基礎編：MZ Platformを使ってみよう

基礎編では MZ Platform を実際に操作しながら、アプリケーションを構築します。MZ Platform が提供する「アプリケーションビルダー」の基本的な操作方法を習得しましょう。

◆目次

Lesson.3 MZ Platformにさわろう

実際に MZ Platform を使ってみましょう。

Step.1 MZ Platformの種類

MZ Platform には次の種類があります。

MZ Platform の種類	役　割
アプリケーションビルダー	アプリケーションを構築する時に使用します。 アプリケーションを実行することができます。 アプリケーションを修正することができます。
アプリケーションローダー	アプリケーションを実行することができます。

Step.2 アプリケーションビルダーを起動し初期画面を確認する

1）アプリケーションビルダーの起動

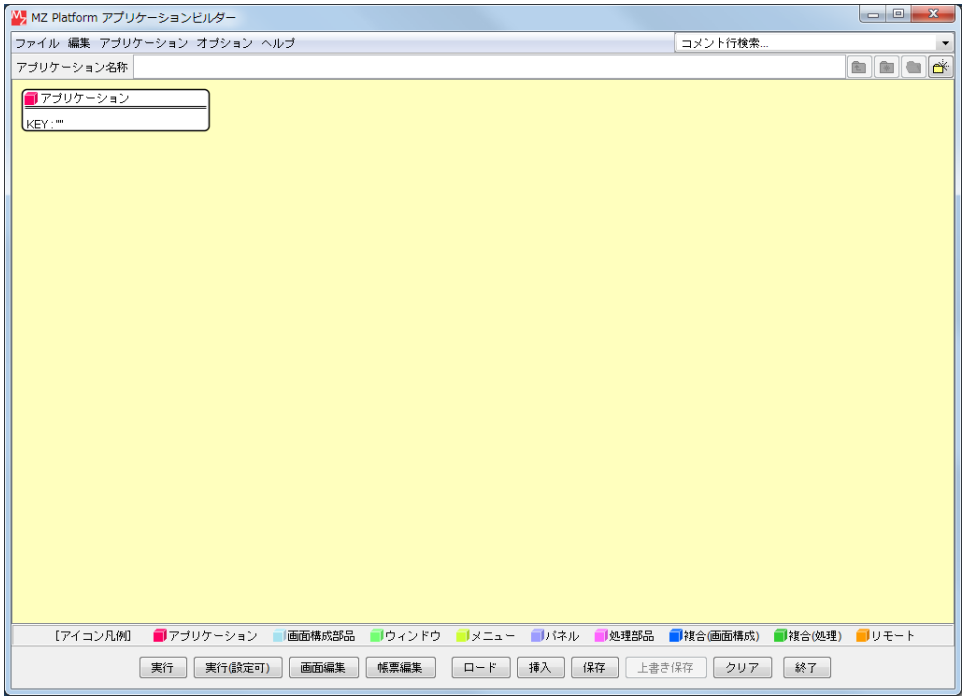
アプリケーションビルダー（以下ビルダー）の起動手順を確認しましょう。

操 作 アプリケーションビルダーを起動しましょう。

① ［スタート］－［すべてのプログラム］－［MZ Platform 2.9］－［アプリケーションビルダー］とクリックします。

確認

アプリケーションビルダーが起動します。



２）アプリケーションビルダーの起動画面

アプリケーションビルダーの起動画面を確認します。

記号	名　称	役　割
a	メニューバー	機能が登録されています。 ｆ［ツールボタン］と同じことができます。
b	アプリケーション名称	作成するアプリケーションに名前を付けられます。
c	コンポーネント	複数のコンポーネントを組み合わせてアプリケーションを構築します。（コンポーネントはその都度追加します）
d	作業領域	コンポーネントの追加、接続、命令の指示をします。
e	アイコン凡例	コンポーネントの種類を表します。
f	ツールボタン	機能が登録されています。 ａ［メニューバー］と同じことができます。
g	コメント行検索領域	アプリケーション内のコメント行を検索しその位置に移動する際に使います。
h	編集サポートボタン	複合コンポーネントを使用し別の階層に移動するときに使います。

知っていると便利！

１．起動するには、MZ Platform が正しく導入されていることが前提です。
２．起動しない場合、MZ Platform が正しく設定されていないことが考えられます。
　　設定を確認してください。

Step.3 アプリケーションビルダーでアプリケーションを使用する

サンプルアプリケーションを使用してビルダーの動きを確認しましょう。

1）既存ファイルの読み込み（[ロード] ボタン）

既存ファイルを読み込みましょう。

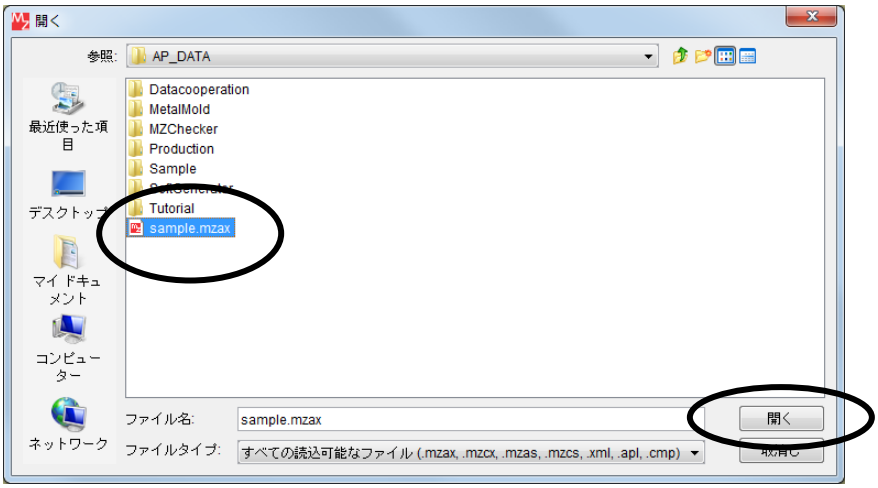
準 備 ここでは以下のボタンを使用します。

ロード

操 作 既存ファイルを読み込みます。

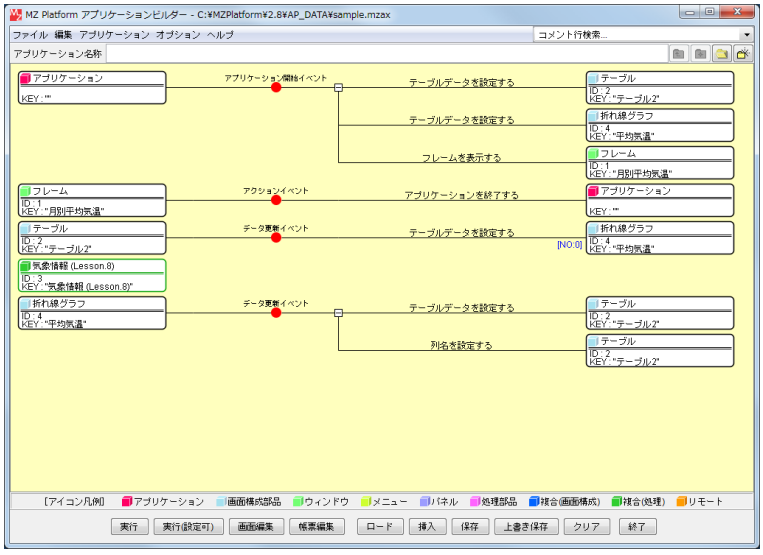
① ロード をクリックします。

② 「sample.mzax」をクリックし、開く をクリックします。



確認

ビルダーに「sample.mzax」ファイルが読み込まれます。



2）アプリケーションの実行（[実行] ボタン）

読み込んだファイルを実行しましょう。

準 備

ここでは以下のボタンを使用します。

実行

操 作

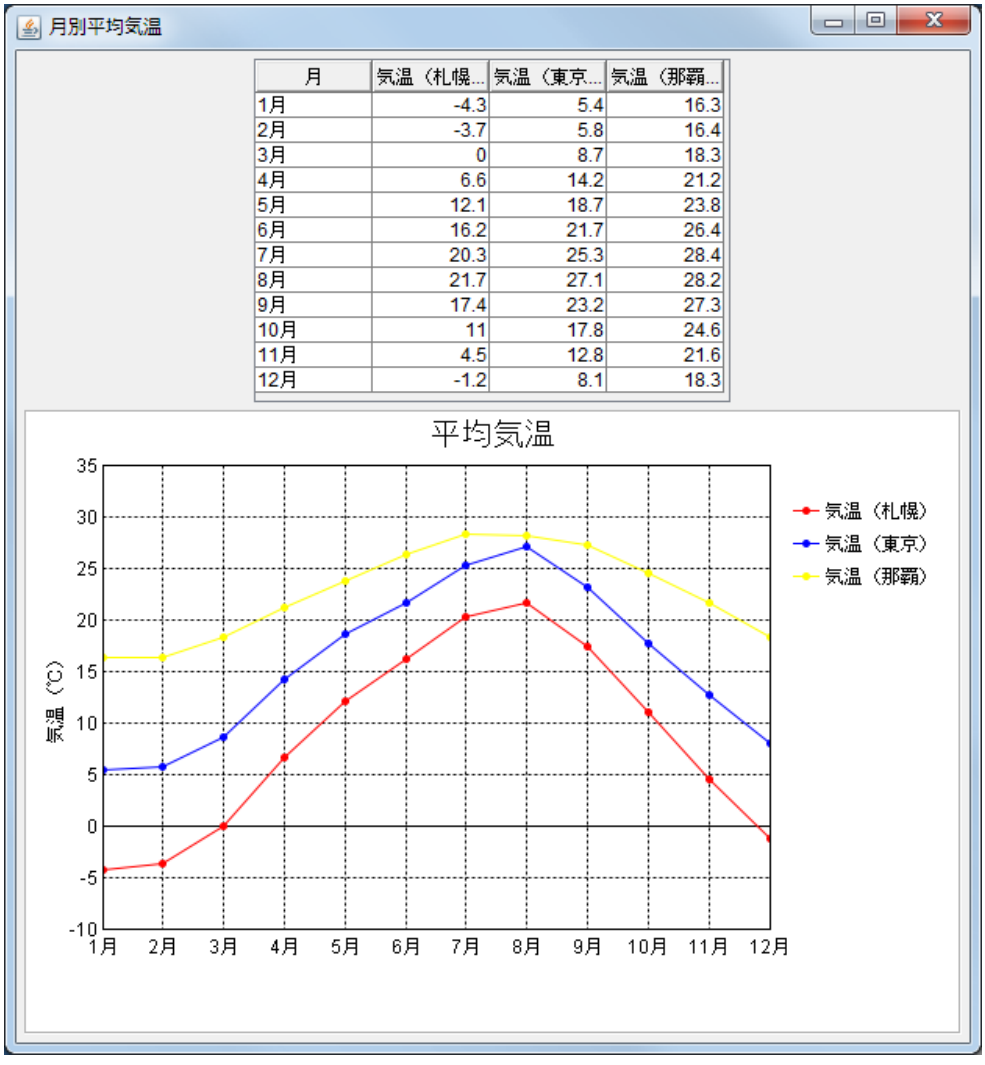
アプリケーションを実行します。

① 実行 をクリックします。

② アプリケーションが実行します。

確認

ここでは、各地の月別平均気温の表とグラフが表示されます。



３）データの変更

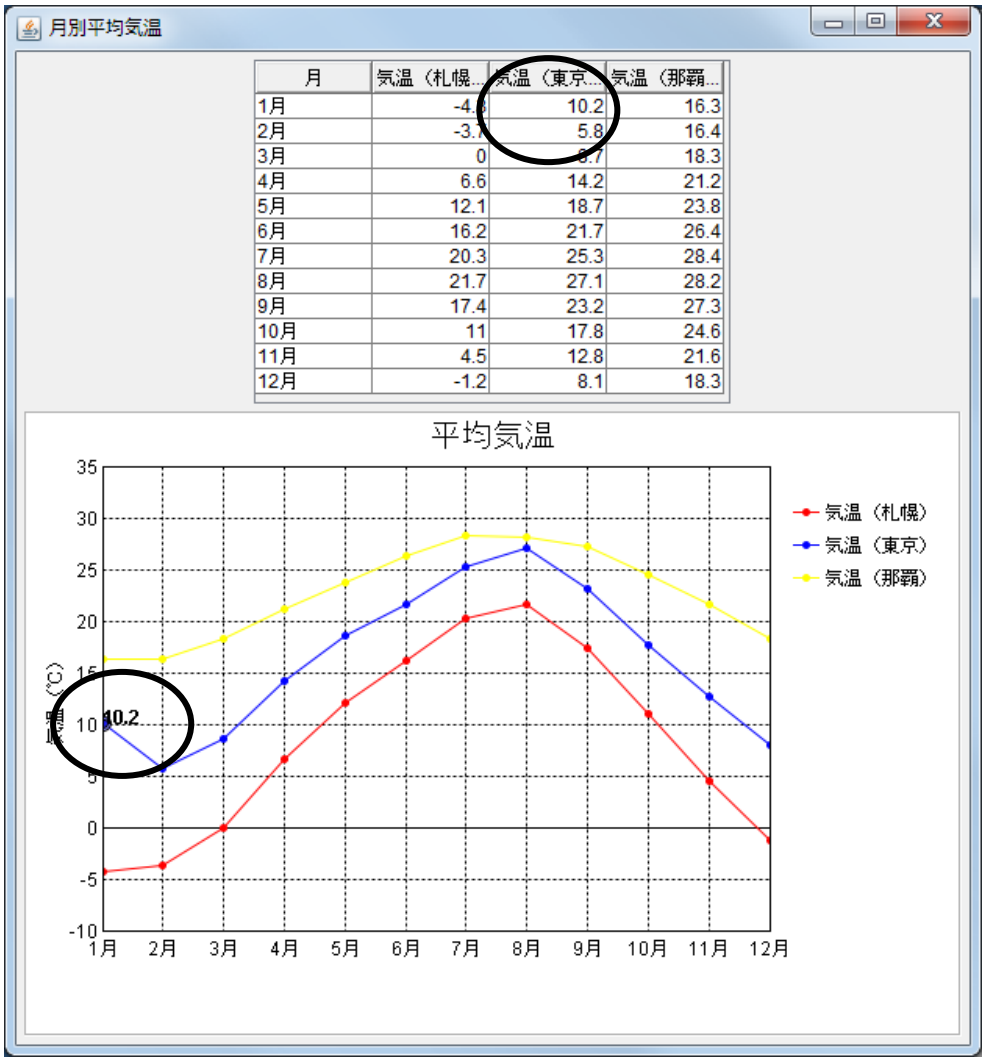
実行しているアプリケーションのデータを変更できます。

操 作 グラフの中の気温データを変更しましょう。

① 「気温（東京)」の「１月」のデータを「10.2」までドラッグします。

確認

グラフデータが変更され、表データも連動して変更されます。



知っていると便利！

データの変更は、表からもできます。

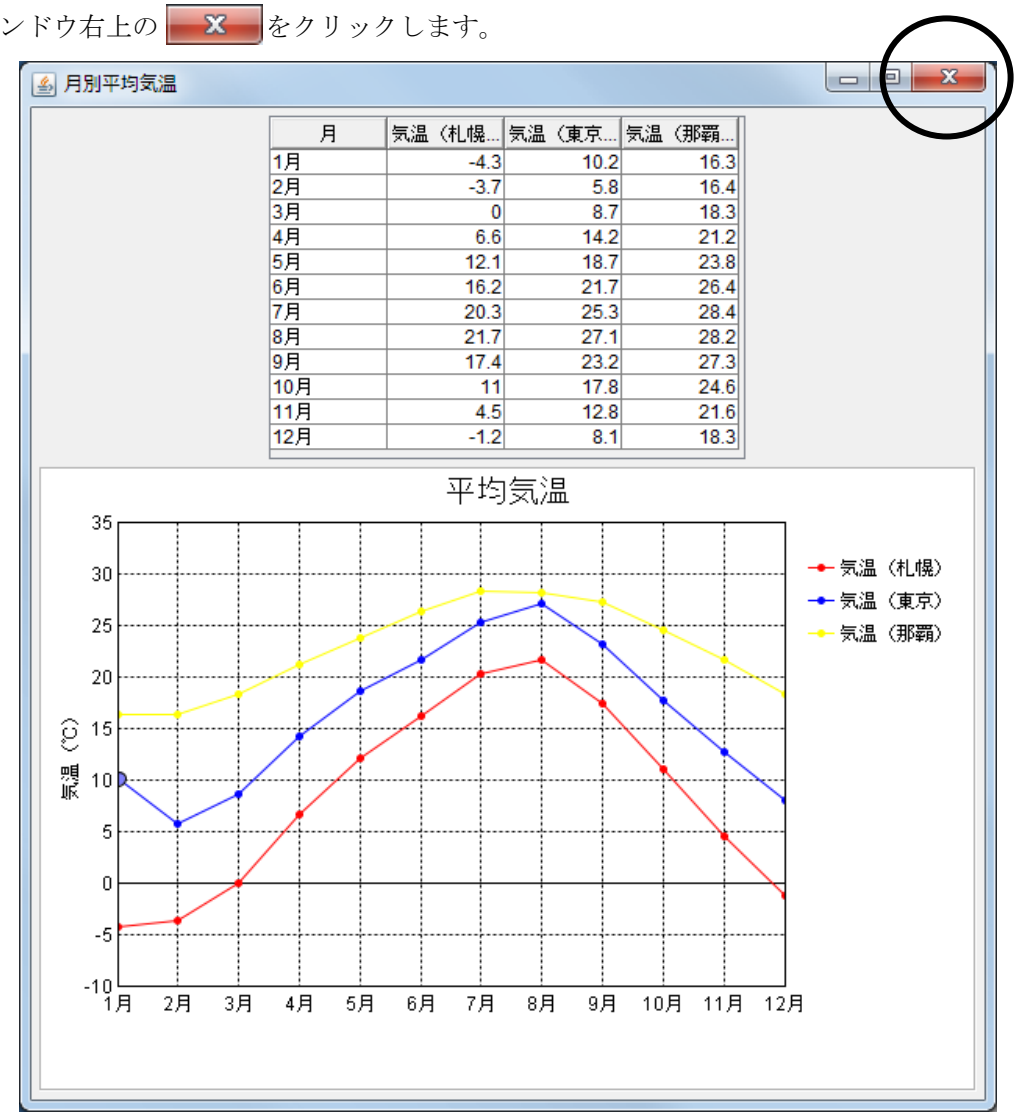
変更したい表中のデータをダブルクリックして変更したい値を上書きします。

４）アプリケーションの終了

実行したアプリケーションを終了しましょう。

操 作 アプリケーションを終了します。

① ウィンドウ右上の [X] をクリックします。



② 表示されていたウィンドウが閉じてアプリケーションが終了します。

５）ビルダーの終了

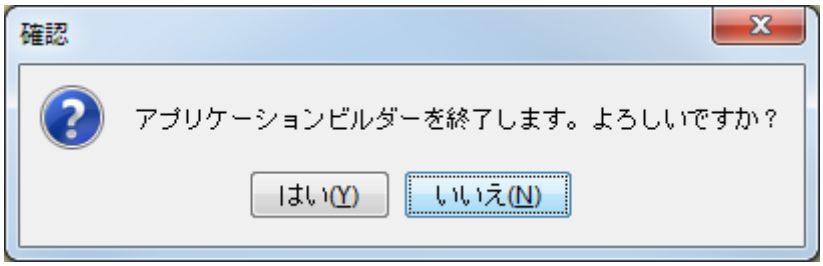
ビルダーを終了します。

準 備 ここでは以下のボタンを使用します。

終了

操 作 ビルダーを終了します。

① 終了 をクリックします。
② 確認画面が表示されるので はい(Y) をクリックします。



③ ビルダーが終了します。

知っていると便利！

ビルダーを終了せずに画面を初期状態にするには クリア をクリックします。

Step.4 アプリケーションローダーでアプリケーションを実行する

サンプルアプリケーションを使用してアプリケーションローダー（以下ローダー）の動きを確認しましょう。ビルダーとの違いを確認しましょう。

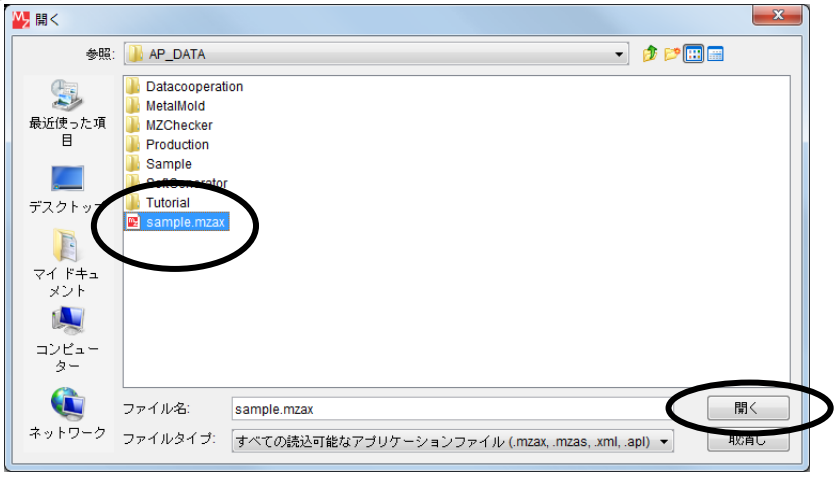
1）アプリケーションローダーの起動

ローダーの起動手順を確認しましょう。

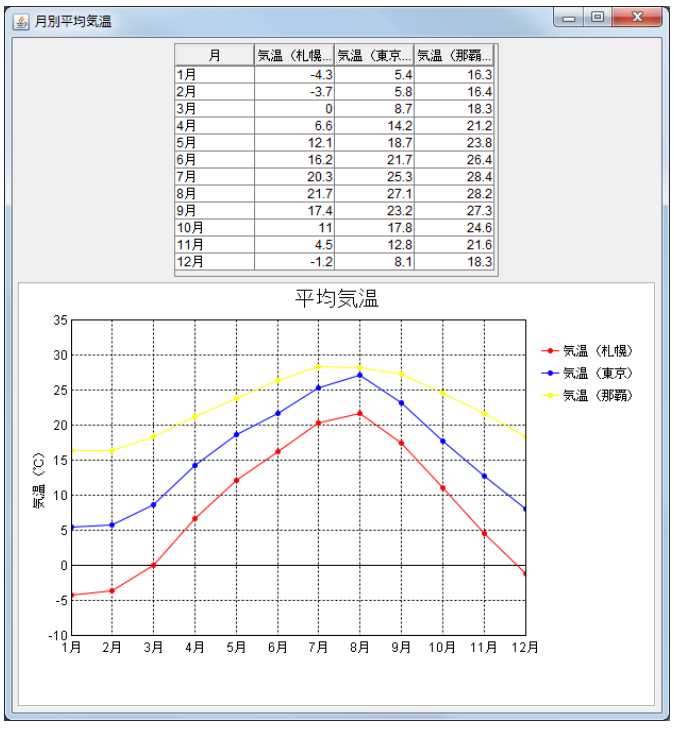
操 作 アプリケーションローダーを起動しましょう。

① ［スタート］―［すべてのプログラム］－［MZ Platform 2.9］－［アプリケーションローダー］と順にクリックします。

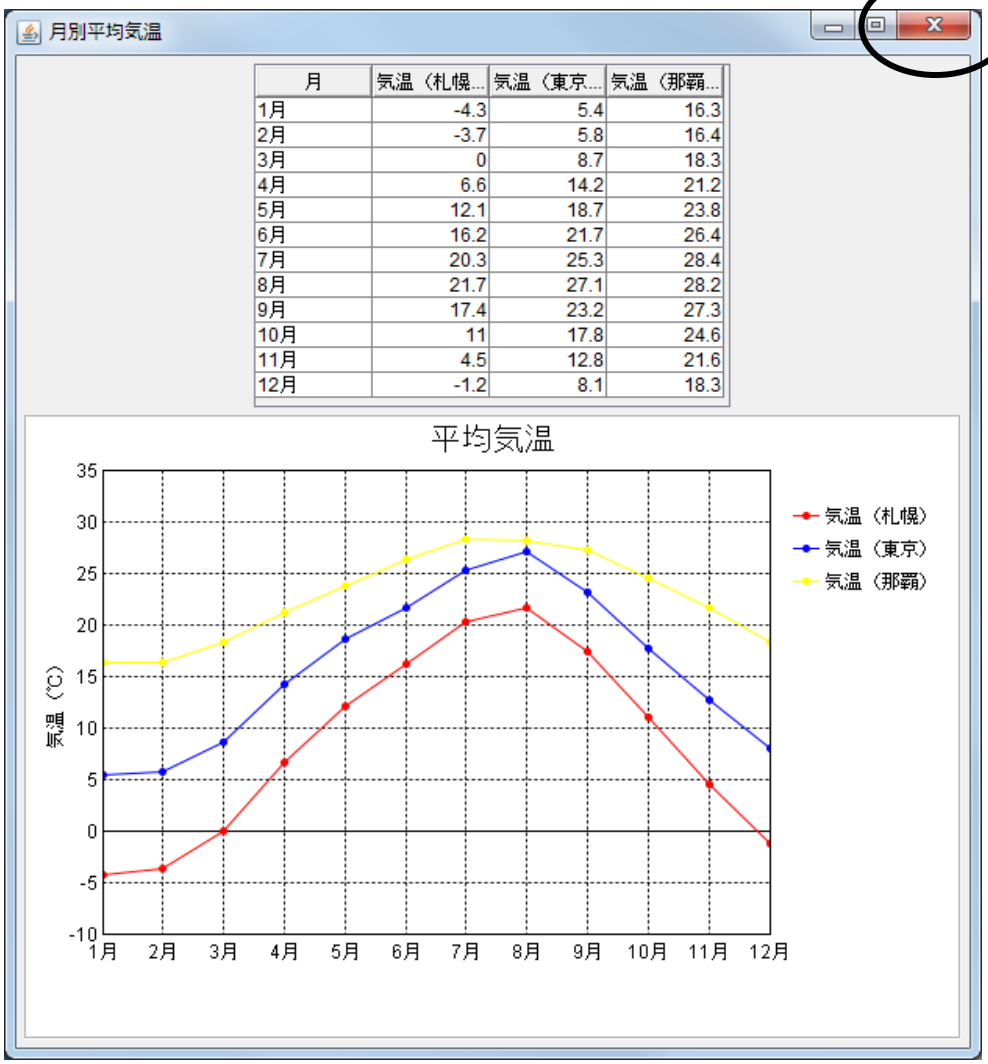
② ［(ファイルを) 開く］画面が表示されますので、「sample.mzax」をクリックし、
開く をクリックします。



