実践編：複雑なアプリケーションを作ってみよう

実践編では、これまでより少し複雑なアプリケーションを構築します。より実用的なアプリケーションを作成し、アプリケーション構築のコツを習得しましょう。

◆目次

実践編： 複雑なアプリケーションを作ってみよう 1

LESSON.11 ガントチャートを使ってみよう 2

Step.1 ガントチャートとは？ 2

Step.2 ガントチャートで使用するデータ 2

Step.3 テーブルデータとガントチャートを表示する 3

Step.4 テーブルとガントチャートを連動させる 16

LESSON.12 いろいろな画像ファイルを表示する 23

Step.1 利用できる画像ファイルの種類 23

Step.2 画像ファイルの入力 23

Step.3 新しいフレームの利用 31

Step.4 イメージビューワーの設定変更 37

Step.5 複合コンポーネントによる階層化 40

Step.6 複合コンポーネントの利用 44

Step.7 機能を追加する 58

LESSON.13 電卓の機能を拡張してみよう 74

Step.1 任意桁の計算 74

Step.2 ［0］ボタンと［.（小数点）］ボタンを追加 86

Step.3 四則演算 94

Step.4 複合コンポーネントによる階層化 119

Step.5 複合コンポーネントの利用 121

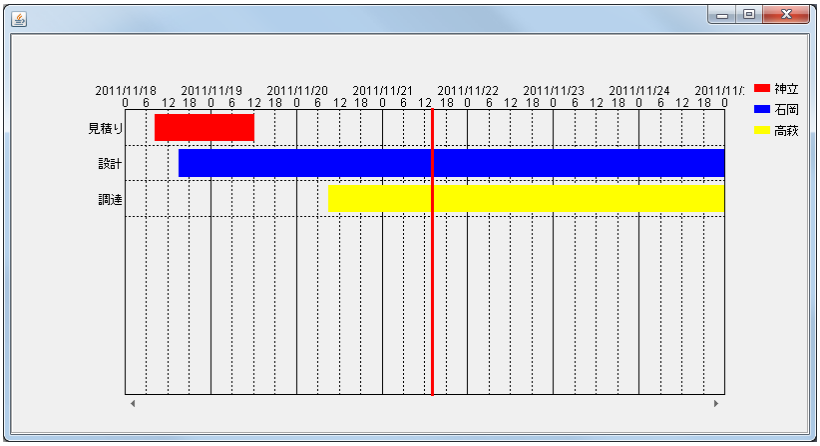
Lesson.11 ガントチャートを使ってみよう

MZ Platform の標準コンポーネントの「ガントチャート」を使ってみましょう。

Step.1 ガントチャートとは？

工場などで人や工程などの管理に用いられる帯状のグラフです。横軸では時間などの期間、縦軸では人・製造工程・作業工程等を表し、それぞれ開始日（時間）、完了日（時間）といった情報を帯状に示します。

MZ Platform では標準コンポーネントでガントチャートを提供しています。



Step.2 ガントチャートで使用するデータ

ガントチャートは［テーブル］コンポーネントのデータを元に作成します。最初にガントチャートに必要なデータが入力されている［テーブル］コンポーネントを作成する必要があります。

１）ガントチャート用データの作成

ガントチャート用データをテーブルに入力します。

ガントチャートの帯の部分のデータは、「開始」データと「終了」データが必要になります。

つまり１つの帯に２列必要です。

ここには開始時刻と終了時刻が入力されています。

WorkName	神立		石岡		高萩	
見積り	2011/11/18 8:30:00	2011/11/19 12:00:00				
設計			2011/11/18 15:00:00	2011/11/25 9:00:00		
調達					2011/11/20 9:00:00	2011/11/25 17:00:00

開始時刻

終了時刻

神立さんの作業（見積り）の開始時刻と終了時刻が入力されています。

２）ガントチャートで使用できるデータ

列　名	データの意味
１列目	各行の項目名となる文字列データです。 グラフにした場合、この項目が軸の作業名として表示されます。
２列目以降（２列１組）	実際にガントチャートで表示する開始・終了日時のデータです。 データ型は日付と文字列が使えます。 “年/月/日 時:分:秒”という形式で書きます。 例えば、“2015/05/15 15:00:00（数字や記号、空白などすべて半角で入力)”となります。

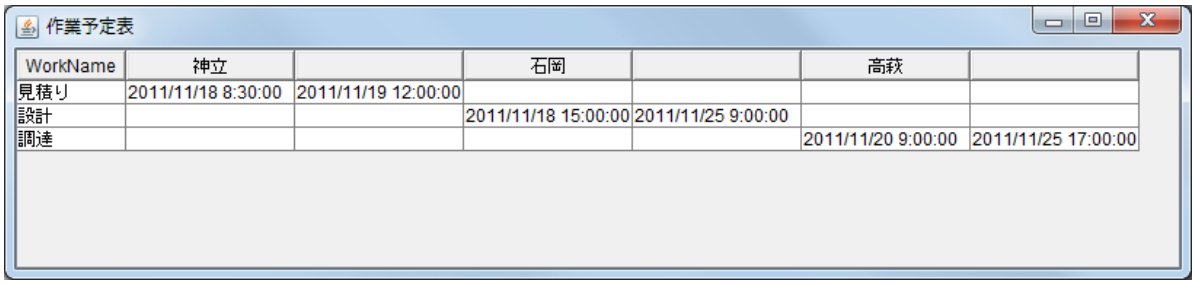
Step.3 テーブルデータとガントチャートを表示する

実際に操作して、テーブルデータとガントチャートを表示しましょう。

１）ガントチャートに必要なデータを入力する

テーブルコンポーネントを準備してガントチャートに必要なデータを入力しましょう。

完成図 テーブルデータを用意します。



準 備 ここでは以下のコンポーネントを使用します。

コンポーネント名	必要数	
■アプリケーション	(1)	
■フレーム	1	［画面構成部品］－［ウィンドウ］－［フレーム］
■テーブル	1	［画面構成部品］－［テーブル］－［テーブル］

操 作

① 必要なコンポーネントを追加します。
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［ウィンドウ］－［フレーム］、
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［テーブル］－［テーブル］
とクリックします。

接続確認

コンポーネント同士の接続を確認します。

開始

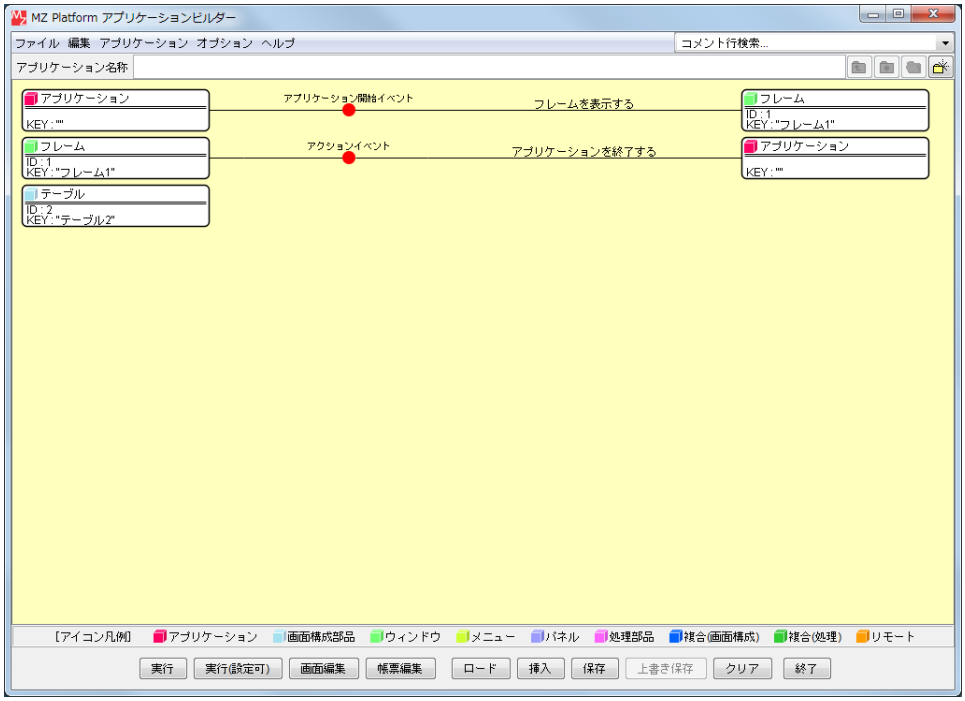
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■アプリケーション
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■フレーム（ID:1）
起動メソッド	フレームを表示する()

終了

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■フレーム（ID:1）
発生イベント	アクションイベント
接続先コンポーネント	■アプリケーション
起動メソッド	アプリケーションを終了する()

操 作

① ［フレーム］コンポーネントと［アプリケーション］コンポーネントを接続します。

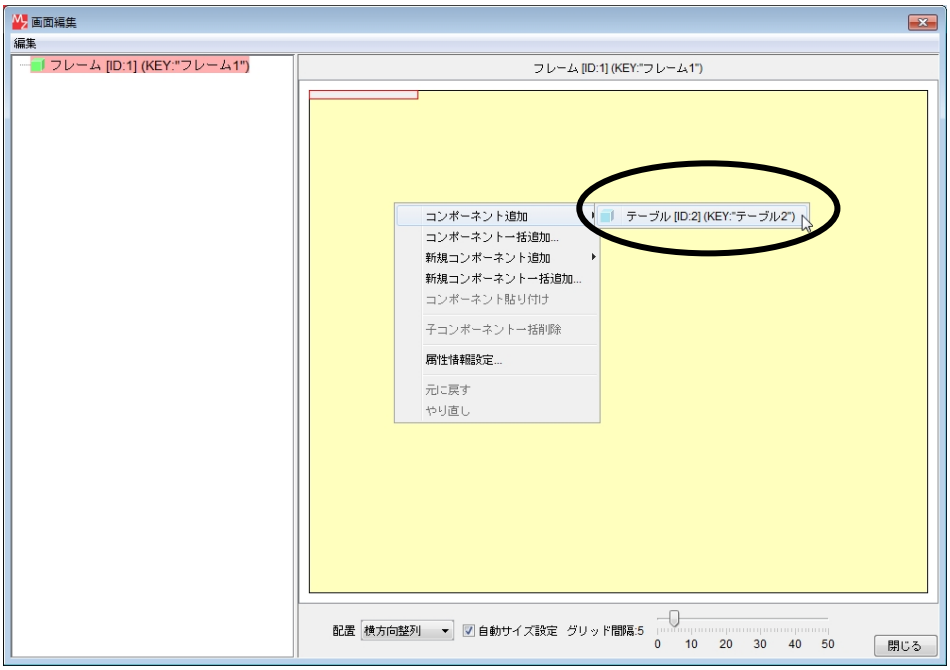


画面編集

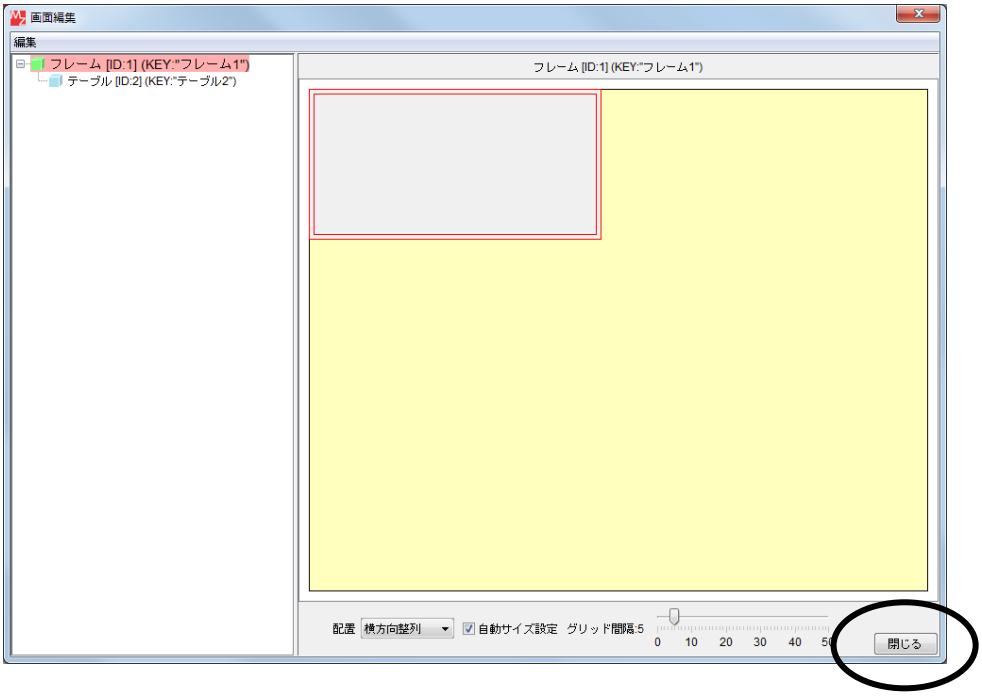
ウィンドウ（フレーム）に［テーブル］を配置しましょう。

① 画面編集ツールボタンをクリックし、［画面編集］画面に入ります。

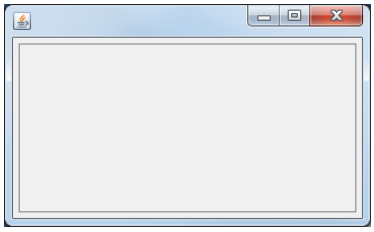
② ［テーブル］コンポーネントをフレームに貼り付けます。
［画面編集］画面上で右クリック－［コンポーネント追加］－［テーブル］コンポーネントとクリックします。



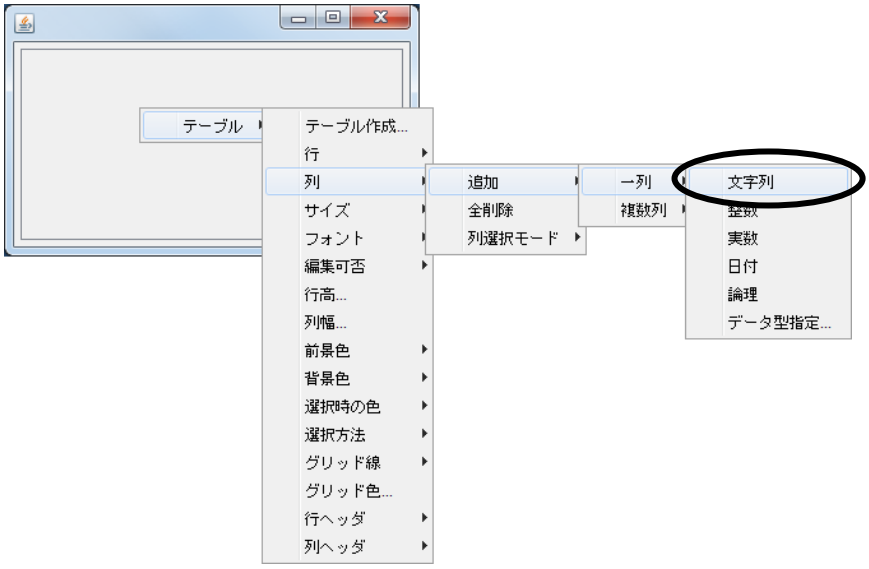
③ 追加できたら閉じるをクリックし、ビルダー画面に戻ります。



④　テーブルの画面が完成したことを確認します。
実行（設定可）で実行します。

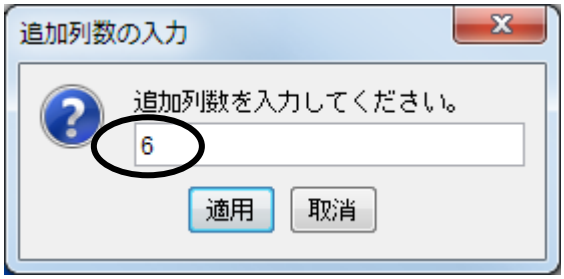


⑤　テーブルを作成します。
項目名を入力する列を文字列型で作成します。
実行（設定可）で実行したテーブル上で右クリックします。
［テーブル］－［列］－［追加］－［一列］－［文字列］をクリックします。

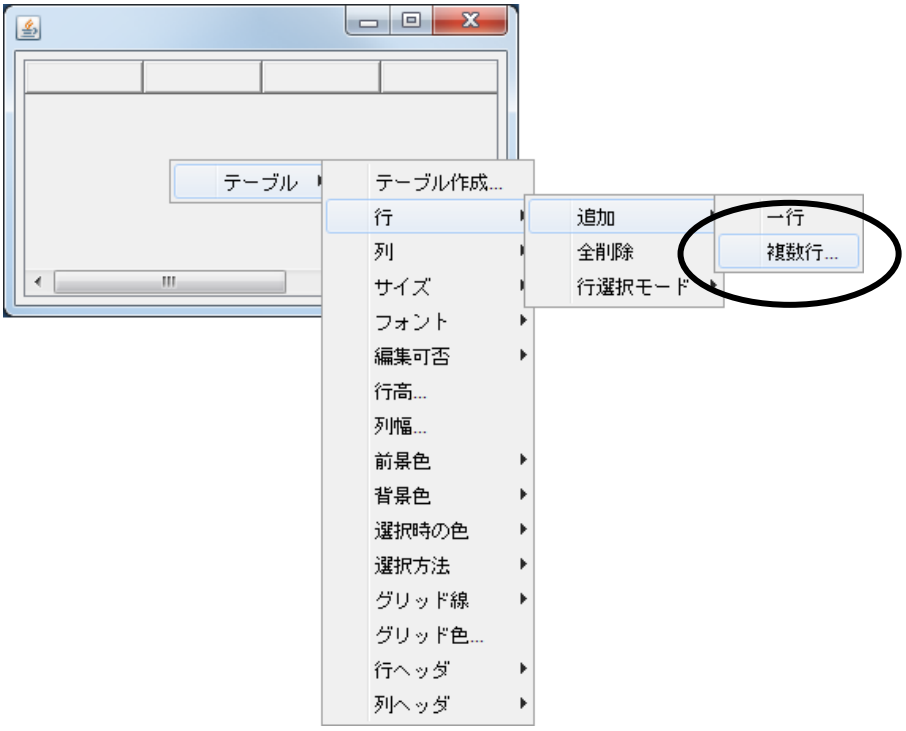


⑥　開始・終了日時を入力する列を日付型で作成します。
実行（設定可）で実行したテーブル上で右クリックします。
［テーブル］－［列］－［追加］－［複数列］－［日付...］をクリックします。

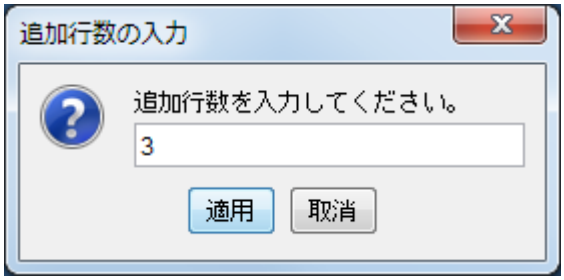
⑦　列数を指定します。
ここでは「6」列にします。



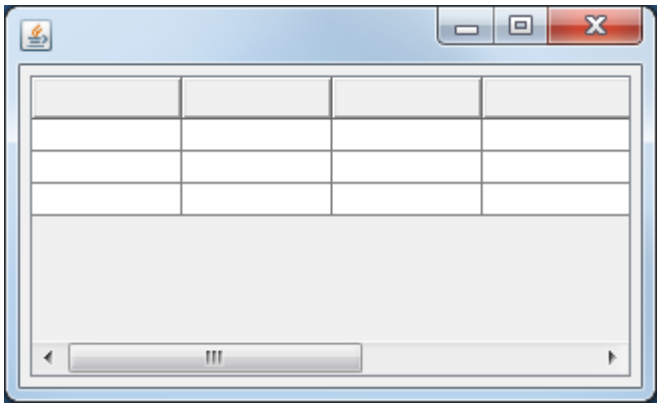
⑧　テーブルの行数を指定します。
実行（設定可）で実行したテーブルの上で右クリックします。
［テーブル］－［行］－［追加］－［複数行...］をクリックします。



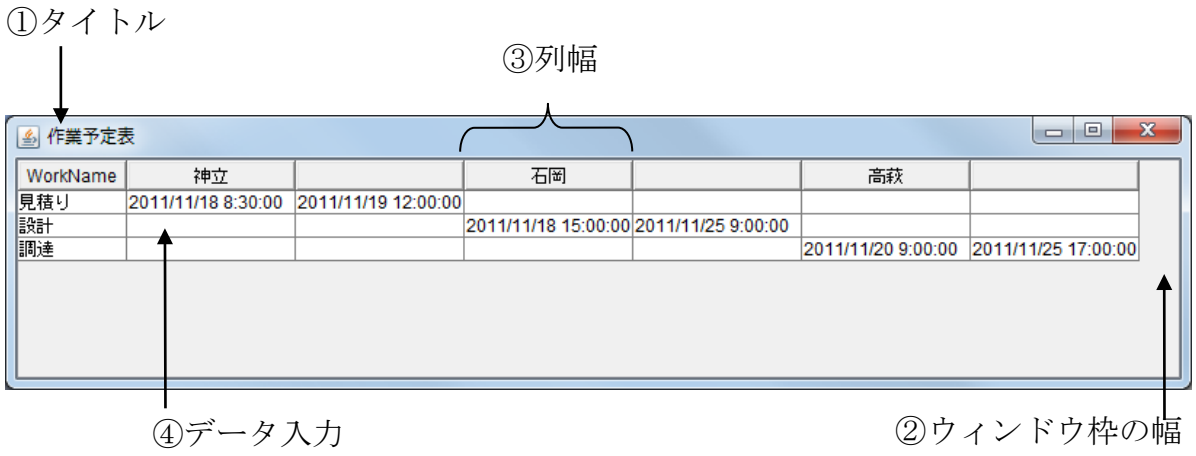
⑨　行数を指定します。
　　ここでは「3」行にします。



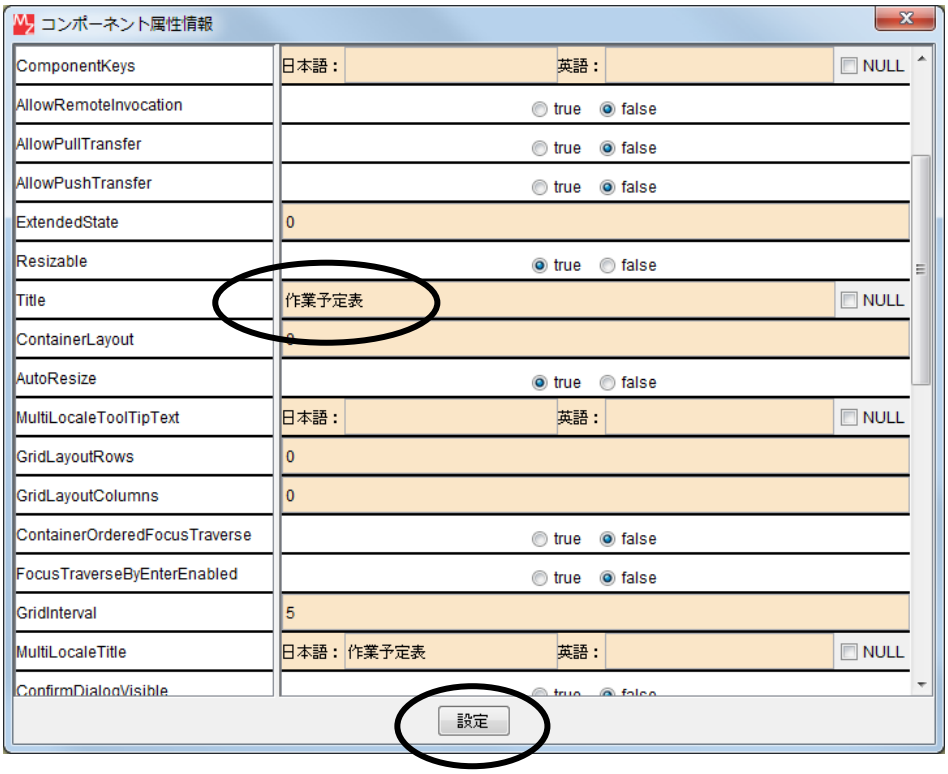
テーブルが完成します。



⑩　テーブルの体裁を整えデータを入力します。



ウィンドウの名前（タイトル）を入力します。
ビルダー上のフレームコンポーネント上で右クリック－［属性情報設定...］をクリックします。
［Title］に「作業予定表」と入力します。
設定をクリックします。

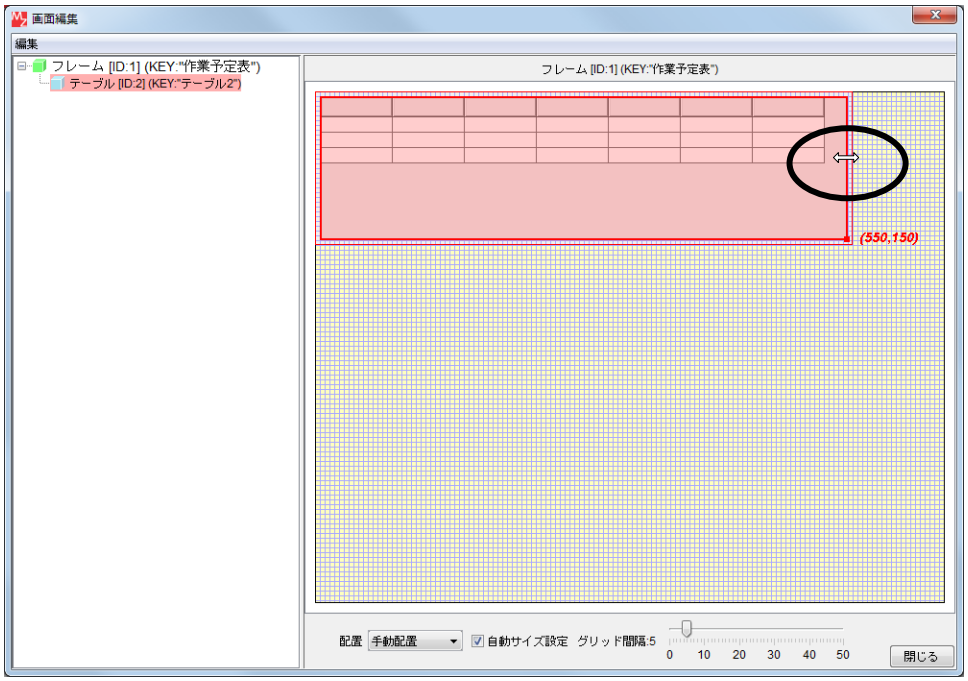


⑪　テーブル全体の幅を広げます。

画面編集をクリックします。

テーブルの周りの赤い線をドラッグし幅を広げます。

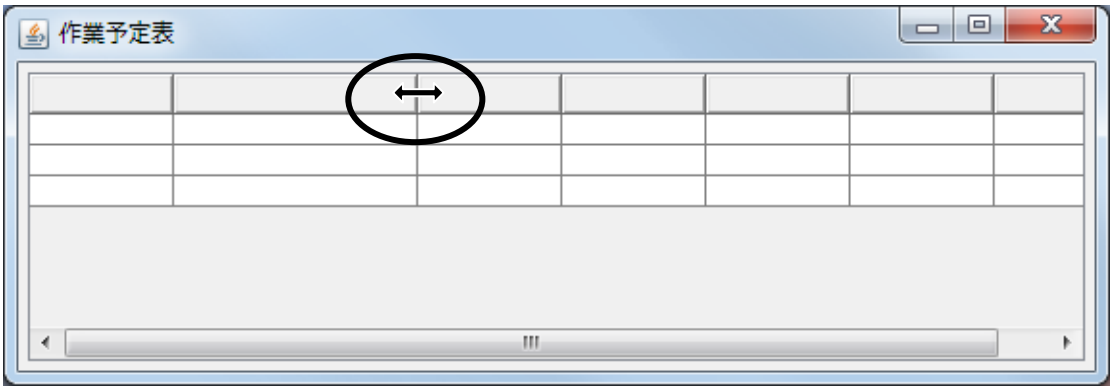
閉じるをクリックします。



⑫　列幅を調整します。

実行（設定可）で実行します。

列名の列と列の間にマウスポインタを合わせてドラッグして列幅を調整します。

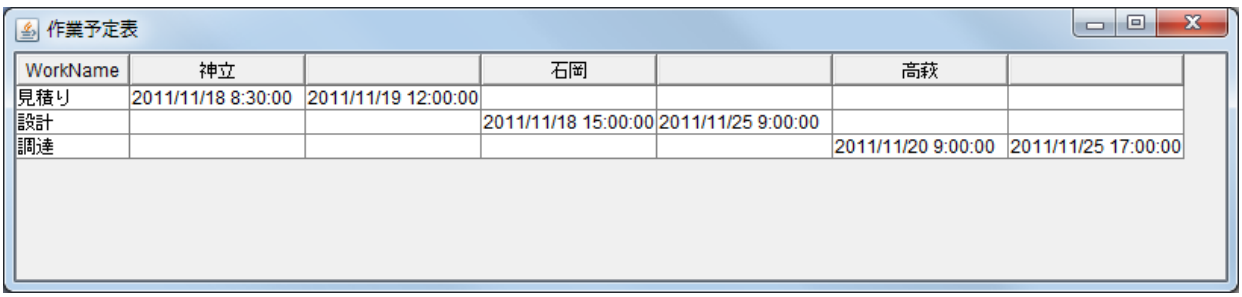


⑬　データを入力します。

セルをダブルクリックしてデータを入力し、以下のように完成してください。

入力する日時データは自由です。例えば開始日は本日の日付とし、終了日は数日先の日付とします。

時刻データは必ず「秒」まで入力してください。

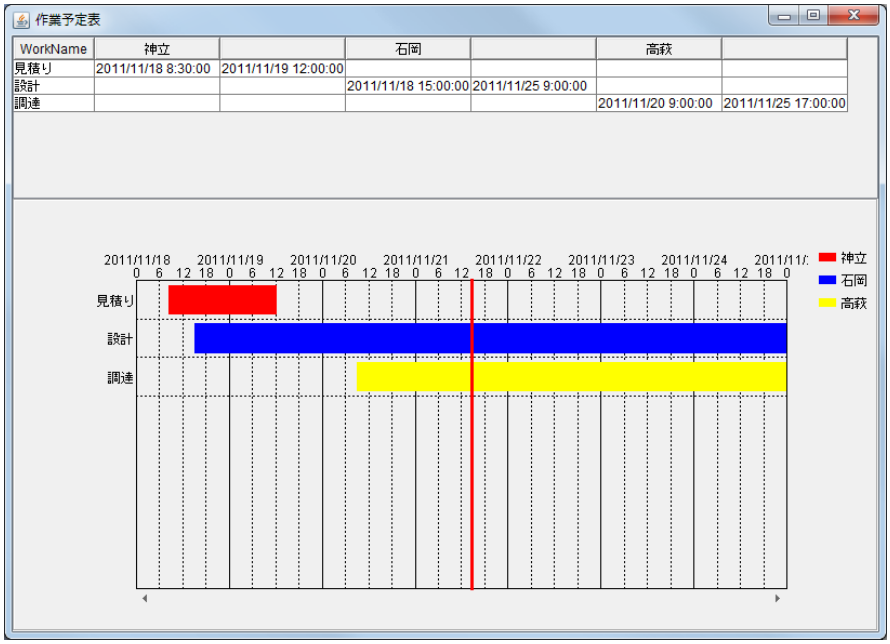


WorkName	神立		石岡		高萩	
見積り	2011/11/18 8:30:00	2011/11/19 12:00:00				
設計			2011/11/18 15:00:00	2011/11/25 9:00:00		
調達					2011/11/20 9:00:00	2011/11/25 17:00:00

2）ガントチャートを表示する

テーブルのデータをガントチャートに表示しましょう。

完成図 テーブルデータをガントチャートに表示します。



準 備 ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■ガントチャート	1	［画面構成部品］－［グラフ］－［ガントチャート］

操 作 ガントチャートを追加しましょう。

① 必要なコンポーネントを追加します。
ここでは［ガントチャート］コンポーネントを追加します。
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［グラフ］－［ガントチャート］とクリックします。

接続確認

コンポーネント同士の接続を確認します。

ガントチャートのデータを設定する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■アプリケーション
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■ガントチャート（ID:3）
起動メソッド	ガントチャートのデータを設定する(PFObjectTable)
＜引数＞	説明：ガントチャートのデータ 取得方法：メソッド戻り値 コンポーネント：テーブル メソッド／値：テーブルデータを取得する

操 作

計算処理を設定しましょう。

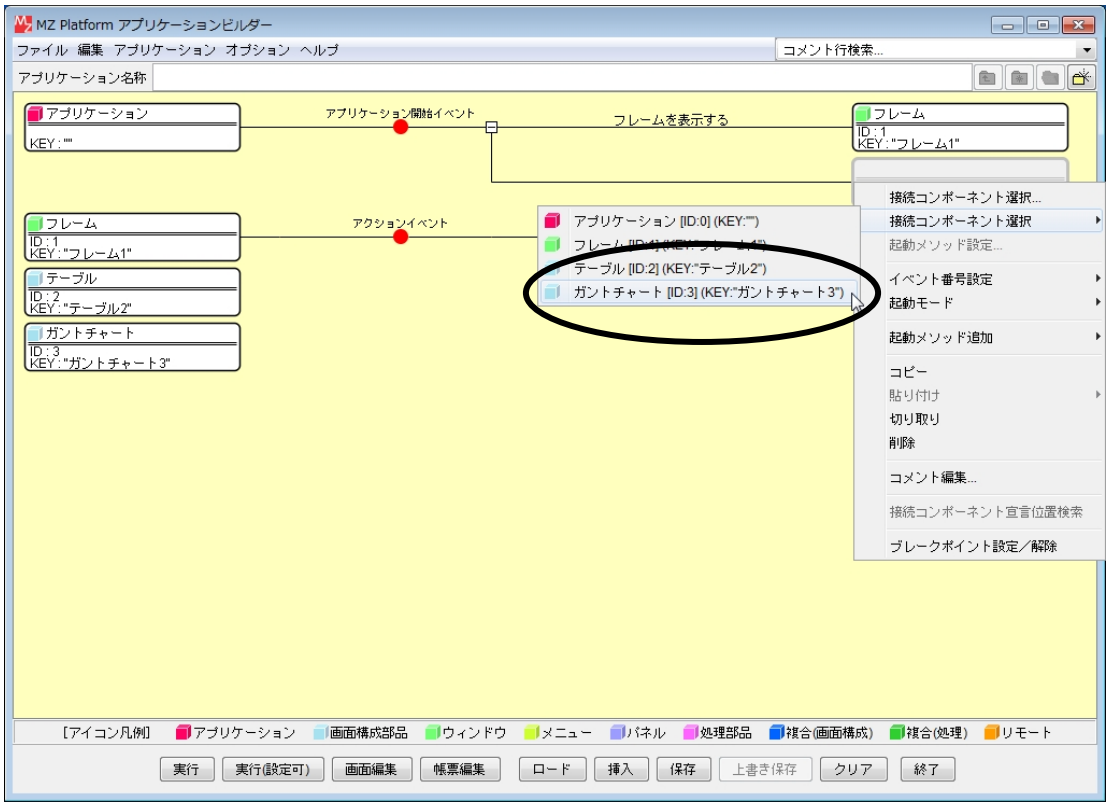
① イベントの接続先コンポーネントを選びます。

［アプリケーション］コンポーネントの［アプリケーション開始イベント］に２つめの起動メソッドを追加します。

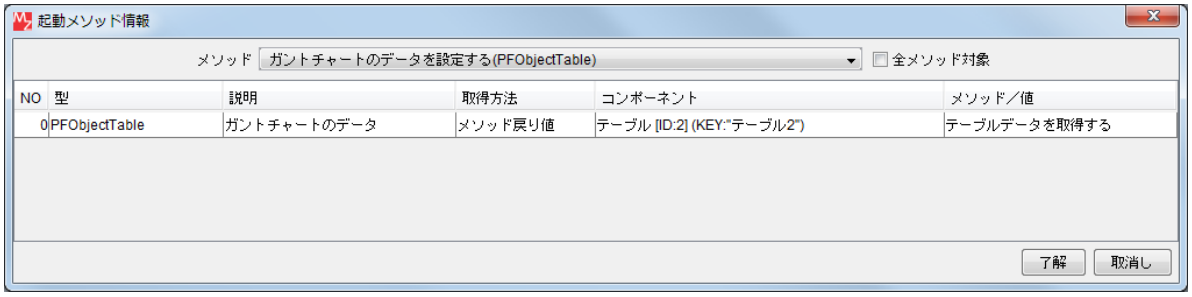
左側の［アプリケーション］コンポーネントの［アプリケーション開始イベント］上で右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。

右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。

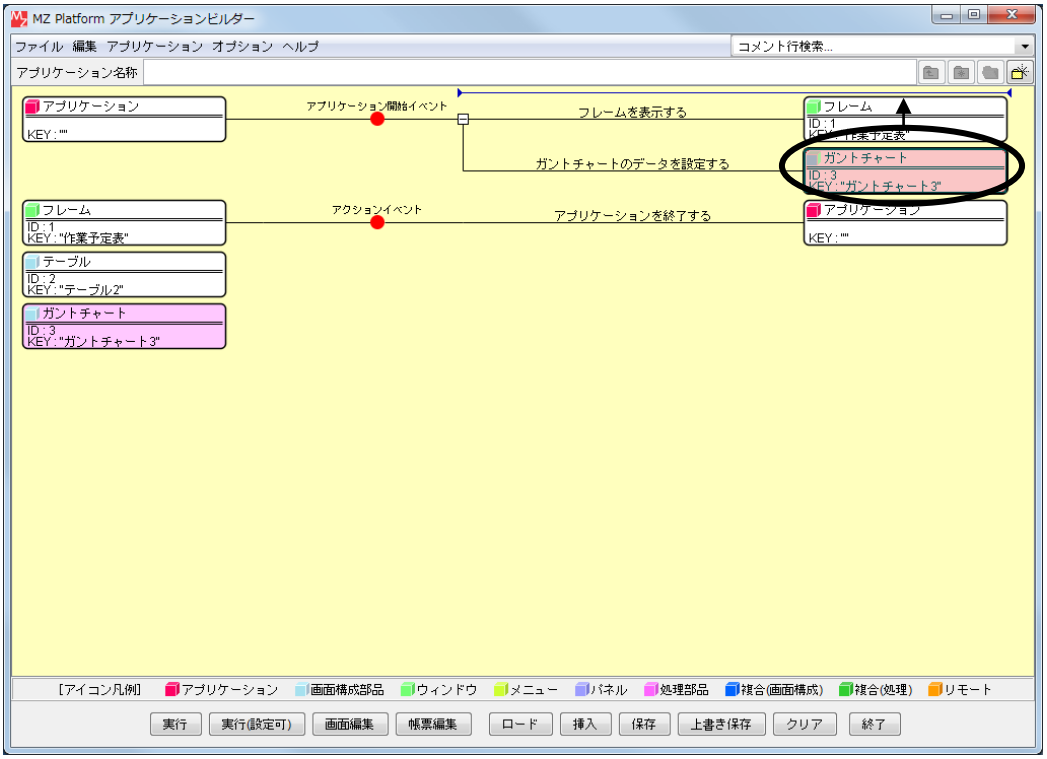
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－［ガントチャート(ID:3)］をクリックします。



② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［ガントチャートのデータを設定する(PFObjectTable)］をクリックします。
引数を設定します。
説明：ガントチャートのデータ
取得方法：メソッド戻り値
コンポーネント：テーブル
メソッド／値：テーブルデータを取得する
設定後、了解ボタンをクリックします。



③ ［アプリケーション］コンポーネントと［ガントチャート］コンポーネントが接続されます。
フレームはデータを設定してから開くように変更します。
［ガントチャート］コンポーネントと［フレーム］コンポーネントを入れ替えます。
［ガントチャート］コンポーネントをドラッグして［フレーム］コンポーネントの上に移動します。



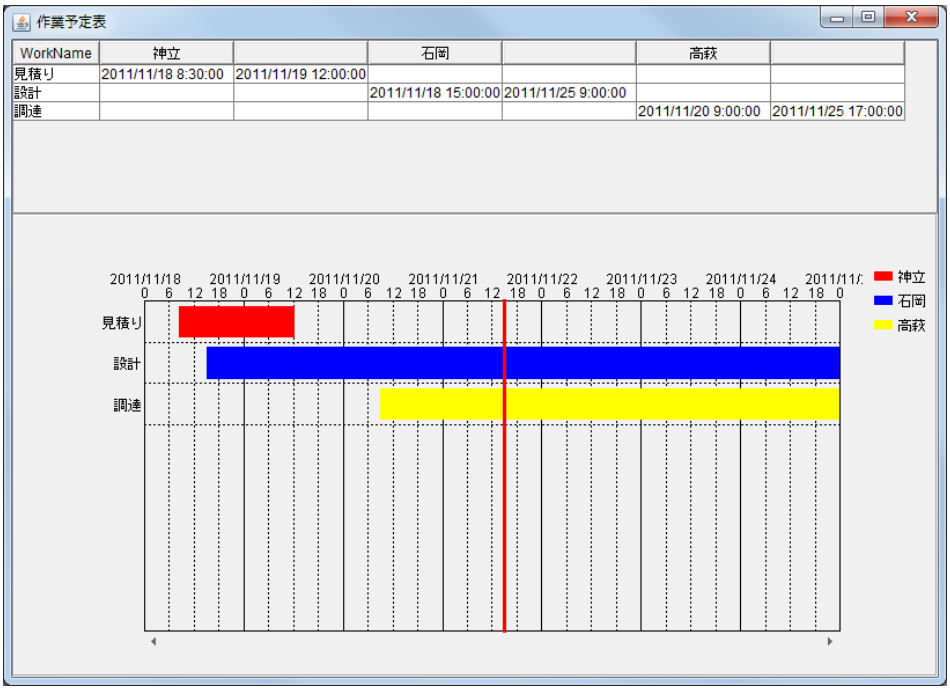
④　画面を作成します。
画面編集をクリックします。

⑤　［ガントチャート］コンポーネントをフレームに追加します。
画面上で右クリックー［コンポーネント追加］ー［ガントチャート］コンポーネントとクリックします。
横方向に配置されるので［手動配置］にして上下にバランスよく配置しましょう。
追加できたら閉じるをクリックします。

⑥　ガントチャートの画面が追加できたことを確認します。
実行（設定可）で実行します。

⑦以下を参考にガントチャート表示の調整をします。
グラフエリアで右クリックし、

1）グラフ表示の項目数　　:［ガントチャート］ー［作業項目軸］ー［表示項目数...］
2）グラフ表示の開始日時　　:［ガントチャート］ー［時間軸］ー［表示開始日時...］
3）グラフの表示期間　　:［ガントチャート］ー［時間軸］ー［表示期間...］
4）数値軸の目盛り間隔　　:［ガントチャート］ー［時間軸］ー［大目盛間隔...］
　　:［ガントチャート］ー［時間軸］ー［小目盛間隔...］
5）時間軸ラベルの表示形式　　:［ガントチャート］ー［時間軸］ー［大項目ラベル表示形式...］ー［設定する］
　　:［ガントチャート］ー［時間軸］ー［小項目ラベル表示形式...］ー［設定する］
　　＊yyyy/MM/dd(年/月/日)、H:mm:ss（時:分:秒）等で設定
6）余白の設定　　:［パネル］ー［余白］
7）ラベル位置の調整　　:［ガントチャート］ー［作業項目軸］ー［ラベル表示幅］
　　:［ガントチャート］ー［時間軸］ー［ラベル表示高さ］

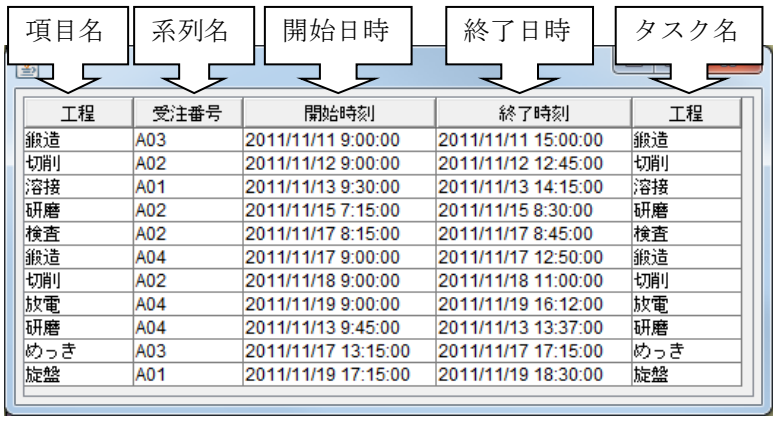


知っていると便利！

［ガントチャート］へは Step2 で紹介した基本のデータ形式だけではなく、順に「項目名」、「系列名」、「開始日時」、「終了日時」、「タスク名」となる列データを持つ、5 列ないし 4 列（「タスク名」の設定は任意）のテーブルデータを設定できます。このメソッドでデータを設定した時は、次の Step.4 で紹介する「テーブルとガントチャートの連動」もこのチュートリアルとは違った設定方法になります。

例）

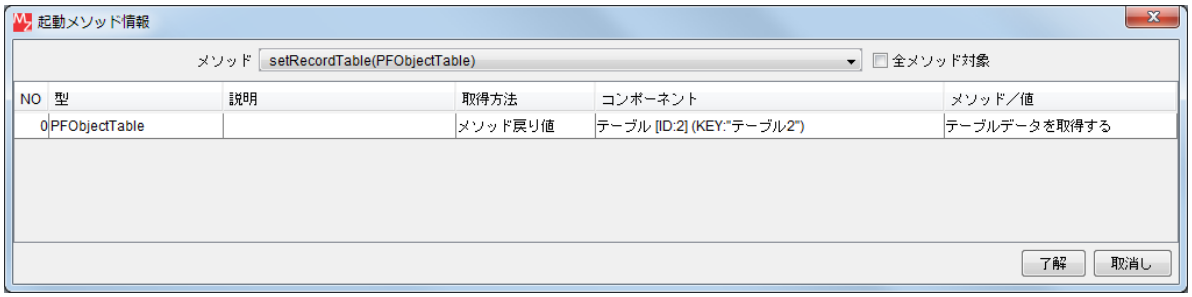
①テーブルデータを用意します。



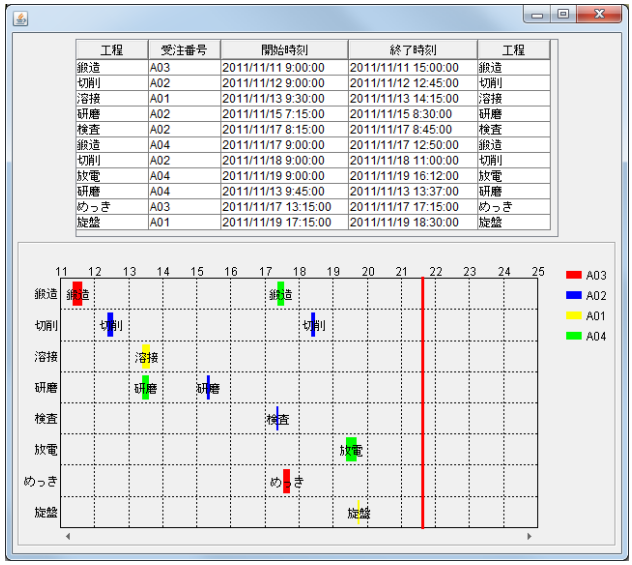
工程	受注番号	開始時刻	終了時刻	工程
鍛造	A03	2011/11/11 9:00:00	2011/11/11 15:00:00	鍛造
切削	A02	2011/11/12 9:00:00	2011/11/12 12:45:00	切削
溶接	A01	2011/11/13 9:30:00	2011/11/13 14:15:00	溶接
研磨	A02	2011/11/15 7:15:00	2011/11/15 8:30:00	研磨
検査	A02	2011/11/17 8:15:00	2011/11/17 8:45:00	検査
鍛造	A04	2011/11/17 9:00:00	2011/11/17 12:50:00	鍛造
切削	A02	2011/11/18 9:00:00	2011/11/18 11:00:00	切削
放電	A04	2011/11/19 9:00:00	2011/11/19 16:12:00	放電
研磨	A04	2011/11/13 9:45:00	2011/11/13 13:37:00	研磨
めっき	A03	2011/11/17 13:15:00	2011/11/17 17:15:00	めっき
旋盤	A01	2011/11/19 17:15:00	2011/11/19 18:30:00	旋盤

②［ガントチャート］にデータを設定します。

起動メソッド設定の際に「全メソッド対象」にチェックを入れ、［setRecordTable（PFObjectTale)］を選びます。



③［ガントチャート］が表示されます。



Step.4 テーブルとガントチャートを連動させる

テーブルのデータを変更したらガントチャートに反映されるように、またはガントチャートのデータを変更したらテーブルに反映されるように設定を変更しましょう。

１）テーブルのイベント番号

コンポーネントとコンポーネントは「イベント」で接続しています。

ここでは［テーブル］コンポーネントと［ガントチャート］コンポーネントを接続します。
一方のデータを変更したときにもう一方のデータも変更したいので［データ更新イベント］を利用します。

テーブルデータの更新の種類は複数のケースがあります。キーボードから値を修正した場合や、行や列の追加／削除のようにデータ構造を変更する場合も［データ更新イベント］が発生します。このように同じイベント（ここではデータ更新イベント）でも複数の意味を持つ場合があります。

こうした複数のイベントの内容を識別するために『イベント番号』を使用します。
『イベント番号』とは同じイベントが複数の意味を持つ場合、それぞれのイベントに番号が振ってあり、どのイベントなのか識別することができるというものです。

［データ更新］イベントのイベント番号は以下のとおりです。

イベント番号	内　容
0	セルの値が更新されたケース
1	行が更新されたケース
2	列が更新されたケース
10	行が追加されたケース
11	複数行が追加されたケース
12	列が追加されたケース
13	複数列が追加されたケース
20	行が削除されたケース
21	全行が削除されたケース
22	列が削除されたケース
23	全列が削除されたケース
24	全行列が削除されたケース

