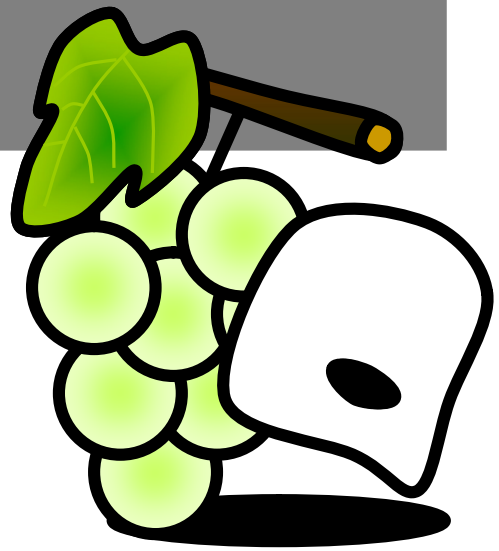
マスカットIDE Eclipse 版 ver 2.0.0
ユーザマニュアル



Rev. 1.0
2008/5/21

http://maskat.sourceforge.jp/

目次

1. インストールガイド

この章では、マスカット IDE Eclipse 版のインストールと環境設定の方法について説明します。

1.1 動作環境と前提条件

マスカットIDE Eclipse 版 (ver. 2.0.0) は以下の環境で動作します。

項目	前提条件	動作環境 (試験済み)
OS	Win32	Windows XP Professional SP2（x86, 32bit）
Java	JRE 1.4 以降	Sun JDK 1.4.2_15
Eclipse	Eclipse 3.2 以降	Eclipse 3.2.2
	GEF 3.2 以降	GEF 3.2.2

1.2 インストール手順

マスカット IDE Eclipse 版のインストールは以下の手順で行います。

1. Eclipse の公式サイト (http://www.eclipse.org/) から以下のファイルをダウンロードします。

Eclipse本体	Eclipse 3.2 SDK	http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.2.2-200702121330/eclipse-SDK-3.2.2-win32.zip
	Language Pack	http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/L-3.2.1_Language_Packs-200609210945/NLpack1-eclipse-platform-3.2.1-win32.zip
GEF	GEF Runtime 3.2.2	http://www.eclipse.org/downloads/download.php?file=/tools/gef/downloads/drops/R-3.2.2-200702081315/GEF-runtime-3.2.2.zip
	Language Pack	http://www.eclipse.org/downloads/download.php?file=/tools/gef/downloads/translations/NLpack1-GEF-runtime-3.2.zip

2. Eclipse SDK の ZIP ファイルを任意のフォルダ（例: C:¥eclipse）に展開します。
3. GEFの ZIP ファイルを解凍し、features と plugins フォルダを 2. のフォルダの配下に上書きコピーします。
4. Eclipse と GEF のランゲージパックの ZIP ファイルをそれぞれ解凍し、features と plugins フォルダを2. のフォルダの配下に上書きコピーします。
5. マスカットプロジェクトサイトから Eclipse版 マスカットIDE（maskat-ide-2.0.0.v20080521.zip）をダウンロードします。
6. 5.の ZIP ファイルを解凍し、featuresとpluginsフォルダを 2. のフォルダの配下にコピーします。

1.3 起動方法

1.2 節の手順 2 で解凍したフォルダにある eclipse.exe を 実行するとマスカットIDE が起動します。

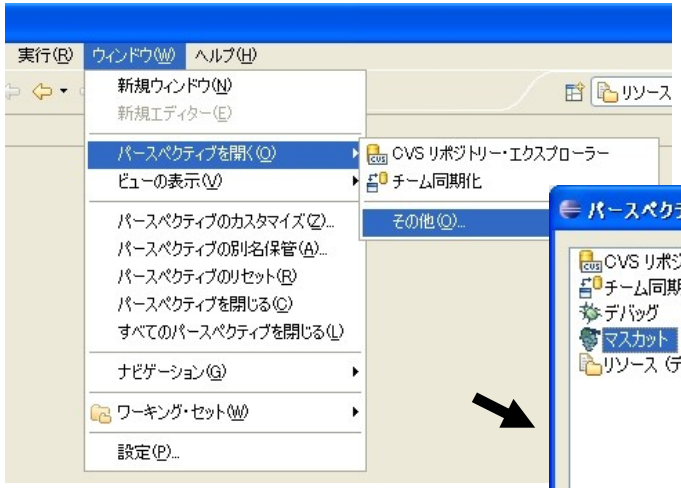
2. 画面構成と機能説明

この章では、マスカット IDE Eclipse 版の画面構成と各画面の
機能について簡単に紹介します。

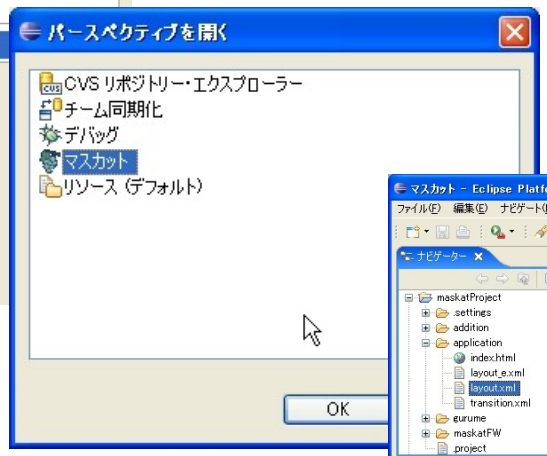
2.1 マスカットパースペクティブの開き方

手動でマスカットパースペクティブに切り替えるには以下の手順で行います。

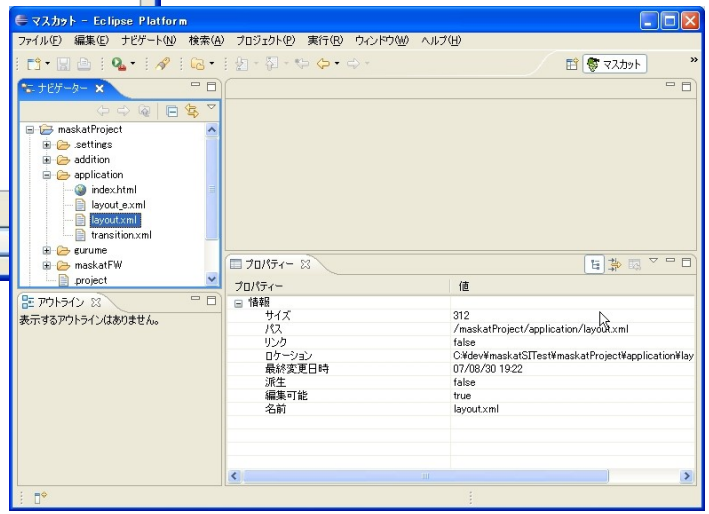
① [ファイル] メニューから [ウィンドウ] > [パースペクティブを開く] > [その他] を選択します。



② [パースペクティブを開く] ダイアログから [マスカット] を選択します。



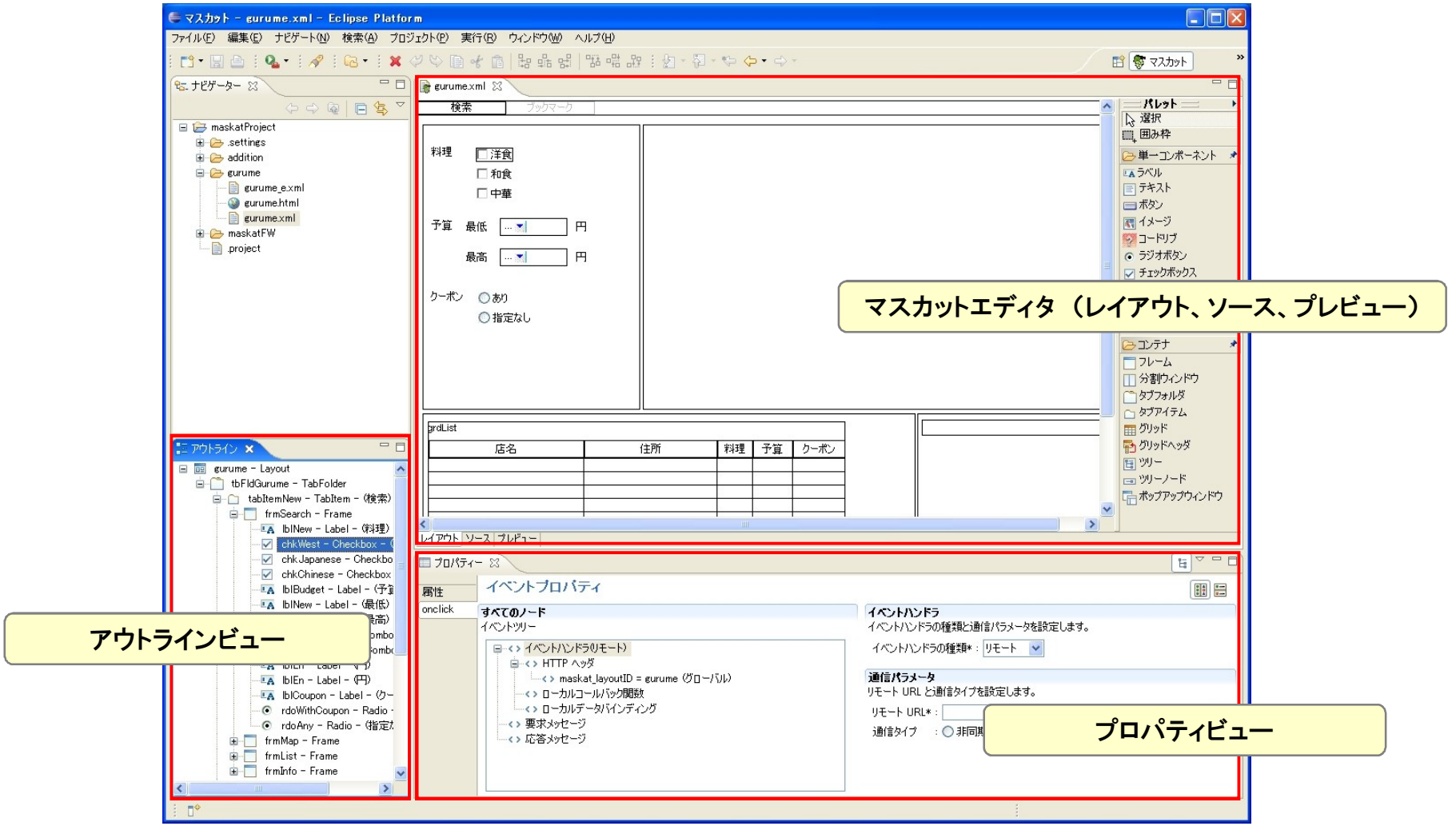
③ マスカットパースペクティブが表示されます。



メモ： 新規アプリケーションウィザードの最後のステップで、マスカットパースペクティブに自動的に切り替えることができます。

2.2 画面構成

Eclipse版IDEの画面構成（マスカットパースペクティブ）は以下のようになっています。



2.3 機能説明

パースペクティブはマスカットエディタ、プロパティビュー、アウトラインビューで構成されています。

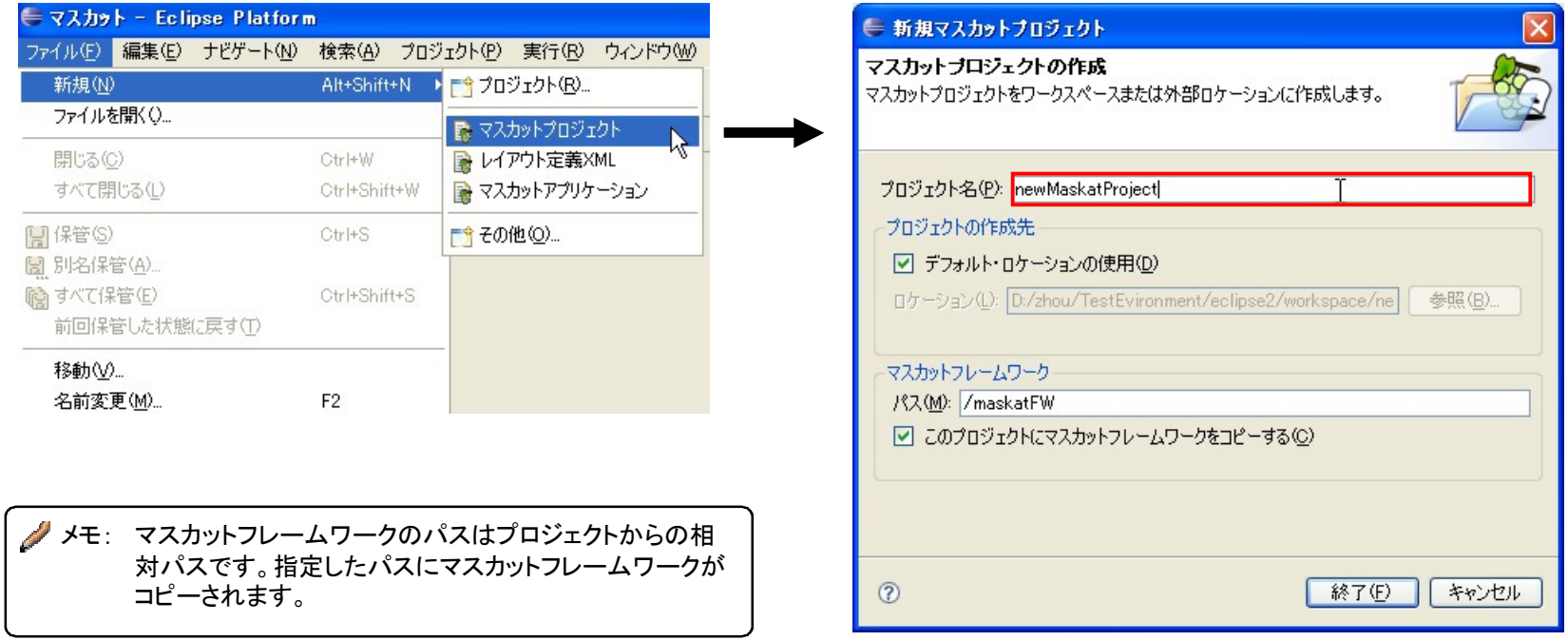
項番	エディタまたはビュー名		説明
1	マスカットエディタ		マスカットエディタは3つのエディタから構成されています。各エディタはタブを切り替えることで利用することができます。
		レイアウト	グラフィカルユーザインタフェースによるレイアウト定義の作成を行います。 主な機能は以下のとおりです。 ・コンポーネントの配置 ・コンポーネントの移動、サイズ変更 ・コンポーネントの削除 ・コンポーネントのラベル直接編集
		ソース	テキストエディタによるレイアウト定義XMLの編集を行います。
		プレビュー	作成したレイアウト定義をブラウザで表示します。
2	プロパティビュー		コンポーネントの属性、イベント定義を編集するエディタです。 以下のビューおよびエディタから構成されています。
		属性	コンポーネントの属性値を編集するビューです。
		イベントプロパティエディタ	イベントハンドラの設定を行うエディタです。
3	アウトラインビュー		レイアウト定義を行ったコンポーネントのアウトライン表示を行うビューです。 ・アウトライン表示機能の他にオブジェクト名の編集機能があります。

3. プロジェクトの作成

この章ではマスカットでアプリケーションを作成するために必要な
ファイルやフォルダの作成方法を説明します。

3.1 マスカットプロジェクトの作成

1. Eclipse を起動して [ファイル] メニューから [新規] > [マスカットプロジェクト] を選択し、新規マスカットプロジェクトダイアログを表示します。
2. プロジェクト名を入力して、[終了] ボタンをクリックします。



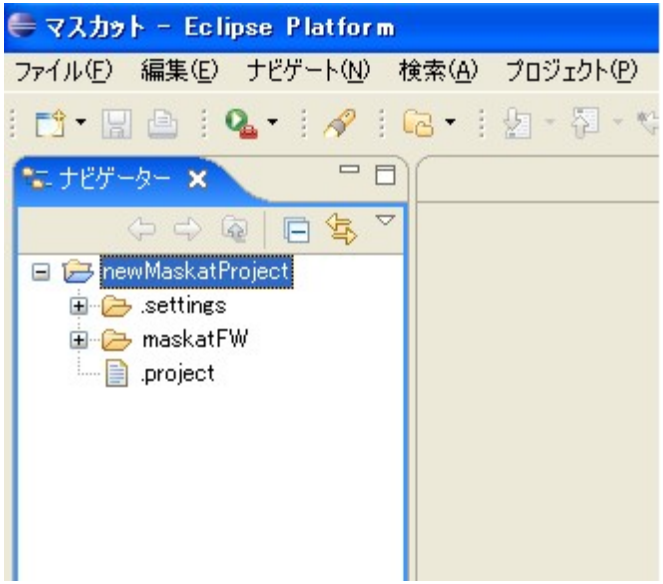
メモ： マスカットフレームワークのパスはプロジェクトからの相対パスです。指定したパスにマスカットフレームワークがコピーされます。

用語:【マスカットプロジェクト】

ワークスペースの最上位ディレクトリです。Eclipse 版 IDE でアプリケーションを開発するには、マスカットプロジェクトを作成する必要があります。

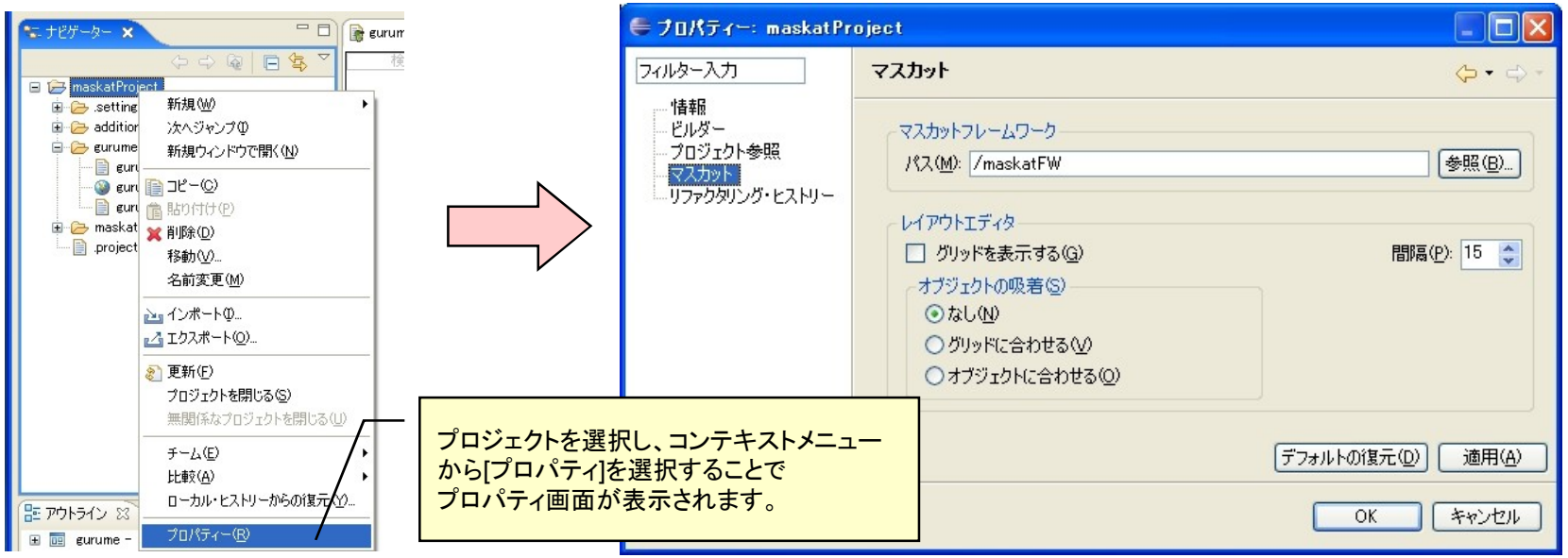
3.1 マスカットプロジェクトの作成

3. マスカットプロジェクトの作成が完了しました。
ナビゲーターから作成したマスカットプロジェクトを確認してください。



3.2 マスカットプロジェクトのプロパティ

マスカットプロジェクトのプロパティ画面では、プロジェクトの設定情報を表示または編集することができます。



項番	分類		項目	説明
1	マスカットフレームワーク		パス	マスカットフレームワークのパスを指定します。 「新規マスカットプロジェクト」ウィザードのマスカットフレームワークパスに指定した値が表示されます。
2	レイアウトエディタ		グリッドを表示する	レイアウトエディタでグリッドを表示するか指定します。
			間隔	レイアウトエディタのグリッド表示間隔を指定します。
		オブジェクトの吸着	なし	オブジェクトの移動単位に何も指定しません。
			グリッドにあわせる	オブジェクトの移動をグリッド単位にします。
			オブジェクトに合わせる	オブジェクトの移動時に他のオブジェクトと中心、上下左右の位置を合わせるためのガイドラインを表示します。

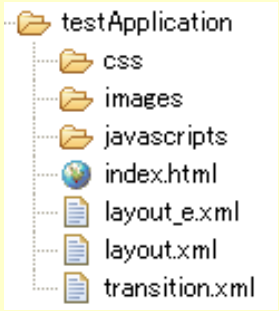
3.3 マスカットアプリケーションの作成

1. [ファイル] メニューから [新規] > [マスカットアプリケーション] を選択して下さい。

用語:【マスカットアプリケーション】

マスカットプロジェクト内に作成するフォルダです。ユーザがブラウザ上から URL でアクセスする単位となり、以下のようなファイル群を格納します。

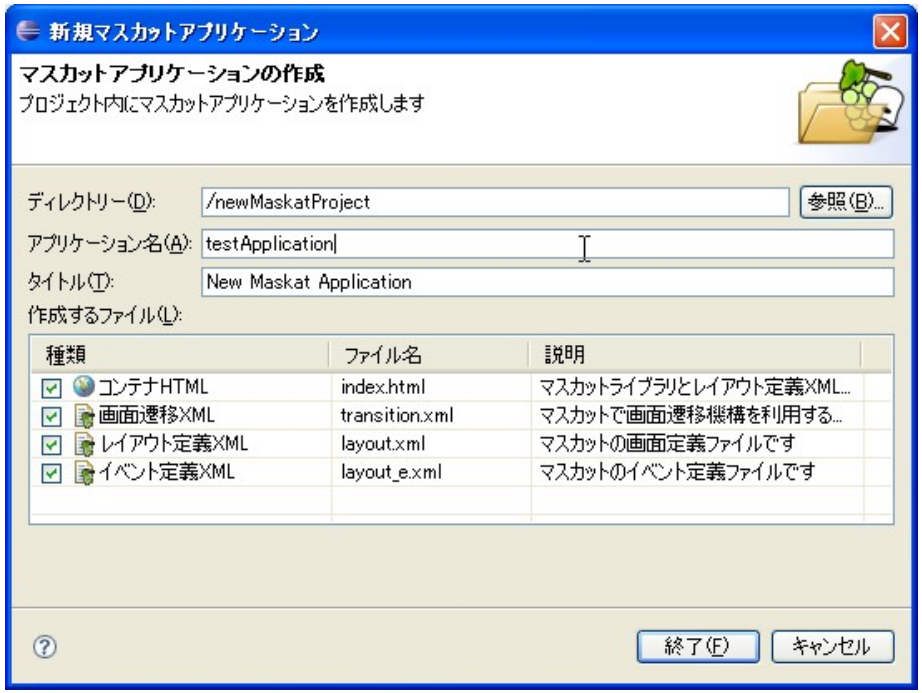
- **レイアウト定義XML**
- **コンテナ HTML**
- **CSS, JavaScript ファイル**
- **イベント定義XML**
- **画面遷移定義XML**
- **画像ファイル**



