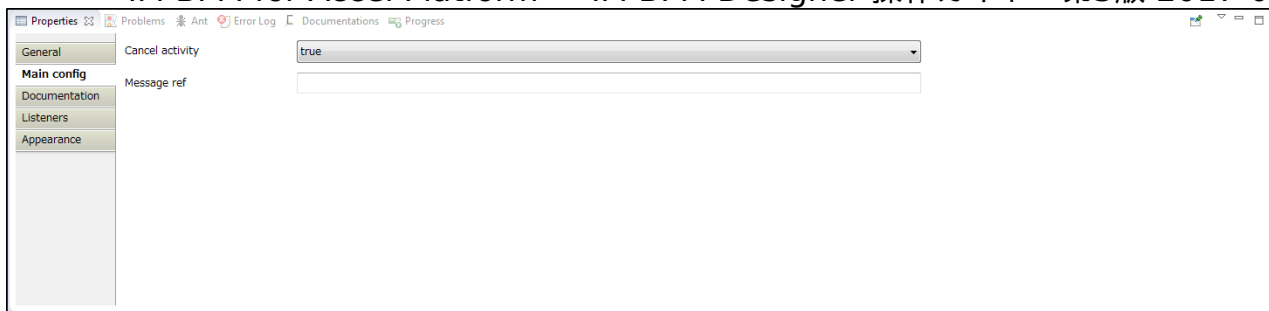


3. これでメッセージ境界イベントが定義されました。



プロパティの設定

メッセージ境界イベントのメインコンフィグの設定について解説します。



項目名	説明
Cancel activity(アクティビティの取消)	メッセージ境界イベントが開始したときに接続しているアクティビティを終了するか継続するかを設定します。
Message ref(参照メッセージ)	受信するメッセージを文字列で定義します。 ここで定義したメッセージを受信すると、メッセージ境界イベントから引かれたシーケンスフローへとフローを進行させます。

SignalBoundaryEvent(シグナル境界イベント)

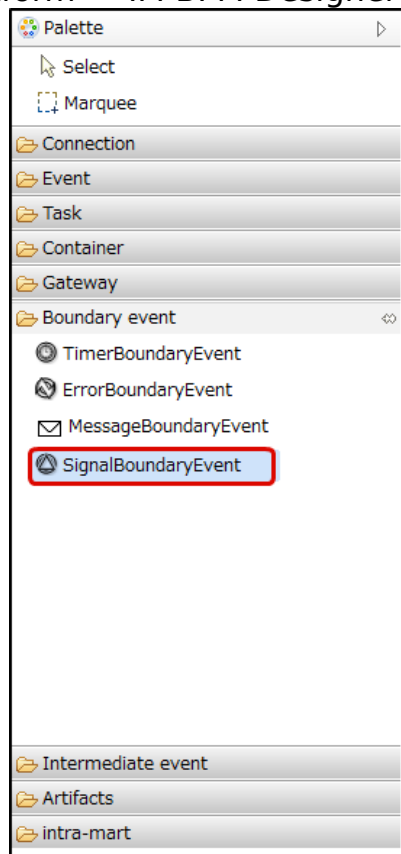
シグナル境界イベントは接続されたアクティビティ内から送信されたシグナルを受信する境界イベントです。シグナル境界イベントで設定したシグナルを受信した場合、シグナル境界イベントから引かれたシーケンスフローへとフローを進行させます。また、シグナル境界イベントがシグナルを受信した際、通常のプロセスフローの継続が中止を設定できます。

シグナル境界イベントの配置

シグナル境界イベントを配置するためには、事前にシグナル境界イベントが接続可能なアクティビティを配置しておく必要があります。接続可能なアクティビティは以下の通りです。

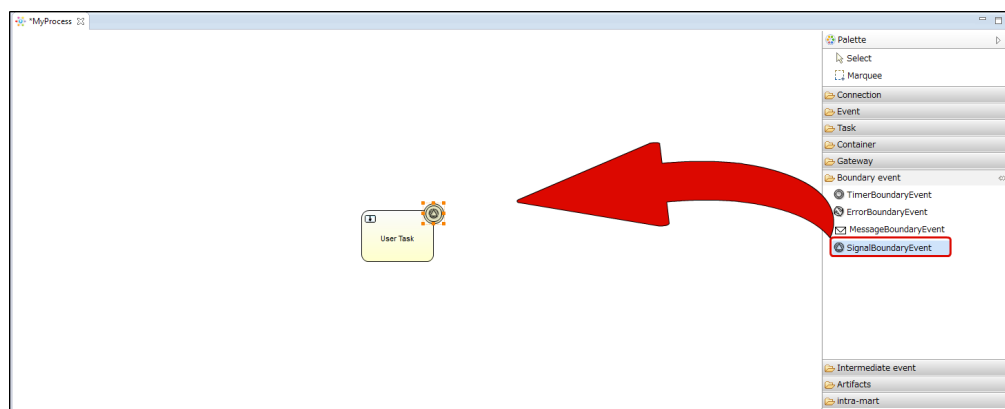
アクティビティ名 Palette	
UserTask	Task
ServiceTask	Task
ReciveTask	Task
CallActivity	Task
SubProcess	Container
ApplyTask	intra-mart
DraftTask	intra-mart

1. IM-BPM Designerの画面右側PaletteのBoundary eventにあるSignalBoundaryEventをクリックします。

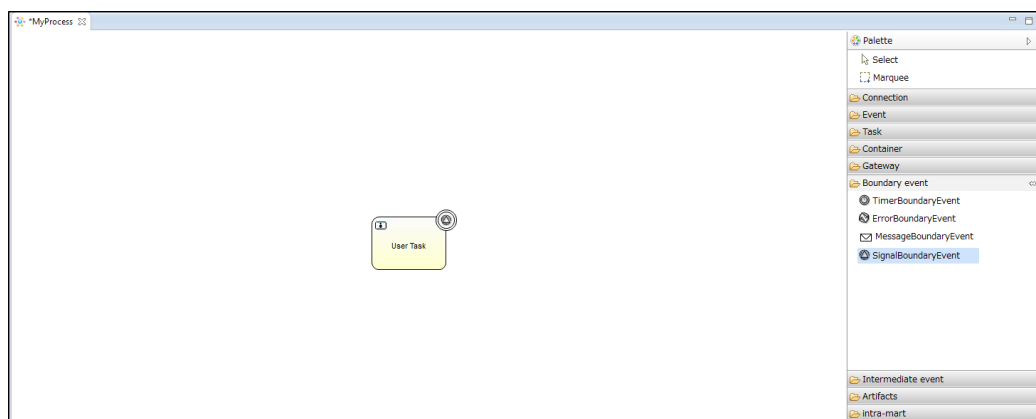


2. エディタ上のシグナル境界イベントを定義したいアクティビティをクリックします。

- ここではユーザタスクに対してシグナル境界イベントを定義しています。

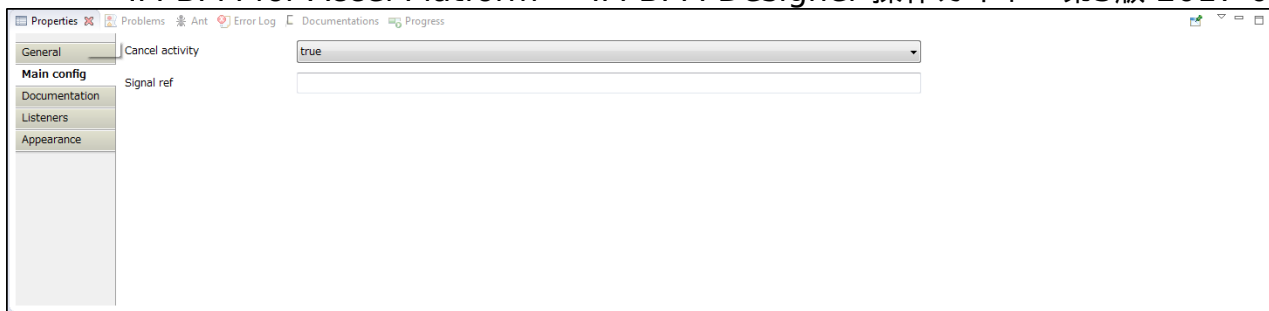


3. これでシグナル境界イベントが定義されました。



プロパティの設定

シグナル境界イベントのメインコンフィグの設定について解説します。



項目名	説明
Cancel activity(アクティビティの取消)	シグナル境界イベントが開始したときに接続しているアクティビティを終了するか継続するかを設定します。
Signal ref(参照シグナル)	受信するシグナルを文字列で定義します。 ここで定義したメッセージを受信すると、シグナル境界イベントから引かれたシーケンスフローへとフローを進行させます。

中間イベント

進行中のプロセス定義で、発生するイベントを表します。

中間イベントはイベント発生起因となるトリガーに反応したり、設定したりするためのフローエレメントです。

IM-BPM Designerでは、中間イベントは発生するイベントの種類によって5つに分類されます。

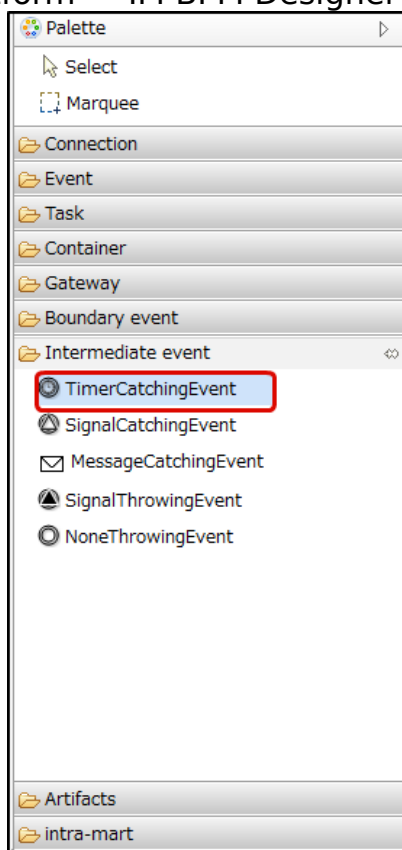
- [TimerCatchingEvent](#)
- [SignalCatchingEvent](#)
- [MessageCatchingEvent](#)
- [SignalThrowingEvent](#)
- [NoneThrowingEvent](#)

TimerCatchingEvent(タイマーキャッチイベント)

指定された時間が経過するか特定の日時になるまで、プロセスフローの実行を停止する中間イベントです。

タイマーキャッチイベントの配置

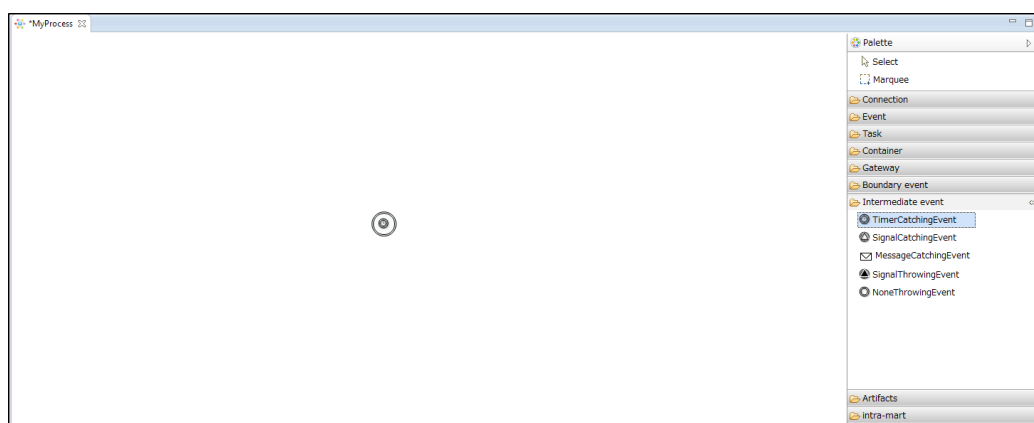
1. IM-BPM Designerの画面右側PaletteのIntermediate eventにあるTimerCatchingEventをクリックします。



2. エディタ上のタイマーキャッチイベントを定義したい場所をクリックします。

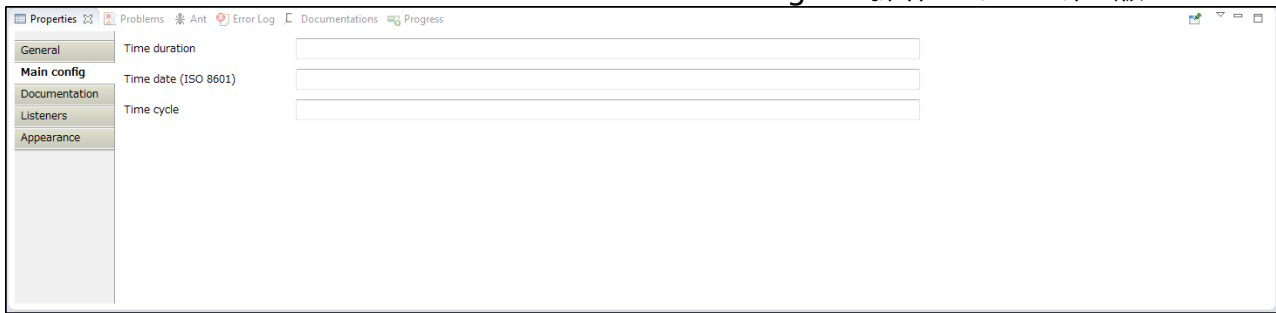


3. これでタイマーキャッチイベントが定義されました。



プロパティの設定

タイマーキャッチイベントのメインコンフィグの設定について解説します。



入力方式は日付と時刻の表記に関する国際規格であるISO 8601の形式に則って記述する必要があります。
 具体的な記述方法については説明欄を参照してください。

項目名	説明
Time duration(期間)	停止する期間を指定します。 TimerCatchingEventに到達したときから指定した期間が経過したとき1回のみ開始されます。 例:10日後まで停止→P10D
Time date(日付)	停止を解除する日付を指定します。 指定日に停止が解除され、プロセスフローが再度進行します。 例:2016年1月1日12時まで停止→ 2016-01-01T12:00:00
Time cycle(周期)	タイマーキャッチイベントでは使用されません、

SignalCatchingEvent(シグナルキャッチイベント)