ここでは、上記すべての場合にデータを更新したいので『イベント番号』を設定するのではなく、
『定常起動』（イベント発生時は常に起動する）を利用します。

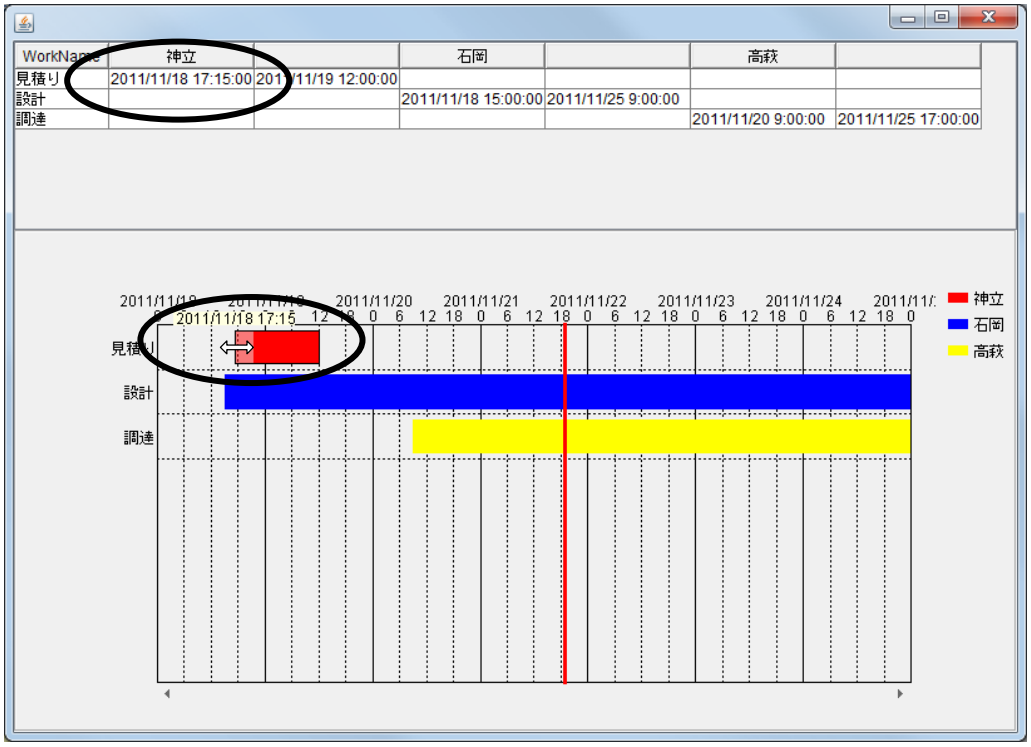
２）ガントチャートのイベント番号

ガントチャートのデータ更新は「マウスでドラッグ」した場合のみイベントが発生しますので､複数のケースはありません。イベントをイベント番号で区別する必要がない場合、接続されているメソッドをイベント発生時に常に起動するように指定します。

３）テーブルデータとガントチャートを連動

［テーブル］コンポーネントと［ガントチャート］コンポーネントを連動させましょう。

完成図 テーブルとガントチャートを連動させます。



接続確認 コンポーネント同士の接続を確認します。

テーブルのデータをガントチャートへ設定する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■テーブル（ID:2）
発生イベント	データ更新イベント
イベント番号	定常起動
接続先コンポーネント	■ガントチャート（ID:3）
起動メソッド	ガントチャートのデータを設定する(PFObjectTable)
＜引数＞	説明：ガントチャートのデータ 取得方法：イベント内包 メソッド／値：イベント対象データ

ガントチャートのデータをテーブルに設定する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ガントチャート（ID:3）
発生イベント	データ更新イベント
イベント番号	定常起動
接続先コンポーネント	■テーブル（ID:2）
起動メソッド	テーブルデータを設定する(PFObjectTable)
＜引数＞	説明：テーブルデータ 取得方法：イベント内包 メソッド／値：イベント対象データ

操 作 テーブルコンポーネントとガントチャートコンポーネントを接続しましょう

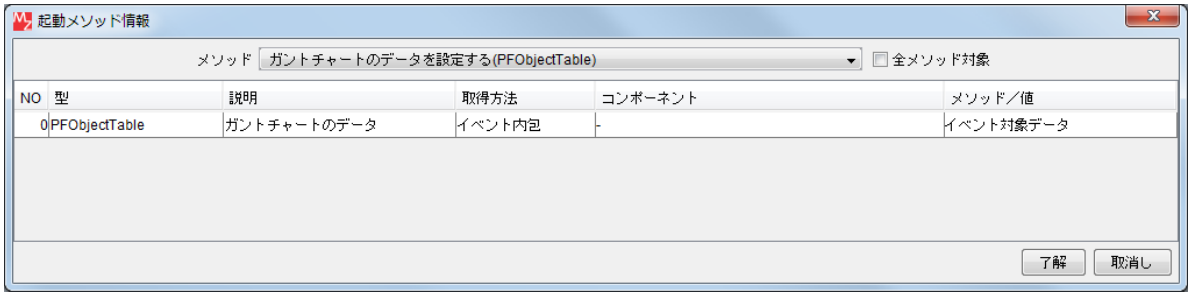
はじめに、テーブルのデータをガントチャートへ設定するよう接続します。

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［テーブル(ID:2)］コンポーネント上で右クリックー［イベント処理追加］ー［データ更新イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［テーブル］コンポーネントの［データ更新イベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー［ガントチャート(ID:3)］をクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［ガントチャートのデータを設定する(PFObjectTable)］をクリックします。
引数を設定します。
説明：ガントチャートのデータ
取得方法：イベント内包
メソッド／値：イベント対象データ

設定後、了解ボタンをクリックします。



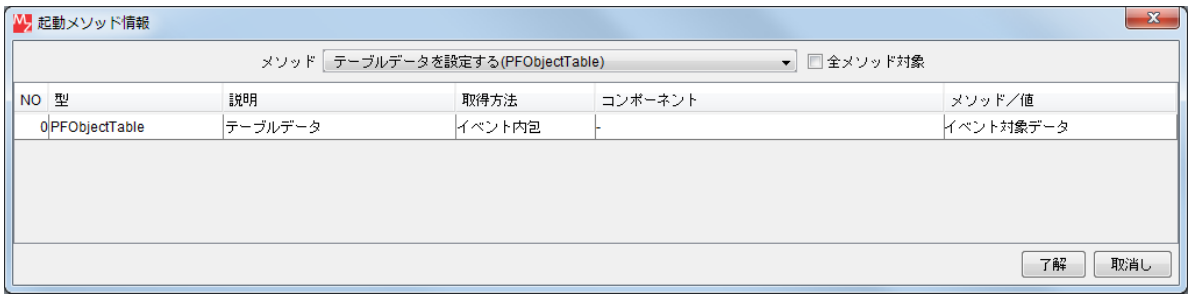
次にガントチャートのデータをテーブルに設定するように接続します。

④　使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ガントチャート(ID:3)］コンポーネント上で右クリックー［イベント処理追加］ー［データ更新イベント］とクリックします。

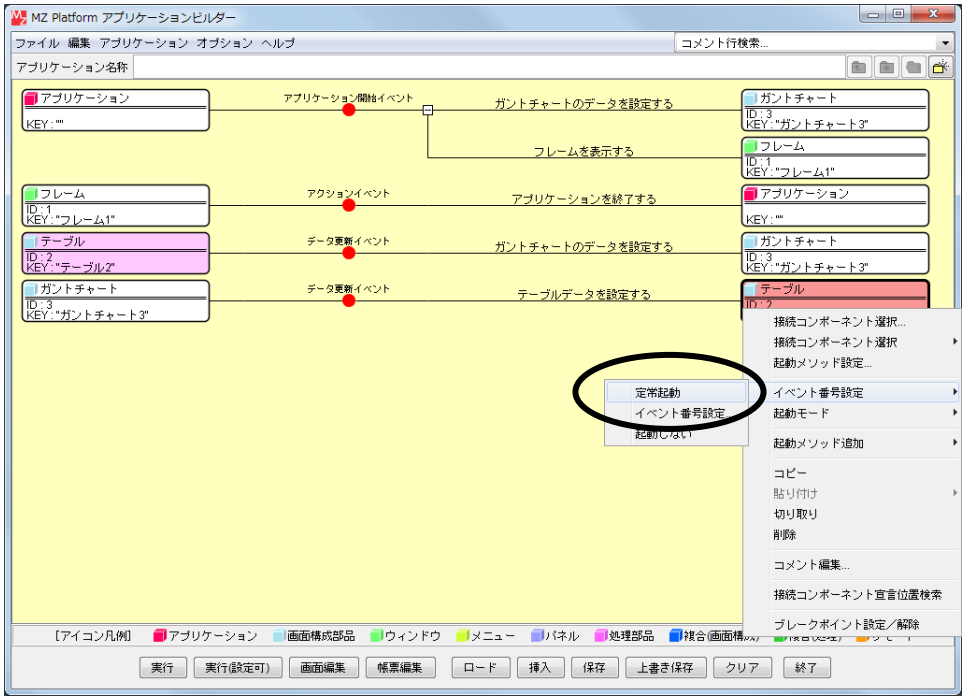
⑤　イベントの接続先コンポーネントを選びます。
左側の［ガントチャート(ID:3)］コンポーネントの［データ更新イベント］上で
右クリックー［起動メソッド追加］とクリックします。
薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー［テーブル(ID:2)］をクリックします。

⑥　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［テーブルデータを設定する(PFObjectTable)］をクリックします。
引数を設定します。
説明：テーブルデータ
取得方法：イベント内包
メソッド／値：イベント対象データ

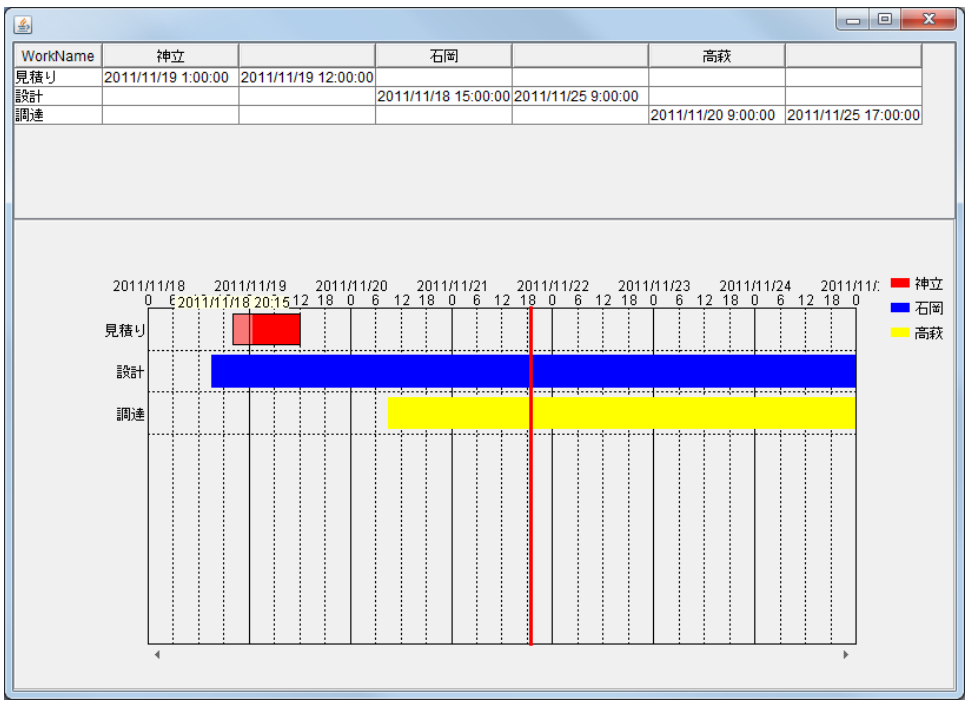
設定後、了解ボタンをクリックします。



⑦　イベント番号を確認します。
今回は常に起動するので『定常起動』になっていることを確認します。
接続先コンポーネント（ここではテーブルコンポーネント）上で右クリックー［イベント番号設定］ー［定常起動］をクリックします。

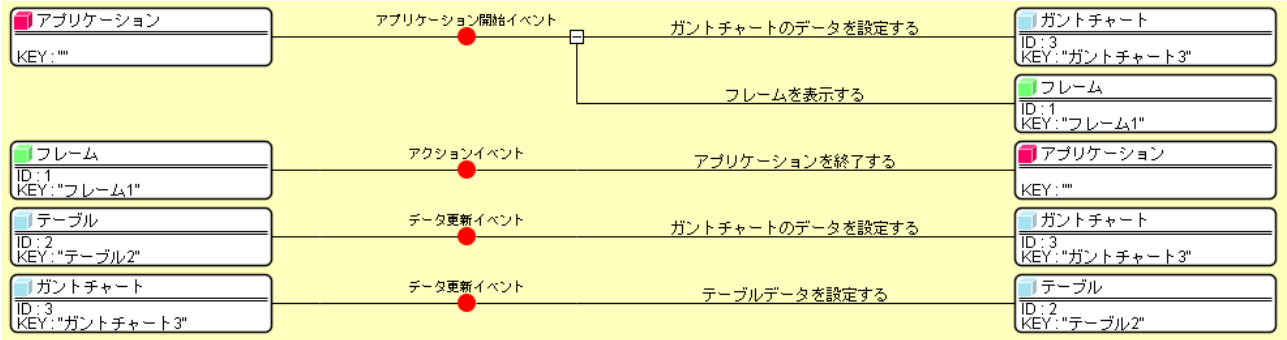


⑧　テーブルとガントチャートが連動します。
テーブルのデータやガントチャートのデータを変更しましょう。



まとめ

ここまで進めるとビルダー上では以下のようになります。

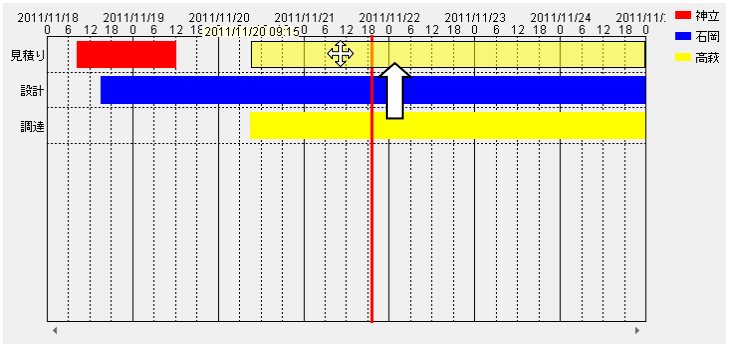


知っていると便利！

ガントチャートには「縦ドラッグ」操作や「コネクタ」、「アイコン」なども設定することができます。

① 縦ドラッグ操作

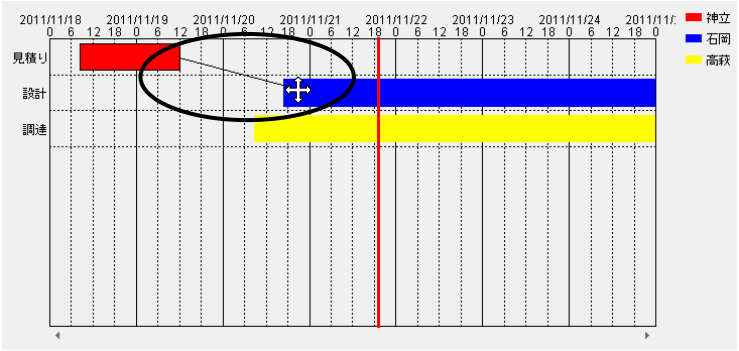
実行（設定可）で実行し、グラフエリアで右クリックー［ガントチャート］ー［縦ドラッグ］ー［有効］とします。マウスでタスクをドラッグします。



② コネクタ設定

コネクタを開始したいタスクを右クリックして、ドラッグしたまま接続したいタスク上にカーソルを移動し、離します。

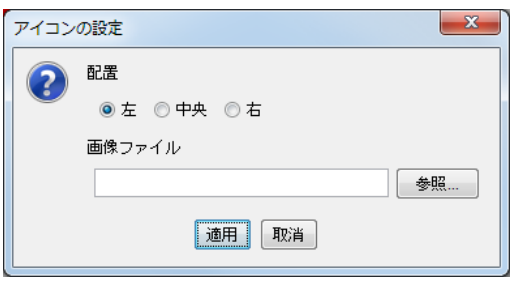
コネクタを削除するときは、マウスで選択した後 Delete キーを押してください。

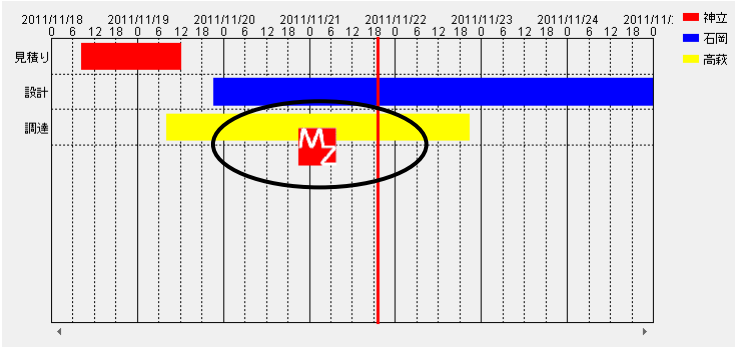


③ アイコン設定

実行（設定可）で実行し、グラフエリアで右クリックー［ガントチャート］ー[アイコン]ー［表示する］とします。

タスク上で右クリックー［ガントチャート］ー[アイコン]ー［設定する...］を選びます。アイコン設定の画面が出るので、表示したい画像のファイルを参照ボタンを押し、選択します。





Lesson.12 いろいろな画像ファイルを表示する

ここでは「ファイル入出力」コンポーネントを利用して、ファイルを呼び出す方法をご紹介します。
これまでフレームを１つしか使用していませんでしたが、ここでは２つのフレームを使用します。
また「複合コンポーネント」という MZ Platform の考え方を使った効率の良いアプリケーション構築方法をご紹介します。

Step.1 利用できる画像ファイルの種類

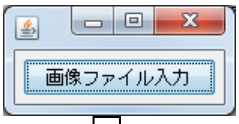
MZ Platform では画像ファイルを扱うことが可能です。取り扱い可能な主なファイル形式は以下のとおりです。

・GIF
・JPEG
・PNG
・BMP

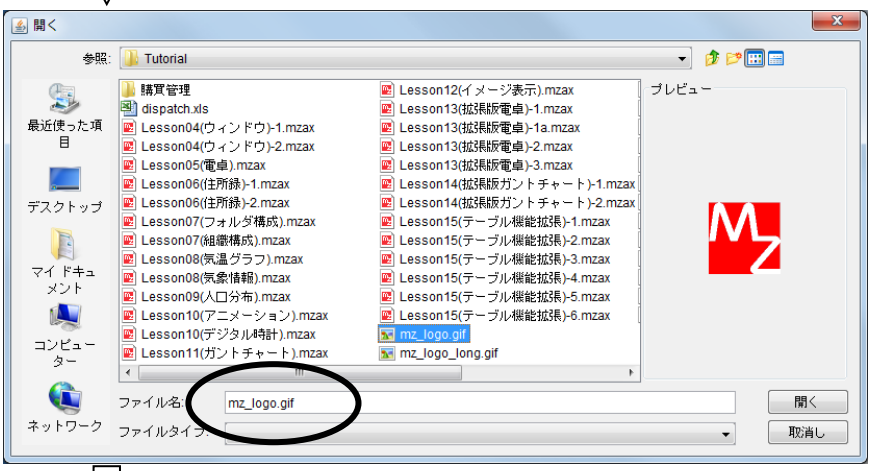
Step.2 画像ファイルの入力

画像ファイルを取り込んで表示しましょう。

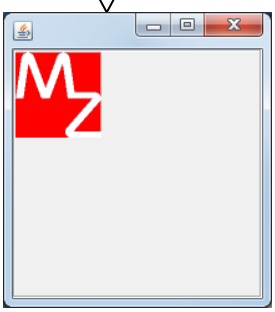
完成図 画像ファイルを取り込みます。



① ボタンをクリックして［画像入出力］の画面を呼び出します。



② ファイルを選択します。



③ 画像ファイルが表示されます。

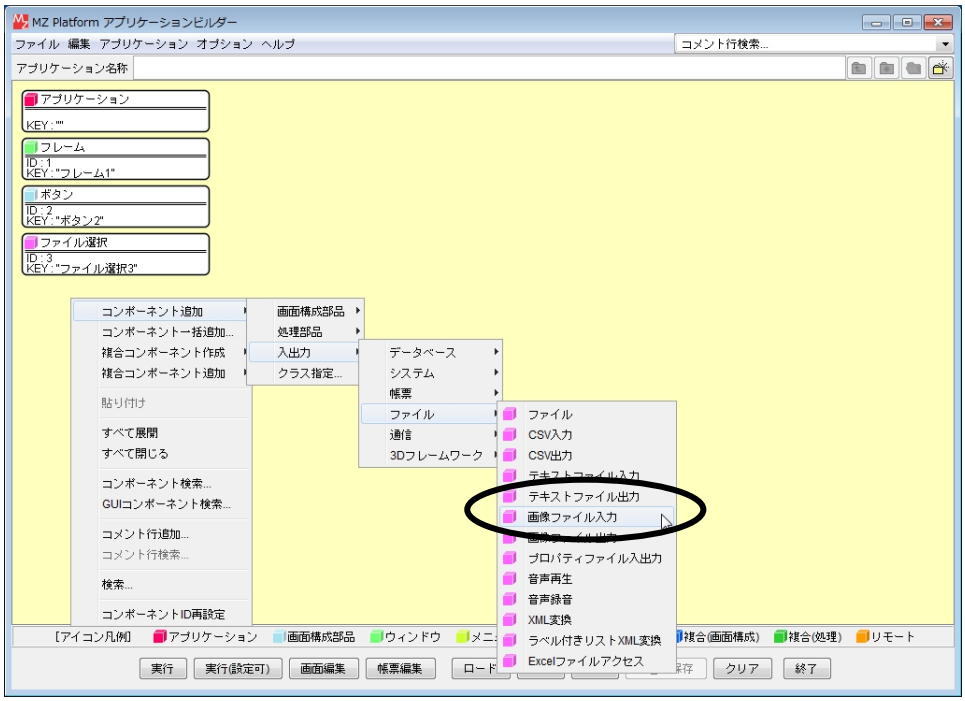
準 備

ここでは以下のコンポーネントを使用します。

コンポーネント名	必要数	
■アプリケーション	(1)	
■フレーム	1	［画面構成部品］－［ウィンドウ］－［フレーム］
■ボタン	1	［画面構成部品］－［ボタン］－［ボタン］
■ファイル選択	1	［画面構成部品］－［ダイアログ］－［ファイル選択］
■画像ファイル入力	1	［入出力］－［ファイル］－［画像ファイル入力］

操 作

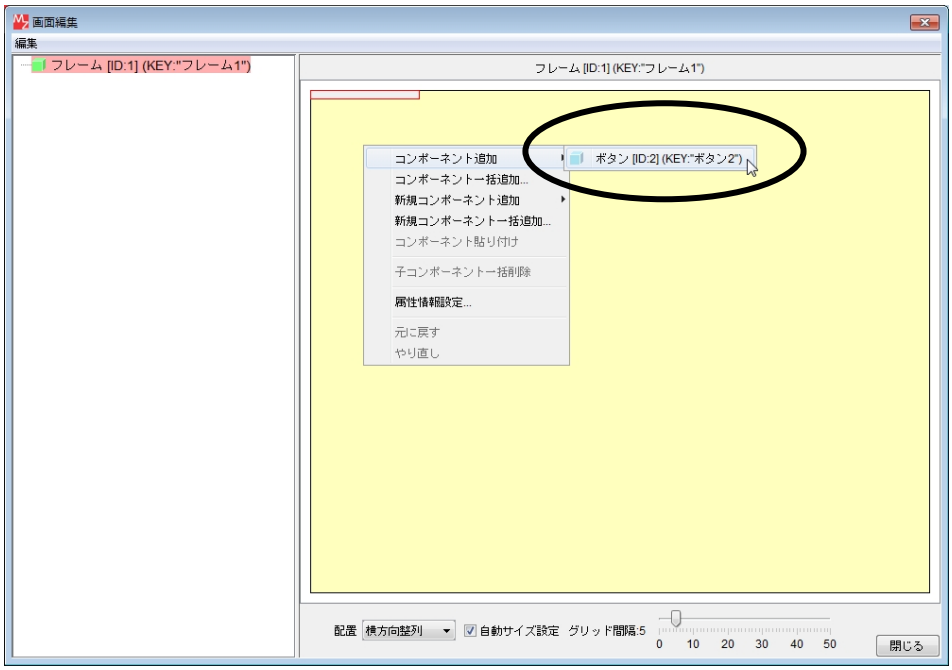
① 必要なコンポーネントを追加します。
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［ウィンドウ］－［フレーム］、
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［ボタン］－［ボタン］、
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［ダイアログ］
－［ファイル選択］、
作業領域で右クリック－［コンポーネント追加］－［入出力］－［ファイル］－［画像ファイル入力］
とクリックします。



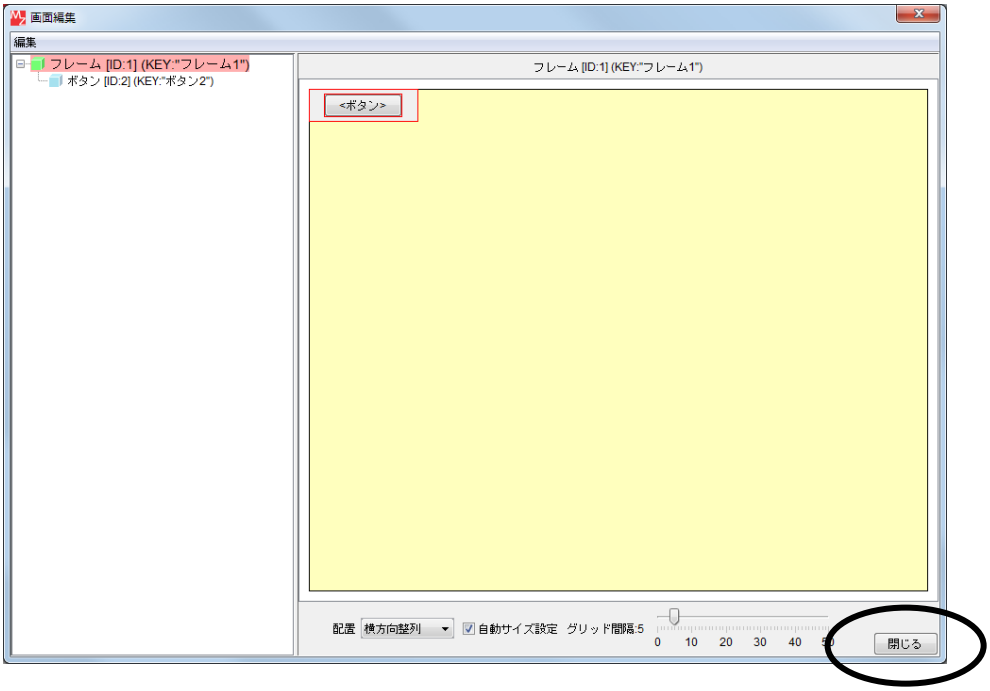
画面編集

画面を作成します。

① 画面編集をクリックします。
［ボタン］コンポーネントをフレームに追加します。
[画面編集]画面上で右クリックー［コンポーネント追加］ー［ボタン］コンポーネントとクリックします。



④ 追加できたら閉じるをクリックし、ビルダー画面に戻ります。



接続確認

コンポーネント同士の接続を確認します。

開始

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■アプリケーション
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■フレーム（ID:1）
起動メソッド	フレームを表示する()

終了

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■フレーム（ID:1）
発生イベント	アクションイベント
接続先コンポーネント	■アプリケーション
起動メソッド	アプリケーションを終了する()

ボタンをクリックしたらファイル選択画面が表示される

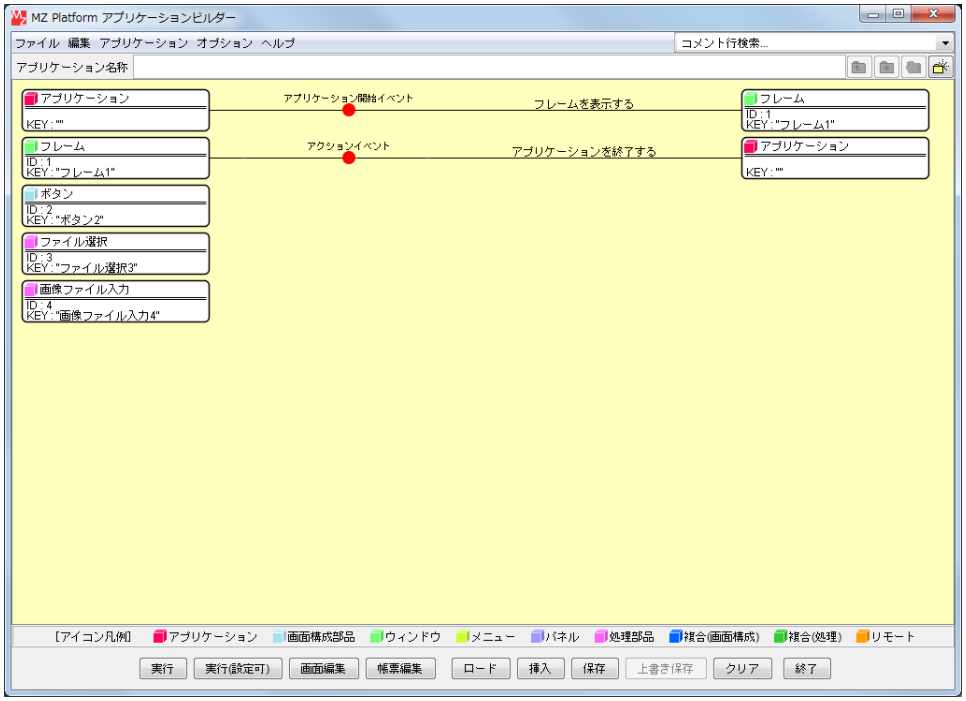
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（ID:2）
発生イベント	アクションイベント
接続先コンポーネント	■ファイル選択（ID:3）
起動メソッド	単数 Open 用ファイル選択ダイアログを表示する(Component)
＜引数＞	説明：親コンポーネント 取得方法：コンポーネント コンポーネント：フレーム(ID:1)

ファイル選択画面からファイル名を指定して画像を読み込む

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ファイル選択（ID:3）
発生イベント	データ選択イベント
接続先コンポーネント	■画像ファイル入力（ID:4）
起動メソッド	ファイル名を指定して画像を読み込む(String)
＜引数＞	説明：読み込むファイル名 取得方法：イベント内包 コンポーネント：選択データ
イベント番号	1

操 作

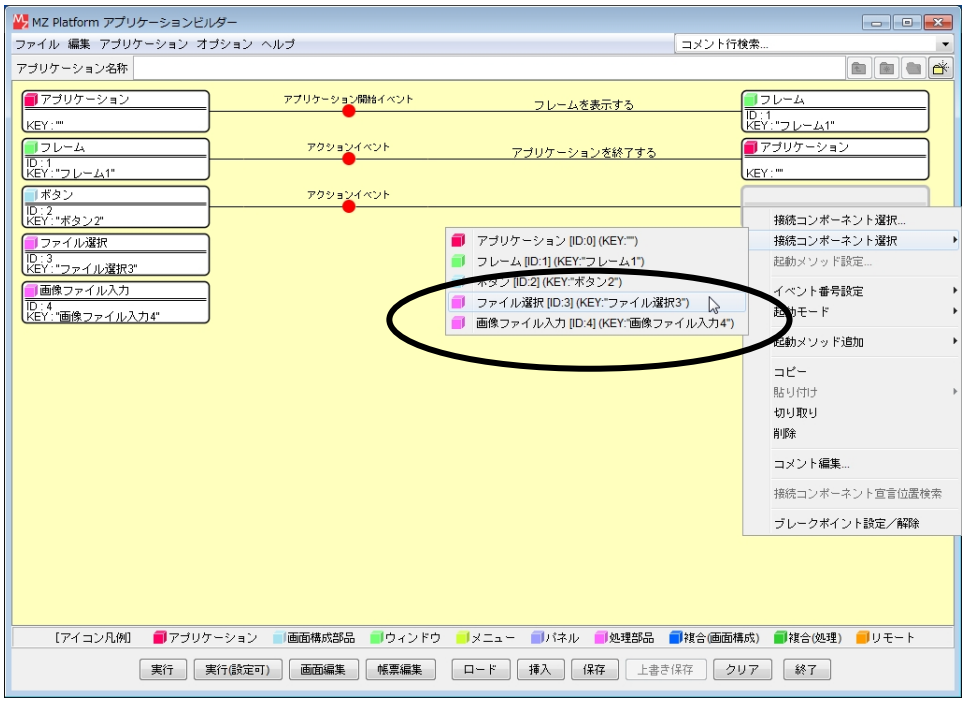
① ［フレーム］コンポーネントと［アプリケーション］コンポーネントを接続します。



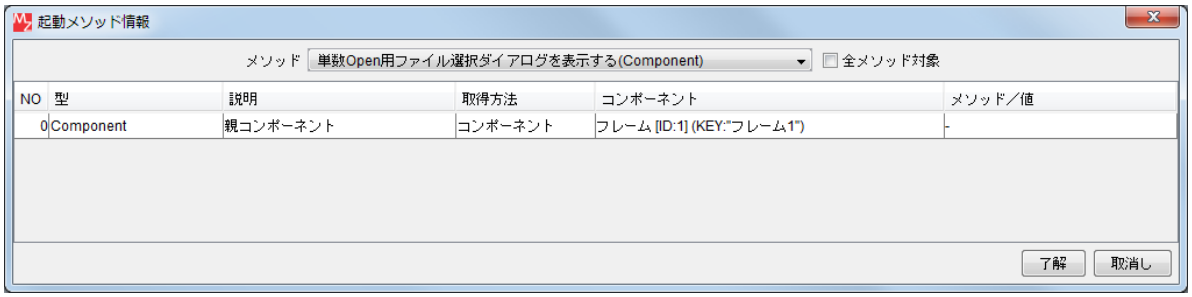
ボタンをクリックしたらファイル選択画面が表示されるように接続します。

② 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ボタン(ID:2)］コンポーネント上で右クリック－［イベント処理追加］－［アクションイベント］とクリックします。

③ イベントの接続先コンポーネントを選びます。
左側の［ボタン(ID:2)］コンポーネントの［アクションイベント］上で右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－［ファイル選択（ID:3)］コンポーネントをクリックします。



④　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［単数 Open 用ファイル選択ダイアログを表示する(Component)］をクリックします。
引数を設定します。
説明：親コンポーネント
取得方法：コンポーネント
コンポーネント：フレーム(ID:1)
設定後、了解ボタンをクリックします。



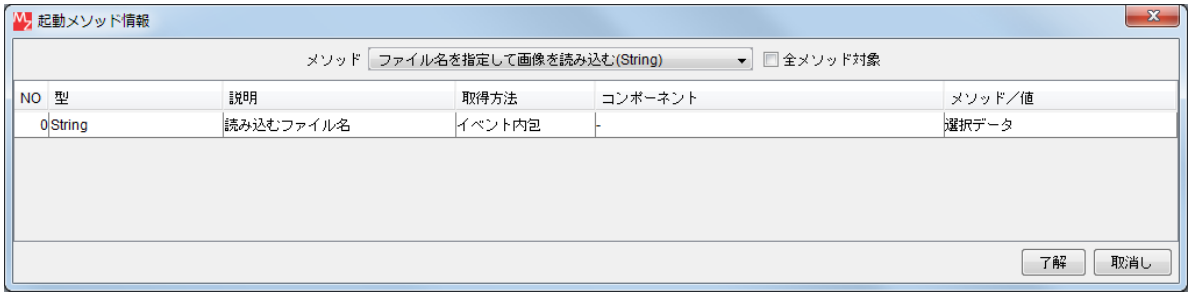
ファイルが選択されたら画像ファイルが読み込まれるように接続します。

⑤　使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ファイル選択（ID:3)］コンポーネント上で右クリックー［イベント処理追加］
ー［データ選択イベント］とクリックします。

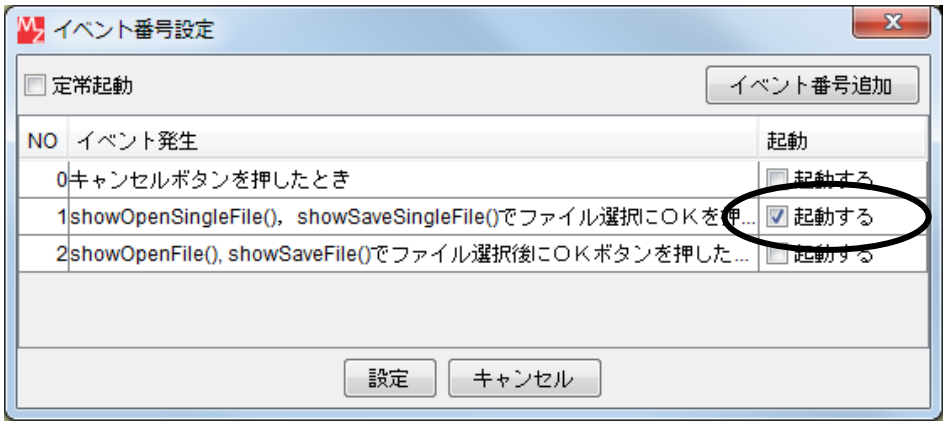
⑥　イベントの接続先コンポーネントを選びます。
左側の［ファイル選択（ID:3)］コンポーネントの［データ選択イベント］上で

右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［画像ファイル入力(ID:4)］コンポーネントをクリックします。

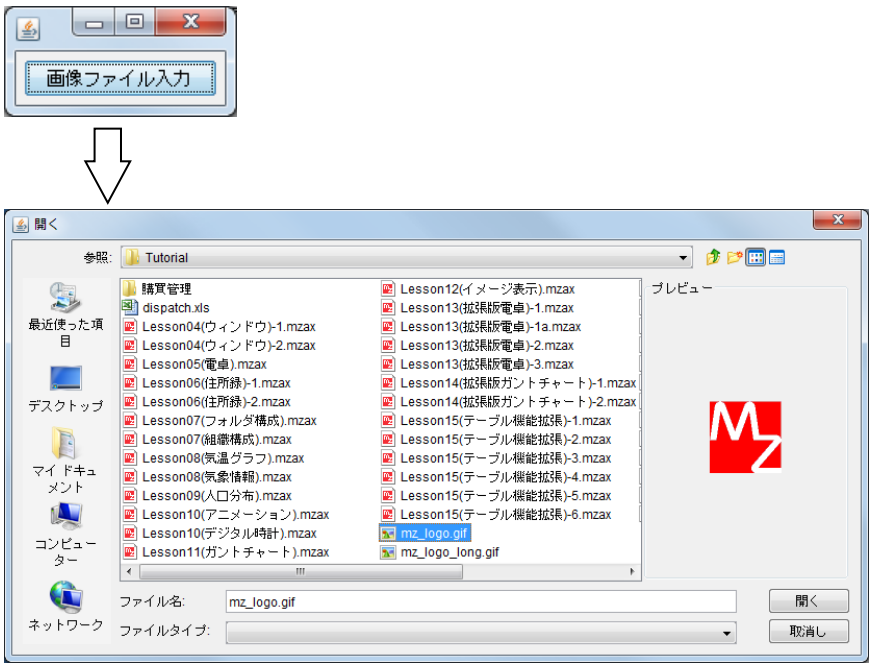
⑦ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［ファイル名を指定して画像を読み込む(String)］をクリックします。
引数を設定します。
説明：読み込むファイル名
取得方法：イベント内包
メソッド／値：選択データ
設定後、了解ボタンをクリックします。



⑧ イベント番号を設定します。
［画像ファイル入力(ID:4)］コンポーネントの上で右クリックー［イベント番号設定］
ー［イベント番号設定］をクリックします。
定常起動のチェックをオフにして［NO:1］をチェックします。



⑨　ボタンが表示され、ボタンをクリックするとファイル選択の画面が表示することを確認します。
実行（設定可）で実行します。
ボタン名を［画像ファイル入力］に変更しましょう。



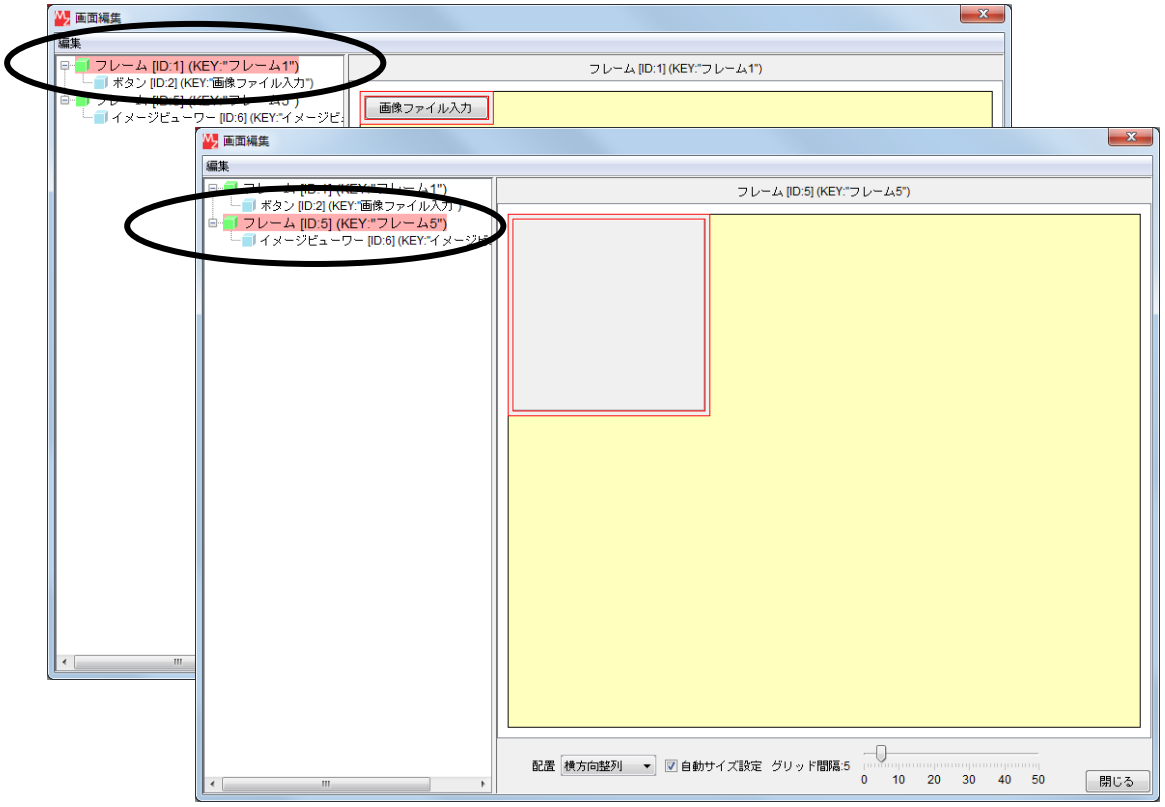
※画像ファイルはまだ表示されません。

Step.3 新しいフレームの利用

ファイル選択画面で選んだファイルを新しいフレームに表示します。

これまでの Lesson では１つのフレームを利用していました。ここでは２つ目のフレームを利用します。複数のフレームを使用するときには、[画面編集] 画面の左側の領域を利用します。この領域には、そのアプリケーションに準備してあるフレームが表示されます。フレームのＩＤを確認しながら使います。

ここまでで［フレーム(ID:1)］コンポーネントに［ボタン］コンポーネントが追加されています。ここでは新たに［フレーム］コンポーネントを追加して［イメージビューワー］コンポーネントを追加します。



考え方

１．ボタンとは別のウィンドウ（フレーム）を追加しそのウィンドウに画像を表示する。

準 備

ここでは以下のコンポーネントを追加します。

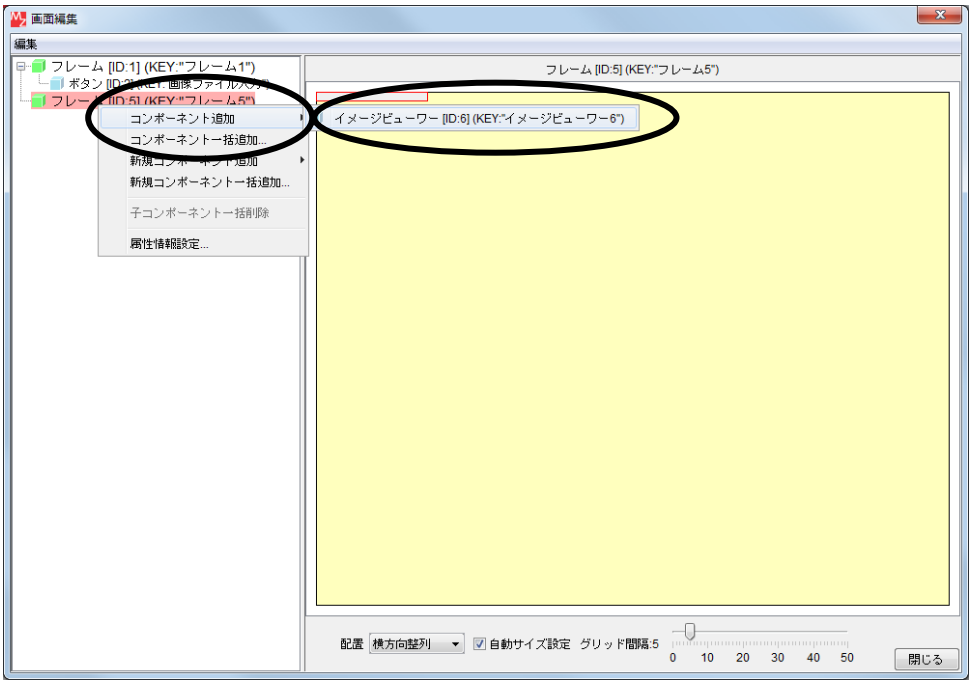
コンポーネント名	必要数	
■フレーム	1	［画面構成部品］－［ウィンドウ］－［フレーム］
■イメージビューワー	1	［画面構成部品］－［グラフィックス］－［イメージビューワー］

操 作

① 必要なコンポーネントを追加します。
作業領域で右クリックー［コンポーネント追加］ー［画面構成部品］ー［ウィンドウ］ー［フレーム］、
作業領域で右クリックー［コンポーネント追加］ー［画面構成部品］ー［グラフィックス］
ー［イメージビューワー］とクリックします。

画面編集

① 画面を作成します。
画面編集をクリックします。
左側の領域から［フレーム(ID:5)］をクリックします。
ここに［イメージビューワー(ID:6)］コンポーネントを追加します。
［画面編集］画面上で右クリックー［コンポーネント追加］ー［イメージビューワー(ID:6)］と
クリックします。



② 追加できたら閉じるをクリックし、ビルダー画面に戻ります。

接続確認

コンポーネント同士の接続を確認します。

ファイルが選択されたら新しいフレームが起動する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ファイル選択（ID:3）
発生イベント	データ選択イベント
接続先コンポーネント	■フレーム（ID:5）
起動メソッド	フレームを表示する()
イベント番号	1

ファイルが読み込まれたら新しいフレームに画像が表示される

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■画像ファイル入力（ID:4）
発生イベント	データ生成イベント
接続先コンポーネント	■イメージビューワー（ID:6）
起動メソッド	イメージデータを設定する(Image)
＜引数＞	説明：イメージ 取得方法：イベント内包 メソッド／値：イベント対象データ

［フレーム(ID:1)］が閉じる時、イメージデータをクリアする

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■フレーム（ID:1）
発生イベント	アクションイベント
接続先コンポーネント	■イメージビューワー（ID:6）
起動メソッド	イメージをクリアする()

［フレーム(ID:1)］が閉じる時［フレーム(ID:5)］も同時に閉じる

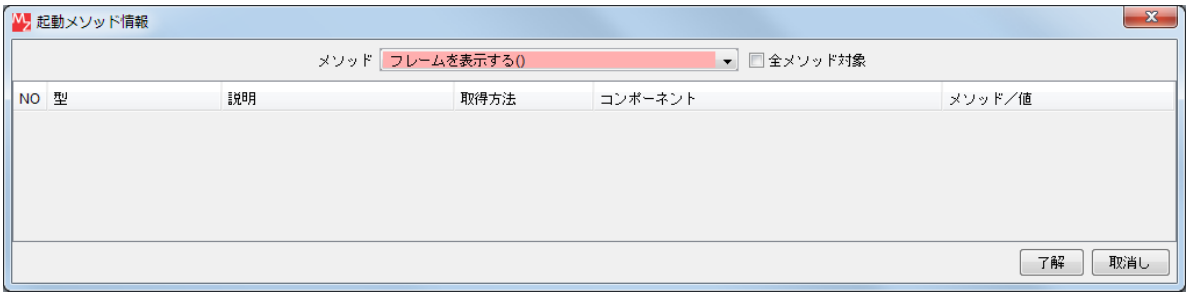
接続先コンポーネント	■フレーム（ID:5）
起動メソッド	フレームを閉じる()

操 作ファイルが読み込まれたら新しいフレームに画像が表示されるようにしましょう。

―――ファイルが選択されたら新しいフレームが起動する―――

① イベントの接続先コンポーネントを選びます。
左側の［ファイル選択(ID:3)］コンポーネントの［データ選択イベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［フレーム(ID:5)］をクリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［フレームを表示する()］をクリックします。
設定後、了解ボタンをクリックします。