確認 アプリケーションが実行されます。



③　アプリケーションを終了しておきましょう。



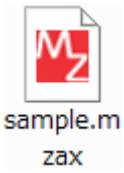
月	気温（札幌...	気温（東京...	気温（那覇...
1月	-4.3	5.4	16.3
2月	-3.7	5.8	16.4
3月	0	8.7	18.3
4月	6.6	14.2	21.2
5月	12.1	18.7	23.8
6月	16.2	21.7	26.4
7月	20.3	25.3	28.4
8月	21.7	27.1	28.2
9月	17.4	23.2	27.3
10月	11	17.8	24.6
11月	4.5	12.8	21.6
12月	-1.2	8.1	18.3

知っていると便利！

アプリケーションビルダー終了時およびアプリケーション実行時には自動でバックアップファイルが作られるようになっています。バックアップファイル「ApplicationBackup.mzas」は MZ Platform インストールフォルダ¥log 内（例：C:¥MZPlatform¥2.9¥log）に作成されます。

バックアップファイルを呼び出すにはアプリケーションビルダーを起動し、 ロード ボタンを押します。
［(ファイルを）開く］画面で log フォルダに移動し「ApplicationBackup.mzas」をクリックし、
開く をクリックします。
ファイルの内容を確認したら別名で保存してください。

MZPlatform のアプリケーションファイルには MZPlatform 独自のアイコンが付きます。
このアイコンをダブルクリックすると、自動的にアプリケーションローダーで
実行されるようになっています。



また、コマンドプロンプトの使い方をご存知の方は、次の方法でも実行できます。
バッチファイルを作成して以下のコマンドを実行すれば、ファイル選択画面を表示しなくても、
直接アプリケーションを実行することができます。

```
CD C:¥MZPlatform¥2.9          (MZPlatform のインストールフォルダーへ移動)
PFLoader  〈Application Data Filename〉
```

※PFLoader は C:¥MZPlatform¥2.9¥PFLoader.exe です。

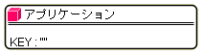
Lesson.4 画面を作ってみよう

ビルダーを利用して、簡単なアプリケーションを作成してみましょう。
実際に操作しながら、ビルダーの基本操作を覚えましょう。

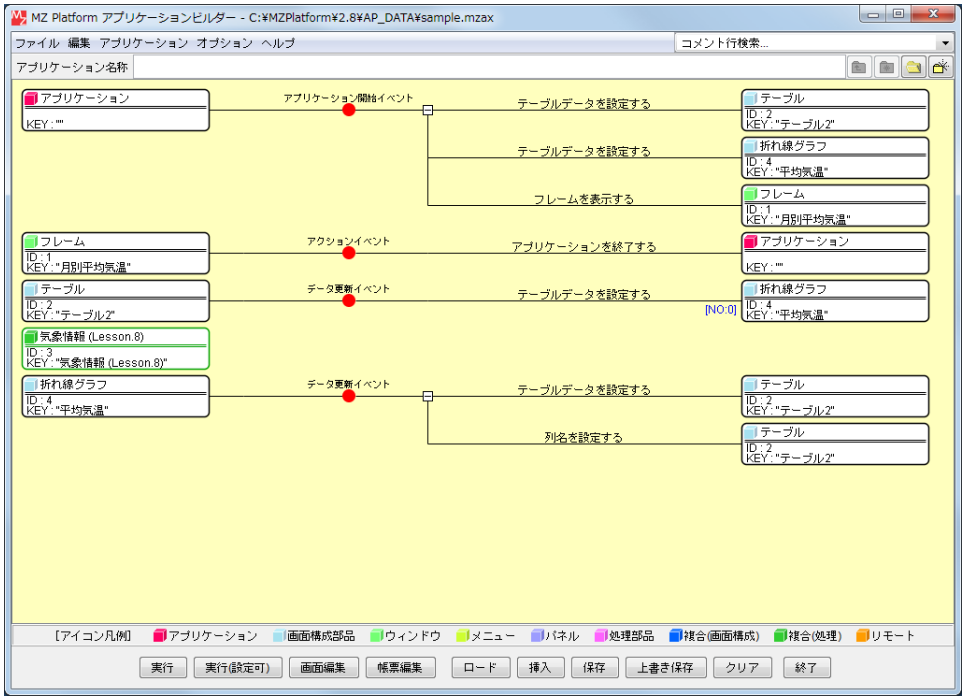
Step.1 アプリケーション設計図（作業画面）の見方

アプリケーションを構築するために必要な設計図（作業画面）の見方を覚えましょう。
ここでは「sample.mzax」ファイルを例に紹介します。

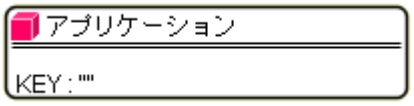
1）コンポーネント

画面上に表示されている白い矩形 

を「コンポーネント」と呼びます。
これらを線で結んでアプリケーションを構築していきます。



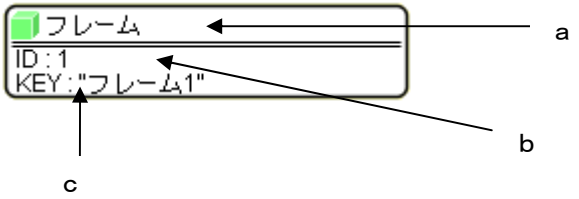
2）特別なコンポーネント



［アプリケーション］コンポーネントは、アプリケーション自身を表す特別なコンポーネントです。
追加操作をしなくても、起動した時点で表示されます。
削除することはできません。

３）コンポーネントの名称

コンポーネントの名称と役割を覚えましょう。



a	コンポーネント名	コンポーネントの名前です。 コンポーネントが持っている機能の情報です。
b	コンポーネント ID	コンポーネントの識別番号を示します。 この番号はアプリケーションの中で重なることはありません。
c	コンポーネントキー	コンポーネントの識別情報を示します。 このコンポーネントを識別するための説明や名前で、 アプリケーション構築時に自由に設定できます。

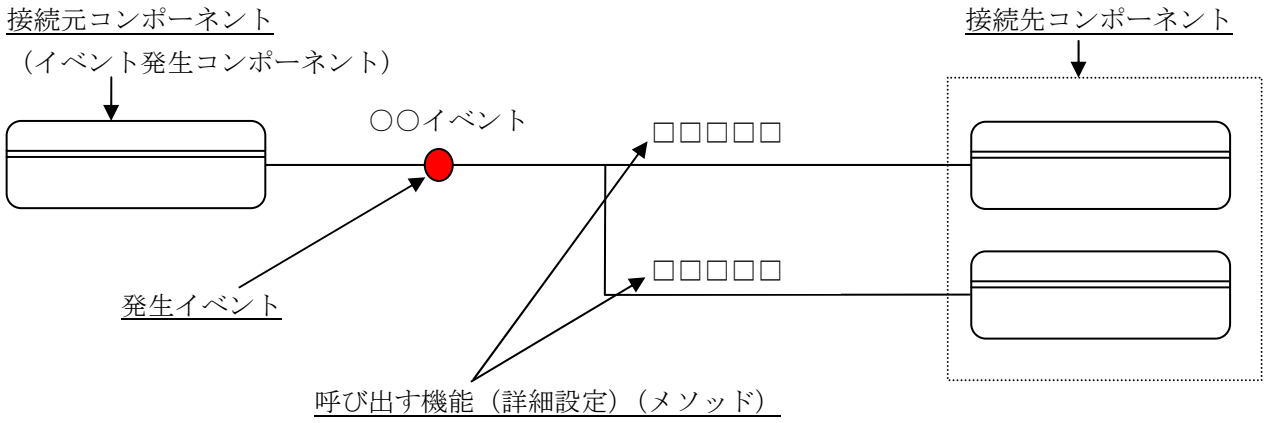
４）コンポーネント同士を結ぶ（接続する）

アプリケーションを構築するには、コンポーネント同士を結びながら組み立てていきます。

コンポーネント間を結ぶには、左側にあるコンポーネントから右側に追加したコンポーネントに結びます。

直接結ぶのではなく、途中に「イベント」が発生し接続のきっかけを作ります。

「イベント」で結んだ右側のコンポーネントには「メソッド」という詳細設定を行います。



イベント	コンポーネントの中で起きるデータや状態の変更をイベントといいます。 このイベントの発生が、コンポーネントから他のコンポーネントへ接続するきっかけになります。 コンポーネントから外部に接続する唯一の方法です。 このような処理形式を『イベント駆動処理』といいます。
メソッド	コンポーネントが持っている機能を呼び出すには、コンポーネントが提供するメソッドを使用します。 メソッドはコンポーネントに対して処理を指示する唯一の方法です。 コンポーネントはその機能を外から使用できるようにするために様々なメソッドを提供します。

MZ Platformにおけるコンポーネントの動作とメソッド起動（**イベント駆動処理**）の詳細については、「*付録 MZ Platformにおけるコンポーネント動作とメソッド起動*」をご覧ください。

Step.2 作成手順

ビルダーを使ったアプリケーションの構築手順は次のとおりです。

１）必要なコンポーネントを追加

ビルダーで提供している標準コンポーネントはたくさんあります。
必要なコンポーネントを画面上に追加しながら作業をします。
追加手順は統一されています。

手 順

① 作業画面上で右クリックー［コンポーネント追加］（または［コンポーネント一括追加］）

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する

必要なコンポーネントが画面上に用意できたら、コンポーネント同士を線で結びます。
線で結ぶにはコンポーネントから発生するイベントを表示し、使用するイベントを選択します。
イベントの種類や数はコンポーネントによって異なります。

手 順

① 左側のコンポーネント上で右クリックー［イベント処理追加］ー状況に応じたイベントを追加

３）イベントの接続先コンポーネントを選ぶ

選択したイベントの「接続先コンポーネント」を選びます。

手 順

① イベント上で右クリックー［起動メソッド追加］
② 右側に表示された空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］

４）接続したコンポーネントの処理を選ぶ

接続したコンポーネントで何をするか処理を選びます。

手 順

① 右側の接続先コンポーネントの上で右クリックー［起動メソッド選択］
② 接続先コンポーネントへ処理を指示

Step.3 ウィンドウの作成

作成手順を確認しながら簡単なアプリケーションを作成してみましょう。
アプリケーションのおおもととなる「ウィンドウ（フレーム)」を作成します。

完成図

ウィンドウ（フレーム）の完成図を確認しましょう。

※ウィンドウの中身は何も用意していないので、枠だけが表示されます。

考え方

1．アプリケーションが開始したら、フレームが表示される。
2．フレームが閉じたら、アプリケーションが終了する。

準 備

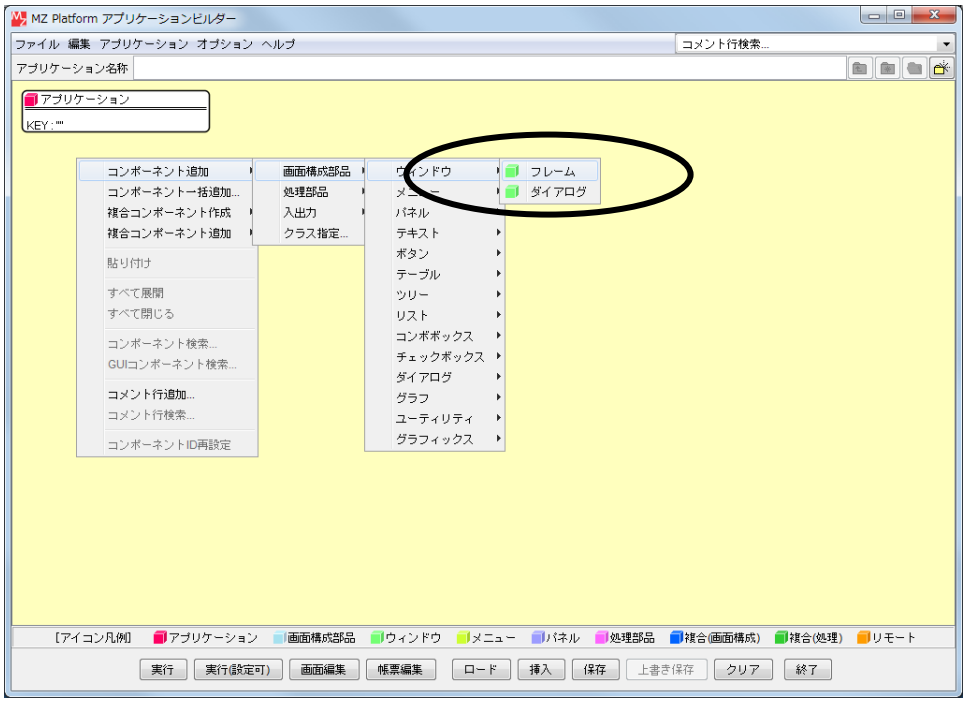
ここでは以下のコンポーネントを使用します。

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］

操 作

ウィンドウ（フレーム）を作成しましょう。

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－［コンポーネント追加］－［画面構成部品］－［ウィンドウ］－［フレーム］とクリックします。



［フレーム］コンポーネントが追加されます。

接続確認

コンポーネント同士の接続を確認します。

開始

項　目	内　容
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

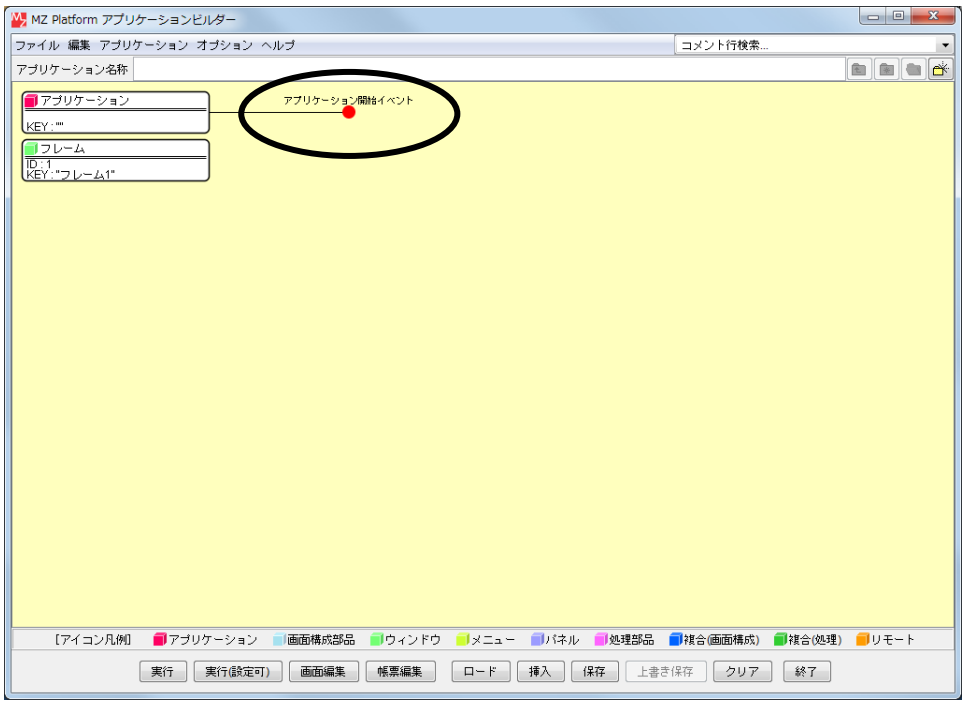
項　目	内　容
接続元コンポーネント （イベント発生コンポーネント）	■フレーム（ID:1）
発生イベント	アクションイベント
接続先コンポーネント	■アプリケーション
起動メソッド	アプリケーションを終了する()

操 作

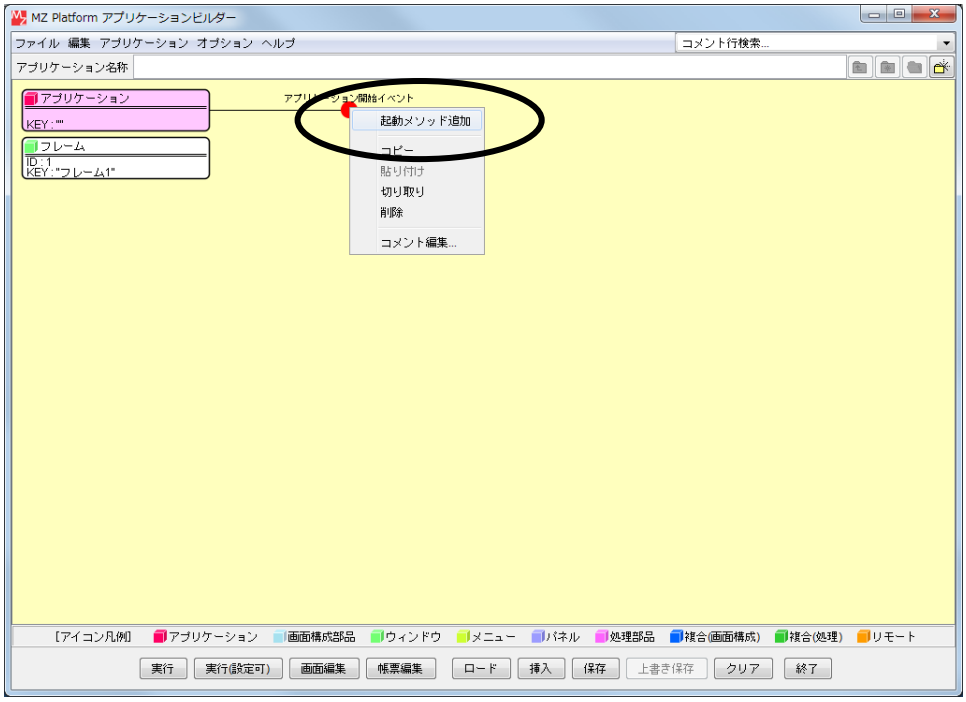
① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［アプリケーション］コンポーネント上で右クリックー［イベント処理追加］ー［アプリケーション開始イベント］とクリックします。

確認

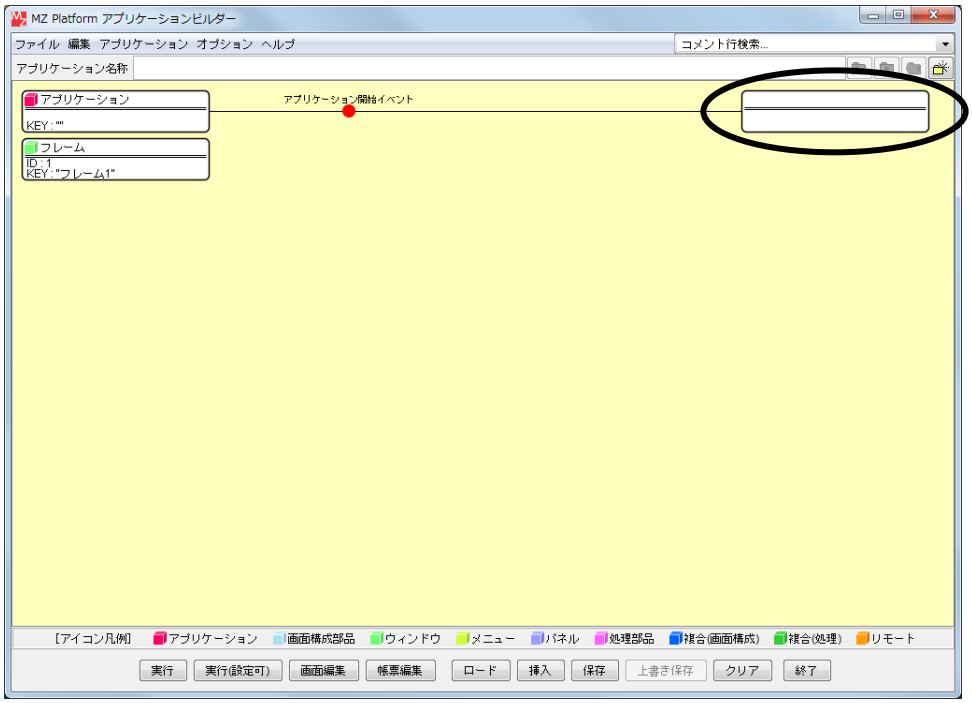
［アプリケーション開始イベント］が表示されます。



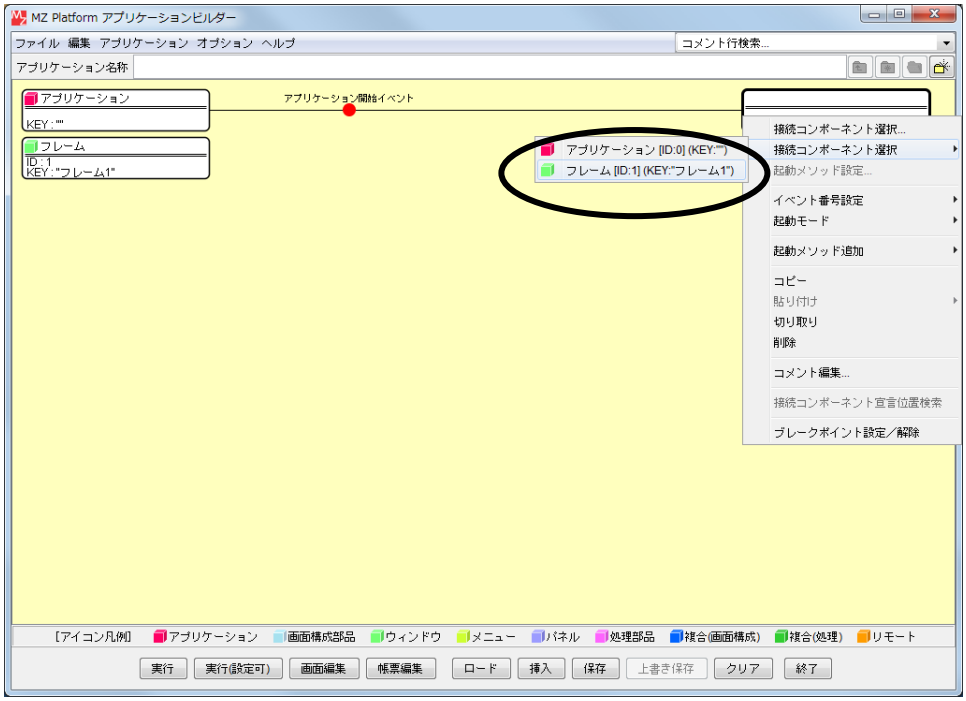
② イベントの接続先コンポーネントを選びます。
イベントの上で右クリック－［起動メソッド追加］とクリックします。