シグナルを受信する事をトリガーとして実行されることを示す中間イベントです。

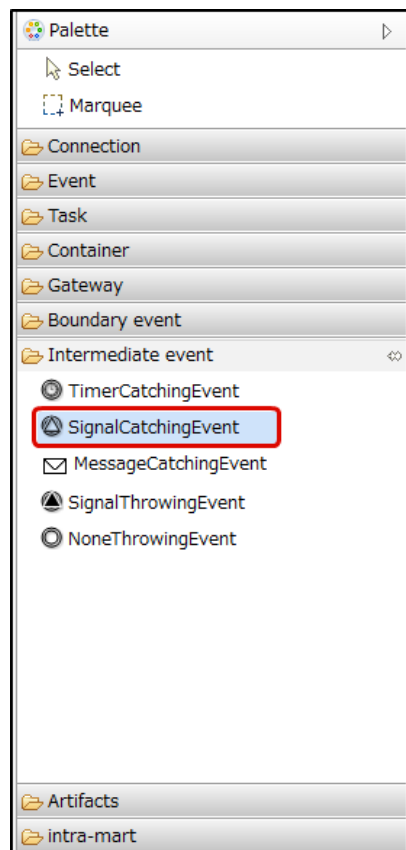
実行のトリガーとなるシグナルに特定の相手はいません。

異なるプロセスインスタンスからシグナルが送信された場合でも、このイベントはシグナルを受信して動作します。

また、同一のシグナルを受信するシグナルキャッチイベントが複数のプロセス定義にまたがっていた場合でも、定義されたシグナルを受信すると一斉に動作を開始します。

シグナルキャッチイベントの配置

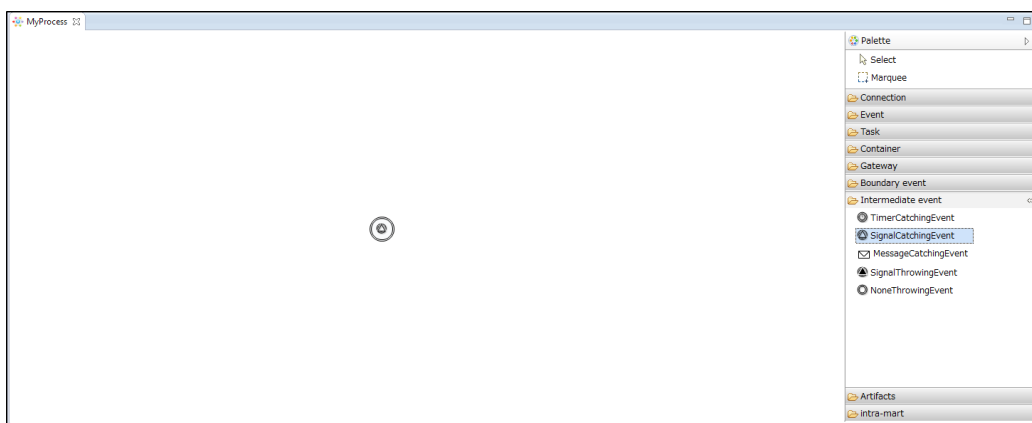
1. IM-BPM Designerの画面右側PaletteのIntermediate eventにあるSignalCatchingEventをクリックします。



2. エディタ上のシグナルキャッチイベントを定義したい場所をクリックします。

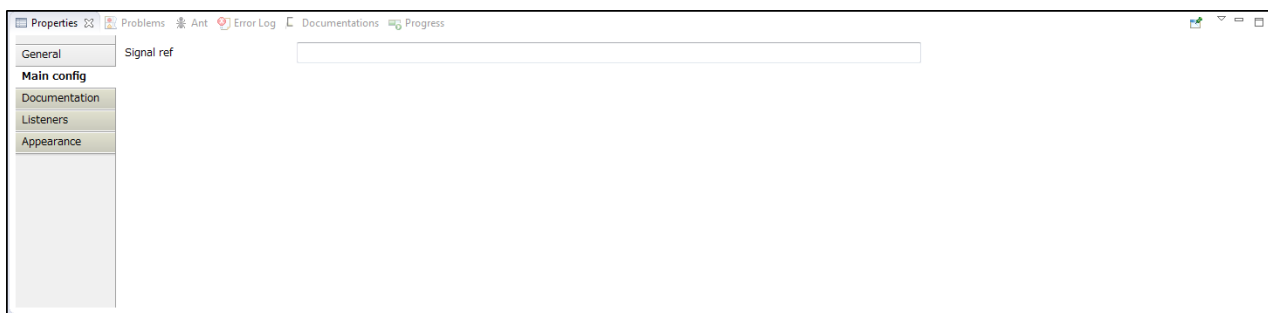


3. これでシグナルキャッチイベントが定義されました。



プロパティの設定

シグナルキャッチイベントのメインコンフィグの設定について解説します。



項目名	説明
Signal ref(参照シグナル)	受信するシグナルを定義します。 文字列で定義します。

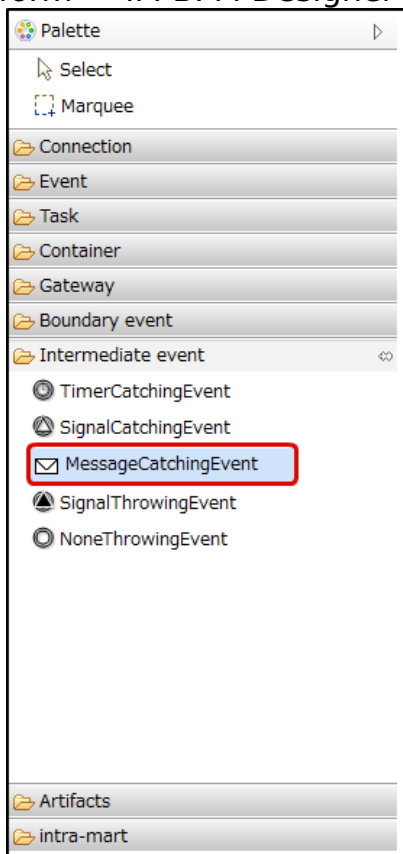
MessageCatchingEvent(メッセージキャッチイベント)

指定された名前のメッセージを受信する事を表します。

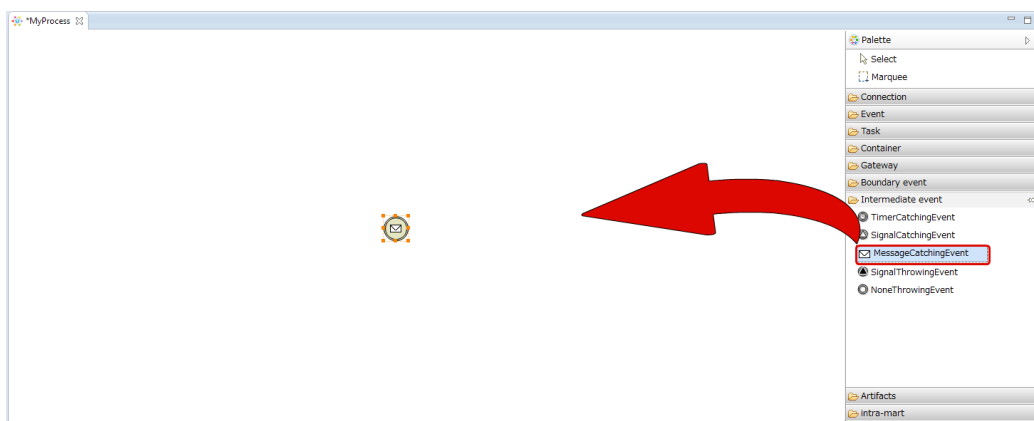
メッセージキャッチイベントでは、常に送信者と受信者の関係は「1対1」となります。

メッセージキャッチイベントの配置

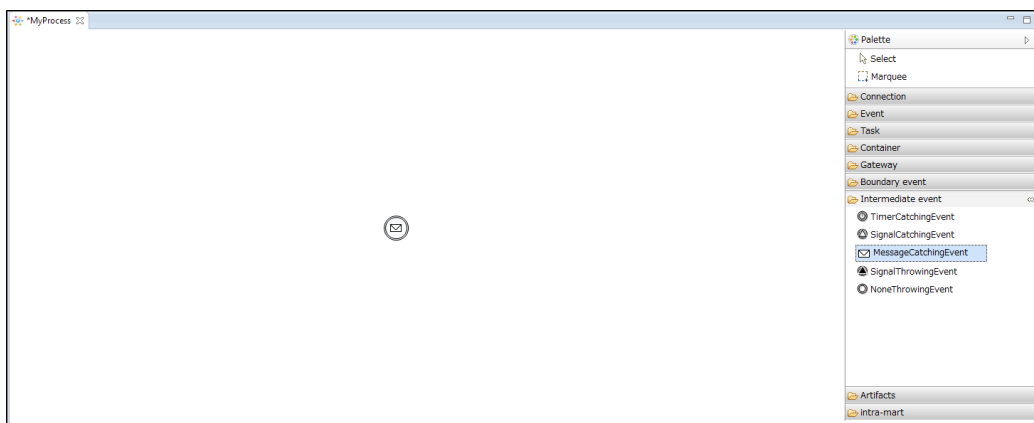
1. IM-BPM Designerの画面右側PaletteのIntermediate eventにあるMessageCatchingEventをクリックします。



2. エディタ上のメッセージキャッチイベントを定義したい場所をクリックします。

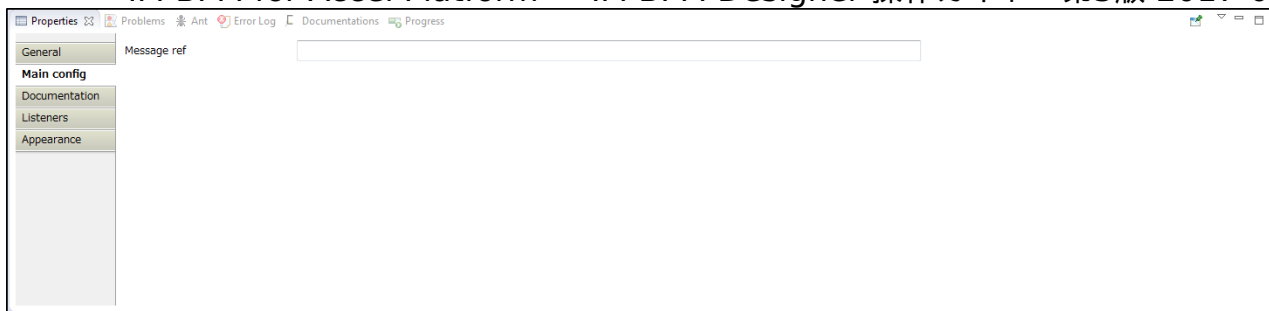


3. これでメッセージキャッチイベントが定義されました。



プロパティの設定

メッセージキャッチイベントのメインコンフィグの設定について解説します。



項目名	説明
Message ref(参照メッセージ)	受信するメッセージを定義します。 文字列で定義します。

SignalThrowingEvent(シグナルスローイベント)

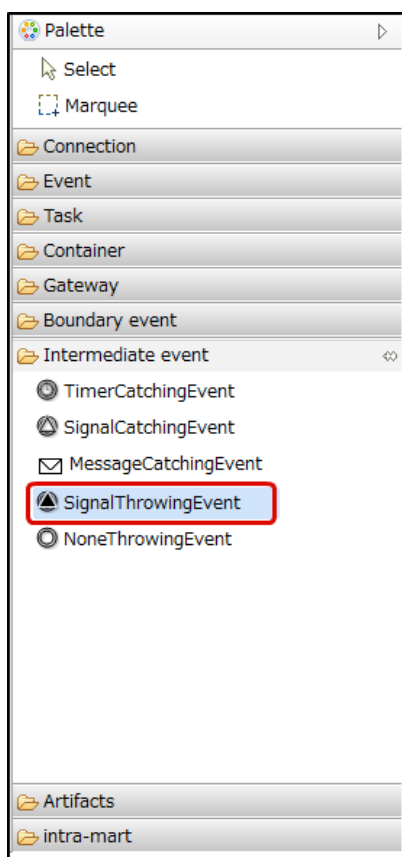
シグナルをブロードキャストする事を表します。

[SignalCatchingEvent](#) にも記述したように、シグナルをこのイベントで送信すると異なるプロセスにいるシグナルキャッチイベントもシグナルを受信します。

そのためシグナルは、送信者と受信者の関係が「1対多」となります。

シグナルスローイベントの配置

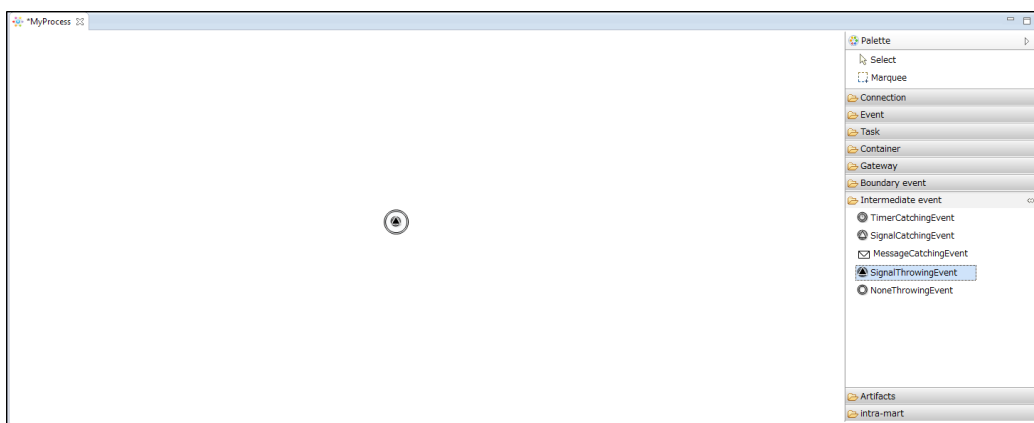
1. IM-BPM Designerの画面右側PaletteのIntermediate eventにあるSignalThrowingEventをクリックします。