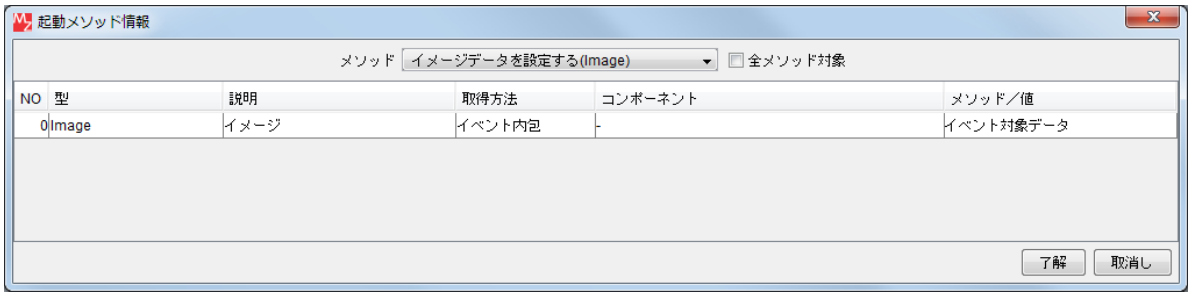
③ イベント番号を設定します。
［フレーム(ID:5)］コンポーネントの上で右クリック－［イベント番号設定］－［イベント番号設定］
をクリックします。
定常起動のチェックをオフにして［NO:1］をチェックし設定をクリックします。

―――ファイルが読み込まれたら新しいフレームに画像が表示される―――

④ 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［画像ファイル入力(ID:4)］コンポーネント上で右クリック－［イベント処理追加］
－［データ生成イベント］とクリックします。

⑤ イベントの接続先コンポーネントを選びます。
左側の［画像ファイル入力(ID:4)］コンポーネントの［データ生成イベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［イメージビューワー(ID:6)］をクリックします。

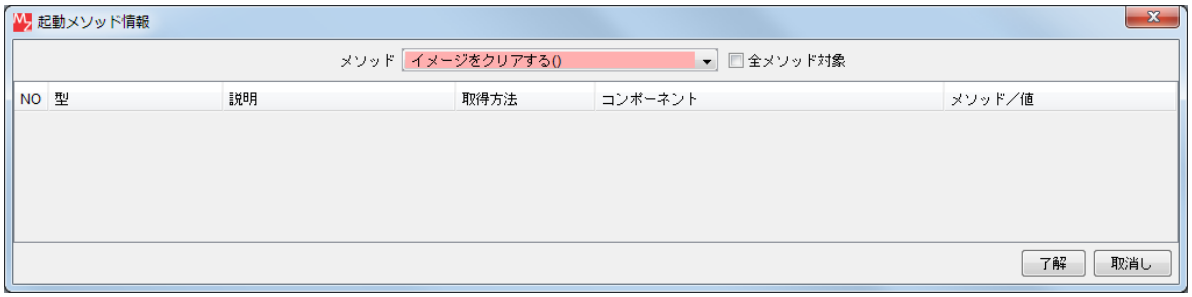
⑥　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の　をクリックします。
［イメージデータを設定する(Image)］をクリックします。
引数を設定します。
説明：イメージ
取得方法：イベント内包
メソッド／値：イベント対象データ
設定後、了解ボタンをクリックします。



―――ボタンが付いているフレームが閉じたら、[イメージビューアー]に表示されている画像を
クリアし、フレームも閉じる―――

⑦　イベントの接続先コンポーネントを選びます。
左側の［フレーム(ID:1)］コンポーネントの［アクションイベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［イメージビューワー(ID:6)］をクリックします。

⑧　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の　をクリックします。
［イメージをクリアする()］をクリックします。
設定後、了解ボタンをクリックします。



⑨ イベントの接続先コンポーネントを選びます。
左側の［フレーム(ID:1)］コンポーネントの［アクションイベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［フレーム(ID:5)］をクリックします。

⑩ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の をクリックします。
［フレームを閉じる()］をクリックします。
設定後、了解ボタンをクリックします。

⑪ ［イメージビューワー(ID:6)］上の画像をクリアし、［フレーム(ID:5)］コンポーネントを閉じてから
アプリケーションを終了するように変更します。
［イメージビューワー(ID:6)］、［フレーム(ID:5)］コンポーネントと
［アプリケーション］コンポーネントを入れ替えます。
［アプリケーション］コンポーネントをドラッグして一番下に移動します。

⑫ ここまでの動きを確認します。
実行（設定可）で実行します。

Step.4 **イメージビューワーの設定変更**

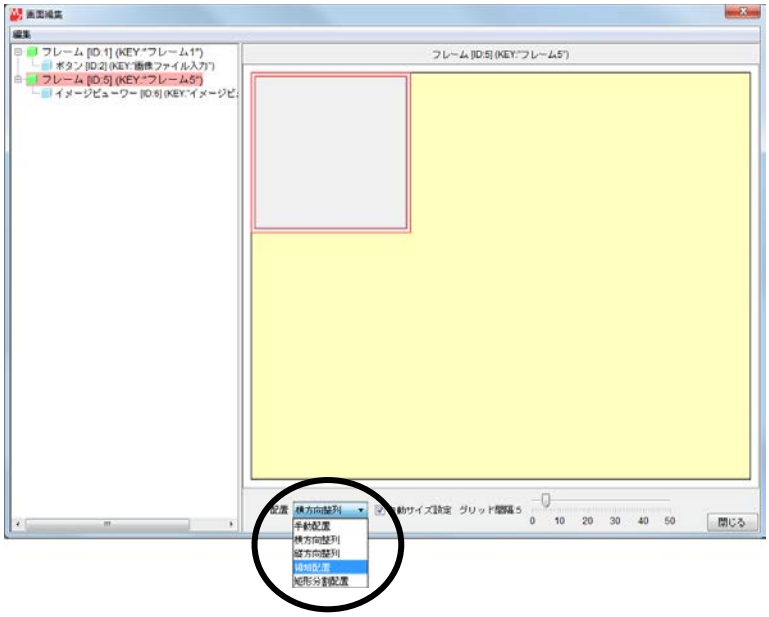
イメージビューワーの設定を変更しましょう。

操 作 ウィンドウのサイズ変更に伴って、イメージビューワーのサイズも変更するようにしましょう（画像のサイズは変わりません）。

① 画面編集をクリックします。

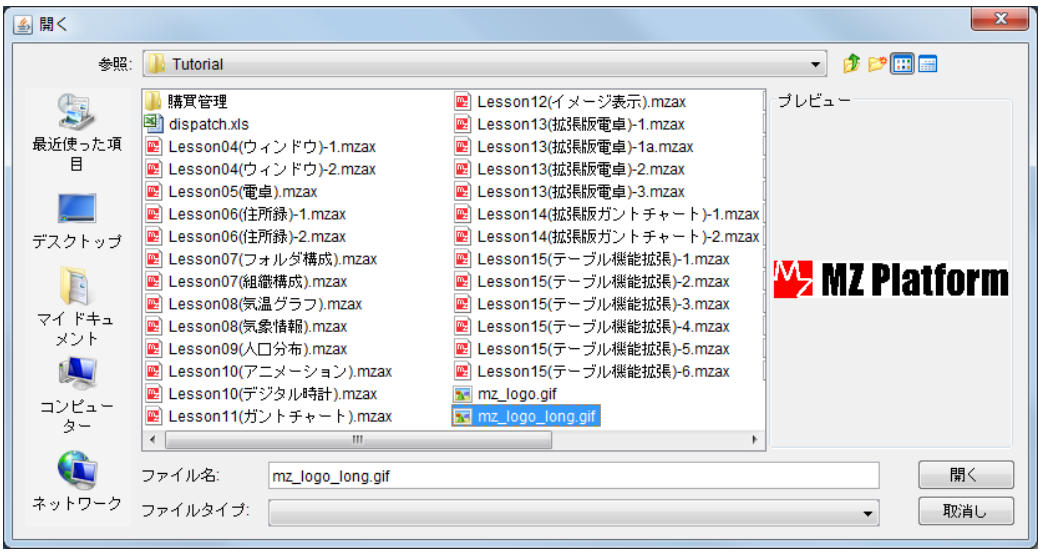
② 左側の領域の［フレーム(ID:5)］をクリックして選択します。

③ ［フレーム(ID:5)］をクリックし、［配置］－［領域配置］に変更します。



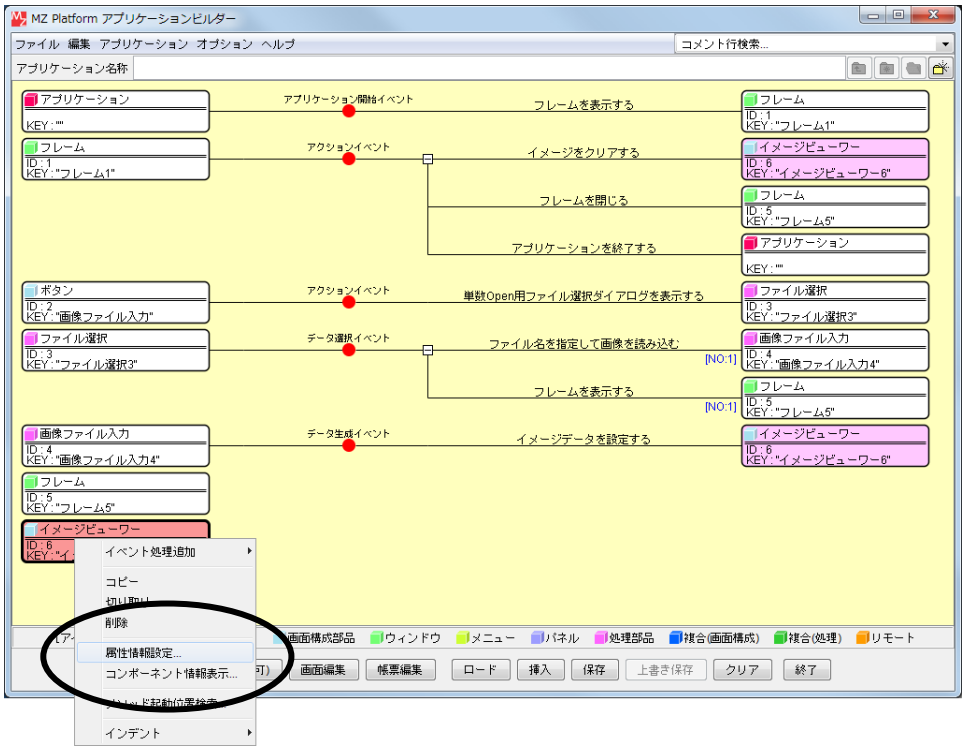
④ 閉じるをクリックします。

⑤ 実行をクリックし、動作確認します。
ここでは画像ファイル「mz_logo_long.gif」を使用して確認します。

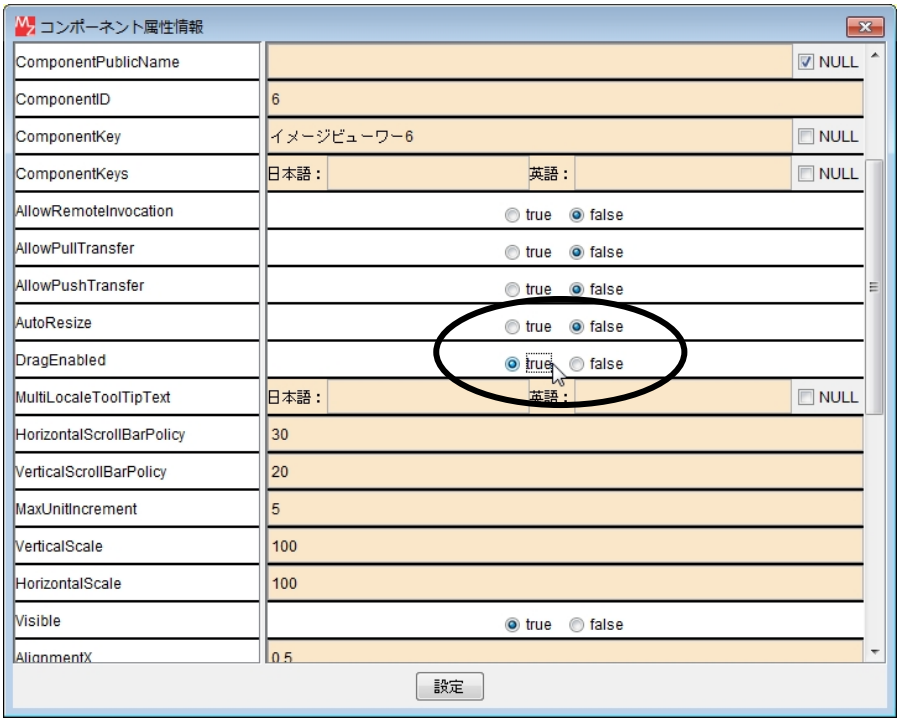


操 作 ウィンドウの中で画像をドラックできるようにしましょう。

① ［イメージビューワー］コンポーネントの上で右クリックー［属性情報設定...］をクリックします。



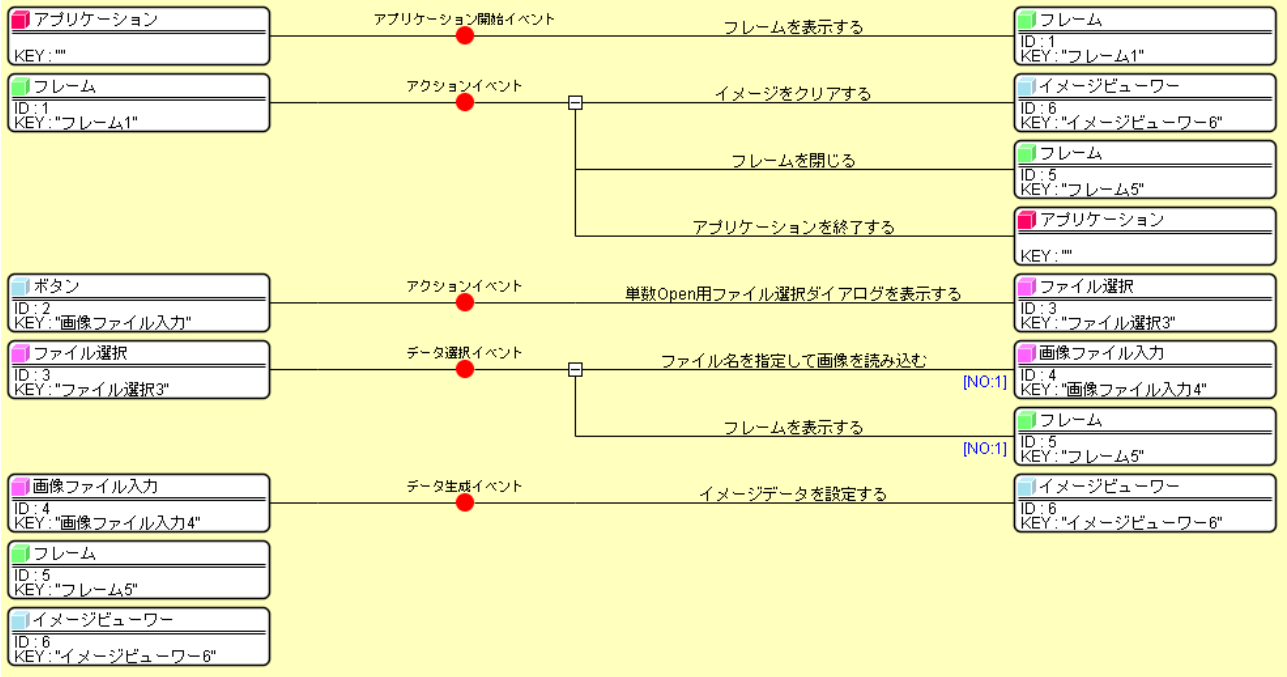
② ［DragEnabled］を［true］にします。



③ ［実行］ボタンをクリックし、ウィンドウの中で画像がドラッグできることを確認します。
ここでは画像ファイル「mz_logo_long.gif」を使用して確認します。

まとめ

ここまで進めるとビルダー上では以下のようになります。

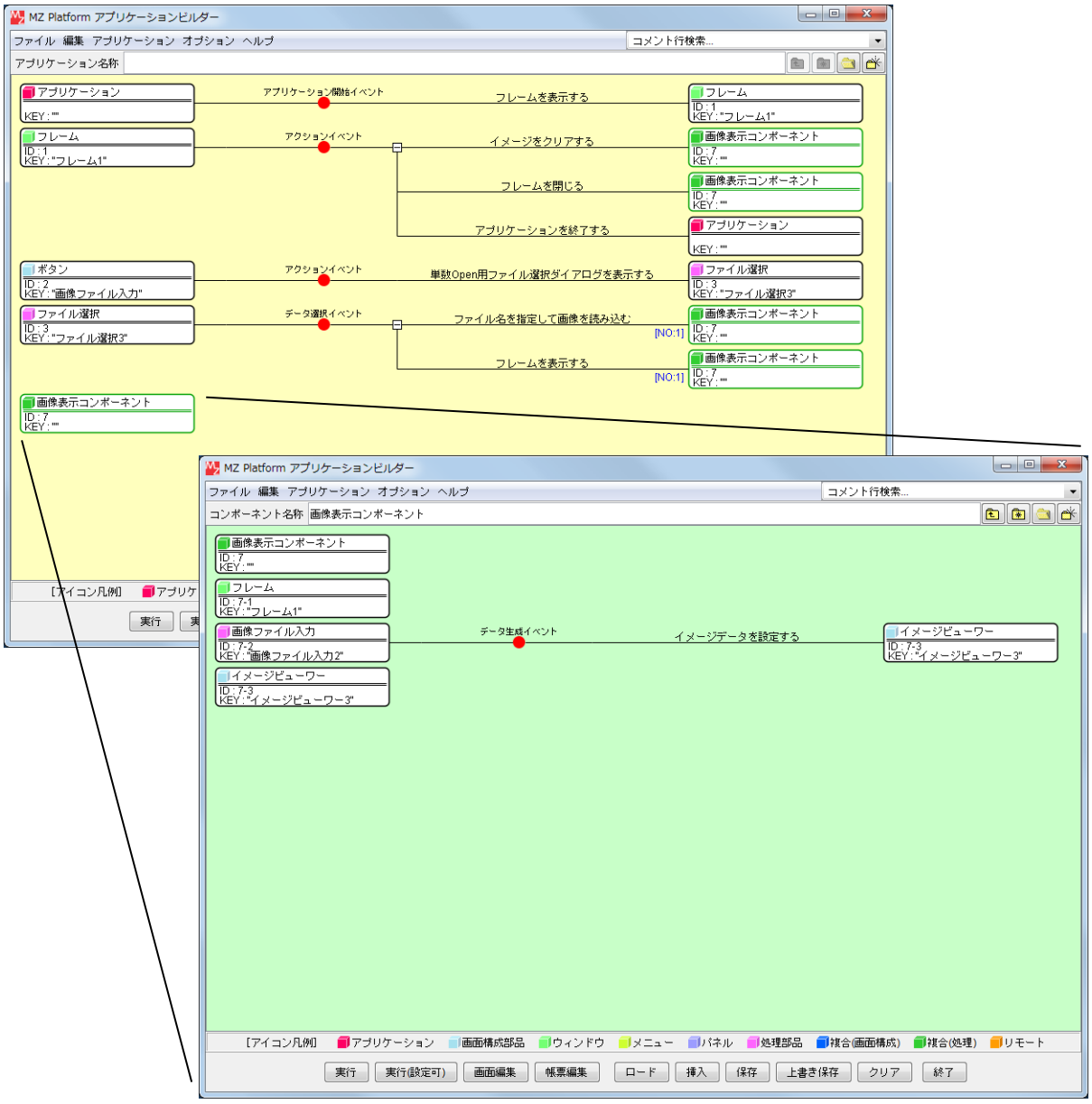


Step.5 **複合コンポーネントによる階層化**

複合コンポーネントとは、いくつかのコンポーネントとひとまとめにした新たなコンポーネントです。
以下のような場合に便利です。

1．繰り返し使われる操作をまとめておく
2．ビルダー上の記述が長くなってしまった場合に整理する
3．機能単位にまとめておき開発作業の効率を上げたい
4．後のメンテナンス時に見やすくしたい

複合コンポーネントは、違う階層にコンポーネントをまとめておく方法です。



考え方

［画像ファイル］コンポーネント、［フレーム］コンポーネント、［イメージビューワー］コンポーネントを、以下の手順で複合コンポーネント内にまとめます：

1．［複合コンポーネント］を追加する

2．［画像ファイル］コンポーネント、［フレーム］コンポーネント、
　　［イメージビューワー］コンポーネントを複合コンポーネントに追加する
3．元の階層から［画像ファイル］コンポーネント、［フレーム］コンポーネント、
　　［イメージビューワー］コンポーネントの３つのコンポーネントを削除する

準 備ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■複合コンポーネント	1	

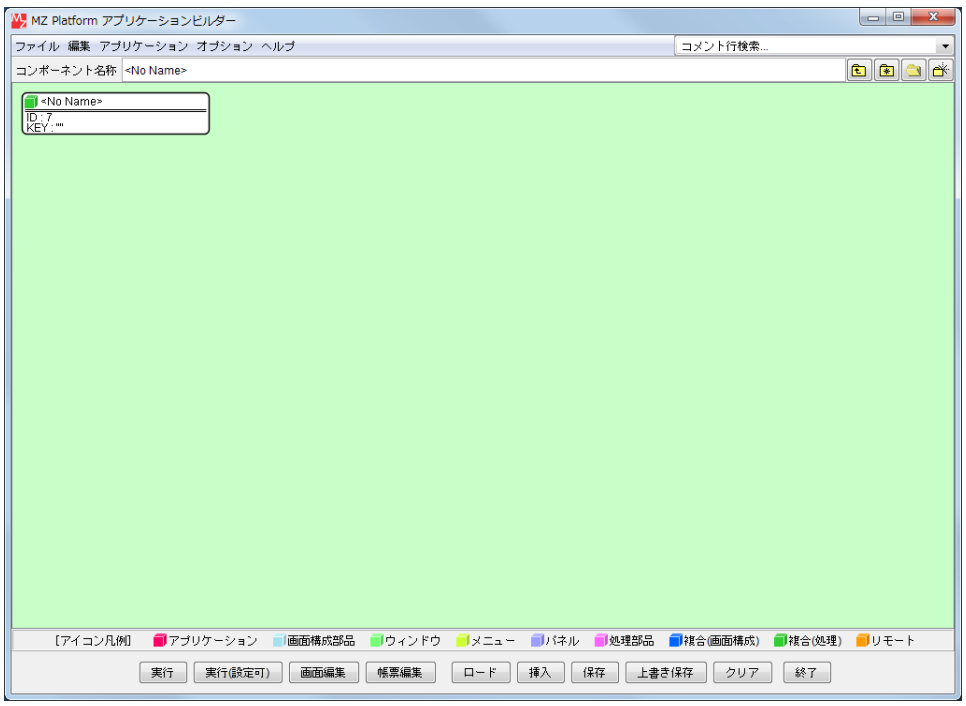
操 作複合コンポーネントを追加しましょう。

① 必要なコンポーネントを追加します。
　作業領域で右クリック－［複合コンポーネント作成］－［コンポーネント］を追加します。

② 追加した［複合コンポーネント］をダブルクリックします。

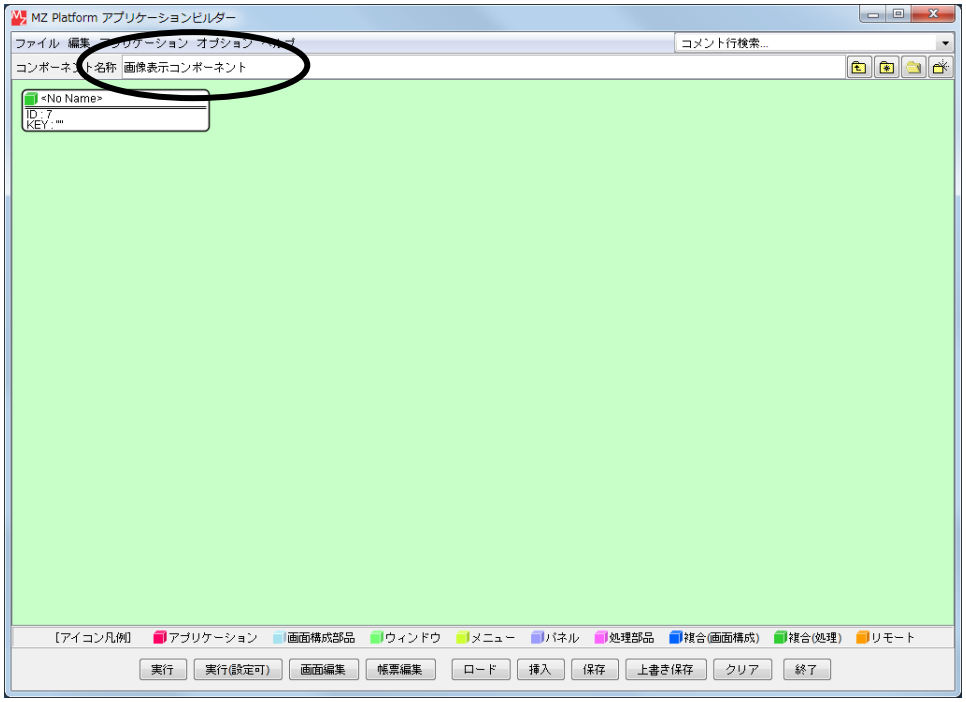
確認

［複合コンポーネント］の中に入ります。画面が緑色に変わります。

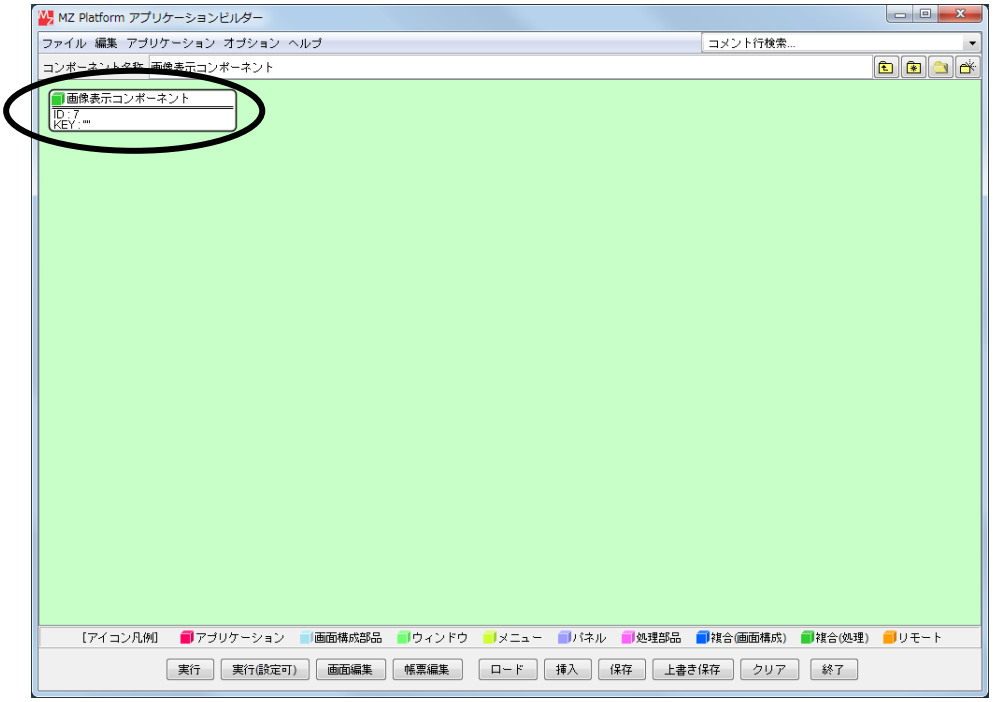


③　複合コンポーネントに名前を付けます。

［コンポーネント名称］の「<No Name>」を消して「画像表示コンポーネント」と入力します。



コンポーネントに名前が付きます。



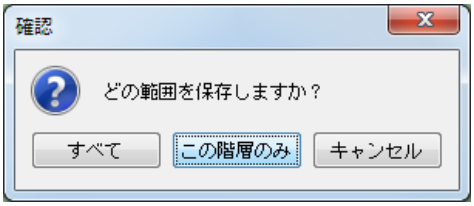
知っていると便利！

複合コンポーネントの中に入っている時、保存ボタンを押して表示されるダイアログでこの階層のみボタンを押すと、複合コンポーネントだけをアプリケーションとは別に保存することができます。

デフォルトではXML形式（拡張子：.mzcx）とバイナリ形式（.mzcs）の2種類で保存されます。

この時 MZPlatform のインストールフォルダー¥AP_DATA_COMB フォルダ（デフォルトではC:¥MZPlatform¥3.2¥AP_DATA_COMB）の下にファイルを保存すると、[複合コンポーネント追加]の際、ファイルが一覧表示されるようになります。

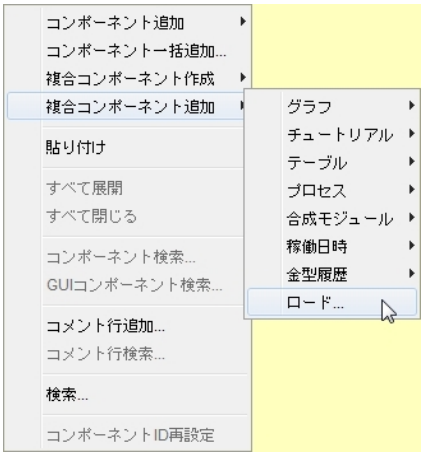
① 複合コンポーネント内で保存ボタンを押し、表示されるダイアログでこの階層のみボタンをクリックします。



② 保存フォルダに「AP_DATA_COMB」を選択します。フォルダ内にさらに新規フォルダを作成しても構いません。

③ [複合コンポーネント追加]時に内容が一覧表示されます。

「AP_DATA_COMB」フォルダ以外に保存してあるファイルを追加する際には、[ロード...]をクリックします。



Step.6 複合コンポーネントの利用

複合コンポーネントを利用しましょう。

１）複合コンポーネントの作成

複合コンポーネントの中を作ります。これまで使用していた階層と同じ方法で作成します。

準 備

ここでは以下のコンポーネントを使用します。

コンポーネント名	必要数	
■複合コンポーネント （画像表示コンポーネント）	(1)	
■フレーム	1	［画面構成部品］－［ウィンドウ］－［フレーム］
■画像ファイル入力	1	［入出力］－［ファイル］－［画像ファイル入力］
■イメージビューワー	1	［画面構成部品］－［グラフィックス］－ ［イメージビューワー］

操 作

① 必要なコンポーネントを追加します。
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［ウィンドウ］－［フレーム］、
作業領域で右クリック－［コンポーネント追加］－［入出力］－［ファイル］－［画像ファイル入力］、
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［グラフィックス］
－［イメージビューワー］とクリックします。

画面編集

① 画面を作成します。
画面編集をクリックします。
［イメージビューワー(ID:7-3)］コンポーネントを複合コンポーネントのフレームに追加します。
［画面編集］画面上で右クリック－［イメージビューワー(ID:7-3)］コンポーネントと
クリックします。
追加できたら閉じるをクリックし、ビルダー画面に戻ります。

接続確認

コンポーネント同士の接続を確認します。

接続は複合コンポーネントの上の階層（前の Step）と同じです。

ファイルが読み込まれたら新しいフレームに画像が表示される

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■画像ファイル入力（ID:7-2）
発生イベント	データ生成イベント
接続先コンポーネント	■イメージビューワー（ID:7-3）
起動メソッド	イメージデータを設定する(Image)
＜引数＞	説明：イメージデータ 取得方法：イベント内包 メソッド／値：イベント対象データ

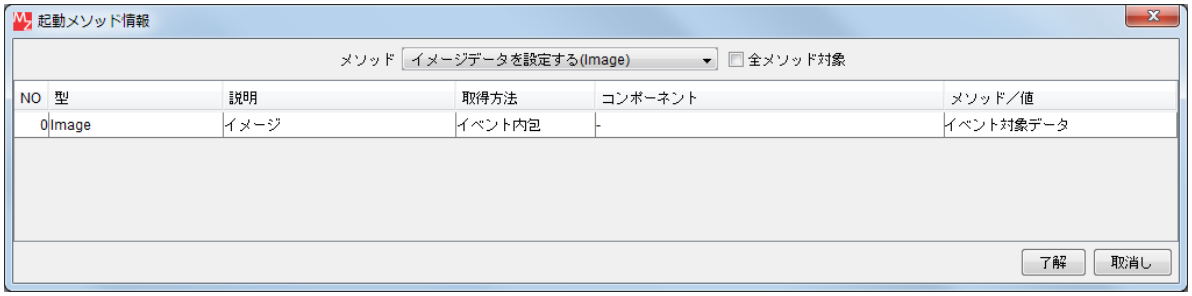
操 作

複合コンポーネントの中を作りましょう。

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［画像ファイル入力(ID:7-2)］コンポーネント上で
右クリックー［イベント処理追加］ー［データ生成イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［画像ファイル入力(ID:7-2)］コンポーネントの［データ生成イベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［イメージビューワー(ID:7-3)］をクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［イメージデータを設定する(Image)］をクリックします。
引数を設定します。
説明：イメージ
取得方法：イベント内包
メソッド／値：イベント対象データ
設定後、了解ボタンをクリックします。



知っていると便利！

複合コンポーネントの中を作成する際、コンポーネントの［コピー］及び［貼り付け］を使用すると効率良く作成できます。

① 複合コンポーネントにするコンポーネント上で右クリック－[コピー]とクリックします。
② ［複合コンポーネント］をダブルクリックし、複合コンポーネント内に入ります。
③ 作業領域で右クリック－[貼り付け]とクリックします。

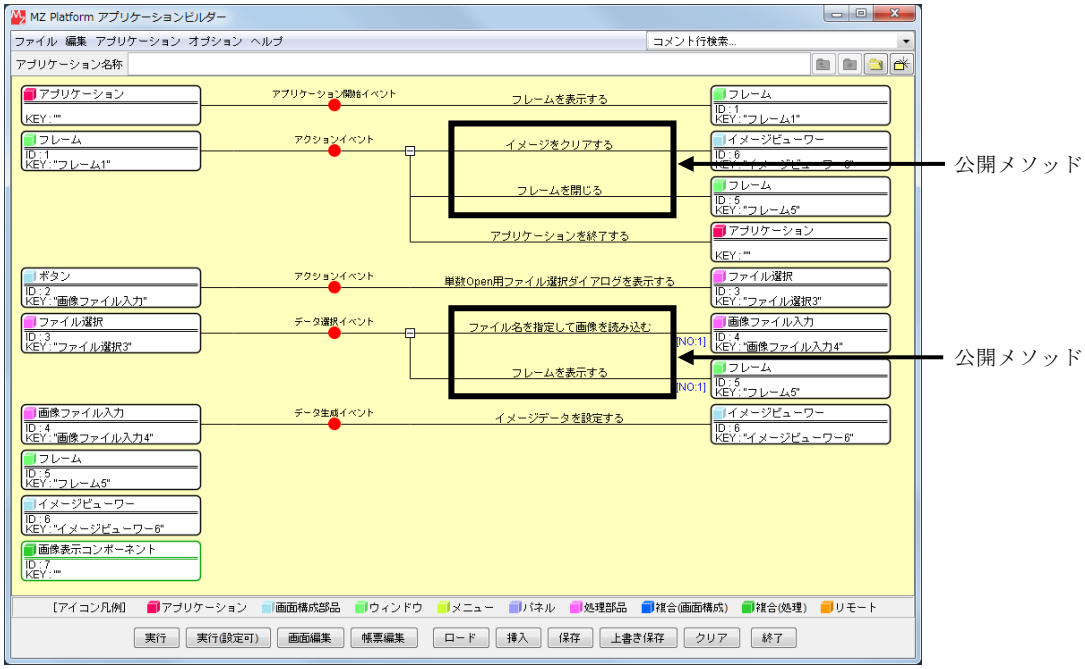
２）複合コンポーネントの中のメソッドを公開する

複合コンポーネントの中に設定したメソッドを上の階層から呼び出します。

上の階層からメソッドを呼び出すには複合コンポーネントの中のメソッドを上の階層に「公開」して使えるようにする必要があります。

「公開」するメソッドは上の階層で必要なものだけを公開します。

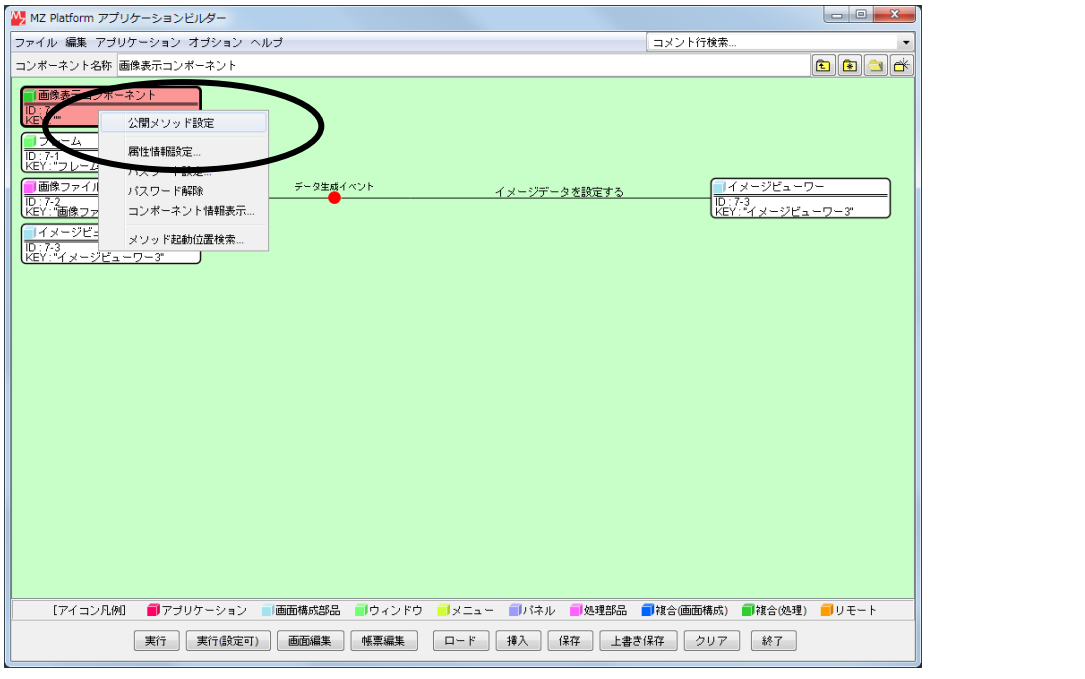
ここで必要なのは以下の４つのメソッドであることがわかります。これらを公開します。



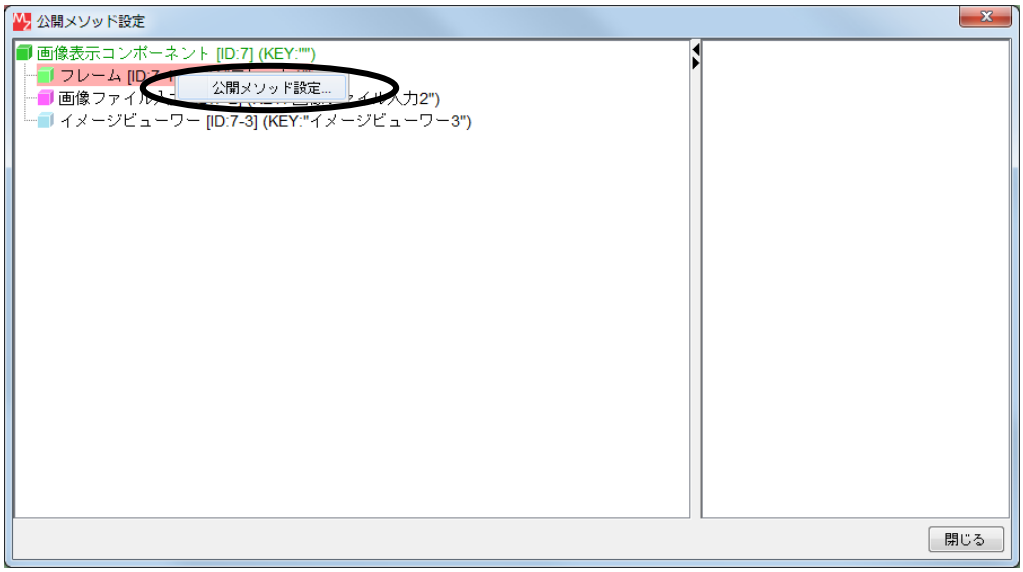
操 作 複合コンポーネントのメソッドを公開しましょう。

① ［画面表示コンポーネント］をダブルクリックして複合コンポーネントに入ります。

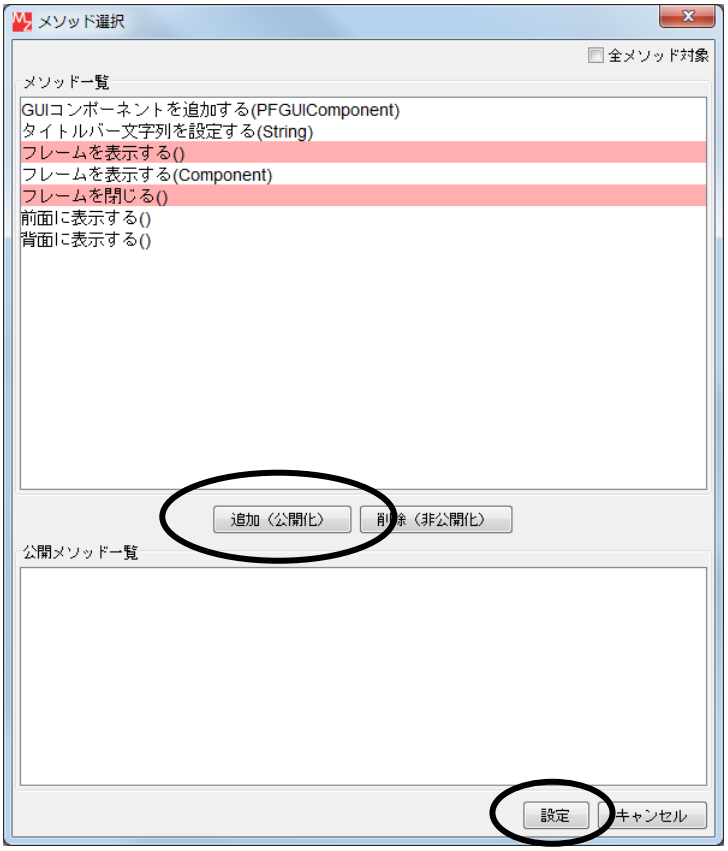
② ［画面表示コンポーネント］の複合コンポーネントで［右クリック］－［公開メソッド設定］をクリックします。



③　公開メソッド設定の窓が表示されます。
公開するメソッドを選びます。
［フレーム］から選びます。
［フレーム(ID:7-1)］で右クリック－［公開メソッド設定...］をクリックします。

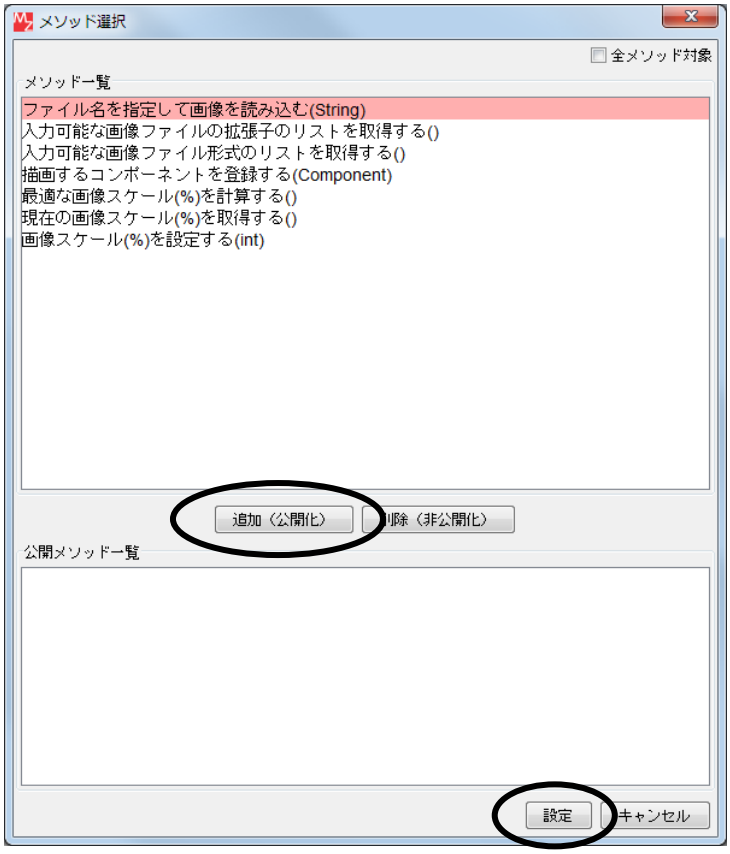


④　［フレームを表示する()］［フレームを閉じる()］を公開します。
［フレームを表示する()］をクリック、［フレームを閉じる()］を【Shift】＋クリックします。
追加（公開化）をクリックし、設定をクリックします。



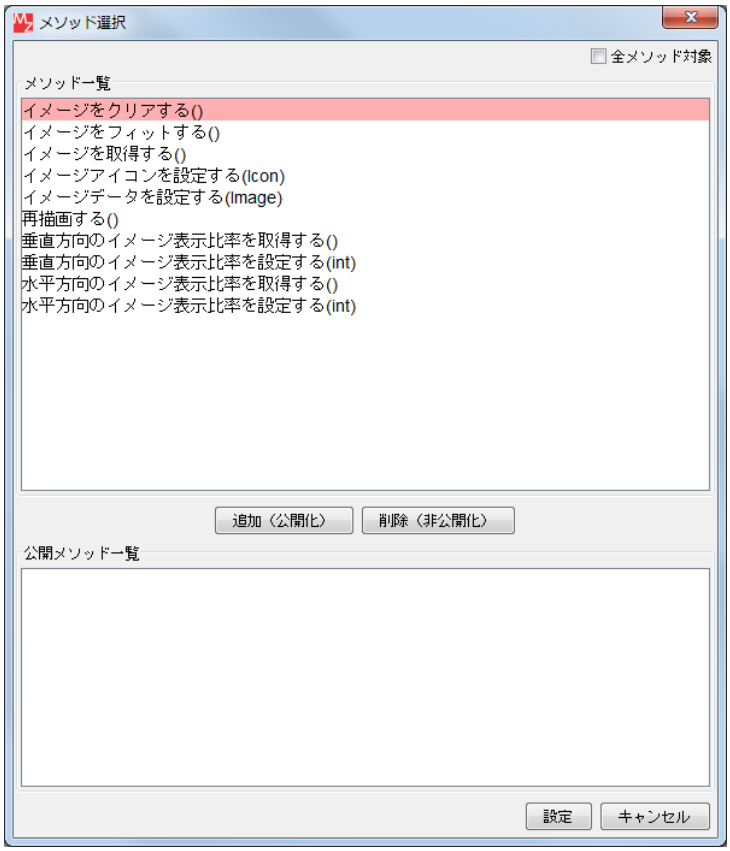
⑤　公開するメソッドを選びます。
［画像ファイル入力(ID:7-2)］から選びます。
［画像ファイル入力(ID:7-2)］で右クリック－［公開メソッド設定...］をクリックします。

⑥　［ファイル名を指定して画像を読み込む(String)］をクリックします。
追加（公開化）をクリックし、設定をクリックします。

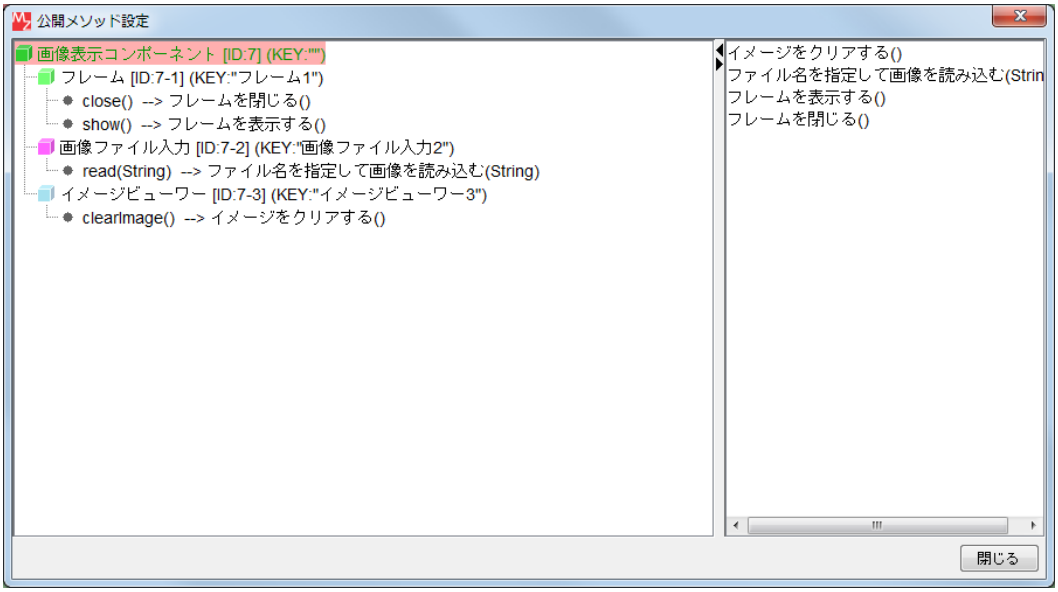


⑦　公開するメソッドを選びます。
［イメージビューワー(ID:7-3)］から選びます。
［イメージビューワー(ID:7-3)］で右クリック－［公開メソッド設定...］をクリックします。