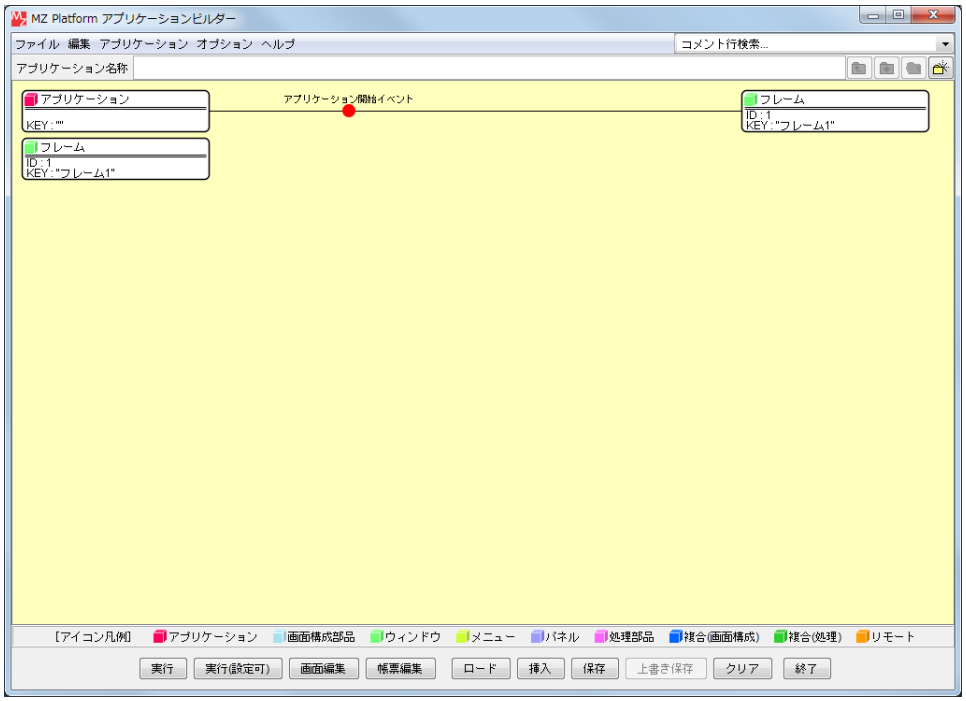
空の四角い枠が追加されます。



③　右側に表示された空の四角い枠にコンポーネントを割り当てます。
右側に表示された空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:1)］をクリックします。



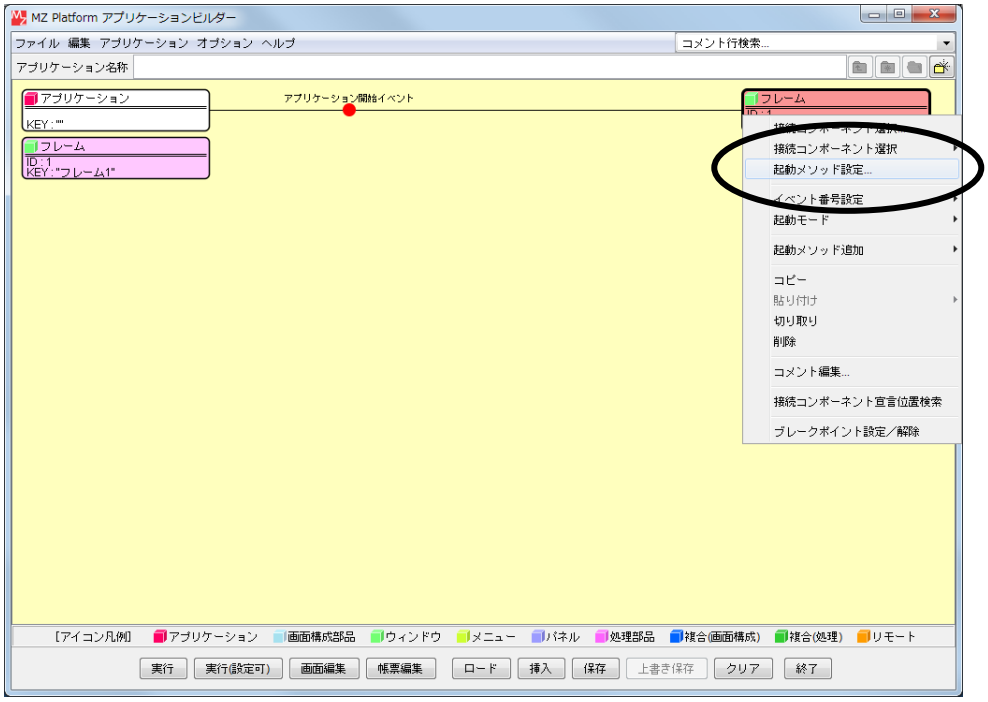
［アプリケーション］コンポーネントと［フレーム］コンポーネントが接続されます。



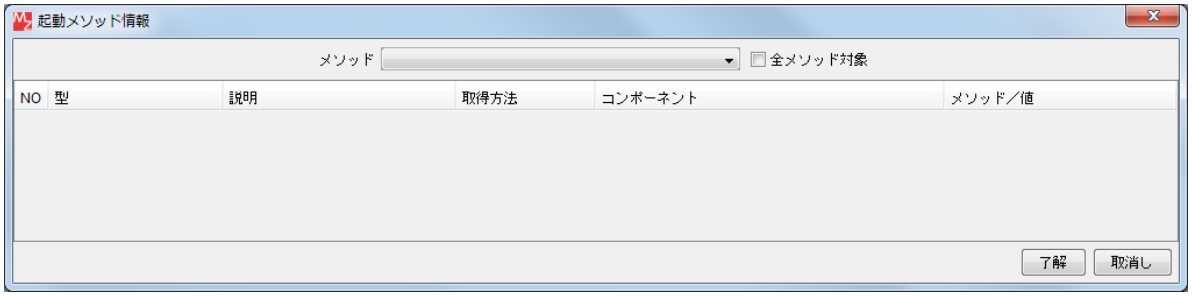
知っていると便利！

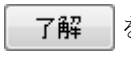
右側に表示された四角い枠にコンポーネントを割り当てる場合、
選択用の専用画面を使用することもできます。
右側に表示された空の四角い枠の上で右クリック－［接続コンポーネント選択...］を
クリックします。

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－［起動メソッド設定...］をクリックします。

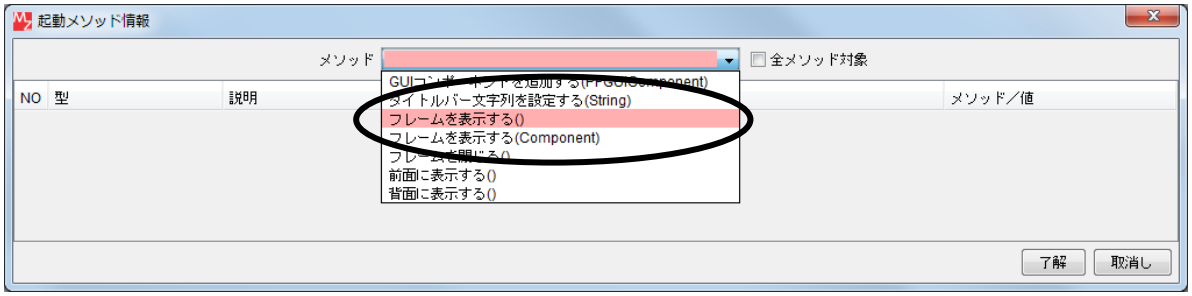


起動メソッド設定画面が表示されます。

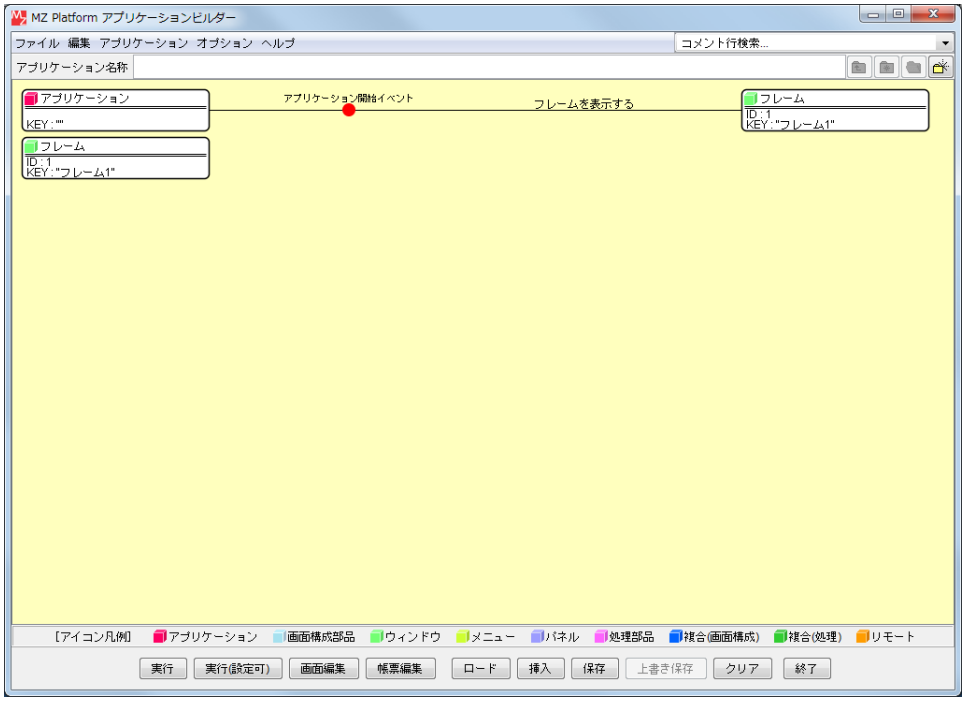


⑤　起動メソッド（処理）を選びます。
［メソッド］の  をクリックします。
［フレームを表示する()］をクリックします。


をクリックします。



起動メソッドが設定されます。



練 習

［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しょう。

項　目	内　容
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

作成したアプリケーションを実行しましょう。

① 実行 をクリックしてアプリケーションが完成したことを確認します。

Step.4 保存

作成したアプリケーションを保存します。

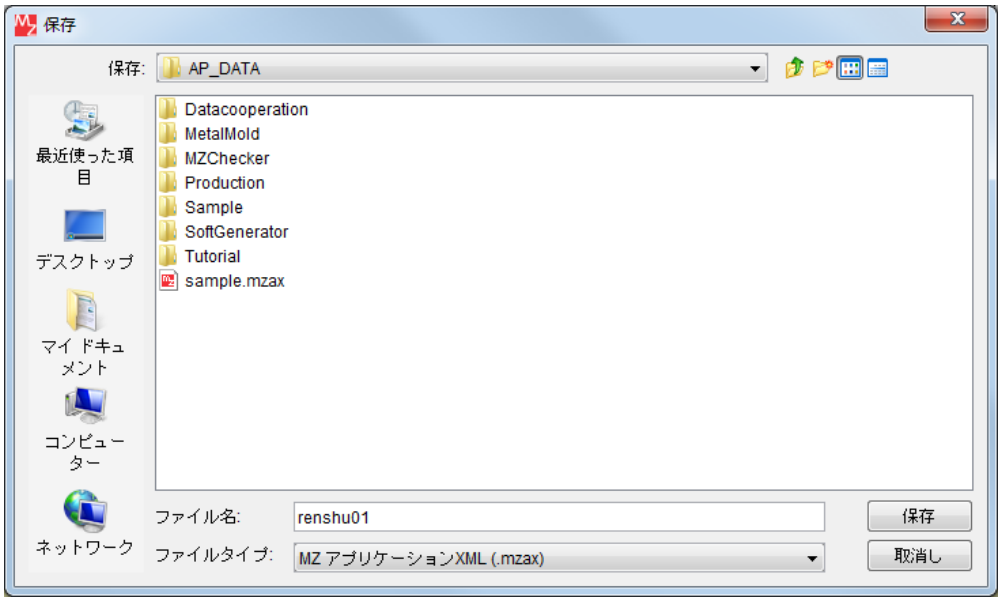
準 備 ここでは以下のボタンを使用します。

保存

操 作 作成したアプリケーション（ファイル）を保存します。

① 保存 をクリックします。

② ［保存］ダイアログボックスが表示されます。
ファイル名を入力して 保存 ボタンをクリックします。

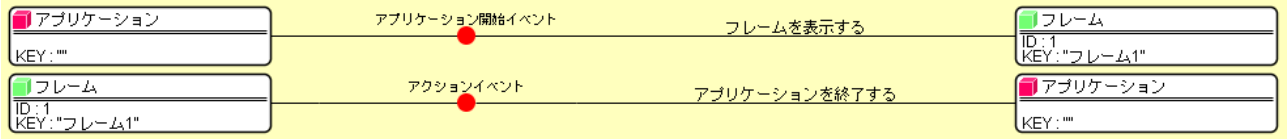


知っていると便利！

上書き保存 は、ファイルを 1 度保存しないと有効になりません。
保存をすると既定の設定では「.mzax」と「.mzas」の拡張子が付いた 2 種類のファイルができます。「.mzax」は XML 形式、「.mzas」はバイナリ形式のファイルです。
アプリケーションビルダーの[オプション]ー[バイナリデータ自動保存]のチェックマークを外すと、バイナリ形式のファイルは自動的に保存されなくなります。

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。



Step.5 閉じるボタンを作成する

フレームに閉じるボタンを作成しましょう。

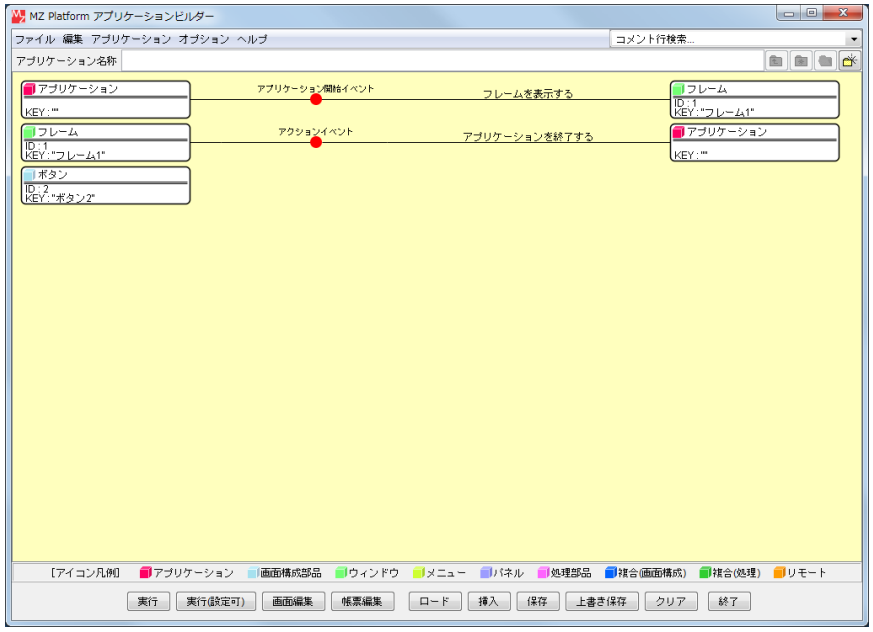
１）［実行］と［実行（設定可)］

ビルダーには［実行］と［実行（設定可)］の２種類の実行があります。
［実行］はビルダーで作成したアプリケーションを実行するだけですが、［実行（設定可)］はアプリケーション実行後、実行しているアプリケーションの上で編集ができる状態です。例えば、［ボタン］コンポーネントを表示してボタン名を変更したい場合などに使用します。

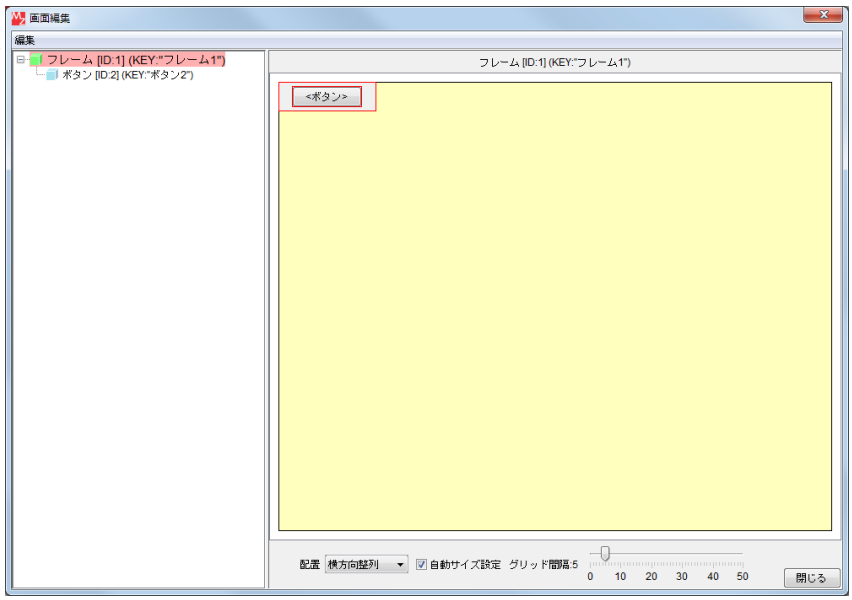
２）［画面編集］画面

起動画面は内部処理を記述します。画面設計は［画面編集］画面で行います。

起動画面
（内部処理を記述する）



画面編集画面
（画面設計をする）



３）閉じるボタンをフレームに表示

閉じるボタンをフレーム上に作成しましょう。

完成図

閉じるボタンが作成されている状態を確認しましょう。

準 備

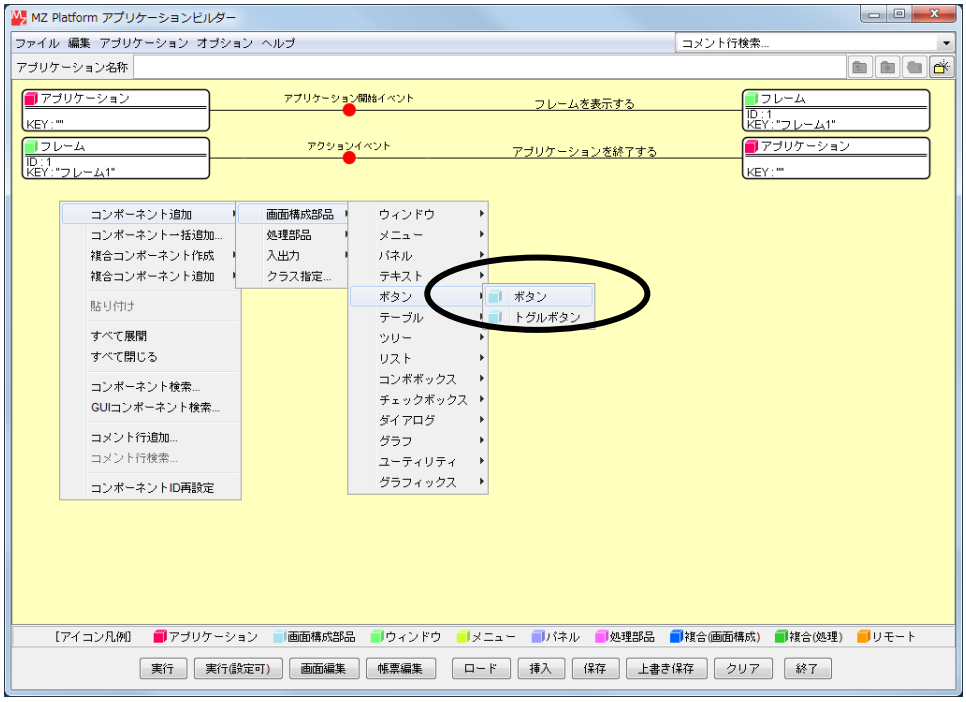
ここでは以下のコンポーネントを追加します。

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

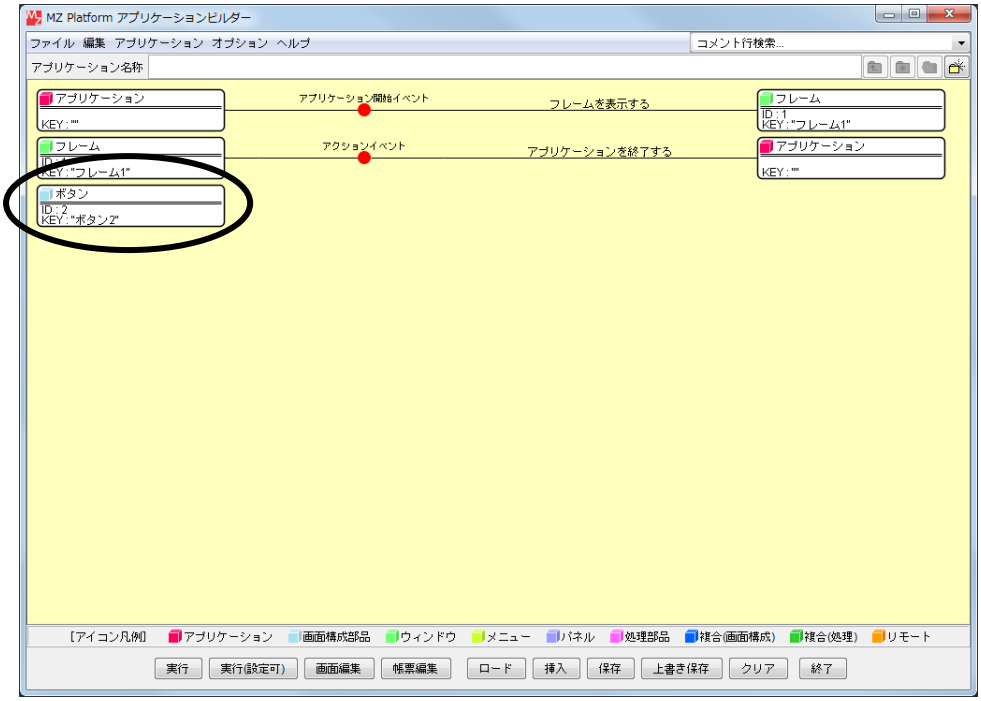
操 作

ウィンドウ（フレーム）を使用しボタンコンポーネントを表示しましょう。

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－［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。