2. エディタ上のシグナルスローイベントを定義したい場所をクリックします。

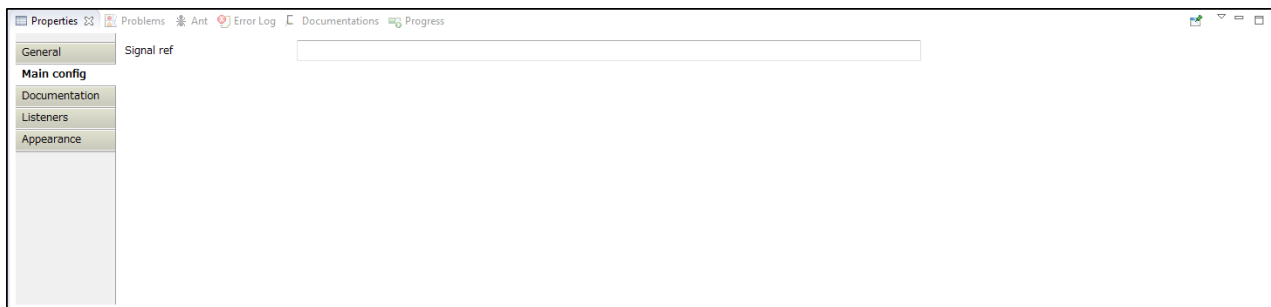


3. これでシグナルスローイベントが定義されました。



プロパティの設定

シグナルスローイベントのメインコンフィグの設定について解説します。



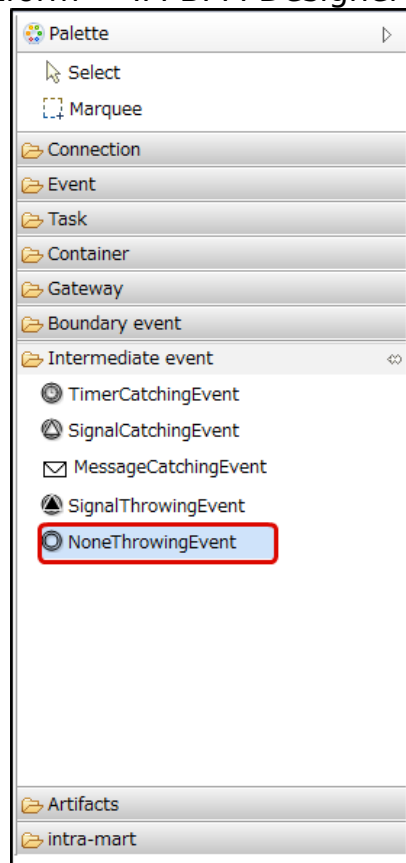
項目名	説明
Signal ref(参照シグナル)	送信するシグナルを定義します。 文字列で定義します。

NoneThrowingEvent(中間イベント)

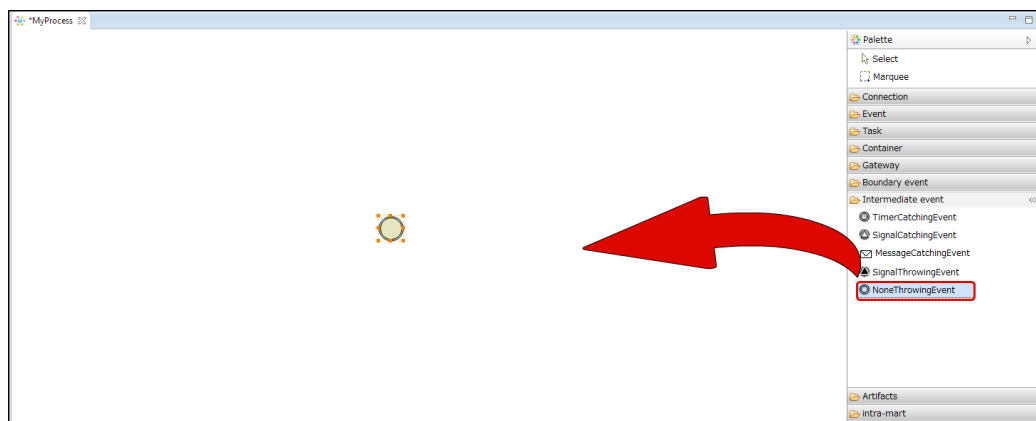
イベントの種類を示さない場合に利用され、プロセス定義の途中で何らかの出来事が生じ、状態や状況が変化することを表します。

中間イベントの配置

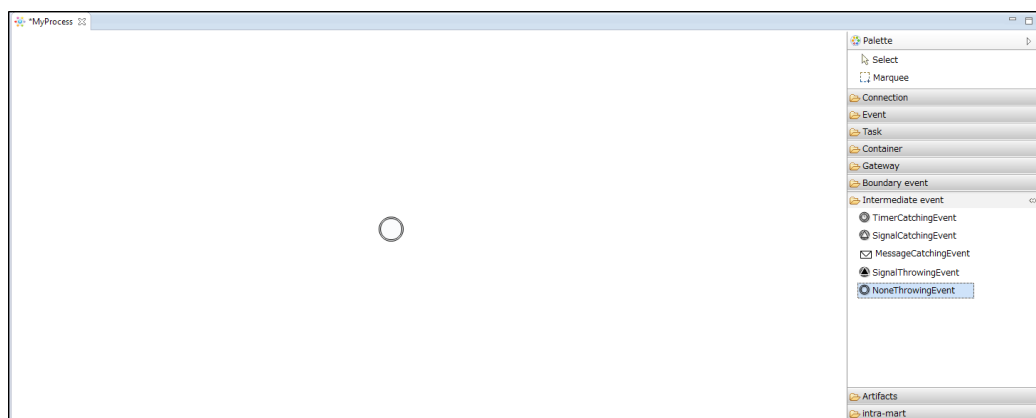
1. IM-BPM Designerの画面右側PaletteのIntermediate eventにあるNoneThrowingEventをクリックします。



2. エディタ上の中間イベントを定義したい場所をクリックします。



3. これで中間イベントが定義されました。



アーティファクト

アーティファクトとは、プロセス定義中をよりわかりやすくするために補助的に利用するオブジェクトです。

アーティファクトは、プロセス定義中のオブジェクトとアソシエーションで紐付けされるため、システム上でのフローの実行に影響はありません。

IM-BPM Designerでは以下の2種類のアーティファクトが設定可能です。

- 注釈
- データオブジェクト

注釈

プロセス定義中に配置されたオブジェクトに対して、補助的な説明を行うために利用されるアーティファクトです。

プロセス定義のフロー中にあるオブジェクトに対して、アソシエーションで接続して注釈と紐付けされ、その状態を破線で示します。

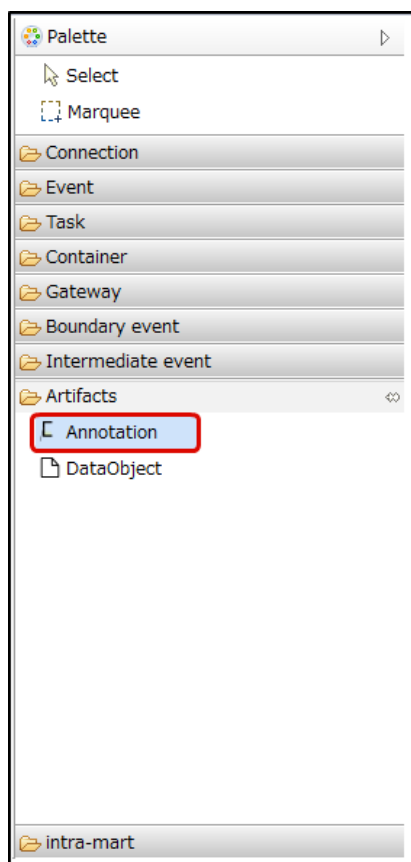
また、プロセス定義全体や特定のプール、レーンに配置する場合は対象となるプロセスの内部に注釈を配置することで紐付けます。

注釈の配置

例としてユーザタスクに注釈を定義する方法について解説します。

フロー中にあるタスク、イベント、ゲートウェイは同様の方法で注釈の設定ができます。

1. IM-BPM Designerの画面右側PaletteのArtifactsにあるAnnotationをクリックします。

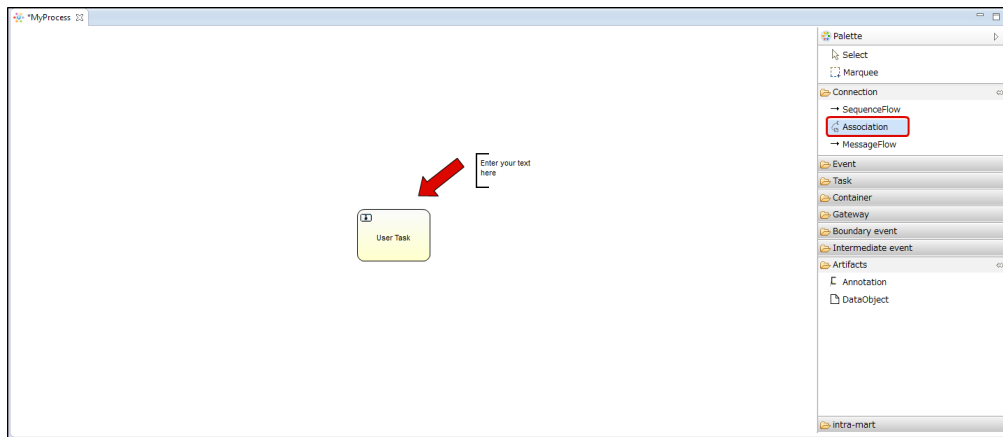


2. エディタ上の注釈を配置したい場所をクリックします。

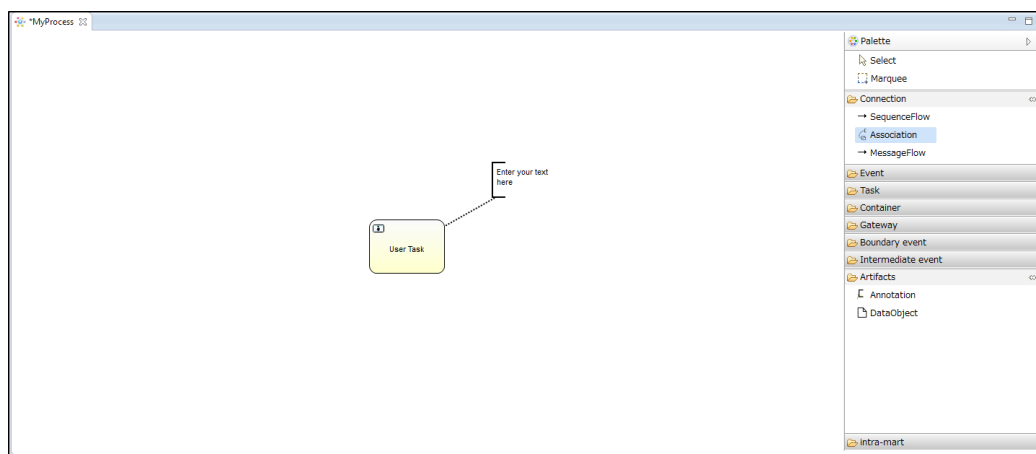


3. 配置した注釈と注釈を行いたいアクティビティをアソシエーションで接続します。

- PaletteのConnectionにあるAssociationをクリックします。
- エディタに配置した注釈をクリックし、注釈を定義したい接続先のアクティビティをクリックします。

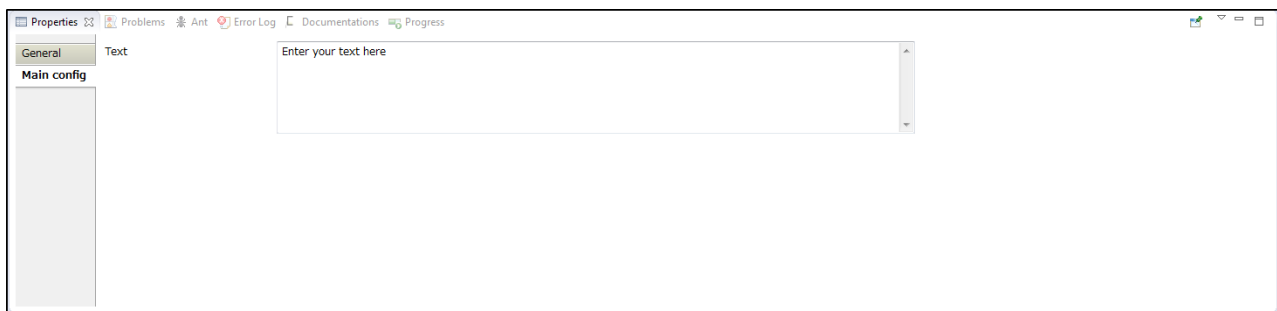


4. これで注釈が定義されました。



プロパティの設定

注釈のメインコンフィグの設定について解説します。



項目名	説明
Text(説明)	注釈の内容を記載します。

データオブジェクト

プロセス定義中にあるタスク、サブプロセス、コールアクティビティ（この後、これらを総称してアクティビティとします）と関連する成果物を示すアーティファクトです。

データオブジェクトとアクティビティをアソシエーションで紐付けすると、破線の矢印で関連の様子を示します。

データオブジェクトからアソシエーションで紐付けした際の矢印の方向は以下のような意味を持ちます

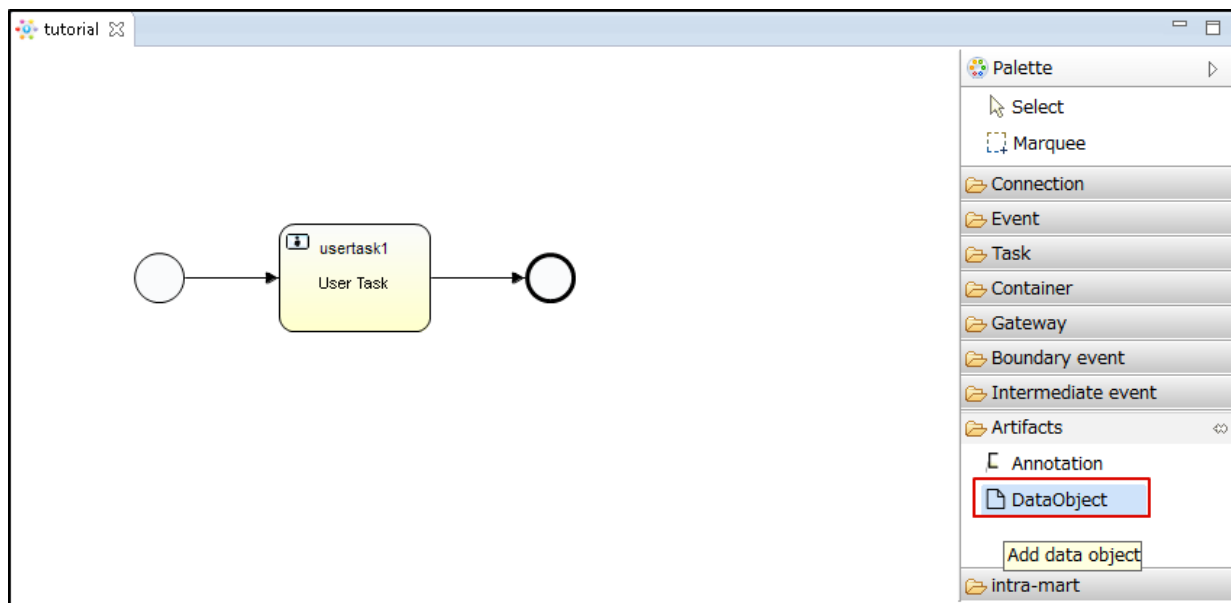
- データオブジェクトからアクティビティへと**Association**を引く
アクティビティで定義されたタスクを実行する際にあらかじめ必要となる成果物を示します。
- アクティビティからデータオブジェクトへと**Association**を引く