⑧　［イメージをクリアする()］をクリックします。
追加（公開化）をクリックし、設定をクリックします。

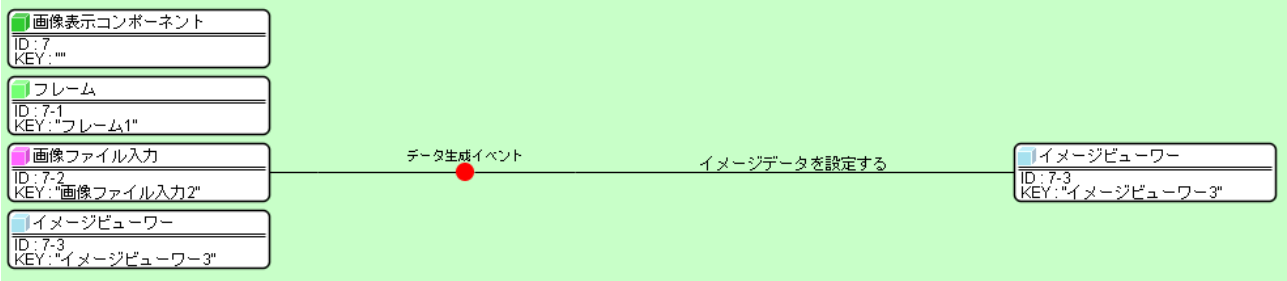


公開メソッドは以下のようになります。



まとめ

ここまで進めるとビルダー上では以下のようになります。

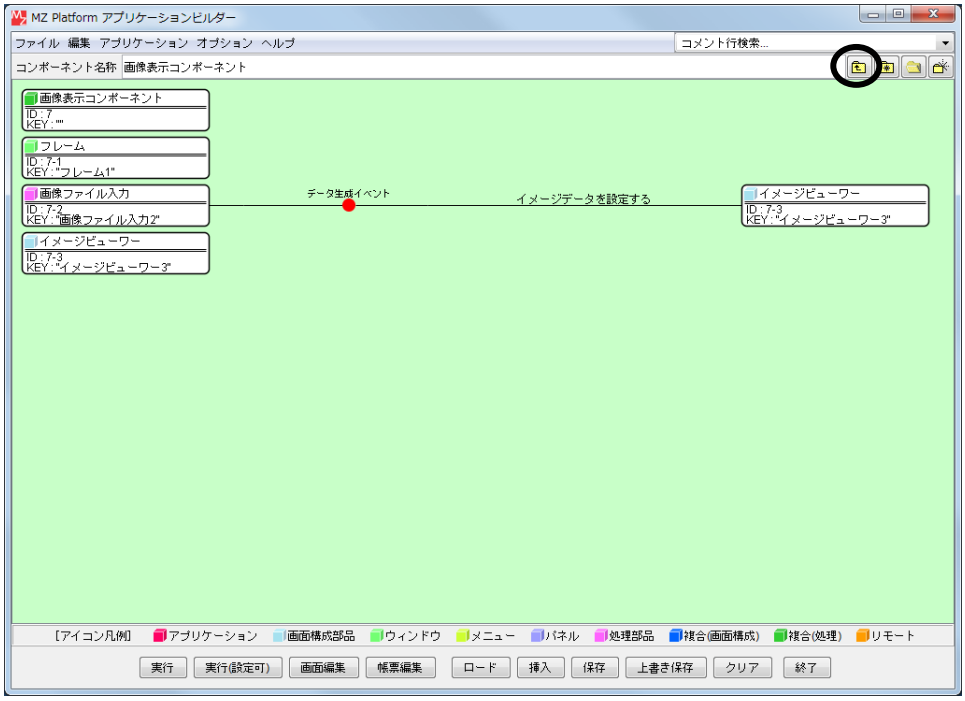


３）公開してあるメソッドを上の階層から使用する

複合コンポーネントで公開したメソッドを上の階層から使用します。
現在設定されているメソッドを削除して複合コンポーネントのメソッドに置き換えます。

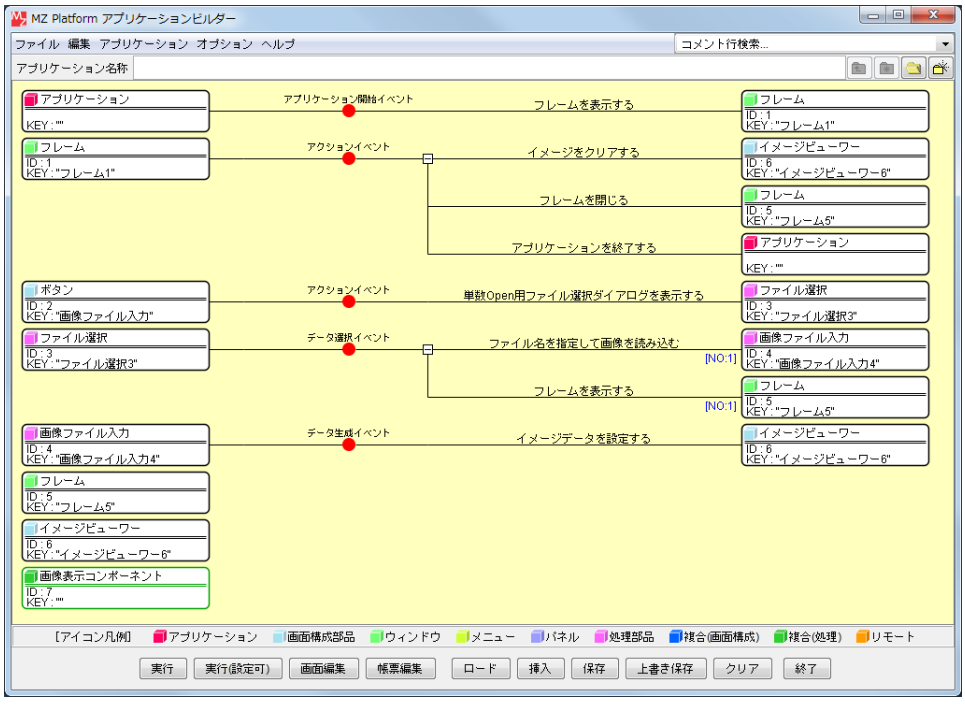
操 作 現在設定されているメソッドを削除しましょう。

① 複合コンポーネントから元の階層に戻ります。
右上の［編集サポートボタン］をクリックして１階層上に上がります。

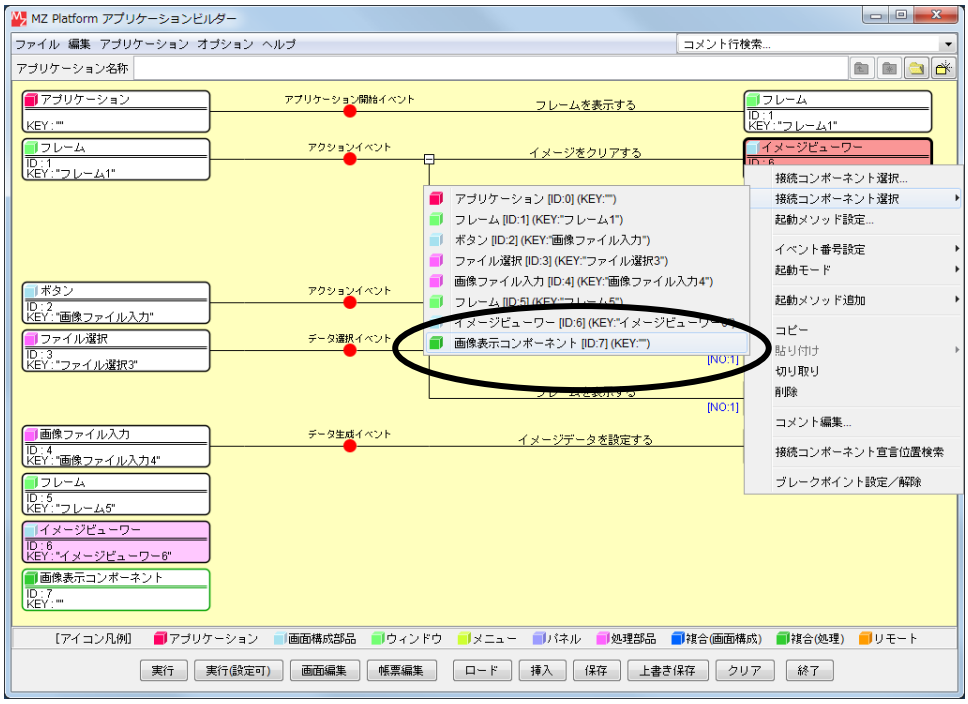


確認

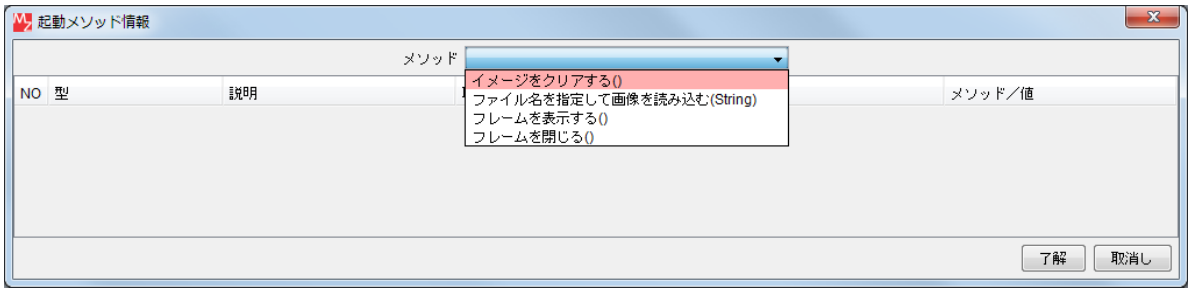
元の階層に戻ります。



② 複合コンポーネントになっている接続コンポーネントを変更します。
［フレーム(ID:1)］と接続されている［イメージビューワー(ID:6)］を
［画像表示コンポーネント(ID:7)］に変更します。
右側の［イメージビューワー(ID:6)］の上で右クリックー［接続コンポーネント選択］ー
［画像表示コンポーネント(ID:7)］をクリックします。

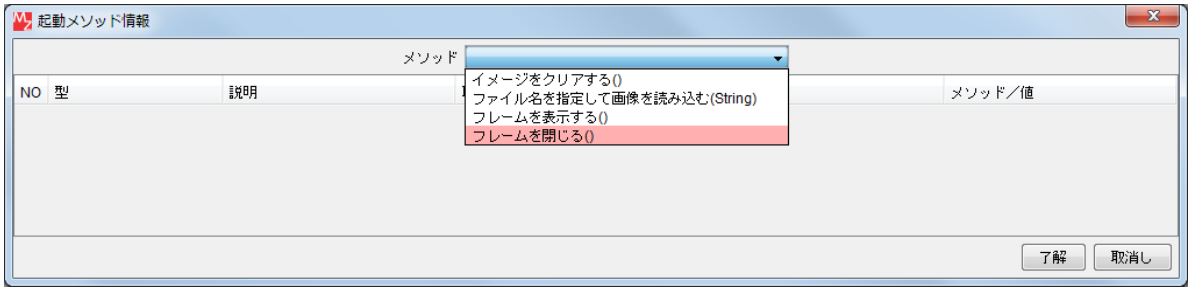


③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［イメージをクリアする()］をクリックします。
設定後、了解ボタンをクリックします。



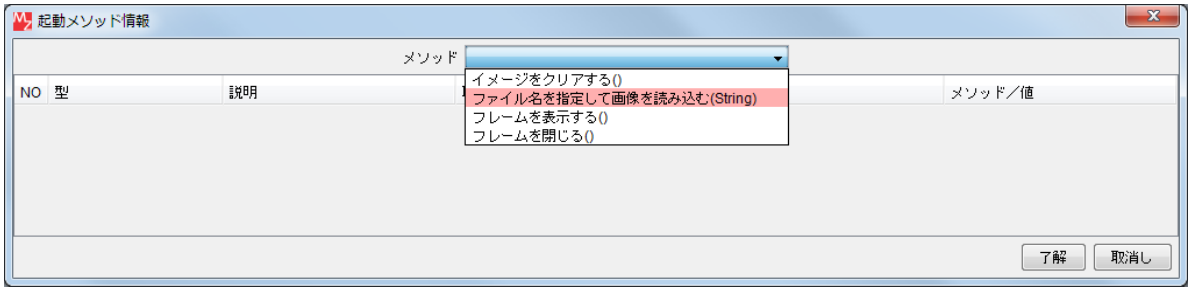
④ ［フレーム(ID:1)］と接続されている［フレーム(ID:5)］を
［画像表示コンポーネント(ID:7)］に変更します。
右側の［フレーム(ID:5)］の上で右クリックー［接続コンポーネント選択］ー
［画像表示コンポーネント(ID:7)］をクリックします。

⑤　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の　をクリックします。
［フレームを閉じる()］をクリックします。
設定後、了解ボタンをクリックします。



⑥　［ファイル選択(ID:3)］と接続されている［画像ファイル入力(ID:4)］を
［画像表示コンポーネント(ID:7)］に変更します。
右側の［画像ファイル入力(ID:4)］の上で右クリック－［接続コンポーネント選択］－
［画像表示コンポーネント(ID:7)］をクリックします。

⑦　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の　をクリックします。
［ファイル名を指定して画像を読み込む(String)］をクリックします。
　説明：読み込むファイル名
　取得方法：イベント内包
　メソッド／値：選択データ
設定後、了解ボタンをクリックします。



⑧　［ファイル選択(ID:3)］と接続されている［フレーム(ID:5)］を
［画像表示コンポーネント(ID:7)］に変更します。
右側の［画像ファイル入力(ID:4)］の上で右クリック－［接続コンポーネント選択］－
［画像表示コンポーネント(ID:7)］をクリックします。

⑨　接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。

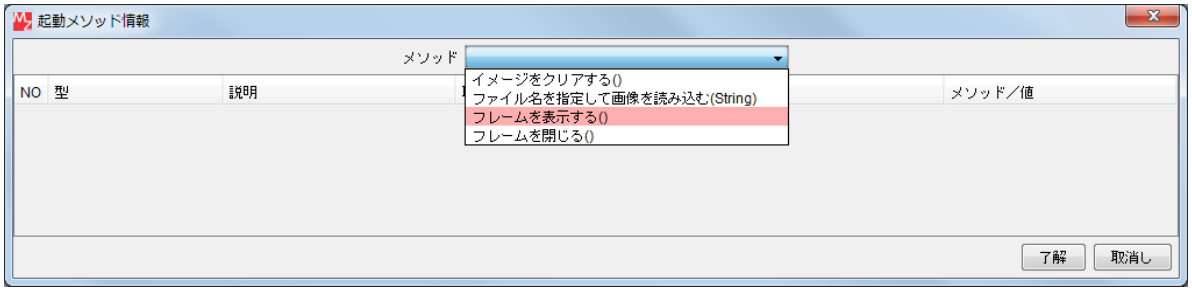
起動メソッド設定画面が表示されます。

起動メソッド（処理）を選びます。

［メソッド］の ▼ をクリックします。

［フレームを表示する()］をクリックします。

設定後、了解ボタンをクリックします。

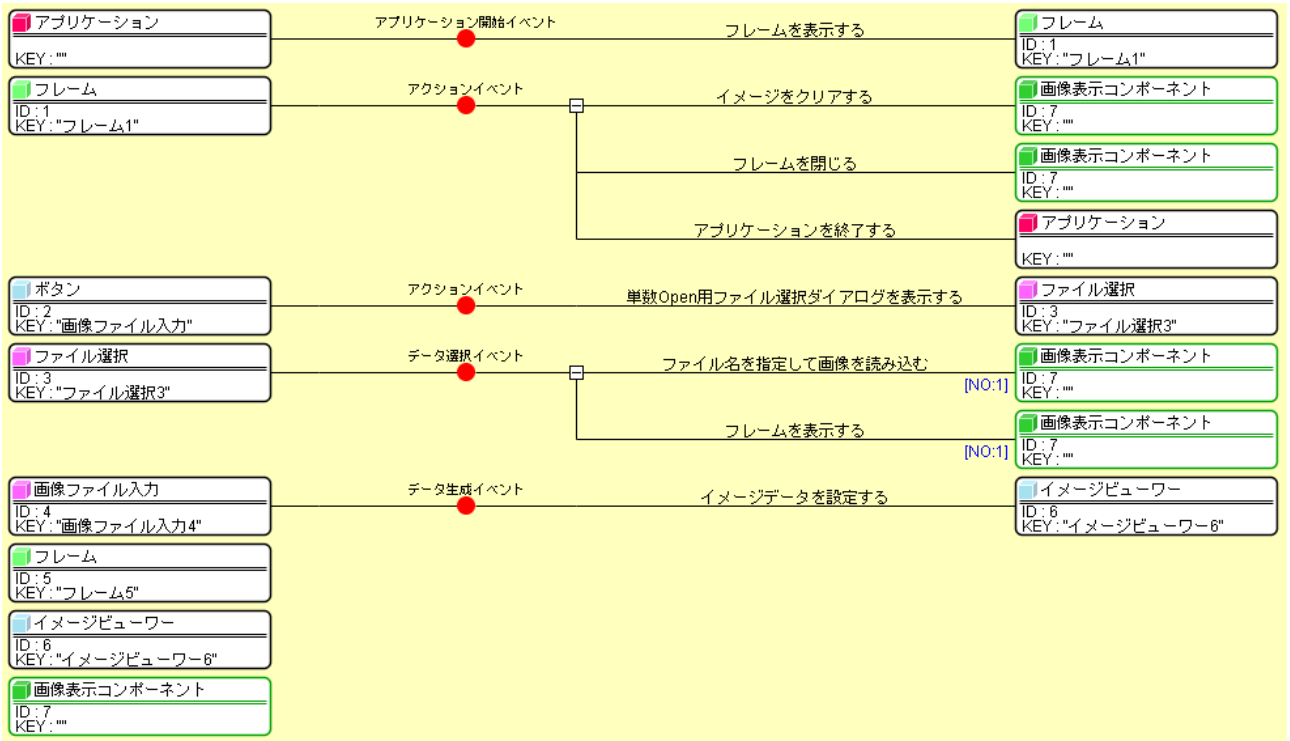


⑩　確認します。

実行（設定可）で実行します。

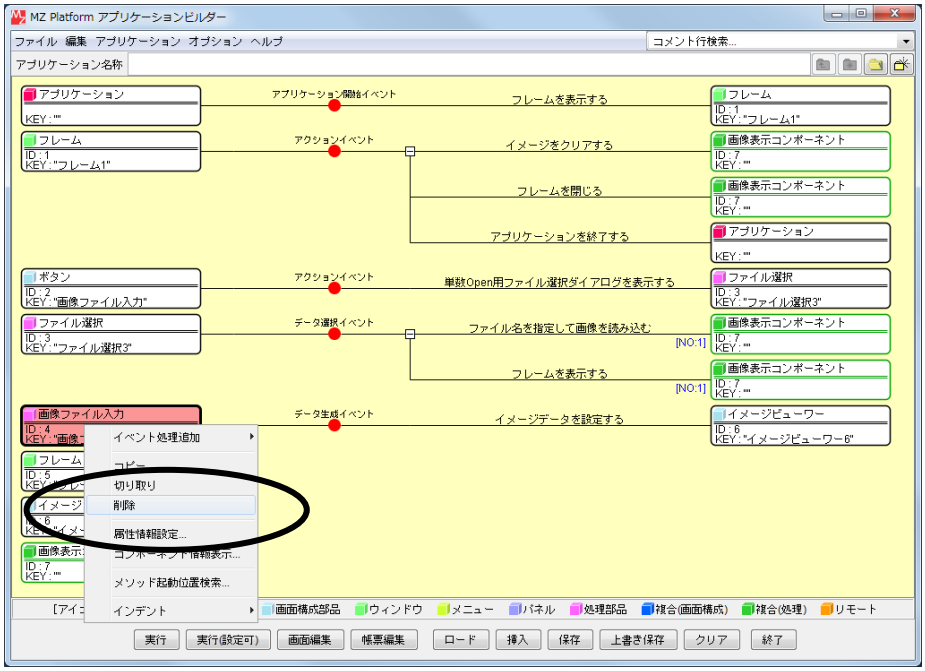
複合コンポーネントを作成する前と同じ動作ができることを確認します。

以下のようになります。

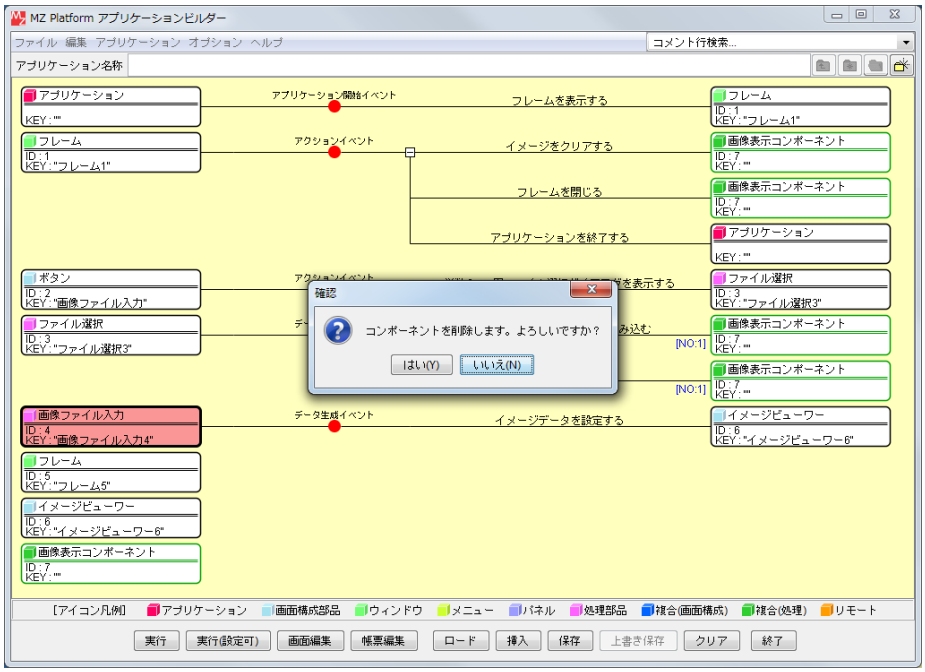


操 作不要なコンポーネントを削除しましょう。

① コンポーネントを削除します。
［画像ファイル入力(ID:4)］を削除します。
［画像ファイル入力(ID:4)］の上で右クリック－［削除］をクリックします。



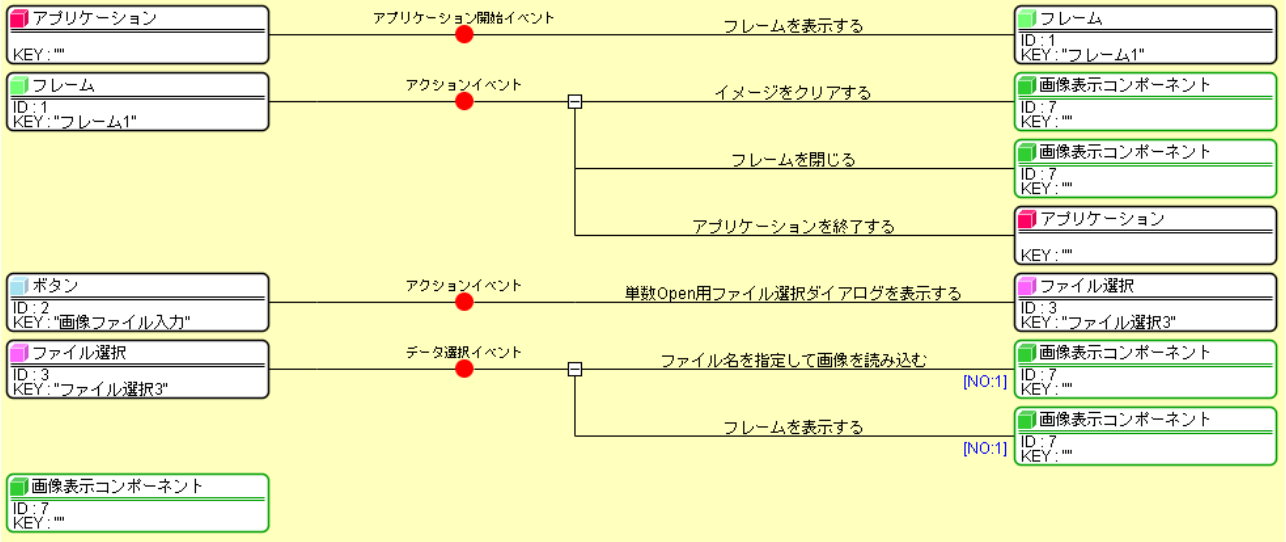
② ［コンポーネントを削除します。よろしいですか？］のメッセージが表示されるので
［はい］をクリックします。



③ ①～②を繰り返して
［フレーム(ID:5)］、［イメージビューワー(ID:6)］を削除します。

まとめ

ここまで進めるとビルダー上では以下のようになります。



Step.7 機能を追加する

ここまで作成してきたアプリケーションに機能を追加します。

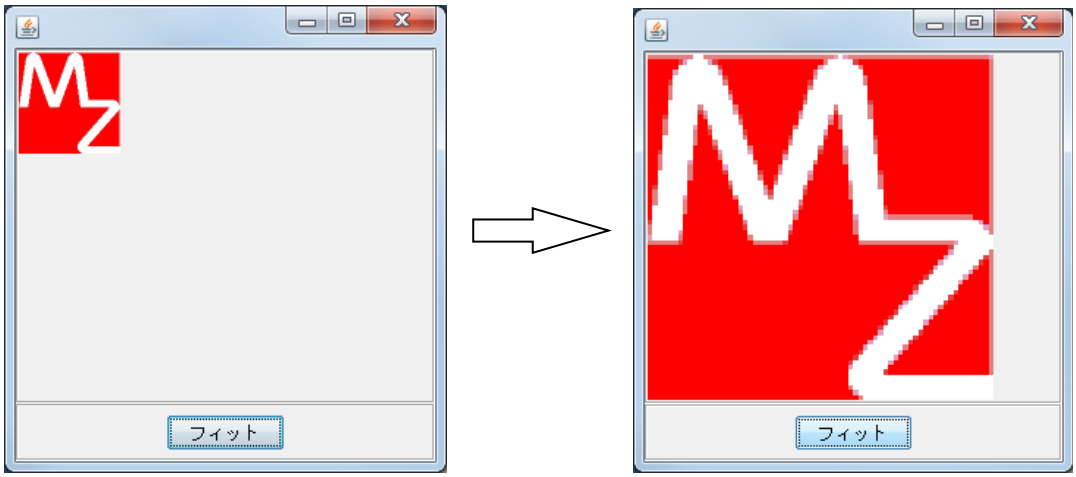
１）画像をウィンドウの領域表示の大きさに合わせる

画像を表示領域の大きさに合わせることができます。
画像の大きさと表示領域の大きさを比較して画像のスケールを計算し、そのスケールを指定して画像を再表示することができます。

画像のスケールについては［イメージビューワー］コンポーネントが計算の機能を持っています。

完成図

画像をウィンドウの表示領域の大きさに合わせて再表示します。



準 備

ここでは以下のコンポーネントを複合コンポーネントに追加します。

コンポーネント名	必要数	
■パネル	1	［画面構成部品］－［パネル］－［パネル］
■ボタン	1	［画面構成部品］－［ボタン］－［ボタン］

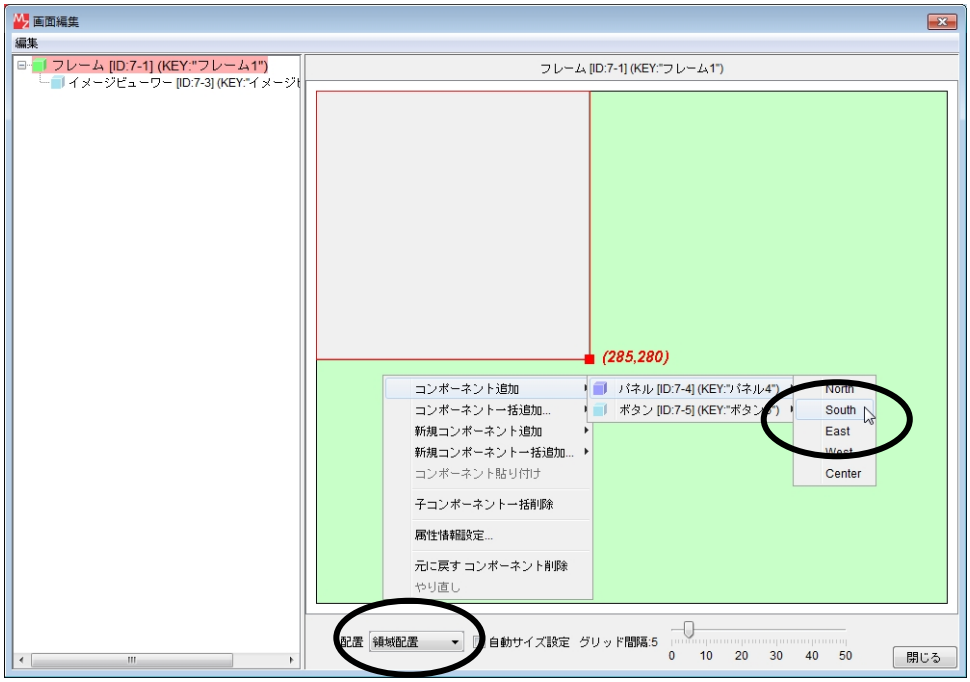
操 作

① 必要なコンポーネントを複合コンポーネントに追加します。
作業領域（緑）で右クリック－［コンポーネント追加］－［画面構成部品］－［パネル］－［パネル］
作業領域（緑）で右クリック－［コンポーネント追加］－［画面構成部品］－［ボタン］－［ボタン］
とクリックします。

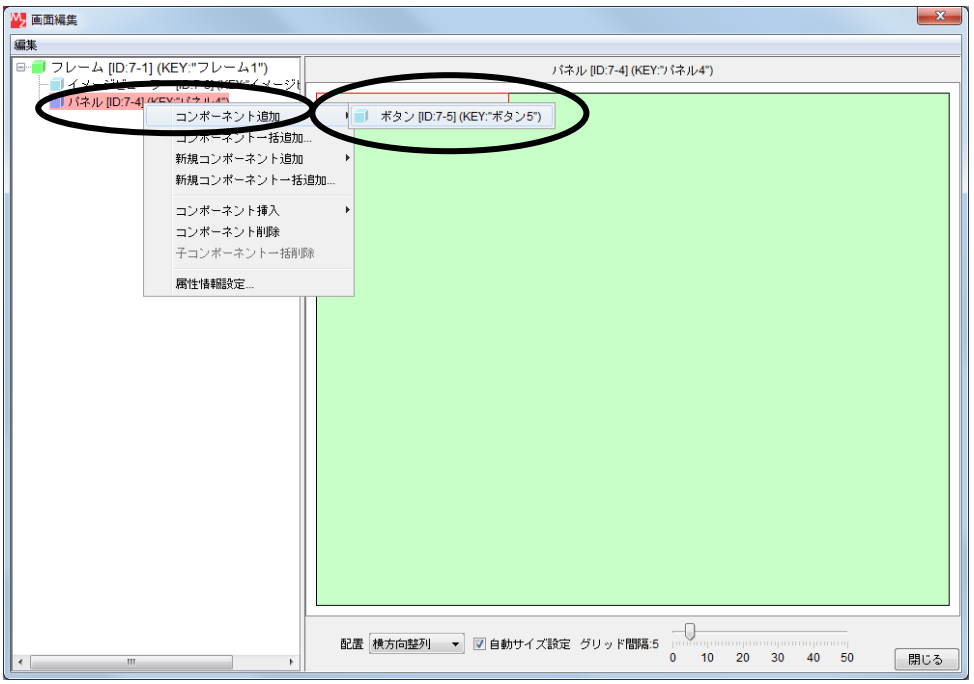
画面編集

① 画面を作成します。
画面編集をクリックします。
［パネル］コンポーネントをフレームに追加します。

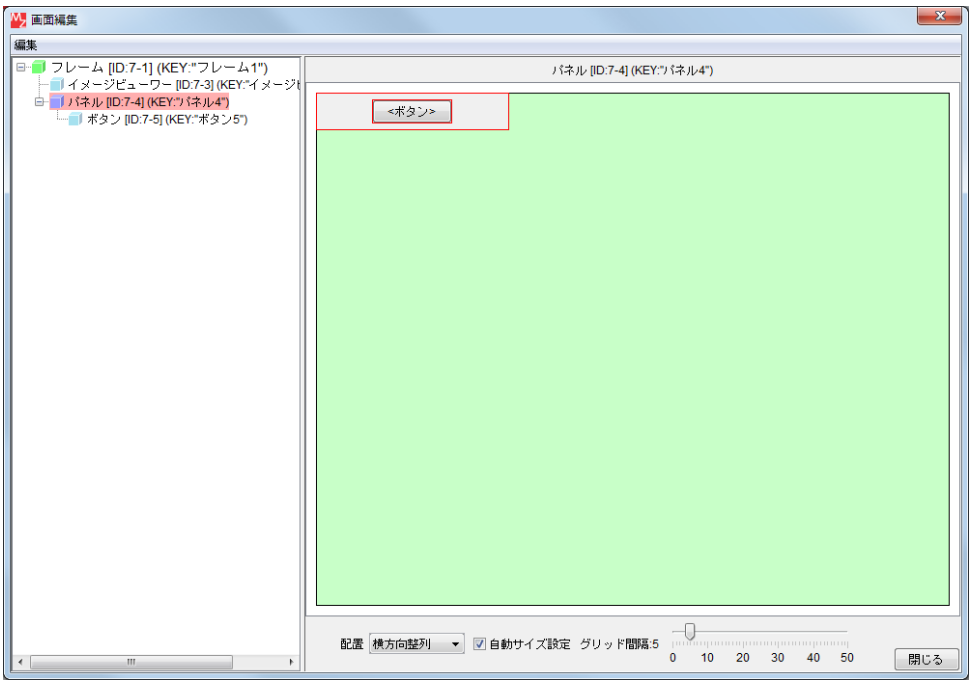
［配置］を［領域配置］に変更します。
画面編集画面上で右クリック－［コンポーネント追加］－［パネル(ID:7-4)］
－［South］とクリックします。



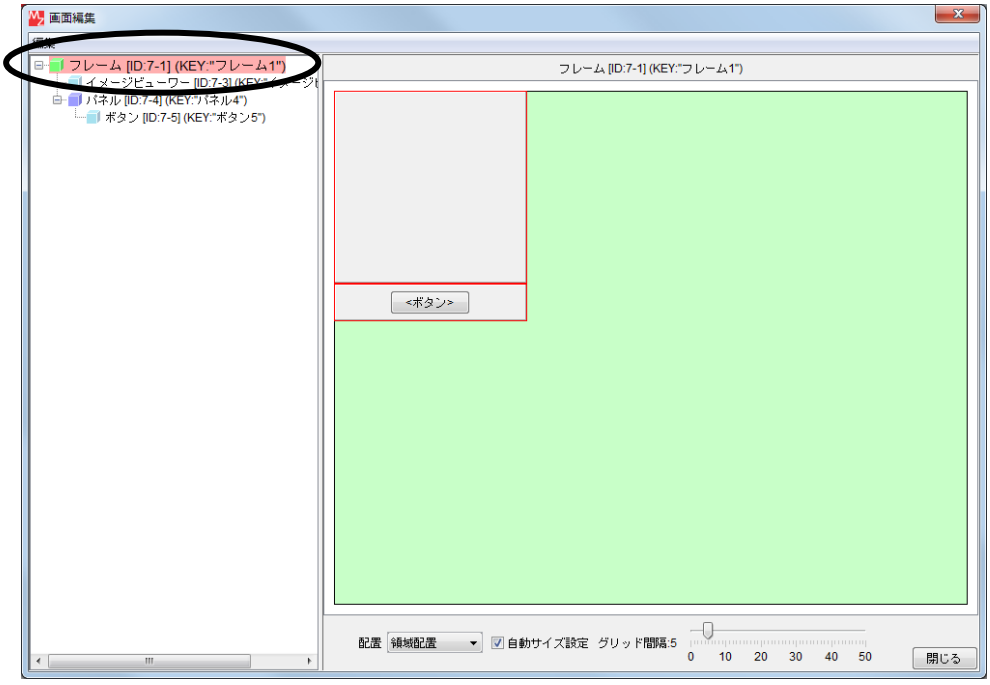
② ［ボタン(ID:7-5)］コンポーネントを［パネル(ID:7-4)］コンポーネントに追加します。
左側の領域を見ると［フレーム(ID:7-1)］コンポーネントに［イメージビューワー(ID:7-3)］コンポーネントと［パネル(ID:7-4)］コンポーネントが並列に追加されています。このうち、［パネル(ID:7-4)］コンポーネントに［ボタン(ID:7-5)］コンポーネントを追加します。［パネル(ID:7-4)］コンポーネント上で右クリックして、［ボタン(ID:7-5)］コンポーネントを追加します。



パネルの上にボタンが追加されます。



[フレーム(ID:7-1)]をクリックすると[イメージビューワー(ID:7-3)]と[パネル(ID:7-4)]、[ボタン(ID:7-5)] の位置が確認できます。



③　[ボタン(ID:7-5)] に「フィット」の文字列を設定します。

[ボタン(ID:7-5)] 上でマウス右クリックし[属性情報設定...]を選択します。コンポーネント情報設定画面内の[TEXT]の欄に「フィット」と設定します。

設定ボタンを押し設定を確定し、閉じるをクリックし画面編集を終了します。

接続確認

コンポーネント同士の接続を確認します。

画像をイメージビューワーにフィットさせる

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（ID:7-5）
発生イベント	アクションイベント
接続先コンポーネント	■イメージビューワー（ID:7-3）
起動メソッド	イメージをフィットする()

操 作

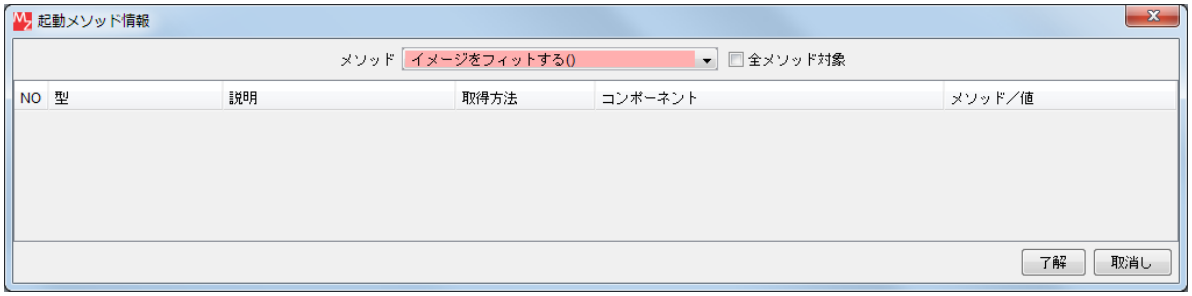
［フィット］ボタンに機能を割り当てましょう。

―――画像をイメージビューワーにフィットさせる―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ボタン(ID:7-5)］コンポーネント上で
右クリック－［イベント処理追加］－［アクションイベント］とクリックします。

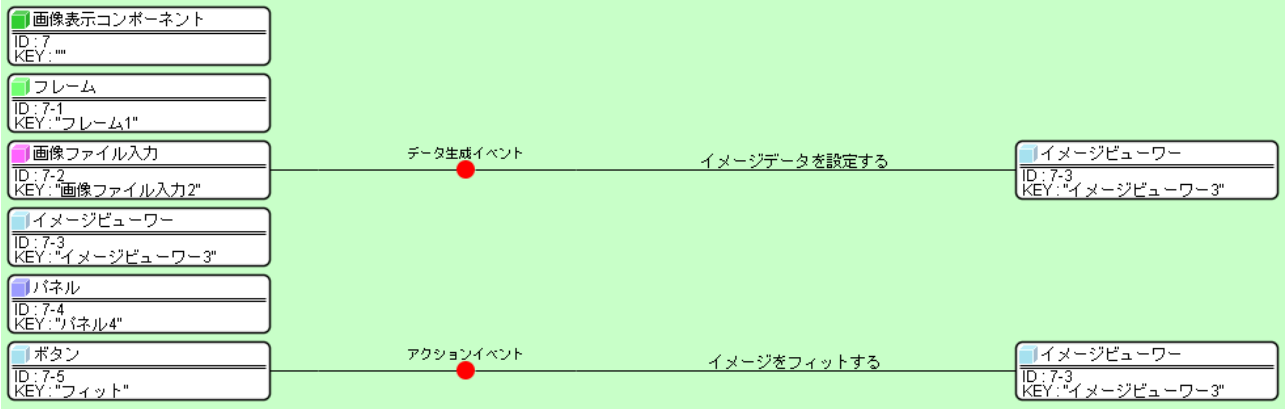
② イベントの接続先コンポーネントを選びます。
左側の［ボタン(ID:7-5)］コンポーネントの［アクションイベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［イメージビューワー(ID:7-3)］コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［イメージをフィットする ()］をクリックします。
設定後、了解ボタンをクリックします。



まとめ

ここまで進めるとビルダー上では以下のようになります。

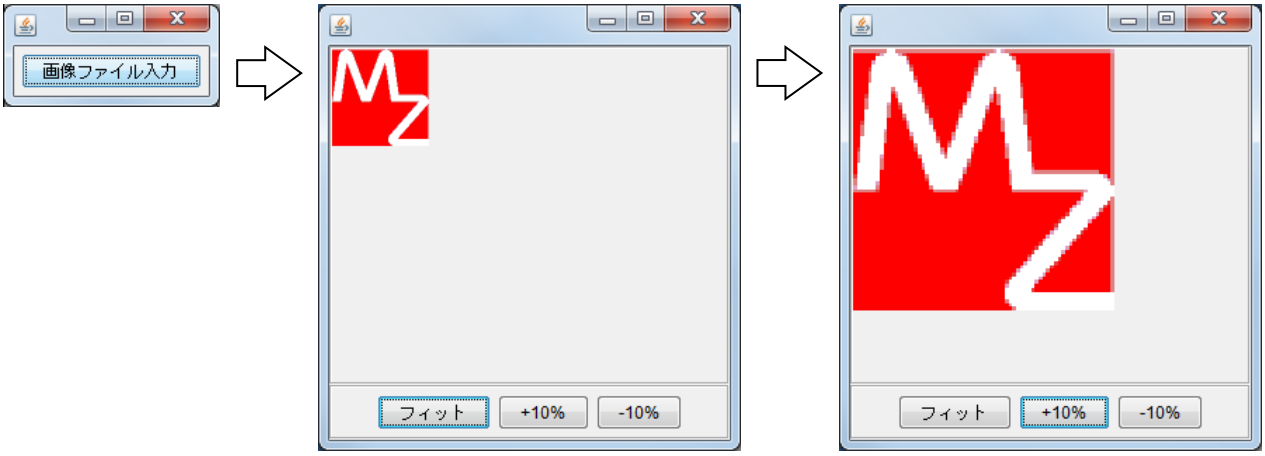


２）画像を拡大・縮小する

画像を拡大・縮小する機能を設定します。10%ずつサイズが変更するようにします。

完成図

画像を拡大・縮小する機能を設定します。



準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■ボタン	2	［画面構成部品］－［ボタン］－［ボタン］
■加算(+)	1	［処理部品］－［演算制御］－［加算(+)］
■比較演算(>)	1	［処理部品］－［条件制御］－［比較演算(>)］

操 作

複合コンポーネント「画像表示コンポーネント」に必要なコンポーネントを追加します。

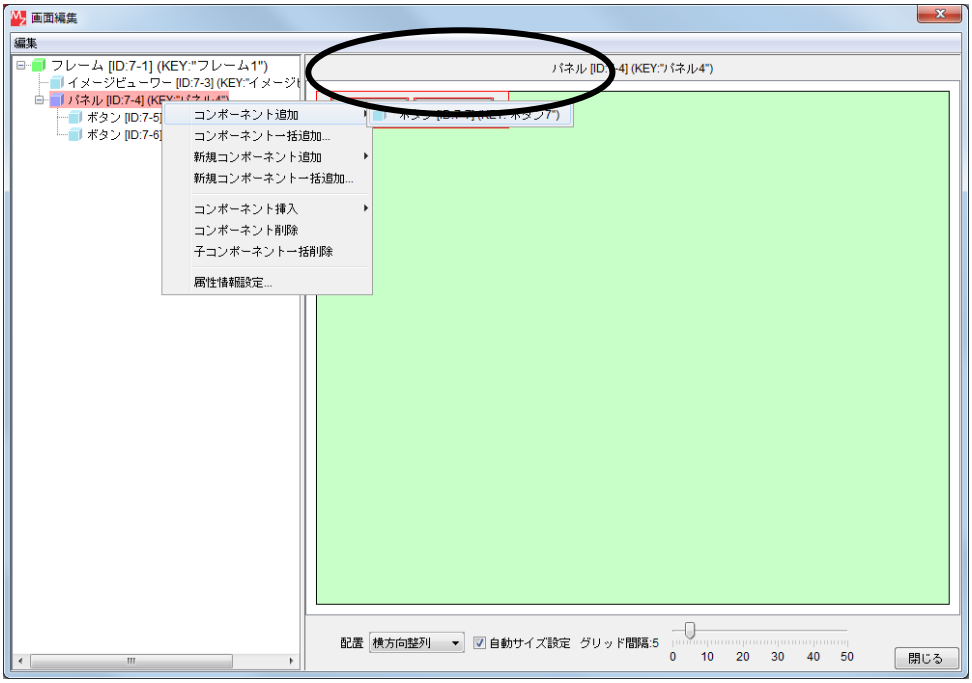
① 複合コンポーネントに入ります。
作業領域（緑）で右クリック－［コンポーネント追加］－［画面構成部品］－［ボタン］－［ボタン］と、クリックします。(２回繰り返します)
作業領域（緑）で右クリック－［コンポーネント追加］－［処理部品］－［演算制御］－［加算(+)］、
作業領域（緑）で右クリック－［コンポーネント追加］－［処理部品］－［条件制御］
－［比較演算(>)］とクリックします。

追加した２つのボタンは以下で［＋１０％］ボタン、［－１０％］ボタンになります。

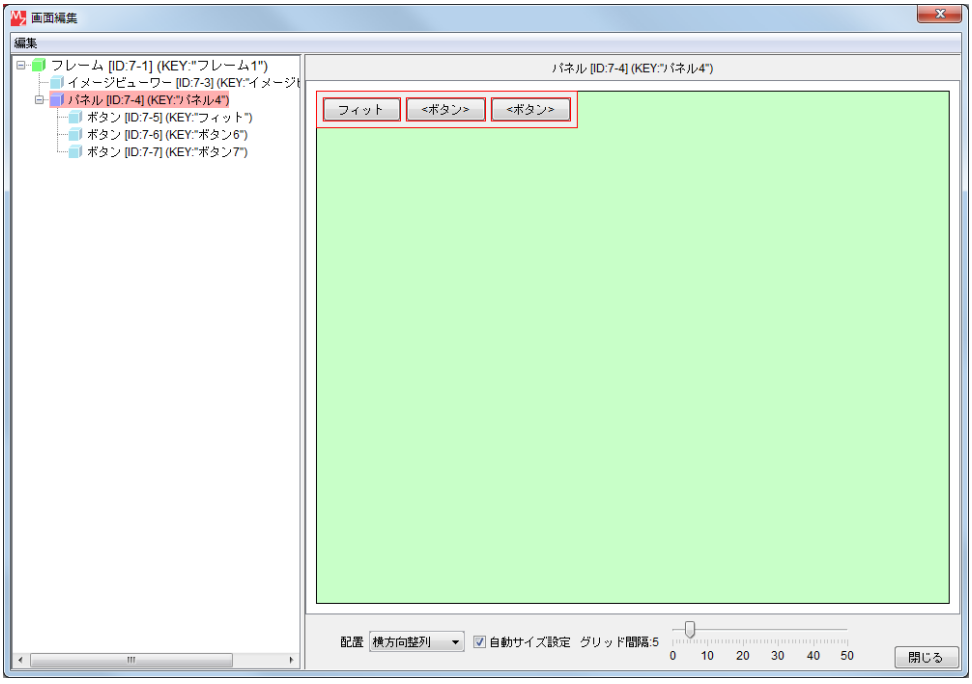
画面編集

① 画面を作成します。
画面編集をクリックします。

［パネル］コンポーネントに［ボタン］コンポーネントを２つ追加します。
［パネル(ID:7-5)］上で右クリック－［コンポーネント追加］－［ボタン(ID:7-6)］、
［パネル(ID:7-5)］上で右クリック－［コンポーネント追加］－［ボタン(ID:7-7)］、
とクリックします。



［フィット］ボタンの周りに２つボタンが追加されます。



③　追加後、閉じるをクリックします。

接続確認コンポーネント同士の接続を確認します。

初期状態を登録する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■イメージビューワー（ID:7-3）
発生イベント	データ設定イベント
接続先コンポーネント	■加算(+)（ID:7-8）
起動メソッド	数値に変換後、左右オペランドを設定する(String,String)
＜引数０＞	説明：左オペランド 取得方法：固定値 メソッド／値：100
＜引数１＞	説明：右オペランド 取得方法：固定値 メソッド／値：0
イベント番号	1

フィット後の状態を登録する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（ID:7-5）
発生イベント	アクションイベント
接続先コンポーネント	■加算(+)（ID:7-8）
起動メソッド	数値に変換後、左右オペランドを設定する(String,String)
＜引数０＞	説明：左オペランド 取得方法：メソッド戻り値 コンポーネント：イメージビューワー(ID:7-3) メソッド／値：垂直方向のイメージ表示比率を取得する
＜引数１＞	説明：右オペランド 取得方法：固定値 メソッド／値：0

画像を 10%ずつ大きくする

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（ID:7-6）
発生イベント	アクションイベント
接続先コンポーネント	■加算(+)（ID:7-8）
起動メソッド	数値変換／左右オペランド設定後、演算を行う (String,String)
＜引数０＞	説明：左オペランド 取得方法：メソッド戻り値 コンポーネント：加算(+)(ID:7-8) メソッド／値：演算結果（左オペランド＋ 右オペランド）を取得する
＜引数１＞	説明：右オペランド 取得方法：固定値 メソッド／値：10

画像を 10%ずつ小さくする

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（ID:7-7）
発生イベント	アクションイベント
接続先コンポーネント	■加算(+)（ID:7-8）
起動メソッド	数値変換／左右オペランド設定後、演算を行う (String,String)
＜引数０＞	説明：左オペランド 取得方法：メソッド戻り値 コンポーネント：加算(+)(ID:7-8) メソッド／値：演算結果（左オペランド＋ 右オペランド）を取得する
＜引数１＞	説明：右オペランド 取得方法：固定値 メソッド／値：-10