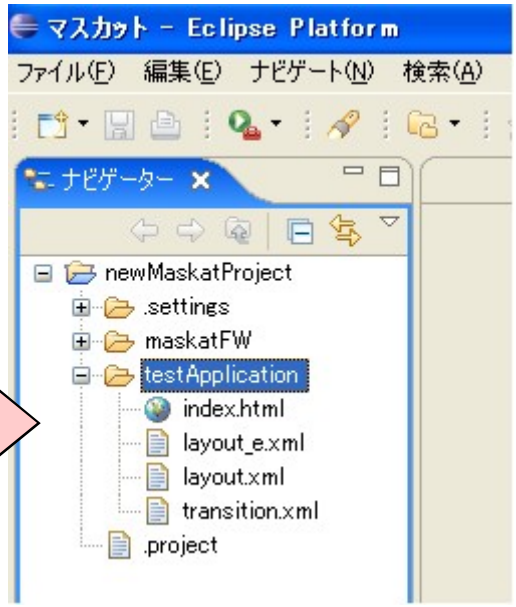
3.3 マスカットアプリケーションの作成

2. アプリケーション名を入力して、[終了]ボタンをクリックします。

- 参照ボタンをクリックして、マスカットプロジェクトを選択できます。
- ファイル名はダブルクリックで変更できます。
  ※ イベント定義 XML のファイル名はレイアウト定義 XML のファイル名に応じて自動的に設定されます。
- タイトルにはコンテナHTMLの title タグに挿入される文字列を指定します。



3. ナビゲーターから作成されたマスカットアプリケーションを確認してください。



3.4 レイアウト定義XMLの追加

1. [ファイル] メニューから [新規] > [レイアウト定義XML] を選択します。

2. 各フィールドを入力して、[終了] ボタンをクリックします。

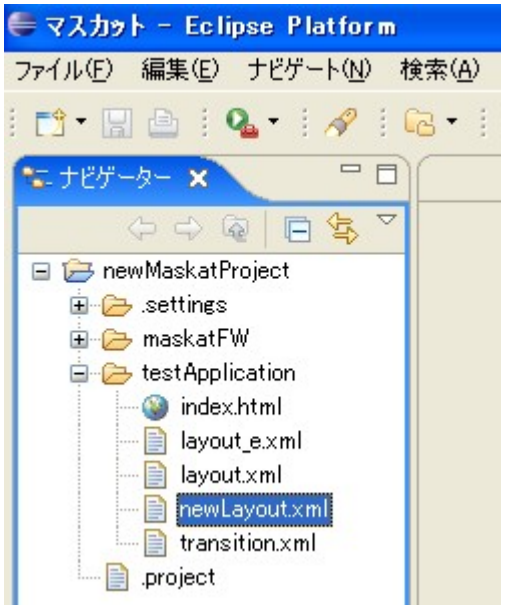
- [参照] ボタンを押して、保存するフォルダを選択します。
- ファイル名とレイアウト名を入力します。

メモ: 画面遷移を伴うようなマスカットアプリケーションでは、複数のレイアウト定義XMLを作成する必要があります。

なお、作成したレイアウト定義 XMLは画面遷移定義 XML (transition.xml)に登録する必要があります。
この作業はテキストエディタで行ってください。

3.4 レイアウト定義XMLの追加

3. レイアウト定義XMLの作成が完了しました。
ナビゲーターから作成したファイルを確認してください。



メモ: レイアウト定義XMLに対応するイベント定義XMLは、このウィザードでは作成されません。
イベント定義 XML は、マスカットエディタがレイアウト定義XMLを保存する際に自動的に作成されます。

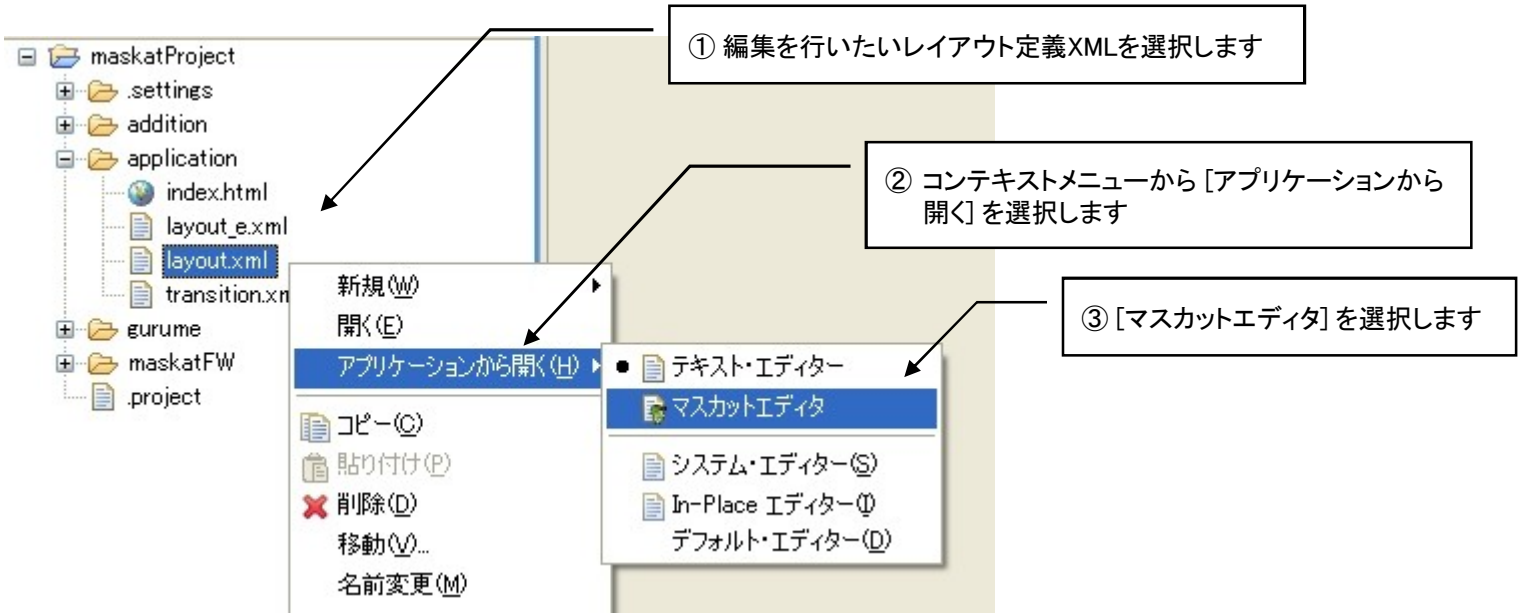
4. レイアウト定義の編集

この章では、マスカットエディタを使用したレイアウト定義 XML の
編集方法について説明します。

4.1 マスカットエディタの開き方

マスカットエディタでレイアウト定義XMLを開くには、以下の手順を行います。

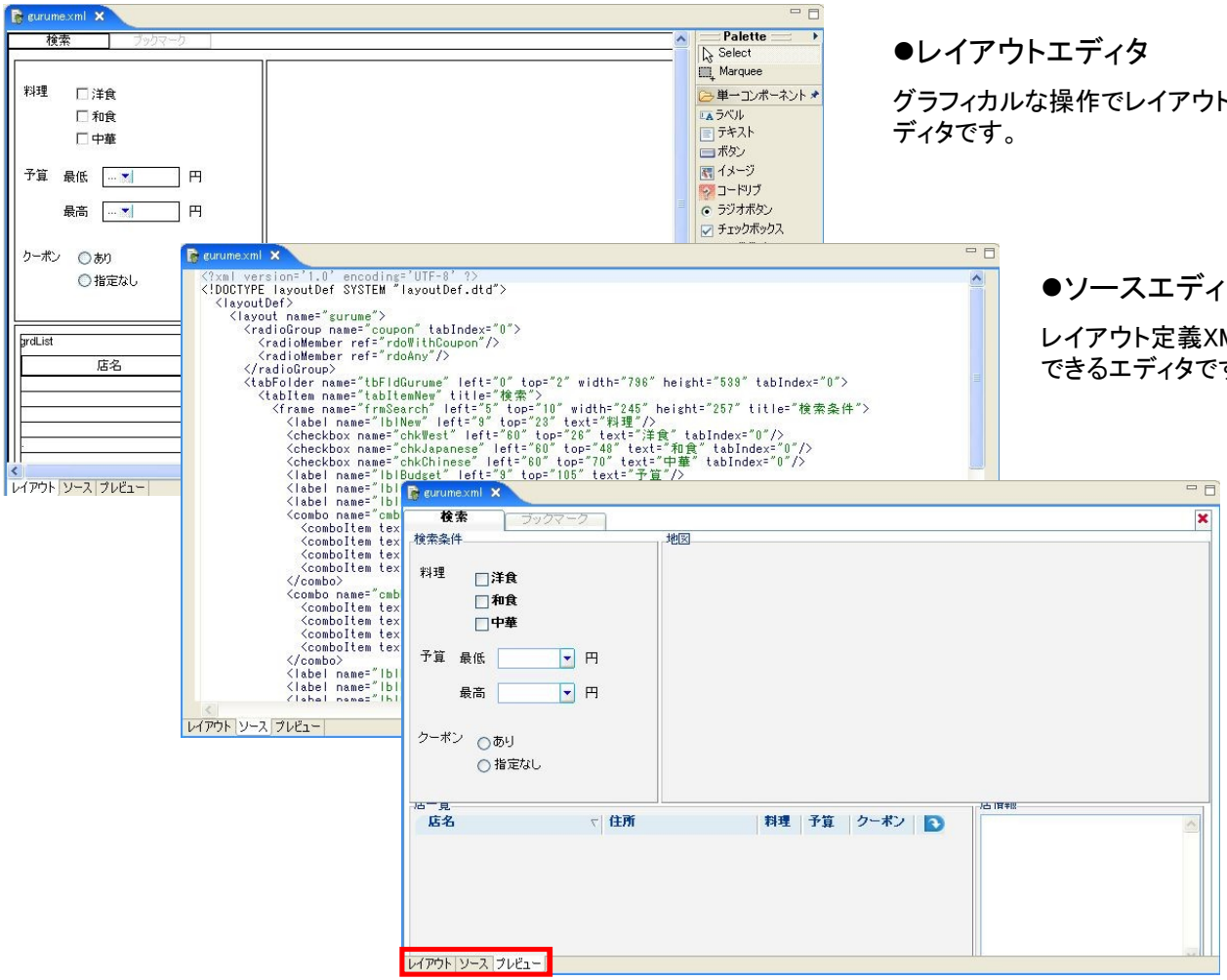
1. 編集を行いたいレイアウト定義XMLを選択します。
2. コンテキストメニューから [アプリケーションから開く] を選択します。
3. [マスカットエディタ] を選択します。



メモ： 新規に作成したレイアウト定義XMLはマスカットエディタに関連付けされていないため、ダブルクリックでは開きません。
2回目以降はダブルクリックでマスカットエディタが起動します。

4.2 マスカットエディタの構成

マスカットエディタは3つのエディタおよびビューで構成されています。



エディタ下部のタブでエディタやプレビューを
切り替えることができます。

●レイアウトエディタ

グラフィカルな操作でレイアウト定義の編集を行うエディタです。

●ソースエディタ

レイアウト定義XMLをテキスト編集できるエディタです。

●プレビュー

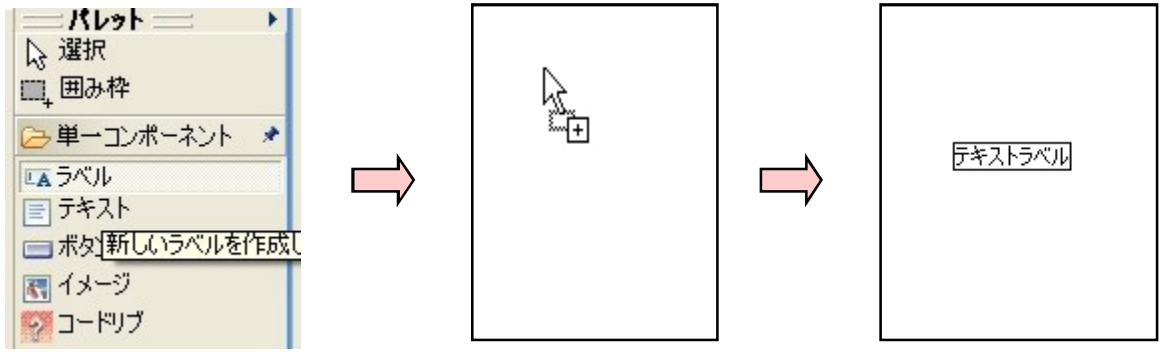
実際にブラウザ上に表示される画面を表示するビューです。

4.3 レイアウトエディタの使い方

4.3.1 コンポーネントの作成

レイアウトエディタではパレットを使用してコンポーネントを作成します。

●単一コンポーネントの作成（例: ラベル）

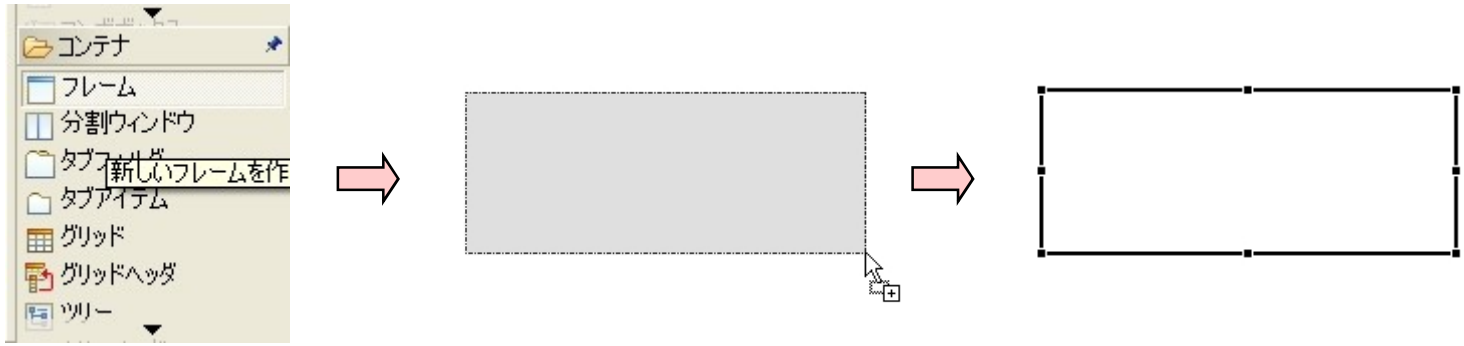


① パレットからラベルをクリックします。

② レイアウトエディタ上で配置を行いたい場所でクリックします。

③ テキストが配置されます。

●コンテナの作成（例: フレーム）



① パレットからフレームをクリックします。

② レイアウトエディタ上で配置を行いたい場所でマウスをドラッグして任意サイズにします。

③ フレームが配置されます。

4.3 レイアウトエディタの使い方

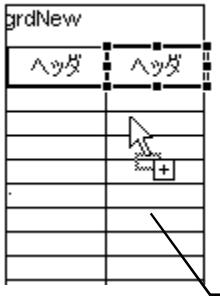
4.3.1 コンポーネントの作成

●複雑なコンポーネントの作成

以下のコンポーネントは子要素との組み合わせにより1つのコンポーネントとして機能します。このようなコンポーネントの作成には複数回のパレット操作が必要です。

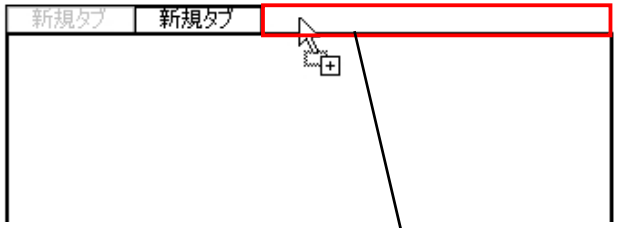
項番	コンポーネント名	子要素となるオブジェクト名	説明
1	グリッド	グリッドヘッダ	グリッドヘッダはグリッドの1要素（列）として機能します。
2	タブフォルダ	タブアイテム	タブアイテムはタブフォルダの1要素（タブ）として機能します。
3	コンボ	コンボアイテム	コンボアイテムはコンボの1要素（ノード）として機能します。
4	ツリービュー	ツリーノード	ツリーノードはツリービューの1要素（ノード）として機能します。

(例) グリッドの作成



グリッドヘッダの追加はパレットからグリッドヘッダを選択しグリッドの上でクリックすることで追加されます。

(例) タブフォルダの作成



タブアイテムの追加はパレットからタブアイテムを選択しタブフォルダのタイトル部分の空き領域（赤い枠）でクリックすることで追加されます。

4.3 レイアウトエディタの使い方

4.3.2 コンポーネントの移動とリサイズ

●コンポーネントの移動

コンポーネントの移動はマウスのドラック＆ドロップで行います。

移動を行いたいコンポーネントを選択しマウスでドラックすると移動枠が表示されます。移動させたい場所でドロップさせると移動を行います。

コンポーネントが移動可能な場合、移動枠上に表示されるマウスカーソルです。

コンポーネントが移動できない場合、移動枠上に表示されるマウスカーソルです。

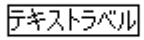
●コンポーネントのリサイズ

コンポーネントのリサイズはマウスのドラッグ＆ドロップで行います。各コンポーネントの種類によってリサイズできる方向が異なっています。リサイズできる方向は選択枠上にある黒い点（ハンドラ）で確認することができます。

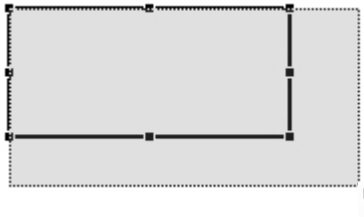
・高さ、幅が変更可能

・幅が変更可能

・高さが変更可能



・変更不可



リサイズを行いたいコンポーネントを選択しハンドラをドラッグするとリサイズ枠が表示されます。変更させたいサイズでドロップさせるとリサイズが行われます。

4.3 レイアウトエディタの使い方

4.3.3 コンポーネントの削除

コンポーネントの削除は以下の2種類の方法で行えます。

●レイアウトエディタ上から削除する方法

① レイアウトエディタ上で削除を行いたいコンポーネントを選択します。

② コンテキストメニューから [削除] を選択します。または、[DEL] キーでも削除することができます。

●アウトラインビュー上から削除する方法

① アウトラインビュー上で削除を行いたいコンポーネントを選択します。

② コンテキストメニューから[削除]を選択します。

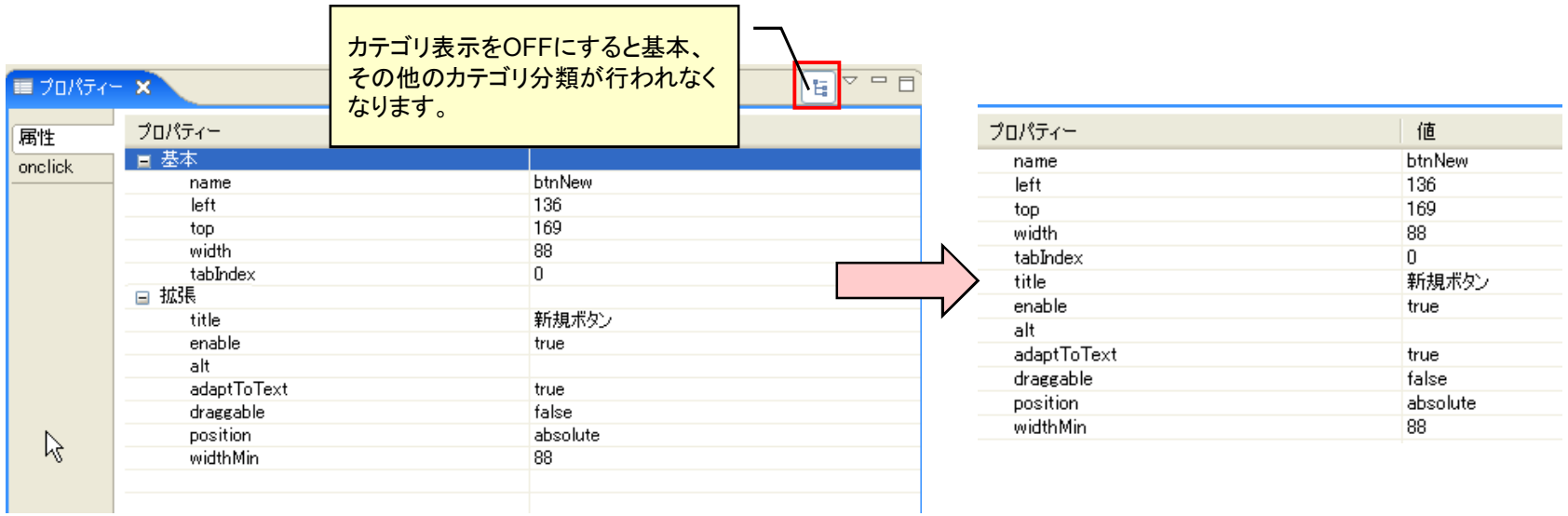
4.3 レイアウトエディタの使い方

4.3.4 プロパティ値の編集

プロパティビューの[属性]タブを選択すると、レイアウトエディタ上で選択されているコンポーネントのプロパティを編集することができます。

プロパティは基本プロパティと拡張プロパティの2種類に分類されます。基本プロパティはオブジェクト名や座標などの共通プロパティであり、
拡張プロパティは各コンポーネントごとに固有のプロパティです。

項番	種別	項目	説明
1	基本プロパティ	name	コンポーネントを識別するオブジェクト名
2		left	X座標
3		top	Y座標
4		width	幅
5		height	高さ
6		tabIndex	タブ移動のインデックス
7	拡張プロパティ	ー	各コンポーネント固有のプロパティ値



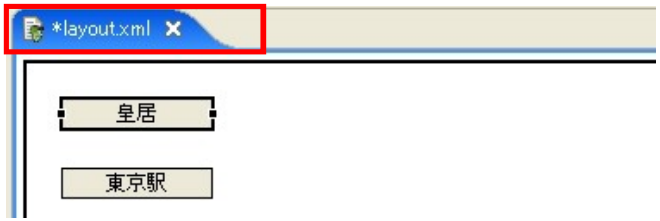
4.3 レイアウトエディタの使い方

4.3.5 保存

レイアウトエディタで編集されたレイアウト情報はレイアウト定義XMLに保存されます。

また、イベントプロパティエディタで編集されたイベント情報はイベント定義XMLに保存されます。イベント定義XMLが存在しない場合には自動的に作成されます。

① マスカットエディタに未保存の変更内容がある場合、ファイル名の前に* (アスタリスク) が付きます。



② レイアウト定義を保存するためには[ファイル]メニューから[保管]を選択します。

> メモ： ショートカットキー [CTRL] + [S] で保管することもできます。

4.3 レイアウトエディタの使い方

4.3.6 その他の機能

マスカットエディタはレイアウト定義の編集を補助するため、下記の機能を持っています。

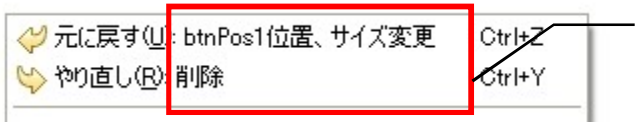
項番	機能名	説明
1	元に戻す、やり直し	直前に実行した操作を取り消したり、取り消した操作を再度実行したりする機能です。
2	コピー（切り取り）＆貼り付け	既に配置しているコンポーネントをコピーし新たなコンポーネントを貼り付ける機能です。 複数のコンポーネントを操作することもできます。
3	整列	配置しているコンポーネントの整列を行う機能です。以下の整列が可能です。 ・中央そろえ（縦）、上そろえ、右そろえ、左そろえ、下そろえ、中央そろえ（横）
4	グリッド	コンポーネントを配置しやすいように、レイアウトエディタ上に枠を表示する機能です。 その枠の格子に合わせてコンポーネントを移動させることが可能です。 また、コンポーネントを移動する時に、ガイドラインを表示させることもできます。
5	直接編集	ラベルなどを持つコンポーネントはレイアウト上のコンポーネントをダブルクリックすることで直接編集することができます。
6	アウトラインビュー	配置したコンポーネントの階層構造をツリーで表示します。 コンポーネントのオブジェクト名(nameプロパティ)の直接編集を行うこともできます。

4.3 レイアウトエディタの使い方

4.3.6 その他の機能

① 元に戻す、やり直し

コンテキストメニューから [元に戻す] (アンドゥ)、[やり直し] (リドゥ)を選択します。



元に戻す、やり直しの操作作象となる処理が表示されます。

② コピー＆ペースト

コピーを行いたいオブジェクトをコンテキストメニューからコピーし、貼り付け先の親となるオブジェクトのコンテキストメニューから貼り付けを行います。複数のオブジェクトやコンテナを選択してコピーすることもできます。