アクティビティを遂行した結果、生成される成果物を示します。

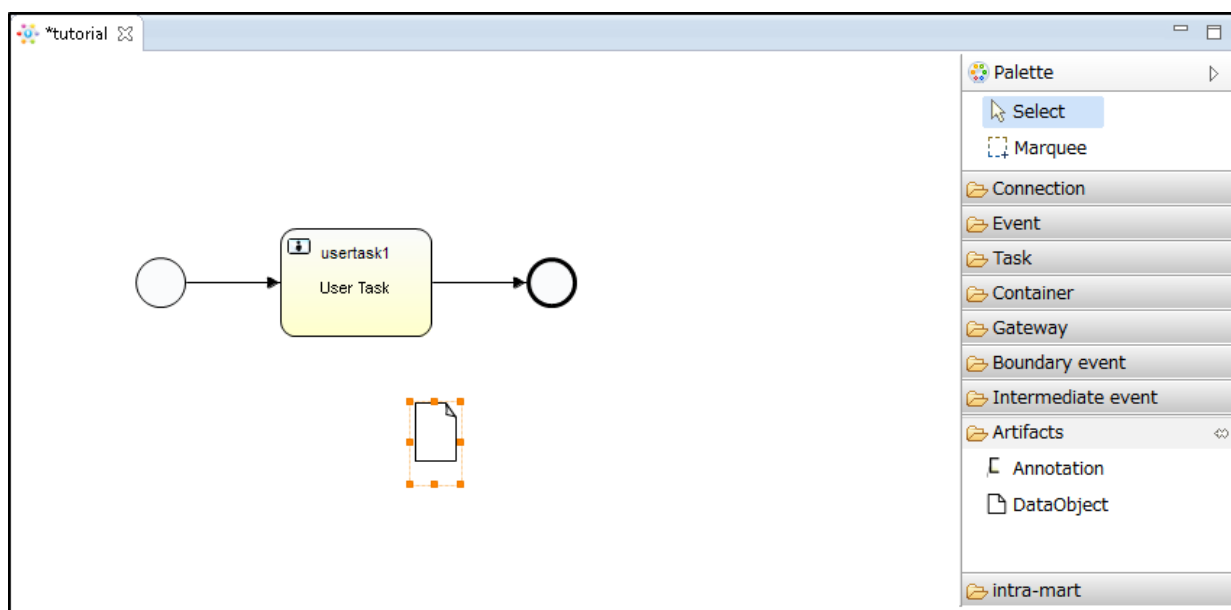
データオブジェクトの配置

例としてチュートリアルで作成したプロセス定義中のユーザタスクで「見積書」という成果物が生成されることを表現する方法について解説します。

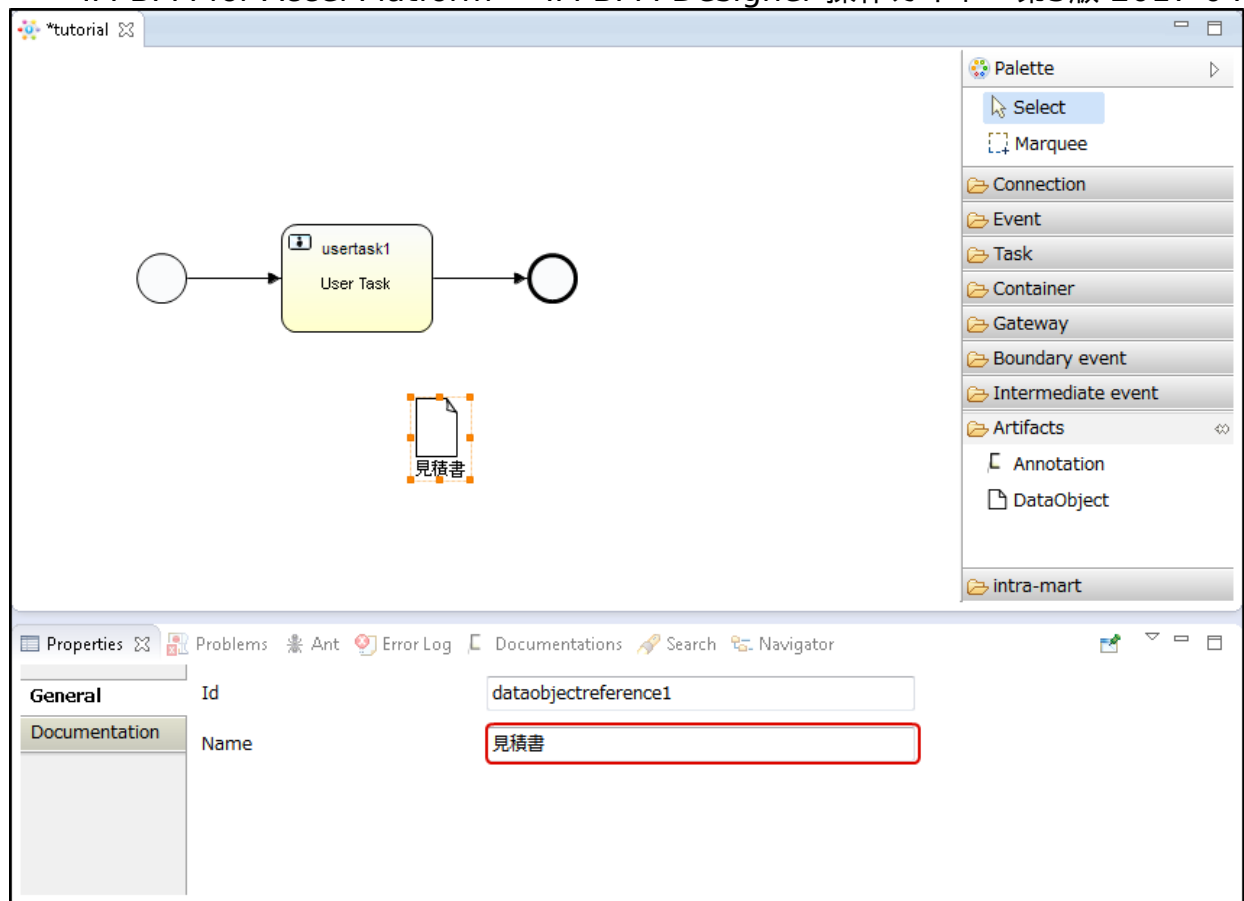
1. PaletteのArtifactsにあるDataObjectをクリックします。



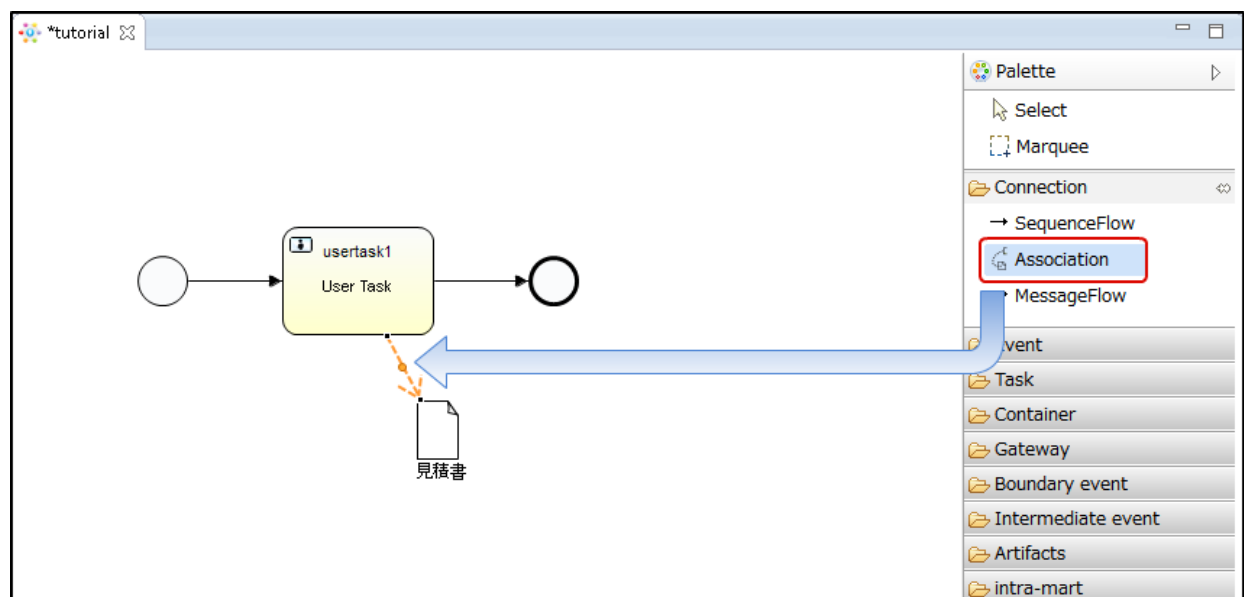
2. エディタ上のデータオブジェクトを配置したい場所をクリックします。



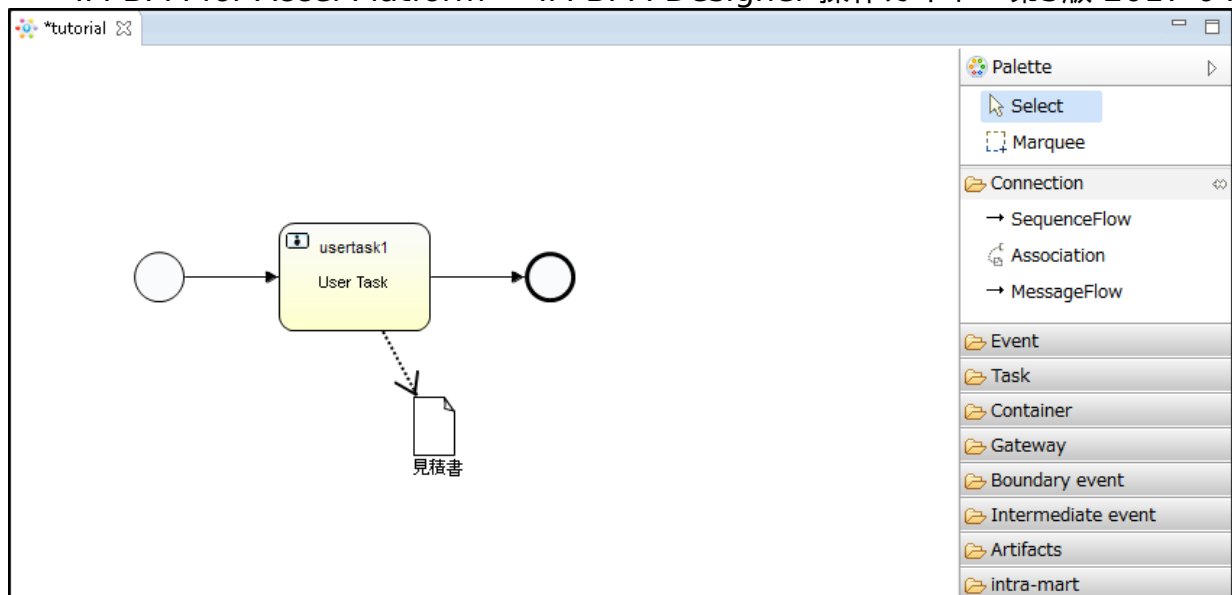
3. 2. で配置したデータオブジェクトが「見積書」であることがわかるように名前を設定します。
プロパティ・ビューの一般の名前に対して、「見積書」と設定します。



4. フロー中のユーザタスクから配置したデータオブジェクトに対してアソシエーションを設定します。
 PaletteのConnectionからAssociationを選択した後、ユーザタスクからデータオブジェクトに対して設定すると、ユーザタスクからデータオブジェクトへの破線の矢印が引かれます。

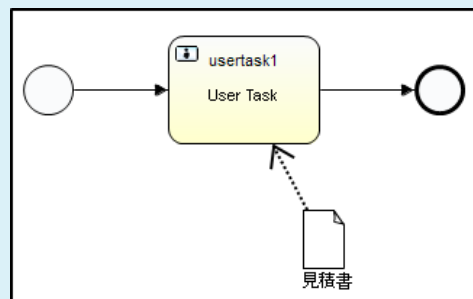


5. これで、ユーザタスクから「見積書」という成果物が生成されることをデータオブジェクトで表現できました。



i コラム

4. の操作において、アソシエーションをデータオブジェクトからユーザタスクへと引くように設定すると、アソシエーションの矢印の方向が反対になり、ユーザタスクを実行する際に「見積書」という成果物が必要であることを表現できます。



intra-mart

intra-martで独自拡張したタスクです。

ここに配置されているタスクの一部は、intra-martのアプリケーションと連動したタスクを設定することができます。

対象となるタスクは、以下の3つです。

- IM-LogicDesigner
 - [IMLogicDesignerTask](#)
- IM-Workflow、IM-FormaDesigner for Accel Platform、IM-BIS for Accel Platform
 - [ApplyTask](#)
 - [DraftTask](#)

i コラム

上記の各タスクを利用する場合、各アプリケーションの基本知識を習得していることを前提とします。

- [IM-LogicDesigner仕様書](#)
- [IM-Workflow 仕様書](#)
- [IM-FormaDesigner 仕様書](#)
- [IM-BIS 仕様書](#)

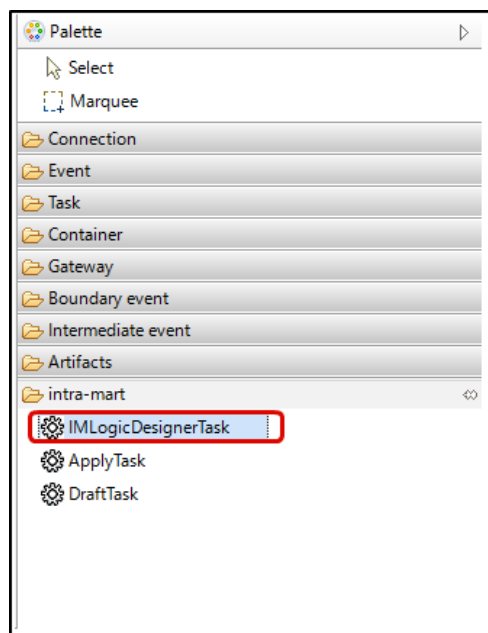
IMLogicDesignerTask(IM-LogicDesignerタスク)