左右オペランドを計算する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■加算(+)（ID:7-8）
発生イベント	処理完了イベント
接続先コンポーネント	■比較演算(>)（ID:7-9）
起動メソッド	数値変換／左右オペランド設定後、演算を行う (String,String)
＜引数０＞	説明：左オペランド 取得方法：イベント内包 メソッド／値：処理結果データ
＜引数１＞	説明：右オペランド 取得方法：固定値 メソッド／値：0
イベント番号	０

画像スケールを設定する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■比較演算(>)（ID:7-9）
発生イベント	処理完了イベント
接続先コンポーネント	■イメージビューワー（ID:7-3）
起動メソッド	setScale（int）
＜引数＞	説明：スケール(%) 取得方法：メソッド戻り値 コンポーネント：比較演算(>)(ID:7-9) メソッド／値：左オペランドを取得する
イベント番号	１

操 作

［＋１０％］ボタン、［－１０％］の機能を設定しましょう。

―――初期状態を登録する―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［イメージビューワー(ID:7-3)］コンポーネント上で
右クリック－［イベント処理追加］－［データ設定イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［イメージビューワー(ID:7-3)］コンポーネントの［データ設定イベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－

［加算(+)（ID:7-8)］コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値変換後、左右オペランドを設定する(String,String)］をクリックします。
引数を設定します。
＜引数０＞
説明：左オペランド
取得方法：固定値
メソッド／値：100
＜引数１＞
説明：右オペランド
取得方法：固定値
メソッド／値：0
設定後、了解ボタンをクリックします。

④ イベント番号を設定します。
［加算(+)(ID:7-8)］コンポーネントの上で右クリックー［イベント番号設定］
ー［イベント番号設定］をクリックします。
定常起動のチェックをオフにして［NO:1］をチェックし設定をクリックします。

―――フィット後の状態を登録する―――
① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ボタン(ID:7-5)］コンポーネント上で
右クリックー［イベント処理追加］ー［アクションイベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［ボタン(ID:7-5)］コンポーネントの［アクションイベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［加算(+)（ID:7-8)］コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値変換後、左右オペランドを設定する(String,String)］をクリックします。
引数を設定します。
＜引数０＞
説明：左オペランド
取得方法：メソッド戻り値

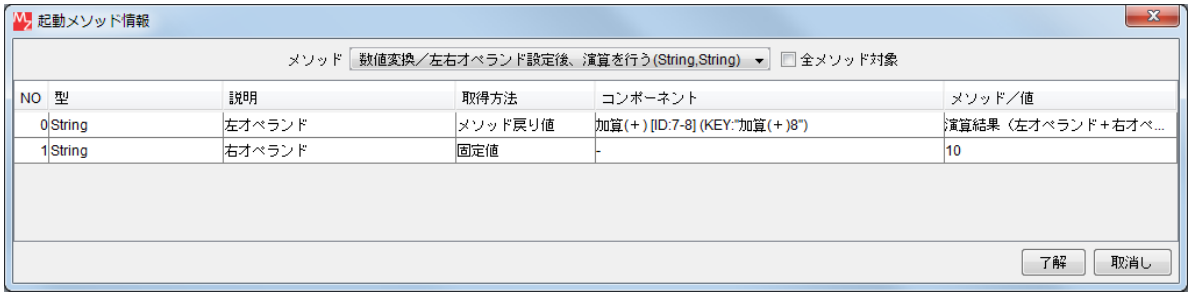
コンポーネント：イメージビューワー(ID:7-3)
メソッド／値：垂直方向のイメージ表示比率を取得する()
＜引数１＞
説明：右オペランド
取得方法：固定値
メソッド／値：0
設定後、了解ボタンをクリックします。

―――画像を１０％ずつ大きくする―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ボタン(ID:7-6)］コンポーネント上で
右クリックー［イベント処理追加］ー［アクションイベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［ボタン(ID:7-6)］コンポーネントの［アクションイベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［加算(+)（ID:7-8)］コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値変換／左右オペランド設定後、演算を行う(String,String)］をクリックします。
引数を設定します。
＜引数０＞
説明：左オペランド
取得方法：メソッド戻り値
コンポーネント：加算(+)（ID:7-8)
メソッド／値：演算結果（左オペランド＋右オペランド）を取得する()
＜引数１＞
説明：右オペランド
取得方法：固定値
メソッド／値：10
設定後、了解ボタンをクリックします。

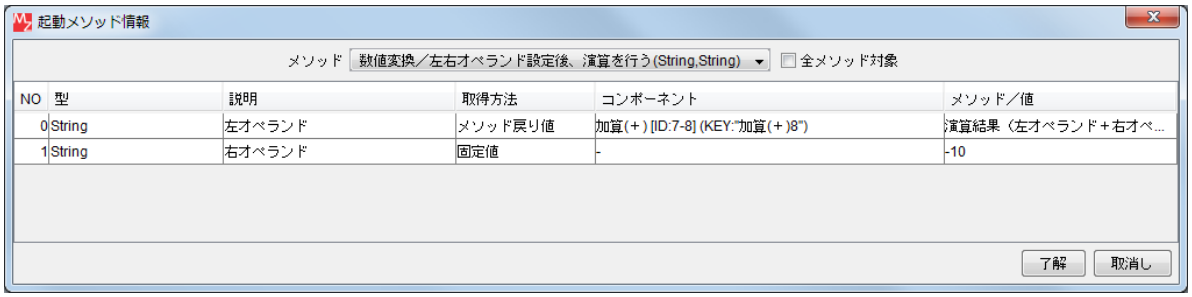


―――画像を１０％ずつ小さくする―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ボタン(ID:7-7)］コンポーネント上で
右クリック－［イベント処理追加］－［アクションイベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［ボタン(ID:7-7)］コンポーネントの［アクションイベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［加算(+) (ID:7-8)］コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値変換／左右オペランド設定後、演算を行う(String,String)］をクリックします。
引数を設定します。
＜引数０＞
説明：左オペランド
取得方法：メソッド戻り値
コンポーネント：加算(+) (ID:7-8)
メソッド／値：演算結果（左オペランド＋右オペランド）を取得する()
＜引数１＞
説明：右オペランド
取得方法：固定値
メソッド／値：-10
設定後、了解ボタンをクリックします。



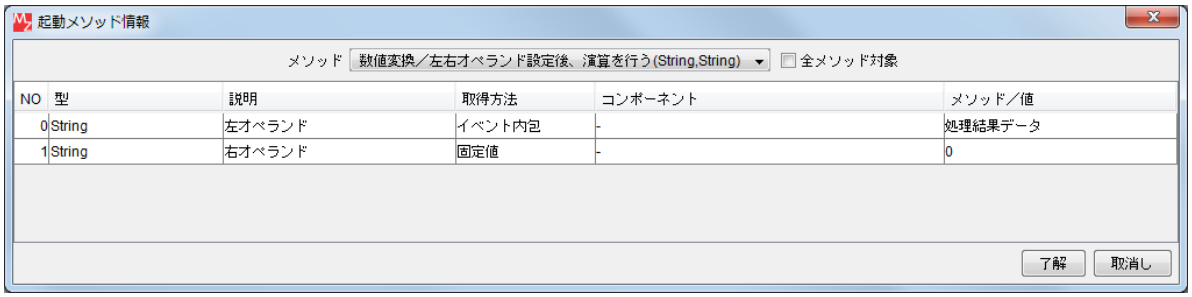
―――左右オペランドを計算する―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［加算(+) (ID:7-8)］コンポーネント上で
右クリック－［イベント処理追加］－［処理完了イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［加算(+) (ID:7-8)］コンポーネントの［処理完了イベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。

右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［比較演算(>)（ID:7-9)］コンポーネントをクリックします。

③　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値変換／左右オペランド設定後、演算を行う(String,String)］をクリックします。
引数を設定します。
＜引数０＞
説明：左オペランド
取得方法：イベント内包
メソッド／値：処理結果データ
＜引数１＞
説明：右オペランド
取得方法：固定値
メソッド／値：0
設定後、了解ボタンをクリックします。

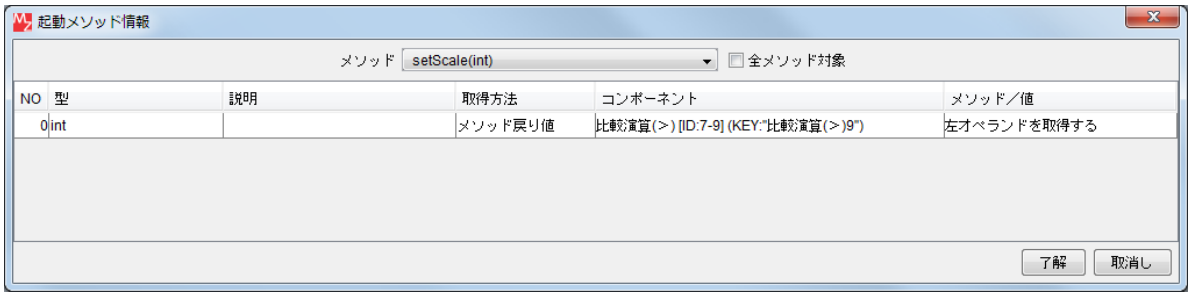


④　イベント番号を設定します。
［比較演算(>)（ID:7-9)］コンポーネントの上で右クリックー［イベント番号設定］
ー［イベント番号設定］をクリックします。
定常起動のチェックをオフにして［NO:0］をチェックし設定をクリックします。

―――比較結果を受け取る―――
①　使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［比較演算(>)（ID:7-9)］コンポーネント上で
右クリックー［イベント処理追加］ー［処理完了イベント］とクリックします。

②　イベントの接続先コンポーネントを選びます。
左側の［比較演算(>)（ID:7-9)］コンポーネントの［処理完了イベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［イメージビューワー(ID:7-3)］コンポーネントをクリックします。

③　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［setSale（int)］をクリックします。
引数を設定します。
説明：スケール(%)
取得方法：メソッド戻り値
コンポーネント：比較演算(>)（ID:7-9)
メソッド／値：左オペランドを取得する
設定後、了解ボタンをクリックします。

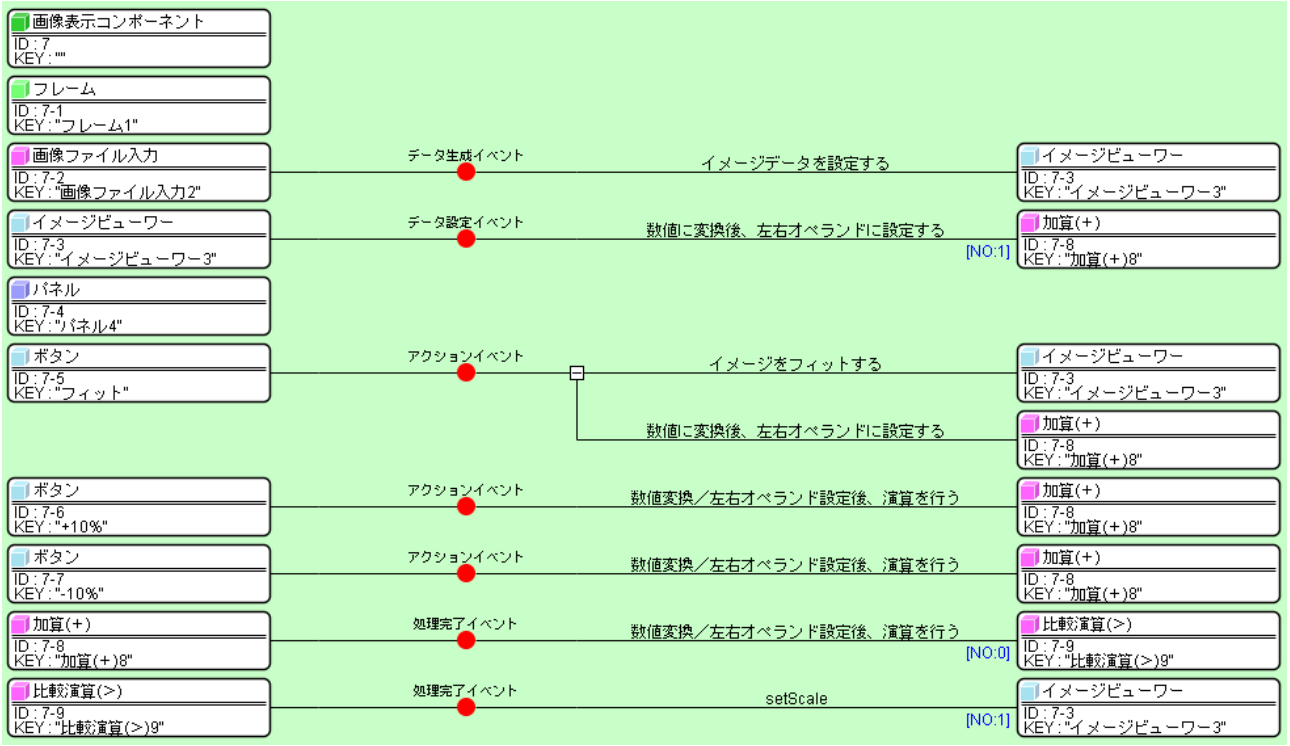


④　イベント番号を設定します。
［画像ファイル入力（ID:7-2)］コンポーネントの上で右クリックー［イベント番号設定］ー［イベント番号設定］をクリックします。
定常起動のチェックをオフにして［NO:1］をチェックし設定をクリックします。

⑤　［実行（設定可)］で実行し確認します。
さらにボタン名を「+10%」「-10%」にそれぞれ変更します。

まとめ

ここまで進めるとビルダー上では以下のようになります。



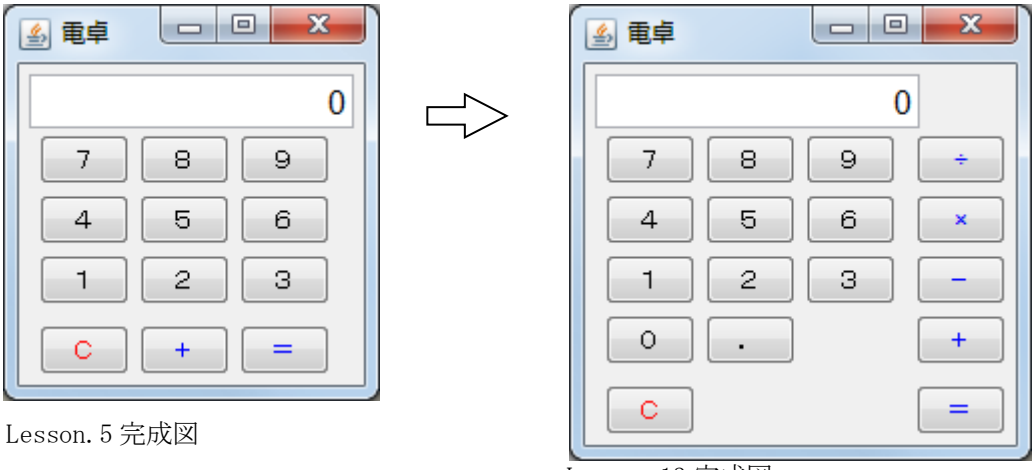
Lesson.13 電卓の機能を拡張してみよう

ここではLesson.5で作成した電卓の機能を次のように拡張していきます。

1．２桁以上の桁が扱える
2．四則演算できる

完成図

Lesson.5と比較して電卓の完成図を確認しましょう。



Lesson.5 完成図

Lesson.13 完成図

Step.1 任意桁の計算

１桁同士の計算ではなく、任意桁同士の計算ができるようにします。
通常の電卓では、任意桁の数値を入力するには数字ボタンを連続して押すと、後から押した数字が表示されている数字の右側に追加されて複数の桁になっていきます。その後、演算子ボタン（＋やーなど）やイコールボタンを押すと表示されている数値を使って計算します。

このような任意桁を利用するには以下のことが必要になります。

1．[数字]ボタンからの一連の入力を文字列の連結として処理する
2．連結された文字列を数値に変換する

これらの処理を可能にするために『変数』コンポーネントを利用します。

１）変数コンポーネント

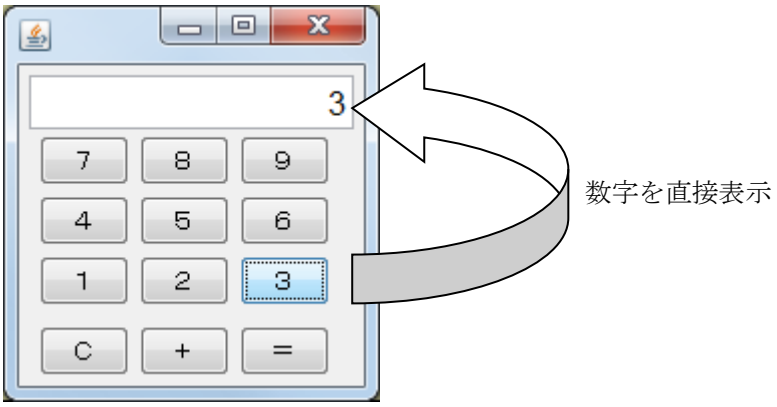
［変数］コンポーネントとは各種のデータを一時的に保持して（格納して）、保持したままそのデータに対する操作を行うコンポーネントの総称です。文字データが格納できる変数や、数値データが格納できる変数など多くの種類があります。

ここでは、①［数字］ボタンから連続して入力された数字を文字列として連結する処理を変数コンポーネントで行います。また②連結された文字列を数値に変換する処理も変数コンポーネントで行います。

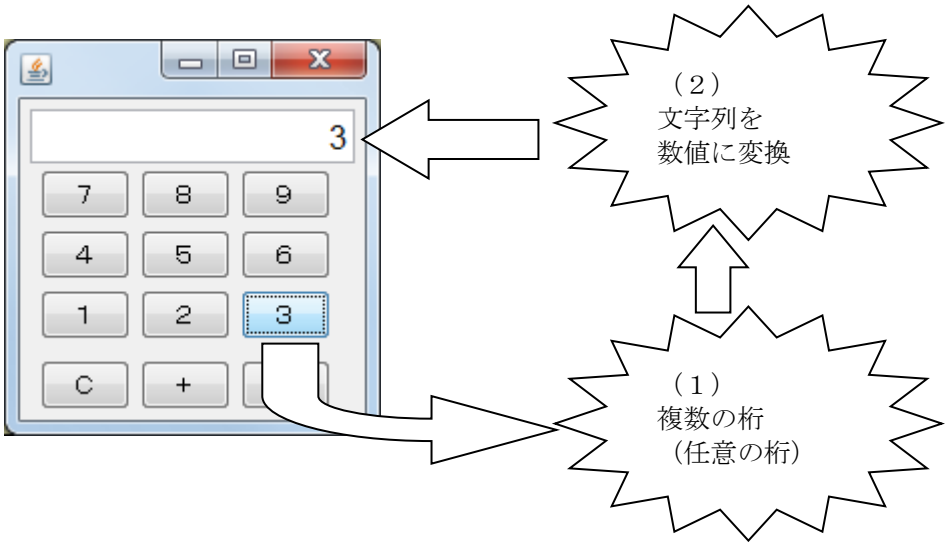
考え方

Lesson.5 と Lesson.13 の数値入力フィールドへの数値の表示の違いを確認します。

Lesson.5 では［数字］ボタンを押したら、直接数値が［数値入力フィールド］に表示されました。



Lesson.13 では［数字］ボタンが押されたら、変数コンポーネントを用いて数字を文字列として連結して（１）複数の桁にして、（２）文字列から数値に変換して、その後、数値入力フィールドに表示します。



準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■文字列格納変数	1	［処理部品］－［変数］－［文字列格納変数］
■ 任 意 精 度 実 数 (BigDecimal)格納変数	1	［処理部品］－［変数］－ ［任意精度実数(BigDecimal)格納変数］

操 作

変数コンポーネントを追加し、変数コンポーネントの名称を変更します。

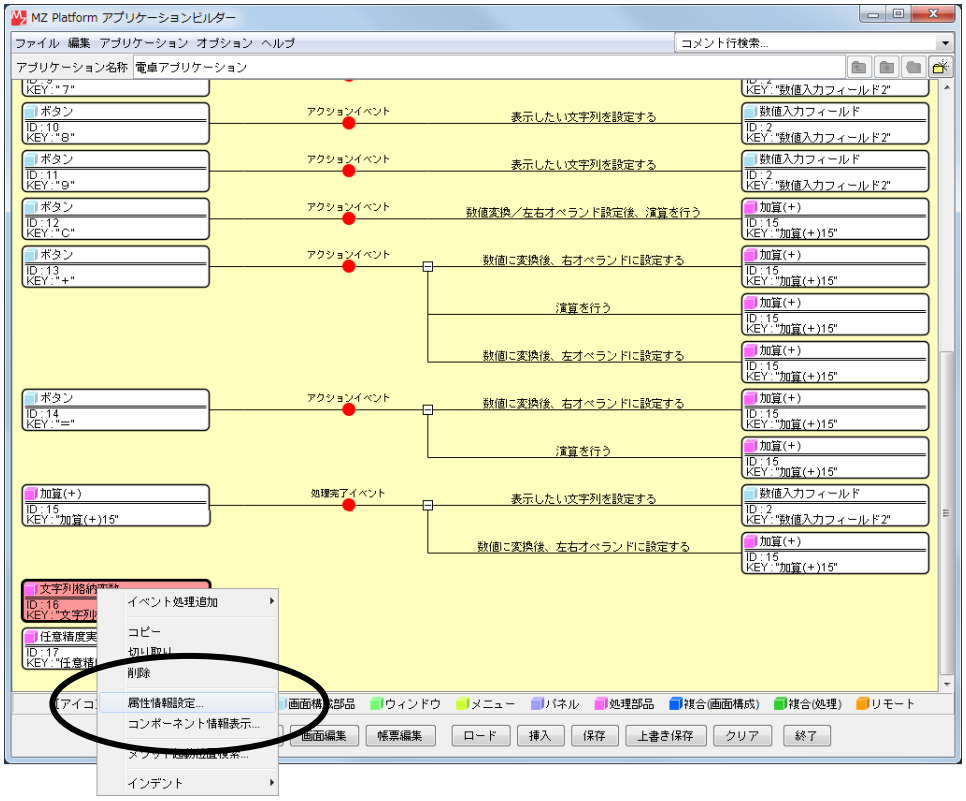
（完成している Lesson05(電卓).mzax の続きに追加していきます）

① 必要なコンポーネントを追加します。

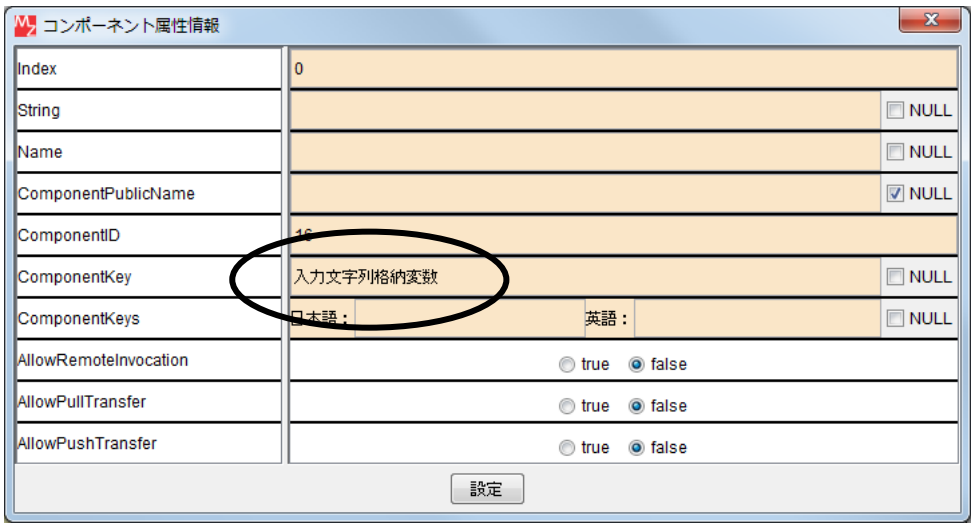
ここでは［文字列格納変数］コンポーネントと［任意精度実数(BigDecimal)格納変数］コンポーネン

トを追加します。
作業領域で右クリックー［コンポーネント追加］ー［処理部品］ー［変数］ー［文字列格納変数］、
作業領域で右クリックー［コンポーネント追加］ー［処理部品］ー［変数］
ー［任意精度実数(BigDecimal)格納変数］とクリックします。

② わかりやすいように変数コンポーネントの名前を変更します（名前の変更はしなくても構いません）。
追加した［文字列格納変数(ID:16)］の上で右クリックー［属性情報設定...］をクリックします。



③ ComponentKey を「入力文字列格納変数」に変更します。
入力後［設定］をクリックします。



④ ②～③を繰り返して［任意精度実数(BigDecimal)格納変数］も「内部数値格納変数」に変更します。

２）起動時の変数の初期化

電卓アプリケーションを起動したときに「文字列格納変数（Key:入力文字列格納変数）」に 0 を設定して初期化しておきます。

接続確認

コンポーネント同士の接続を確認します。

開始

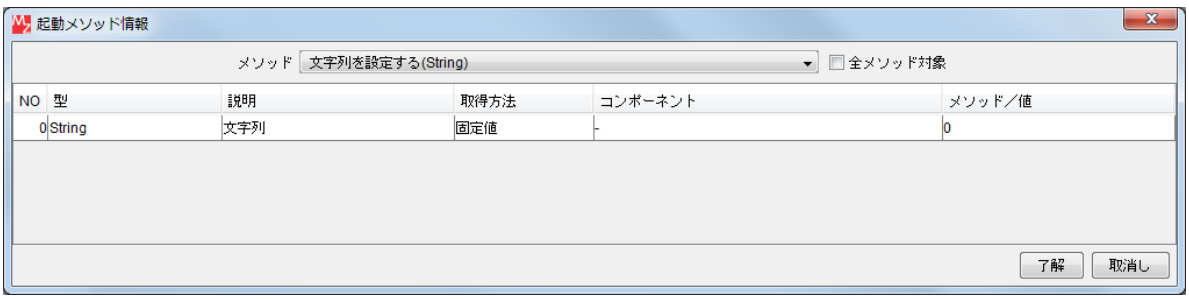
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■アプリケーション
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■文字列格納変数（ID：16, Key:入力文字列格納変数）
起動メソッド	文字列を設定する(String)
＜引数＞	説明：文字列 取得方法：固定値 メソッド／値：0

操 作

アプリケーション開始時に［入力文字列格納変数］を 0 に初期化します。

① イベントの接続先コンポーネントを選びます。
左側の［アプリケーション］コンポーネントの［アプリケーション開始イベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［文字列格納変数（ID:16 Key:入力文字列格納変数)］をクリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［文字列を設定する(String)］をクリックします。
引数を設定します。
説明：設定する文字列
取得方法：固定値
メソッド／値：0
設定後、了解ボタンをクリックします。



３）数字ボタンの起動メソッドの変更

Lesson.5 では［数字］ボタンを押すと数字が直接［数値入力フィールド］に表示されていましたが、［数字］ボタンを押すと［文字列格納変数（Key:入力文字列格納変数)］に格納するように変更します。

接続確認 コンポーネント同士の接続を確認します。

数字ボタンを押したら文字列格納変数に格納し、次に押された数字と連結する

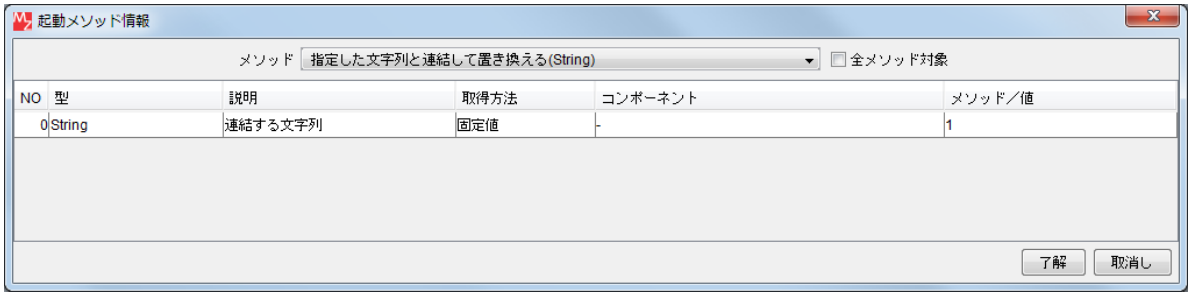
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン(1, ID:3)
発生イベント	アクションイベント
接続先コンポーネント	■文字列格納変数（ID：16, Key:入力文字列格納変数）
起動メソッド	指定した文字列と連結して置き換える(String)
＜引数＞	説明：連結する文字列 取得方法：固定値 メソッド／値：1

操 作 ［数字］ボタンの起動メソッドを変更します。

数字ボタンを押したら文字列格納変数に格納し、その変数内で、次に押された数字と連結していきます。

① イベントの接続先コンポーネントを変更します。
左側の［ボタン(1, ID:3)］コンポーネントから接続されている
［数値入力フィールド(ID:2)］メソッド上で右クリックー［接続コンポーネント選択］ー
［文字列格納変数（ID:16, Key:入力文字列格納変数］
とクリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［指定した文字列と連結して置き換える(String)］をクリックします。
引数を設定します。
説明：連結する文字列
取得方法：固定値
メソッド／値：1
設定後、了解ボタンをクリックします。



③　①～②の操作を繰り返して数字ボタン２～９まで設定をします。

４）文字列データを数値データに変換

［文字列格納変数（Key:入力文字列格納変数)］に格納してあるデータを数値データに変更するため［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］にセットします。

接続確認 コンポーネント同士の接続を確認します。

文字列格納変数に格納してあるデータを内部数値格納変数に格納する（数値に変換される）

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■文字列格納変数（ID：16, Key:入力文字列格納変数）
発生イベント	データ設定イベント
接続先コンポーネント	■任意精度実数(BigDecimal)格納変数（ID：17, Key:内部数値格納変数）
起動メソッド	数値を文字列で設定する(String)
＜引数＞	説明：数値の文字列表現 取得方法：イベント内包 メソッド／値：イベント対象データ

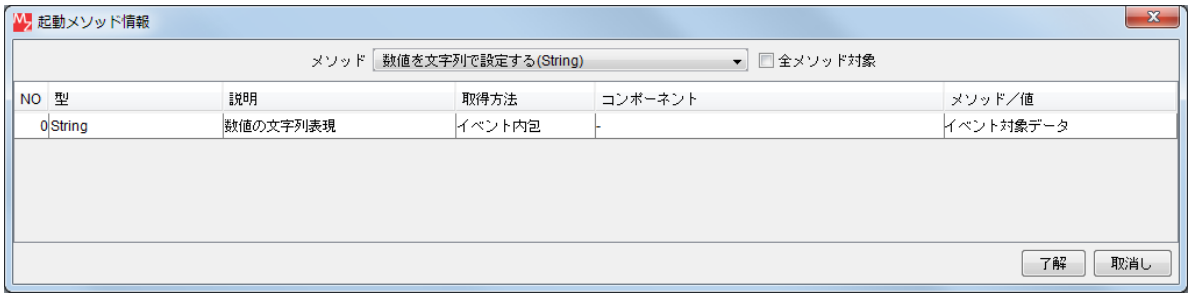
操 作 ［文字列格納変数（Key:入力文字列格納変数)］に格納してあるデータを［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］にセットします。

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［文字列格納変数（Key:入力文字列格納変数）(ID:16)］コンポーネント上で右クリックー［イベント処理追加］ー［データ設定イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［文字列格納変数（Key:入力文字列格納変数）(ID:16)］コンポーネントの［データ設定イベント］上で右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー［内部数値格納変数(ID:17)］をクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値を文字列で設定する(String)］をクリックします。
引数を設定します。
説明：数値の文字列表現
取得方法：イベント内包
メソッド／値：イベント対象データ

設定後、了解ボタンをクリックします。



５）数値データを［数値入力フィールド］に表示

［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］にセットされているデータを［数値入力フィールド］に表示します。

接続確認 コンポーネント同士の接続を確認します。

内部数値格納変数に値がセットされたら数値入力フィールドに表示する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■任意精度実数(BigDecimal)格納変数（ID：17, Key:内部数値格納変数）
発生イベント	データ設定イベント
接続先コンポーネント	■数値入力フィールド（ID:2）
起動メソッド	表示したい文字列を設定する(String)
＜引数＞	説明：文字列 取得方法：イベント内包 メソッド／値：イベント対象データ

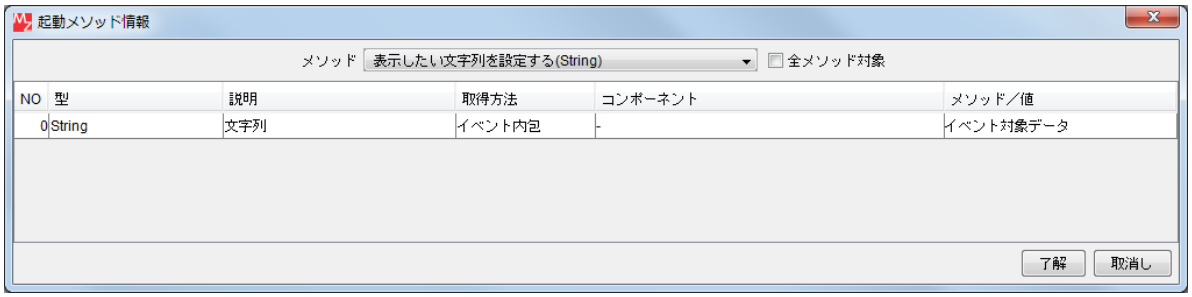
操 作 ［内部数値格納変数］にセットされているデータを［数値入力フィールド］に表示します。

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［任意精度実数(BigDecimal)格納変数（ID:17, Key:内部数値格納変数)］コンポーネント上で右クリックー［イベント処理追加］ー［データ設定イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［任意精度実数(BigDecimal)格納変数（ID:17, Key:内部数値格納変数)］コンポーネントの［データ設定イベント］上で右クリックー［起動メソッド追加］とクリックします。
薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー［数値入力フィールド(ID:2)］をクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［表示したい文字列を設定する(String)］をクリックします。
引数を設定します。
説明：文字列
取得方法：イベント内包
メソッド／値：イベント対象データ

設定後、了解ボタンをクリックします。



６）［加算］コンポーネントの処理の変更

Lesson.5 では加算コンポーネントの処理結果が直接［数値入力フィールド］に設定されていましたが、［文字列格納変数（Key:入力文字列格納変数)］と［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］を追加したので以下のように修正を行います。

１．演算を行った後に［文字列格納変数（Key:入力文字列格納変数)］の文字列を「０」に初期化する

２．［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］の数値を処理結果に置き換える

接続確認 コンポーネント同士の接続を確認します。

［内部数値格納変数］の数値を処理結果に置き換える❶

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■加算(+)（ID:15）
発生イベント	処理完了イベント
接続先コンポーネント❶ ※Lesson.5 で設定した［接続先：数値入力フィールド］を置き換える	■任意精度実数(BigDecimal)格納変数（ID：17, Key:内部数値格納変数）
起動メソッド	数値を文字列で設定する(String)
＜引数＞	説明：数値の文字列表現 取得方法：イベント内包 メソッド／値：処理結果データ

演算を行った後に［入力文字列格納変数］の文字列を「０」に初期化する❷

接続先コンポーネント❷ ※新しく追加する	■文字列格納変数（ID：16, Key:入力文字列格納変数）
起動メソッド	文字列を設定する（イベント発生なし）(String)
＜引数＞	説明：なし 取得方法：固定値 メソッド／値：0

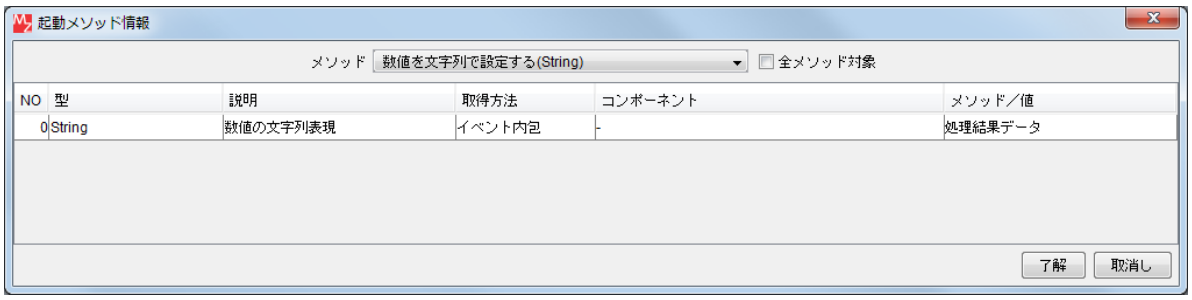
操 作 ［加算(ID:15)］コンポーネントの接続を変更します。

――［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］の数値を処理結果に置き換える❶――

① イベントの接続先コンポーネントを変更します。
左側の［加算(ID:15)]］コンポーネントから接続されている右側［数値入力フィールド(ID:2)］上で右クリック－［接続コンポーネント選択］－［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数）(ID:17)］を選択します。

② 接続したコンポーネントの処理を選びます。

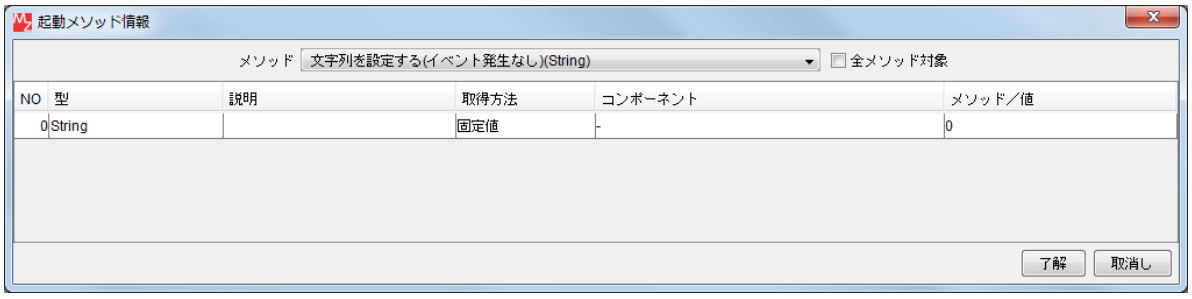
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値を文字列で設定する(String)］をクリックします。
引数を設定します。
　説明：数値の文字列表現
　取得方法：イベント内包
　メソッド／値：処理結果データ
設定後、了解ボタンをクリックします。



―――演算を行った後に［入力文字列格納変数］の文字列を「０」に初期化する❷―――

③　イベントの接続先コンポーネントを選びます。
左側の［加算(ID:15)］コンポーネントの［処理完了イベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［文字列格納変数（ID:16 Key:入力文字列格納変数)］をクリックします。

④　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［文字列を設定する（イベント発生なし）(String)］をクリックします。
引数を設定します。
　説明：なし
　取得方法：固定値
　メソッド／値：0
設定後、了解ボタンをクリックします。



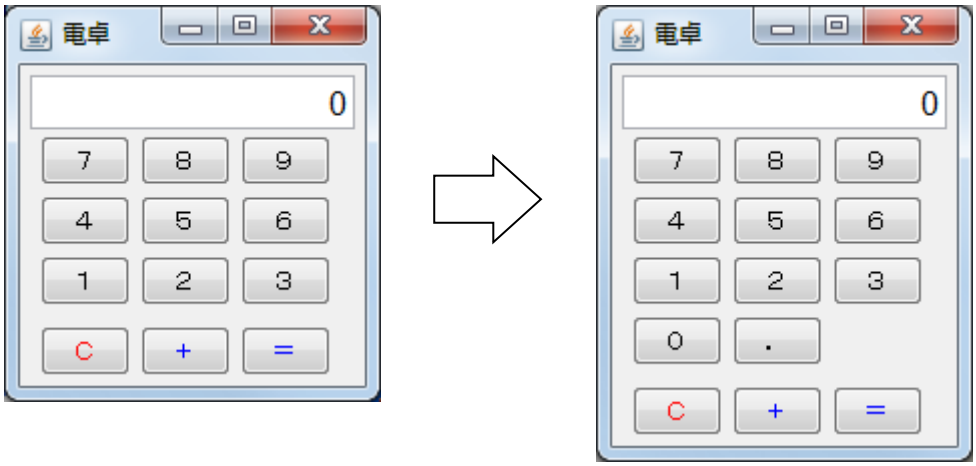
Step.2 ［０］ボタンと［.（小数点）］ボタンを追加

任意桁の表示が可能になったので［０］ボタンと［.（小数点）］ボタンを追加しましょう。

１）［０］ボタンと［.（小数点）］ボタンの追加

完成図

［０］ボタンと［.（小数点）］ボタンを追加します。



準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■ボタン	2	［画面構成部品］－［ボタン］－［ボタン］

操 作

ボタンを追加しましょう。

① 必要なコンポーネントを追加します。
ここでは［ボタン］コンポーネントを２つ追加します。
作業領域で右クリック－［コンポーネント追加］－［画面構成部品］－［ボタン］－［ボタン］とクリックします。（２回繰り返します）

画面編集

① 画面を作成します。
画面編集をクリックします。
［ボタン］コンポーネントをフレームに追加します。
［画面編集］画面上で右クリック－［ボタン］コンポーネントとクリックします。（２回繰り返します）
追加できたら、ボタンの名前を変更し、体裁を整え閉じるをクリックします。

接続確認

コンポーネント同士の接続を確認します。

［0］ボタン、[.]ボタンの設定

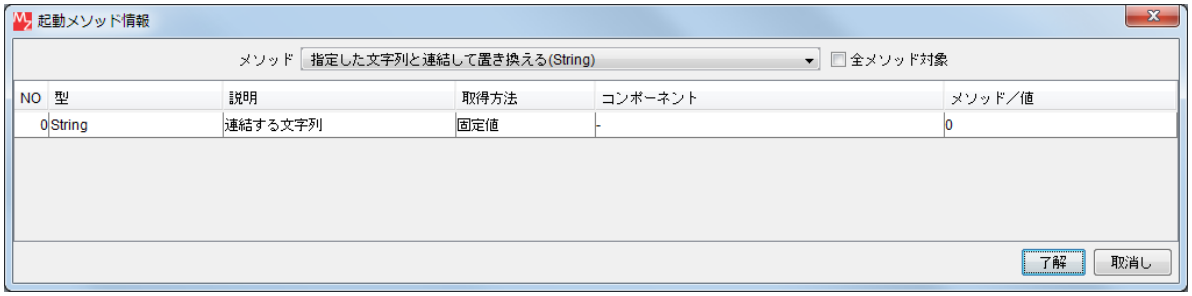
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（0, ID:18）
発生イベント	アクションイベント
接続先コンポーネント	■文字列格納変数（ID：16, Key:入力文字列格納変数）
起動メソッド	指定した文字列と連結して置き換える(String)
＜引数＞	説明：連結する文字列 取得方法：固定値 メソッド／値：0

操 作

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ボタン(ID:18)］コンポーネント上で右クリックー［イベント処理追加］
ー［アクションイベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［ボタン(ID:18)］コンポーネントの［アクションイベント］上で
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［文字列格納変数（ID:16 Key:入力文字列格納変数)］コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の をクリックします。
［指定した文字列と連結して置き換える(String)］をクリックします。
引数を設定します。
説明：連結する文字列
取得方法：固定値
メソッド／値：0
設定後、了解ボタンをクリックします。



④　①～③の操作を繰り返して（または起動メソッドのコピーを利用して）[.]ボタンを設定します。
固定値の変更をします。(ボタンの名前に合った固定値を入力します)

２）エラー処理

［.（小数点)］ボタンが追加されたことで、この時点でアプリケーションはボタンの組み合わせでエラーが発生するようになります。

［.（小数点)］ボタンを２回押すとエラーになり、エラーダイアログが表示されます。文字列の数値への変換が失敗するためです。この状態になった場合、そのまま続けて数字を入力しても［文字列格納変数（Key:入力文字列格納変数)］に保持されているデータは数値に変換できない状態になっているので、エラーが繰り返し表示されます。このような場合には、［Ｃボタン］を押して文字列をクリアすることでその状態を抜けることができます。

このようにエラーが発生した場合の処理をビルダー上で記述しておくことができます。エラー処理の方法としては、［文字列格納変数（Key:入力文字列格納変数)］から最後に入力された文字（２回目の小数点）を１つ取り除く必要があります。(例：　0.2345.　→　0.2345)

エラーが発生するのは文字列を数値に変換するときなのでそこにエラー処理を記述しておきます。

［任意精度整数(BigInteger)格納変数］を追加して、エラーが発生したら［文字列格納変数（Key:入力文字列格納変数)］の最後尾の文字を１文字削除するようにします。

準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■整数(BigInteger)格納変数	1	［処理部品］－［変数］－ ［任意精度整数(BigInteger)格納変数］

操 作

①　必要なコンポーネントを追加します。
作業領域で右クリック－［コンポーネント追加］－［処理部品］－［変数］
－［任意精度整数(BigInteger)格納変数］とクリックします。

接続確認

コンポーネント同士の接続を確認します。

エラーが発生したときの文字列の長さを取得する❶

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■文字列格納変数（ID：16, Key:入力文字列格納変数）
発生イベント	データ設定イベント
接続先コンポーネント❶	■整数(BigInteger)格納変数（ID：20）
起動メソッド	数値(BigInteger)を設定する(BigInteger)
＜引数＞	説明：数値 取得方法：メソッド戻り値 コンポーネント： 文字列格納変数（Key:入力文字列格納変数） メソッド／値：文字列の長さを取得する
起動モード	エラー発生時起動

エラーが発生したときに文字列の最後尾の１文字を削除する❷

接続先コンポーネント❷	■整数(BigInteger)格納変数（ID：20）
起動メソッド	値を１減らす()
起動モード	エラー発生時起動

一文字削除した文字列に置き換える❸

接続先コンポーネント❸	■文字列格納変数（ID：16, Key:入力文字列格納変数）
起動メソッド	指定インデックス間の部分文字列に置き換える(int, int)
＜引数０＞	説明：部分文字列の開始インデックス 取得方法：固定値 メソッド／値：0
＜引数１＞	説明：部分文字列の終了インデックス 取得方法：メソッド戻り値 コンポーネント：整数(BigInteger)格納変数 メソッド／値：int 数値(BigInteger)を取得する
起動モード	エラー発生時起動

操 作

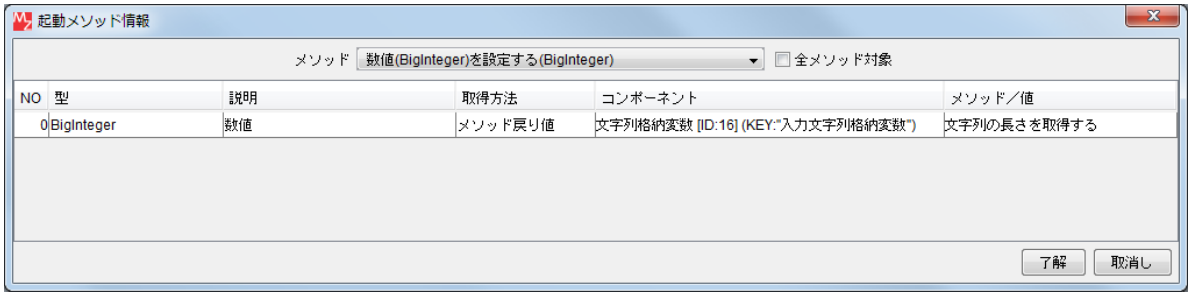
エラー処理を追加しましょう。

―――エラーが発生したときの文字列の長さを取得する❶―――

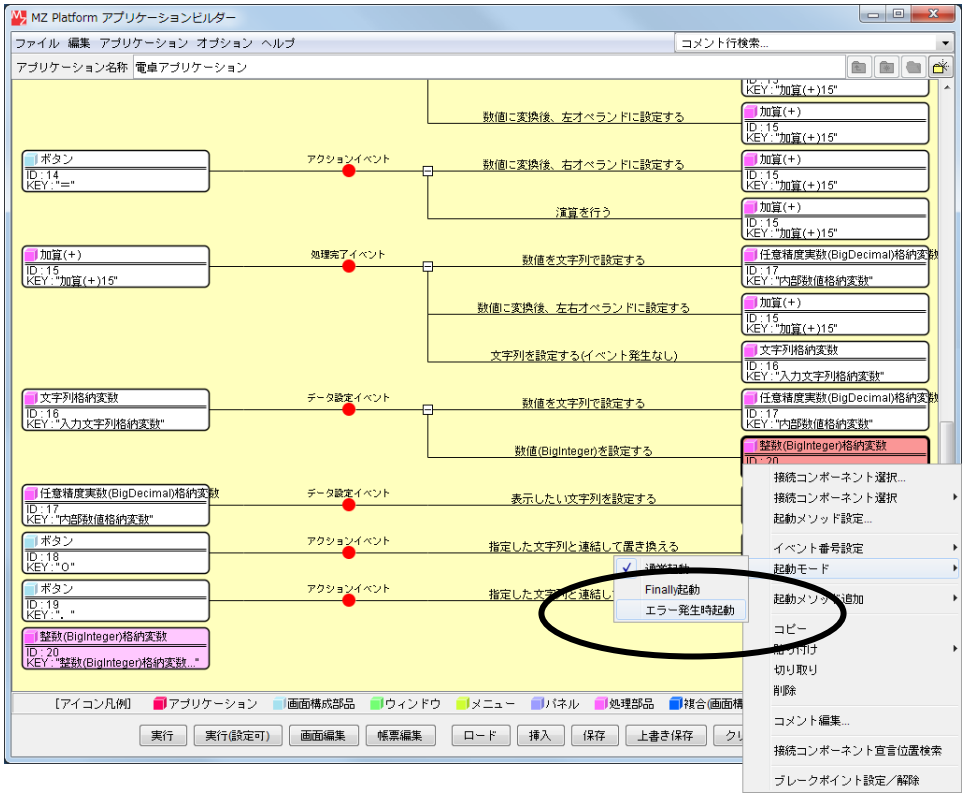
① イベントの接続先コンポーネントを選びます。
左側の［文字列格納変数（ID:16 Key:入力文字列格納変数)］コンポーネントの［データ設定イベント］