1. コピーを行いたいオブジェクトを選択しコンテキストメニューから[コピー]を選択します

2. コピー先オブジェクトを選択しコンテキストメニューから貼り付けを選択します

3. オブジェクトがコピーされます

メモ: これらの操作はショートカットキーでも実行できます。

Eclipse 標準のキー設定では、元に戻すには [CTRL] + [Z]、やり直しには [CTRL] + [Y]、切り取りには [CTRL] + [X]、コピーには [CTRL] + [C]、貼り付けは [CTRL] + [V] がそれぞれ割り当てられています。

4.3 レイアウトエディタの使い方

4.3.6 その他の機能

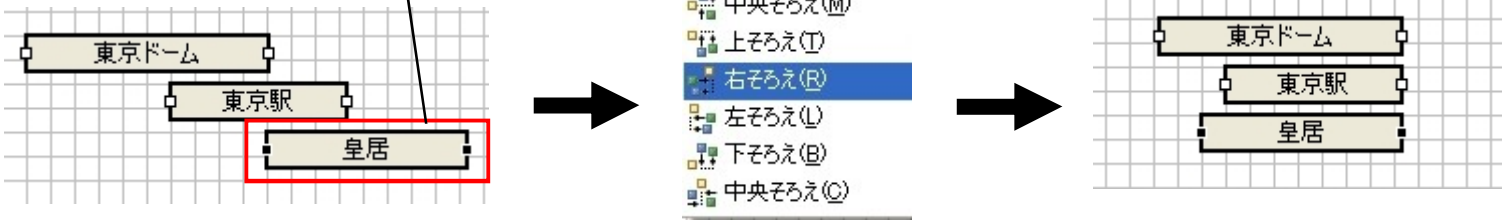
③ 整列

整列を行いたい複数のコンポーネントを選択し、コンテキストメニューの[整列] サブメニューから任意の項目を選択します。

種別	説明
中央そろえ（縦）	基準コンポーネント(*1)の高さの中央が他のコンポーネントの高さの中央になるように整列します。
上そろえ	基準コンポーネントの上に合わせて整列します。
右そろえ	基準コンポーネントの右に合わせて整列します。
左そろえ	基準コンポーネントの左に合わせて整列します。
下そろえ	基準コンポーネントの下に合わせて整列します。
中央そろえ（横）	基準コンポーネントの幅の中央が他のコンポーネントの幅の中央になるように整列します。

メモ*1： 最後に選択したコンポーネントが基準コンポーネントとなります。

黒い点（ハンドラ）を持つコンポーネントが整列の基準となります。



1. 整列を行いたい複数のコンポーネントを選択します

2. 整列の種別を選択します

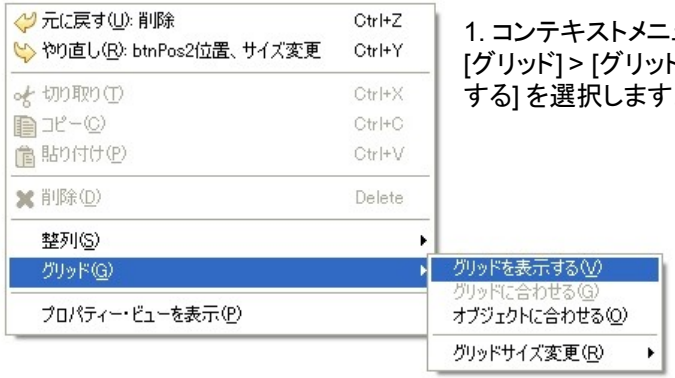
3. 選択されたコンポーネントが整列されます

4.3 レイアウトエディタの使い方

4.3.6 その他の機能

④ グリッド

コンテキストメニューからグリッドの設定が行えます。また「マスカットプロジェクトのプロパティ(p.13)」でも同様の設定が行えます。

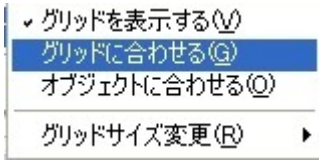


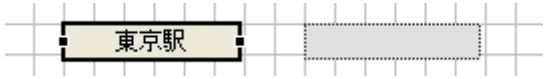
1. コンテキストメニューから[グリッド] > [グリッドを表示する] を選択します。

2. グリッド線がレイアウト上に表示されます。

メモ:
[グリッドサイズ変更] サブメニューからグリッドの幅を変更できます。

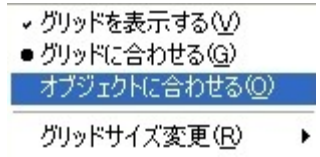
● グリッドに合わせる

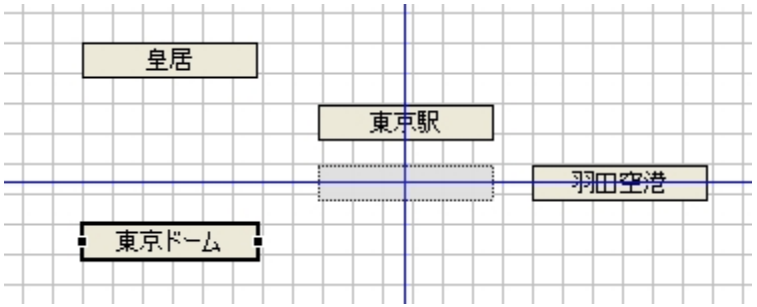




移動単位がグリッドの格子単位となります。

● オブジェクトに合わせる





オブジェクトをドラッグしている際に他のオブジェクトの中央または四隅の座標と一致すると、図のような青いガイドラインが表示されます。

4.3 レイアウトエディタの使い方

4.3.6 その他の機能

⑤ 直接編集

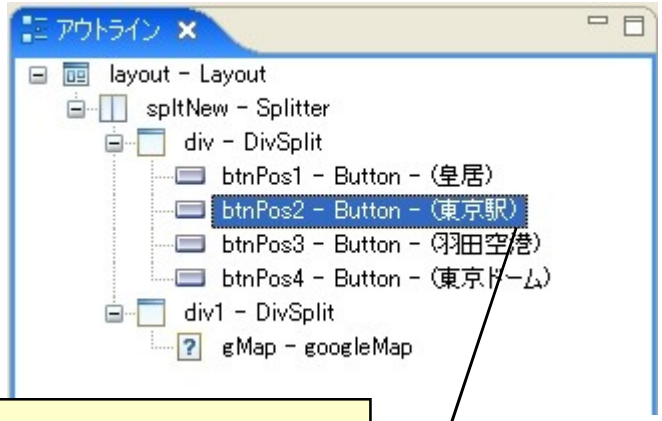
すでに配置済みのコンポーネントをダブルクリックすることで特定のプロパティを編集することができます。

直接編集が行えるコンポーネントと編集対象のプロパティ名

コンポーネント名	プロパティ名	コンポーネント名	プロパティ名
ラベル	text	コンボアイテム	value
テキスト	initValue	HTMLテキスト	html
ボタン	title	タブアイテム	title
ラジオボタン	text	グリッドヘッダ	title
チェックボックス	text	ツリーノード	sText

⑥ アウトラインビュー

レイアウト定義 XML の構造をツリービューアで表示します。



アウトライン上のコンポーネントは選択、削除がレイアウトエディタのコンポーネントと同期します。

●コンテキストメニュー

コンテキストメニューからオブジェクトの削除とオブジェクト名の変更が行えます。

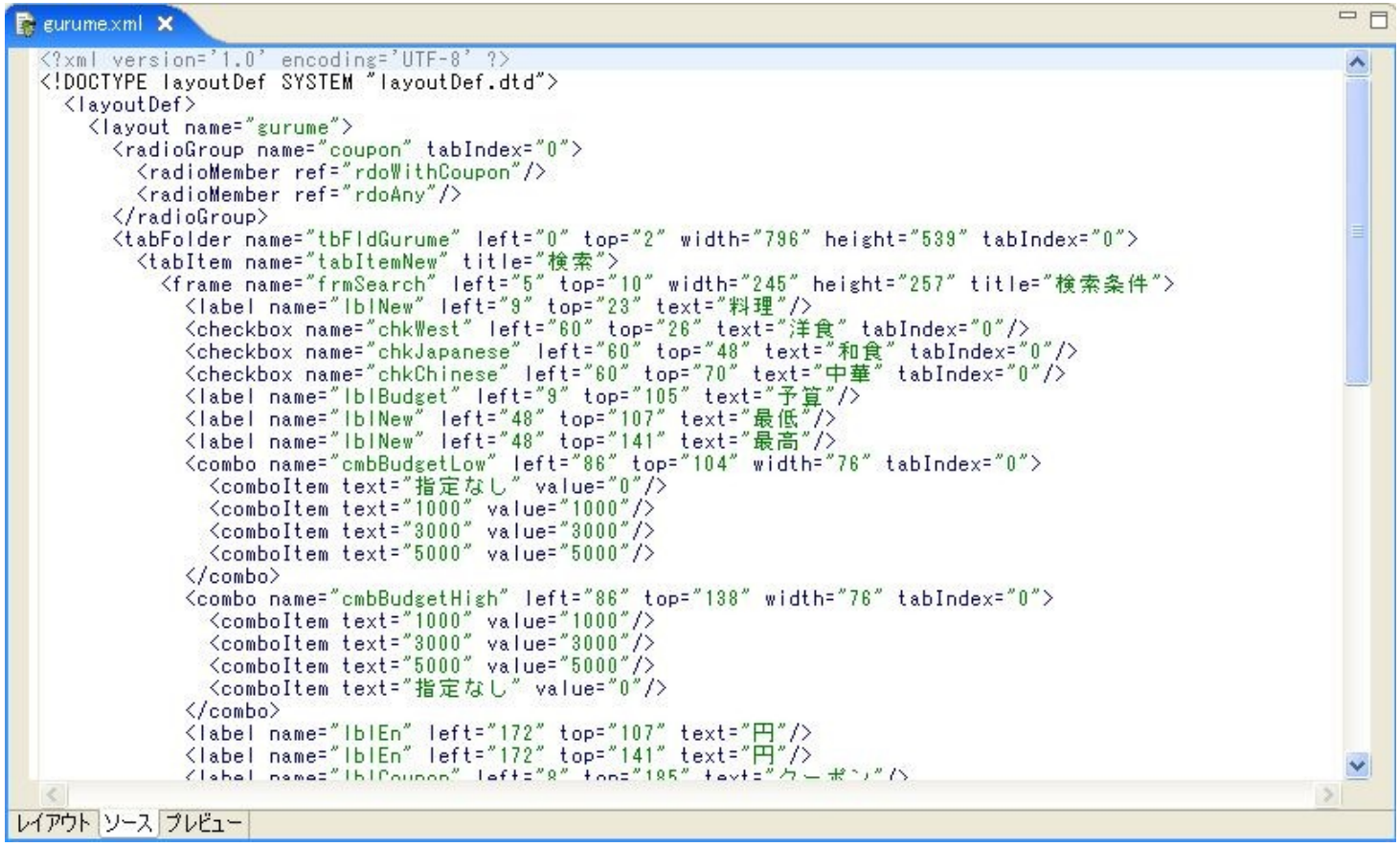
●オブジェクト名の変更

アウトラインビューのノードをダブルクリックするとオブジェクト名(nameプロパティ)の編集が行えます。

4.4 ソースコード表示と編集

ソースエディタはマスカットエディタのソースタブを選択することで利用することができます。

このエディタではレイアウト定義XMLの内容を確認および編集することができます。レイアウトエディタでは編集できない未対応のコンポーネントや属性値を直接 XML に記述する場合に利用します。

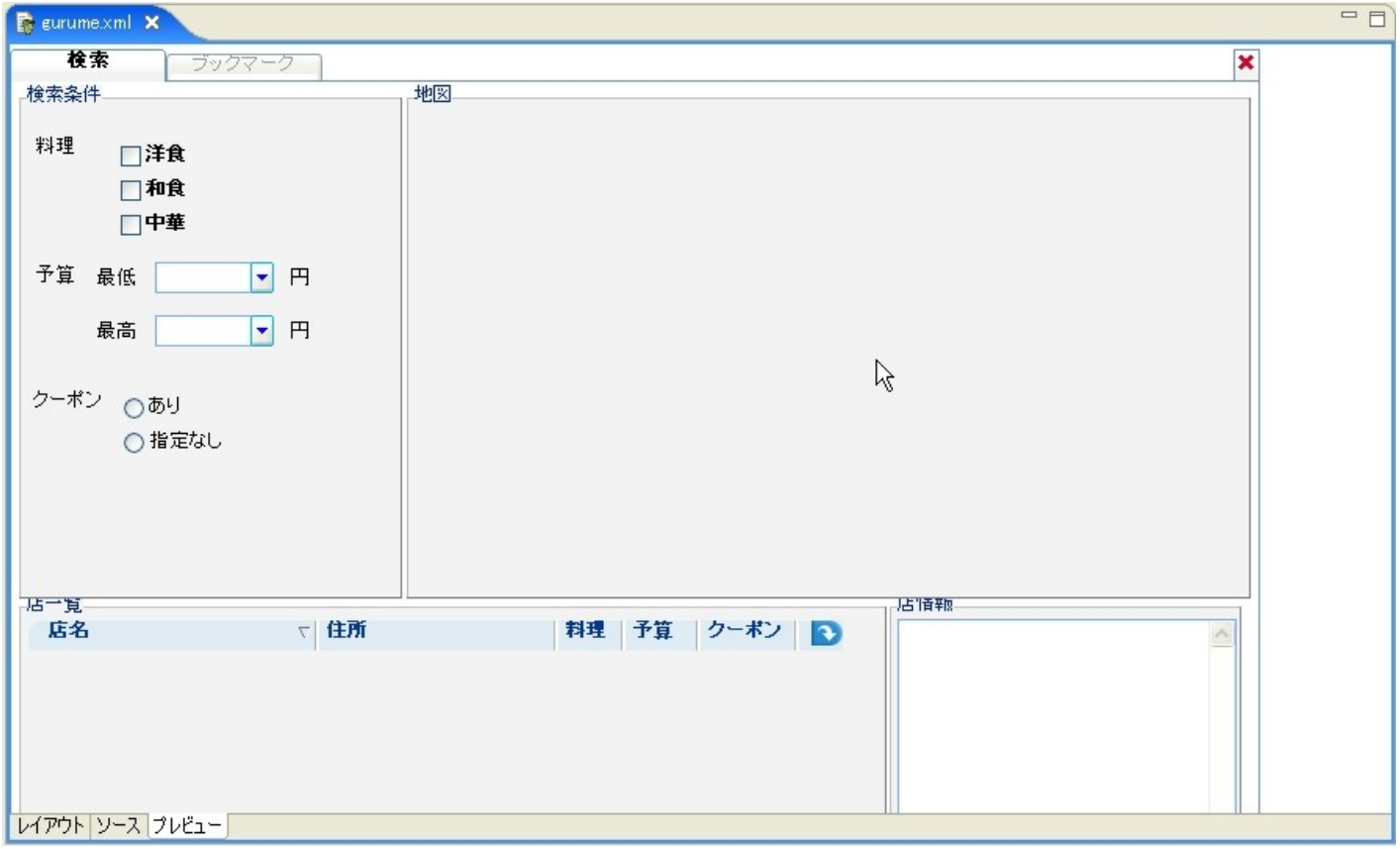


メモ： ソースエディタで編集した内容を他のエディタ(レイアウト、プレビュー)に反映させたい場合は保存を行ってください。

4.5 プレビュー機能

プレビューはマスカットエディタのプレビュータブを選択することで利用することができます。

編集中のレイアウトについて、Webブラウザでの実際の表示イメージを確認することができます。
なお、プレビューではイベント定義は動作しません。



メモ: レイアウトエディタやソースエディタで編集中のレイアウトを表示させるには、マスカットエディタで保存を行ってからプレビューを表示させてください。

4.5 プレビュー機能

プレビューによるレイアウトの表示には以下の制限事項があります。

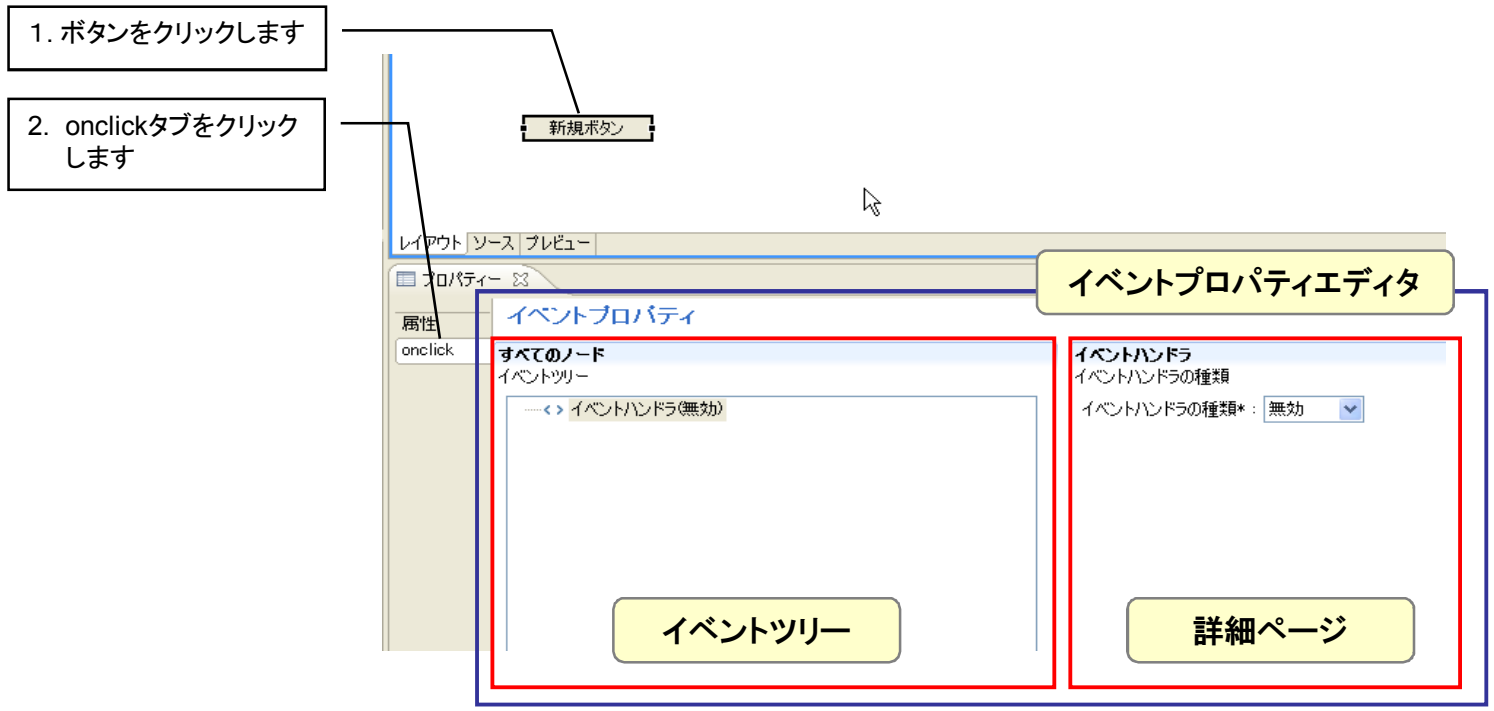
項番	項目
1	グリッドに1つもグリッドヘッダが存在しない場合、エラーとなります。
2	タブフォルダに1つもタブアイテムが存在しない場合、エラーとなります。
3	イメージが配置されている場合、エラーダイアログが表示されます。これはイメージファイルにアクセスできないため表示されるダイアログです。
4	配置した拡張コンポーネントがマスカットフレームワークに登録されていない場合、エラーとなります。

5. イベント定義の編集

この章では、イベントエディタを使用したコンポーネントのイベント
定義の編集方法について説明します。

5.1 イベントプロパティエディタの開き方

1. レイアウトエディタでイベント定義を行いたいコンポーネントを選択します。
2. プロパティビューから編集したいイベントタブをクリックします。
3. イベントプロパティエディタが表示されます。
    - 左側にイベントツリー、右側に選択されたツリーノードの詳細ページが表示されます。
    - 詳細ページに * (アスタリスク) が付いてる項目は必須項目です。



メモ: プロパティビューが表示されてない場合は、コンポーネントのコンテキストメニューから [プロパティビューを表示] を選択してください。

5.2 概要 (イベントの種類)

イベント定義にはローカルイベントとリモートイベントの2種類があります。

- 5.3 節 (p.38) では、ローカルイベントの編集方法について説明します。
- 5.4 節 (p.44) では、リモートイベントの編集方法について説明します。

> 📖 **用語:【ローカルイベント】**
>
> **イベントの発生時に、Web ブラウザ内部で JavaScript 関数を実行して処理する方式です。**
> **ローカルイベントではサーバとの HTTP 通信は行いません。**

> 📖 **用語:【リモートイベント】**
>
> **イベントの発生時に、HTTP 通信を行ってサーバ側のアプリケーションで処理する方式です。**
> **リモートイベントでは HTTP 要求、応答として XML を送受信します。**

5.3 ローカルイベントの編集

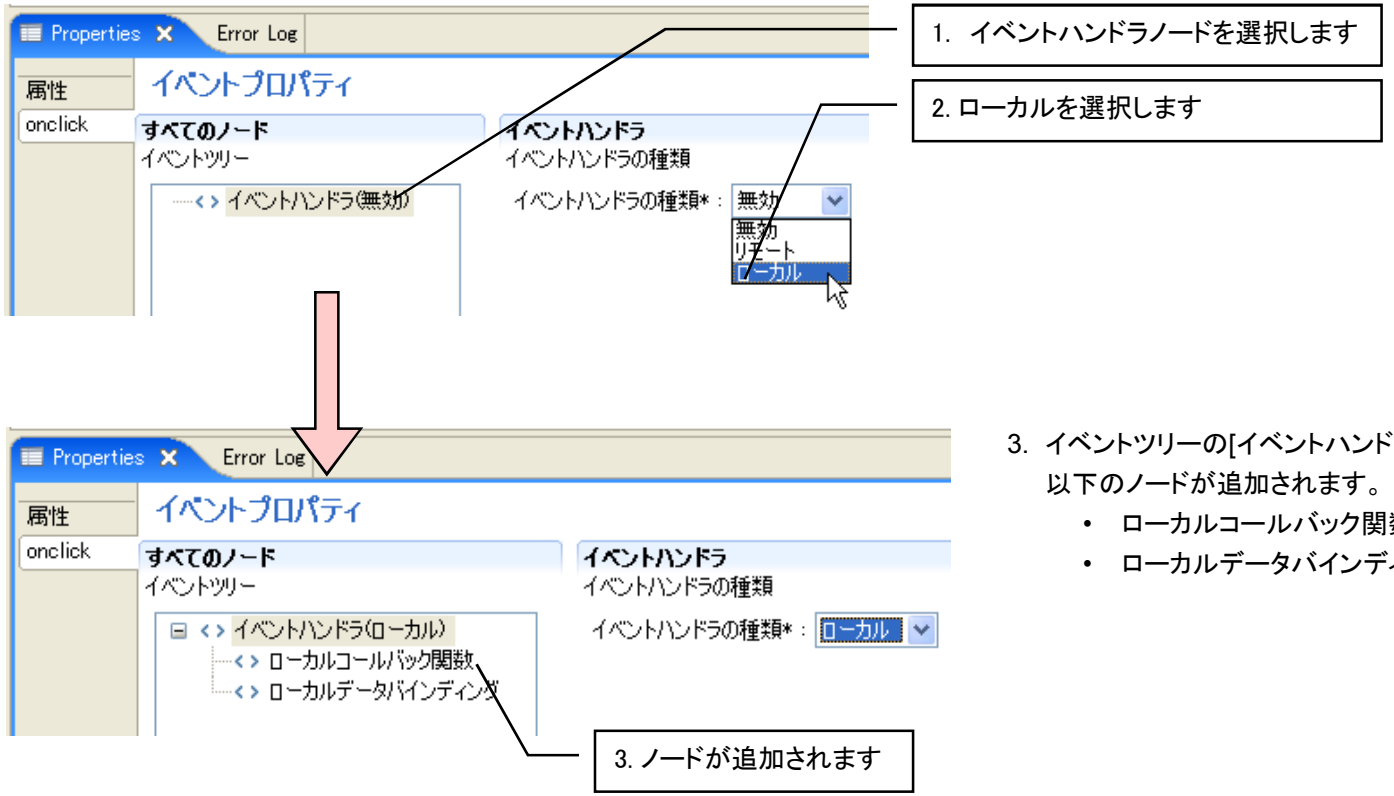
ローカルイベントの編集項目は以下のとおりです。

項番	設定項目		設定内容	必須
1	イベントタイプ		イベントタイプをローカルに設定します。	○
2	ローカルコールバック関数	イベント開始時	イベント開始時に呼び出される関数名	
3		イベント終了時	イベント終了時に呼び出される関数名	
4	ローカルデータバインディング	データ取得元	データ取得元のオブジェクト名	
5		データ格納先	データ格納先のオブジェクト名	
6		データ生成者	拡張ローカルマッピングのマッピング名	