IM-LogicDesignerの機能を呼び出すタスクです。

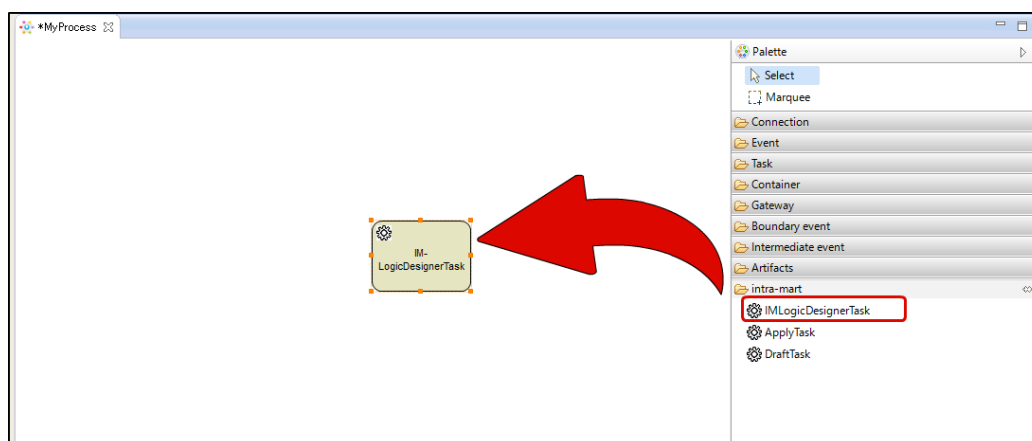
IM-LogicDesignerタスクに到達すると、タスク中に設定された情報からIM-LogicDesignerで定義されたロジックフローの処理を行います。

IM-LogicDesignerタスクはロジックフローが実行完了になった段階でタスクも終了となり、次のフローエレメントへと遷移します。

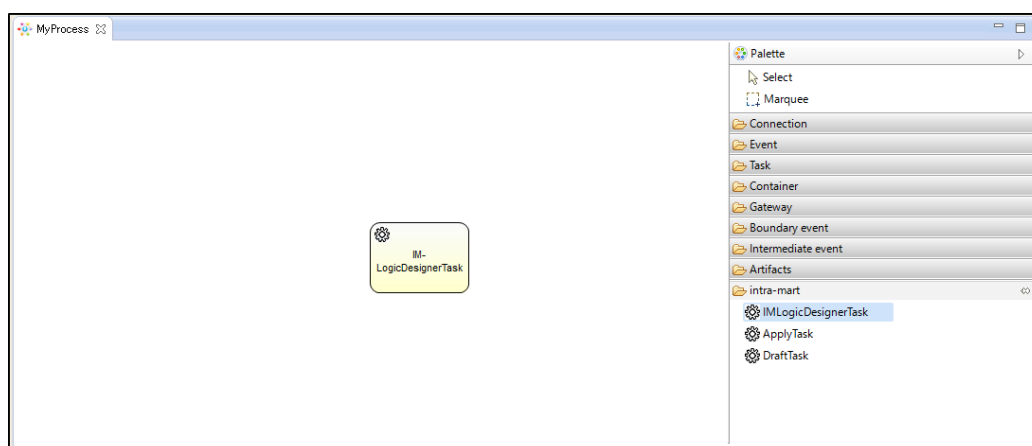
1. IM-BPM Designerの画面右側Paletteのintra-martにあるIMLogicDesignerTaskをクリックします。



2. エディタ上のIM-LogicDesignerタスクを定義したい場所をクリックします。



3. これでIM-LogicDesignerタスクが定義されました。



プロパティの設定

IM-LogicDesignerタスクのメインコンフィグの設定について解説します。

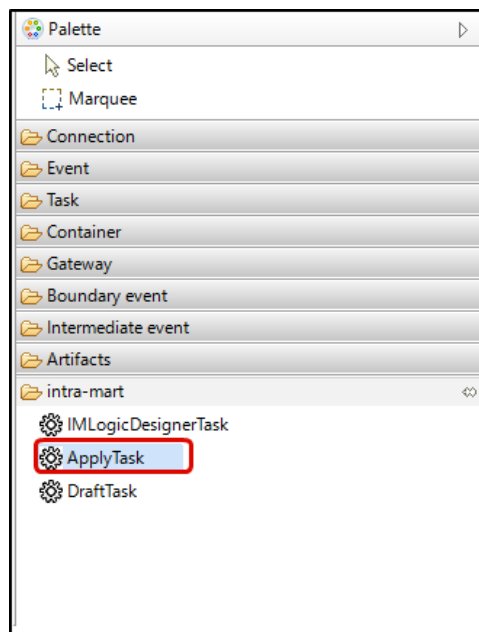
項目名	説明
Flow id(フローID)	IM-LogicDesignerのロジックフローIDを指定します。
Version(バージョン)	ロジックフローのバージョンを指定します。 「最新バージョンの利用」にチェックを入れた状態でバージョンも設定すると、バージョンで設定された情報が優先されます。 バージョンに指定が無く「最新バージョンの利用」もチェックが入っていない場合、デプロイエラーが発生します。
Use latest version(最新バージョンの利用)	最新のバージョンを使用するかの設定です。
Input data(入力データ)	ロジックフローに渡す入力値の設定です。
Store result variable(結果変数を格納する)	ロジックフローの実行結果をプロセスフローに返却するかどうかを設定します。
Result variable(結果変数名)	「結果変数を格納する」がONで「結果変数名」に入力が無い場合 ロジックフローの実行結果で返却された変数をそのままプロセス変数に格納します。 「結果変数を格納する」がONで「結果変数名」に入力がある場合 結果変数名に設定された変数名のマップに、ロジックフローの実行結果で返却された変数を格納します。 「結果変数を格納する」がOFFの場合 ここで入力された値は扱われません。

ApplyTask(申請タスク)

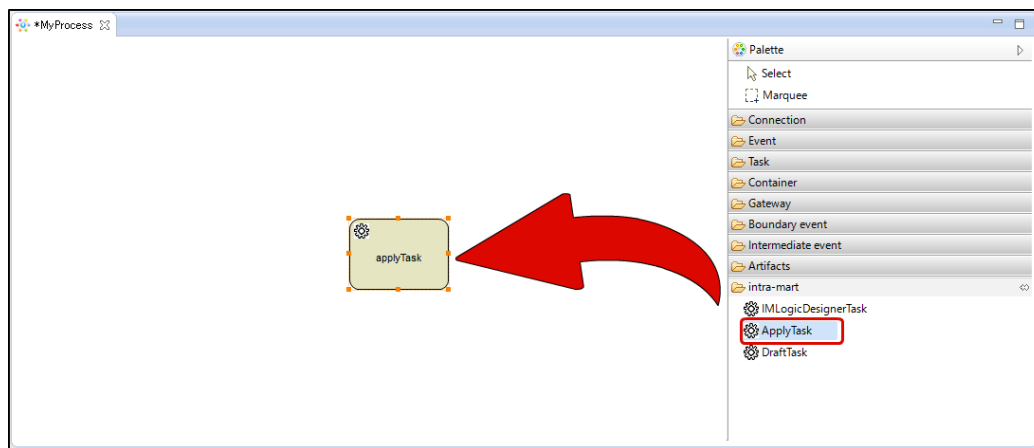
IM-Workflow、IM-FormaDesigner、IM-BISいずれかのフローを申請するタスクです。

申請タスクの配置

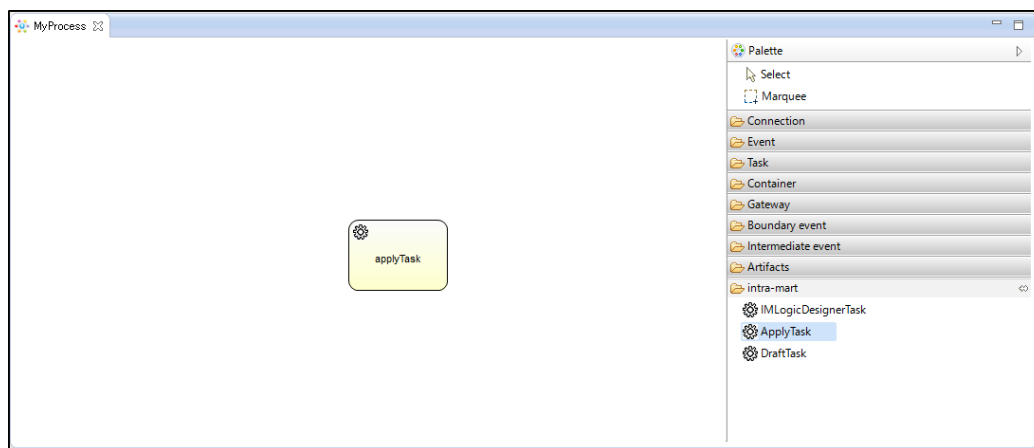
1. IM-BPM Designerの画面右側Paletteのintra-martにあるApplyTaskをクリックします。



2. エディタ上の申請タスクを定義したい場所をクリックします。



3. これで申請タスクが定義されました。



申請タスクの設定

申請タスクのメインコンフィグの設定について解説します。

Properties Problems Ant Error Log Documentations Search Navigator

基本情報 アプリケーション IM-Workflow

メインコンフィグ 前処理ユーザ...ラムのクラス名 クラスの選択

説明

リスナ フローID (?)

表現 ユーザーデータID (?)

案件番号 (?)

案件名 (?)

申請画面目 (?)

申請実行者コード (?)

申請権限者コード (?)

権限者会社コード (?)

権限者組織セットコード (?)

権限者組織コード (?)

優先度 (?)

起票コメント (?)

タスク終了時の起票案件の操作 terminate

☐ 参照可能な変数を継承する

入力データ:

名前	値

案件の処理結果を格納する変数名

☐ 案件のユーザデータを格納する

案件のユーザデ...格納する変数名

新規 編集 削除

項目名	説明
application(アプリケーション)	申請タスクで利用するアプリケーションを選択します。 選択できるアプリケーションは以下の3つです。 - IM-Workflow → Forma利用なしのワークフロー - IM-FormaDesigner → コンテンツ定義にIM-FormaDesignerを利用したワークフロー - IM-BIS → IM-BISのワークフロー
preprocess class(前処理のクラス名)	より詳細な設定を行いたい場合にjavaファイルで設定が行えます。 利用するアプリケーションによって実装するインタフェースが異なります。 - IM-Workflow - jp.co.intra_mart.activiti.engine.delegate.ImWorkflowApplyPreprocess - IM-FormaDesigner - jp.co.intra_mart.activiti.engine.delegate.ImFormaApplyPreprocess - IM-BIS - jp.co.intra_mart.activiti.engine.delegate.ImBisApplyPreprocess
flow id(フローID)	フローIDを設定します。
User data id(ユーザデータID)	アプリケーションに紐づくユニークなkeyを設定します。
mutter no.(案件番号)	案件番号を設定します。
mutter name(案件名)	案件名を設定します。
apply base date(申請基準日)	案件の申請基準日を設定します。 ISO8601フォーマットで指定します。（例）2017-04-01 指定しなかった場合は、現在日が設定されます。
apply execute user(申請実行者コード)	申請実行者のコードを設定します。
apply auth user(実行権限者コード)	申請権限者のコードを設定します。
apply auth company(権限者会社コード)	権限会社のコードを設定します。
apply organization set(権限者組織セットコード)	権限者組織のセットコードを設定します。
apply organization(権限者組織コード)	権限者組織のコードを設定します。
Priority(優先度)	案件の優先度を設定します。
process comment(起票コメント)	起票コメントを設定します。
manipulating of matter cancellation (タスク終了時の起票案件の操作)	タスクが終了した際に、起票案件の処理を継続について選択します。 terminateを選択した場合、タスク終了と同時に起票案件の処理も終了します。 continueを選択した場合、タスクが終了しても起票案件の処理は継続します。
inherit variables(参照可能な変数を継承する)	呼び出し元プロセスの変数を申請時に入力データとしてすべて引き継ぐかを選択します。
Input data(入力データ)	申請時にフローに渡す入力データを定義します。
Variable to store matter process result (案件の処理結果を格納する変数名)	設定された変数名で案件の処理結果の情報が格納されます。
Store matter user data (案件のユーザデータを格納する)	案件内のユーザデータをBPMのプロセスフローに返却するかどうかを設定します。 IM-Workflowの場合は、案件プロパティが返却されます。 IM-FormaDesignerやIM-BISの場合は、アプリケーションデータが返却されます。
Variable to store matter user data (案件のユーザデータを格納する変数名)	「案件のユーザデータを格納する」がONで「案件のユーザデータを格納する変数名」に入力が無い場合 案件内のユーザデータをそのままプロセス変数に格納します。 「案件のユーザデータを格納する」がONで「案件のユーザデータを格納する変数名」に入力がある場合 案件のユーザデータを格納する変数名に設定された変数名のマップに案件内のユーザデータを格納します。 「案件のユーザデータを格納する」がOFFの場合 ここで入力された値は扱われません。

- ワークフローの終了結果によって「案件の処理結果を格納する変数名」に保存される値が大きく変わります。

1. 通常に案件が終了した場合

キー	データ型	補足
locale	Object	
processDate	String	
systemMatterId	String	
userDataId	String	
contentsId	String	
contentsVersionId	String	
routeId	String	
routeVersionId	String	
flowId	String	
flowVersionId	String	
actFlag	String	
lastAuthUserCd	String	
lastExecUserCd	String	
lastNodeType	String	
lastProcessNodeId	String	
lastNodeName	String	
lastResultStatus	String	discontinue、approveend、mattercomplete、deny、matterhandleのいずれかが返却されます。

2. 案件削除でワークフロー自体が削除された場合

キー	データ型	補足
locale	Object	
systemMatterId	String	
userDataId	String	
lastResultStatus	String	activedeleteが返却されます。

3. BPM側で終了した場合

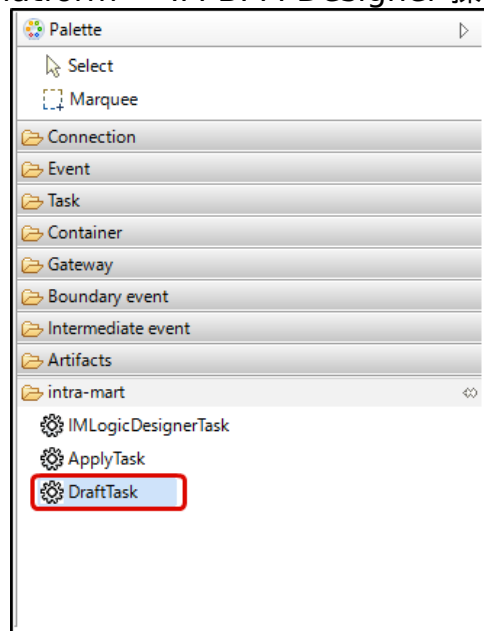
キー	データ型	補足
systemMatterId	String	
userDataId	String	
lastResultStatus	String	null値が返却されます。