上で右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［整数(BigInteger)格納変数(ID:20)］コンポーネントをクリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値(BigInteger)を設定する(BigInteger)］をクリックします。
引数を設定します。
説明：数値
取得方法：メソッド戻り値
コンポーネント：文字列格納変数（ID:16 Key:入力文字列格納変数）
メソッド／値：文字列の長さを取得する
設定後、了解ボタンをクリックします。



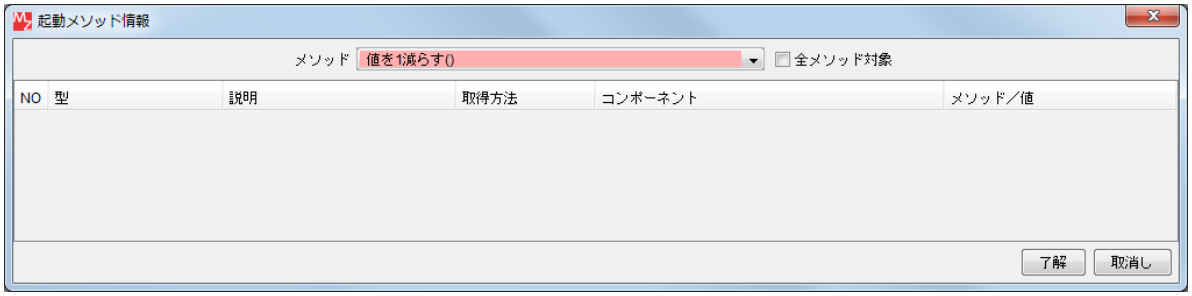
③ エラーの時だけ起動したいので［起動モード］の［エラー発生時起動］を選択します。



――エラーが発生したときに文字列の最後尾の１文字を削除する❷―――

④　イベントの接続先コンポーネントを選びます。
左側の［文字列格納変数（ID:16 Key:入力文字列格納変数)］コンポーネントの［データ設定イベント］上で右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－［整数(BigInteger)格納変数(ID:20)］コンポーネントをクリックします。

⑤　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［値を１減らす()］をクリックします。
設定後、了解ボタンをクリックします。



⑥　エラーの時だけ起動したいので［起動モード］の［エラー発生時起動］を選択しておきます。

―――一文字削除した文字列に置き換える❸―――

⑦　イベントの接続先コンポーネントを選びます。
左側の［文字列格納変数（ID:16 Key:入力文字列格納変数)］コンポーネントの［データ設定イベント］上で右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－［文字列格納変数（ID:16 Key:入力文字列格納変数)］コンポーネントをクリックします。

⑧　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［指定インデックス間の部分文字列に置き換える(int, int)］をクリックします。
引数を設定します。
引数０を設定します
　　説明：部分文字列の開始インデックス
　　取得方法：固定値
　　メソッド／値：0
引数１を設定します

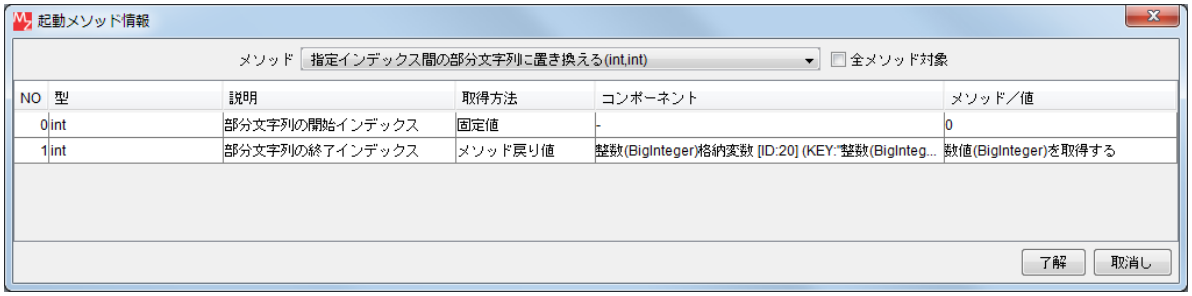
説明：部分文字列の終了インデックス

取得方法：メソッド戻り値

コンポーネント：整数(BigInteger)格納変数(ID:20)

メソッド／値：数値（BigInteger）を取得する

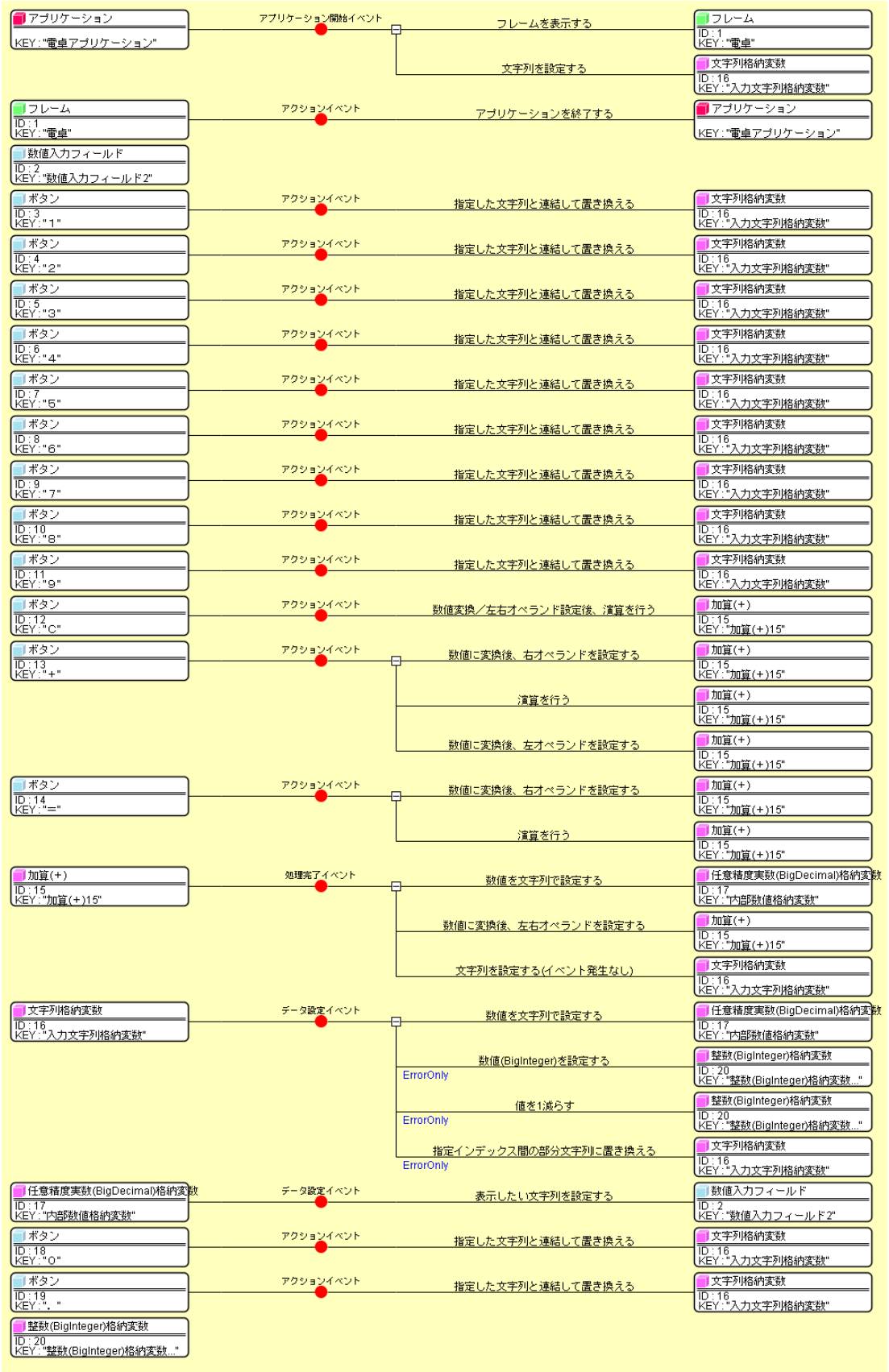
設定後、了解ボタンをクリックします。



⑨　エラーの時だけ起動したいので［起動モード］を［エラー発生時起動］を選択しておきます。

まとめ

ここまで進めるとビルダー上では以下のようになります。



Step.3 四則演算

これまでは［＋（加算)］のみでしたので四則演算ができるように設定しましょう。

Lesson.5 では演算として［＋ボタン］のみが用意されていましたが、ここでは四則演算を実現することを目的とします。通常の電卓では、演算子ボタン（[＋ボタン］など）や［＝ボタン］を押した際に、直前に押された演算子ボタンがあればそれに対応する演算処理を実行します。これを実現するためには、直前に押された演算子ボタンの情報をどこかに記憶しておき、最後に演算子ボタンや［＝ボタン］が押された際に前もって記憶しておいた演算処理を実行することができなくてはいけません。このような処理の実現方法はいくつか考えられますが、ここでは［算術演算子コンポーネント格納変数］を用いたアプリケーションを作成します。

１）変数の利用

ここでは２つの変数を使用します。

・[算術演算子コンポーネント格納変数（入力演算子格納変数)]
・[算術演算子コンポーネント格納変数（内部演算子格納変数)]

この変数には直前に押された［演算子ボタン］に対応する［演算子］コンポーネントを格納しておきます。その後、次の［演算子ボタン］または［＝ボタン］が押された際に、変数に格納された［演算子］コンポーネントが演算を実行し、また次の［演算子］コンポーネントを格納します。
これを繰り返すことで任意回数の四則演算を実現します。

考え方

1．［算術演算子コンポーネント格納変数（入力演算子格納変数)］は、演算子が入力される度に[内部演算子格納変数]に記憶されている演算を実行してから、［内部演算子格納変数］にその［演算子］コンポーネントを設定します。

2．［算術演算子コンポーネント格納変数（内部演算子格納変数)］は、［演算子］コンポーネントが格納されるたびにその左側の数値を［**任意精度実数(BigDecimal)格納変数**（Key:内部数値格納変数)］から取得して設定します。

準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■算術演算子コンポーネント格納変数	2	［処理部品］－［変数］－［算術演算子コンポーネント格納変数］
■サブルーチン	2	［処理部品］－［サブルーチン］－［サブルーチン］
■Null 判定	1	［処理部品］－［条件制御］－［Null 判定］

操 作

演算子を格納する変数を用意しましょう。

① 必要なコンポーネントを追加します。
ここでは四則演算で使用する５つのコンポーネントを追加します。
作業領域で右クリックー［コンポーネント追加］ー［処理部品］ー［変数］
ー［算術演算子コンポーネント格納変数］とクリックします（２回繰り返します)。
作業領域で右クリックー［コンポーネント追加］ー［処理部品］ー［サブルーチン］
ー［サブルーチン］とクリックします（２回繰り返します)。
作業領域で右クリックー［コンポーネント追加］ー［処理部品］ー［条件制御］ー［Null 判定］
とクリックします。

② コンポーネントの名前を変更しておきます。
１つめの［算術演算子コンポーネント格納変数］を［入力演算子格納変数］
２つめの［算術演算子コンポーネント格納変数］を［内部演算子格納変数］
１つめの［サブルーチン］を［演算実行サブルーチン］
２つめの［サブルーチン］を［初期化サブルーチン］
［Null 判定］を［内部演算子 Null 判定］
と変更します。

接続確認

コンポーネント同士の接続を確認します。

演算実行サブルーチンの処理を呼び出す❶

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■算術演算子コンポーネント格納変数（ID:21, Key:入力演算子格納変数）
発生イベント	データ設定イベント
接続先コンポーネント❶	■サブルーチン（ID:23, Key: 演算実行サブルーチン）
起動メソッド	処理を呼び出す()

入力（格納）された［演算子］を［内部演算子格納変数］に設定する❷

接続先コンポーネント❷	■算術演算子コンポーネント格納変数（ID:22, Key:内部演算子格納変数）
起動メソッド	演算子を設定する(PFArithmeticOperator)
＜引数＞	説明：算術演算子コンポーネント 取得方法：イベント内包 メソッド／値：イベント対象データ

演算子の左側の数値を数値変換する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■算術演算子コンポーネント格納変数（ID:22, Key:内部演算子格納変数）
発生イベント	データ設定イベント
接続先コンポーネント	■算術演算子コンポーネント格納変数（ID:22, Key:内部演算子格納変数）
起動メソッド	演算子の左側の数値を文字列で設定して数値変換する (String)
＜引数＞	説明：演算子の左側の数値の文字列表現 取得方法：メソッド戻り値 コンポーネント：任意精度実数(BigDecimal)格納変数 (ID:17) メソッド／値：数値(BigDecimal)を取得する

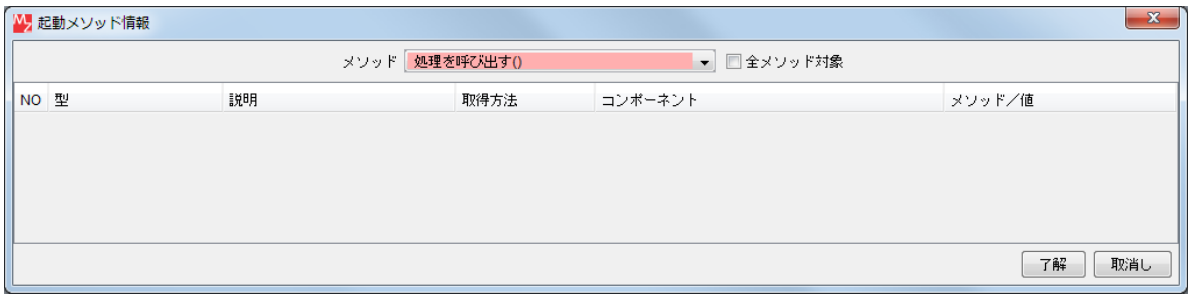
操 作

―――演算実行サブルーチンの処理を呼び出す❶―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［入力演算子格納変数(ID:21)］コンポーネント上で右クリックー［イベント処理追加］ー［データ設定イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［入力演算子格納変数(ID:21)］コンポーネントの［データ設定イベント］上で右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー［演算実行サブルーチン(ID:23)］コンポーネントをクリックします。

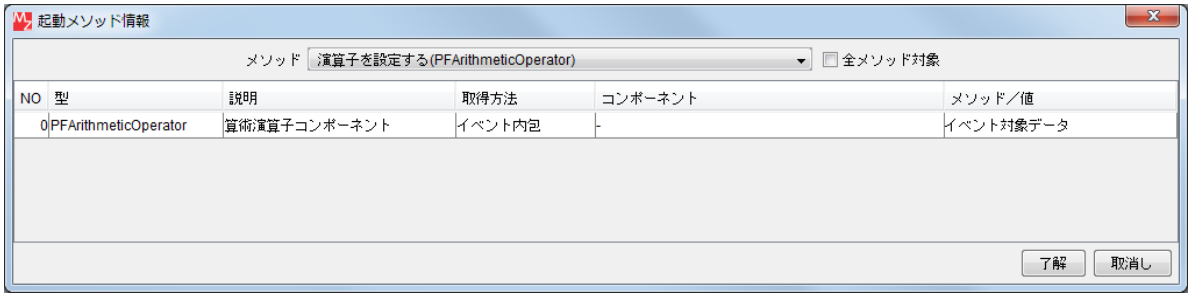
③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［処理を呼び出す()］をクリックします。
設定後、了解ボタンをクリックします。



―――入力（格納）された［演算子］を［内部演算子格納変数］に設定する❷―――

④　イベントの接続先コンポーネントを選びます。
左側の［入力演算子格納変数(ID:21)］コンポーネントの［データ設定イベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［内部演算子格納変数(ID:22)］コンポーネントをクリックします。

⑤　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［演算子を設定する(PFArithmeticOperator)］をクリックします。
引数を設定します。
　説明：算術演算子コンポーネント
　取得方法：イベント内包
　メソッド／値：イベント対象データ
設定後、了解ボタンをクリックします。



―――演算子の左側の数値を数値変換する―――

⑥　使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［内部演算子格納変数(ID:22)］コンポーネント上で右クリック－［イベント処理追加］
－［データ設定イベント］とクリックします。

⑦　イベントの接続先コンポーネントを選びます。
左側の［内部演算子格納変数(ID:22)］コンポーネントの［データ設定イベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［内部演算子格納変数(ID:22)］コンポーネントをクリックします。

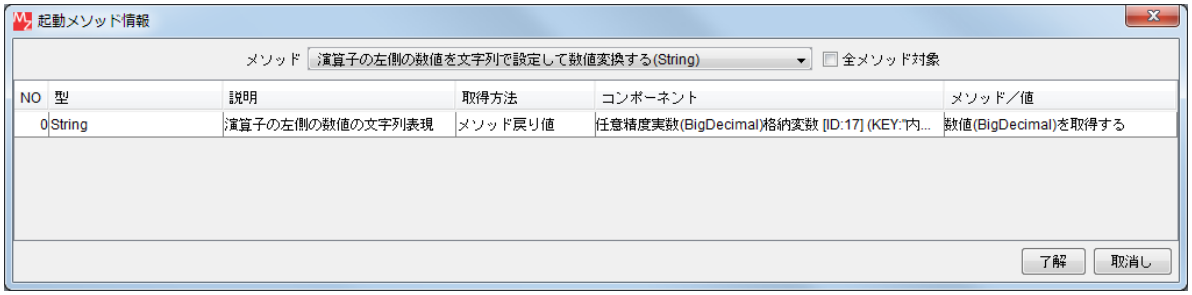
⑧　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［演算子の左側の数値を文字列で設定して数値変換する(String)］をクリックします。
引数を設定します。
　説明：演算子の左側の数値の文字列表現

取得方法：メソッド戻り値

コンポーネント：任意精度実数(BigDecimal)格納変数（ID:17 Key:内部数値格納変数）

メソッド／値：数値(BigDecimal)を取得する

設定後、了解ボタンをクリックします。

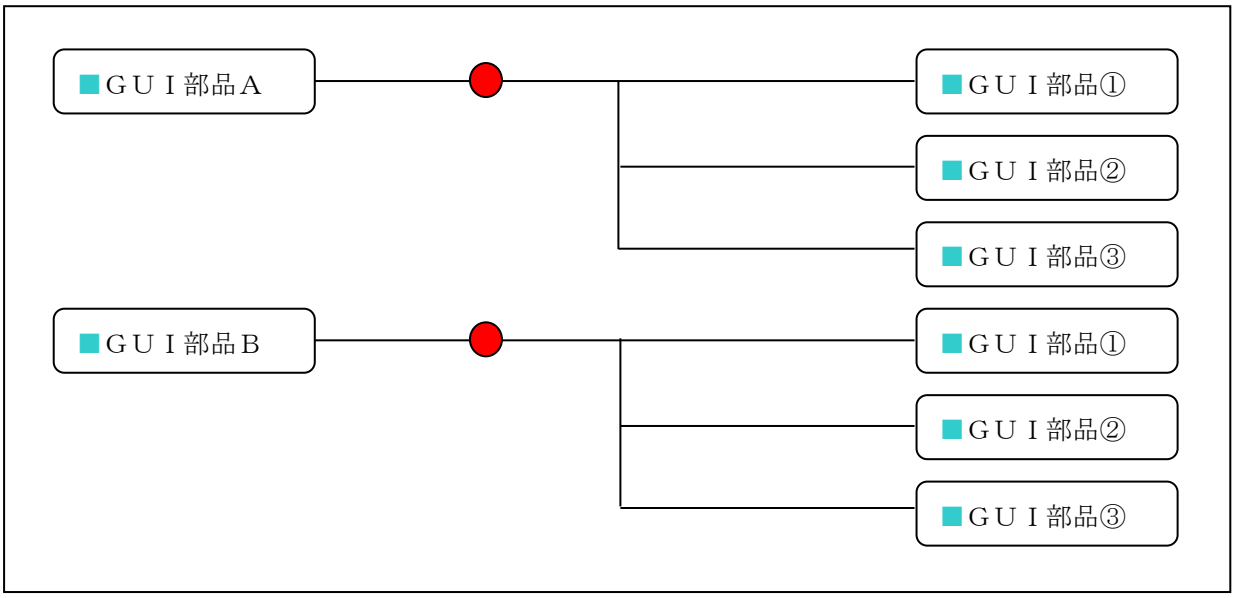


２）サブルーチンとは

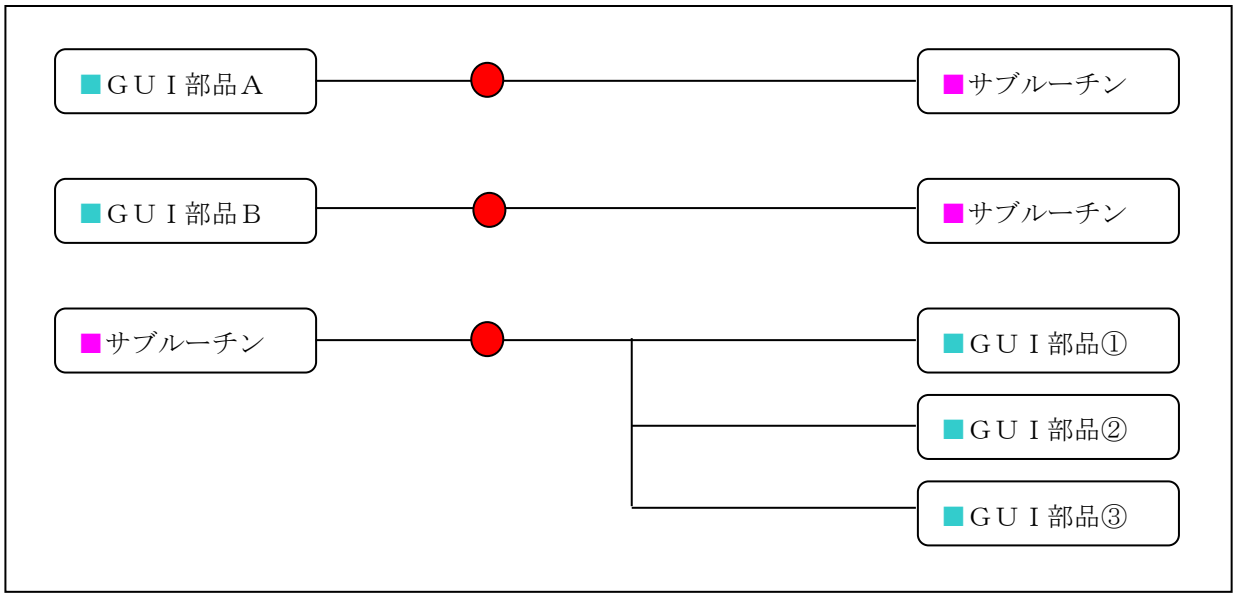
１つのイベント処理で複数のメソッドを起動している場合、それらのメソッドを「サブルーチン」としてまとめておくと、他のイベント処理で再利用するときなどに便利です。

例

サブルーチン化していない場合：ＧＵＩ部品ＡとＧＵＩ部品ＢでＧＵＩ部品①～③の処理をしている
これをそれぞれに記述しているためわかりづらい



サブルーチン化している場合：ＧＵＩ部品①～③の処理をまとめてサブルーチンとすると、
同じ処理の繰り返しをスッキリ記述できる



３）サブルーチンの利用

ここでは２つのサブルーチンを使用します。

・[演算実行サブルーチン]

・[初期化サブルーチン]

考え方

1．[サブルーチン（演算実行サブルーチン)]は、[算術演算子コンポーネント格納変数（Key:内部演算子格納変数)]と[文字列格納変数（Key:入力文字列格納変数)]がセットされていたらその演算を実行し、その後[内部演算子格納変数]と[文字列格納変数（Key:入力文字列格納変数)]を初期化するという処理を行います。

2．[サブルーチン（初期化サブルーチン)]は、アプリケーション起動時と[C（クリア）ボタン]を押したときに、初期化処理を呼び出します。初期化処理とは[内部演算子格納変数]を初期化（空に）して[文字列格納変数（Key:入力文字列格納変数)]に「0」を設定するということです。

接続確認

コンポーネント同士の接続を確認します。

変数が NULL（空）でなかったら演算する❶

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■サブルーチン（ID:23, Key: 演算実行サブルーチン）
発生イベント	アクションイベント
接続先コンポーネント❶	■Null 判定（ID:25）
起動メソッド	オペランド設定後、演算を行う(Object)
＜引数＞	説明：オペランド 取得方法：メソッド戻り値 コンポーネント：算術演算子コンポーネント格納変数 （ID:22, Key:内部演算子格納変数） メソッド／値：演算子を取得する

初期化する❷

接続先コンポーネント❷	■算術演算子コンポーネント格納変数（ID:22, Key:内部演算子格納変数）
起動メソッド	初期化する()

入力文字列格納変数に０を入れて初期化する❸

接続先コンポーネント❸	■文字列格納変数（ID:16, Key:入力文字列格納変数）
起動メソッド	文字列を設定する(イベント発生なし)(String)
＜引数＞	説明：なし 取得方法：固定値 メソッド／値：0

内部演算子格納変数を初期化する❶

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■サブルーチン（ID:24, Key: 初期化サブルーチン）
発生イベント	アクションイベント
接続先コンポーネント❶	■算術演算子コンポーネント格納変数（ID:22, Key:内部演算子格納変数）
起動メソッド	初期化する()

入力文字列格納変数に０を入れて初期化する❷

接続先コンポーネント❷	■文字列格納変数（ID:16, Key:入力文字列格納変数）
起動メソッド	文字列を設定する(String)
＜引数＞	説明：設定する文字列 取得方法：固定値 メソッド／値：0

操 作演算子を格納する変数を用意しましょう。

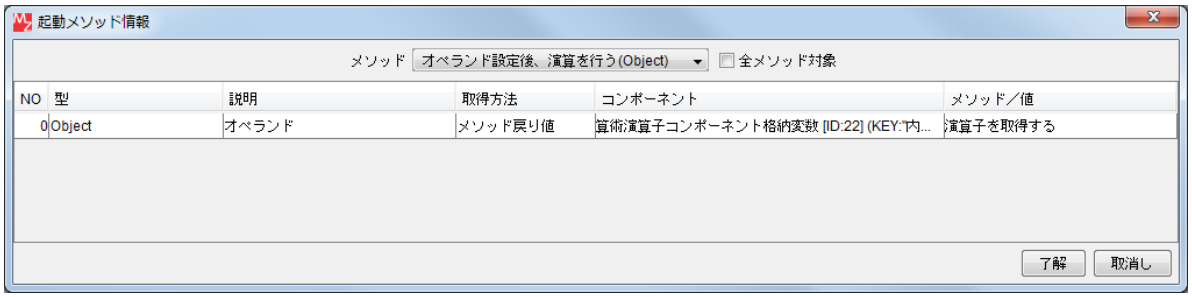
―――変数が NULL（空）でなかったら演算する❶―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［演算実行サブルーチン(ID:23)］コンポーネント上で右クリックー［イベント処理追加］ー［アクションイベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［演算実行サブルーチン(ID:23)］コンポーネントの［アクションイベント］上で、右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー［内部演算子 Null 判定(ID:25)］コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。

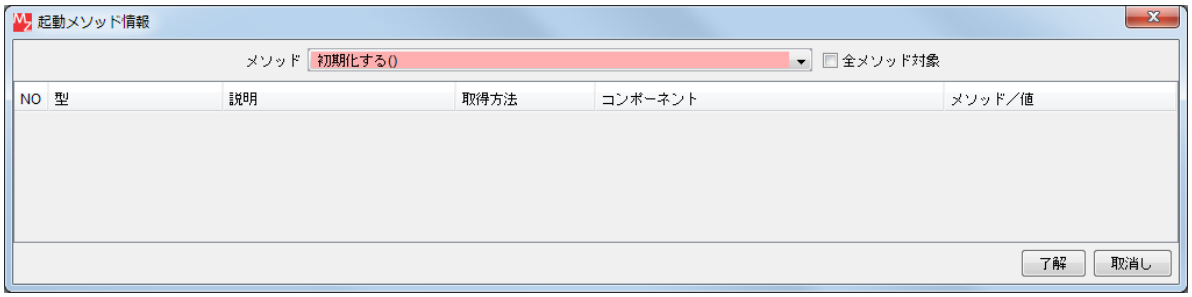
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［オペランド設定後、演算を行う(Object)］をクリックします。
引数を設定します。
説明：オペランド
取得方法：メソッド戻り値
コンポーネント：算術演算子コンポーネント格納変数（ID:22, Key:内部演算子格納変数）
メソッド／値：演算子を取得する
設定後、了解ボタンをクリックします。



―――初期化する❷―――

④　イベントの接続先コンポーネントを選びます。
左側の［演算実行サブルーチン(ID:23)］コンポーネントの［アクションイベント］上で、
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［内部演算子格納変数(ID:22)］コンポーネントをクリックします。

⑤　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［初期化する()］をクリックします。
設定後、了解ボタンをクリックします。

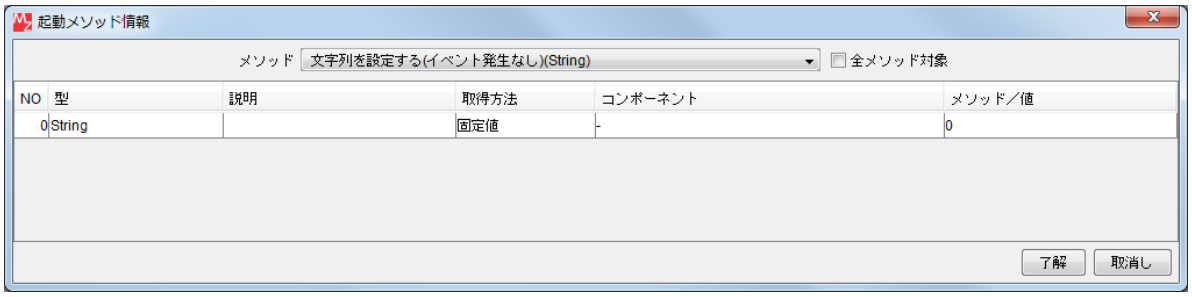


―――変数に０を入れて初期化する❸―――

⑥　イベントの接続先コンポーネントを選びます。
左側の［演算実行サブルーチン(ID:23)］コンポーネントの［アクションイベント］上で、
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－

［文字列格納変数（ID:16 Key:入力文字列格納変数)］コンポーネントをクリックします。

⑦ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［文字列を設定する（イベント発生なし）(String)］をクリックします。
引数を設定します。
説明：なし
取得方法：固定値
メソッド／値：0
設定後、了解ボタンをクリックします。

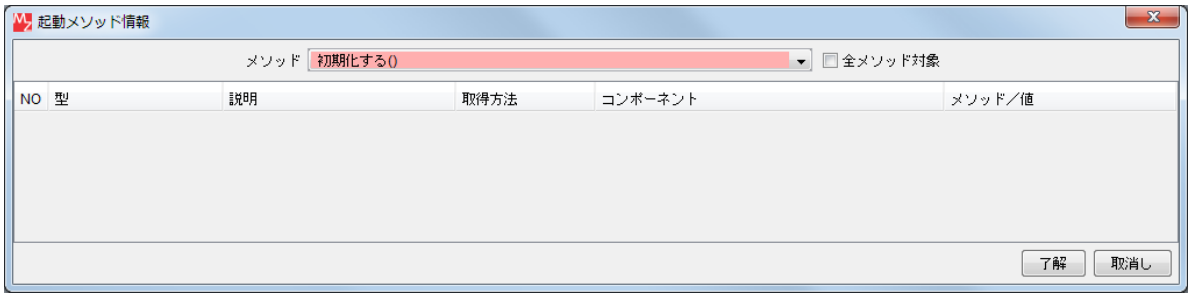


―――内部演算子格納変数を初期化する❶―――

⑧ 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［初期化サブルーチン(ID:24)］コンポーネント上で右クリックー［イベント処理追加］ー［アクションイベント］とクリックします。

⑨ イベントの接続先コンポーネントを選びます。
左側の［初期化サブルーチン(ID:24)］コンポーネントの［アクションイベント］上で、
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［内部演算子格納変数(ID:22)］コンポーネントをクリックします。

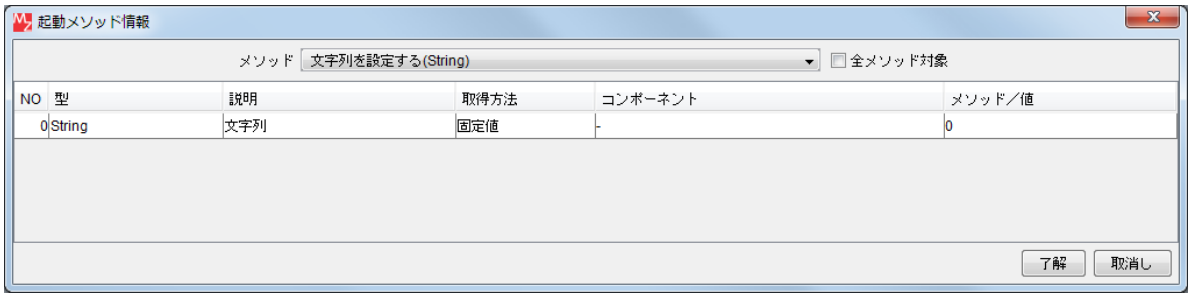
⑩ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［初期化する()］をクリックします。
設定後、了解ボタンをクリックします。



―――変数に 0 を入れて初期化する❷―――

⑪　イベントの接続先コンポーネントを選びます。
左側の［初期化サブルーチン(ID:24)］コンポーネントの［アクションイベント］上で、
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［文字列格納変数（ID:16 Key:入力文字列格納変数）()］コンポーネントをクリックします。

⑫　接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［文字列を設定する(String)］をクリックします。
引数を設定します。
　説明：文字列
　取得方法：固定値
　メソッド／値：0
設定後、了解ボタンをクリックします。



４）Ｎｕｌｌ判定

与えられた値が空(Null)かどうかを判定するメソッドを起動されたときに、処理完了イベントから判定結果をイベント番号で取得します。

接続確認 コンポーネント同士の接続を確認します。

演算子の右側の数値を設定する❶

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■Null 判定（ID:25）
発生イベント	処理完了イベント
接続先コンポーネント❶	■算術演算子コンポーネント格納変数（ID:22, Key:内部演算子格納変数）
起動メソッド	演算子の右側の数値を文字列で設定して数値変換する (String)
＜引数＞	説明：演算子の右側の数値の文字列表現 取得方法：メソッド戻り値 コンポーネント：任意精度実数(BigDecimal)格納変数 (ID:17) メソッド／値：数値(BigDecimal)を取得する
イベント番号	0（Null でない場合＝内部演算子が設定されている）

演算する❷

接続先コンポーネント❷	■算術演算子コンポーネント格納変数（ID:22, Key:内部演算子格納変数）
起動メソッド	演算を実行する()
イベント番号	0（Null でない場合＝内部演算子が設定されている）

操 作 Null 判定を設定しましょう。

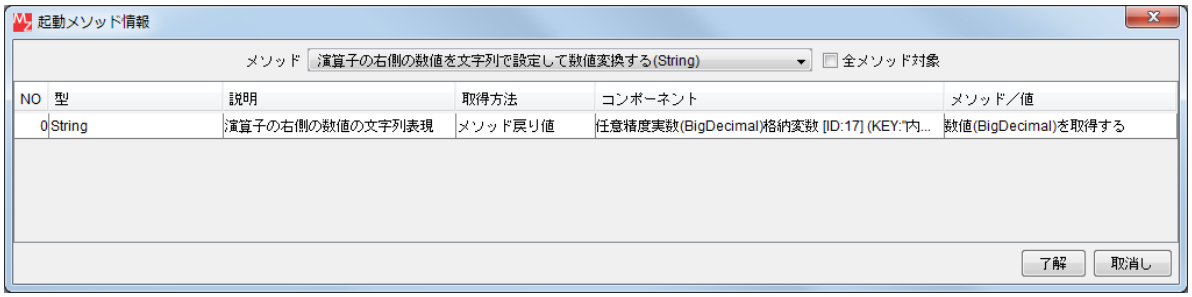
―――内部演算子が設定されていたら演算子の右側の数値を設定する❶―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［内部演算子 Null 判定(ID: 25)］コンポーネント上で右クリックー［イベント処理追加］ー［処理完了イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［内部演算子 Null 判定(ID: 25)］コンポーネントの［処理完了イベント］上で、
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。

右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［内部演算子格納変数(ID: 22)］コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［演算子の右側の数値を文字列で設定して数値変換する(String)］をクリックします。
引数を設定します。
説明：演算子の右側の数値の文字列表現
取得方法：メソッド戻り値
コンポーネント：任意精度実数(BigDecimal)格納変数(ID:17)
メソッド／値：数値(BigDecimal)を取得する
設定後、了解ボタンをクリックします。

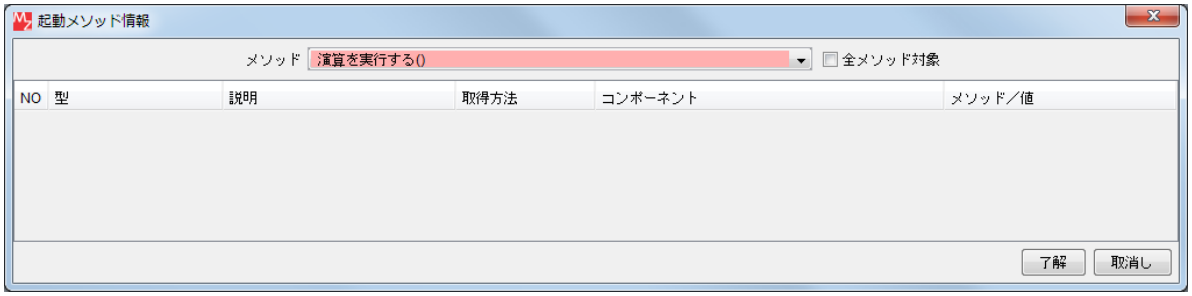


④ イベント番号を設定します。
［内部演算子格納変数(ID: 22)］コンポーネントの上で右クリックー［イベント番号設定］
ー［イベント番号設定...］をクリックします。
定常起動のチェックをオフにして[evaluate メソッドで演算結果が false のとき]をチェックします。

―――演算する❷―――

⑤ イベントの接続先コンポーネントを選びます。
左側の［内部演算子 Null 判定(ID: 25)］コンポーネントの［処理完了イベント］上で、
右クリックー［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［内部演算子格納変数(ID: 22)］コンポーネントをクリックします。

⑥ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［演算を実行する()］をクリックします。
設定後、了解ボタンをクリックします。



⑦　イベント番号を設定します。
［内部演算子格納変数(ID: 22)］コンポーネントの上で右クリック－［イベント番号設定］
－［イベント番号設定...］をクリックします。
定常起動のチェックをオフにして[evaluate メソッドで演算結果が false のとき]をチェックします。

５）［＋ボタン］コンポーネントの起動メソッド変更

［(＋) ボタン］コンポーネントにはこれまでは３つの起動メソッドが設定されていましたが、その３つの作業を［入力演算子格納変数］で行うように設定を変更したので［入力演算子格納変数］へ接続を変更します。

起動メソッド１つを変更し、残りの起動メソッド（２つ）は削除します。

接続確認　コンポーネント同士の接続を確認します。

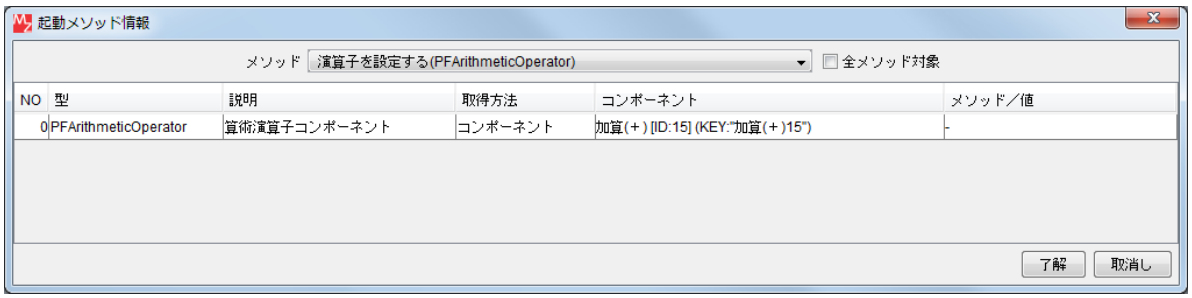
入力演算子を設定する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（+）
発生イベント	アクションイベント
接続先コンポーネント	■算術演算子コンポーネント格納変数（ID:21, Key:入力演算子格納変数）
起動メソッド	演算子を設定する(PFArithmeticOperator)
＜引数＞	説明：算術演算子コンポーネント 取得方法：コンポーネント コンポーネント：加算（＋）(ID:15)

操 作　［(＋) ボタン］コンポーネントの起動メソッドを変更します。

①　イベントの接続先コンポーネントを変更します。
左側の［(＋) ボタン(ID:11)］コンポーネントから接続されている一番上の［加算（＋）(ID:15)］
メソッド上で右クリック－［接続コンポーネント選択］－［入力演算子格納変数(ID:21)］と
クリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［演算子を設定する(PFArirhmeticOperator)］をクリックします。
引数を設定します。
説明：算術演算子コンポーネント
取得方法：コンポーネント
コンポーネント：加算（＋）(ID:15)
設定後、了解ボタンをクリックします。



③ 左側の［(＋）ボタン］コンポーネントから接続されている［加算（＋）(ID:15)］メソッドの上で右クリックー［起動メソッド削除］をクリックします。(もう１つも同様の操作で削除します)

６）［＝ボタン］、［加算（＋）］コンポーネントの起動メソッド変更・削除

［(＝）ボタン］コンポーネントにはこれまでは２つの起動メソッドが設定されていましたが、その２つの作業を［演算実行サブルーチン］で行うように設定を変更したので［演算実行サブルーチン］へ接続を変更します。
起動メソッド１つを変更し、残りの起動メソッド（１つ）は削除します。

［加算（＋)］コンポーネントにはこれまでは３つの起動メソッドが設定されていましたが、その３つの作業を［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］１つで行うように設定を変更したので起動メソッド１つを変更し、残りの起動メソッド（２つ）は削除します。

接続確認 コンポーネント同士の接続を確認します。

［(＝）ボタン］コンポーネントの接続先を変更する

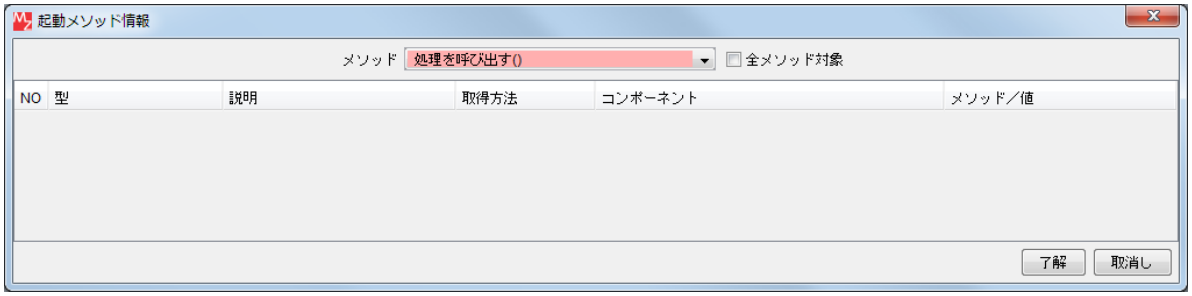
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（=)
発生イベント	アクションイベント
接続先コンポーネント	■サブルーチン（ID:23, Key:演算実行サブルーチン）
起動メソッド	処理を呼び出す()

操 作

―――［(＝) ボタン］コンポーネントの起動メソッドを変更―――

① イベントの接続先コンポーネントを変更します。
左側の［(＝) ボタン］コンポーネントから接続されている一番上の［加算 (＋) (ID:15)］メソッド上で右クリック－［接続コンポーネント選択］－［演算実行サブルーチン(ID:23)］とクリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド (処理) を選びます。
［メソッド］の ▼ をクリックします。
［処理を呼び出す()］をクリックします。
設定後、了解ボタンをクリックします。



③ 左側の［(＝) ボタン］コンポーネントから接続されている［加算 (＋) (ID:15)］メソッドの上で右クリック－［起動メソッド削除］をクリックします。

―――［加算 (＋)］コンポーネントの起動メソッドを変更―――

④ 左側の［加算 (＋) (ID:15)］コンポーネントから接続されている［任意精度実数(BigDecimal)格納変数 (Key:内部数値格納変数) (ID:17)］のメソッドだけを残して他の２つ ([加算 (＋) (ID:15)］メソッドと［文字列格納変数 (ID:16 Key:入力文字列格納変数)］メソッド) を削除します。

7）［アプリケーション］、［Cボタン］コンポーネントの起動メソッド変更・削除

アプリケーション起動時と［C（クリア）ボタン］を押したときに前に定義した初期化処理を実行するように変更します。

［アプリケーション］コンポーネントは起動時に［文字列格納変数（Key:入力文字列格納変数)］を初期化していましたが、その作業を［初期化サブルーチン］で行うように設定を変更したので［初期化サブルーチン］へ接続を変更します。

［C（クリア）ボタン］コンポーネントはイベント発生時に［加算（＋)］を初期化していましたが、その作業を［初期化サブルーチン］で行うように設定を変更したので［初期化サブルーチン］へ接続を変更します。

接続確認 コンポーネント同士の接続を確認します。

［アプリケーション］コンポーネントの変数を初期化する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■アプリケーション
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■サブルーチン（ID:24, Key:初期化サブルーチン）
起動メソッド	処理を呼び出す()

［C（クリア）ボタン］コンポーネントの変数を初期化する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（C)
発生イベント	アクションイベント
接続先コンポーネント	■サブルーチン（ID:24, Key:初期化サブルーチン）
起動メソッド	処理を呼び出す()

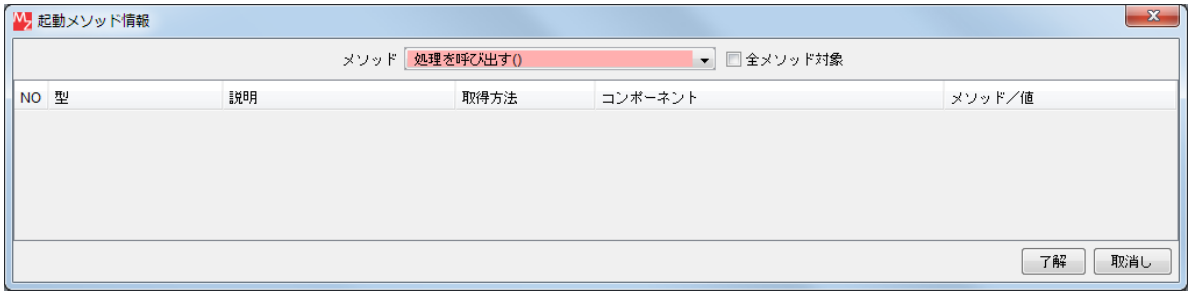
操 作

―――［アプリケーション］コンポーネントの起動メソッドを変更―――

① イベントの接続先コンポーネントを変更します。
左側の［アプリケーション］コンポーネントから接続されている［文字列格納変数（ID:16 Key:入力文字列格納変数)］メソッド上で右クリック－［接続コンポーネント選択］－［初期化サブルーチン(ID:24)］とクリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。

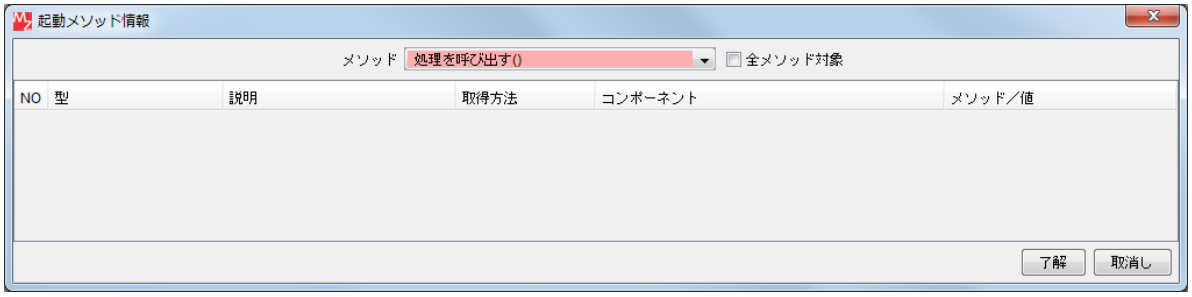
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［処理を呼び出す()］をクリックします。
設定後、了解ボタンをクリックします。