5.3 ローカルイベントの編集

5.3.1 イベントハンドラの種類の設定

1. イベントツリーの [イベントハンドラ] ノードを選択します。
2. 詳細ページのイベントハンドラの種類を [ローカル] に変更します。

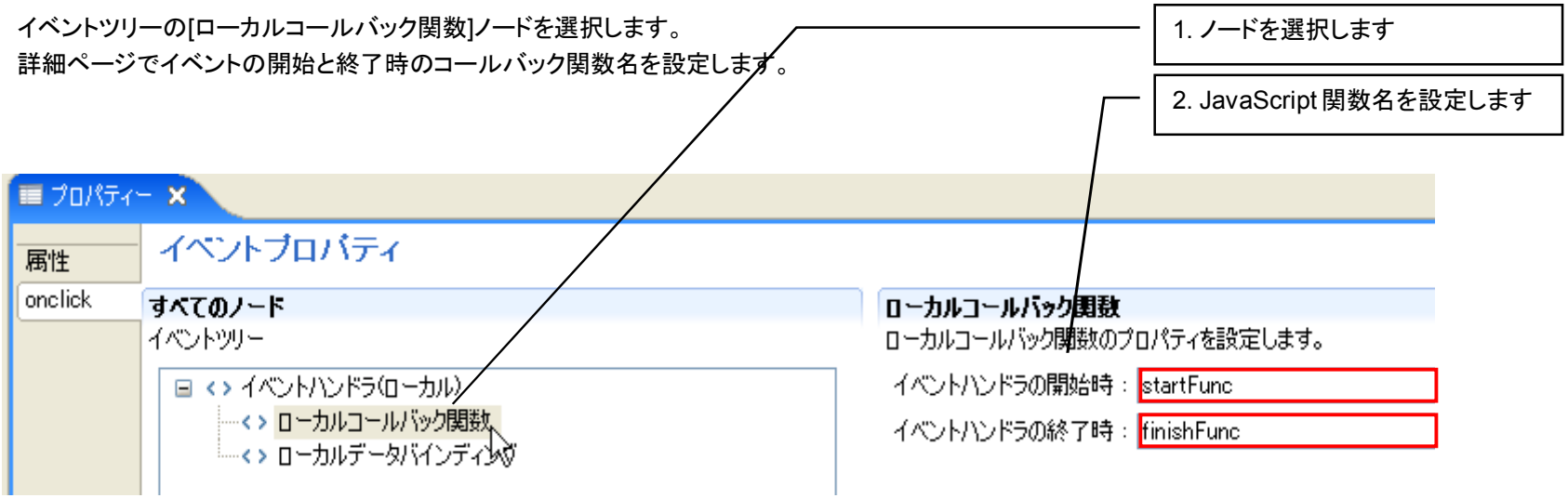


3. イベントツリーの[イベントハンドラ]ノードの下に以下のノードが追加されます。
   - ローカルコールバック関数
   - ローカルデータバインディング

5.3 ローカルイベントの編集

5.3.2 ローカルコールバック関数の設定

1. イベントツリーの[ローカルコールバック関数]ノードを選択します。
2. 詳細ページでイベントの開始と終了時のコールバック関数名を設定します。



用語:【ローカルコールバック関数】

イベント処理の特定のタイミングで、ユーザが指定した任意の JavaScript 関数を呼び出す機能です。
以下のタイミングが指定可能です:

① イベント処理の開始時 (start)
② イベント処理の終了時 (finish)

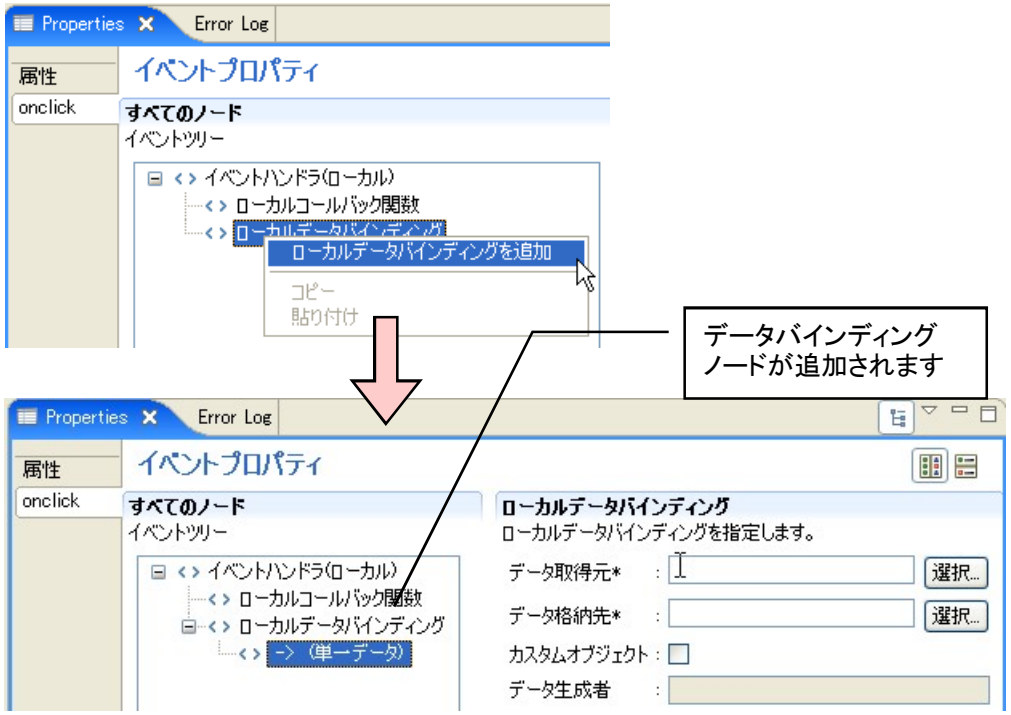
リモートイベントの場合、さらに以下の 3 種類のタイミングが指定できます。

③ HTTP 要求の送信前 (before)
④ HTTP 応答の受信後 (after)
⑤ HTTP 要求のタイムアウト発生時 (onTimeoutError)

5.3 ローカルイベントの編集

5.3.3 ローカルデータバインディングの追加

1. イベントツリーの[ローカルデータバインディング]ノードを右クリックします。
2. コンテキストメニューの[ローカルデータバインディングを追加]を選択します。



3. データバインディングノードが追加されます。

用語:【ローカルデータバインディング】

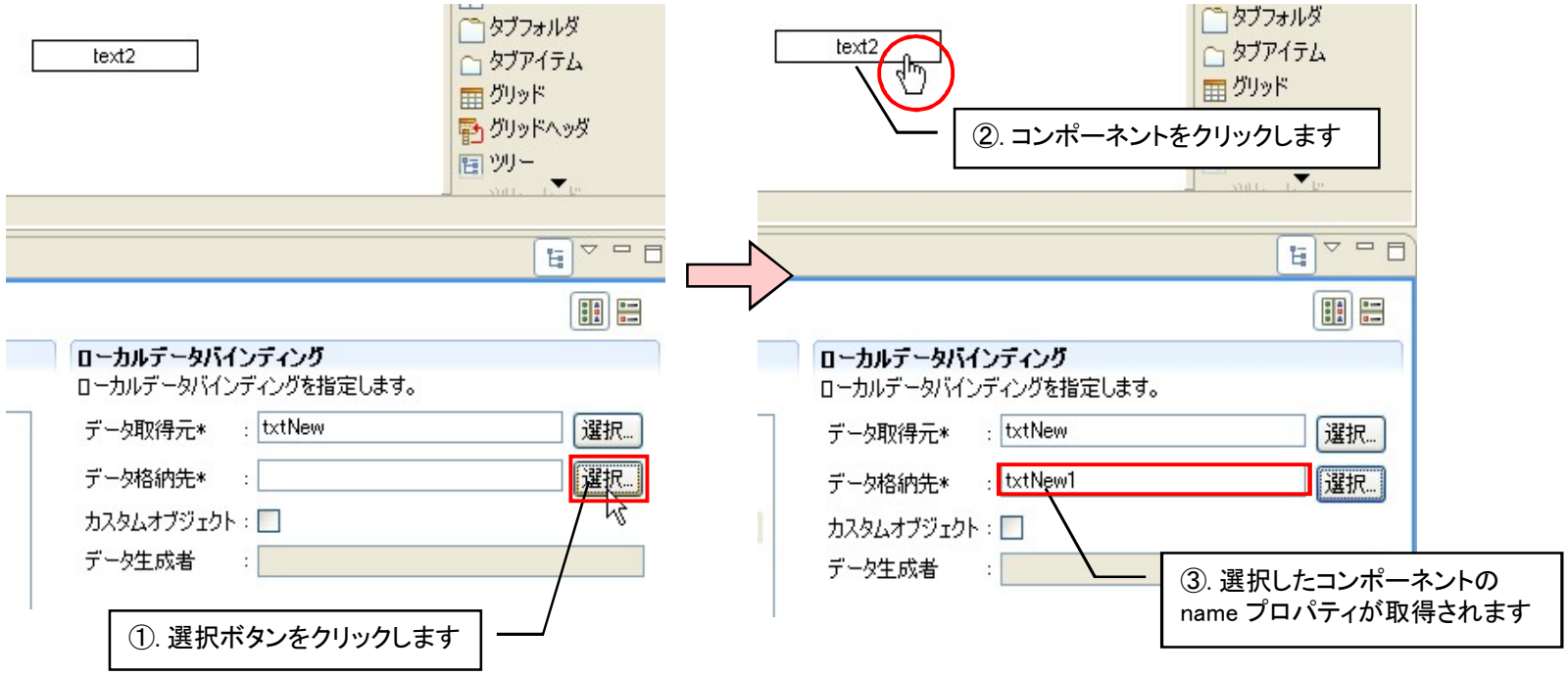
マスカットのコンポーネントや JavaScript 変数の間で、データ値の代入を行う操作を指します。
データの取得元、取得先のコンポーネント名や変数名を指定する必要があります。

5.3 ローカルイベントの編集

5.3.3 ローカルデータバインディングの追加

4. 追加されたノードを選択して、詳細ページでデータ取得元とデータ格納先を編集します。

> メモ: 詳細ページ右側にある [選択] ボタンをクリックすると、レイアウトエディタからコンポーネントを選択することができます。

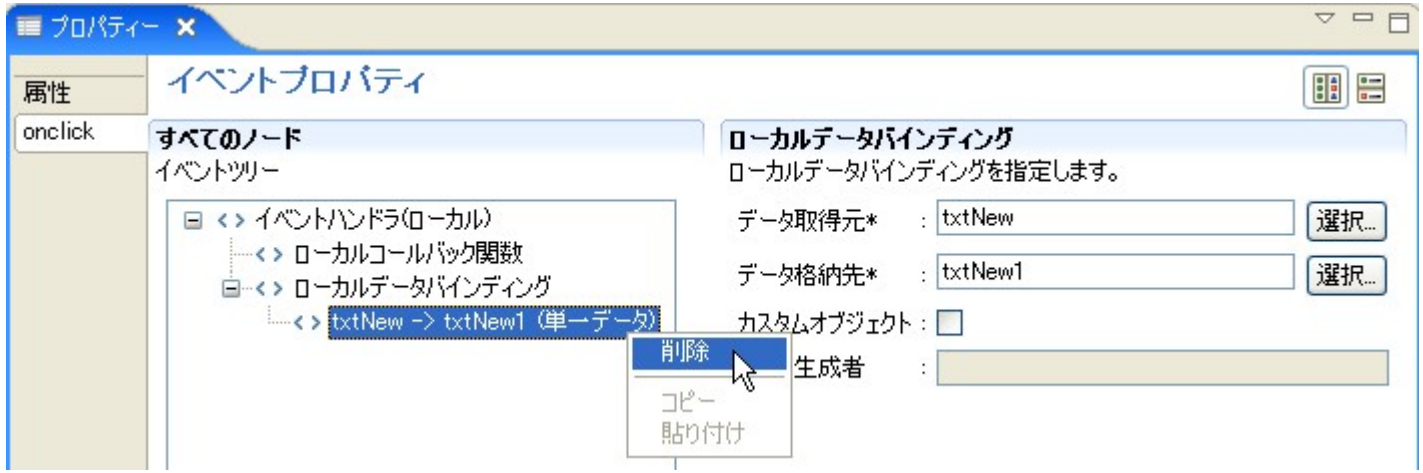


5. 1～4の操作を繰り返して、複数のローカルデータバインディングを追加できます。

5.3 ローカルイベントの編集

5.3.4 ローカルデータバインディングの削除

1. データバインディングノードを右クリックして、コンテキストメニューから [削除] を選択します。
2. 該当するデータバインディングが削除されます。



5.4 リモートイベントの編集

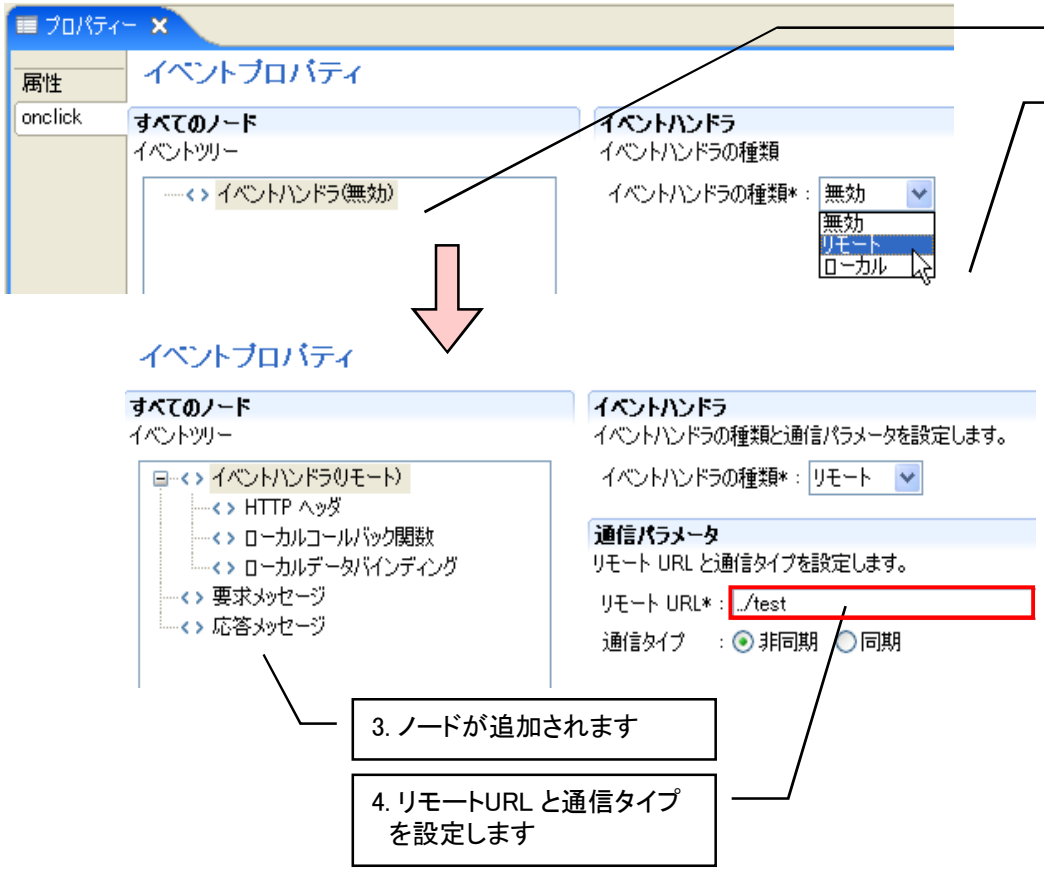
リモートイベントの編集項目は以下のとおりです。

項番	主な設定項目	主な設定内容	必須
1	イベント属性	イベントタイプをリモートに設定します。	○
2		URL	○
3		通信タイプ	○
4	HTTPヘッダ	名前、値	
5		ヘッダの種類（ローカル／グローバル）	
6	ローカルコールバック関数	イベント開始時、イベント終了時	
7		要求メッセージ送信前、応答メッセージ受信後	
8		タイムアウト時間、タイムアウト発生時	
9	ローカルデータバインディング	データ取得元、データ格納先	
10		データ生成者	
11	要求メッセージ	XML構文、送信するデータ	
12	応答メッセージ	XML構文、受信するデータ	

5.4 リモートイベントの編集

5.4.1 イベントハンドラの種類の設定

1. イベントツリーの [イベントハンドラ] ノードを選択します。
2. 詳細ページのイベントハンドラの種類を [リモート] に変更します。



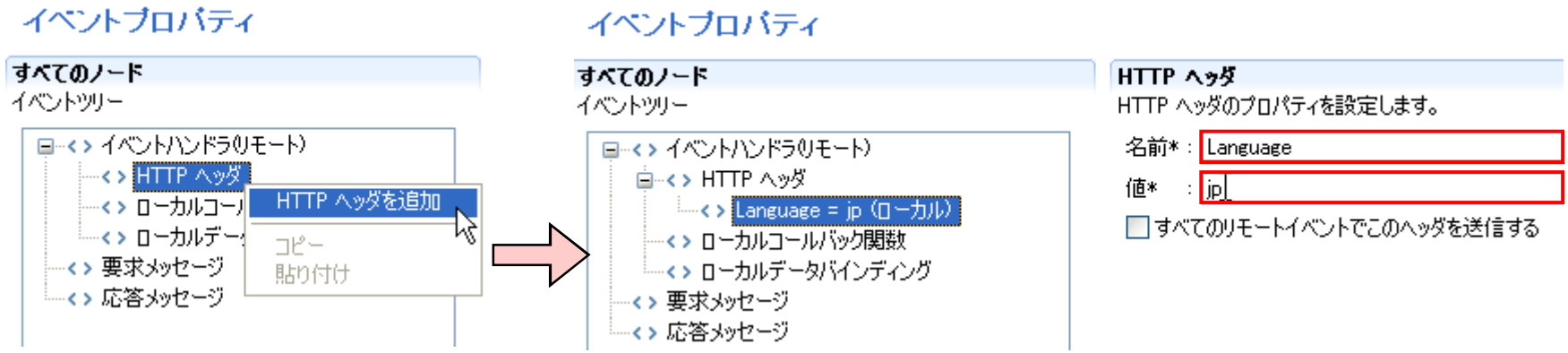
3. イベントツリーに以下のノードが追加されます。
   - HTTPヘッダ
   - ローカルコールバック関数
   - ローカルデータバインディング
   - 要求メッセージ
   - 応答メッセージ
4. 詳細ページでリモートURLと通信タイプを編集します。

5.4 リモートイベントの編集

5.4.2 HTTPヘッダの追加

1. イベントツリーの [HTTPヘッダ] ノードを右クリックして、コンテキストメニューの [HTTPヘッダを追加] を選択します。

2. ローカルHTTPヘッダが追加されます。追加されたノードを選択して、詳細ページでヘッダの名前と値を編集します。
   - [すべてのリモートイベントでこのヘッダを送信する] チェックボックスをチェックすると、グローバルヘッダが設定できます。



用語:【HTTPヘッダ】

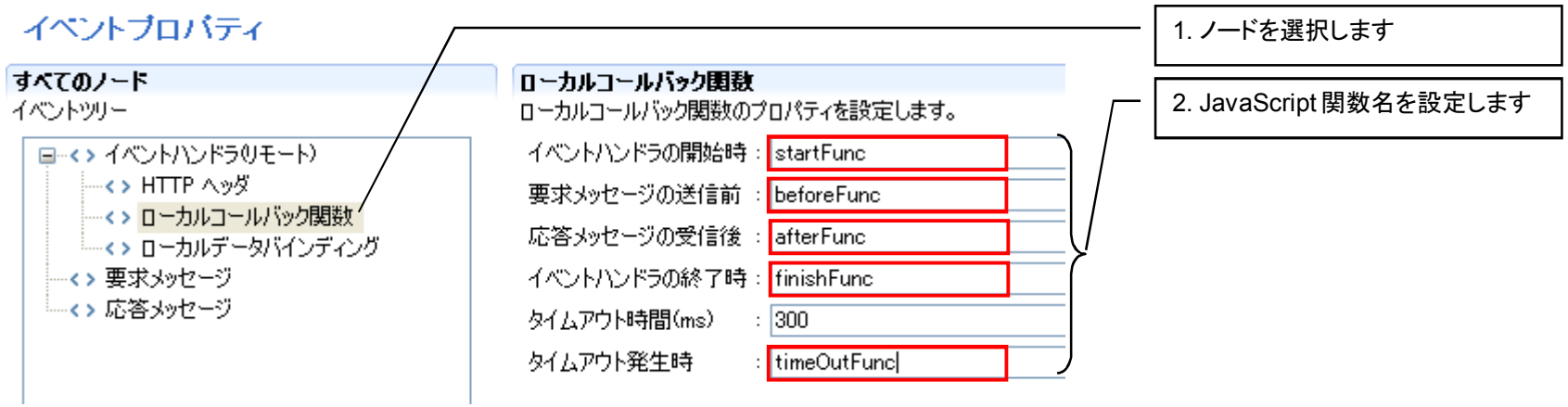
リモートイベントでは HTTP 要求の送信時に任意の HTTP ヘッダを付加することができます。
HTTPヘッダにはローカルとグローバルの2種類があります。

種類	説明
ローカル HTTPヘッダ	現在編集しているイベントのみで送信されるHTTPヘッダ
グローバル HTTPヘッダ	イベントファイルに含まれるすべてのリモートイベントで送信される HTTPヘッダ

5.4 リモートイベントの編集

5.4.3 ローカルコールバック関数の設定

1. イベントツリーの[ローカルコールバック関数]ノードを選択します。
2. 詳細ページで呼び出しタイミングごとのコールバック関数名を設定します。



5.4.4 ローカルデータバインディングの追加と削除

ローカルデータバインディングの設定方法は、ローカルイベントの場合と同様です。
ローカルイベントの編集（P.41～P.43）を参照してください。

メモ： リモートイベントでローカルデータバインディング機能を使用するためには、応答メッセージを受信する必要があります。

5.4 リモートイベントの編集

5.4.5 要求メッセージの設定

要求メッセージの設定項目は以下の通りです。

項番	主な設定項目			主な設定内容	必須
1	ルート要素			ルートノード名	○
2	子要素	単一データ送信		ノード名、データ取得元	
3		データ検証		データ検証タイプ、検証属性など	
4		複数データ送信	第1階層	ノード名、データ取得元	
5			第2階層	ノード名	
6			第3階層	ノード名、データ取得キー	