DraftTask(起票タスク)

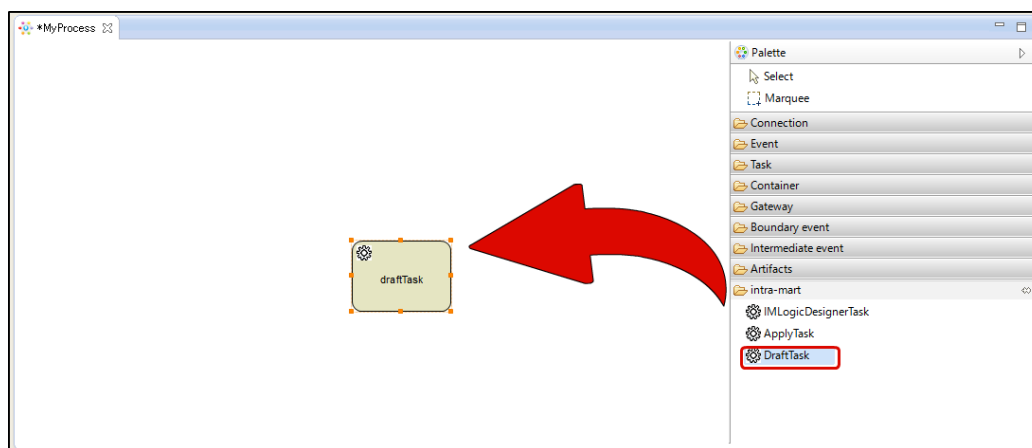
- IM-Workflow、IM-FormaDesigner、IM-BISいずれかのフローを起票するタスクです。

起票タスクの配置

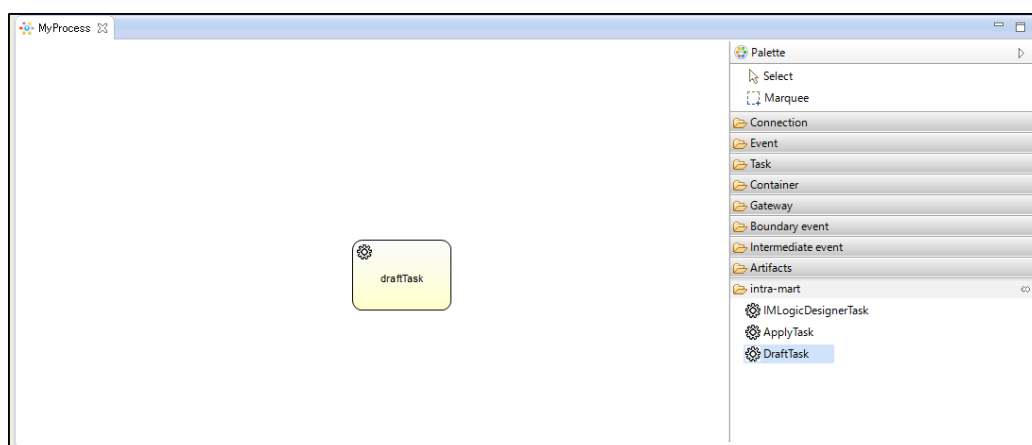
- IM-BPM Designerの画面右側Paletteのintra-martにあるDraftTaskをクリックします。



2. エディタ上の起票タスクを定義したい場所をクリックします。

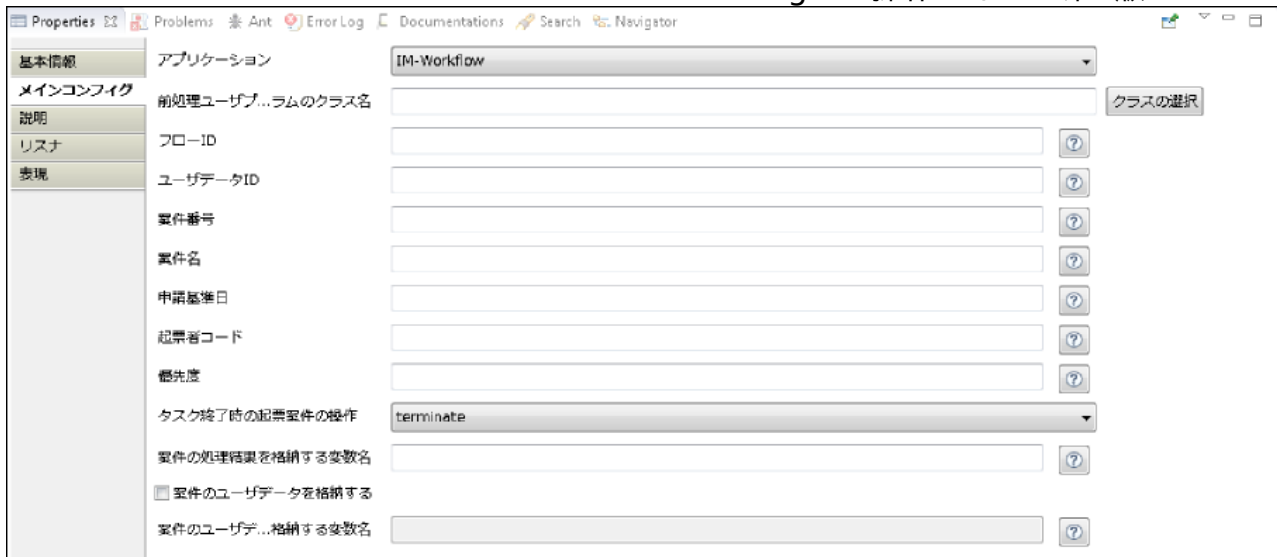


3. これで起票タスクが定義されました。



起票タスクの設定

起票タスクのメインコンフィグの設定について解説します。



項目名	説明
application(アプリケーション)	起票タスクで利用するアプリケーションを選択します。 選択できるアプリケーションは以下の3つです。 - IM-Workflow → Forma利用なしのワークフロー - IM-FormaDesigner → コンテンツ定義にIM-FormaDesignerを利用したワークフロー - IM-BIS → IM-BISのワークフロー
preprocess class(前処理のクラス名)	より詳細な設定を行いたい場合にjavaファイルで設定が行えます。 - IM-Workflow - jp.co.intra_mart.activiti.engine.delegate.ImWorkflowDraftPreprocess - IM-FormaDesigner - jp.co.intra_mart.activiti.engine.delegate.ImFormaDraftPreprocess - IM-BIS - jp.co.intra_mart.activiti.engine.delegate.ImBisDraftPreprocess
flow id(フローID)	フローIDを設定します。
User data id(ユーザデータID)	フローに紐づくユニークなkeyを設定します。
mutter no.(案件番号)	案件番号を設定します。
mutter name(案件名)	案件名を設定します。
apply base date(申請基準日)	案件の申請基準日を設定します。 ISO8601のフォーマットで指定します。(例) 2017-04-01 指定しなかった場合は現在日が設定されます。
draft user code(起票者コード)	起票者コードを設定します。
案件のPriority(優先度)	優先度を設定します。
manipulating of matter cancellation (タスク終了時の起票案件の操作)	タスクが終了した際に、起票案件の処理を継続について選択します。 terminateを選択した場合、タスク終了と同時に起票案件の処理も終了します。 continueを選択した場合、タスクが終了しても起票案件の処理は継続します。
Variable to store matter process result (案件の処理結果を格納する変数名)	設定された変数名で案件の処理結果の情報が格納されます。
Store matter user data (案件のユーザデータを格納する)	案件内のユーザデータをBPMのプロセスフローに返却するかどうかを設定します。 IM-Workflowの場合は、案件プロパティが返却されます。 IM-FormaDesignerやIM-BISの場合は、アプリケーションデータが返却されます。

項目名	説明
Variable to store matter user data (案件のユーザデータを格納する変数名)	「案件のユーザデータを格納する」がONで「案件のユーザデータを格納する変数名」に 入力が無い場合 案件内のユーザデータをそのままプロセス変数に格納します。 「案件のユーザデータを格納する」がONで「案件のユーザデータを格納する変数名」に 入力がある場合 案件のユーザデータを格納する変数名に設定された変数名のマップに案件内のユーザ データを格納します。 「案件のユーザデータを格納する」がOFFの場合 ここで入力された値は扱われません。

コラム

起票タスクの案件のユーザデータを格納するアプリケーションがIM-FormaDesigner/IM-BISの場合、結果変数に返却されるFormaアイテムデータは、ワークフローの申請時に作成されます。

その為、申請処理を行わずに取り下げを行った場合は、Formaアイテムデータは返却されません。

- ワークフローの終了結果によって「案件の処理結果を格納する変数名」に保存される値が大きく変わります。

1. 通常に案件が終了した場合

キー	データ型	補足
locale	Object	
processDate	String	
systemMatterId	String	
userDataId	String	
contentsId	String	
contentsVersionId	String	
routeId	String	
routeVersionId	String	
flowId	String	
flowVersionId	String	
actFlag	String	
lastAuthUserCd	String	
lastExecUserCd	String	
lastNodeType	String	
lastProcessNodeId	String	
lastNodeName	String	
lastResultStatus	String	discontinue、approveend、mattercomplete、deny、matterhandleのいずれかが返却されます。

2. 案件削除でワークフロー自体が削除された場合

キー	データ型	補足
locale	Object	
systemMatterId	String	
userDataId	String	
lastResultStatus	String	activedeleteが返却されます。

3. BPM側で終了した場合

キー	データ型	補足
----	------	----

systemMatterId	String	
userDataId	String	
lastResultStatus	String	null値が返却されます。

i コラム

これから説明する内容において使用される単語は以下の総称として説明します。

- フローエレメント：エディタ上で配置可能な全図形の総称
- アクティビティ：フローエレメントのうち各種タスク、および各種サブプロセスの総称。

! 注意

リンク先で設定している各コンポーネントの「プロパティの設定」は、プロパティ・ビューの「メインコンフィグ」タブに対する設定です。

そのほかのタブの設定情報については「[共通の項目](#)」を参照してください。

共通の項目

本項では、フローエレメントで共通となるプロパティ・ビューの設定項目について解説します。

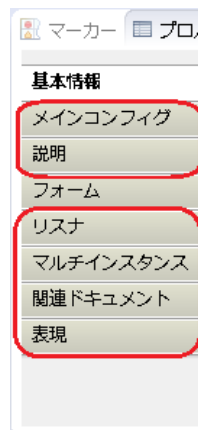
各フローエレメントにおいて共通のプロパティ設定は以下の5つです。

これらは複数のフローエレメントにまたがって個別に設定可能な項目です。

- [General](#)（基本情報）
- [Documentation](#)（説明）
- [Listeners](#)（リスナ）
- [Multi instance](#)（マルチインスタンス）
- [Relation document](#)（関連ドキュメント）
- [Appearance](#)（表現）

それぞれエディタ上にあるフローエレメントにフォーカスを当てたときに、画面下部のプロパティにあるタブで選択する事で設定を行う事ができます。

ただし、一部フローエレメントでは設定できない項目も存在します。



General（一般）の設定

ここでは各フローエレメントで行う一般の設定について解説します。

一般は、エディタ上の設定を行いたいフローエレメントをクリックする事で画面下部のプロパティに編集画面が表示されます。

フローエレメントには、それぞれプロパティで様々な設定が定義できます。

一般の設定は全てのフローエレメントで可能です。



コラム

フローエレメントによって設定の項目数は異なります。

項目名	説明
Id	フローエレメントを一意に特定するための設定です。 英数字を推奨します。
Name（名前）	フローエレメントの名前を設定します。 プロセス定義で表示されます。
Default flow（デフォルトフロー）	同じフローエレメントから複数のシーケンスフローが接続されている場合に表示されます。 他のシーケンスフローの条件式が全て偽の場合に進行するシーケンスフローの設定です。 「通常」の場合という意味ではなく、「それ以外」の場合を表します。
Other name（別名）	フローエレメントの別名の設定です。
Asynchronous（非同期）	非同期実行の設定です。
Exclusive（排他）	排他制御の設定です。