―――［C（クリア）ボタン］コンポーネントの起動メソッドを変更―――

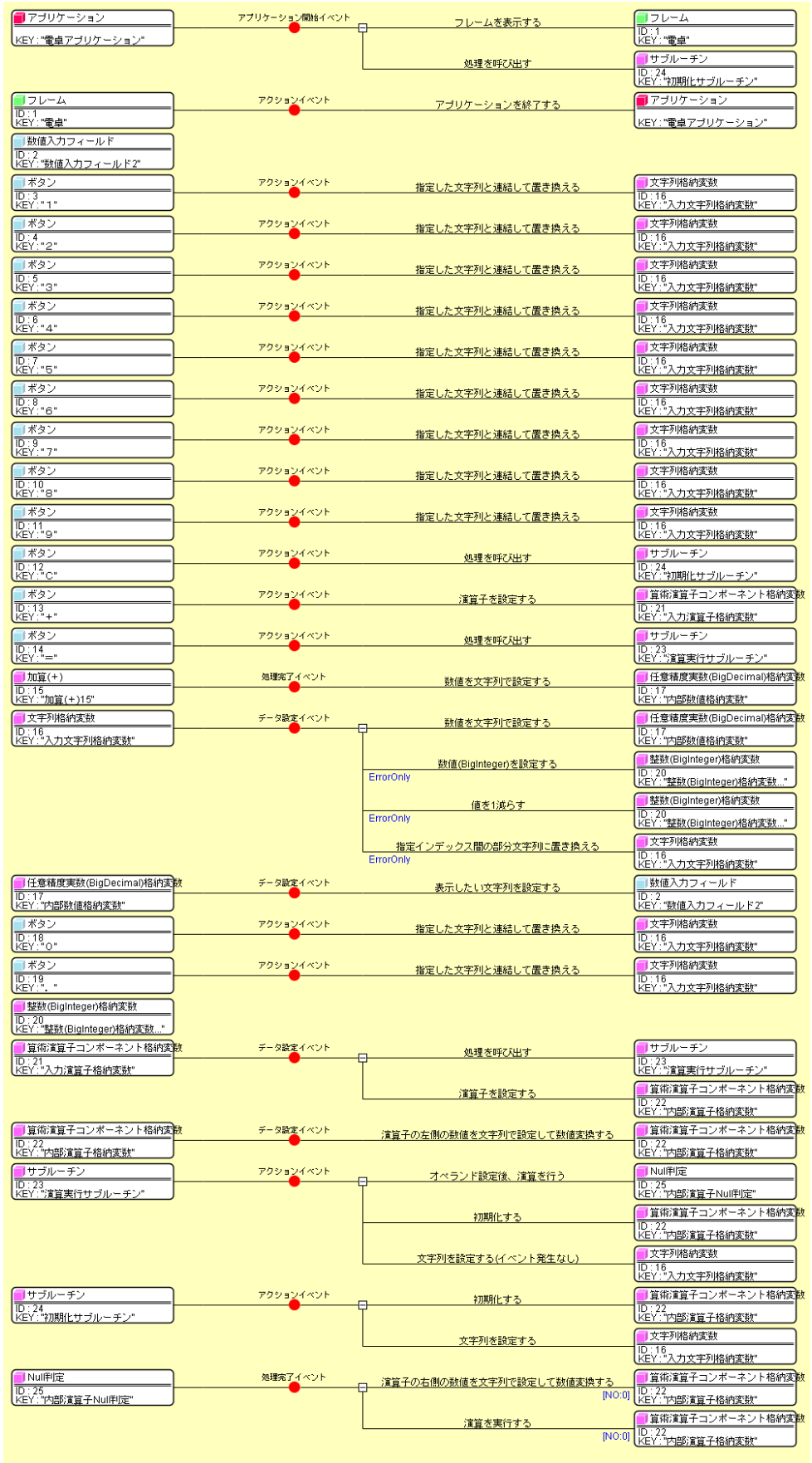
① イベントの接続先コンポーネントを変更します。
左側の［C（クリア）ボタン］コンポーネントから接続されている［加算（＋）(ID:15)］
メソッド上で右クリック－［接続コンポーネント選択］－［初期化サブルーチン(ID:24)］と
クリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［処理を呼び出す()］をクリックします。
設定後、了解ボタンをクリックします。



まとめ

ここまで進めるとビルダー上では以下のようになります。

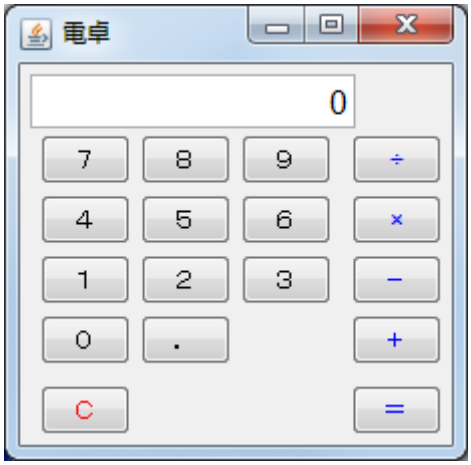


８）四則演算の完成

（－）ボタン、（×）ボタン、（÷）ボタンを追加して四則演算できるようにします。

完成図

四則演算ができる状態にします。



考え方

［(＋）ボタン］や［加算（＋)］を接続した場合と同じように、［(－）ボタン］、［(×）ボタン］、［(÷）ボタン］が押された情報を［入力演算子格納変数］に渡します。また、［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］内で［減算（－)］［乗算（×)］［除算（÷)］を処理します。

準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■ボタン	3	［画面構成部品］－［ボタン］－［ボタン］
■減算(-)	1	［処理部品］－［演算制御］－［減算（－)］
■乗算(×)	1	［処理部品］－［演算制御］－［乗算（×)］
■除算(÷)	1	［処理部品］－［演算制御］－［除算（÷)］

操 作

① 必要なコンポーネントを追加します。
コンポーネントの数が多いので一括追加します。
作業領域で右クリック－［コンポーネント一括追加］とクリックします。
右側の領域にコンポーネントの分類が表示されるのでここから

［画面構成部品］－［ボタン］－［ボタン］を３つ追加、
［処理部品］－［演算制御］－［減算（－)］を１つ追加、
［処理部品］－［演算制御］－［乗算（×)］を１つ追加、
［処理部品］－［演算制御］－［除算（÷)］を１つ追加します。

② コンポーネントの名前を変更しておきます。
１つめの［ボタン(ID:26)］を［－］
２つめの［ボタン(ID:27)］を［×］
３つめの［ボタン(ID:28)］を［÷］
と変更します。

画面編集

① 画面を作成します。
画面編集をクリックします。
［ボタン］コンポーネントを３つフレームに追加し、体裁を整えます。

接続確認

コンポーネント同士の接続を確認します。

［(－）ボタン］［(×）ボタン］［(÷）ボタン］を接続する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（ID:26, Key:-）
発生イベント	アクションイベント
接続先コンポーネント	■算術演算子コンポーネント格納変数（ID:21, Key:入力演算子格納変数）
起動メソッド	演算子を設定する(PFArithmeticOperator)
＜引数＞	説明：算術演算子コンポーネント 取得方法：コンポーネント コンポーネント：減算(-)

［減算（－)］［乗算（×)］［除算（÷)］を接続する

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■減算(-)（ID:29）
発生イベント	処理完了イベント
接続先コンポーネント	■任意精度実数(BigDecimal)格納変数（ID:17, Key:内部数値格納変数）
起動メソッド	数値を文字列で設定する(String)
＜引数＞	説明：数値の文字列表現 取得方法：イベント内包 メソッド／値：処理結果データ

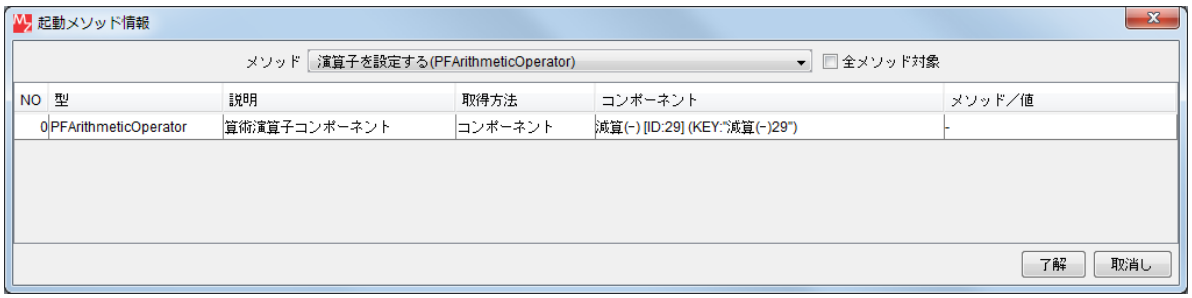
操 作

――― [(－) ボタン] [(×) ボタン] [(÷) ボタン] を接続する―――

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の [(－) ボタン(ID:26)] コンポーネント上で右クリック－ [イベント処理追加] －[アクションイベント] とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の [(－) ボタン(ID:26)] コンポーネントの [アクションイベント] 上で
右クリック－ [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－ [接続コンポーネント選択] －[入力演算子格納変数(ID:21)] コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－ [起動メソッド設定] をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
[メソッド] の ▼ をクリックします。
[演算子を設定する(PFArithmeticOperator)] をクリックします。
引数を設定します。
説明：算術演算子コンポーネント
取得方法：コンポーネント
コンポーネント：減算(-)(ID:29)
設定後、了解ボタンをクリックします。



④ ①～③の操作を繰り返して、[(×) ボタン] [(÷) ボタン] を設定します。
メソッド引数の [コンポーネント] はそれぞれの演算によって異なります。

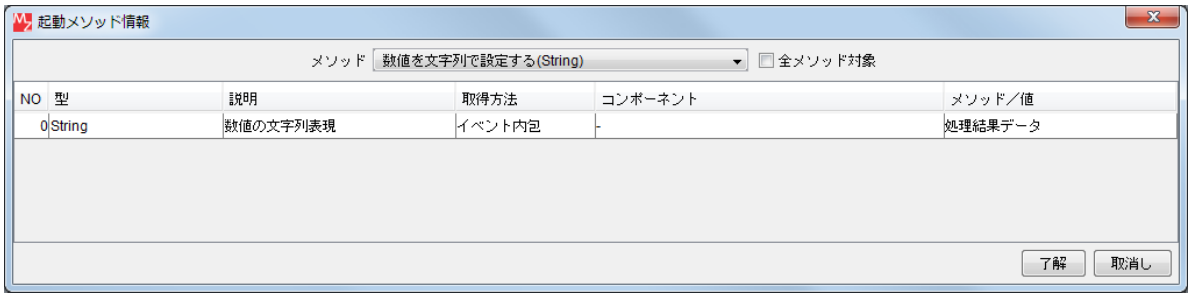
――― [減算 (－)] [乗算 (×)] [除算 (÷)] を接続する―――

⑤ 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の [減算 (－) (ID:29)] コンポーネント上で右クリック－ [イベント処理追加] －[処理完了イベント] とクリックします。

⑥ イベントの接続先コンポーネントを選びます。
左側の [減算 (－) (ID:29)] コンポーネントの [処理完了イベント] 上で
右クリック－ [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。

右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［任意精度実数(BigDecimal)格納変数（ID:17 Key:内部数値格納変数)］コンポーネントをクリックします。

⑦ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［数値を文字列で設定する(String)］をクリックします。
引数を設定します。
説明：数値の文字列表現
取得方法：イベント内包
メソッド／値：処理結果データ
設定後、了解ボタンをクリックします。



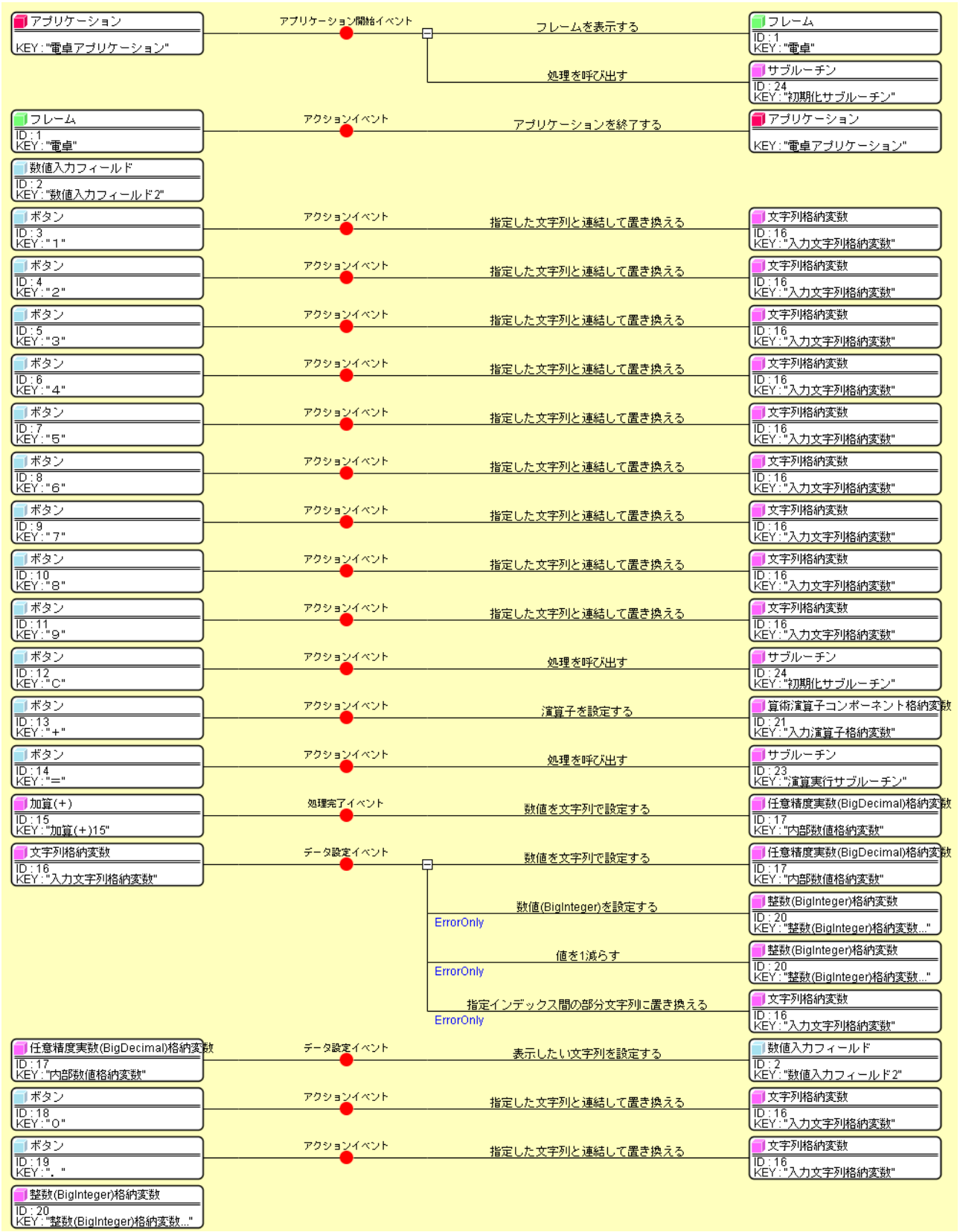
⑧ ⑤～⑦を繰り返して［乗算（×）(ID:30)］コンポーネント、［除算（÷）(ID:31)］コンポーネントに接続します。

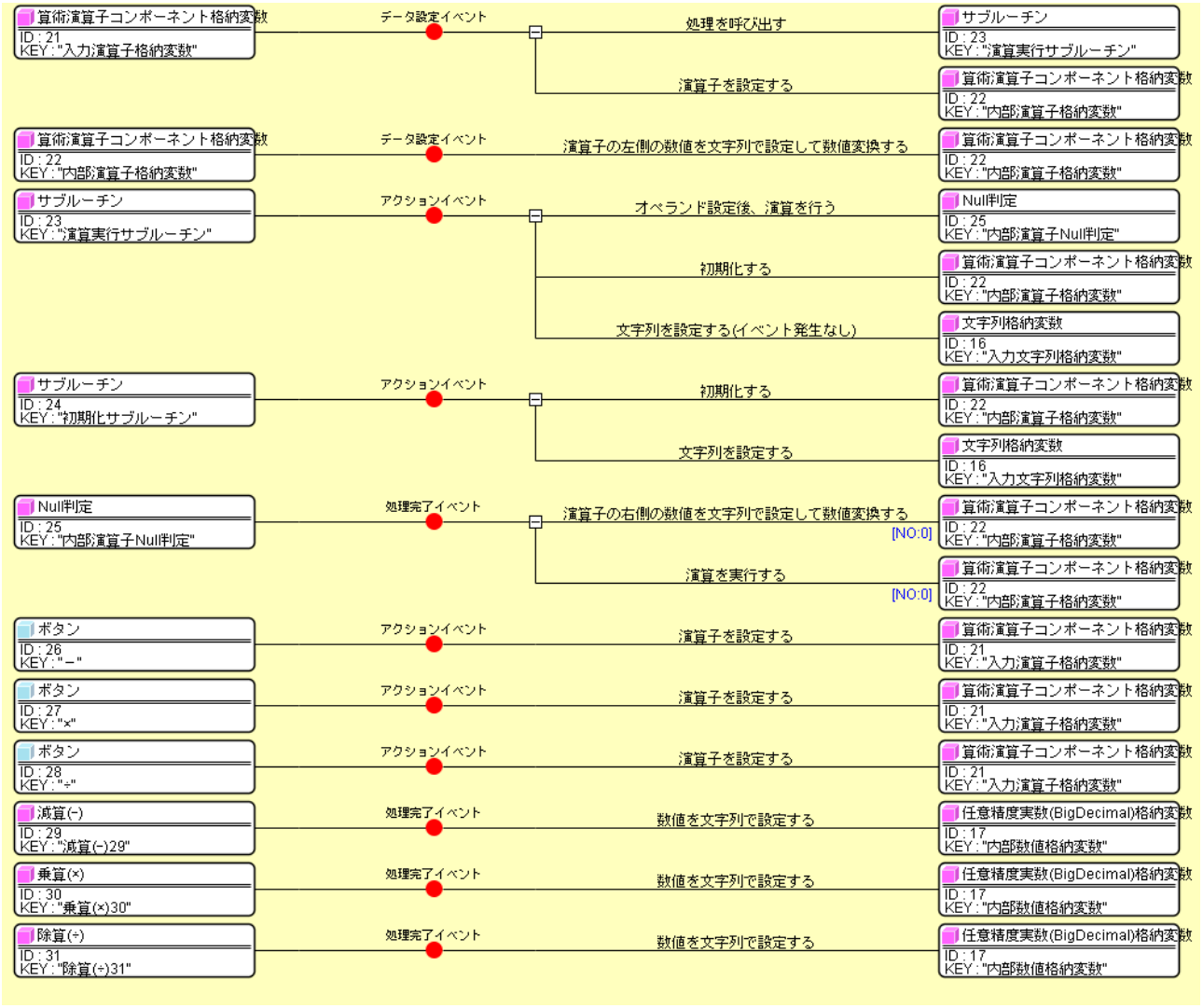
⑨ 確認します。
実行（設定可）で実行します。
電卓アプリケーションで四則演算できることを確認します。

⑬ 保存します。
（次の複合コンポーネントで利用するので必ず保存します）

まとめ

ここまで進めるとビルダー上では以下のようになります。



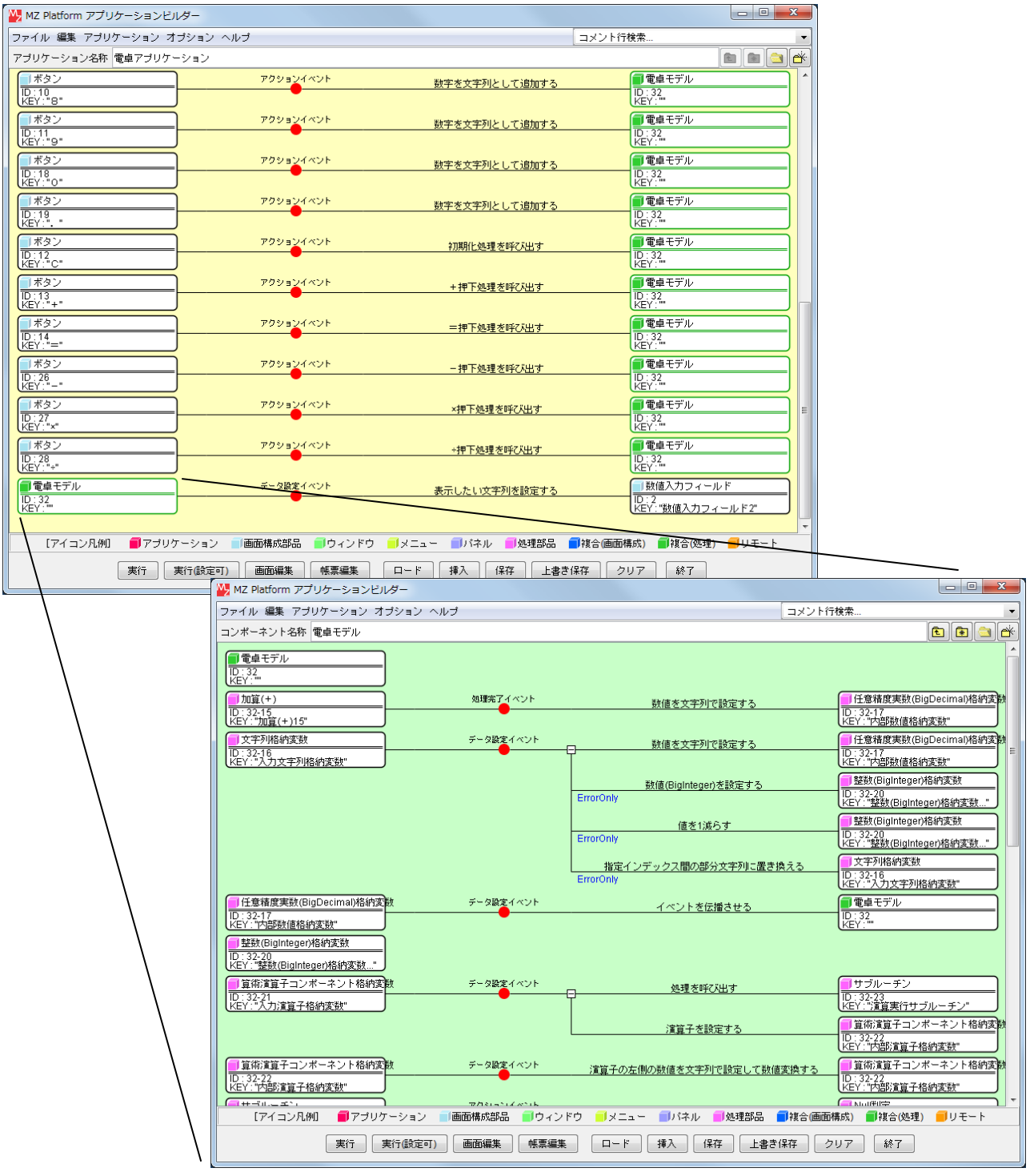

算術演算子コンポーネント格納変数
ID : 21
KEY : "入力演算子格納変数"
データ設定イベント
処理を呼び出す
サブルーチン
ID : 23
KEY : "演算実行サブルーチン"
演算子を設定する
算術演算子コンポーネント格納変数
ID : 22
KEY : "内部演算子格納変数"
算術演算子コンポーネント格納変数
ID : 22
KEY : "内部演算子格納変数"
データ設定イベント
演算子の左側の数値を文字列で設定して数値変換する
算術演算子コンポーネント格納変数
ID : 22
KEY : "内部演算子格納変数"
サブルーチン
ID : 23
KEY : "演算実行サブルーチン"
アクションイベント
オペランド設定後、演算を行う
Null判定
ID : 25
KEY : "内部演算子Null判定"
初期化する
算術演算子コンポーネント格納変数
ID : 22
KEY : "内部演算子格納変数"
文字列を設定する(イベント発生なし)
文字列格納変数
ID : 16
KEY : "入力文字列格納変数"
サブルーチン
ID : 24
KEY : "初期化サブルーチン"
アクションイベント
初期化する
算術演算子コンポーネント格納変数
ID : 22
KEY : "内部演算子格納変数"
文字列を設定する
文字列格納変数
ID : 16
KEY : "入力文字列格納変数"
Null判定
ID : 25
KEY : "内部演算子Null判定"
処理完了イベント
演算子の右側の数値を文字列で設定して数値変換する
[NO:0]
算術演算子コンポーネント格納変数
ID : 22
KEY : "内部演算子格納変数"
演算を実行する
[NO:0]
算術演算子コンポーネント格納変数
ID : 22
KEY : "内部演算子格納変数"
ボタン
ID : 26
KEY : "－"
アクションイベント
演算子を設定する
算術演算子コンポーネント格納変数
ID : 21
KEY : "入力演算子格納変数"
ボタン
ID : 27
KEY : "×"
アクションイベント
演算子を設定する
算術演算子コンポーネント格納変数
ID : 21
KEY : "入力演算子格納変数"
ボタン
ID : 28
KEY : "÷"
アクションイベント
演算子を設定する
算術演算子コンポーネント格納変数
ID : 21
KEY : "入力演算子格納変数"
減算(-)
ID : 29
KEY : "減算(-)29"
処理完了イベント
数値を文字列で設定する
任意精度実数(BigDecimal)格納変数
ID : 17
KEY : "内部数値格納変数"
乗算(×)
ID : 30
KEY : "乗算(×)30"
処理完了イベント
数値を文字列で設定する
任意精度実数(BigDecimal)格納変数
ID : 17
KEY : "内部数値格納変数"
除算(÷)
ID : 31
KEY : "除算(÷)31"
処理完了イベント
数値を文字列で設定する
任意精度実数(BigDecimal)格納変数
ID : 17
KEY : "内部数値格納変数"

Step.4 **複合コンポーネントによる階層化**

複合コンポーネントとは、いくつかのコンポーネントとひとまとめにして新たなコンポーネントを定義することです。以下のような場合に便利です。

1．繰り返し使われる操作をまとめておく
2．ビルダー上の記述が長くなってしまった場合に整理する
3．機能単位にまとめておき開発作業の効率を上げたい
4．後のメンテナンス時に見やすくしたい

複合コンポーネントは、違う階層にコンポーネントをまとめておく方法です。



考え方

1．［複合コンポーネント］を追加する
2．内部処理の部分を複合コンポーネントに追加する
3．元の階層から［複合コンポーネント］で追加したコンポーネントを削除する

準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■複合コンポーネント	1	

操 作

複合コンポーネントを追加しましょう。

① 必要なコンポーネントを追加します。
作業領域で右クリックー［複合コンポーネント作成］ー［コンポーネント］を追加します。

② 追加した［複合コンポーネント］をダブルクリックします。

［複合コンポーネント］の中に入ります。画面が緑色に変わります。

Step.5 複合コンポーネントの利用

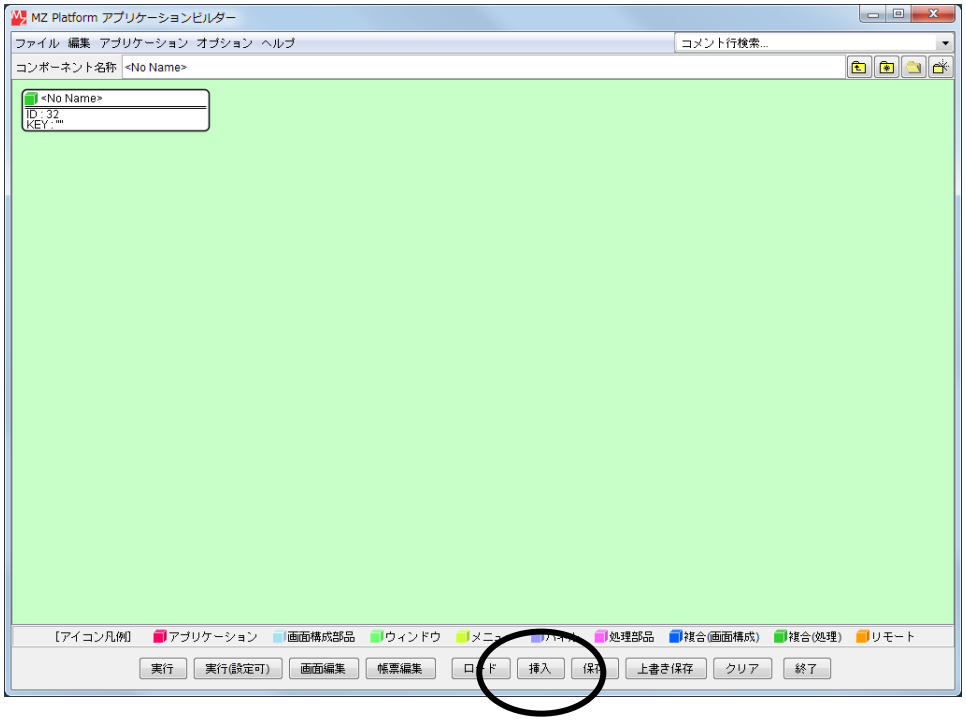
複合コンポーネントを利用しましょう。

１）複合コンポーネントの追加

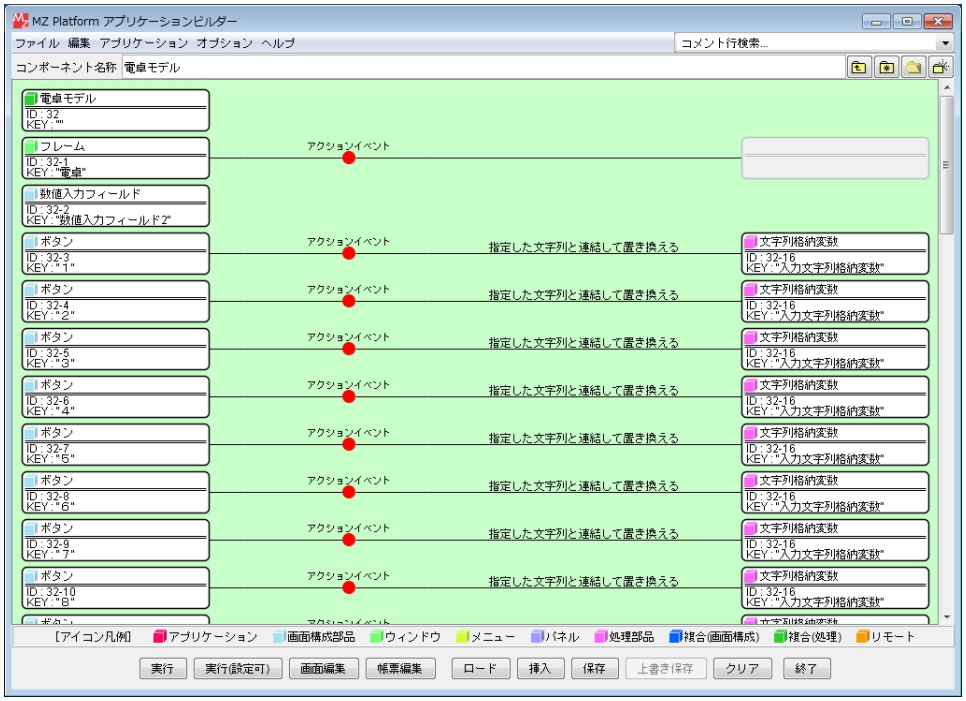
複合コンポーネントの中を作ります。上の階層の内部処理の部分を複合コンポーネントに入れるので、［挿入］メニューを使い保存してあるファイルを挿入します。

操 作 複合コンポーネントの中にファイルを挿入しましょう。

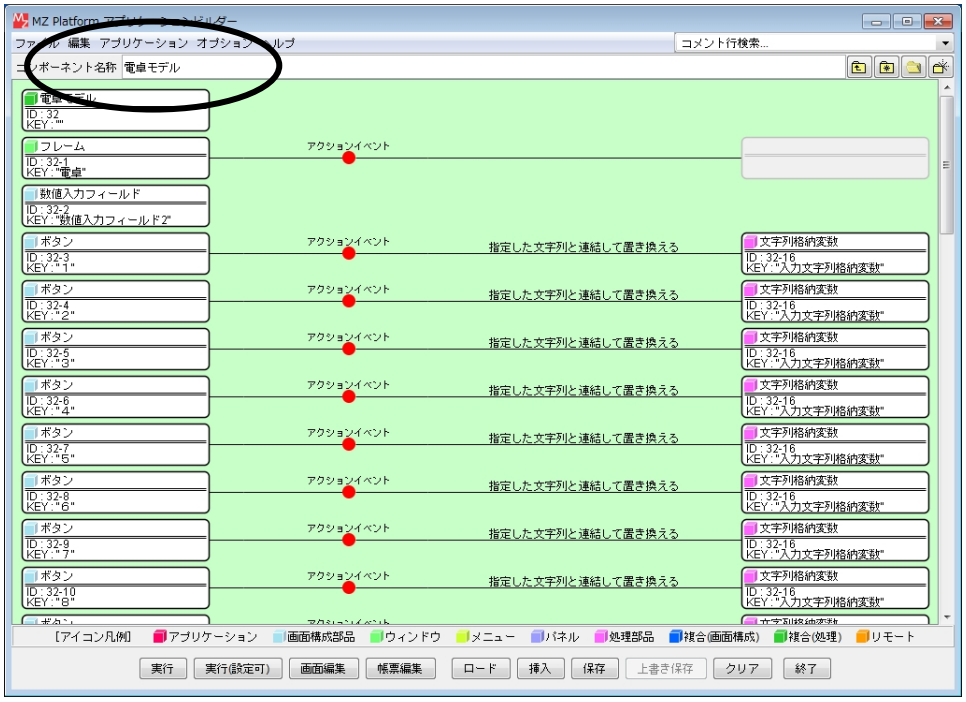
① 複合コンポーネントの中で挿入ボタンをクリックします。
先ほど保存したファイルを選択しロードします。



複合コンポーネントに追加されます。



② 複合コンポーネントに「電卓モデル」と名前をつけておきます。



２）複合コンポーネントの編集

挿入したファイルを整理します。

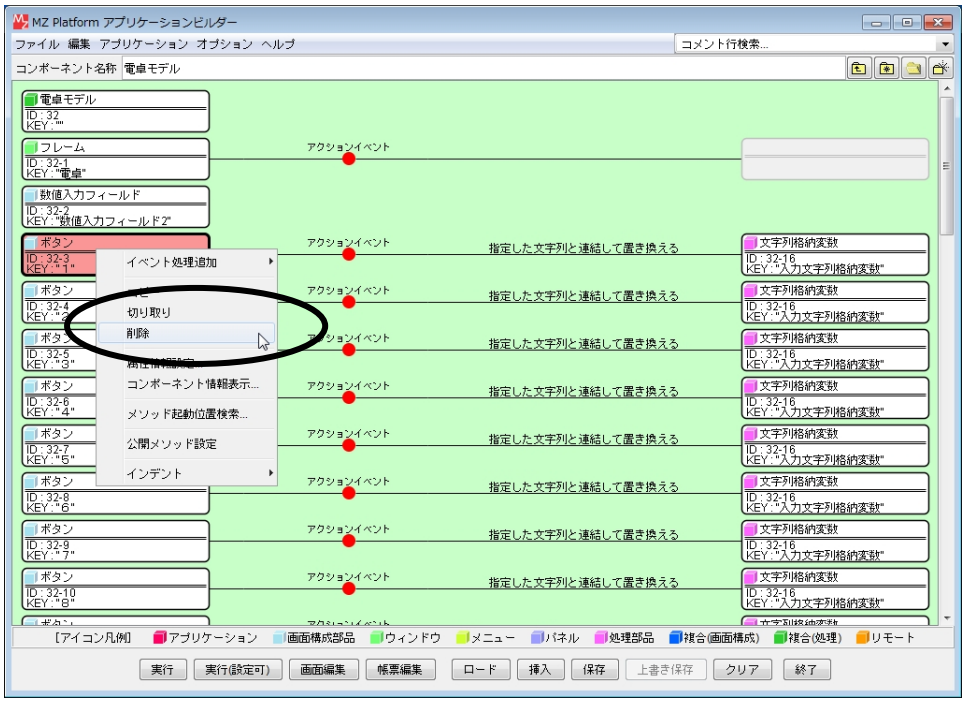
ここで作成しようとしている［電卓モデル］複合コンポーネントにはＧＵＩ部品を含めないことにしますので、複合コンポーネント内から GUI 部品のコンポーネントを削除します。

操 作

複合コンポーネントの中を編集しましょう。

ここでは［ボタン］コンポーネント全部と［数値入力フィールド］コンポーネント、[フレーム]コンポーネントを削除します。

① ［ボタン］コンポーネントを削除します。
［ボタン(1)］コンポーネントの上で右クリック－［削除］をクリックします。

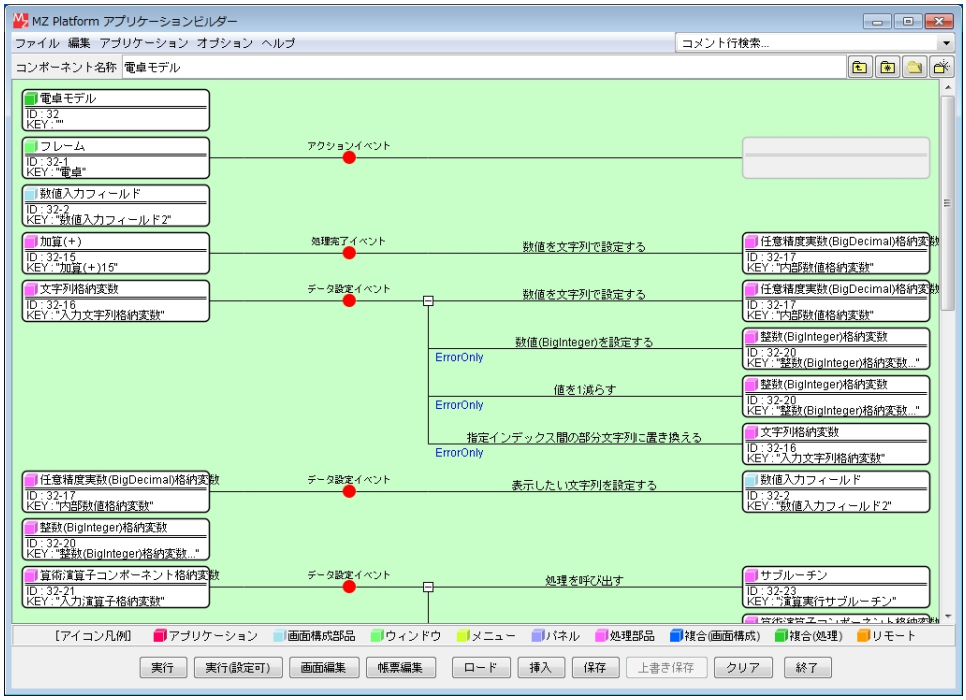


② 確認ダイアログボックスが表示されるので［はい］をクリックします。

③ ［ボタン(1)］コンポーネントが削除されます。

④ 以下のボタンを削除します。
［ボタン(2)］～［ボタン(9)］、［ボタン(0)］、［ボタン(C)］、［ボタン(+)］、［ボタン(=)］
［ボタン(0)］、［ボタン(.)］、［ボタン(-)］、［ボタン(×)］、［ボタン(÷)］
[Shift]キーを押しながら、コンポーネントをクリックしていくと、まとめて選択できます。

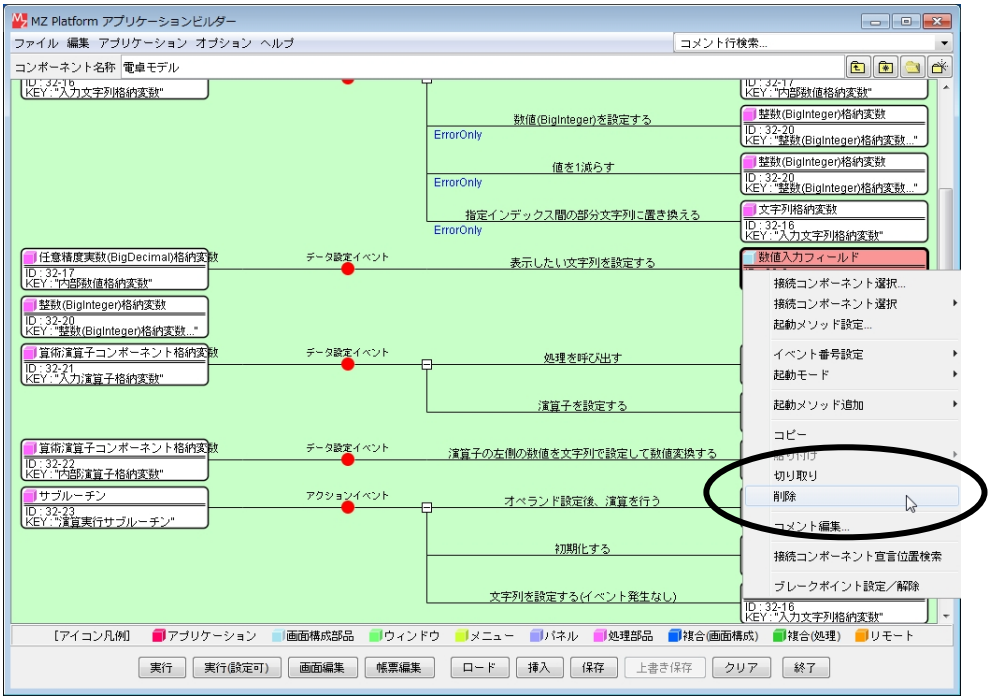
以下のようになります。



⑤　［数値入力フィールド］を削除します。

［数値入力フィールド（ID:32-2)］がメソッドとして接続されているコンポーネントがあるので、その分を削除します。

［任意精度実数(BigDecimal)格納変数（Key:内部数値格納変数)］に接続されている［数値入力フィールド（ID:32-2)］の上で右クリックー［削除］をクリックします。



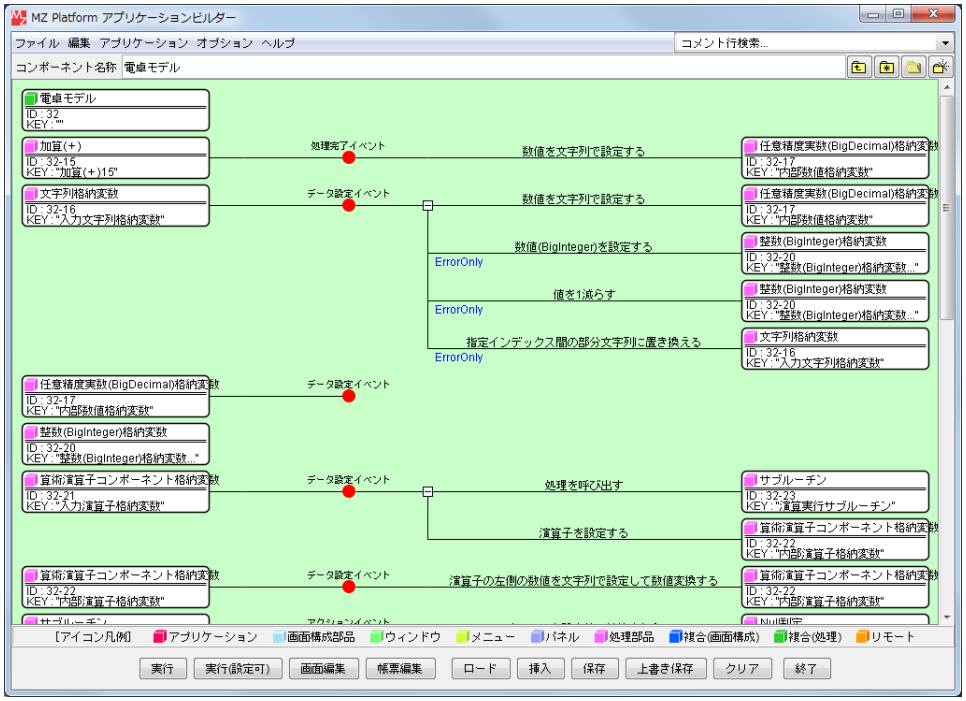
⑥　確認ダイアログボックスが表示されるのではいをクリックします。

⑦　［数値入力フィールド（ID:32-2)］コンポーネントの上で右クリックー［コンポーネント削除］をクリックします。

⑧　確認ダイアログボックスが表示されるのではいをクリックします。

⑨　［フレームボタン］コンポーネントを削除します。
同様に、［フレーム（ID:32-1)］コンポーネントの上で右クリックー［コンポーネント削除］をクリックし、［フレーム（ID:32-1)］コンポーネントを削除します。

以下のようになります。



３）イベント伝播

［数値入力フィールド］コンポーネントが削除されたので、［任意精度実数(BigDecimal)格納変数（ID:32-17 Key:内部数値格納変数)］コンポーネントの接続先がなくなっています。［任意精度実数(BigDecimal)格納変数 ］コンポーネントの接続先は上の階層の［数値入力フィールド］です。それには「外部へのイベント通知」が必要になります。

これを実現するのが［イベント伝播］というメソッドになります。これによって複合コンポーネント内部にある［任意精度実数(BigDecimal)格納変数 ］でデータ設定イベントが発生したときに［電卓モデル］に通知され、外部のコンポーネントにとっては複合コンポーネントの中からデータ設定イベントが発生したように見えることになります。

この反対で外部から複合コンポーネント内部に対してメソッド起動要求が必要な場合には「公開メソッド設定」を行います。

接続確認 コンポーネント同士の接続を確認します。

複合コンポーネントのイベントを上の階層に送る

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■任意精度実数(BigDecimal)格納変数（ID:32-17, Key:内部数値格納変数）
発生イベント	データ設定イベント
接続先コンポーネント	■電卓モデル（ID:32）
起動メソッド	イベントを伝播させる(PFEvent)
＜引数＞	説明：対象イベント 取得方法：イベント

操 作 ［イベント伝播］メソッドを設定します。

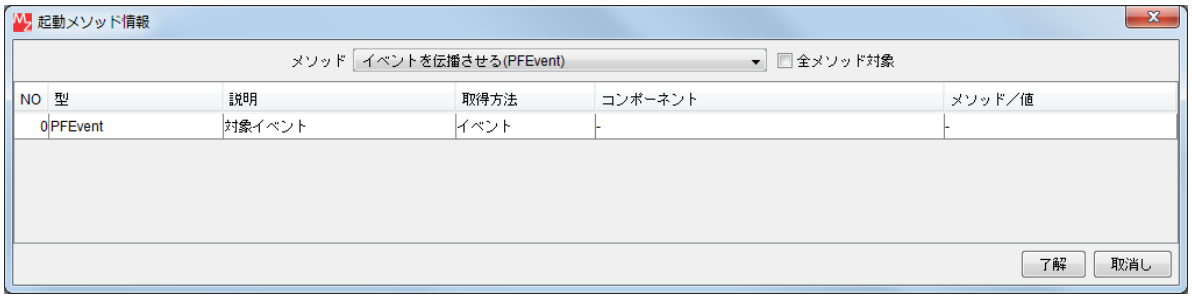
① イベントの接続先コンポーネントを選びます。
左側の［任意精度実数(BigDecimal)格納変数（ID:32-17 Key:内部数値格納変数)］コンポーネントの［データ設定イベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［電卓モデル(ID:32)］コンポーネントをクリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［イベントを伝播させる(PFEvent)］をクリックします。
引数を設定します。

説明：対象イベント

取得方法：イベント

設定後、了解ボタンをクリックします。



4）サブルーチンの利用

複合コンポーネント内の処理の一部をサブルーチンにまとめます。

［入力演算子格納変数］の処理をサブルーチンにまとめます。

準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■サブルーチン	4	［処理部品］－［サブルーチン］－［サブルーチン］

操 作

① 必要なコンポーネントを追加します。
作業領域で右クリック－［コンポーネント一括追加］－［処理部品］－［サブルーチン］
－［サブルーチン］を４つ追加します。

② ［サブルーチン］コンポーネントの名前を変更しておきます。
１つめの［サブルーチン(ID:32-32)］を［＋押下げサブルーチン］
２つめの［サブルーチン(ID:32-33)］を［－押下げサブルーチン］
３つめの［サブルーチン(ID:32-34)］を［×押下げサブルーチン］
４つめの［サブルーチン(ID:32-35)］を［÷押下げサブルーチン］
と変更します。

接続確認

コンポーネント同士の接続を確認します。

演算ボタンの押下げ処理をサブルーチンにまとめる

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■サブルーチン(ID:32-32, Key:＋押下げサブルーチン)
発生イベント	アクションイベント
接続先コンポーネント	■算術演算子コンポーネント格納変数（ID:32-21, Key:入力演算子格納変数）
起動メソッド	演算子を設定する(PFArithmeticOperator)
＜引数＞	説明：算術演算子コンポーネント 取得方法：コンポーネント コンポーネント：加算(+)(ID:32-15)

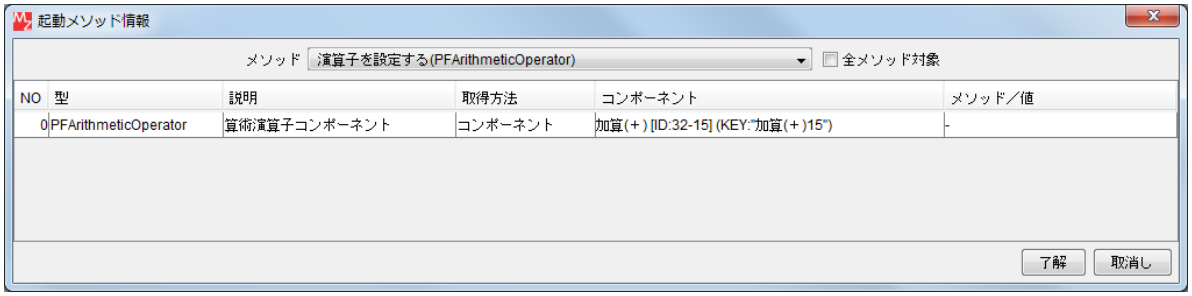
操 作

演算子をサブルーチンにまとめます。

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［＋押下げサブルーチン(ID:32-32)］コンポーネント上で
右クリックー［イベント処理追加］ー［アクションイベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［＋押下げサブルーチン(ID:32-32)］コンポーネントの
［アクションイベント］上で右クリックー［起動メソッド追加］とクリックします。
薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリックー［接続コンポーネント選択］ー
［入力演算子格納変数(ID:32-21)］をクリックします。

③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリックー［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［演算子を設定する(PFArithmeticOperator)］をクリックします。
引数を設定します。
　説明：算術演算子コンポーネント
　取得方法：コンポーネント
　コンポーネント：加算(+)(ID:32-15)
設定後、了解ボタンをクリックします。