用語:【単一データ送信】

単一の XML 要素を用いて、コンポーネントや JavaScript 変数のデータを送信する操作です。

例）テキストのデータ送信

マスカット太郎 → 送信 → `<userName>マスカット太郎</userName>`

用語:【複数データ送信】

複数の XML 要素による構文を用いて、コンポーネントや JavaScript 変数（配列）のデータを送信する操作です。

例）グリッドのデータ送信

商品コード	数量
MA01	10
MA02	15
…	…

送信 →

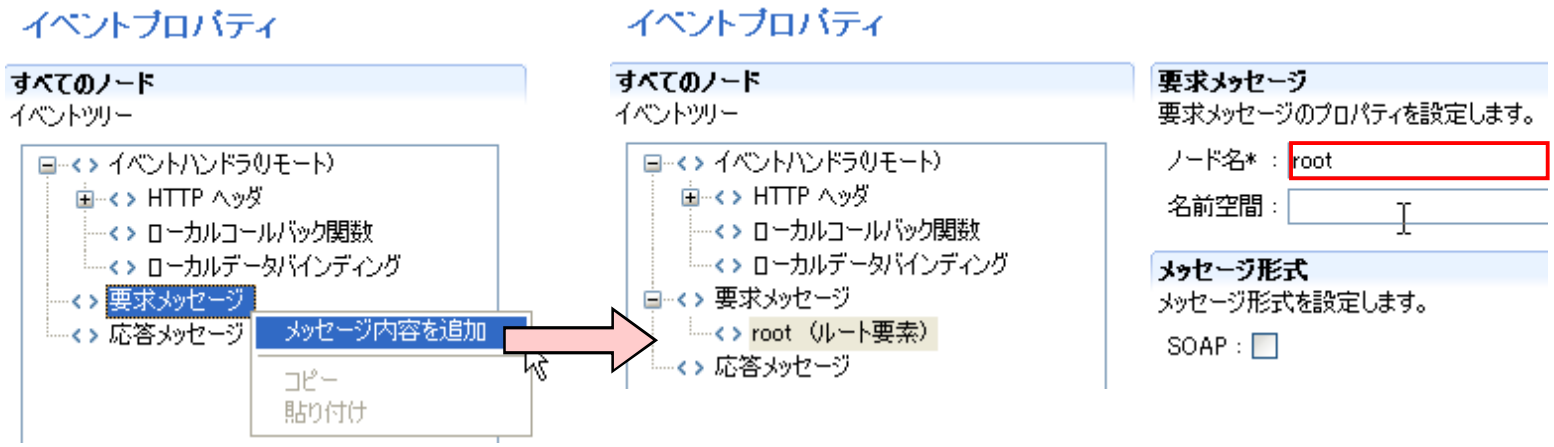
```
<order>
 <orderLine>
  <productId>MA01</productId>
  <quantity>10</quantity>
 </orderLine>
 <orderLine> … </orderLine>
</order>
```

5.4 リモートイベントの編集

5.4.5 要求メッセージの設定

① ルート要素の追加

1. [要求メッセージ] ノードを右クリックして、コンテキストメニューの[メッセージ内容を追加]を選択します。
2. [ルート要素] ノードが追加されます。
3. 追加されたノードを選択して、詳細ページでノード名を編集します。
   必要に応じて送信される XML のデフォルト名前空間 URI も指定することができます。

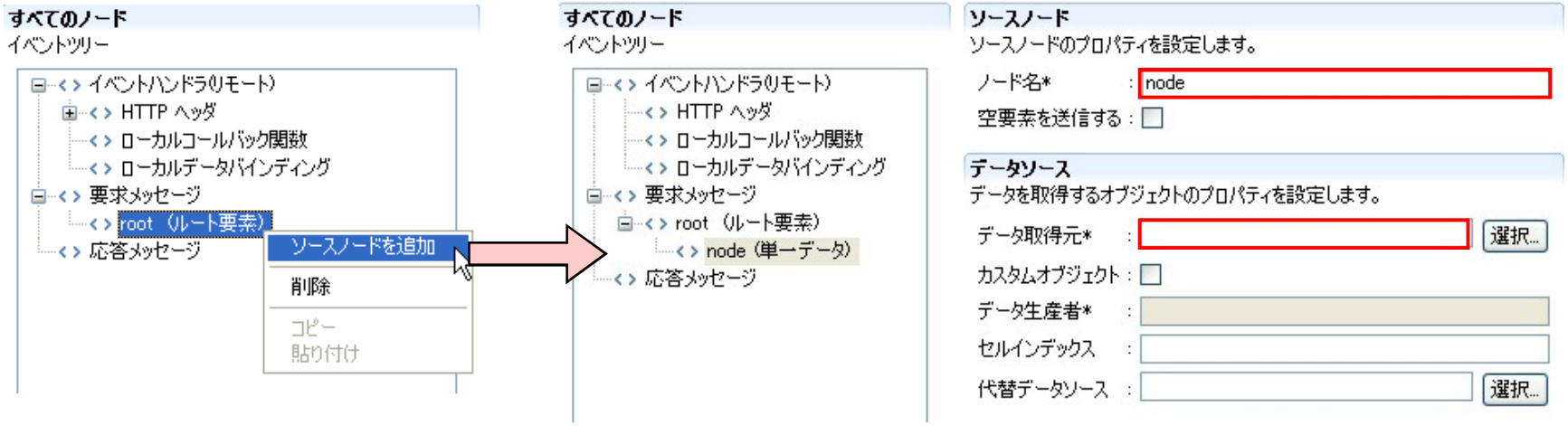


5.4 リモートイベントの編集

5.4.5 要求メッセージの設定

② 単一データ送信の追加

1. [ルート要素]ノードを右クリックして、コンテキストメニューの[ソースノードを追加]を選択します。
2. ソースノード (単一データ) が追加されます。
3. 追加されたノードを選択して、詳細ページでノード名とデータ取得元を編集します。



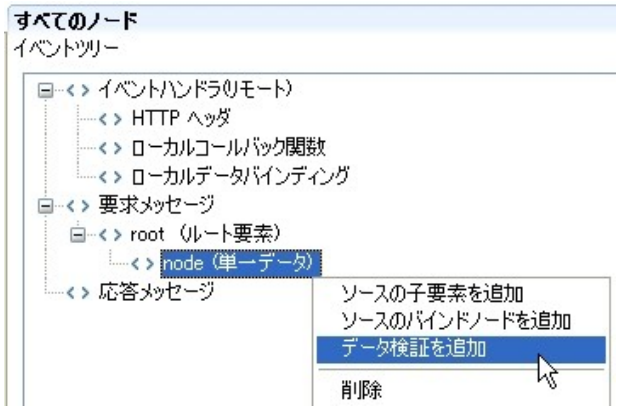
メモ: 詳細ページ右側にある [選択] ボタンをクリックすると、レイアウトエディタからデータ取得元のコンポーネントを選択することができます。

5.4 リモートイベントの編集

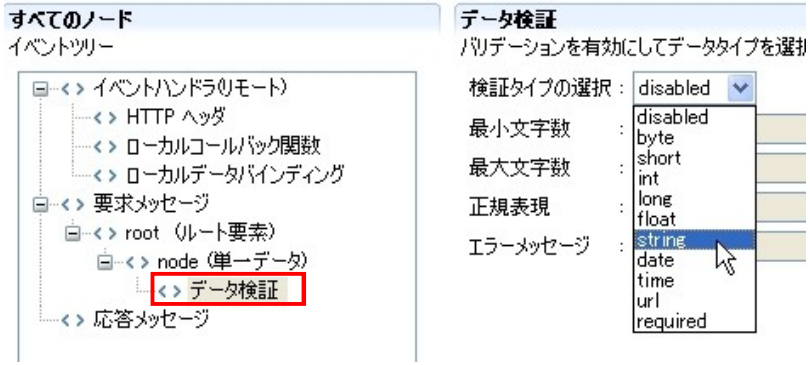
5.4.5 要求メッセージの設定

③ データ検証の追加

1. ソースノードを右クリックして、コンテキストメニューの[データ検証を追加]を選択します。
2. [データ検証]ノードが追加されます。



3. 追加されたノードを選択して、詳細ページで検証内容を編集します。



選択可能な検証タイプの一覧

タイプ	説明
byte	-128~127
short	-32768~32767
int	-2147483648~2147483647
long	-9223372036854776000~9223372036854776000
float	±10^-38～10^38
string	正規表現の設定に従います。
date	yyyy-MM-dd
time	hh:mm:ss
url	(http or https or ftp):// で開始する文字列
required	必須チェック
disabled	データ検証を無効にします。

データ検証の設定項目

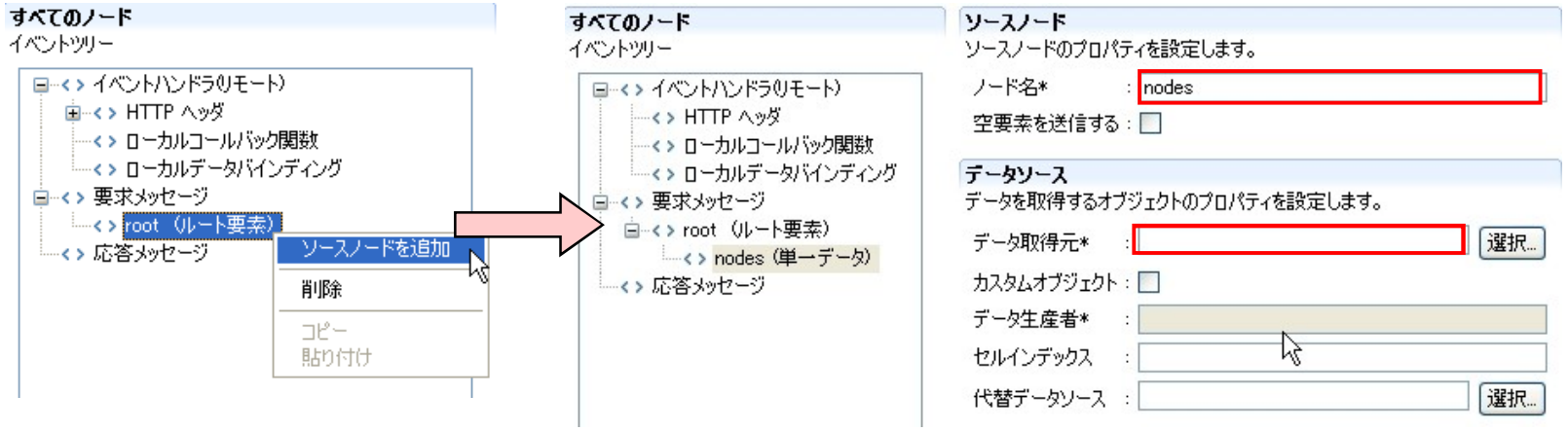
項目	意味
最小文字数	送信されるデータの最小文字数を検査します。
最大文字数	送信されるデータの最大文字数を検査します。
正規表現	ここで設定した正規表現にしたがって送信されるデータを検査します。
エラーメッセージ	エラーダイアログに表示する説明文を指定します 。

5.4 リモートイベントの編集

5.4.5 要求メッセージの設定

④ 複数データ送信の追加 (第1階層)

1. [ルート要素]ノードを右クリックして、コンテキストメニューから[ソースノードを追加]を選択します。
2. 単一データのソースノードが追加されます。
3. 追加されたノードを選択して、詳細ページでノード名およびデータ取得元を編集します。



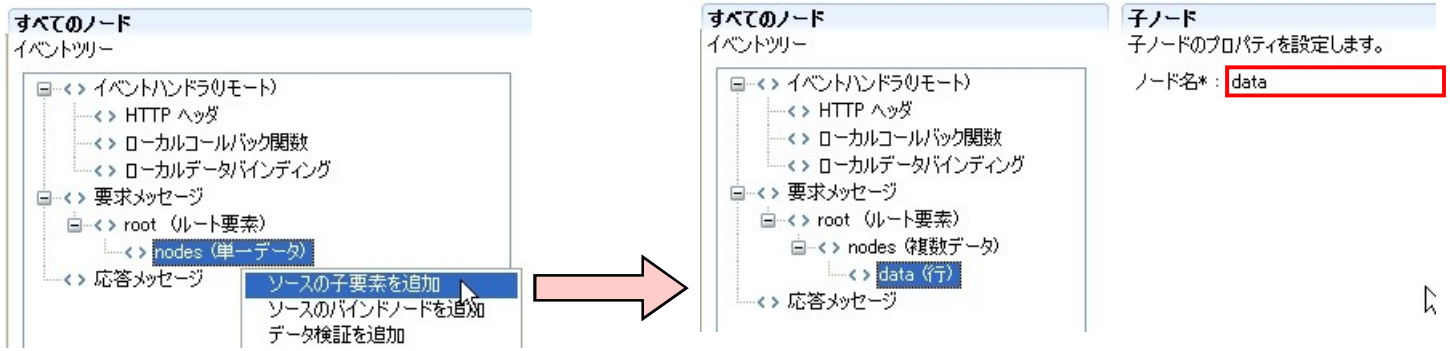
メモ: 詳細ページ右側にある [選択] ボタンをクリックすると、レイアウトエディタからデータ取得元のコンポーネントを選択することができます。

5.4 リモートイベントの編集

5.4.5 要求メッセージの設定

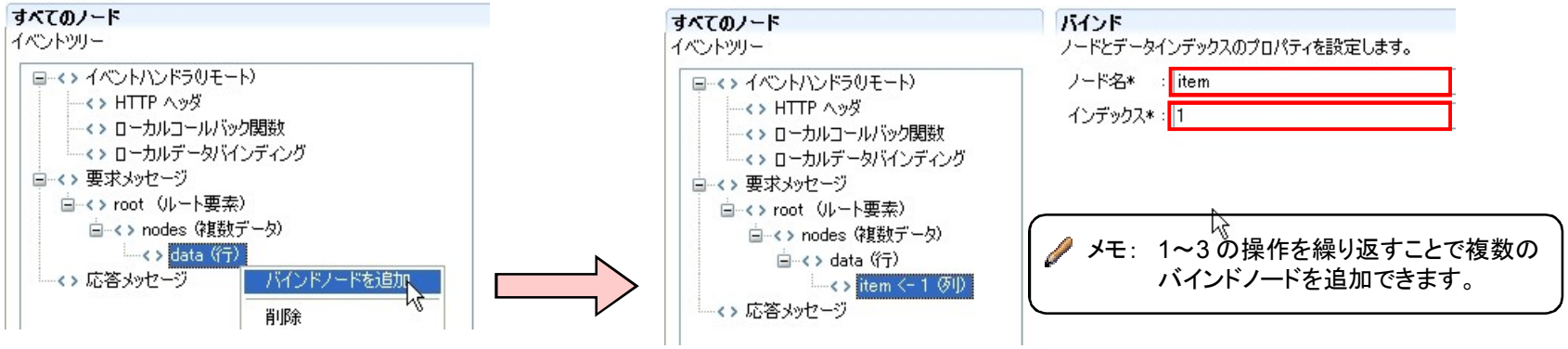
⑤ 複数データ送信の追加 (第2階層)

1. [ソースノード]を右クリックして、コンテキストメニューの[ソースの子要素を追加]を選択します。
2. ソースノードの子要素が追加されます。
3. 追加されたノードを選択して、詳細ページでノード名を編集します。



⑥ 複数データ送信の追加 (第3階層)

1. ソースの子要素ノードを右クリックして、コンテキストメニューから [バインドノードを追加] を選択します。
2. バインドノードが追加されます。
3. 追加されたノードを選択して、詳細ページでノード名とインデックスを編集します。



メモ： 1～3 の操作を繰り返すことで複数のバインドノードを追加できます。

5.4 リモートイベントの編集

5.4.6 応答メッセージの設定

応答メッセージの設定項目は以下の通りです。

項番	主な設定項目			主な設定内容	必須
1	ルート要素			ルートノード名	○
2	子要素	単一データ受信		ノード名、データ格納先	
3		複数データ受信	第一階層	ノード名、データ格納先	
4			第二階層	ノード名	
5			第三階層	ノード名、データ格納キー	

用語:【単一データ受信】

単一の XML 要素に含まれるデータを、コンポーネントや JavaScript 変数のデータに代入する操作です。

例）ラベルのデータ送信

<userName>マスカット太郎</userName>

受信

マスカット太郎

用語:【複数データ受信】

複数の XML 要素で構造化されたデータを、コンポーネントや JavaScript 変数 のデータに代入する操作です。

例）　グリッドのデータ送信

```
<orderList>
  <order>
    <userId>0001</userId>
    <name>マスカット太郎</name>
    <totalAmount>5000</totalAmount>
  </order>
  <order> … </order>
</orderList>
```

受信

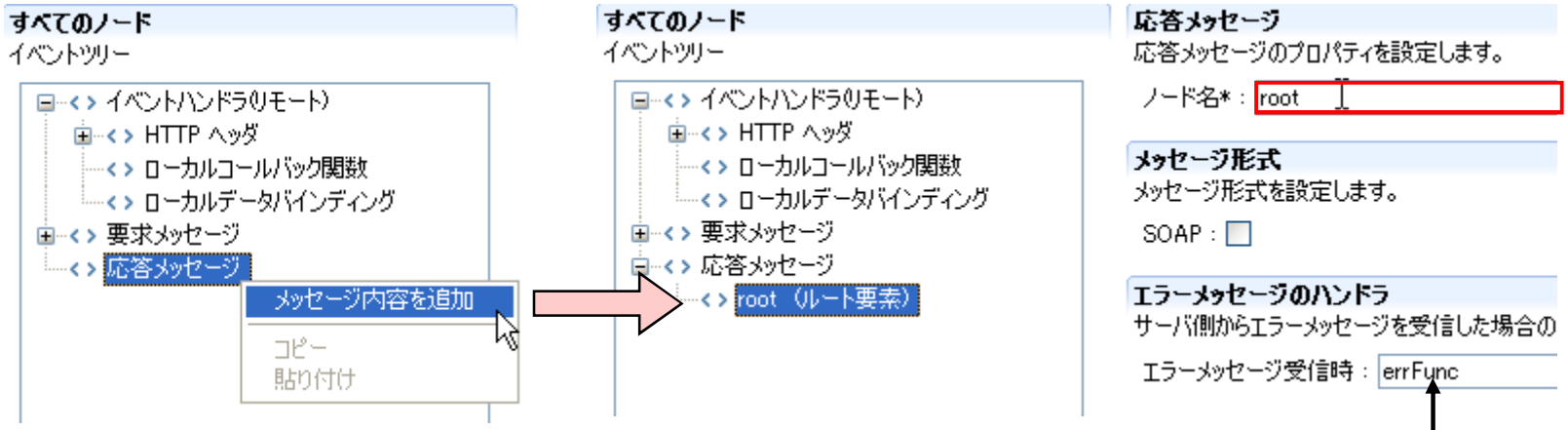
ユーザID	氏名	合計金額
0001	マスカット太郎	5000
0002	マスカット次郎	2500
…	…	

5.4 リモートイベントの編集

5.4.6 応答メッセージの設定

① ルート要素の追加

1. [応答メッセージ] ノードを右クリックして、コンテキストメニューの [メッセージ内容を追加] をクリックします。
2. [ルート要素]ノードが追加されます。
3. 追加されたノードを選択して、詳細ページでノード名を編集します。



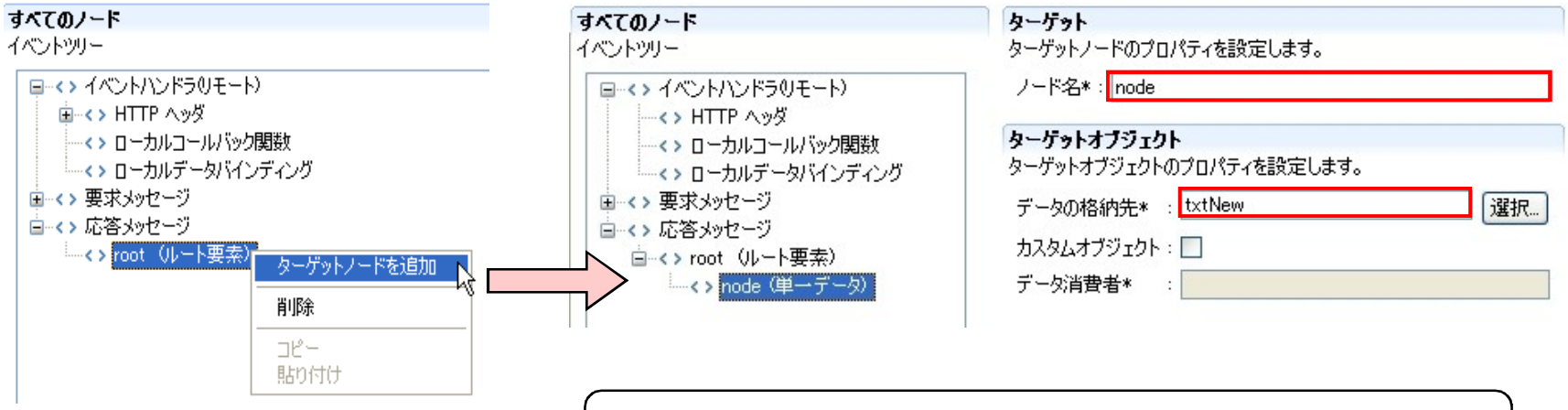
メモ: サーバ側でエラーが発生した場合に受信するエラーメッセージを処理する JavaScript ハンドラ関数 を設定することもできます。

5.4 リモートイベントの編集

5.4.6 応答メッセージの設定

② 単一データ受信の追加

1. [ルート要素]ノードを右クリックして、コンテキストメニューの[ターゲットノードを追加]を選択します。
2. ターゲットノード (単一データ) が追加されます。
3. 追加されたノードを選択して、詳細ページでノード名とデータ格納先を編集します。



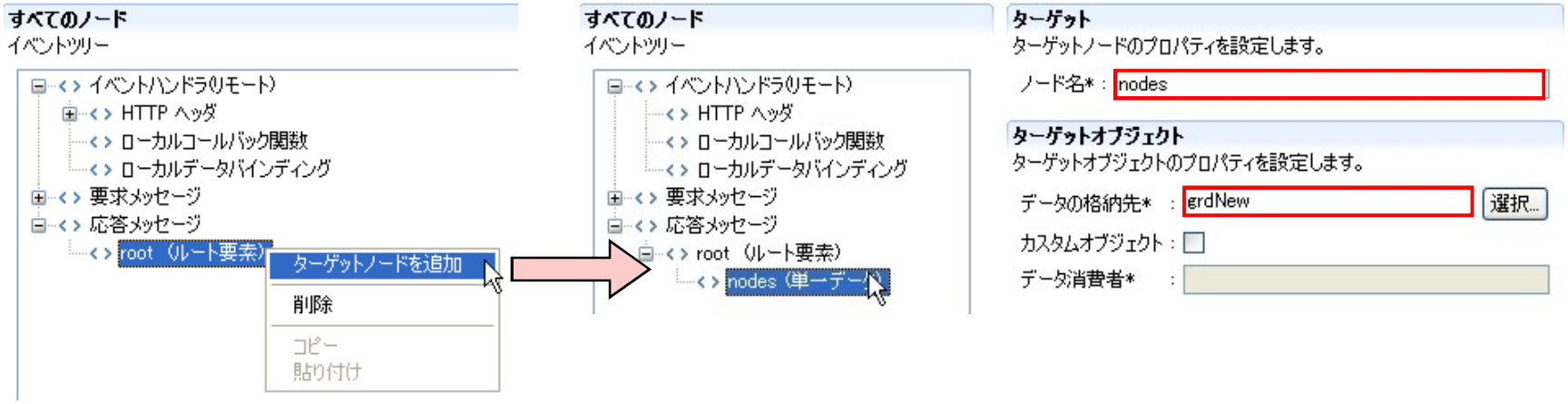
メモ: 詳細ページ右側にある [選択] ボタンをクリックすると、レイアウトエディタからデータ格納先のコンポーネントを選択することができます。

5.4 リモートイベントの編集

5.4.6 応答メッセージの設定

③ 複数データ受信の追加 (第1階層)

1. [ルート要素] ノードを右クリックして、コンテキストメニューから [ターゲットノードを追加] を選択します。
2. 単一データのターゲットノードが追加されます。
3. 追加されたノードを選択して、詳細ページでノード名およびデータ格納先を編集します。



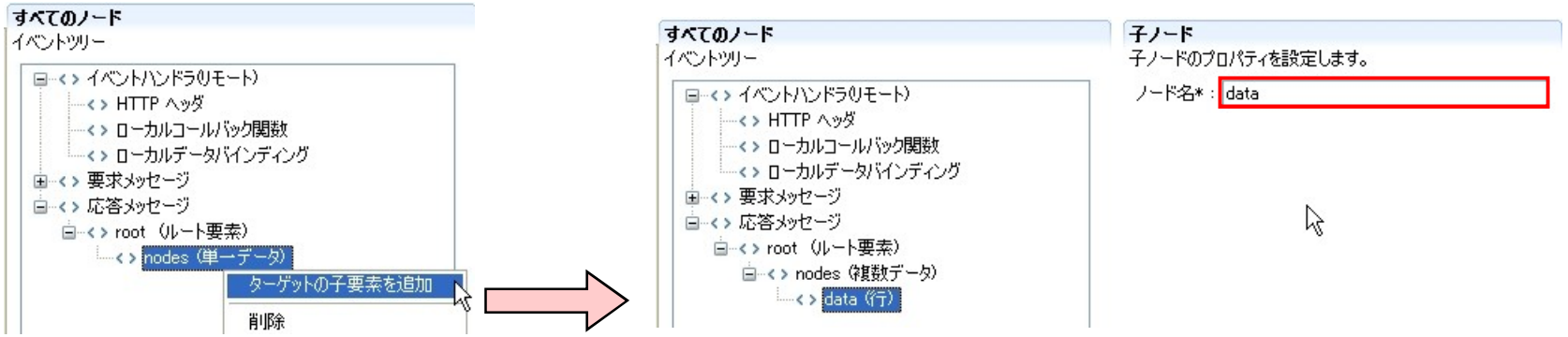
メモ: 詳細ページ右側にある [選択] ボタンをクリックすると、レイアウトエディタからデータ格納先のコンポーネントを選択することができます。