Documentation（説明）

フローエレメントについての説明を記載できます。

（Export to Excel）エクスポートを行った際にExcelファイルに設計書として出力されます。

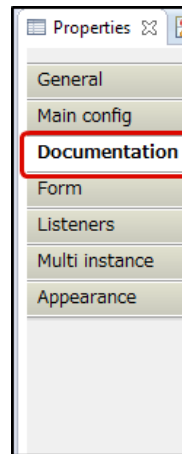
以下の4つのフローエレメントのみドキュメンテーションの設定ができません。

- Pool
- Lane
- MessageFlow
- Association

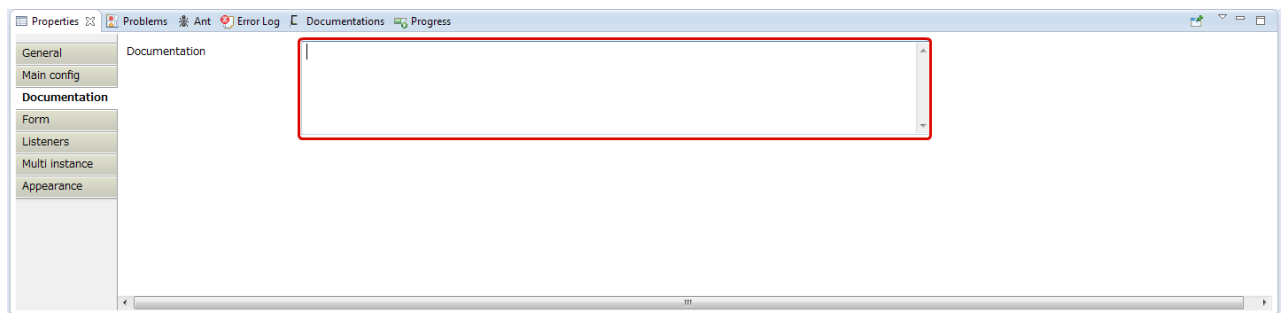
Documentation（説明）の編集

- ここでは、ユーザタスクの説明を編集します。

1. 編集を行うフローエレメントを選択し、画面下部のProperties・ビューにあるDocumentationをクリックします。

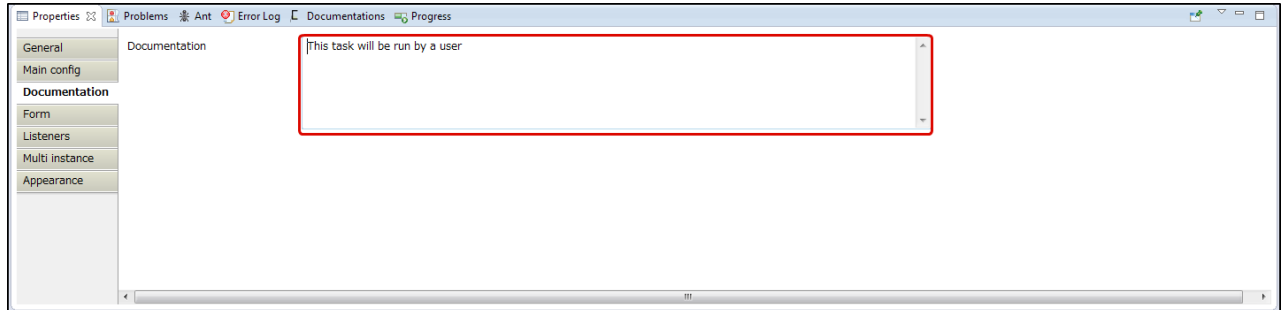


2. 説明にフローエレメントの説明を入力します。



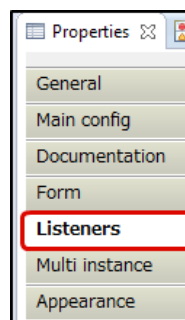
3. これでドキュメンテーションの編集が完了しました。

ここでは「This task will be run by a user」と入力しています。

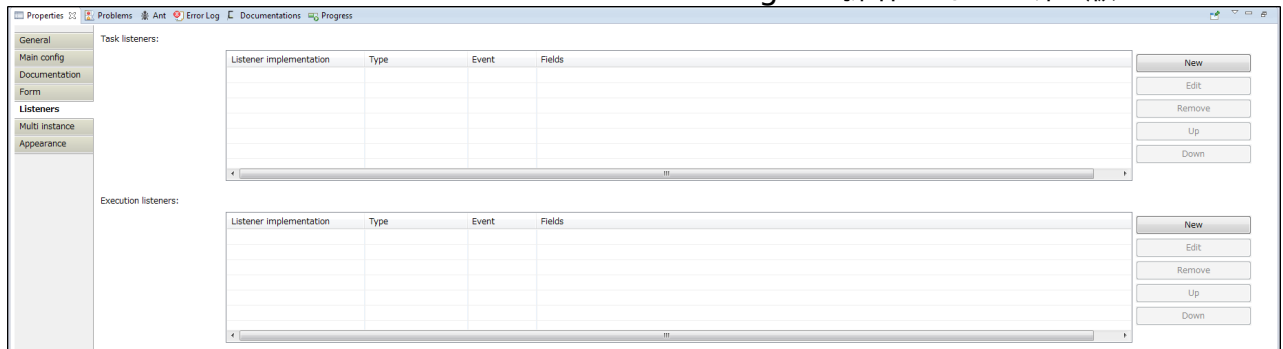


Listeners（リスナ）

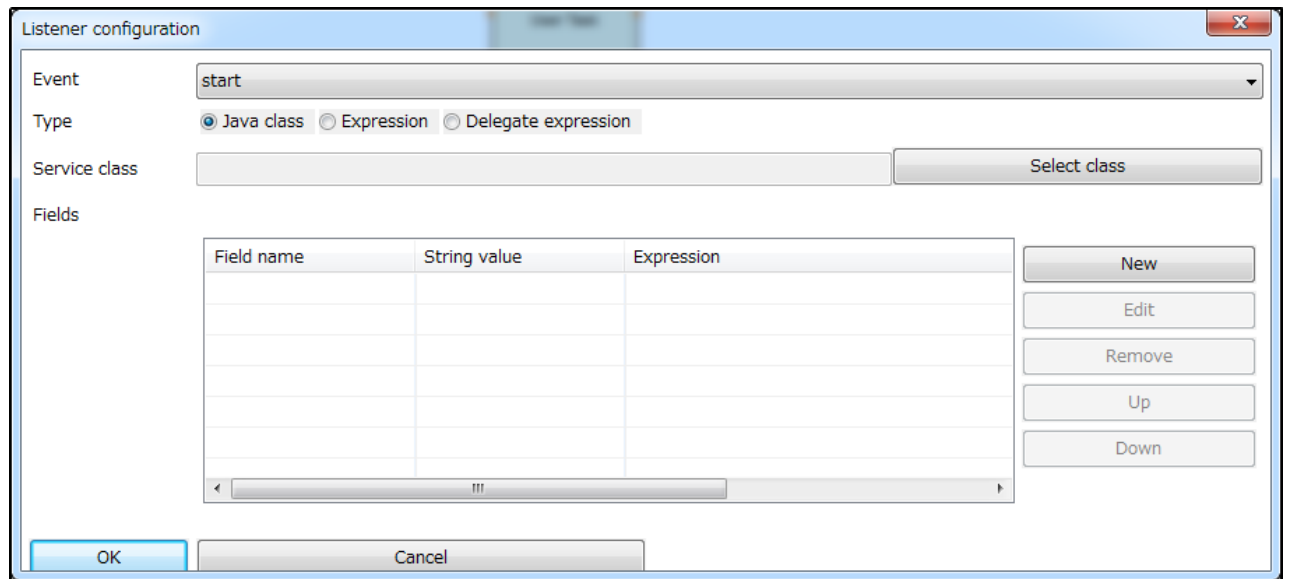
1. 設定を行うフローエレメントを選択し、画面下部のPropertiesにあるListenersをクリックします。



2. リスナを使用するとjavaの実行や、プロセス定義実行中のイベントの評価などが行えます。



- 「New」ボタンで新しいリスナの登録が行えます。
- 「Edit」ボタンで、既にあるリスナの編集が行えます。
- 「Remove」ボタンで既にあるリスナの削除が行えます。
- 「Up」ボタンと「Down」ボタンでリスナの並び替えが行えます。



項目名	説明
Event (イベント)	イベントの種類を選択します。
Task type (タスクタイプ)	呼び出すタスクのタイプを選択します。 Java class・Expression・Delegate expressionの3種類があります。
Class name (クラス名)	呼び出すJava class名を選択します。
Fields (フィールド)	java classのフィールドに値をセットする設定です。

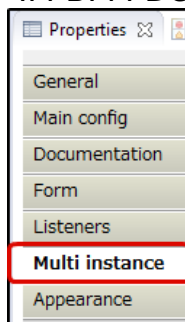
Multi instance (マルチインスタンス)

マルチインスタンスを設定したフローエレメントは複数回実行されます。

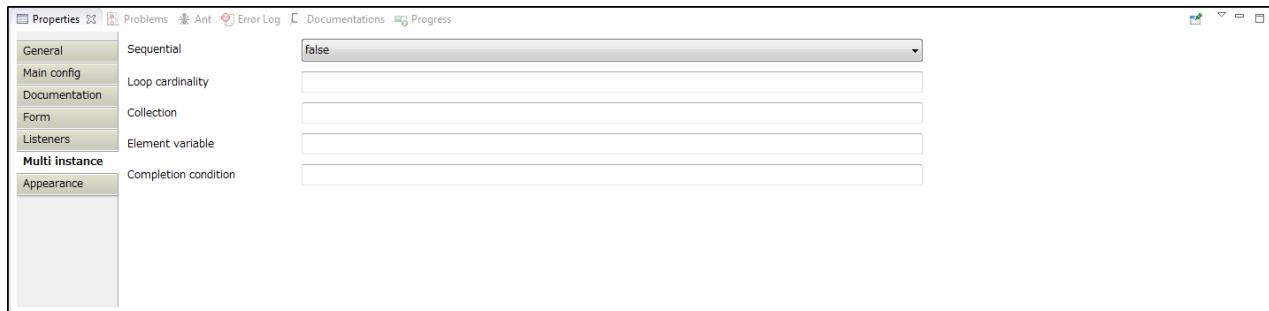
ゲートウェイやイベントでは定義する事ができません。

実行方法として順次実行と並行実行があります。

- 設定を行うフローエレメントを選択し、画面下部のPropertiesにあるMulti instanceをクリックします。

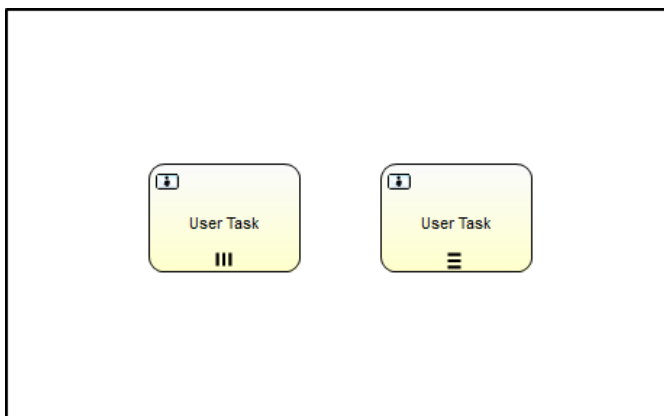


2. この画面では実行の方法や終了条件などMulti instanceの設定が行えます。



項目名	説明
Sequential (連続性)	順次実行か並列実行かを設定します。
Loop Cardinality (ループ回数)	ループ回数を設定します。
Collection (配列)	配列の変数を指定します。 指定された配列の要素数分マルチインスタンスが実行されます。
Element variable (要素変数)	サービスタスク等でRuntime側で実行する際に、実行時に持つ変数を定義できます。 ここで定義した変数はjp.co.intra_mart.activiti.engine.impl.pvm.delegate.ActivityBehaviorの実装時に利用することができます。
Completion (終了条件)	終了条件の式を設定します。 ループ毎に式の評価を行い、評価が真の場合残りのインスタンスを破棄しプロセス定義の実行を続けます。

- 設定を行うとプロセス定義上で、フローエレメントの下部に三本線が表示されます。（左:並列実行、右：順次実行）



ここでは、[UserTask](#)のマルチインスタンスに以下の設定を行いました。

- 並列実行の設定例

項目名	設定
Sequential (連続性)	false

項目名	設定
-----	----

Loop Cardinality (ループ回数)	10
--------------------------	----

- 順次実行の設定例

項目名	設定
-----	----

Sequential (連続性)	true
------------------	------

Loop Cardinality (ループ回数)	10
--------------------------	----

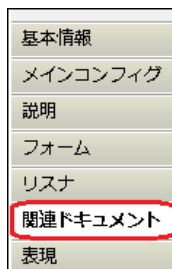
Relation document (関連ドキュメント)

プロセス、または特定のイベントやタスクに関連するドキュメントを設定することができます。
以下のフローエレメントに対して設定します。

- Process
- Pool
- StartEvent (ただし、SubProcess内に配置された StartEventは除く)
- UserTask
- ManualTask

関連ドキュメントは、ドキュメントファイルを指定する方法と外部のサイトを指定する方法があります。

1. 関連ドキュメントを設定できるフローエレメントを選択した状態で、画面下部のプロパティにある関連ドキュメントをクリックします。

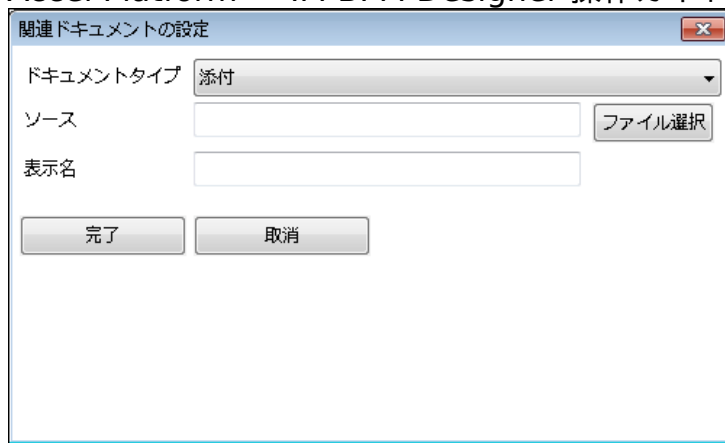


2. テーブルの右に配置されているボタンを押下し、ダイアログを表示して関連ドキュメントの設定を行います。



- 「新規」ボタンで新しい関連ドキュメントの登録が行えます。
- 「編集」ボタンで既にある関連ドキュメントの編集が行えます。
- 「削除」ボタンで既にある関連ドキュメントの削除が行えます。

ドキュメントの設定で表示されるダイアログでは、以下の内容を登録します。



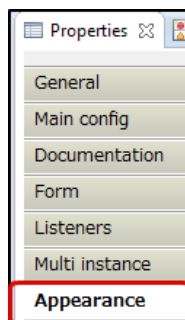
項目名	説明
Document type (ドキュメントタイプ)	関連ドキュメントの形式を設定します。 「添付」にした場合、関連ドキュメントをダウンロードして閲覧します。 「外部リンク」にした場合、閲覧時に指定したサイトを閲覧することができます。
Source (ソース)	関連ドキュメントのソースを指定します。 本項目は必須です。 「添付」の場合、BPMプロジェクトのソース・フォルダー「src/main/resources」にある「documents」パッケージの中にあるファイルを指定します。 テキストの右にある「ファイル選択」ボタンを押下すると、「documents」パッケージ中に配置されたファイルが選択できます。 「外部リンク」の場合、関連ドキュメントにあたる外部サイトのリンク先を入力します。
Label (表示名)	ブラウザ上で表示されるリンクの表示名を指定します。 未入力にした場合、「ソース」で指定した値がそのまま画面上に表示されます。

Appearance (表現)