［ボタン］コンポーネントが追加されます。



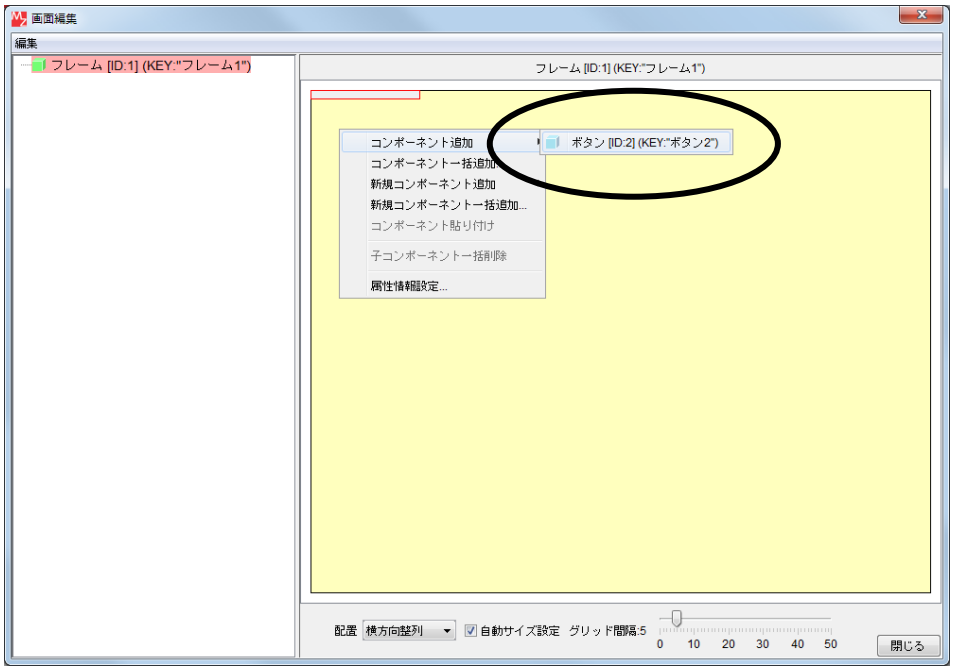
画面編集

① 画面を作成します。

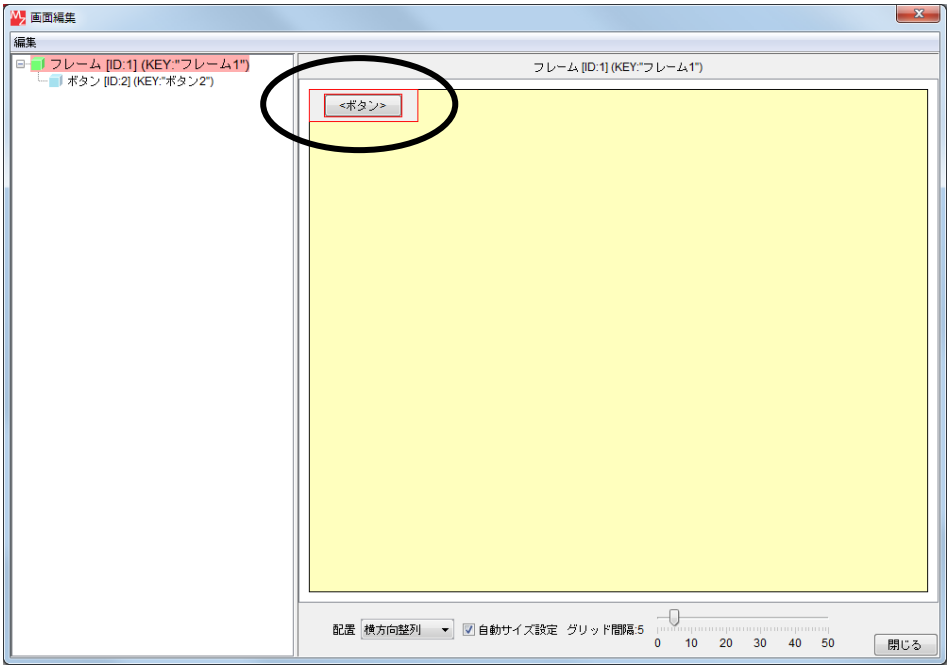
画面編集 をクリックします。

［ボタン］コンポーネントをフレームに追加します。

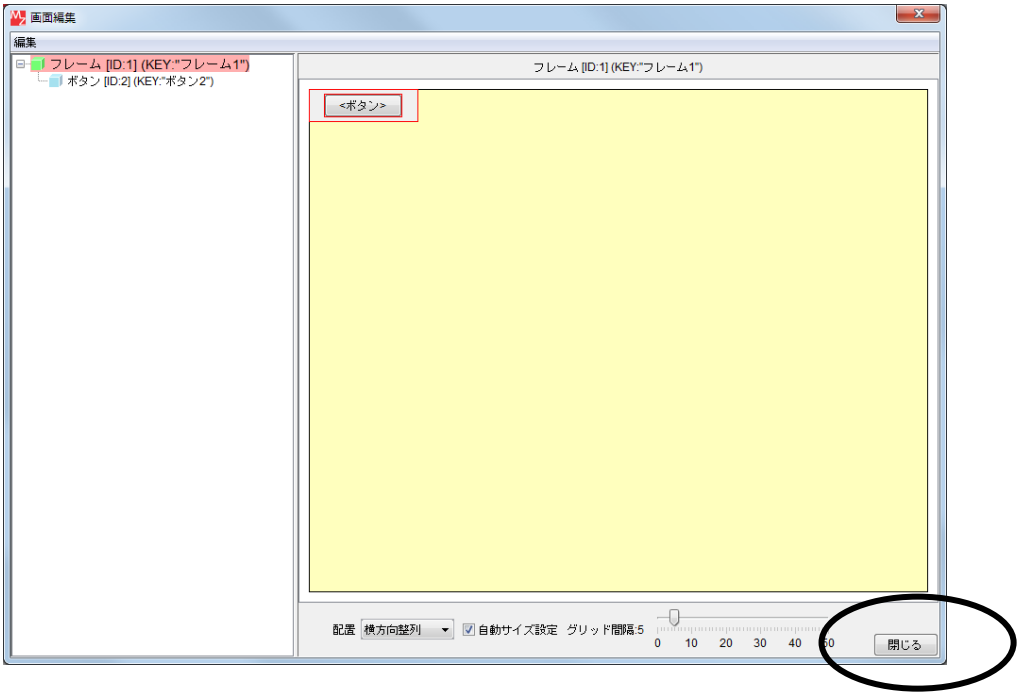
［画面編集］画面上で右クリックー［コンポーネント追加］ー［ボタン］コンポーネントとクリックします。



［ボタン］コンポーネントが［フレーム］に追加されます。

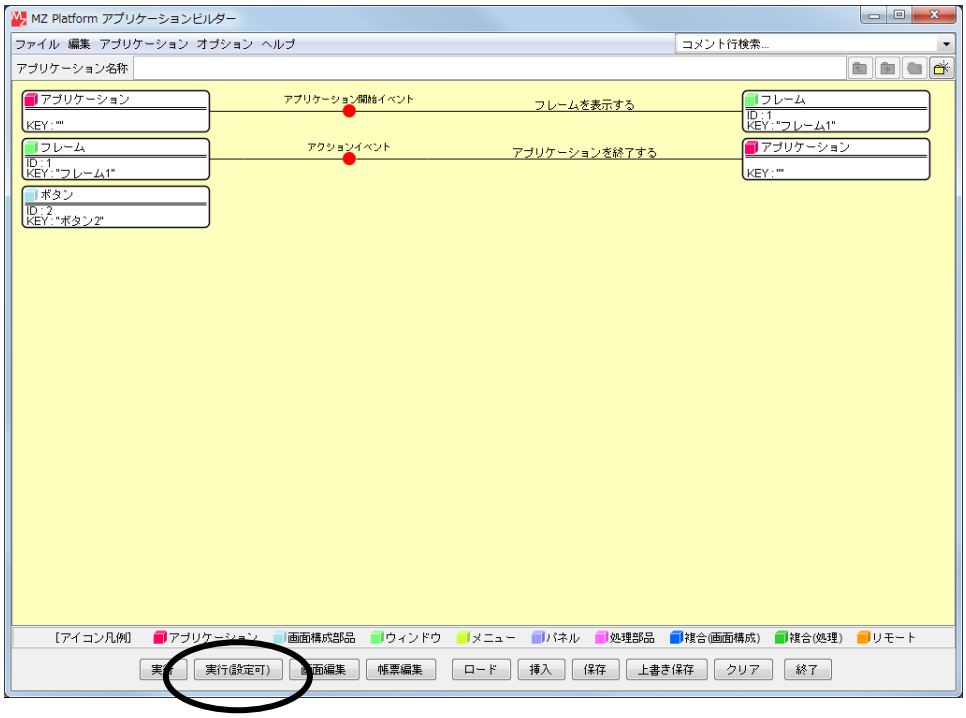


② 追加できたら 閉じる をクリックし、ビルダー画面に戻ります。

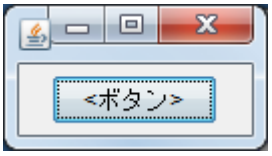


③ ［ボタン］コンポーネントが追加できたことを確認します。

実行(設定可) で実行します。



ボタンコンポーネントが追加されます。

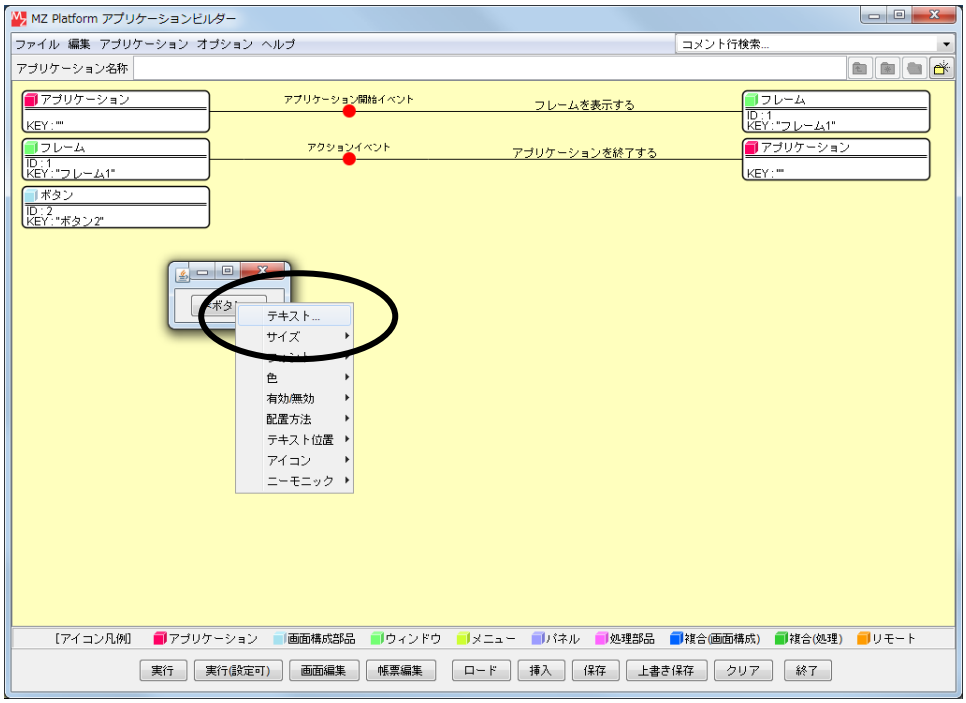


４）ボタン名の変更

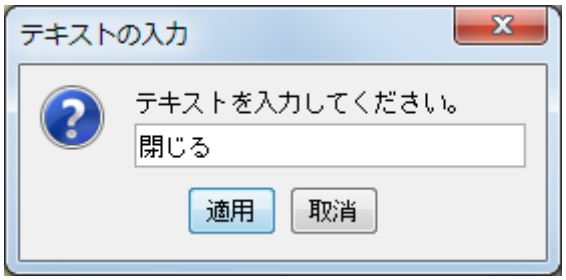
ボタン名を変更することができます。

操 作 ボタン名を変更しましょう。

① 実行(設定可) で実行していることを確認します。
［<ボタン>］の上で右クリックします。
［テキスト...］をクリックします。



② ボタン名を変更します。
［<ボタン>］を消して「閉じる」と入力し、 適用 をクリックします。



確認

適用 をクリックすると表示が変わります。

５）閉じるボタンに機能を割り当てる

閉じるボタンにフレームを閉じる機能を割り当てましょう。

考え方

１．閉じるボタンをクリックしたら、フレームが閉じる。
２．フレームが閉じたら、アプリケーションが終了する。

接続確認

コンポーネント同士の接続を確認します。

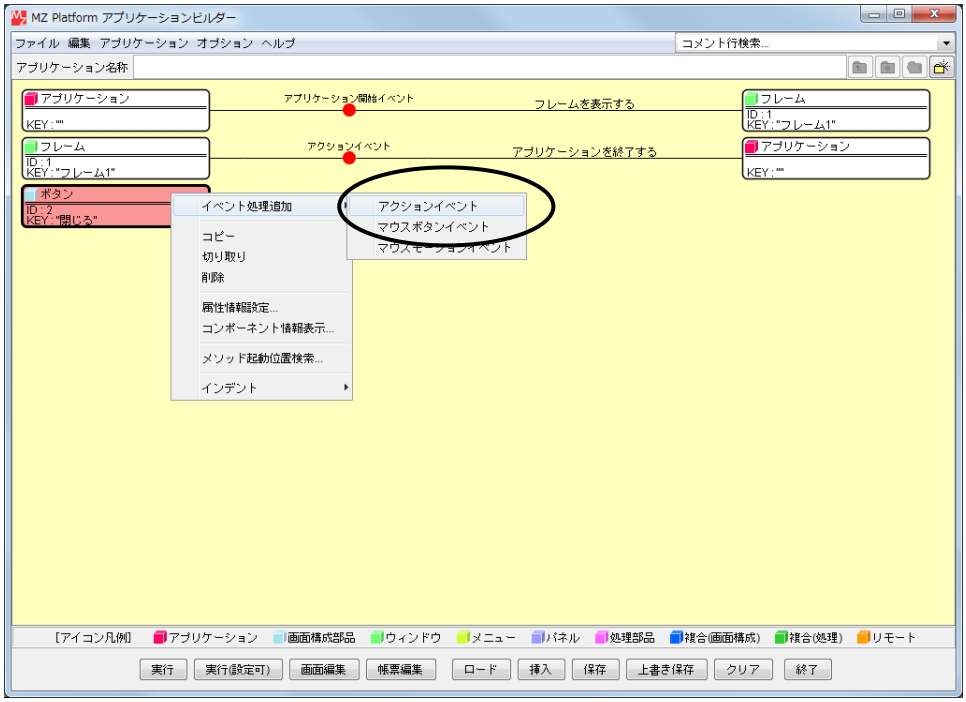
ボタンをクリックしたらフレームが閉じる

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	■フレーム（ID:1）
起動メソッド	フレームを閉じる()

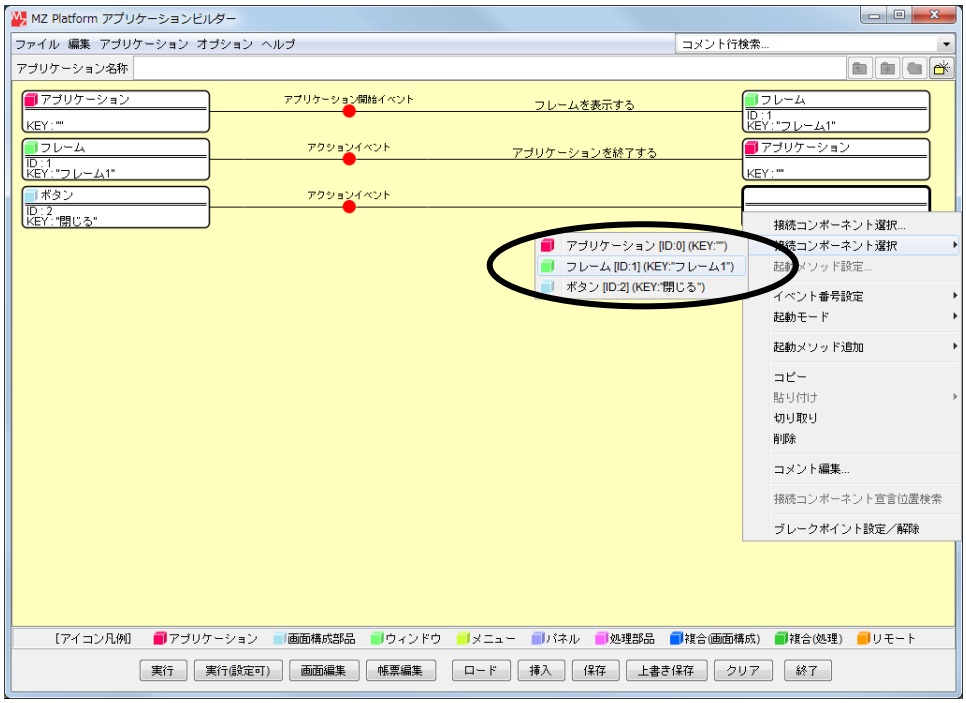
操 作

［閉じる］ボタンをクリックしたらフレームが閉じるように設定しましょう。

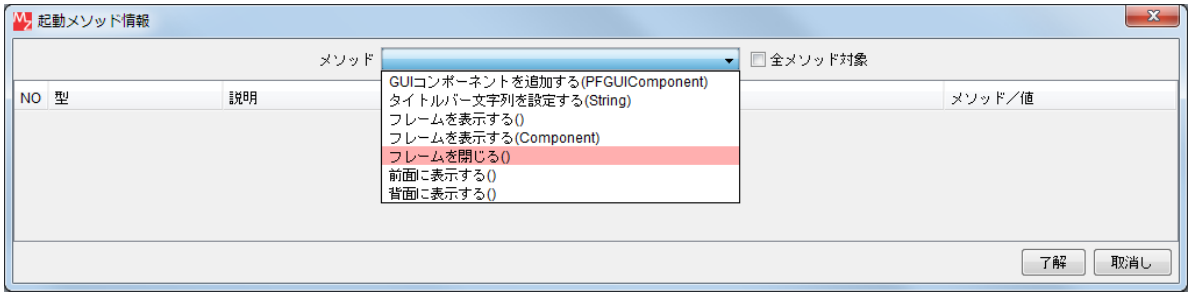
① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ボタン(ID:2)］コンポーネント上で右クリック－［イベント処理追加］－［アクションイベント］とクリックします。



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: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。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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:1)］をクリックします。



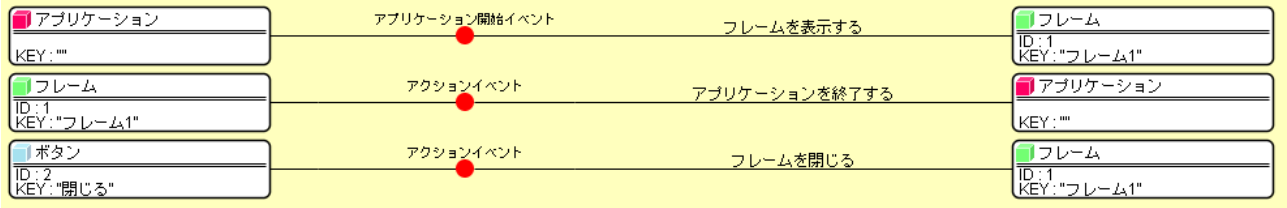
③ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の ▼ をクリックします。
［フレームを閉じる()］をクリックします。
設定後、 了解 ボタンをクリックします。



④ 設定を確認します。
実行(設定可) で実行します。
［閉じる］ボタンをクリックして、フレーム（ウィンドウ）が閉じることを確認します。

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。

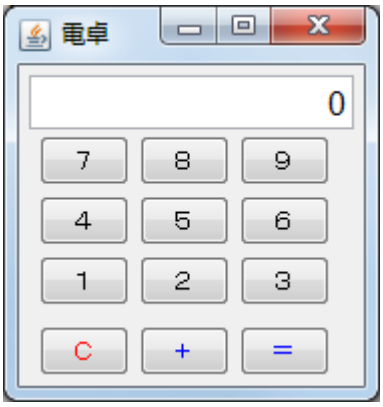


Lesson.5 電卓を作ってみよう

ビルダーを使ったアプリケーション構築の考え方や構築時の操作方法をご紹介します。
ここでは「電卓」を作ってみます。

完成図

電卓の完成図を確認しましょう。



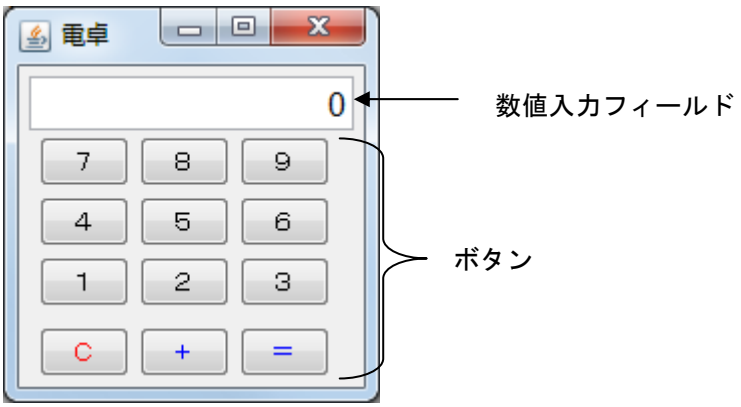
※ここでは１桁の数値が扱える電卓にします。
※計算（演算）は足し算（加算）のみとします。

考え方

１．電卓の画面を作成する。
２．ボタンを押したら数値が表示する。
３．プラス（＋）ボタンとイコール（＝）ボタンを押すと足し算ができ、結果が表示される。

Step.1 画面を作成する

ビルダーでは画面設計と内部処理を分けて考えるとわかりやすくなります。
最初に画面設計をしてから内部処理を考えましょう。
電卓の画面は以下のような構成になっています。