5.4 リモートイベントの編集

5.4.6 応答メッセージの設定

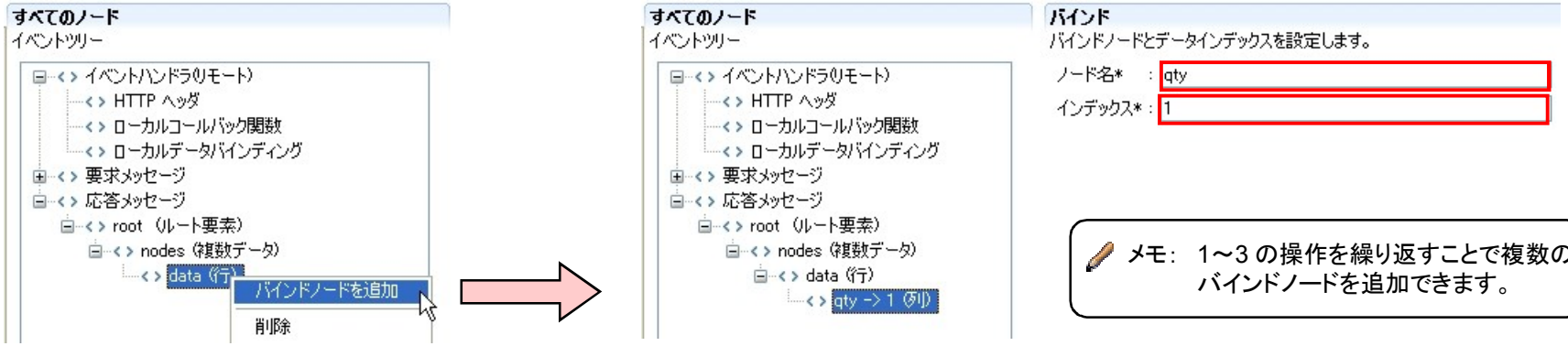
⑤ 複数データ受信の追加 (第2階層)

1. [ターゲットノード] を右クリックして、コンテキストメニューから [ターゲットの子要素を追加] を選択します。
2. ターゲットの子要素ノードが追加されます。
3. 追加されたノードを選択して、詳細ページでノード名を編集します。



⑥ 複数データ受信の追加 (第3階層)

1. ターゲットの子要素ノードを右クリックして、コンテキストメニューから[バインドノードを追加]を選択します。
2. バインドノードが追加されます。
3. 追加されたノードを選択して、詳細ページでノード名とインデックスを編集します。

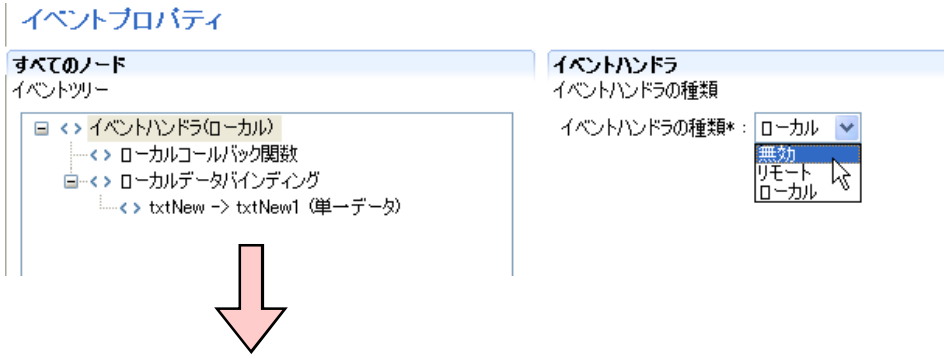


メモ: 1～3 の操作を繰り返すことで複数のバインドノードを追加できます。

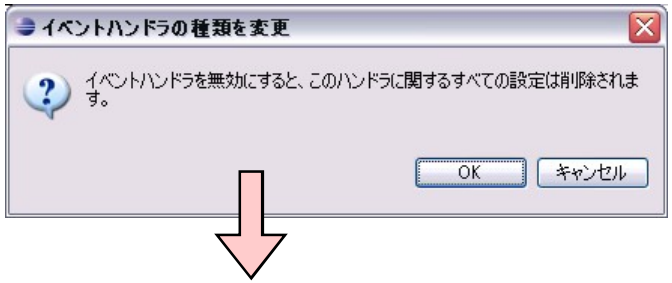
5.5 イベント定義の削除

作成したイベント定義を削除するには以下の手順で行います。

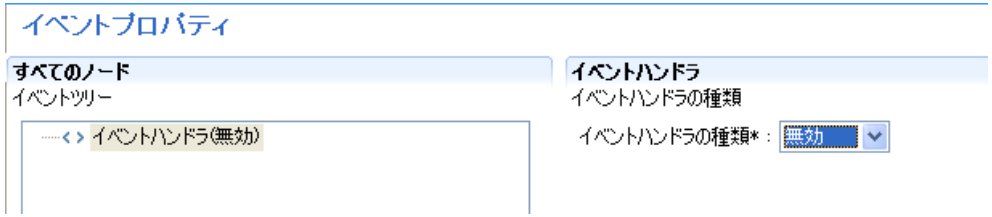
1. イベントツリーから ローカルイベントのイベントハンドラの種類を無効にすると、ローカルイベントが削除されます。



2. 確認ダイアログが表示されるので、[OK] ボタンをクリックします。



3. イベント定義が削除されます。



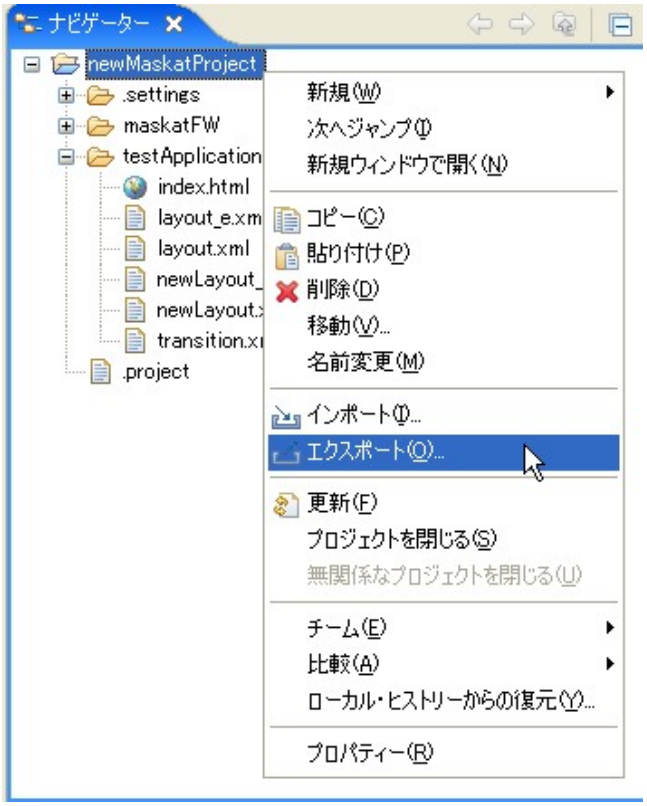
6. XMLスキーマの生成

この章では、イベント定義 XML に記述されているリモートイベントで送受信される XML メッセージのスキーマをファイルに出力する方法について説明します。

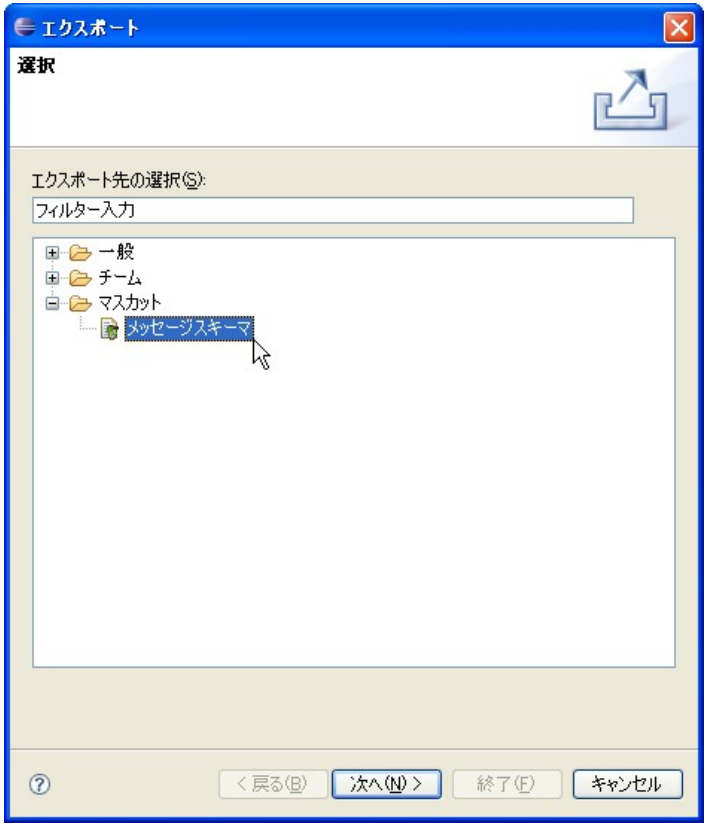
6.1 イベント定義 XML からのスキーマ生成

リモートイベントで送受信する XML メッセージのスキーマは、以下の手順でエクスポートすることができます。

1. マスカットプロジェクトを右クリックして、コンテキストメニューから [エクスポート] を選択します。



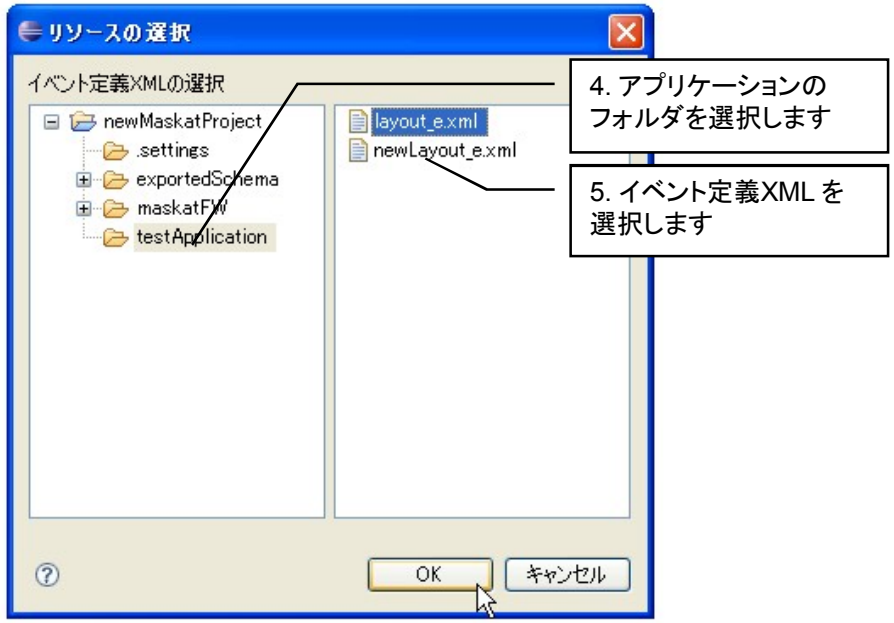
2. エクスポートウィザードの選択ページで、[マスカット] カテゴリの [メッセージスキーマ]を選択し、[次へ] ボタンをクリックします。



6.1 イベント定義 XML からのスキーマ生成

3. [メッセージスキーマエクスポートウィンドウ] ページの上部にある、[参照] ボタンをクリックします。

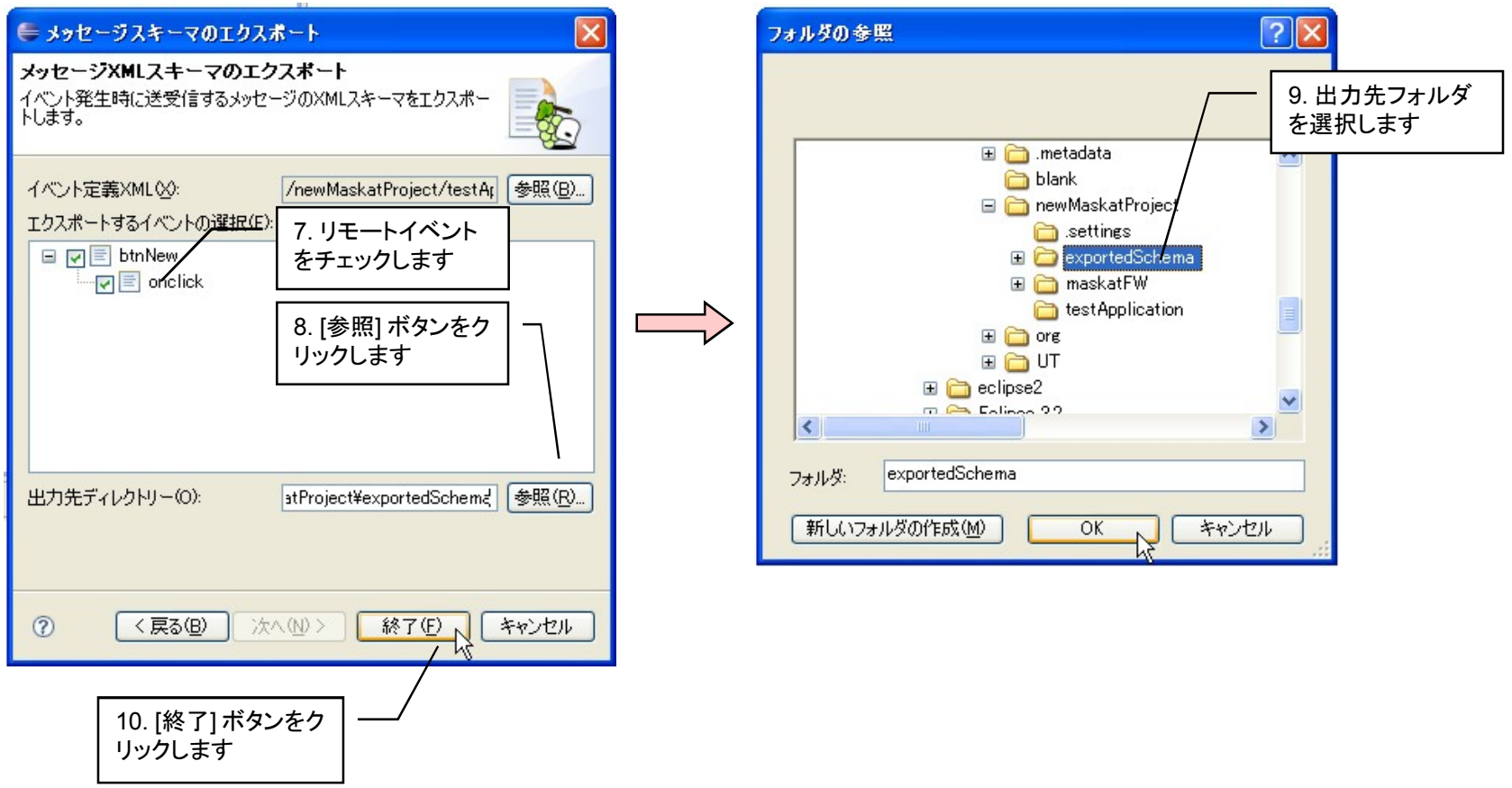
4. リソース選択ウィンドウの左側からマスカットアプリケーションのフォルダを選択します。
5. 右側からエクスポートしたいイベント定義XMLを選択して、[OK] ボタンをクリックします。



メモ: ステップ 1 でイベント定義 XML を選択してコンテキストメニューからウィザードを開始すると、ステップ3 ～5 を省略することができます。

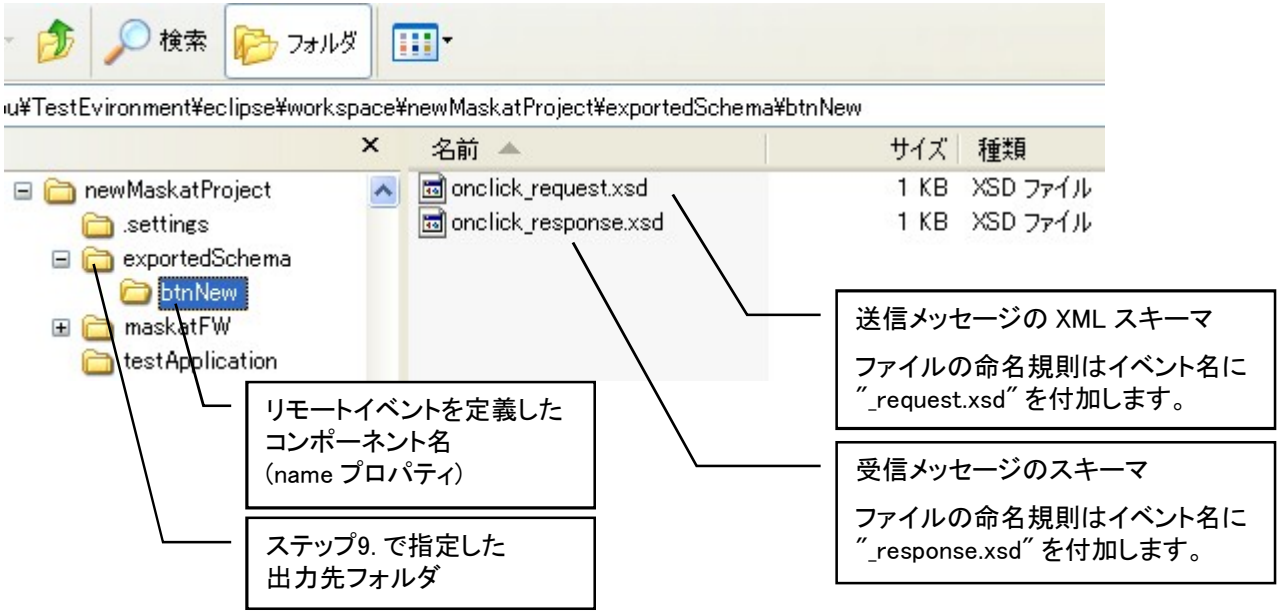
6.1 イベント定義 XML からのスキーマ生成

6. [メッセージスキーマエクスポートウィンドウ] ページに戻ると指定したイベント定義XML に含まれているコンポーネントとイベントの一覧がツリー表示されます。
7. エクスポートしたいリモートイベントのチェックボックスを有効にします。
8. ウィンドウ下部にある [参照] ボタンをクリックします。
9. [フォルダの参照] タイアログで XML スキーマの出力先となるフォルダを選択し、[OK] ボタンをクリックします。
10. [メッセージスキーマのエクスポート] ページに戻り、[終了] ボタンをクリックします。



6.1 イベント定義 XML からのスキーマ生成

ステップ 9. で選択した出力先フォルダには、以下のようにフォルダとファイルが生成されます。



メモ: 「メッセージスキーマをエクスポートすることができません」というエラーが表示された場合は、指定されたフォルダの書き込み権限を確認してください。

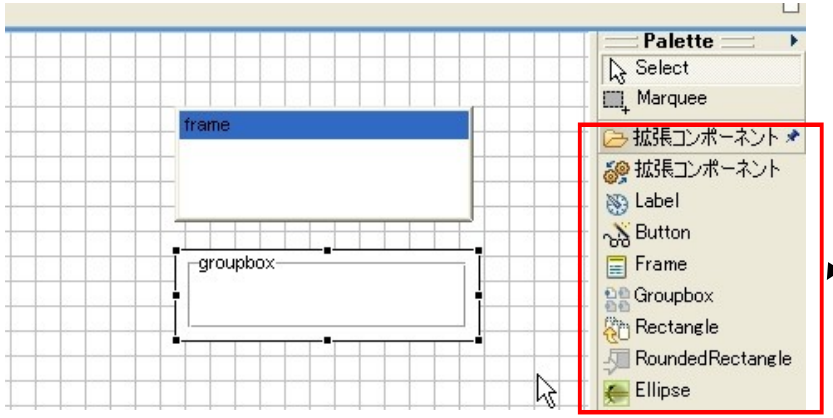
7. 拡張部品

この章では、マスカットフレームワークに追加された拡張部品を
マスカット IDE で表示・編集できるように拡張するための方法を
説明します。

7.1 拡張部品とは

マスカットフレームワークではユーザが独自に部品を拡張することができます。このような部品を拡張部品といいます。

Eclipse版IDEでは拡張部品の作成や編集も容易に対応可能です。



拡張部品を組み込むと左図のようにパレットの生成ボタンが追加されます。通常のコンポーネントと同じようにレイアウトエディタ上で操作することができるようになります。

● 前提条件

拡張部品を利用するにはマスカットフレームワーク上に拡張部品が追加されている必要があります。

開発環境にはPDE（plug-in development environment）とEclipse版IDEが必要です。

● 拡張部品作成手順

作成順番	項目	必須	説明
1	プラグインプロジェクトの作成	○	拡張部品はプラグインとして追加されます。そのためプラグインプロジェクトを作成する必要があります。
2	コンポーネント構文の定義	○	拡張部品のイベントおよびプロパティを設定します。
3	パレットの拡張	○	レイアウトエディタで利用するパレットに拡張部品を表示させます。
4	コンポーネントの形、振る舞いの定義		レイアウトエディタ上に表示されるコンポーネントの形やリサイズパターンを定義します。
5	ビルドとデプロイ	○	ビルドして作成したプラグインをEclipse版IDEにデプロイし拡張コンポーネントを利用できるようにします。

7.2 拡張部品の組み込み手順

拡張部品を Eclipse 版 IDE に組み込む作業フローは以下の通りです。

拡張部品の定義は Eclipse プラグインとして作成します。

レイアウト定義 XML の構文を拡張し、新しいコンポーネントで使用する XML タグや属性を定義します。

Eclipse版IDE のレイアウトエディタのパレットに、定義したコンポーネントを作成するための生成ボタンを配置します。

Eclipse 版IDE のレイアウトエディタ上で、コンポーネントがどのように表示・編集されるかを指定します。

※ このステップは省略可能です。

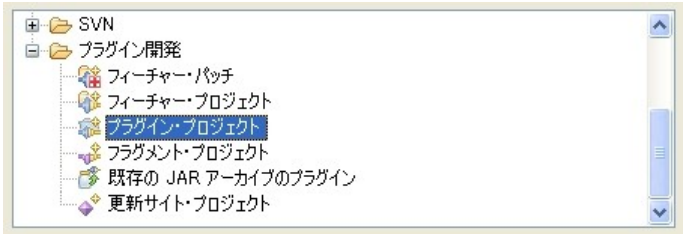
拡張部品をビルドし、Eclipse 版 IDE を拡張するプラグインとしてデプロイします。