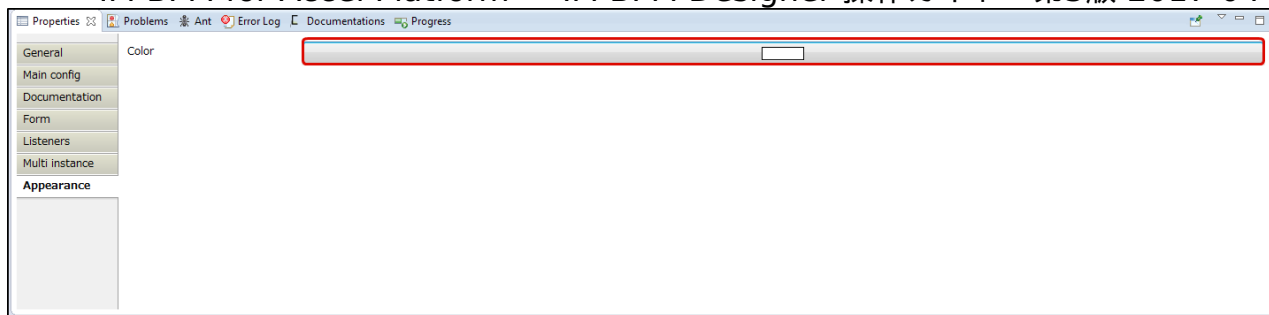
プロセス定義上での任意の色表現を設定します。

Appearanceの設定

1. 設定を行うフローエレメントを選択し、画面下部のPropertiesにあるAppearanceをクリックします。



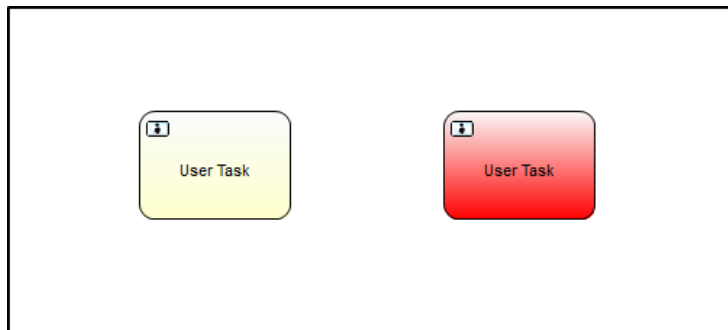
2. Colorをクリックし、パレットを表示させます。



- パレットから任意の色の選択・作成が行えます。
- 現在の色は、破線で囲まれている色です。



- 設定を行うとプロセス定義上で、フローエレメントの色が変更されます。（左:設定前、右:設定後）
- ここでは赤色を設定しています。



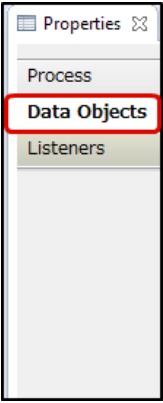
プロセス内の変数

Data Objects(データオブジェクト)

プロパティ・ビュー中にあるData Objectsタブでは、プロセス定義中で使用可能な変数を定義する事ができます。指定された変数情報はプロセス定義実行時に、指定した変数名に対して入力した値が指定されるようになります。

データオブジェクトの設定

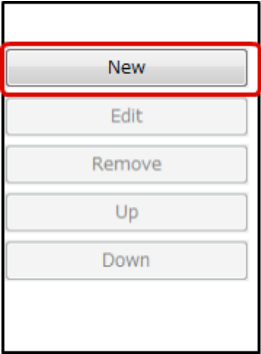
1. BPM Diagram Editorの空白部分をクリックして、Properties・ビューを表示します。
2. 画面下部のProperties・ビューにあるData Objectsタブを選択します。



- Newボタンで新しい変数の登録が行えます。
- Editボタンで、既にある変数の編集が行えます。
テーブル上で編集したい変数を選択した後、Editをクリックしてください。
- Removeボタンで既にある変数の削除が行えます。
テーブル上で削除したい変数を選択した後、Removeボタンをクリックしてください。
- UpとDownで変数の並び替えが行えます。
テーブル上で並び替えたい変数を選択した後、対象のボタンをクリックしてください。

ダイアログを用いた変数定義の作成

1. Newボタンをクリックします。



2. 表示された画面で各項目の入力を行います。

A dialog box titled 'Data property configuration' with a close button in the top right corner. It contains four input fields: 'Id' (text), 'Name' (text), 'Type' (a dropdown menu currently showing 'string'), and 'Value' (text). At the bottom are 'OK' and 'Cancel' buttons.

項目名	説明
Id	変数を一意に特定するためのIDを定義します。 必須項目です。
Name(名前)	変数名を定義します。 必須項目です

項目名	説明
Type(型)	変数の型を選択します。 選択可能な型の種類は、string、boolean、datetime、double、int、longの6種類です。
Value(値)	変数の値を入力します。

- ここでは下記の設定で変数を作成します。

項目名	入力内容
Id	DataObject1
Name	StringTest
Type	string
Value	StringValue

3. 入力を終わったらOKをクリックします。

4. これで変数が定義されました。

Data properties:			
Id	Name	Type	Value
DataObject1	StringTest	string	StringValue

設計書出力機能

本項では作成したBPMダイアグラムの設計書出力する機能について説明します。
設計書はプロセス定義内に設定した各フローエレメントの情報とプロセス図を含んでいます。
また本設計書はExcelをベースに作成されるので、生成後に必要に応じて編集を行うことができます。

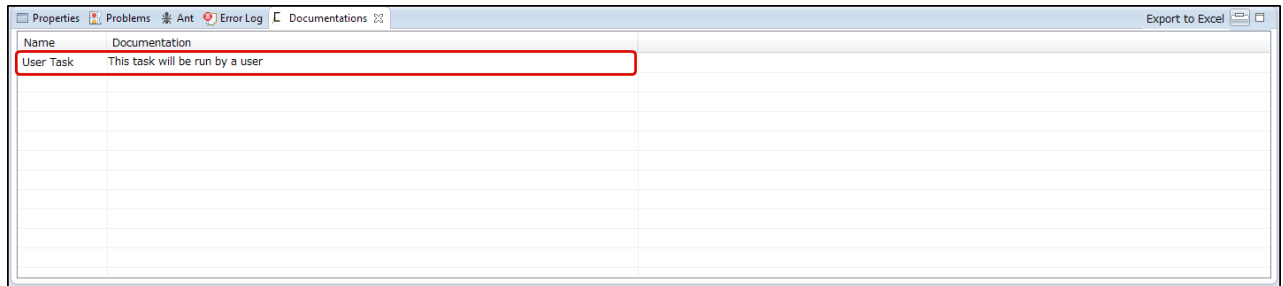
設計書出力の操作

各フローエレメントで編集したドキュメンテーションの内容を、設計書としてExcelファイルに出力することができます。

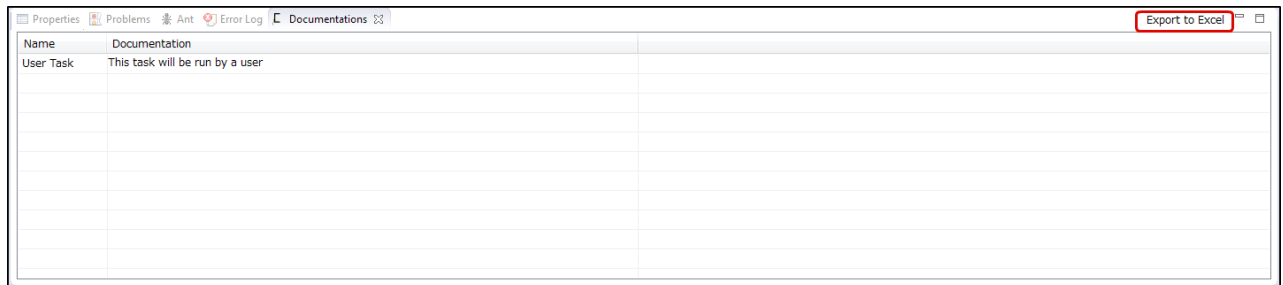
1. 画面下部からDocumentationsタブをクリックします。



2. 各フローエレメントで設定したドキュメンテーションがこの画面に表示されます。



3. Documentationsタブの右側にあるExport to Excelをクリックします。



4. 任意のディレクトリを指定して保存します。

出力される設計書の構成

出力された設計書は大きく分けて4つのシート構成でできています。

利用者はこの設計書をベースにして必要な情報の補填やカスタマイズをすることができます。

- 表紙

設計書の表紙です。

タイトルは固定文字列になっています。

- 改訂情報

設計書の改訂情報です。

改訂情報の日付は設計書の生成日になります。

版数は初版で固定です。

- アクティビティ一覧

各プロセスごとに定義されたフローエレメント情報を出力します。

もしプロセス定義ファイル中に複数のPoolが指定されている場合、各Poolごとに出力されます。

- BPMフロー図

プロセス定義ファイルで定義したプロセスフローの全体像を出力します。

デプロイ資材の作成

本項では、IM-BPM Designerで作成したBPMのプロセス定義ファイル（以下bpmnファイル）をintra-mart Accel Platform上にデプロイする方法について説明します。

プロセス定義のデプロイ方法については以下の2つの単位でデプロイすることが可能です。

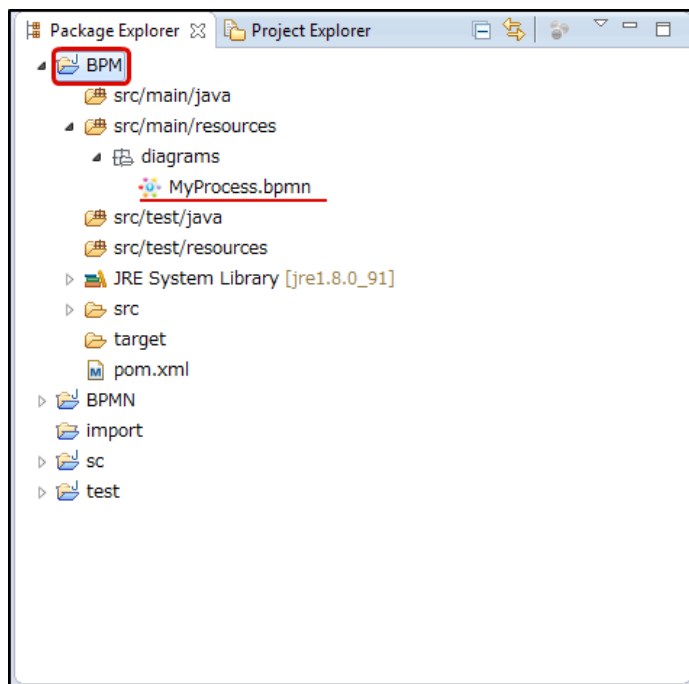
- bpmnファイル
- barファイル（アーカイブファイル）

実際にプロセス定義をデプロイする方法に関しては、「[IM-BPM ユーザ操作ガイド](#)」を参照してください。

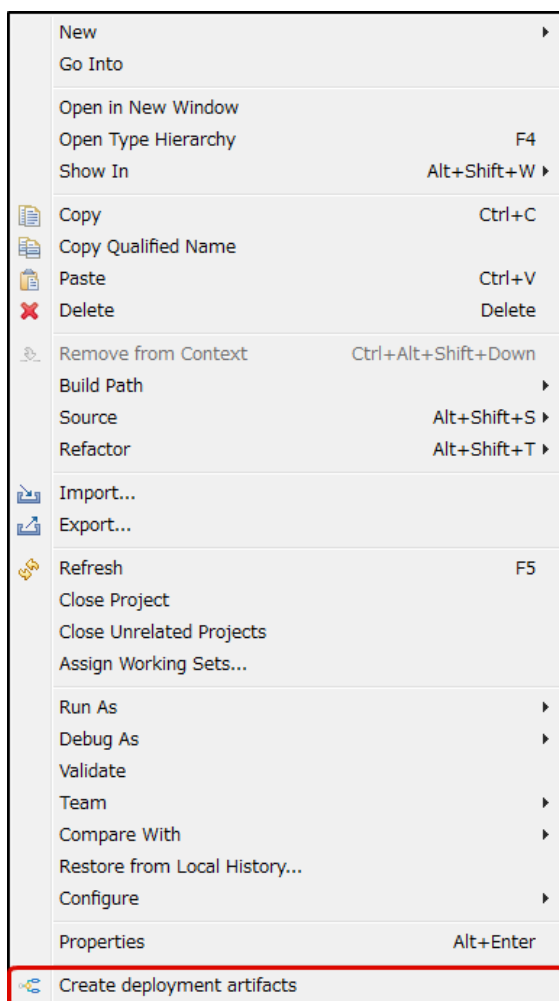
barファイルの作成方法

barファイルについては、以下の方法で作成を行います。

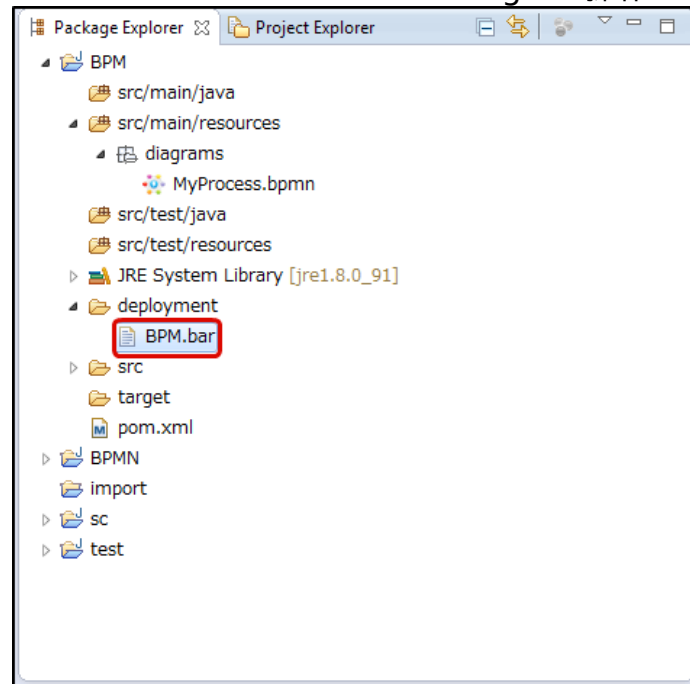
1. デプロイしたいbpmnファイルを含んだプロジェクトを選択して右クリックします。



1. 表示されたメニューバーから「Create development artifacts」を選択します。



1. プロジェクト上にdeploymentフォルダが生成され、その中にbarファイルが作成されます。



各デプロイ方法の特徴

■ barファイルでデプロイする利点

bpmnファイルでは特定のプロセス定義をデプロイできるのに対して、barファイルとしてコミットするとプロジェクト内のプロセス定義ファイルを一括してデプロイすることができます。

またbarファイルを作成した場合、同一プロジェクト中に存在するJavaクラスとJavaScriptファイルを含んだjarファイルと関連ドキュメントとして配置されたファイルが含まれます。

実際の本番環境にデプロイする場合、barファイルという単位で複数のプロセスの追加や更新と同時に、利用されるJavaクラスとJavaScriptファイル、関連ドキュメントの追加が一括でできます。

■ bpmnファイルでデプロイする利点

barファイルでデプロイするときに、含まれているプロセス定義のどれか1つでも設定に誤りがあった場合、エラーが発生してすべてのプロセス定義が以前の状態にロールバックされます。

そのためプロジェクト中に未完成のプロセス定義が含まれている状態で特定のプロセス定義のみデプロイを行いたいときは、デプロイ時にbpmnファイルを選択することで実現できます。

i コラム

IM-BPM Designer では、日時を **ISO8601拡張形式** で入力してください。

入力例

- 年月日 : 2016-12-01
- 日時(UTC) : 2016-12-01T12:34:56Z
- 日時(UTC+9) : 2016-12-01T12:34:56+09:00