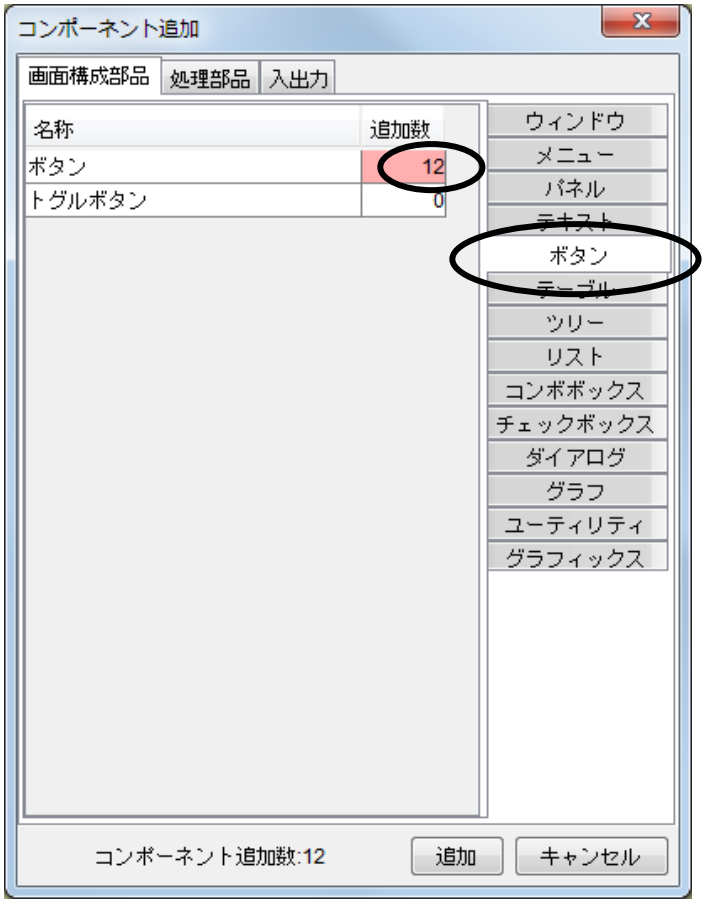
準 備

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］
■数値入力フィールド	1	［画面構成部品］－［テキスト］－［数値入力フィールド］
■ボタン	12	［画面構成部品］－［ボタン］－［ボタン］

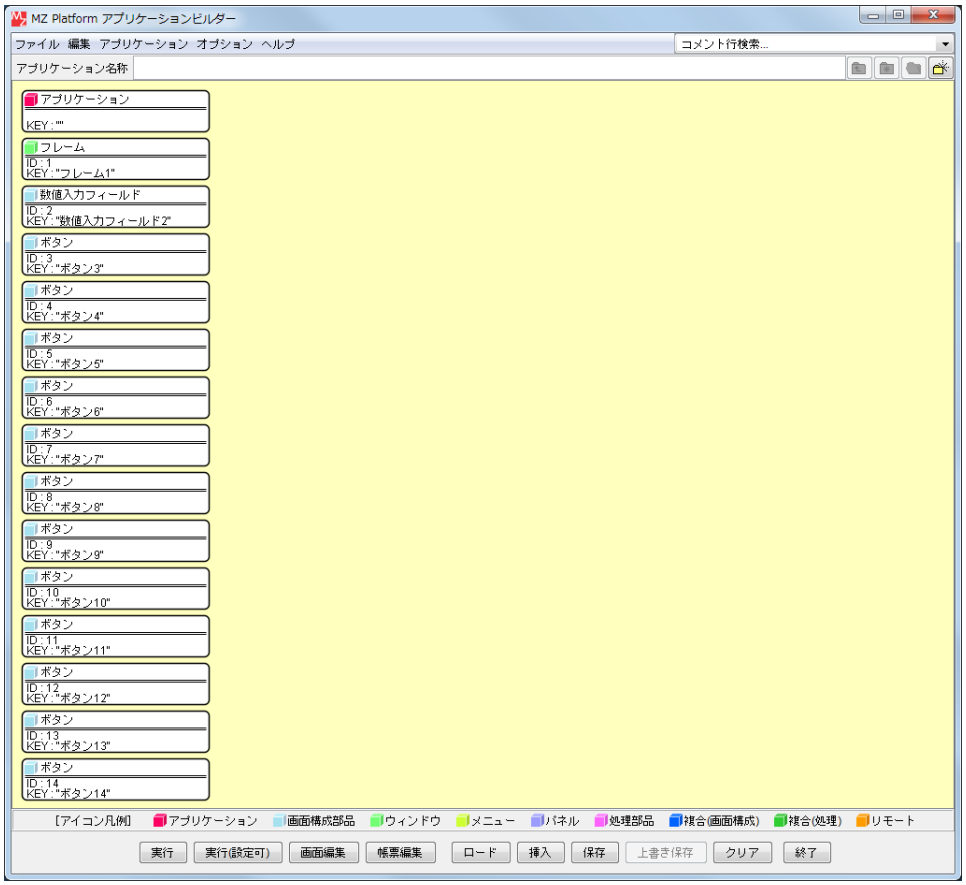
操 作

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］をクリックします。
左側の領域のボタンコンポーネントの追加数に１２（１～９、＋、＝、Ｃ）を入力します。



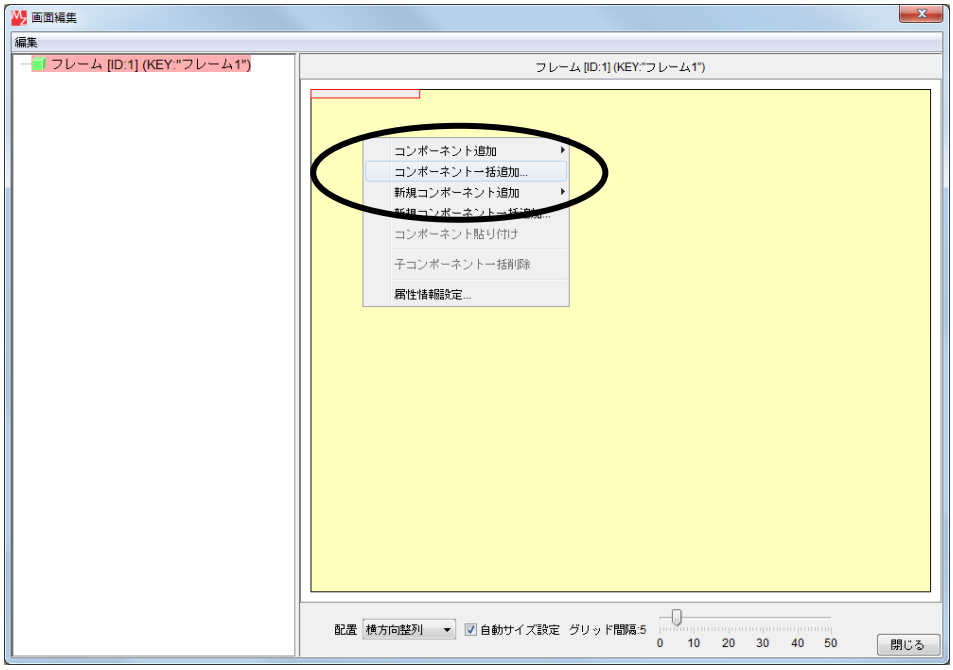
② 分類を切り替えて［画面構成部品］－［ウィンドウ］－［フレーム］の追加数に１を入力します。
分類を切り替えて［画面構成部品］―［テキスト］－［数値入力フィールド］の追加数に１を入力します。
最後に追加ボタンをクリックします。

コンポーネントが追加されます。

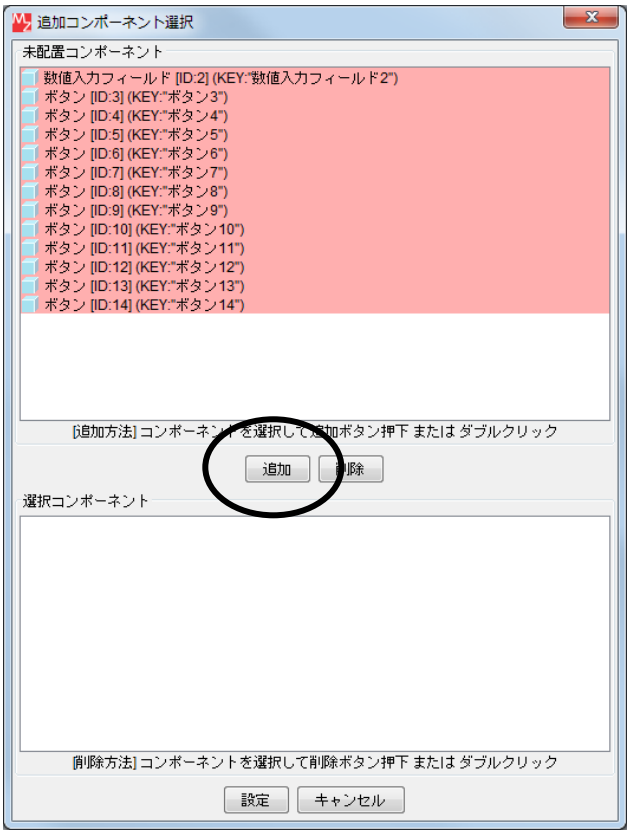


画面編集

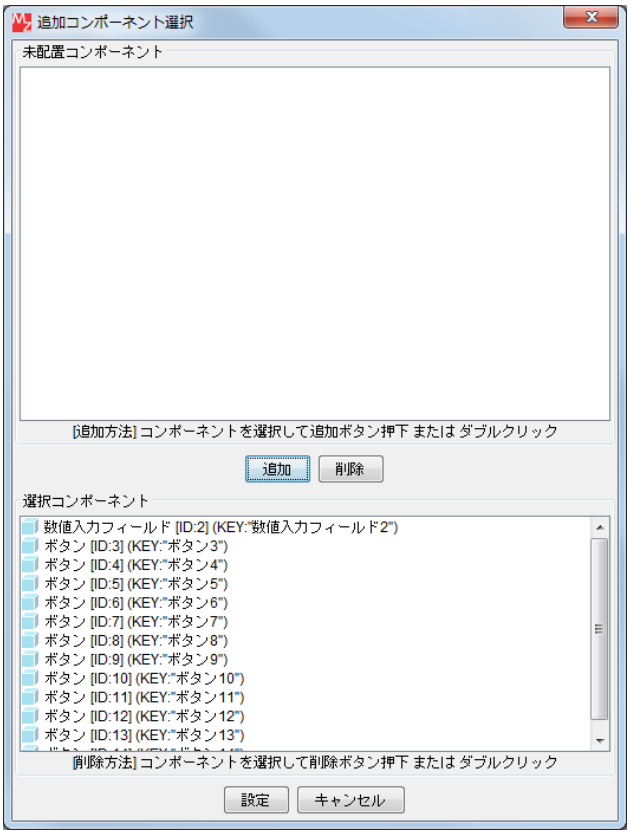
① 画面を作成します。
画面編集をクリックします。
［ボタン］コンポーネントと［数値入力フィールド］コンポーネントをフレームに追加します。
［画面編集］画面上で右クリック－［コンポーネント一括追加］とクリックします。



② ［未配置コンポーネント］を全て［選択コンポーネント］に追加します。
［未配置コンポーネント］の１つめのコンポーネントをクリックし、一番下のコンポーネントを【Shift】＋クリックします。
追加ボタンをクリックします。

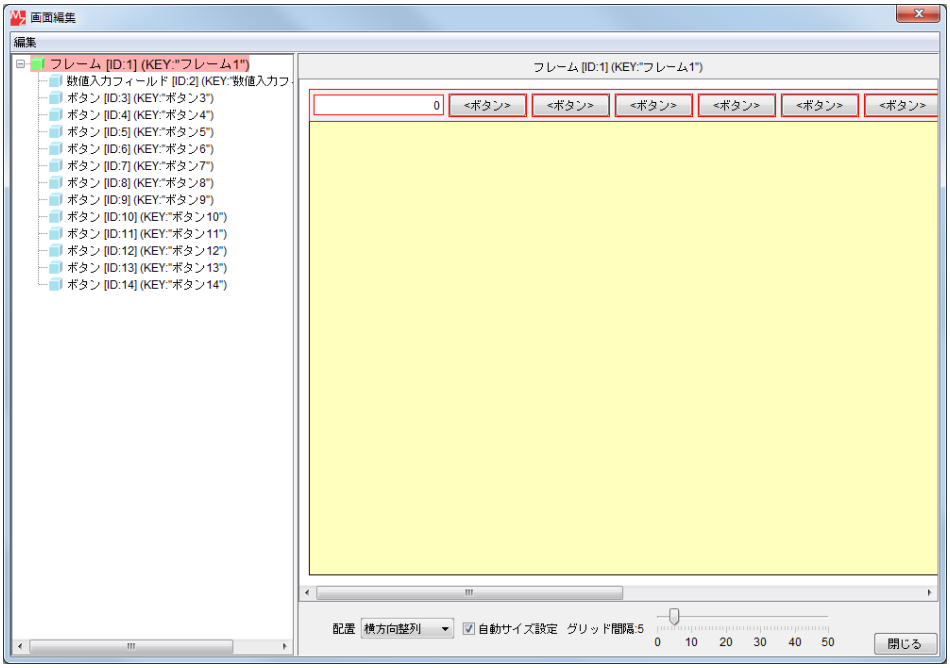


［選択コンポーネント］に追加されます。



③　設定をクリックします。

フレームに追加されます。（横方向に整列します）



④　閉じるボタンをクリックし、ビルダーの画面に戻ります。

接続確認

コンポーネント同士の接続を確認します。

開始

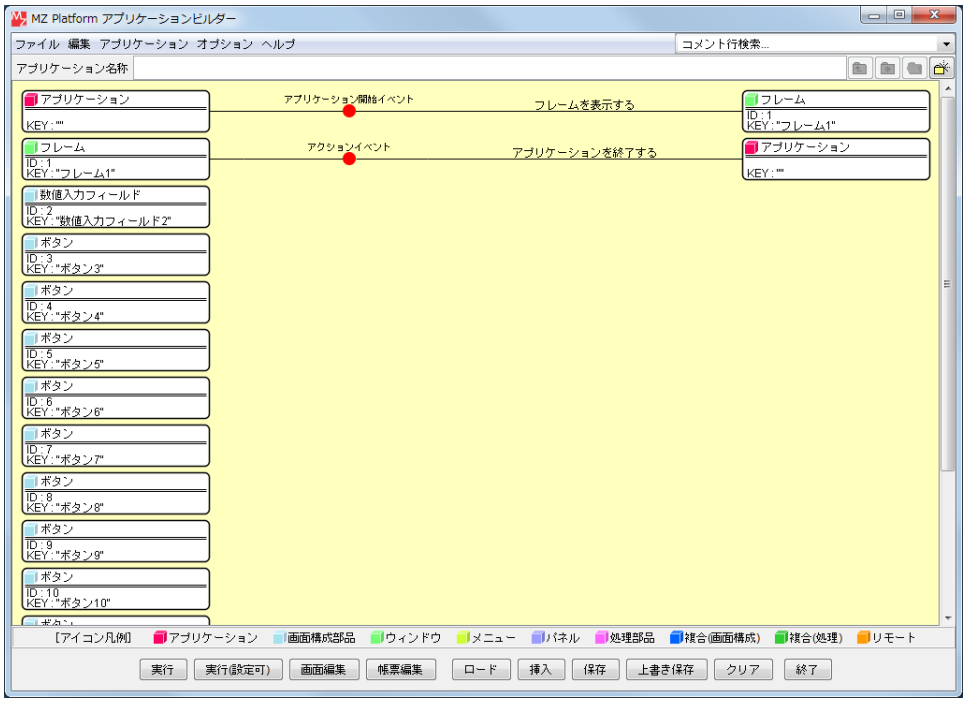
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■アプリケーション
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■フレーム（ID:1）
起動メソッド	フレームを表示する()

終了

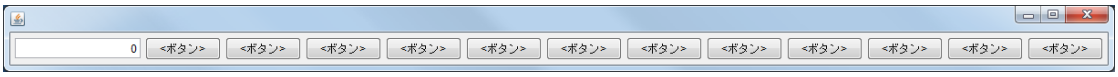
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■フレーム（ID:1）
発生イベント	アクションイベント
接続先コンポーネント	■アプリケーション
起動メソッド	アプリケーションを終了する()

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。
これは全てのアプリケーションで共通です。
アプリケーションを作成する場合には必ず設定しておきます。

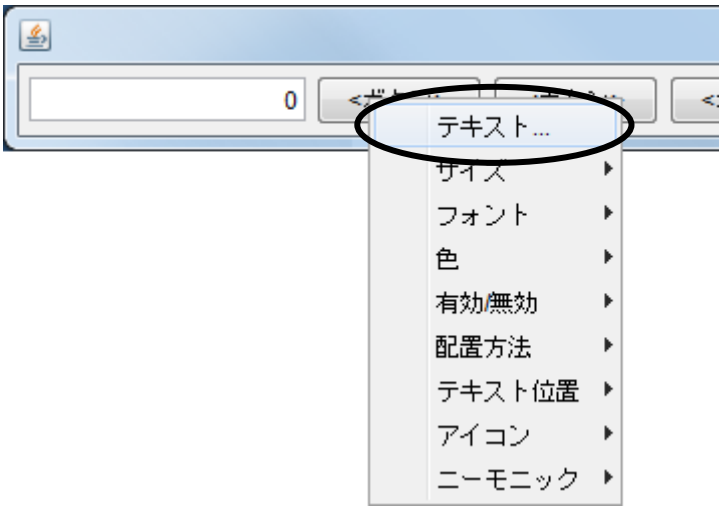


② 実行（設定可）ボタンをクリックし実行します。

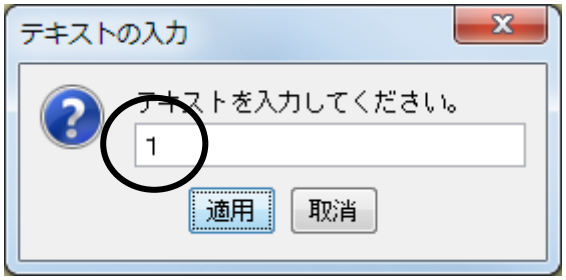


操 作 ボタンと数値入力フィールドを編集しましょう。

① ボタンの名前を変更します。
実行（設定可）で実行していることを確認して、
ボタン名を変更したいボタンの上で右クリック－［テキスト...］をクリックします。

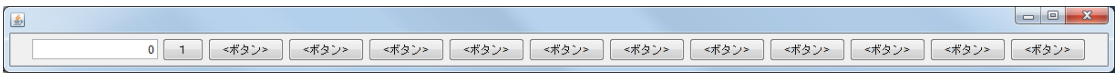


② ボタン名を入力し適用をクリックします。



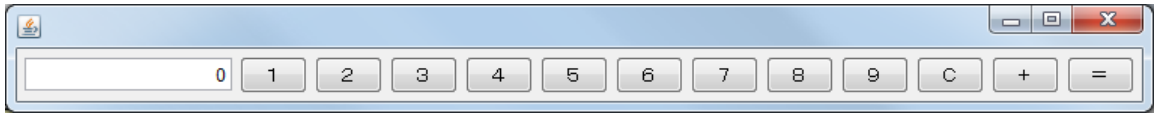
確認

ボタン名が変更されます。



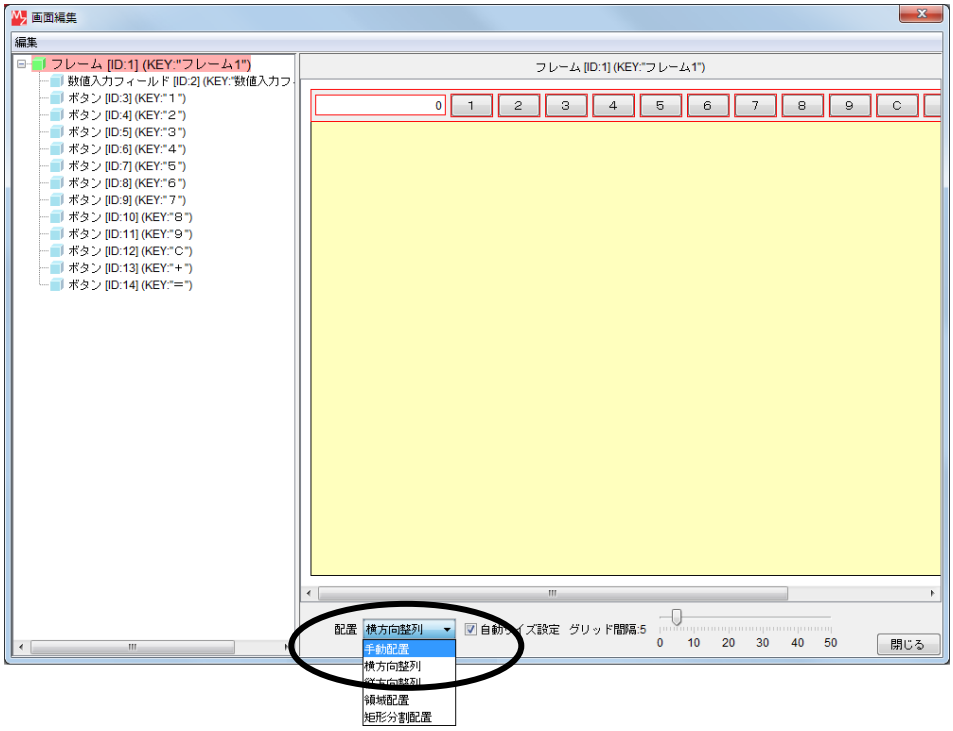
③ ①～②の操作を繰り返してすべてのボタン名を「２」から「９」までと、「Ｃ」「＋」「＝」に変更しましょう。変更すると以下のようになります。確認後ウィンドウを終了しておきます。

確認

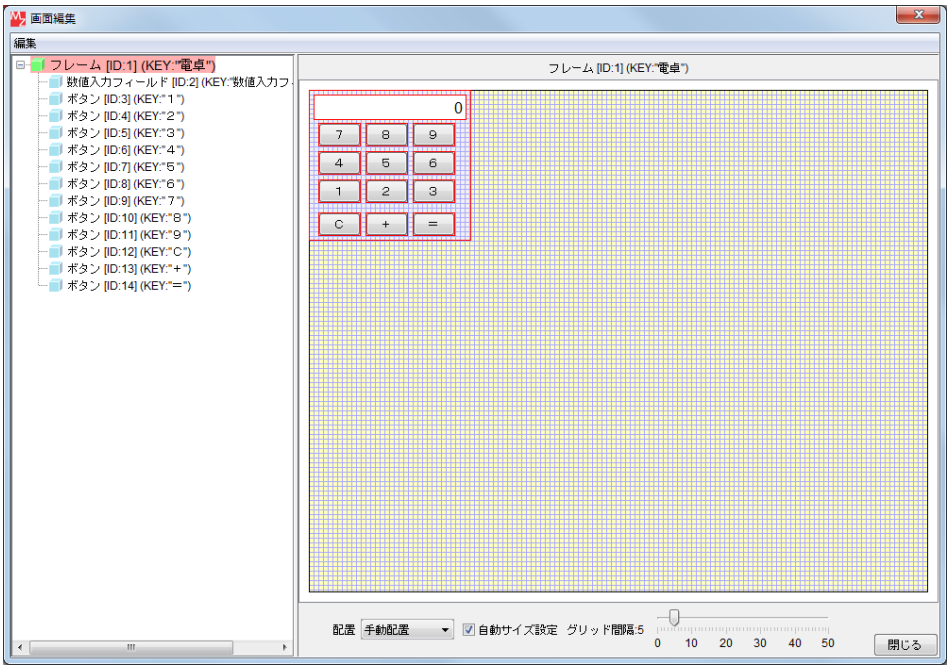


④　ボタンの配置を変更します。
画面編集ボタンをクリックします。

⑤　［配置］を［手動配置］に変更します。

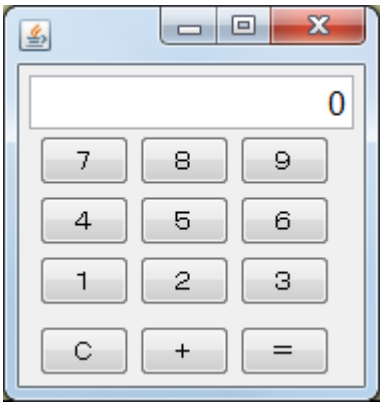


⑥　［ボタン］コンポーネントと［数値入力フィールド］コンポーネントをドラッグで移動し、電卓に見えるように配置します。



⑦　閉じるをクリックし［画面編集］を終了します。

⑧ 実行（設定可）で実行します。



知っていると便利！

1．［画面編集］の「配置」には次の５種類があります。

配置の種類	説　明
手動配置	画面に追加されているコンポーネントを画面に自由に配置できます
横方向整列	画面に追加されているコンポーネントがすべて横方向に並びます 既定値です
縦方向整列	画面に追加されているコンポーネントがすべて縦方向に並びます
領域配置	ウィンドウのサイズに合わせてコンポーネントの表示領域が変更されます
矩形分割配置	画面サイズを変更しても表示されているコンポーネントの大きさは変わりません。