7.3 プラグインプロジェクトの作成

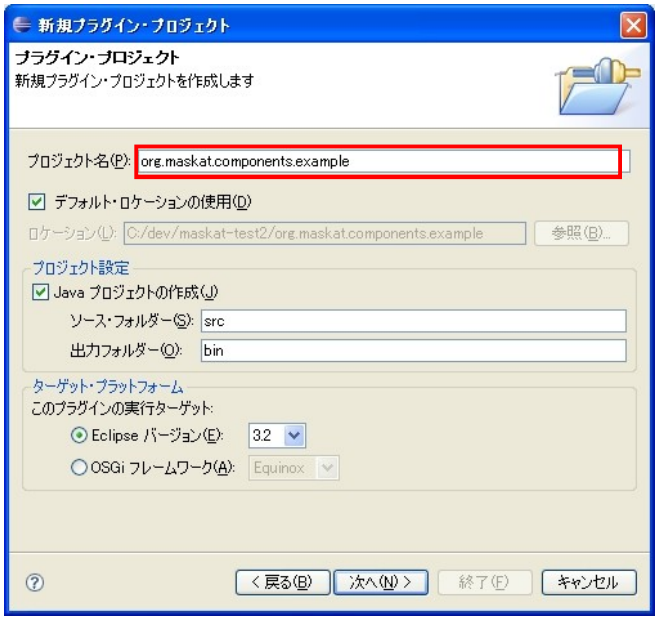
拡張部品を定義するプラグインプロジェクトを以下の手順で作成します。

① [ファイル] > [新規] > [プロジェクト]でウィザード選択画面を表示させます。

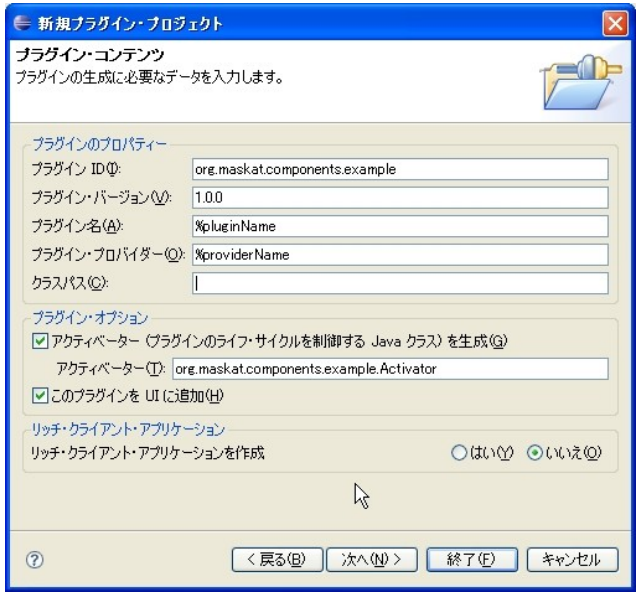
② [プラグイン・プロジェクト]を選択し[次へ]ボタンをクリックしプラグイン・プロジェクト画面を表示させます。



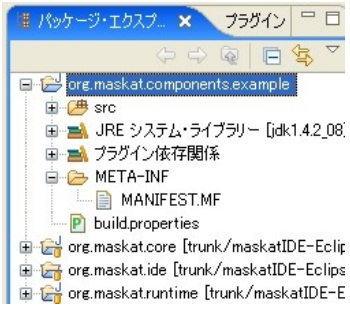
③ 任意のプロジェクト名を入力し[次へ]をクリックし[プラグイン・コンテンツ画面]を表示させます。



④ 任意の値を入力し[終了]をクリックしプラグインプロジェクトの作成を実行します。



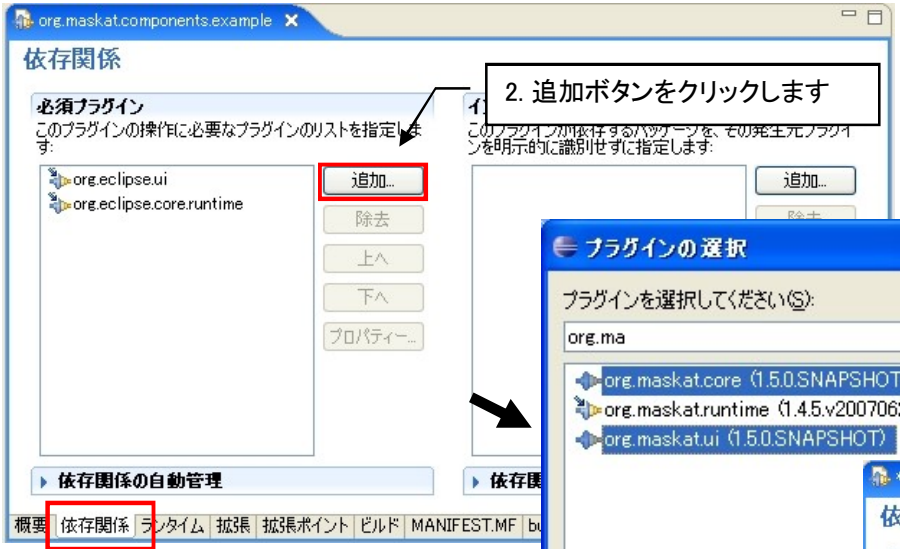
⑤ パッケージエクスプローラにプラグインプロジェクトが作成されることを確認します。



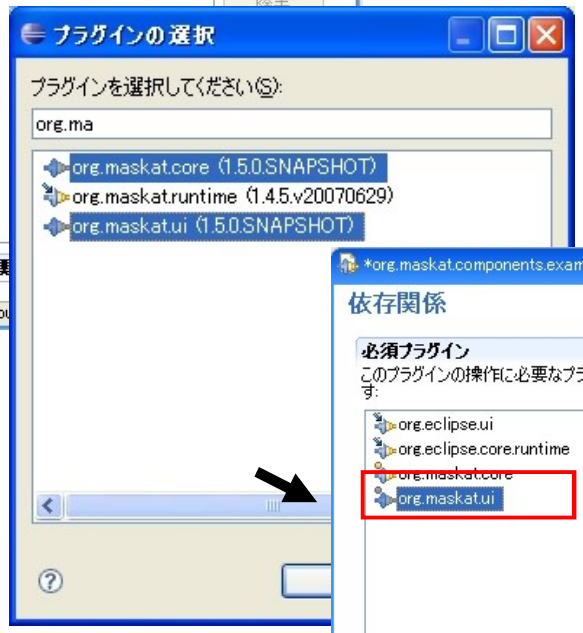
7.3 プラグインプロジェクトの作成

プラグインの設定を行います。プラグイン間の依存関係に “org.maskat.core” と “org.maskat.ui” を含める必要があります。

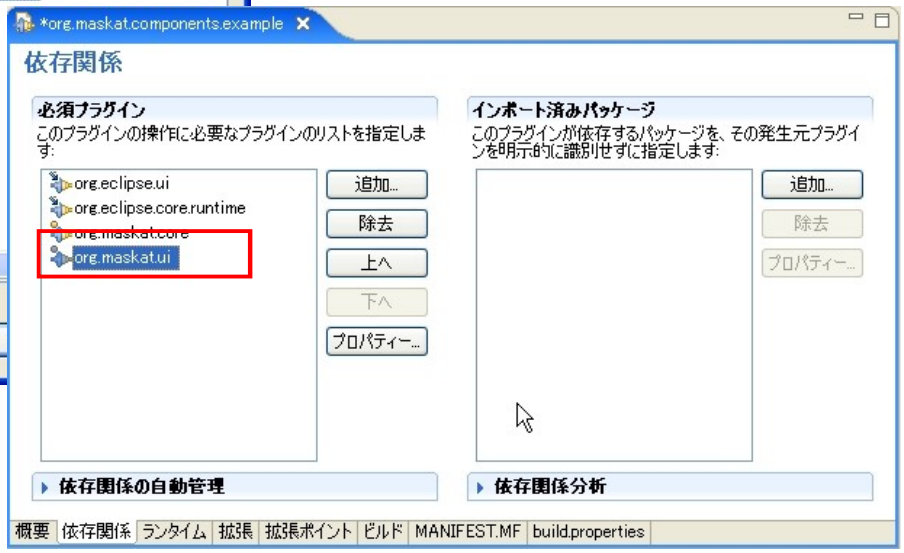
⑥ MANIFEST.MF または plugin.xml を開いてプラグイン・マニフェスト・エディタを実行し、[依存関係] タブの左上にある [追加] ボタンをクリックします。



⑦ プラグイン “org.maskat.core” と “org.maskat.ui” を選択して [OK] ボタンをクリックします。



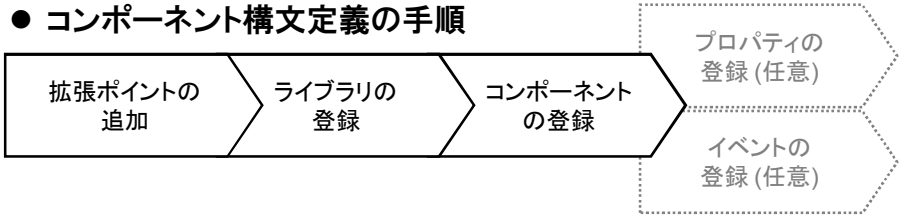
⑧ 必須プラグインに上記2つのプラグインが含まれたことを確認します。



7.4 コンポーネント構文の定義

コンポーネント構文の定義で設定する項目は以下のとおりです。

● コンポーネント構文定義の手順



コンポーネント構文の定義項目一覧

項番	項目	必須	説明
1	library		コンポーネントが属するライブラリです。 このライブラリには複数のコンポーネントを登録することができ、ライブラリでコンポーネントを分類することができます。
2	prefix	○	このライブラリのプリフィクスを指定します。 例）example
3	namespaceURI	○	このライブラリを識別するユニークなURIを指定します。 例）http://maskat.sourceforge.jp/widget/example/2.0.0/
4	class		独自のライブラリクラスを実装する場合指定します。 指定しない場合はorg.maskat.core.layout.custom.ComponentLibraryが使用されます。(※)
5	component		コンポーネントの定義を行います。
6	name	○	このコンポーネントの名前を指定します。この名前はレイアウト定義XMLのタグ名となります。　例）customComponent
7	container		このコンポーネントがコンテナである場合にはtrueにします。
8	size		このコンポーネントのリサイズ条件を指定します。
9	tabFocus		TABキーで移動するタブフォーカスの有無を指定します。
10	property		このコンポーネントが持っているプロパティです。
11	name	○	プロパティの名前を指定します。
12	type	○	プロパティの値のデータ形式を指定します。
13	event		このコンポーネントが持っているイベントです。
14	type	○	このコンポーネントで発生するイベントを指定します。

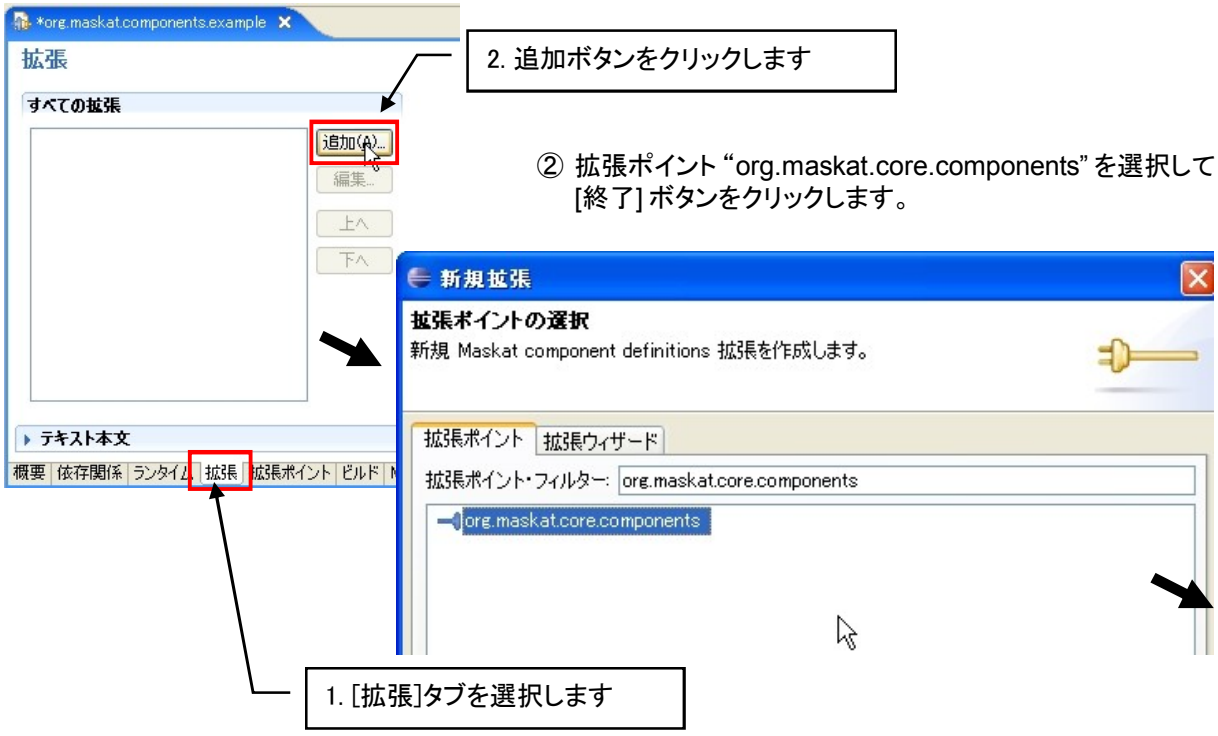
※ この項目の利用には Eclipse や GEF の詳細な知識が必要となるため、本マニュアルの対象外とします。

7.4 コンポーネント構文の定義

7.4.1 拡張ポイントの追加

レイアウト定義 XML の構文にコンポーネントを追加する拡張を行います。
この拡張によって、マスカットエディタで拡張コンポーネントの読み込み／書き込み処理を行うことができるようになります。

① プラグインエディタの拡張タブの [追加] ボタンをクリックします。



② 拡張ポイント “org.maskat.core.components” を選択して [終了] ボタンをクリックします。

③ プラグインエディタの拡張タブに拡張が追加されたことを確認します。

7.4 コンポーネント構文の定義

7.4.2 ライブラリの登録

① コンテキストメニューから [新規] > [library] を選択します。

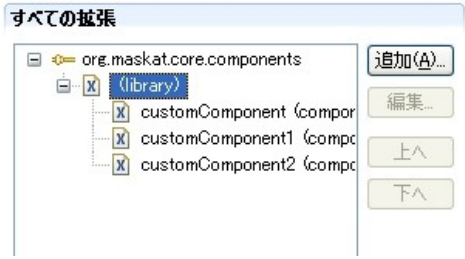
② ライブラリがツリー上に表示されたことを確認し、prefix と namespaceURI を入力します。

7.4 コンポーネント構文の定義

7.4.3 コンポーネントの登録

① コンポーネントを追加したいライブラリを選択し、コンテキストメニューから [新規] > [component] を選択します。

メモ: コンポーネントはライブラリに対して複数登録することができます。



② 追加した component ノードを選択し、コンポーネント名と属性を入力します。
設定された属性値に応じて、コンポーネントに基本プロパティが追加されます。

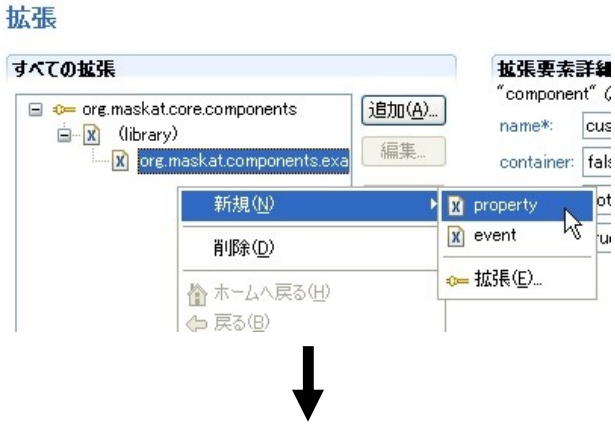
コンポーネントに追加される基本プロパティ

設定項目		値	プロパティ
component		—	name, top, left
	size	none	—
		width	width
		height	height
		both	width, height
	tabFocus	true	tabIndex
		false	—

7.4 コンポーネント構文の定義

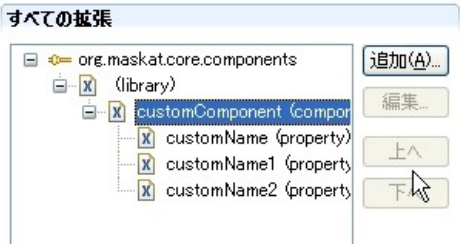
7.4.4 拡張プロパティの登録

① プロパティを追加したいコンポーネントを右クリックし、コンテキストメニューから [新規] > [property] を選択します。



② プロパティがツリー上に表示されたことを確認後、name, typeを入力します。

メモ: プロパティはコンポーネントに対して複数登録することができます。



プロパティの型 (type)

項目	型の説明
boolean	真偽値
int	整数値
float	浮動小数 (単精度)
string	文字列

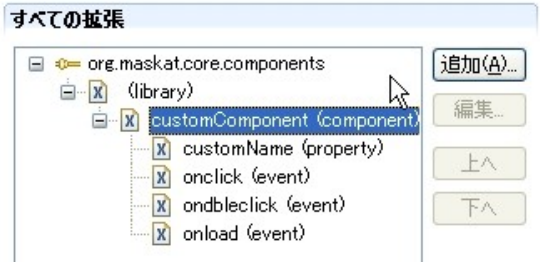
メモ: 基本プロパティ (name, top, left, width, height, tabIndex) は 7.4.3「コンポーネントの登録」で既に追加されているため、ここで登録する必要はありません。

7.4 コンポーネント構文の定義

7.4.5 イベントの登録

① イベントを追加したいコンポーネントを選択し、コンテキストメニューから [新規] > [event] を選択します。

メモ: イベントはコンポーネントに対して複数登録することができます。



② イベントがツリー上に表示されたことを確認後、type を選択します。

選択可能なイベントタイプの一覧

イベント名	
onblur	ondblclick
onCellEdit	onEnableTab
onCellWrite	onfocus
onclick	onload
onclose	onSetDisplay

7.5 パレットの拡張

パレットの拡張で設定する項目は以下のとおりです。

● コンポーネント構文定義の手順

パレット拡張の項目一覧

項番	項目			必須	説明
1	drawer				パレットの引き出しに該当します。
2			id	○	このdrawerのユニークなIDを指定します。
3			label	○	パレットに表示されるラベルを指定します。
4			path		登録するdrawerを指定します。(未指定の場合はルート直下)
5			icon		パレットに表示するアイコンを指定します。
6	creationTool				パレットのコンポーネント生成ボタンに該当します。
7			id	○	このcreationToolのユニークなIDを指定します。
8			namespaceURI	○	このcreationToolで生成するコンポーネントを指定します。コンポーネント構文の定義で設定したコンポーネントをnamespaceURIとnameプロパティで指定します。
9			name	○	
10			label	○	パレットに表示するラベルを指定します。
11			path	○	登録するdrawerのIDを指定します。
12			description		パレットに表示する詳細メッセージを指定します。
13			iconSmall		パレットに表示するアイコン(16x16)を指定します。
14			iconLarge		パレットに表示するアイコン(32x32)を指定します。
15			factory		独自にファクトリクラスを実装する場合に指定します。(※)
16		property			このcreationToolで生成するコンポーネントのプロパティに初期値を設定します。
17			name	○	プロパティの名前を指定します。
18			value	○	プロパティの値を指定します。

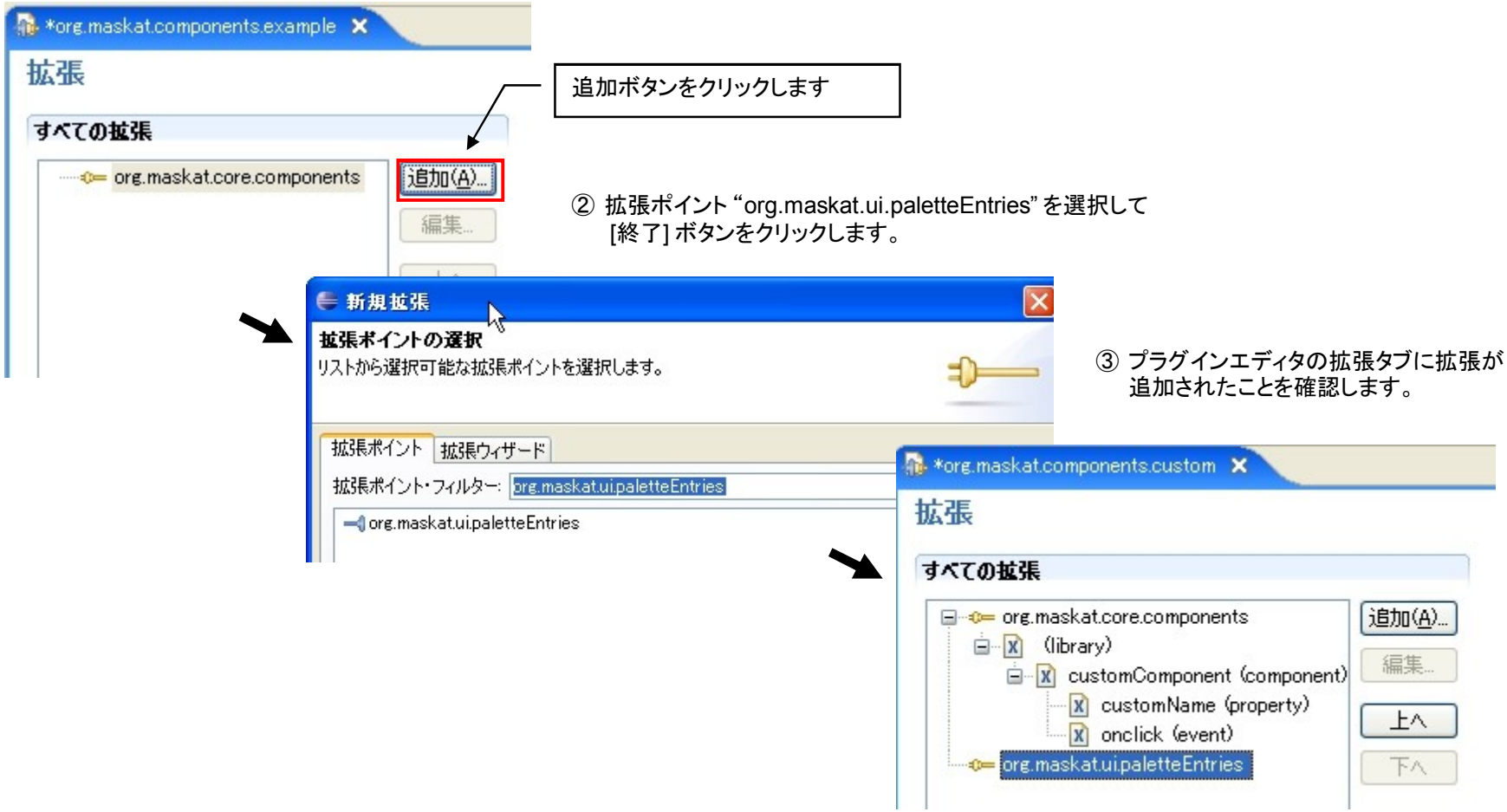
※ この項目の利用には Eclipse や GEF の詳細な知識が必要となるため、本マニュアルの対象外とします。

7.5 パレットの拡張

7.5.1 拡張ポイントの追加

エディタのパレットにボタンを追加する拡張を行います。
この拡張によって、マスカットエディタ(レイアウトエディタ)でコンポーネントを作成できるようになります。

① プラグインエディタの拡張タブの[追加]ボタンをクリックします。

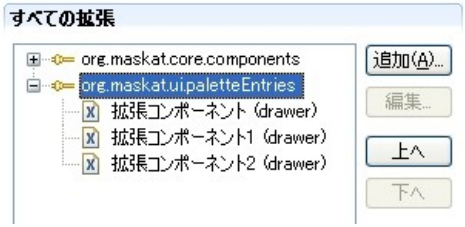


7.5 パレットの拡張

7.5.2 引き出しの登録

① コンテキストメニューから[新規] > [drawer]を選択します。

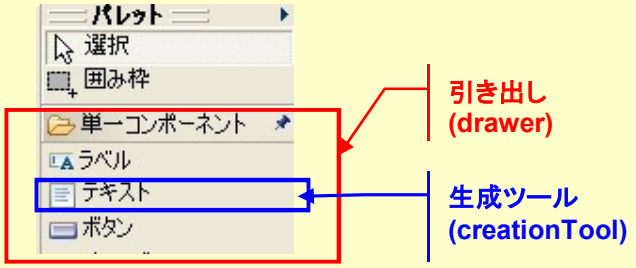
メモ: 複数の引き出し (drawer) をパレットに追加することもできます。



② drawerがツリー上に表示されたことを確認後、各プロパティを入力します。

用語:【引き出し】と【生成ツール】

- **引き出し (drawer) はレイアウトエディタのパレットにツールボタンをグループ化して格納する表示領域です。**
- **生成ツール (creationTool) は新しいコンポーネントを作成して配置するためのツールボタンです。**