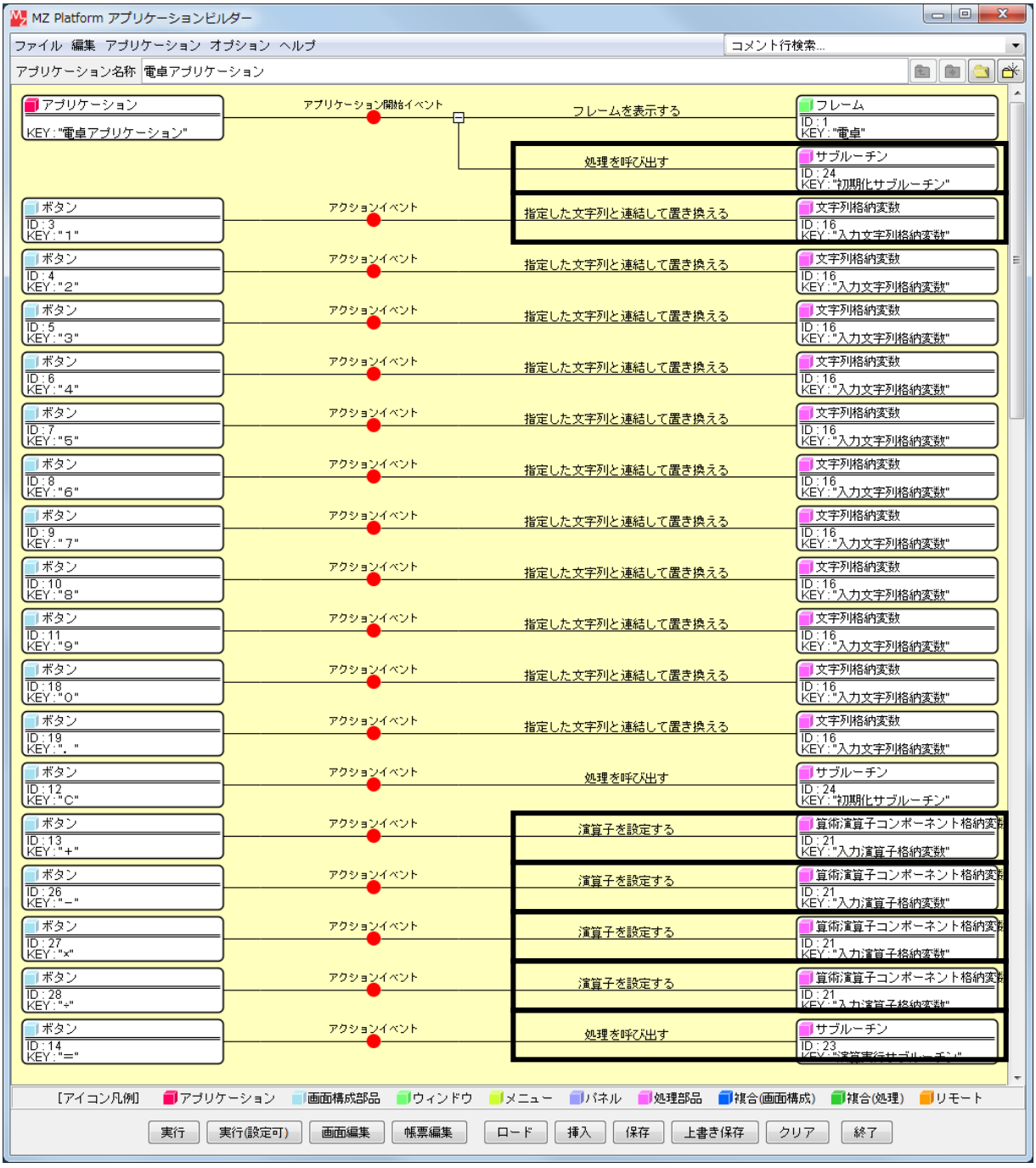
④　①～③の操作を繰り返して、［－押下げサブルーチン］［×押下げサブルーチン］
［÷押下げサブルーチン］を接続します。
（サブルーチンによってメソッドのコンポーネントを変更します）

５）複合コンポーネントの中のメソッドを公開する

複合コンポーネントの中に設定したメソッドを上の階層から呼び出します。
上の階層からメソッドを呼び出すには複合コンポーネントの中のメソッドを上の階層に「公開」して使えるようにする必要があります。
「公開」するメソッドは上の階層で必要なものだけを公開します。

上の階層では［ボタン］コンポーネントと［数値入力フィールド］コンポーネントだけを残して、他のコンポーネントを複合コンポーネントに入れます。したがって、［ボタン］コンポーネントから接続してあるメソッドをすべて公開する必要があります。［数値入力フィールド］は結果を表示するところで使われるので、複合コンポーネントで処理された計算結果を表示します。

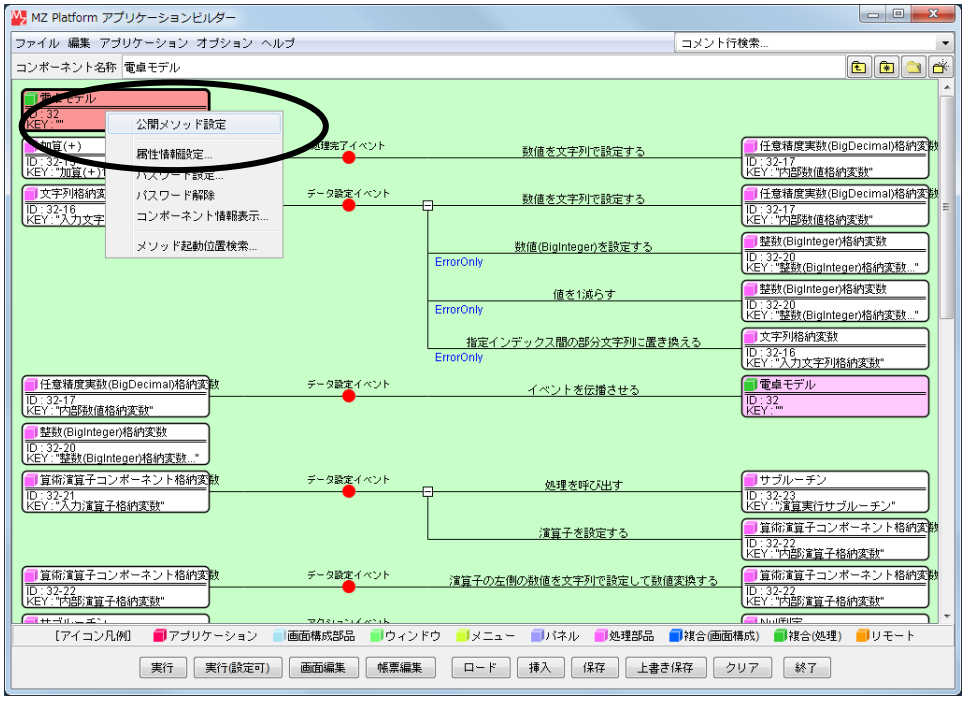
ここで必要なのは以下の７つのメソッドです。これらを公開します。



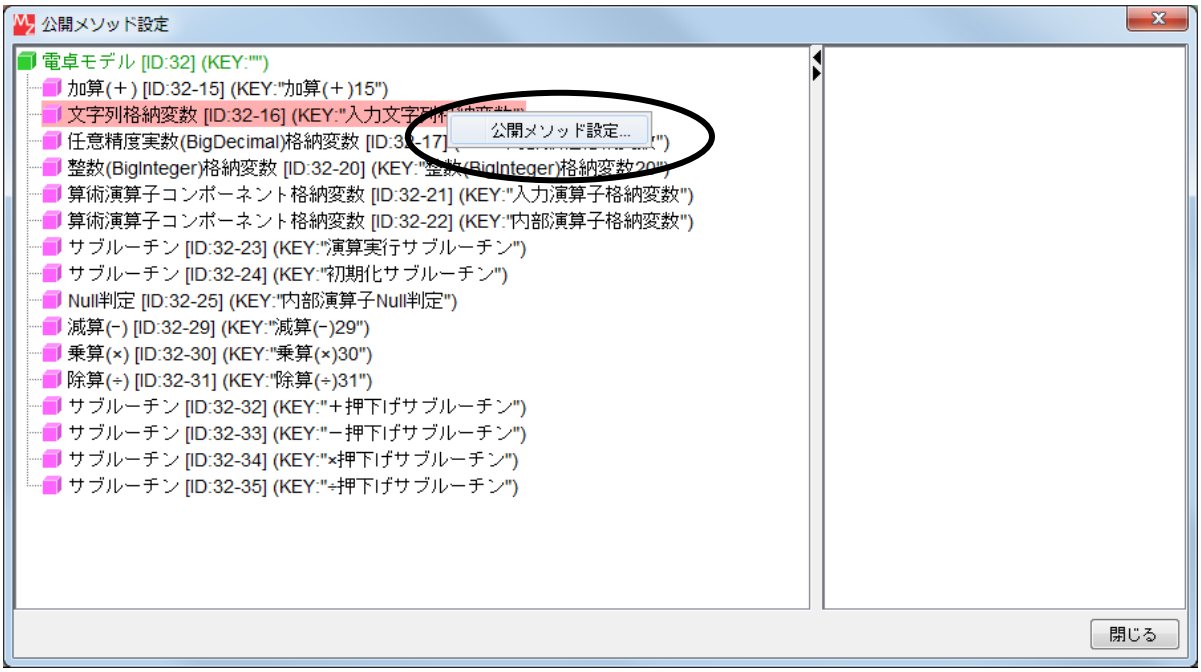
操 作 複合コンポーネントのメソッドを公開しましょう。

① ［電卓モデル］複合コンポーネントをダブルクリックして複合コンポーネントに入ります。

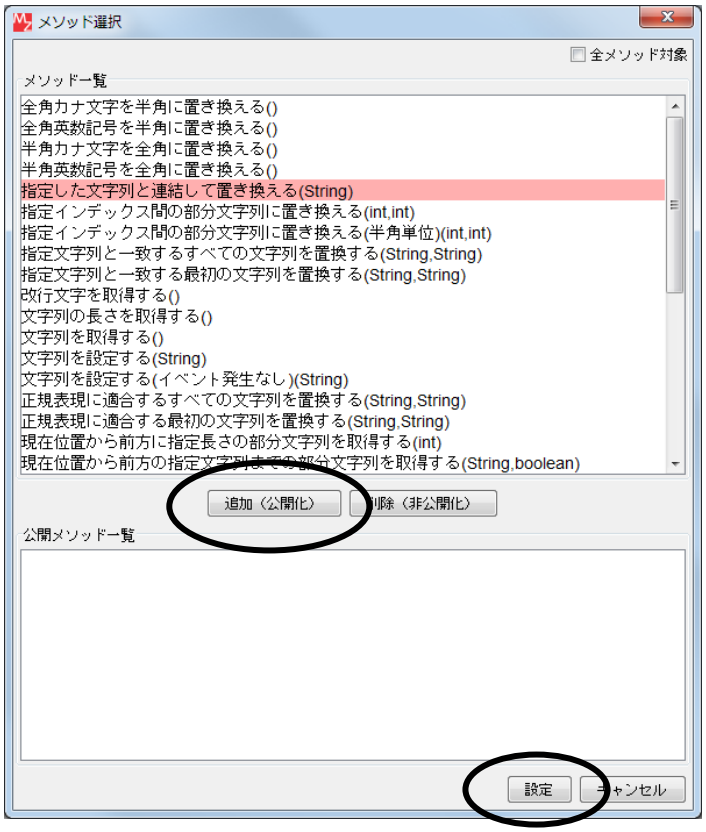
② ［電卓モデル］の複合コンポーネントで［右クリック］－［公開メソッド設定］をクリックします。



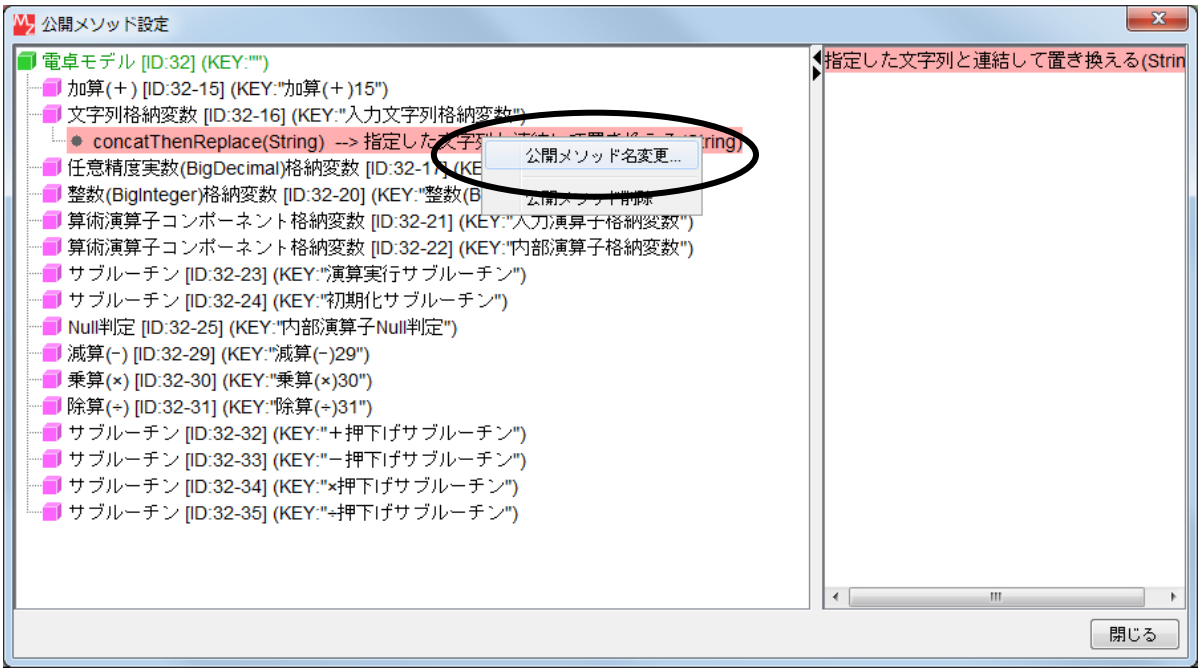
③ 公開するメソッドを選びます。
［文字列格納変数］から選びます。
［文字列格納変数(ID:32-16)］で右クリック－［公開メソッド設定...］をクリックします。



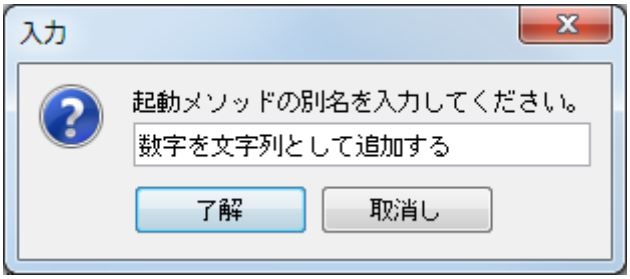
④　［指定した文字列と連結して置き換える(String)］を公開します。
［指定した文字列と連結して置き換える(String)］をクリックします。
追加（公開化）をクリックし、設定をクリックします。



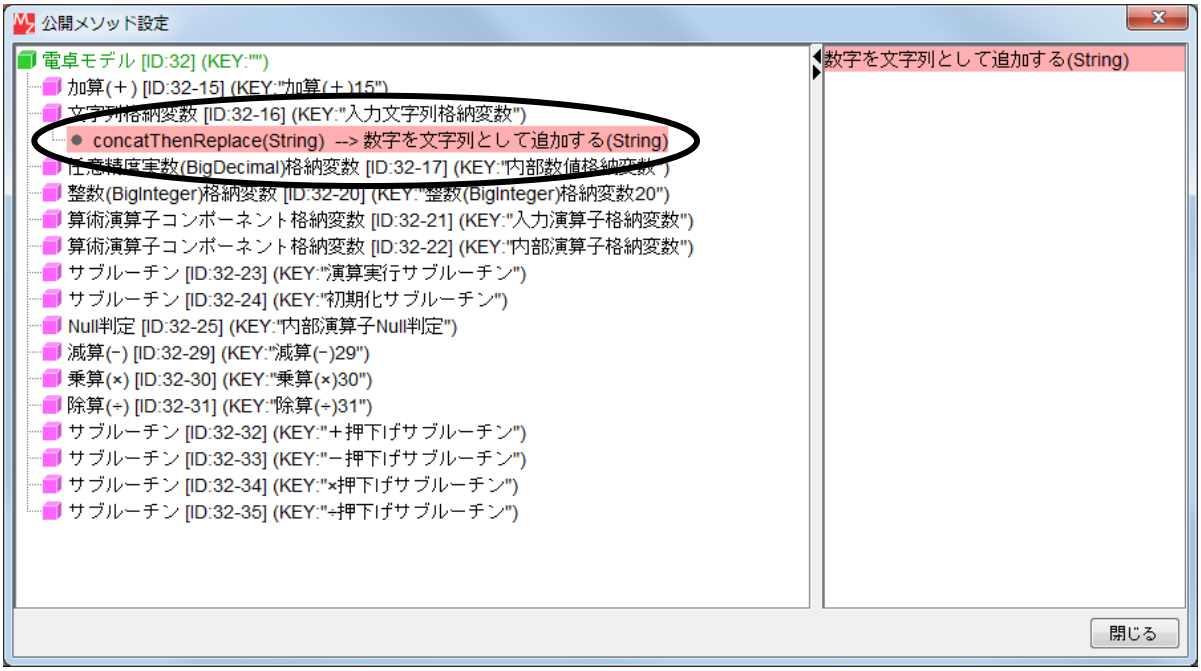
⑤　公開メソッド名を変更します。
公開したメソッドの上で右クリックー［公開メソッド名変更］をクリックします。



⑥　日本語表記として「数字を文字列として追加する」と入力し了解をクリックます。

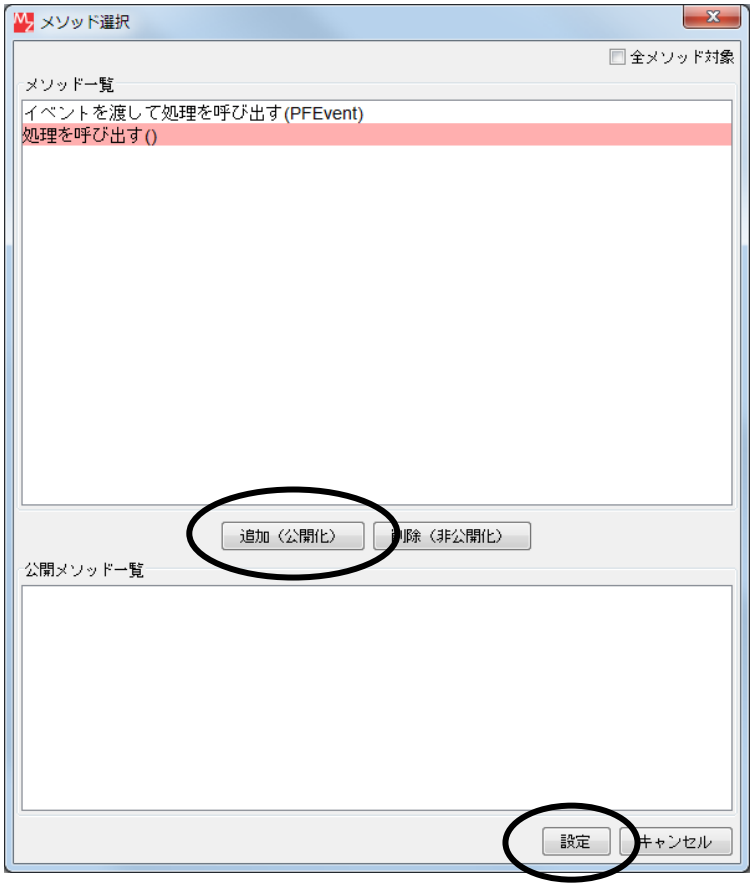


公開メソッドは以下のようになります。



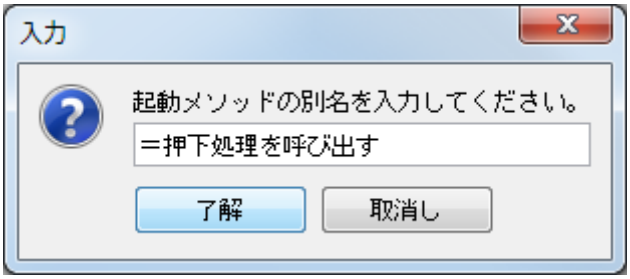
⑦　公開するメソッドを選びます。
［サブルーチン］から選びます。
［演算実行サブルーチン(ID:32-23)］で右クリック－［公開メソッド設定...］をクリックします。

⑧　［処理を呼び出す()］をクリックします。
追加（公開化）をクリックし、設定をクリックします。



⑨　メソッド名をわかりやすくします。
公開したメソッドの上で右クリック－［公開メソッド名変更］をクリックします。

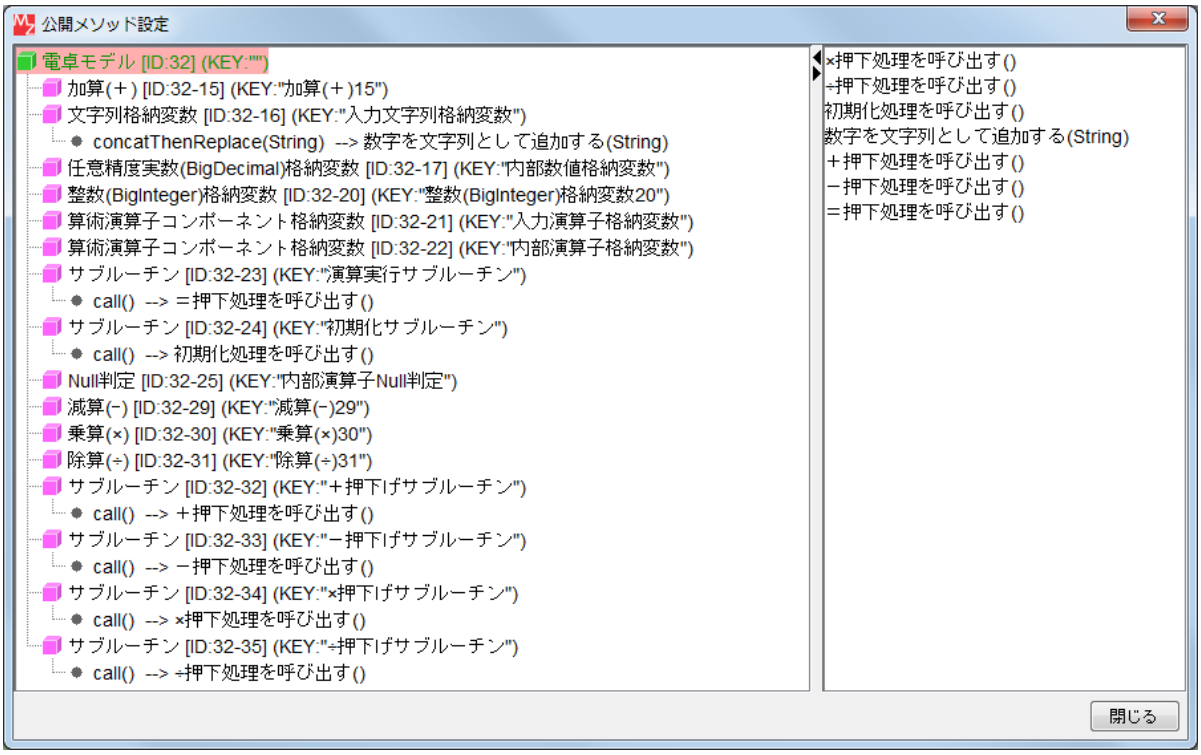
⑩　日本語表記として「＝押下処理を呼び出す」と入力し了解をクリックします。



⑪　⑦～⑩を繰り返して以下のサブルーチンの［処理を呼び出す()］を公開し、次のように名前を変更します。

サブルーチン(ID:32-24)［初期化処理を呼び出す］
サブルーチン(ID:32-32)［＋押下処理を呼び出す］
サブルーチン(ID:32-33)［－押下処理を呼び出す］
サブルーチン(ID:32-34)［×押下処理を呼び出す］
サブルーチン(ID:32-35)［÷押下処理を呼び出す］

公開メソッド設定完了後は次のようになります。閉じるをクリックします。

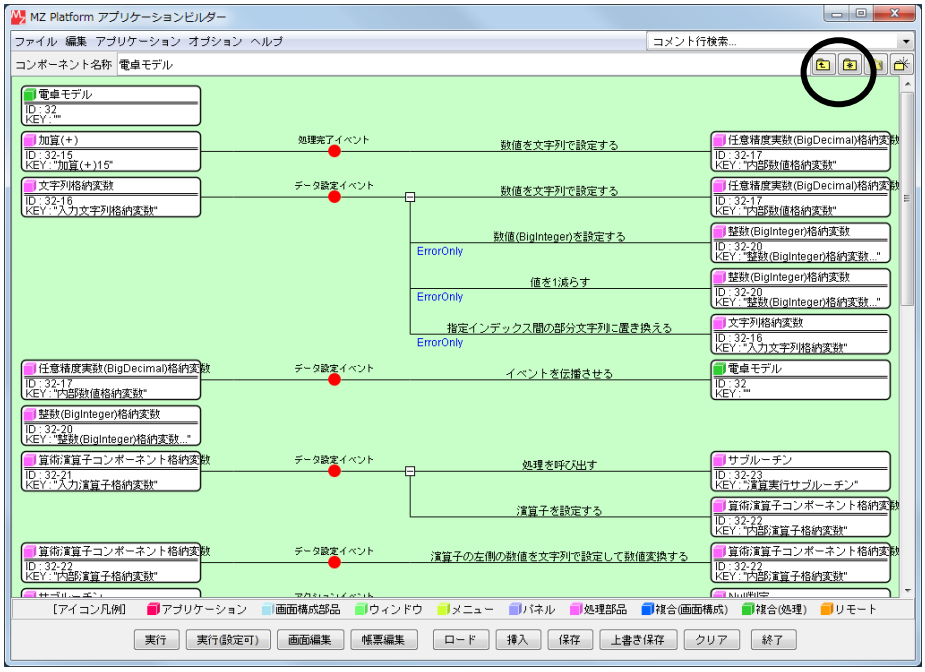


６）公開してあるメソッドを上の階層から使用する

複合コンポーネントで公開したメソッドを上の階層から使用します。
現在設定されているメソッドを削除して複合コンポーネントのメソッドに置き換えます。

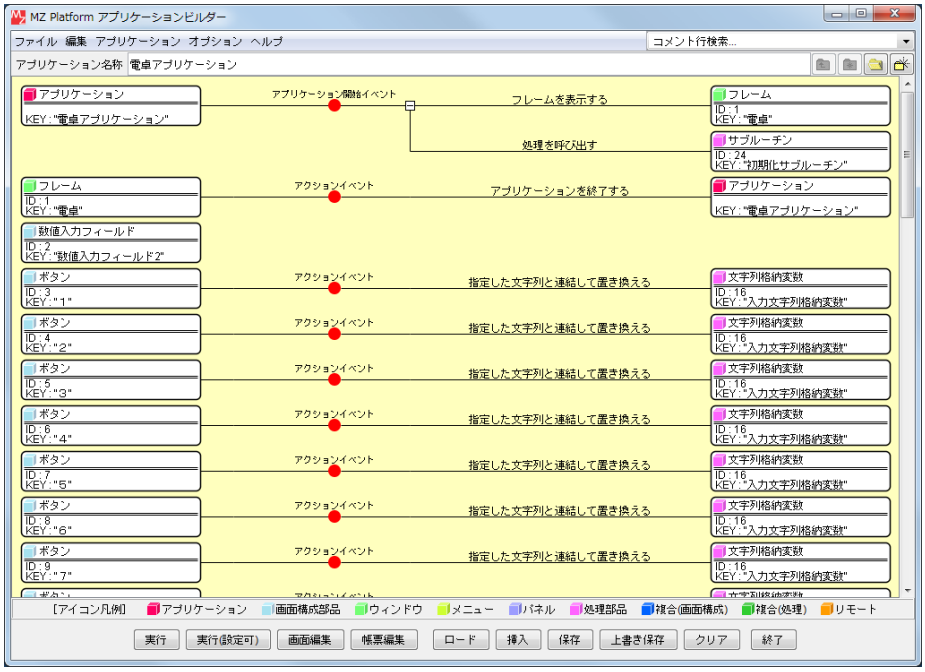
操 作 現在設定されているメソッドを削除しましょう。

① 複合コンポーネントから元の階層に戻ります。
右上の［編集サポートボタン］をクリックして１階層上に上がります。



確認

元の階層に戻ります。

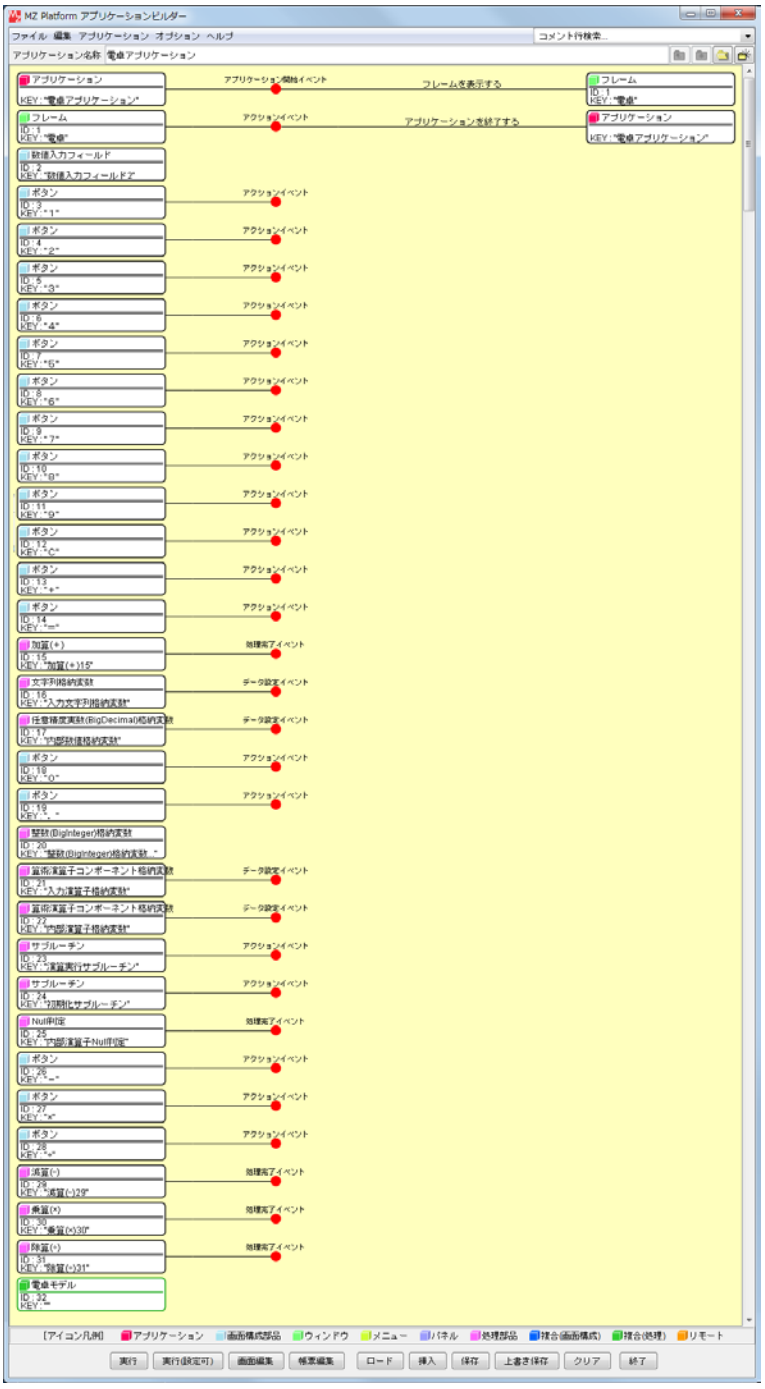


② メソッドを削除します。
［アプリケーション］と接続されている［初期化サブルーチン(ID:24)］を削除します。
［初期化サブルーチン(ID:24)］の上で右クリック－［削除］をクリックします。

③ ［起動メソッドを削除します。よろしいですか？］のメッセージが表示されるので
［はい］をクリックします。

④ ②～③を繰り返して
すべてのメソッドを削除します。

以下のようになります。



操 作 現在設定されている不要なコンポーネントを削除しましょう。
ここでは電卓の画面編集で使用しているコンポーネントは残し、
処理関係の複合コンポーネントに設定したコンポーネントをすべて削除します。

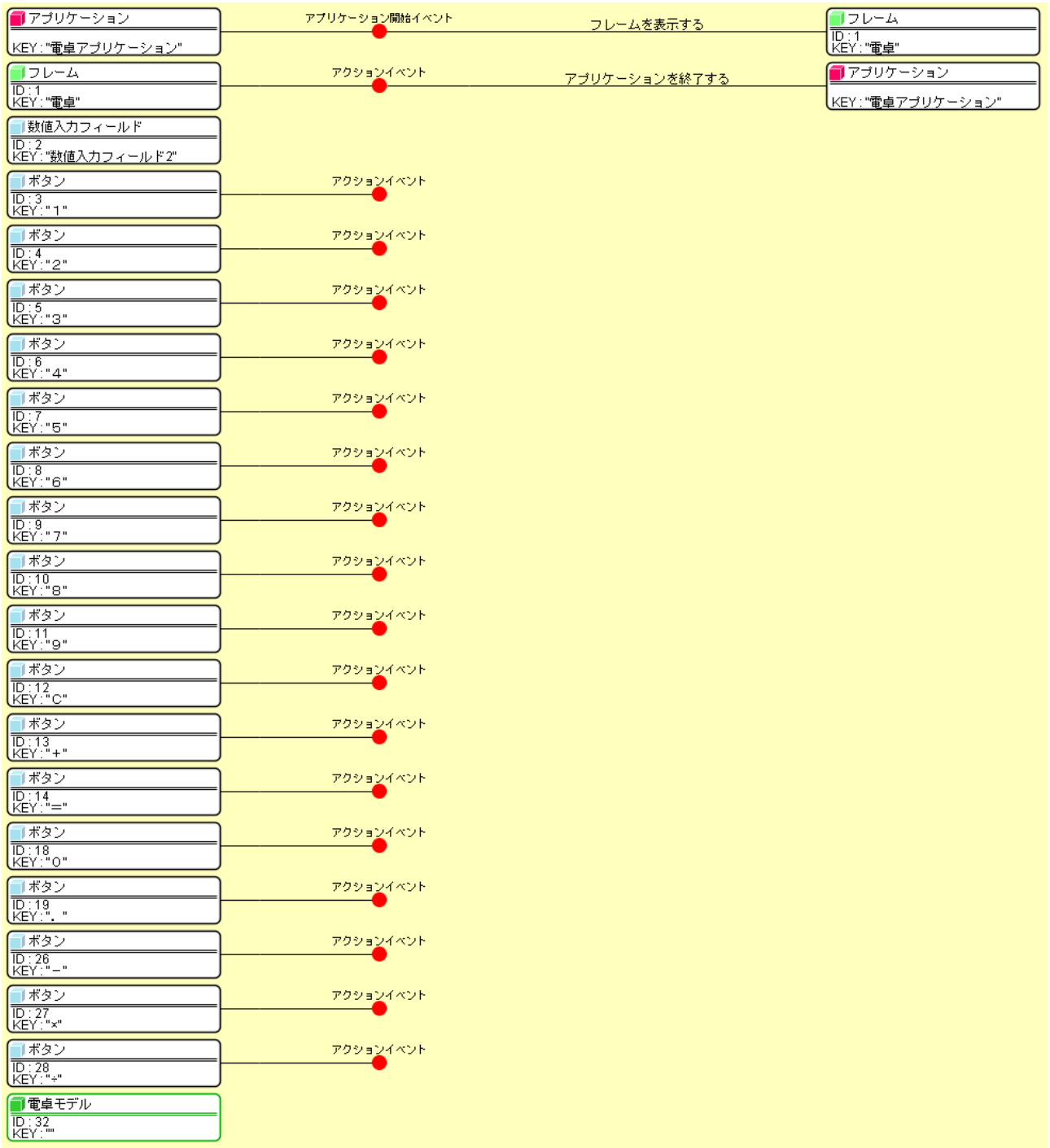
① コンポーネントを削除します。
［加算（＋）(ID:15)］の上で右クリックー［削除］をクリックします。

② ［コンポーネントを削除します。よろしいですか？］のメッセージが表示されるので
［はい］をクリックします。

③ ①～②を繰り返して
以下の１２個のコンポーネントを削除します。
１）［加算（＋）(ID:15)］(①～②で削除済み)
２）［文字列格納変数 (ID:16)］
３）［任意精度実数(BigDecimal)格納変数(ID:17)］
４）［整数(BigInteger)格納変数(ID:20)］
５）［入力演算子格納変数(ID:21)］
６）［内部演算子格納変数(ID:22)］
７）［演算実行サブルーチン(ID:23)］
８）［初期化サブルーチン(ID:24)］
９）［内部演算子 Null 判定(ID:25)］
１０）［減算（－）(ID:29)］
１１）［乗算（×）(ID:30)］
１２）［除算（÷）(ID:31)］

確認

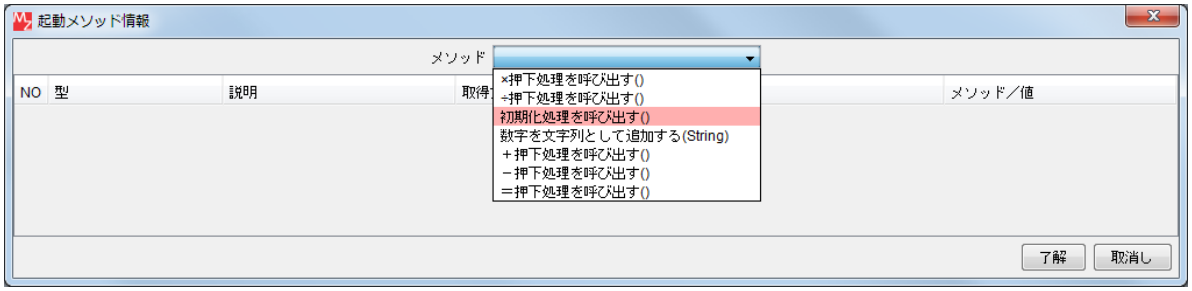
以下のようになります。


アプリケーション
KEY："電卓アプリケーション"
アプリケーション開始イベント
フレームを表示する
フレーム
ID：1
KEY："電卓"
フレーム
ID：1
KEY："電卓"
アクションイベント
アプリケーションを終了する
アプリケーション
KEY："電卓アプリケーション"
数値入力フィールド
ID：2
KEY："数値入力フィールド2"
ボタン
ID：3
KEY："1"
アクションイベント
ボタン
ID：4
KEY："2"
アクションイベント
ボタン
ID：5
KEY："3"
アクションイベント
ボタン
ID：6
KEY："4"
アクションイベント
ボタン
ID：7
KEY："5"
アクションイベント
ボタン
ID：8
KEY："6"
アクションイベント
ボタン
ID：9
KEY："7"
アクションイベント
ボタン
ID：10
KEY："8"
アクションイベント
ボタン
ID：11
KEY："9"
アクションイベント
ボタン
ID：12
KEY："C"
アクションイベント
ボタン
ID：13
KEY："+"
アクションイベント
ボタン
ID：14
KEY："="
アクションイベント
ボタン
ID：18
KEY："0"
アクションイベント
ボタン
ID：19
KEY："."
アクションイベント
ボタン
ID：26
KEY："−"
アクションイベント
ボタン
ID：27
KEY："×"
アクションイベント
ボタン
ID：28
KEY："÷"
アクションイベント
電卓モデル
ID：32
KEY：""

操 作 起動メソッドに複合コンポーネントの公開メソッドを追加しましょう。

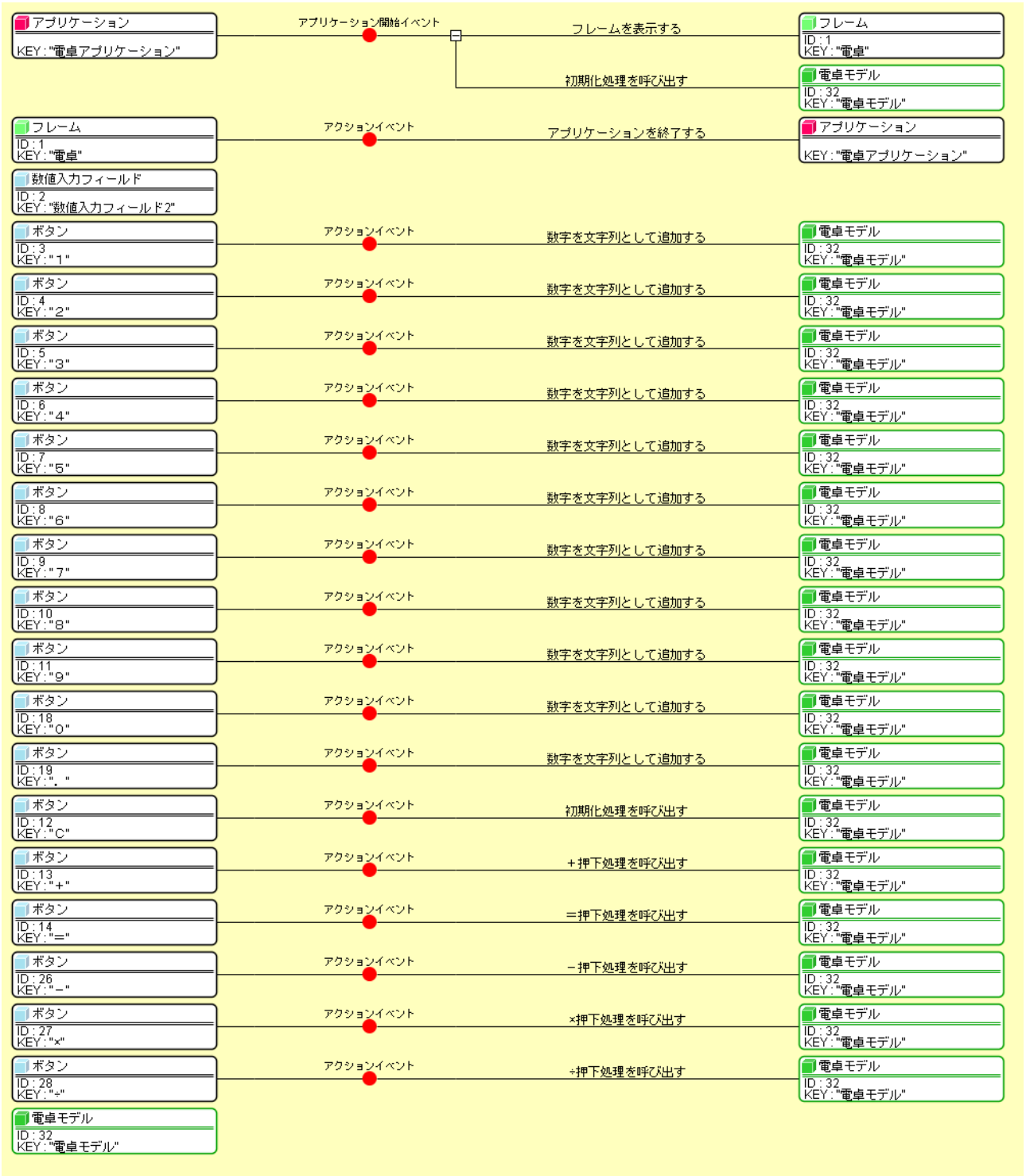
① イベントの接続先コンポーネントを選びます。
左側の［アプリケーション］コンポーネントの［アプリケーション開始イベント］上で
右クリック－［起動メソッド追加］とクリックします。薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［電卓モデル(ID:32)］をクリックします。

② 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［初期化処理を呼び出す()］をクリックします。
設定後、了解ボタンをクリックします。



③ ①～②の操作を繰り返してすべてのコンポーネントを接続します。
接続先は次ページの画面図を参考にしてください。
数字ボタンは引数（固定値、数字）も設定します。

以下のようになります。



※ボタンの順序を整理した後の様子です。

７）電卓モデルからの発生イベントを数値入力フィールドに関連付ける

［電卓モデル］からの発生イベントを［数値入力フィールド］に関連付けます。
［電卓モデル］内部での作業により、［電卓モデル］からはデータ設定イベントが発生することになっています。このイベント処理を追加して、そのイベント内包データを［数値入力フィールド］に設定します。

接続確認 コンポーネント同士の接続を確認します。

［電卓モデル］からの発生イベントを［数値入力フィールド］に関連付ける

接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■電卓モデル（ID:32）
発生イベント	データ設定イベント
接続先コンポーネント	■数値入力フィールド（ID:2）
起動メソッド	表示したい文字列を設定する(String)
＜引数＞	説明：文字列 取得方法：イベント内包 メソッド／値：イベント対象データ

操 作 ［電卓モデル］からの発生イベントを［数値入力フィールド］に関連付けましょう。

① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［電卓モデル(ID:32)］コンポーネント上で
右クリック－［イベント処理追加］－［データ設定イベント］とクリックします。

② イベントの接続先コンポーネントを選びます。
左側の［電卓モデル(ID:32)］コンポーネントの
［データ設定イベント］上で右クリック－［起動メソッド追加］とクリックします。
薄灰色の四角い枠が追加されます。
右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
右側に追加された薄灰色の四角い枠の上で右クリック－［接続コンポーネント選択］－
［数値入力フィールド（ID:2)］をクリックします。

③　接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。

起動メソッド設定画面が表示されます。

起動メソッド（処理）を選びます。

［メソッド］の ▼ をクリックします。

［表示したい文字列を設定する(String)］をクリックします。

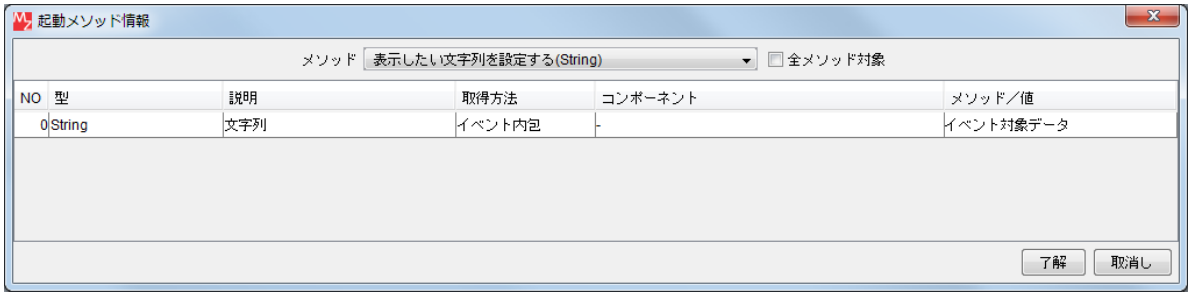
引数を設定します。

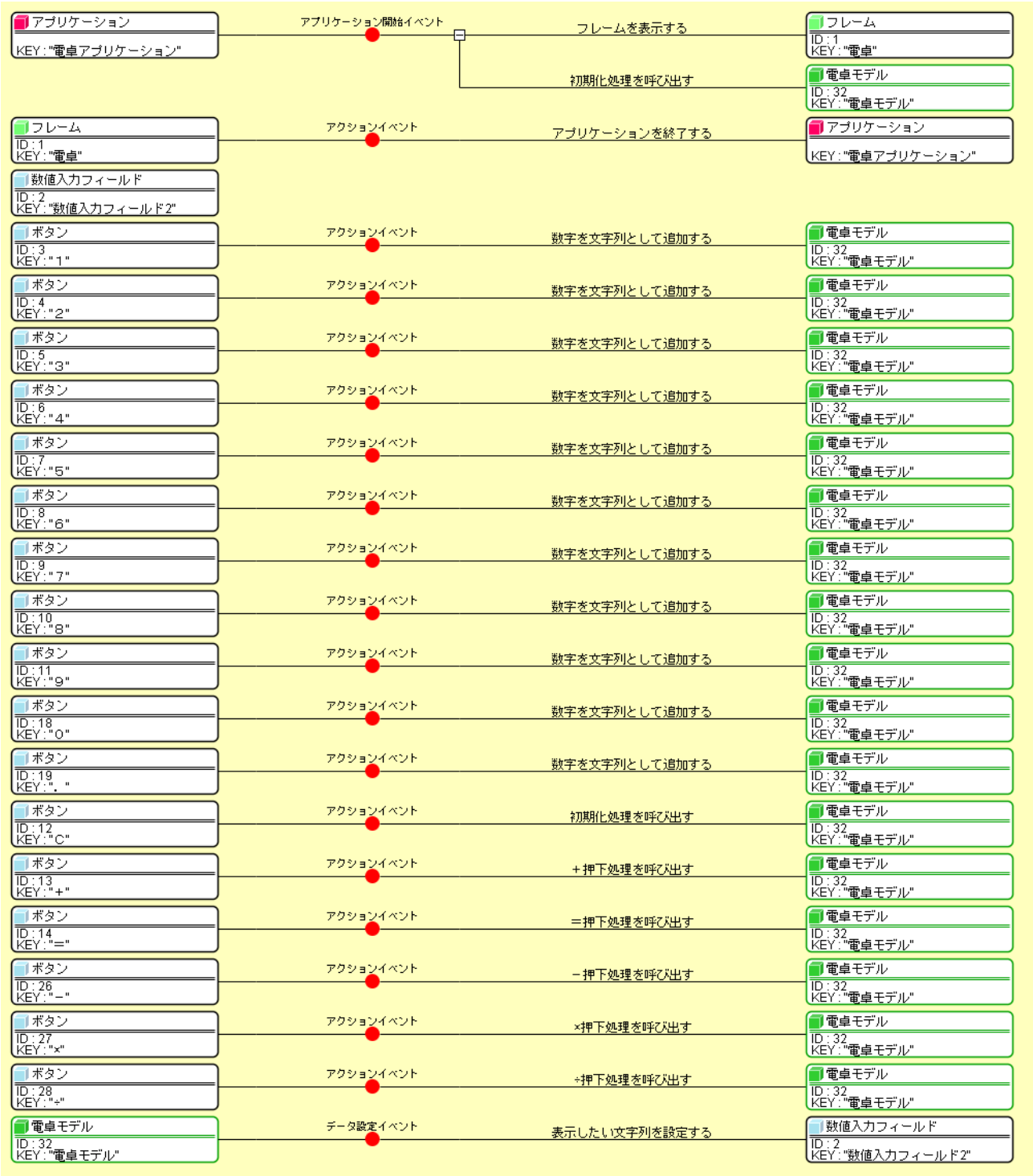
説明：文字列

取得方法：イベント内包

メソッド／値：イベント対象データ

設定後、了解ボタンをクリックします。



まとめ

ここまで進めるとビルダー上では以下のようになります。