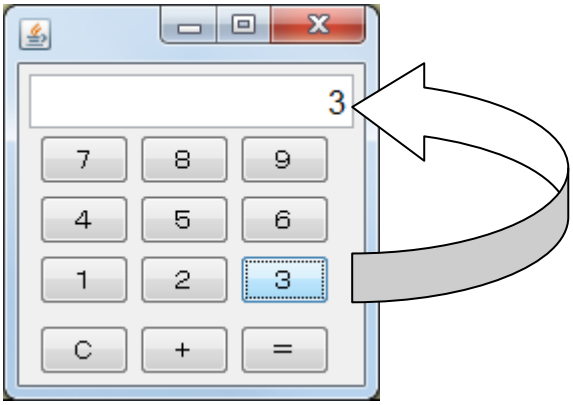
2．［数値入力フィールド］コンポーネントの数値の配置方法には５種類あります。
実行（設定可）で編集します。

位　置	意　味
左端	左端に配置されます
中央	中央に配置されます
右端	右端に配置されます
リーディングエッジ	文頭に配置されます
トレーリングエッジ	文末に配置されます

Step.2 内部処理を設定する

内部処理（ここでは計算処理）の部分を設定します。

１）［数値ボタン］をクリックしたら［数値入力フィールド］にボタンの数字を表示する



考え方

１．電卓の数字ボタンをクリックしたら、そのボタンの数字が数値入力フィールドに出力される。

接続確認コンポーネント同士の接続を確認します。

数値ボタンをクリックしたら数値入力フィールドにボタンの数値を表示

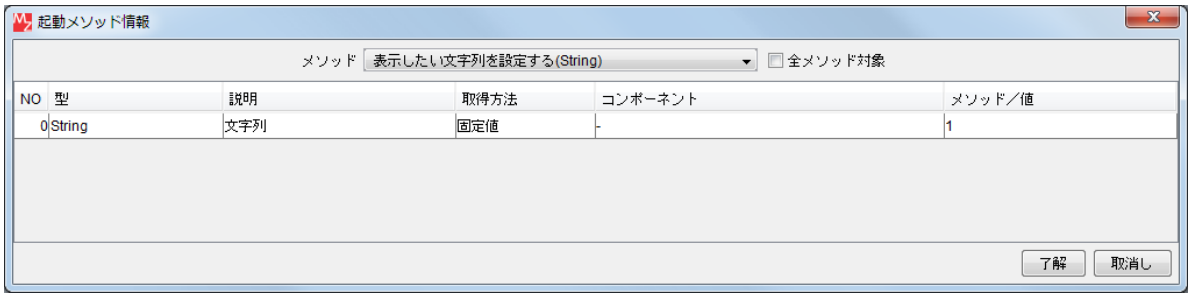
項　目	内　容
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（ID:3）　～　■ボタン（ID:11）
発生イベント	アクションイベント
接続先コンポーネント	■数値入力フィールド（ID:2）
起動メソッド	表示したい文字列を設定する(String)
＜引数＞	説明：文字列 取得方法：固定値 メソッド／値：1 ～ 9

操 作数字ボタンをクリックしたら数値入力フィールドに数字が表示されるように設定しましょう。

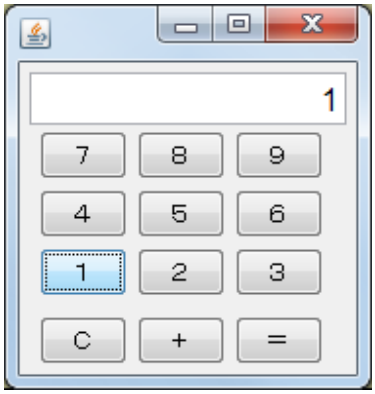
① 使用するイベントを選択し、コンポーネントを接続する準備をします。
左側の［ボタン(ID:3)］コンポーネント上で右クリックー［イベント処理追加］
ー［アクションイベント］とクリックします。

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:3)］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。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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：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。



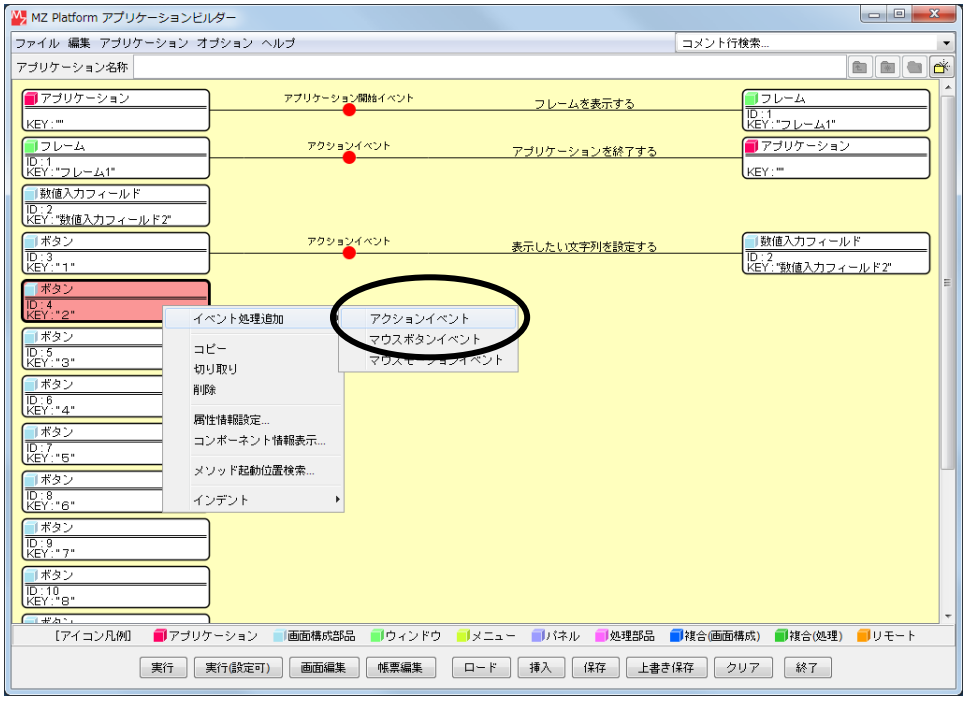
④ 設定を確認します。
実行（設定可）で実行します。［ボタン］の「１」をクリックします。
［数値入力フィールド］に「１」が表示されます。



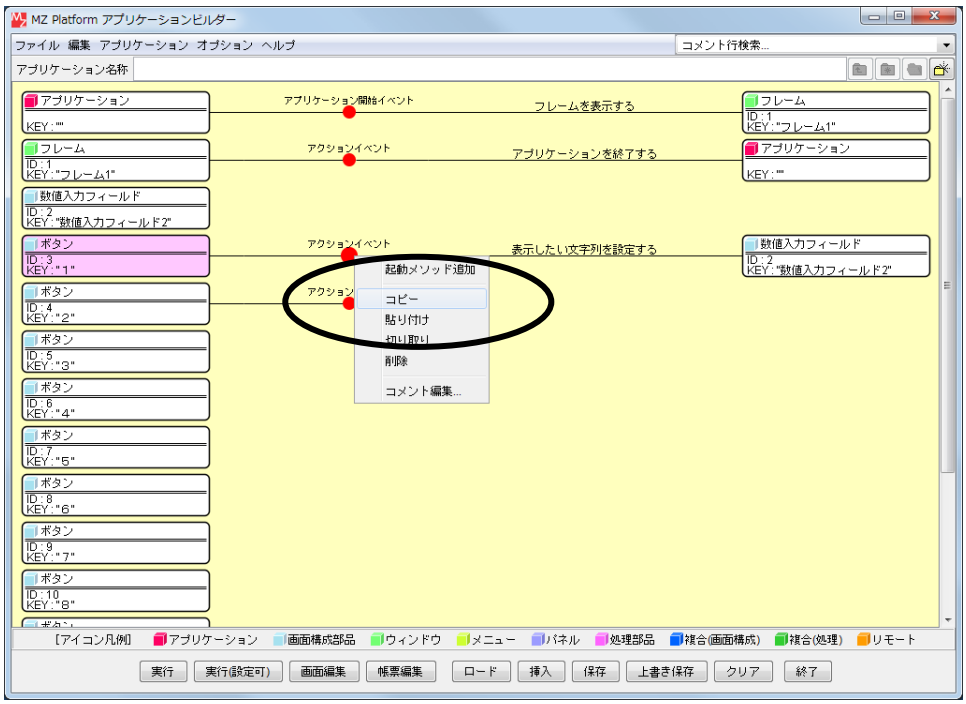
操 作

ボタン「２」～「９」を設定します。前の操作で設定したことをほぼ同じ設定ですので、起動メソッドの［コピー］を使用します。

①　「２」のボタンに、［アクションイベント］を選択しコンポーネントを接続する準備をしておきます。

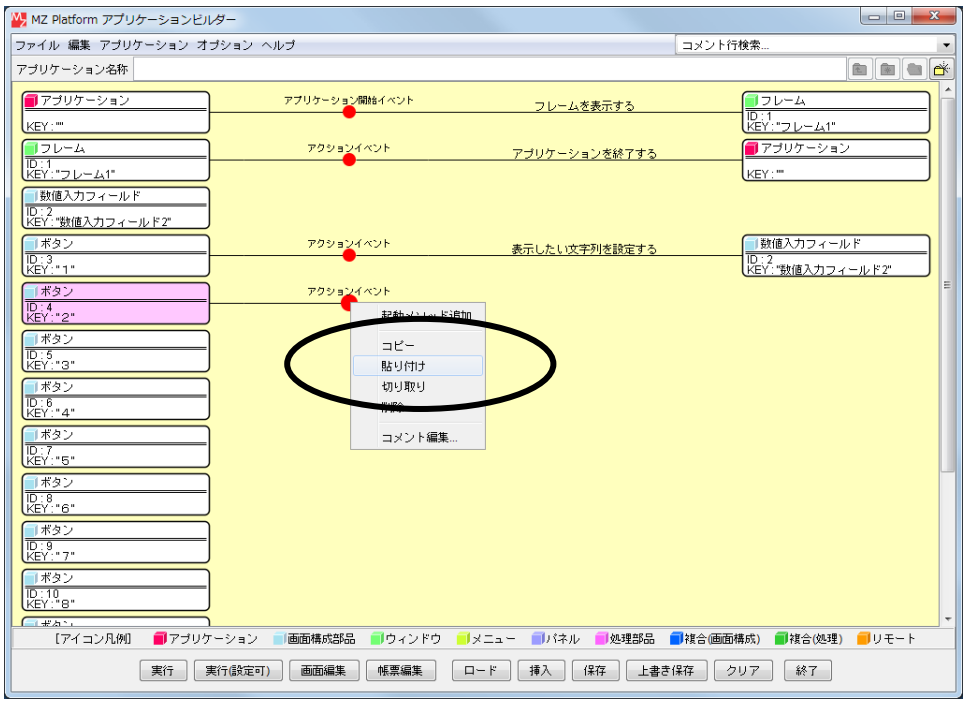


②　［ボタン１］の［アクションイベント］の上で右クリックー［コピー］をクリックします。

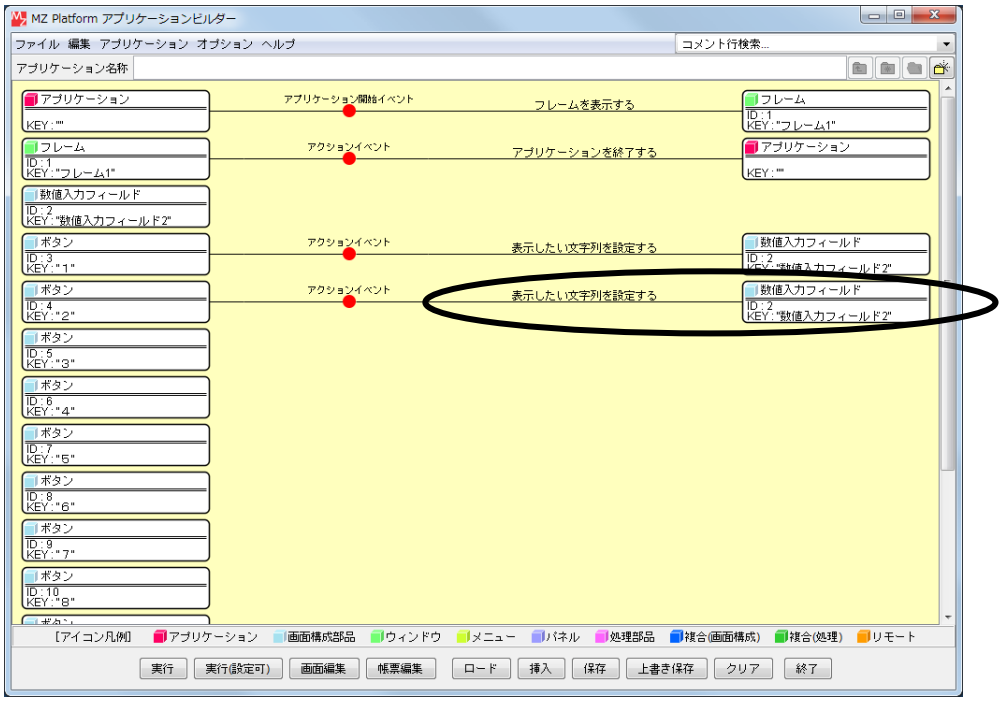


③　［ボタン２］の［アクションイベント］の上で右クリックー［貼り付け］を

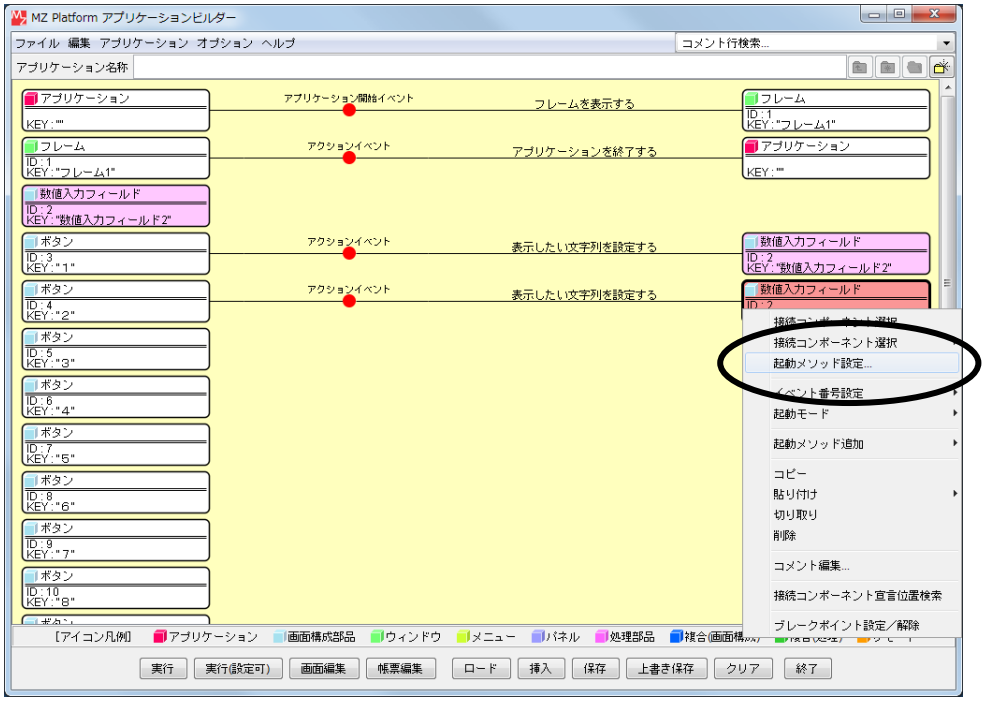
クリックします。



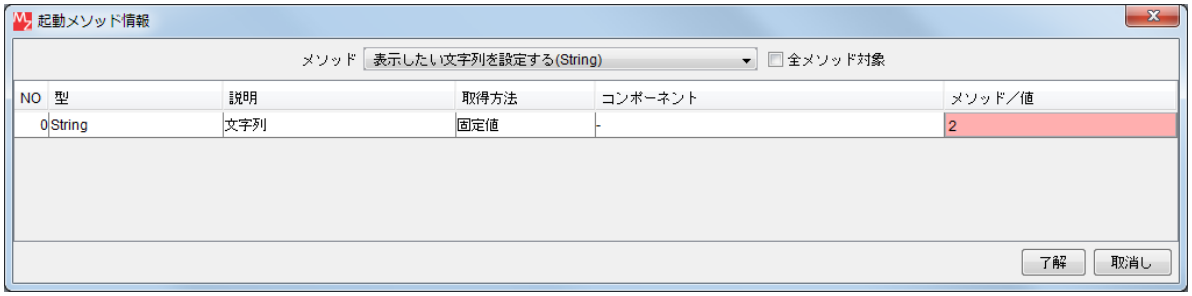
［ボタン１］の起動メソッドが貼り付きます。



④　③で貼り付けた［起動メソッド］の固定値を変更します。
　　貼り付けた［起動メソッド］の上で右クリック－［起動メソッドの設定...］をクリックします。



⑤　［起動メソッド］の［メソッド／値］を「２」に変更します。
　　設定後、了解ボタンをクリックします。



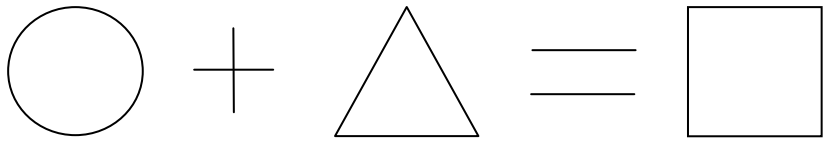
⑥　①～⑤の操作を繰り返し、「３」～「９」のボタンを設定します。

２）入力したデータを計算する

ボタンをクリックして数値入力フィールドに表示された数値を加算します。

考え方

１．加算は次のような式で書くことができる。



加算コンポーネントの中で、○を「左オペランド」、△を「右オペランド」と呼ぶ。

２．＋ボタンや＝ボタンが押されたとき、次のように○と△に数値を入力する。
＋ボタンが押された時、その時に数値入力フィールドに表示されている数字を○（左オペランド）へ
＝ボタンが押された時、その時に数値入力フィールドに表示されている数字を△（右オペランド）へ
それぞれ設定する。

３．最後に加算して、その結果を数値入力フィールドに表示して終了する。

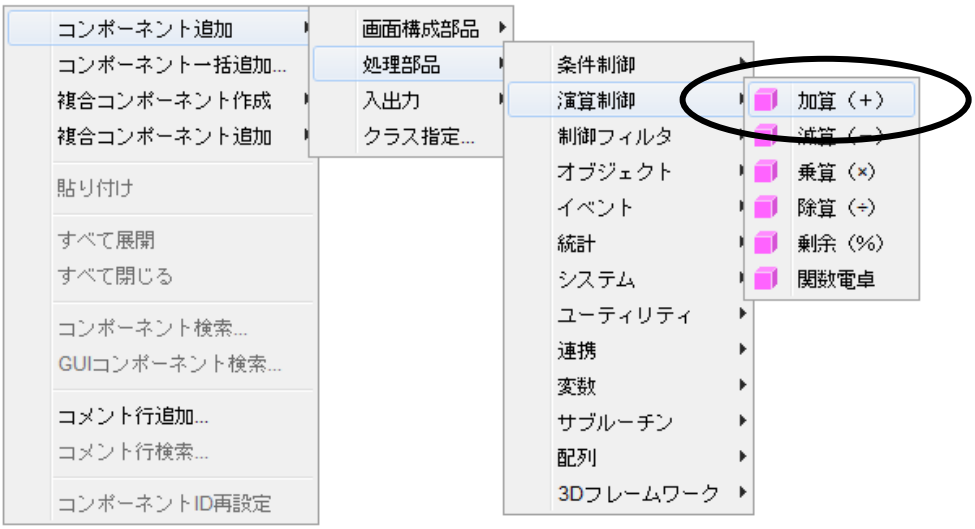
準 備

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	
■加算(+)	1	［処理部品］－［演算制御］－［加算(+)］

操 作

① 必要なコンポーネントを追加します。
ここでは［加算］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］－［演算制御］－［加算(+)］
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。

左オペランドを設定する

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（ID:13, 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	■加算(+)（ID:15）
起動メソッド	数値に変換後、左オペランドを設定する(String)
＜引数＞	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：数値入力フィールド メソッド／値：表示されている文字列を取得する

右オペランドを設定する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）	■ボタン（ID:14, 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❶	■加算(+)（ID:15）
起動メソッド	数値に変換後、右オペランドを設定する(String)
＜引数＞	説明：右オペランド 取得方法：メソッド戻り値 コンポーネント：数値入力フィールド メソッド／値：表示されている文字列を取得する

加算する❷

接続先コンポーネント❷	■加算(+)（ID:15）
起動メソッド	演算を行う()

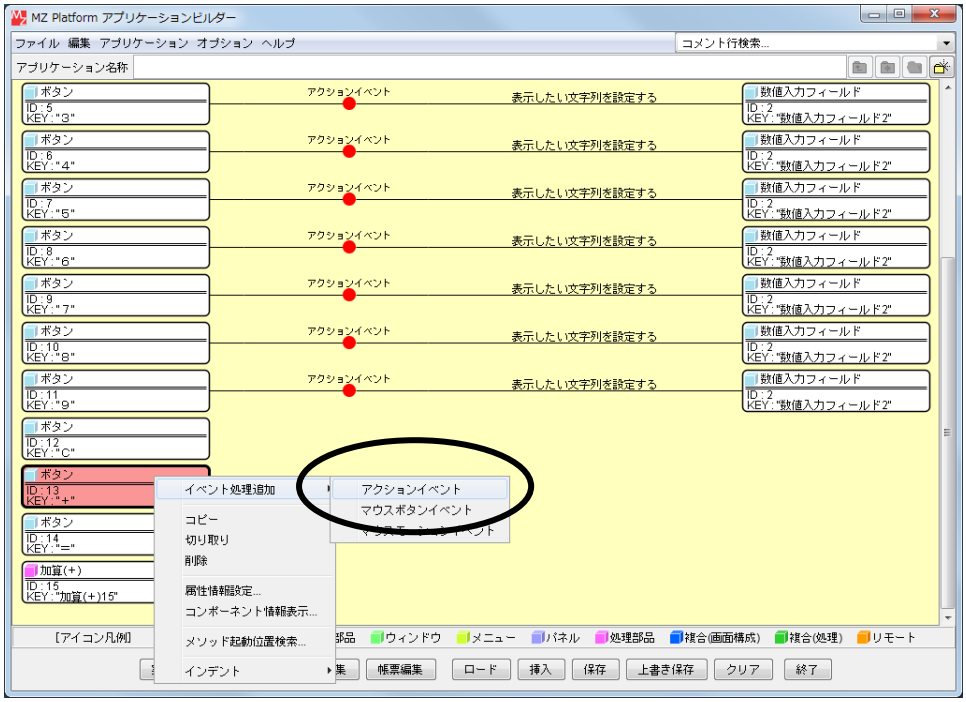
加算結果を表示する

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）	■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	■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：処理結果データ