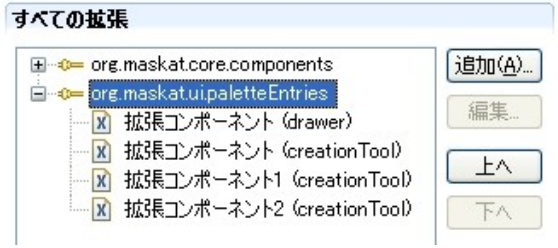


7.5 パレットの拡張

7.5.3 生成ツールの登録

① 拡張ポイントのコンテキストメニューから [新規] > [creationTool] を選択します。

メモ: 複数の生成ツール (creationTool) を引き出しに追加することもできます。



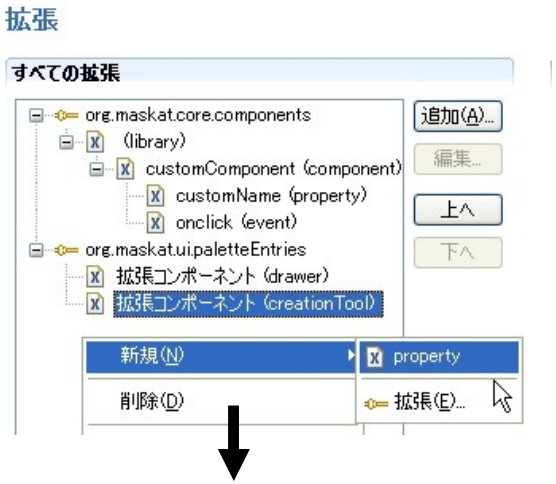
② creationTool がツリー上に表示されたことを確認し、各項目を設定します。

新規追加時にはfactoryに初期値が自動的に入力されます。ファクトリクラスを実装しない場合には空白にしてください。

7.5 パレットの拡張

7.5.4 プロパティの初期値の登録

① プロパティの初期値を登録したいコンポーネントのcreationToolを選択し、
コンテキストメニューから [新規] > [property] を選択します。



メモ: 生成ツール (creationTool)の子要素として複数の
プロパティの初期値を登録することができます。

② プロパティがツリー上に表示されたことを確認後、name, valueを入力します。

プロパティの初期値について

プロパティ名	型	必須	説明
name	String	○	オブジェクト名の初期値 (連番が自動的に付加されます)
width	int		コンポーネントの幅
height	int		コンポーネントの高さ
任意のプロパティ	ー		任意のプロパティに初期値を指定可能

7.6 コンポーネントの形、振る舞いの定義

●コンポーネントの形について

幅、高さの両方を持つ拡張コンポーネントは四角い枠で表示されます（図1）。幅や高さを持たないコンポーネントをレイアウト上に表示させると定義されていないプロパティが無視されます。（図2,3）

コンポーネントの形の定義を行うことで幅、高さを持たないコンポーネントの表示サイズを変更することができます。（図4）

デフォルトでは拡張部品は以下の形になります:

図1: 幅、高さを持つ場合

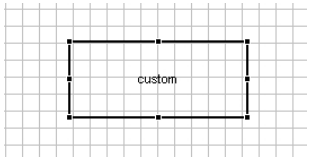


図2: 幅、高さの両方を持たない場合

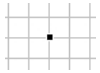


図3: 高さを持たない場合

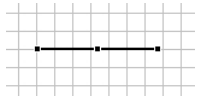


図4: コンポーネントの形の定義を行った場合の例

●コンポーネントの振る舞いについて

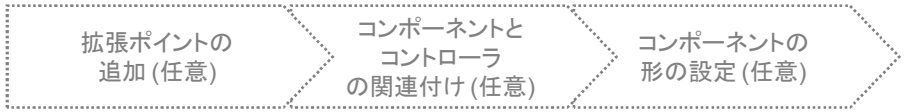
デフォルトでは拡張部品は以下の振る舞いとなります。
コンポーネントの振る舞いの定義を行うことで独自の振る舞いを実装することもできます。

項番	種別	説明
1	削除	コンポーネントを削除することができます。 コンテナの場合はコンテナ内のコンポーネントも全て削除されます。
2	移動	ドラッグ &ドロップで任意の座標に移動することができます。
3	リサイズ	コンポーネントのサイズを変更することができます。 変更できるプロパティはコンポーネント構文のsizeプロパティで設定します。（P.73) ・widthが変更可能であればドラッグ &ドロップでリサイズできます。 ・heightが変更可能であればドラッグ &ドロップでリサイズできます。
4	親子関係	コンポーネント構文のcontainerプロパティ（P.70)がtrueであれば子コンポーネントを持つことができます。

7.6 コンポーネントの形、振る舞いの定義

コンポーネントの形、振る舞いの定義で設定する項目は以下のとおりです。

● コンポーネントの形、振る舞いの定義手順



コンポーネントの形、振る舞いの設定項目の一覧

項番	項目			必須	説明
1	editPart				コンポーネントの形、振る舞いを扱えるコントローラの定義を行います。
2		namespaceURI		○	形、振る舞いを変更したいコンポーネントを定義します。コンポーネント構文の定義で設定したコンポーネントをnamespaceURIとnameプロパティで指定します。
3		name		○	
4		icon			アウトラインビューに表示されるアイコンを指定します。省略した場合、デフォルトのアイコンが表示されます。
5		class			独自のコントローラクラスを実装する場合に指定します。(※)
6		figure			コンポーネントの表示図形の定義を行います。
7			type		このコンポーネントの基本図形です。
8			width		このコンポーネントの幅を定義します。
9			height		このコンポーネントの高さを定義します。
10			factory		独自のコンポーネントの表示図形を実装する場合に指定します。(※)

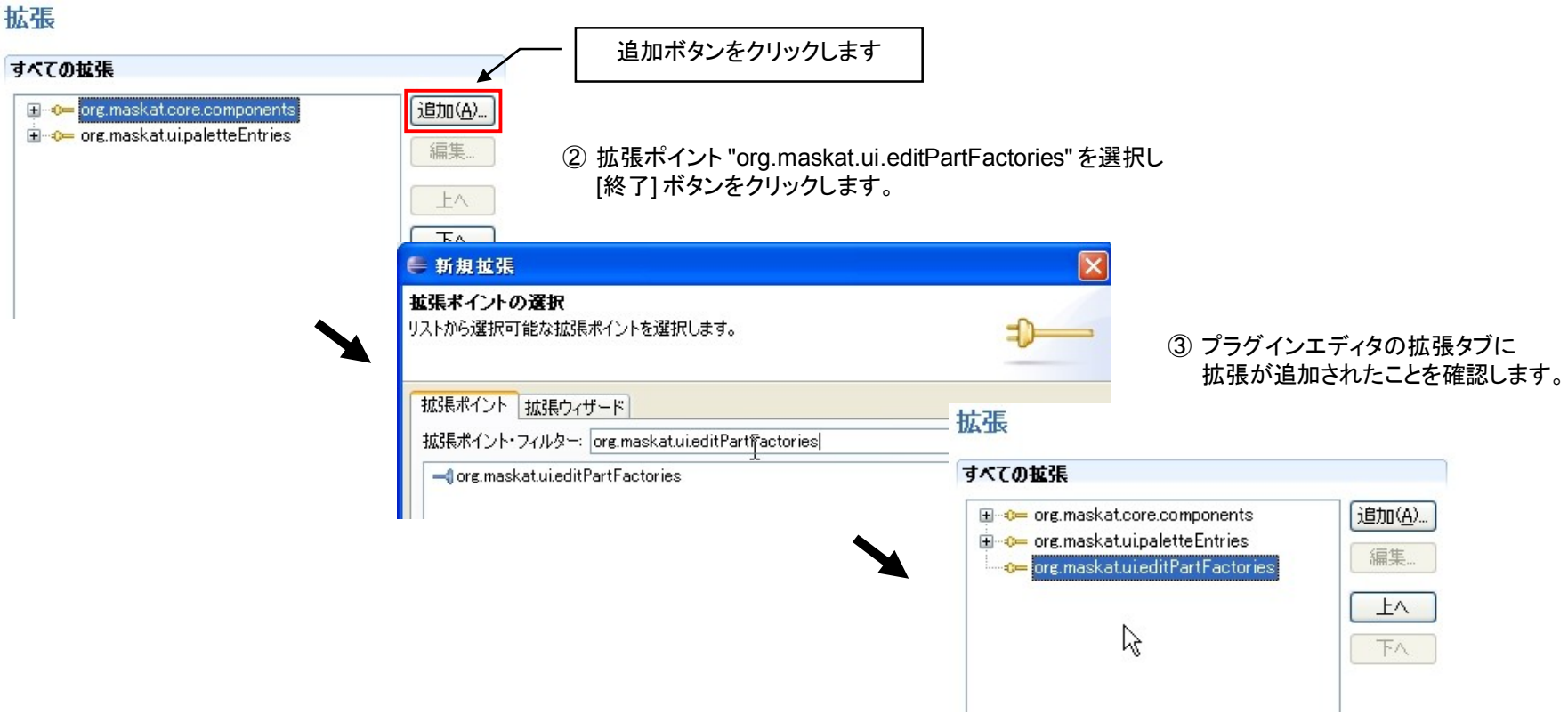
※ これらの項目の利用には Eclipse や GEF の詳細な知識が必要となるため、本マニュアルの対象外とします。

7.6 コンポーネントの形、振る舞いの定義

7.6.1 拡張ポイントの追加

コンポーネントの形や振る舞いを変更する拡張を行います。
この拡張によって、マスカットエディタ(レイアウトエディタ)におけるコンポーネントの表示や編集操作を定義することができます。

①　プラグインエディタの拡張タブの[追加]ボタンをクリックします。

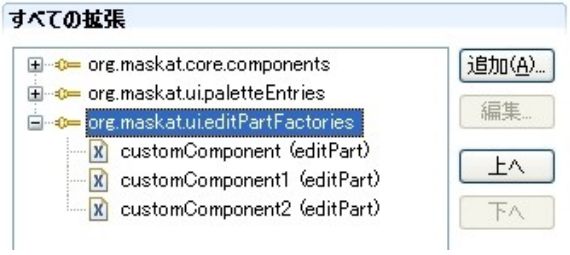


7.6 コンポーネントの形、振る舞いの定義

7.6.2 コンポーネントとコントローラの関連付け

① コンテキストメニューから[新規] > [editPart]を選択します。

メモ: 定義した拡張コンポーネントごとに、コントローラ(editPart)を登録することができます。



② editPartがツリー上に表示されたことを確認後、拡張コンポーネントのnamespaceURIとnameを入力します。

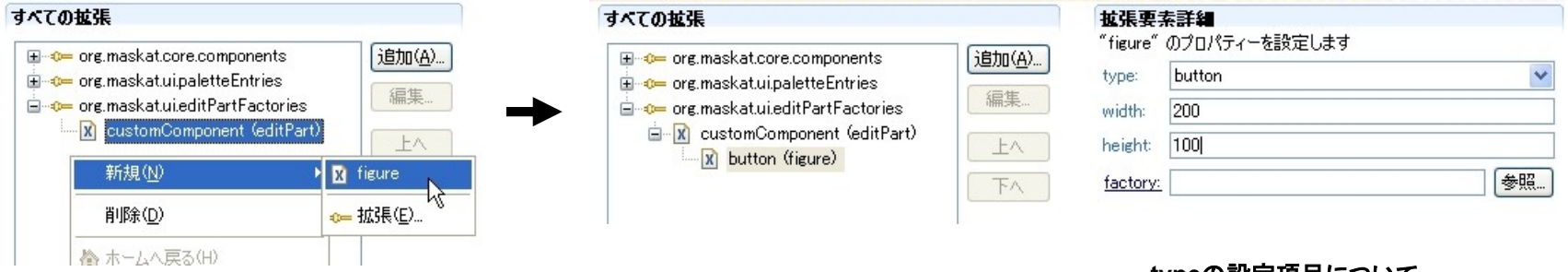
コンポーネントの振る舞いを変更したい場合、実装クラスを指定してください。

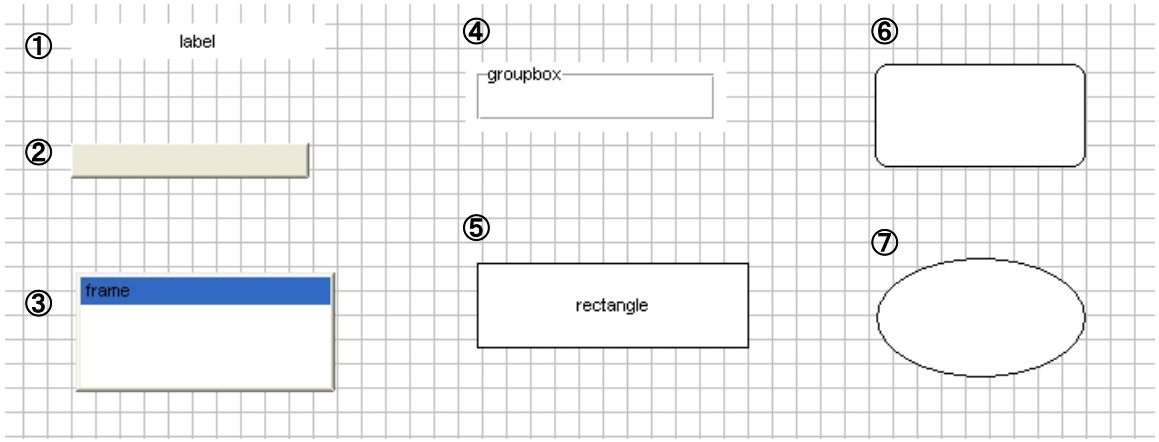
7.6 コンポーネントの形、振る舞いの定義

7.6.3 コンポーネントの形の設定

① コンポーネントの形を指定したいeditPartを選択し、コンテキストメニューから[新規] > [figure]を選択します。

② figureがツリー上に表示されたことを確認後、各プロパティを入力します。





typeの設定項目について

	プロパティ名
①	label
②	button
③	frame
④	groupbox
⑤	rectangle
⑥	roundedRectangle
⑦	ellipse

メモ：typeが未指定（空白）の場合はデフォルトの枠で表示されます。

width, height項目について

コンポーネント定義のsize項目	figureに設定する必要のあるプロパティ
none	width, height
width	height
height	width

figure のwidth, heightはコンポーネントのプロパティに width や height がない場合の表示サイズを指定します。

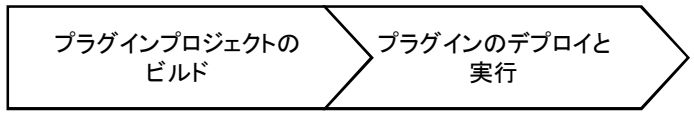
コンポーネント構文の定義で設定するsizeプロパティ(P.73)によりfigureに定義する必要があるプロパティが変わります。

7.7 ビルドとデプロイ

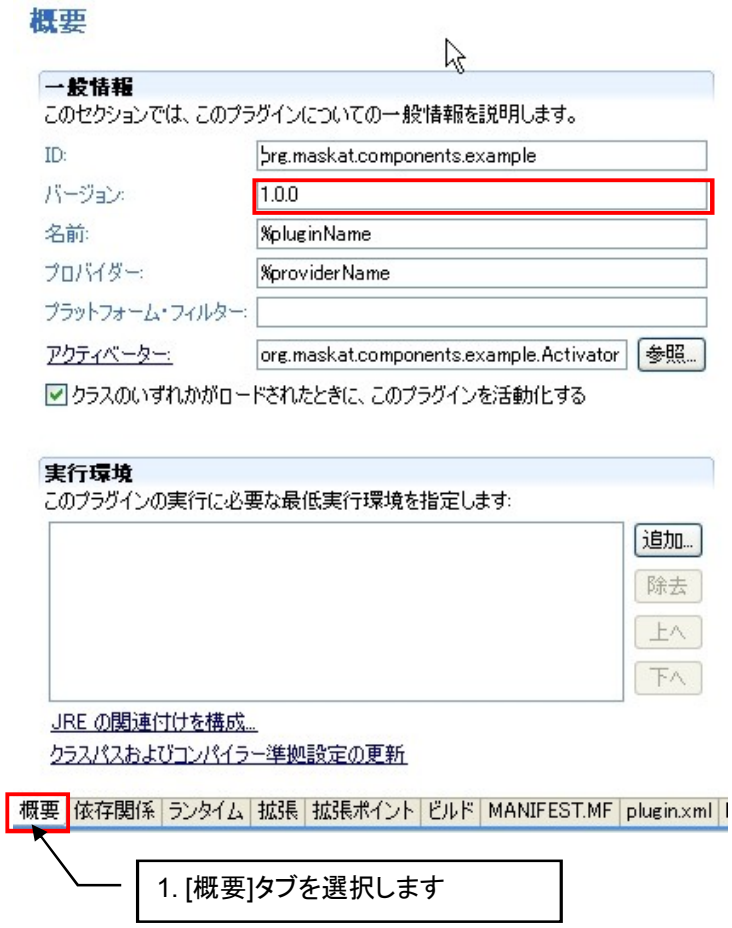
7.7.1 プラグインプロジェクトのビルド

プラグイン定義をビルドして、拡張コンポーネントプラグインとして出力します。

● ビルドとデプロイの手順



① プラグイン・マニフェスト・エディタの [概要] タブでプラグインのバージョンを指定します。



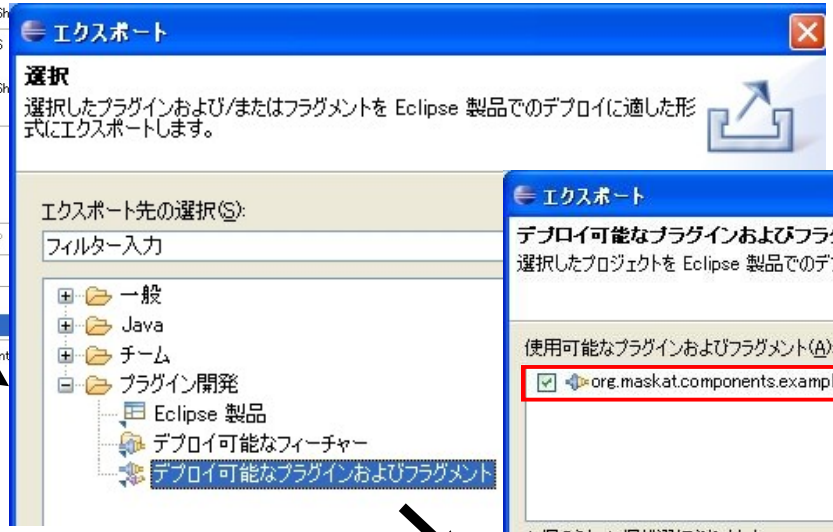
② [ビルド] タブでplugin.xmlが [バイナリー・ビルド] に含まれているか確認し、含まれていない場合は追加します。また、icons/フォルダやplugin.propertiesを作成した場合も同様に追加してください。

7.7 ビルドとデプロイ

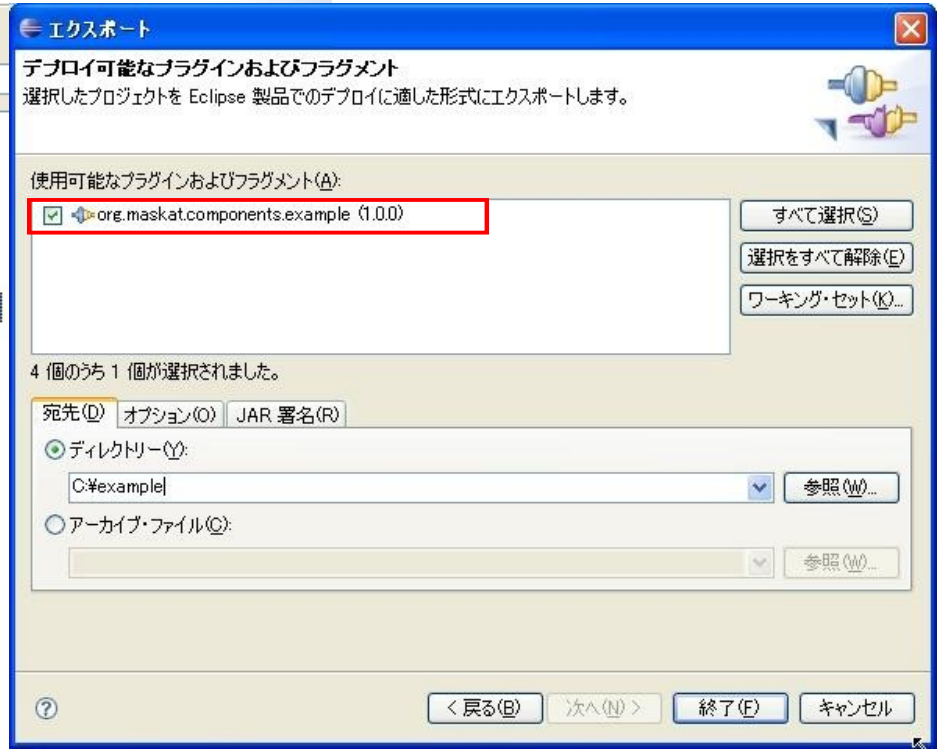
7.7.1 プラグインプロジェクトのビルド

③ ファイルメニューから[エクスポート]を選択します。

④ [デプロイ可能なプラグインおよびフラグメント]を選択し[次へ]ボタンをクリックします。



⑤ プラグインと出力ディレクトリを選択し[終了]ボタンでビルドを行います。



⑥ 出力ディレクトリにpluginsフォルダが作成され中にjarファイルが作成されていることを確認します。

例）C:¥example¥plugins¥org.maskat.components.example_1.0.0.jar

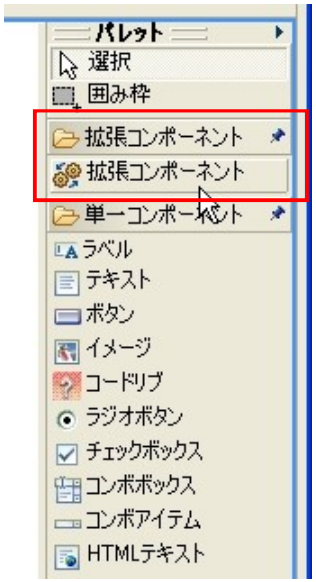
7.7 ビルドとデプロイ

7.7.2 プラグインのデプロイと実行

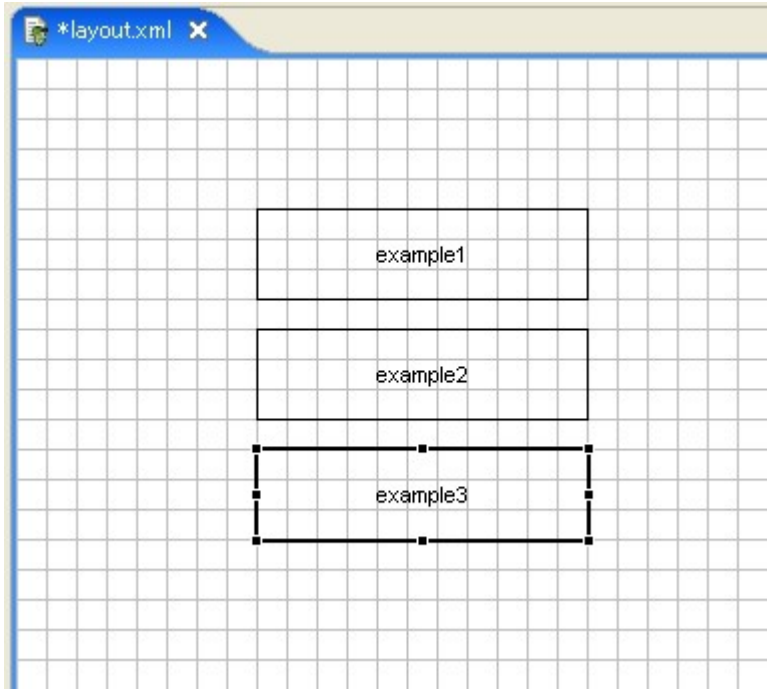
作成したプラグインをマスカットIDEへデプロイし、動作確認を行います。

① マスカットIDE の eclipse¥plugins フォルダにビルドしたプラグインをコピーします。

② マスカットIDEを起動してレイアウト定義 XML を開いたとき、パレットに拡張部品が追加されていることを確認します。



③ パレットから拡張部品を選択しレイアウトエディタ上に正常に配置および操作できることを確認します。



メモ： Eclipseに同一バージョン番号のプラグインを上書きすると、プラグインが更新されない場合があります。
その場合は起動オプションに-cleanを追加してEclipse を再起動してください。

（コマンドプロンプト上での実行例）

```
> C:¥eclipse¥eclipse.exe -clean
```