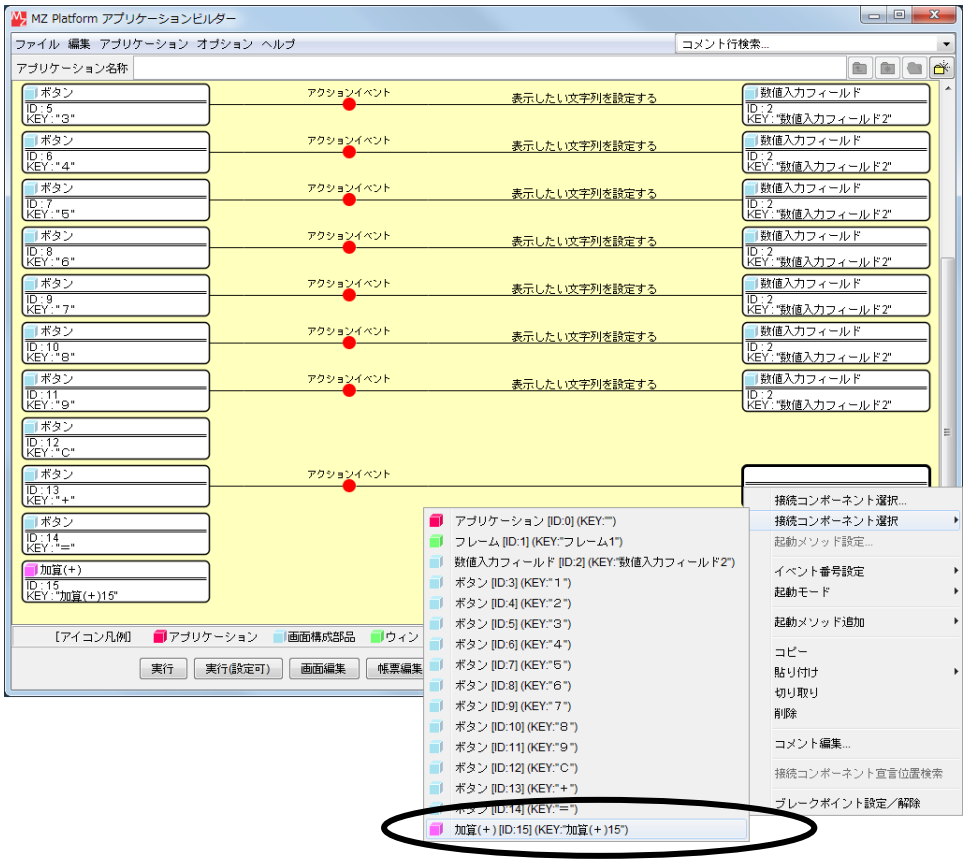
操 作 電卓の計算処理を設定しましょう。

―――左オペランドを設定する―――

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:13)］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。



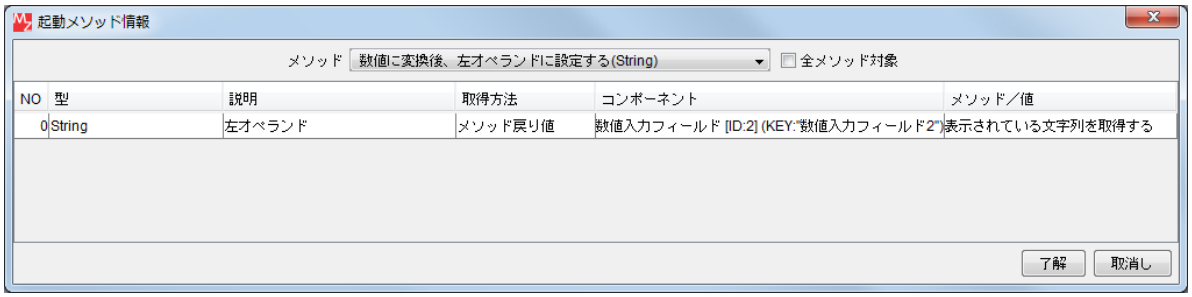
② イベントの接続先コンポーネントを選びます。
左側の［＋ボタン(ID:13)］コンポーネントの［アクションイベント］上で
右クリック－［起動メソッド追加］とクリックします。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空の四角い枠の上で右クリック－［接続コンポーネント選択］－
［加算(ID:15)］コンポーネントをクリックします。



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。
［数値に変換後、左オペランドを設定する(String)］をクリックします。
引数を設定します。

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：数値入力フィールド(ID:2)
メソッド／値：表示されている文字列を取得する

設定後、了解ボタンをクリックして閉じます。

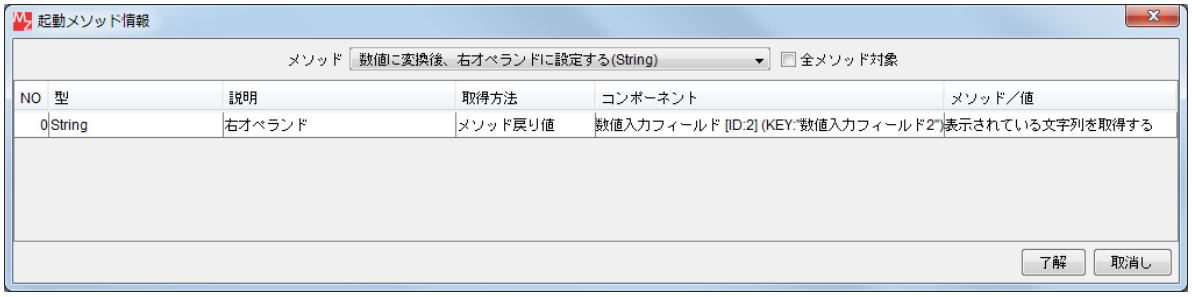


―――右オペランドを設定する❶―――

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［＝ボタン(ID:14)］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。

⑤ 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:14)］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。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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:15)］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。
［数値に変換後、右オペランドを設定する(String)］をクリックします。
引数を設定します。
説明：右オペランド
取得方法：メソッド戻り値
コンポーネント：数値入力フィールド(ID:2)
メソッド／値：表示されている文字列を取得する
設定後、了解ボタンをクリックして閉じます。

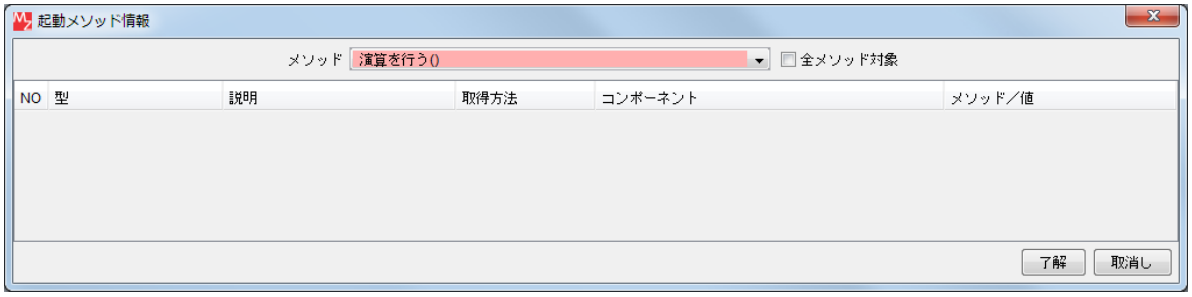


―――加算する❷―――

⑦ 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:14)］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。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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:15)］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。

［演算を行う()］をクリックします。
設定後、了解ボタンをクリックして閉じます。

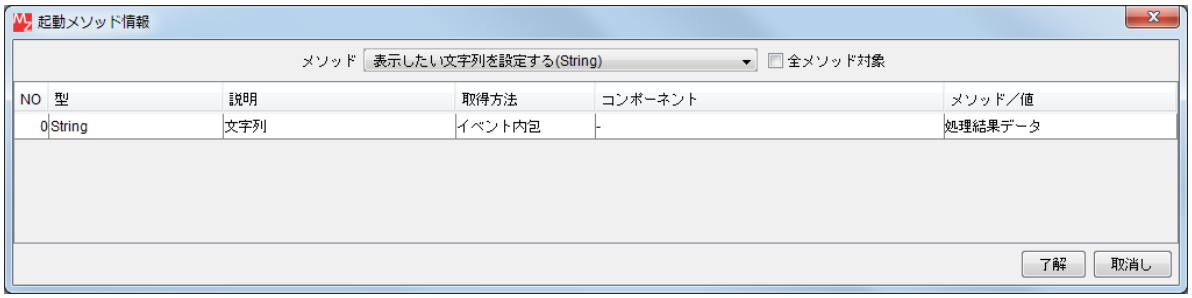


―――加算結果を表示する―――

⑨ 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:15)］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。

⑩ 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－［起動メソッド追加］とクリックします。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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)］コンポーネントをクリックします。

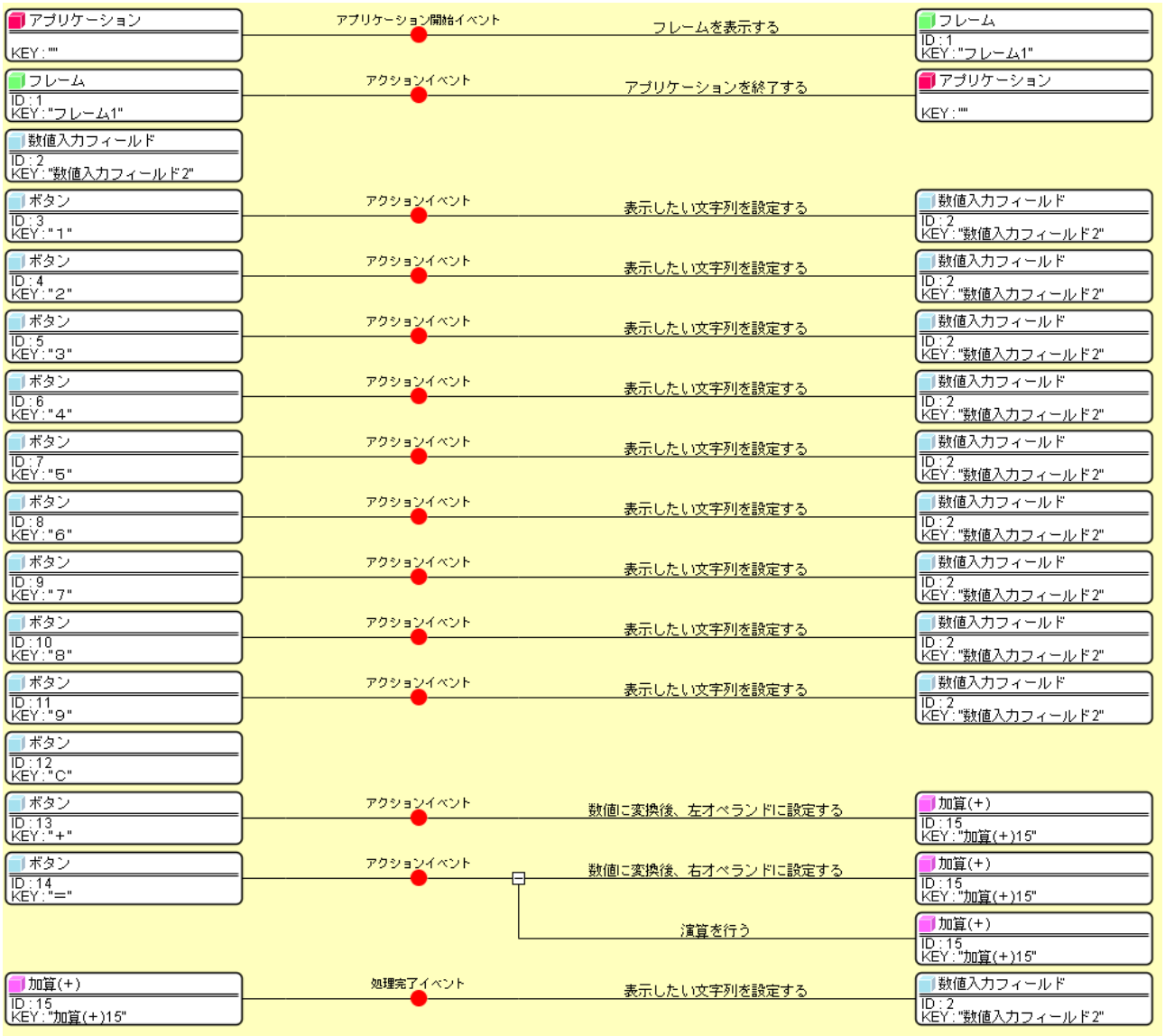
⑪ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック－［起動メソッド設定...］をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド（処理）を選びます。
［メソッド］の をクリックします。
［表示したい文字列を設定する(String)］をクリックします。
引数を設定します。
説明：文字列
取得方法：イベント内包
メソッド／値：処理結果データ
設定後、了解ボタンをクリックして閉じます。



⑫ 確認します。
実行（設定可）で実行します。
電卓上で一桁の足し算を行います。

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 機能を拡張する

以下の機能を追加して使いやすくしましょう。

1）クリアボタン：「C」のボタンを押すと画面表示を「０」にするようにします。
2）連続計算：この状態では、「１＋２＋３＋４＋５」と入力すると「４＋５」だけが有効なので、
　　連続して計算できるようにします。
3）計算終了：計算が終了したら、次の計算のために左右のオペランドをクリアします。

１）クリアボタン

「C」のボタンが押されたら、加算コンポーネントの内容をクリアし、画面表示を「０」に戻します。

考え方

1．［Cボタン］が押されたら［加算］コンポーネント内の左右オペランドに０を入れる。
2．０＋０を計算させて、結果の０を画面上に表示する。

接続確認コンポーネント同士の接続を確認します。

［C（クリア）ボタン］をクリックしたら表示を０にする

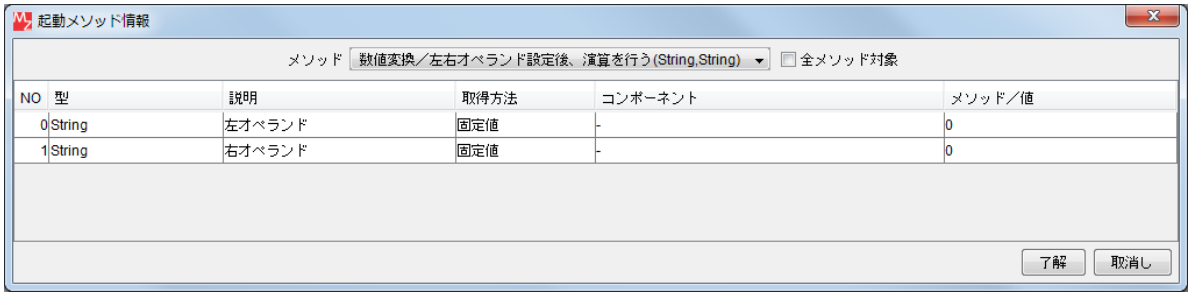
接続項目	接続関係
接続元コンポーネント （イベント発生コンポーネント）	■ボタン（ID:12, Key:C）
発生イベント	アクションイベント
接続先コンポーネント	■加算(+)（ID:15）
起動メソッド	数値変換／左右オペランド設定後、演算を行う (String,String)
＜引数０＞	説明：左オペランド 取得方法：固定値 メソッド／値：0
＜引数１＞	説明：右オペランド 取得方法：固定値 メソッド／値：0

操 作「C」にクリアの機能を追加します。

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［Cボタン(ID:12)］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［Cボタン(ID:12)］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。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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:15)］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０を設定します。
　説明：左オペランド
　取得方法：固定値
　メソッド／値：0
引数１を設定します。
　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。



２）連続計算

これまでの設定では「１＋２＋３＋４＋５」と入力すると「４＋５」だけが有効になり、結果は「９」になります。　これを「１＋２＋３＋４＋５」は「１５」と計算できるようにします。

考え方

1．［＋］ボタンが押されたら［＝］と同じ処理をするように設定しておけば途中で計算が行われ、計算結果が正しくなる。

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］コンポーネントから［加算］コンポーネントに３つ接続します。

右オペランドを設定する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）	■ボタン（ID:13, 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❶	■加算(+)（ID:15）
起動メソッド	数値に変換後、右オペランドを設定する(String)
＜引数＞	説明：右オペランド 取得方法：メソッド戻り値 コンポーネント：数値入力フィールド メソッド／値：表示されている文字列を取得する

演算する❷

接続先コンポーネント❷	■加算(+)（ID:15）
起動メソッド	演算を行う()

左オペランドを設定する❸

接続先コンポーネント❸ （基礎編 Step2 で設定済み）	■加算(+)（ID:15）
起動メソッド	数値に変換後、左オペランド設定する(String)
＜引数＞	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：数値入力フィールド メソッド／値：表示されている文字列を取得する

操 作

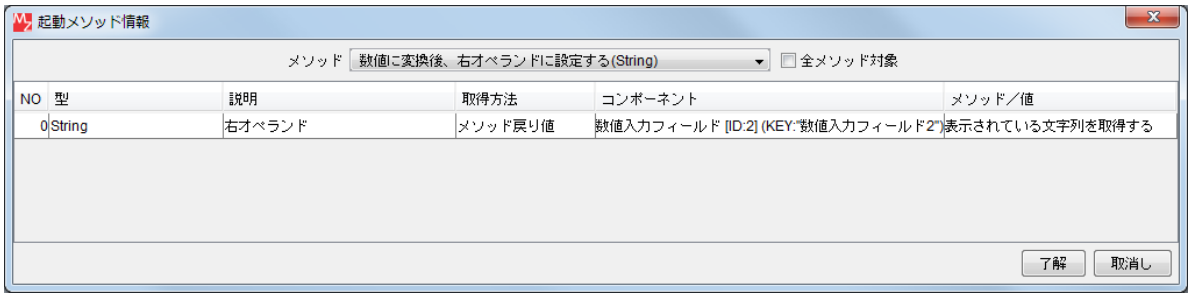
連続計算の機能を設定します。

［＋ボタン(ID:13)］コンポーネントから３つのコンポーネントを接続します。

―――右オペランドを設定する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［＋ボタン(ID:13)］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。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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:15)］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。
［数値に変換後、右オペランドを設定する(String)］をクリックします。
引数を設定します。
説明：右オペランド
取得方法：メソッド戻り値
コンポーネント：数値入力フィールド(ID:2)
メソッド／値：表示されている文字列を取得する
設定後、了解ボタンをクリックします。

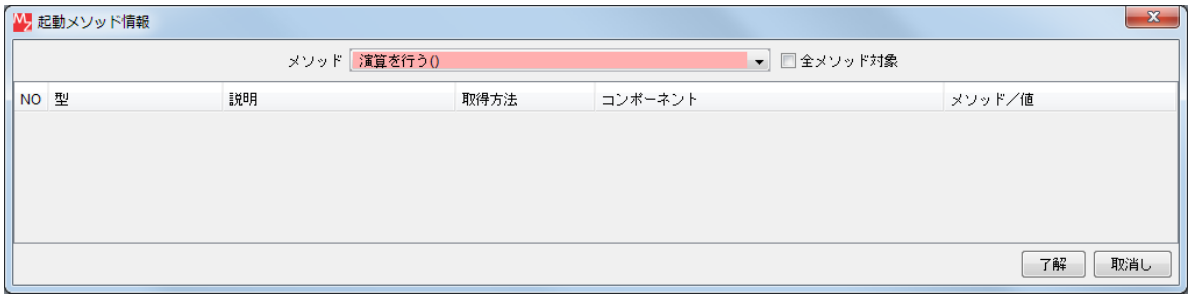


―――演算する❷―――

③ イベントの接続先コンポーネントを選びます。
左側の［＋ボタン(ID:13)］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。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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:15)］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－［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。

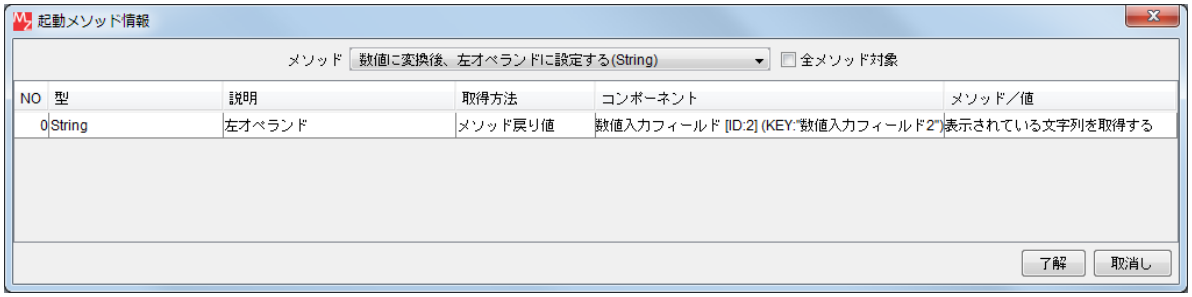
［メソッド］の ▼ をクリックします。
［演算を行う()］をクリックします。
設定後、了解ボタンをクリックします。



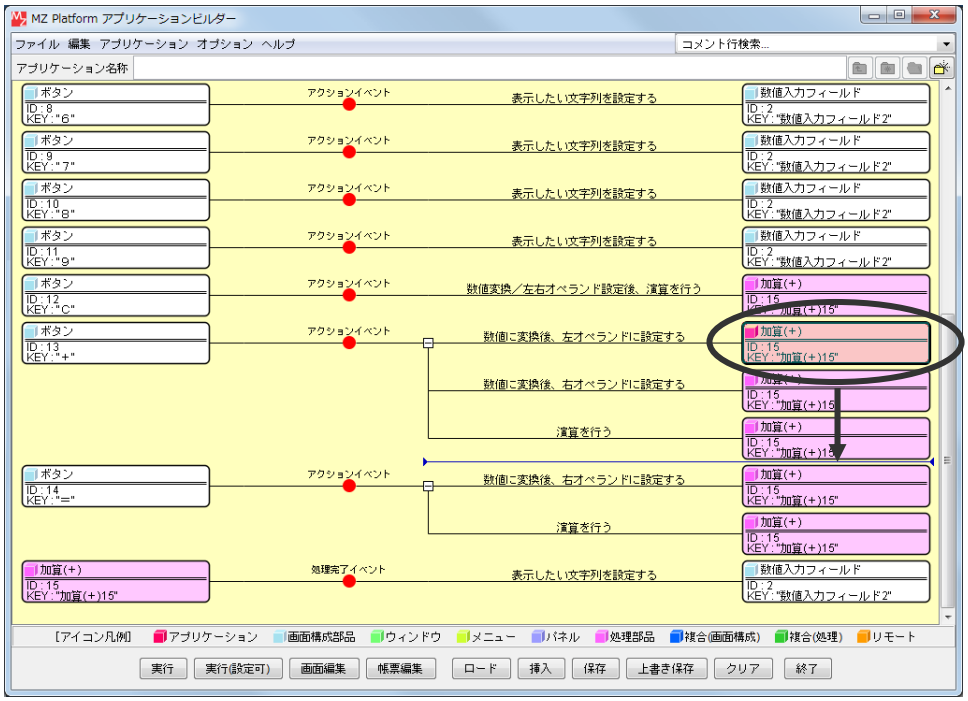
―――左オペランドを設定する❸（基礎編 Step2 で設定済み）―――

⑤ 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:13)］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。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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:15)］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－［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。
［数値に変換後、左オペランドを設定する(String)］をクリックします。
引数を設定します。
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：数値入力フィールド(ID:2)
メソッド／値：表示されている文字列を取得する
設定後、了解ボタンをクリックします。



⑦ 以下のようにメソッドの位置を入れ替えます。
移動したいメソッドをドラッグすると移動できます。


MZ Platform アプリケーションビルダー
ファイル 編集 アプリケーション オプション ヘルプ
コメント行検索...
アプリケーション名称
ボタン ID:8 KEY:"6"
ボタン ID:9 KEY:"7"
ボタン ID:10 KEY:"8"
ボタン ID:11 KEY:"9"
ボタン ID:12 KEY:"C"
ボタン ID:13 KEY:"+"
ボタン ID:14 KEY:"="
加算(+) ID:15 KEY:"加算(+)15"
アクションイベント
処理完了イベント
表示したい文字列を設定する
数値変換／左右オペランド設定後、演算を行う
数値に変換後、左オペランドに設定する
数値に変換後、右オペランドに設定する
演算を行う
数値入力フィールド ID:2 KEY:"数値入力フィールド2"
【アイコン凡例】 アプリケーション 画面構成部品 ウィンドウ メニュー パネル 処理部品 複合(画面構成) 複合(処理) リモート
実行 実行(設定可) 画面編集 帳票編集 ロード 挿入 保存 上書き保存 クリア 終了

３）計算終了

計算が終了したら、次の計算のために左右のオペランドをクリアしておきます。

考え方

１．加算コンポーネントが計算を完了した時点で、左右のオペランドを「０」にクリアする。

接続確認

コンポーネント同士の接続を確認します。

計算結果を表示する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）	■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 （基礎編 Step2 で設定済み）	■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：処理結果データ

左右オペランドを０にクリアする❷

接続先コンポーネント❷	■加算(+)（ID:15）
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：0
＜引数１＞	説明：右オペランド 取得方法：固定値 メソッド／値：0

操 作

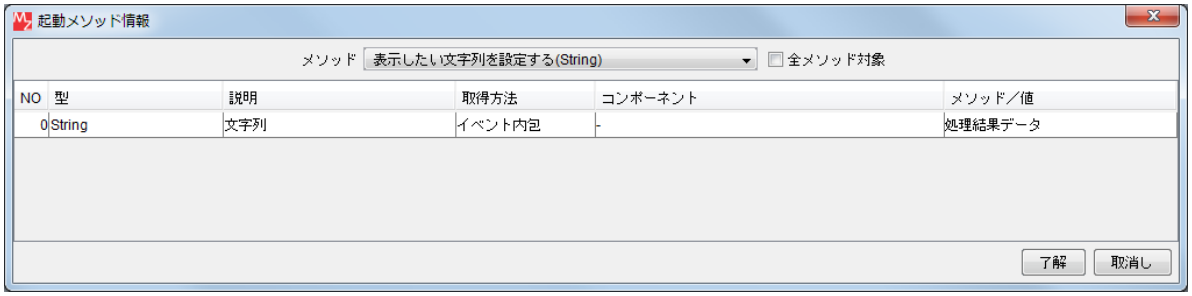
左右オペランドを「０」にクリアしましょう。

［加算］コンポーネントから２つのコンポーネントを接続します。

―――計算結果を表示する❶（基礎編 Step2 で設定済み）―――

① 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－［起動メソッド追加］とクリックします。空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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)］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。
［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。
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。

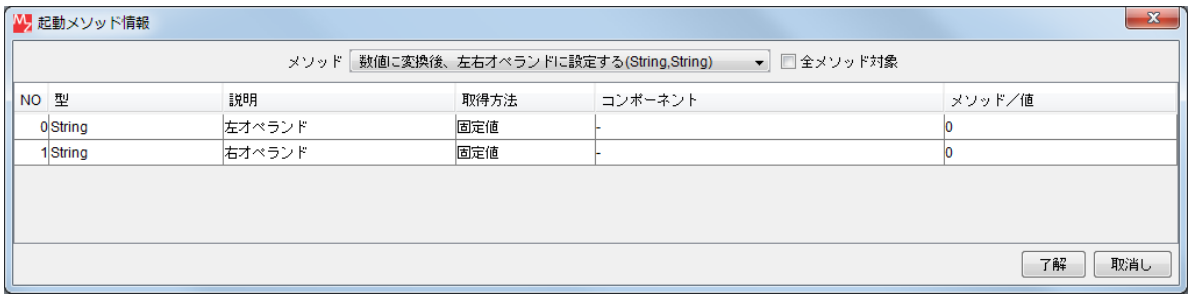


―――左右オペランドを０にクリアする❷―――

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－［起動メソッド追加］とクリックします。
空の四角い枠が追加されます。
右側に追加された空の四角い枠にコンポーネントを割り当てます。
右側に追加された空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:15)］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－［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。

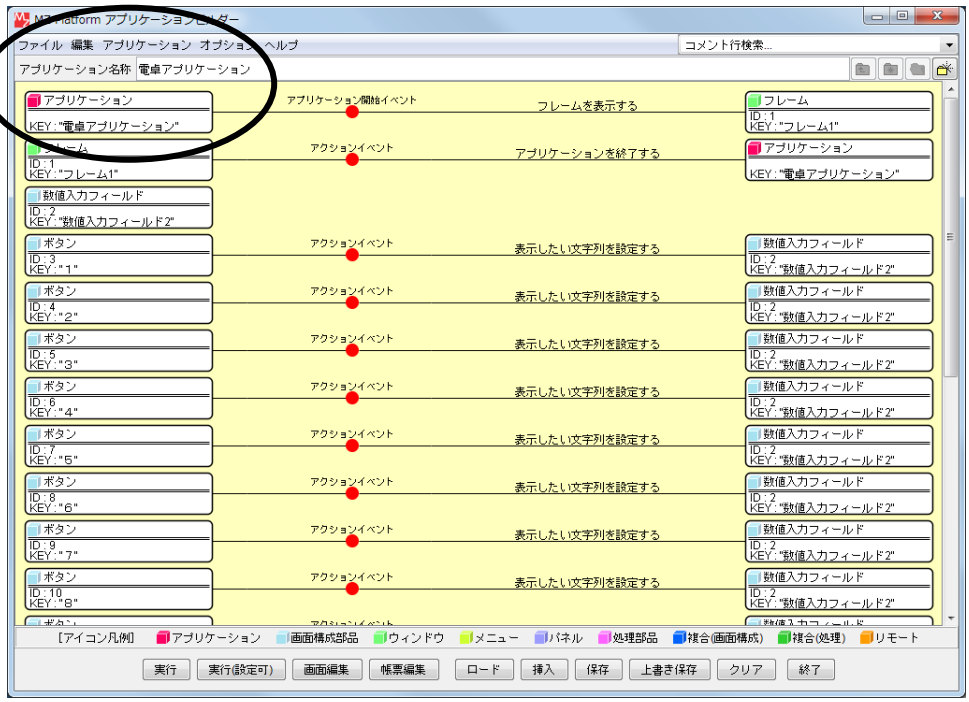
［数値に変換後、左右オペランドを設定する(String,String)］をクリックします。
引数０を設定します。
説明：左オペランド
取得方法：固定値
メソッド／値：0
引数１を設定します。
説明：右オペランド
取得方法：固定値
メソッド／値：0
設定後、了解ボタンをクリックします。



⑤　設定を確認しましょう。
実行（設定可）で実行します。
［C］クリアボタンを使用してみましょう。
連続計算してみましょう。

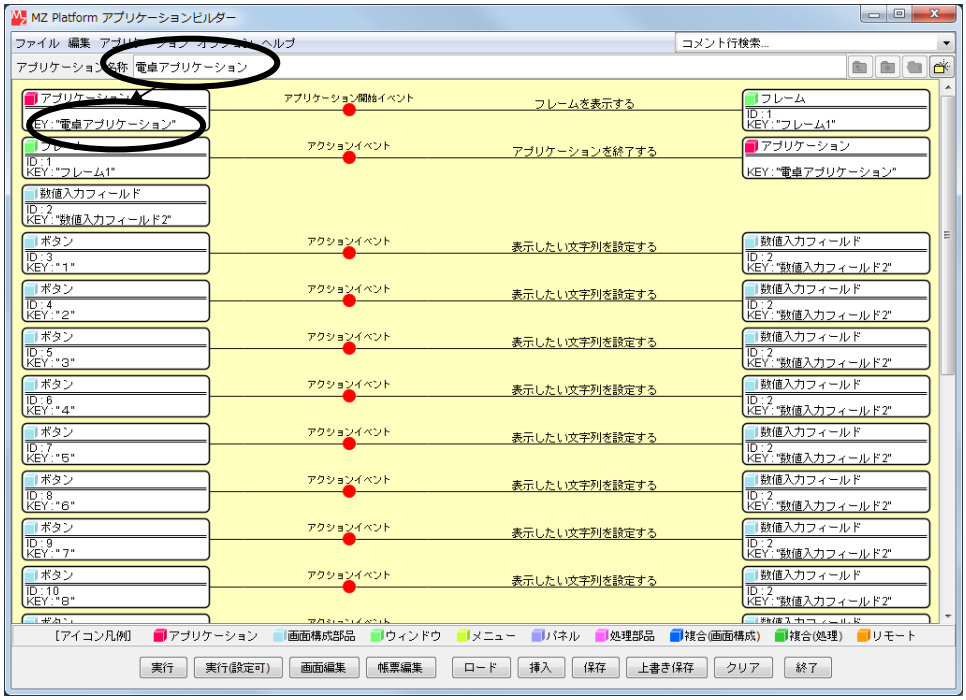
Step.4 アプリケーションの名称を付ける

作成したアプリケーションに名前を付けます。



操 作 アプリケーションに名前を設定します。

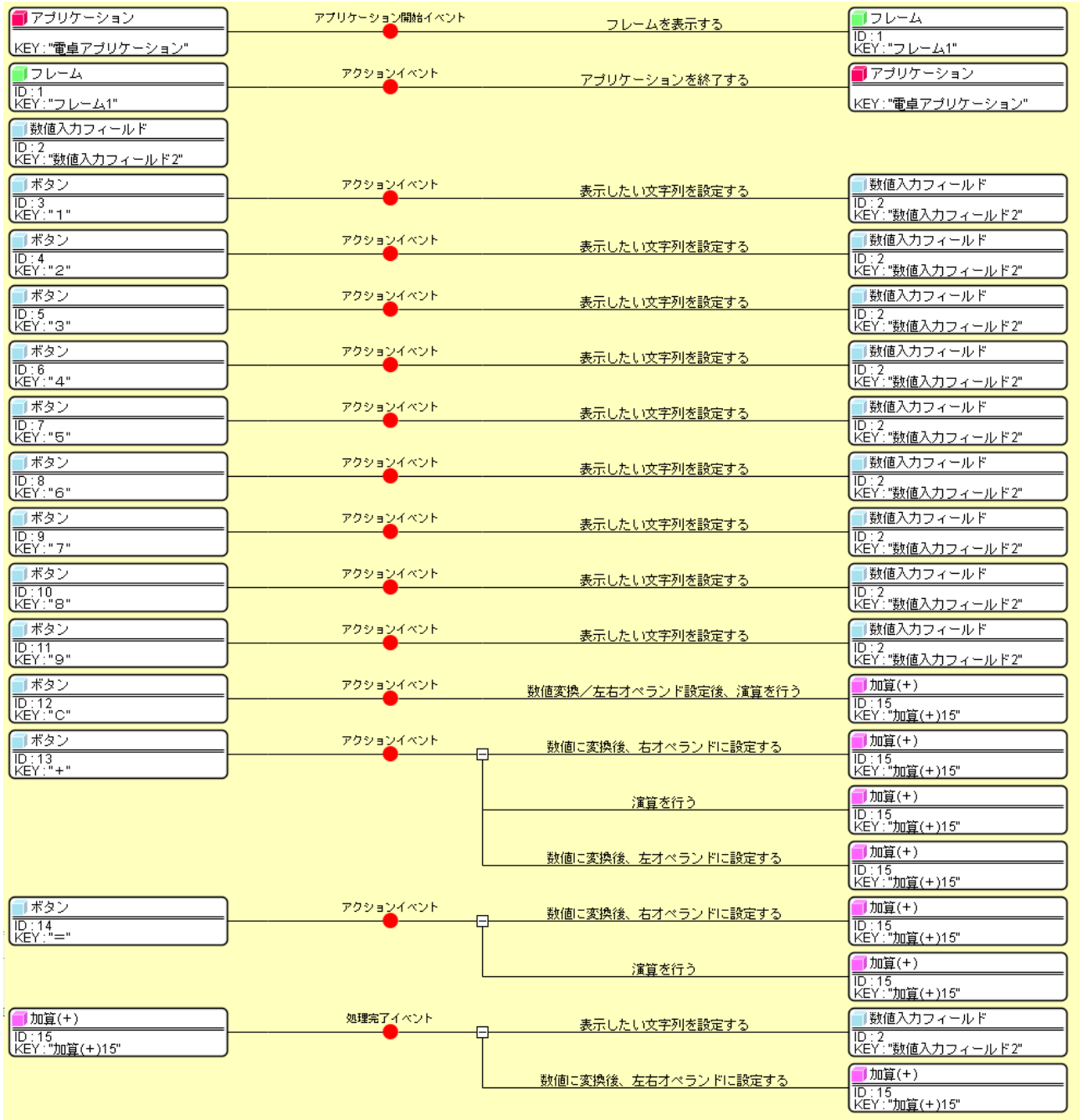
① ［アプリケーション名称］の欄に「電卓アプリケーション」と入力します。



自動的に［アプリケーション］コンポーネントの［Key］にも名前が入力されます。

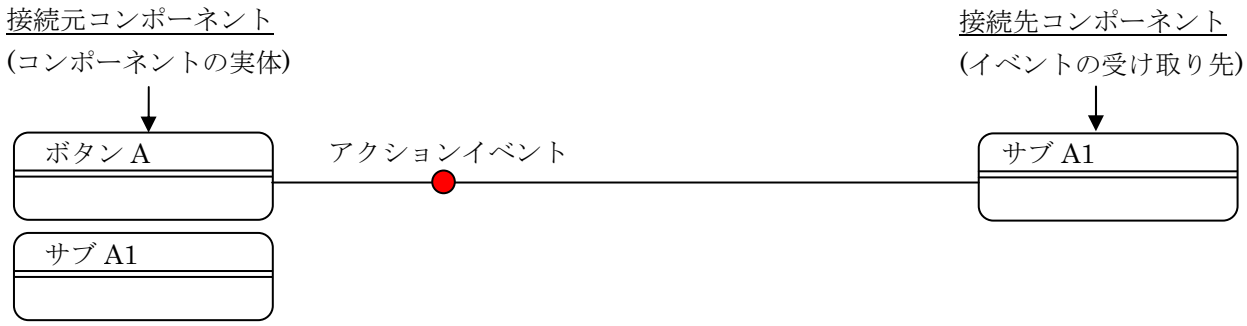
まとめ

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。



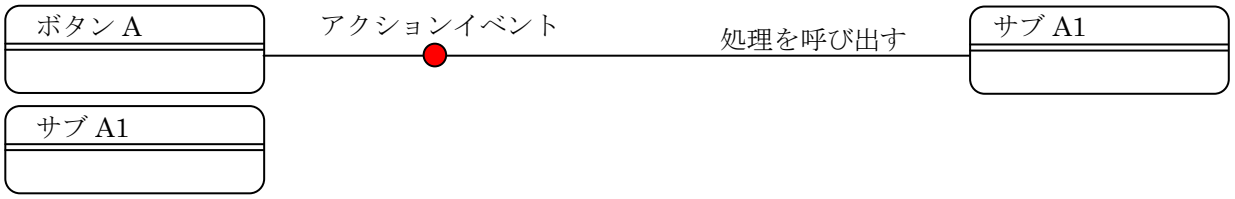
付録　MZ Platformにおけるコンポーネント動作とメソッド起動

MZ Platform のコンポーネントは、それぞれ決められたタイミングでイベントを発生する仕様となっています。イベントは、一種のメッセージであると考えることができます。例えば、ボタンコンポーネントの場合、画面上でボタンがクリックされたときに、「そのボタンがクリックされた」ということを知らせるためのアクションイベントが発生します。そのメッセージの受け取り先は、ビルダー画面上で、線をつなぐことによって指定します。



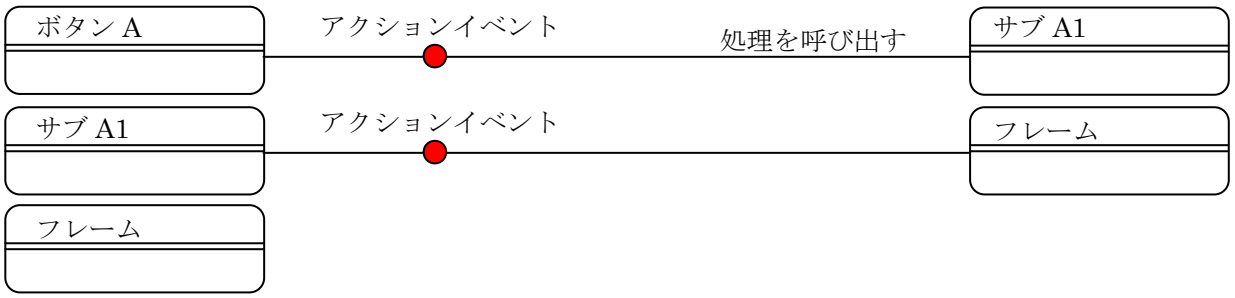
ビルダー画面上で上図のように記述されていた場合、これは、「[ボタン A]がクリックされたとき、そのことを知らせるために発生したアクションイベント（メッセージ）を[サブ A1]が受け取る。」ということを意味します。ビルダー画面上では、コンポーネントの実体は左側に配置されており、線でつながれた右側は、イベントの受け取り先を示しています。この左側のコンポーネントが「接続元コンポーネント」であり、右側が「接続先コンポーネント」となります。繰り返しとなりますが、コンポーネントが発生するイベントの種類と発生タイミングはコンポーネントごとに決まっています。例えば、テーブルコンポーネントでは、テーブルのセルが選択されると、「テーブルのセルが選択された」ということを知らせるためのデータ選択イベントが発生します。

さて、上の図では、アクションイベントの受け取り先が[サブ A1]コンポーネントであることは指定してありますが、イベントを受け取ったときに何をするかまでは指定していません。「イベントを受け取ったときに行わせる動作」を「メソッド」といいます。メソッドは、ビルダー画面上で接続先コンポーネント上でマウス右クリックし、表示されたメニューから[起動メソッド設定...]を選択し指定します。



上の図は、「[ボタン A]がクリックされたとき、そのことを知らせるために発生したアクションイベント（メッセージ）を[サブ A1]が受け取る。[サブ A1]は、イベントを受け取ったときに、『処理を呼び出す』メソッドを実行する。」ということを意味します。ここで、メソッドの中にはイベントを発生させるという動作を伴うものもある、ということに注意が必要です。例えば、サブルーチンコンポーネントは、「処理を呼び出す()」メソッドを実行すると、そのタイミングでアクションイベントを発生させます。

今、仮に[サブ A1]はサブルーチンコンポーネントだとして、「処理を呼び出す」メソッドを実行すると、そのタイミングでアクションイベントを発生させるものだとします。

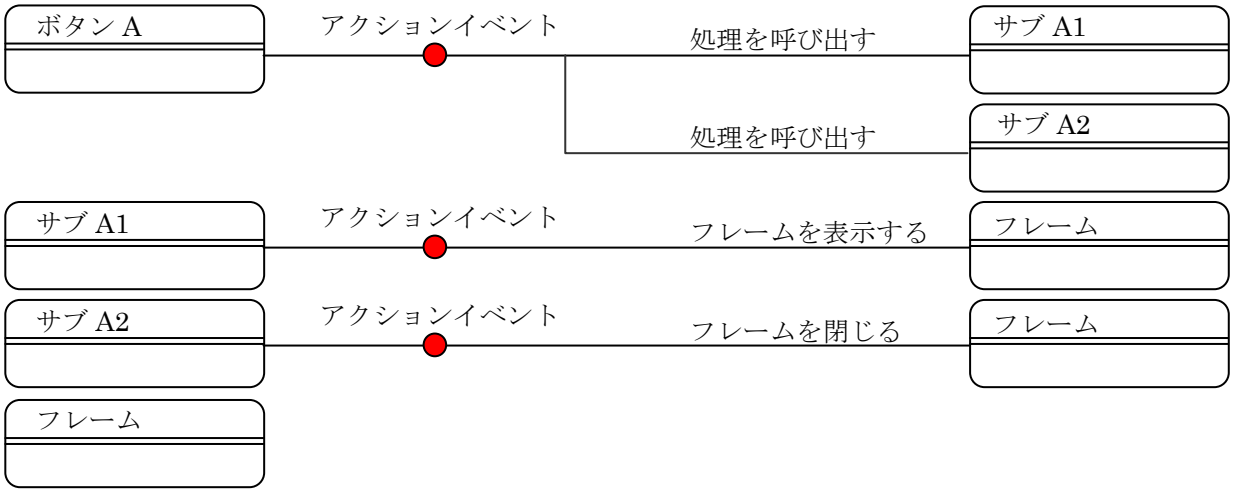


すると、この図は以下のことを意味します。

(1) [ボタン A]がクリックされたとき、そのことを知らせるためにアクションイベントが発生する。
(2) [ボタン A]から発生したアクションイベントを[サブ A1]が受け取り、「処理を呼び出す」メソッドを実行する。
(3) 「処理を呼び出す」メソッドの実行により、[サブ A1]からアクションイベントが発生する。
(4) [サブ A1]から発生したアクションイベントを[フレーム]が受け取る。

このように、MZ Platform では、最初に発生したイベントをきっかけとして、そのイベントを受け取ったコンポーネント（接続先コンポーネント）がメソッドを実行し、さらにそのメソッド実行により新しくイベントが発生して次の動作が行われる、という形で処理が進みます（このような処理実行形態を「イベント駆動型」と呼びます)。

以上が、接続元コンポーネント、発生イベント、接続先コンポーネント、および MZ Platform アプリケーションの動作形態の概要となります。１つのイベントに複数のコンポーネントが接続されている場合には、処理は上から順に実行されます。イベント発生を伴うメソッドが実行される場合には、一見、実行順序が複雑に見えますが、すべて上述の考え方でたどることができます。



この図の場合、処理は以下の順序で行われます（[サブ A2]はサブルーチンコンポーネントとします)。

(1) [ボタン A]がクリックされたとき、そのことを知らせるためにアクションイベントが発生する。
(2) [ボタン A]から発生したアクションイベントを[サブ A1]が受け取り、「処理を呼び出す」メソッドを実行

する。

(3) [サブ A1]の「処理を呼び出す」メソッドの実行により、[サブ A1]からアクションイベントが発生する。

(4) [サブ A1]から発生したアクションイベントを[フレーム]が受け取り、「フレームを表示する」メソッドを実行する。（この時点で、[サブ A1]の「処理を呼び出す」メソッド実行に関わる処理は完了。）

(5) [サブ A2]の「処理を呼び出す」メソッドが実行される。

(6) [サブ A2]の「処理を呼び出す」メソッドの実行により、[サブ A2]からアクションイベントが発生する。

(7) [サブ A2]から発生したアクションイベントを[フレーム]が受け取り、「フレームを閉じる」メソッドを実行する。