

設計・製造支援アプリケーション構築プラットフォーム

# AMZ Platform

バーコード入力導入チュートリアル



独立行政法人  
産業技術総合研究所

## 注意事項

独立行政法人 産業技術総合研究所（以下、産総研）は、本文書の記載内容に関して、一切の明示的または暗示的な責任または保証を負うものではありません。

本文書の一部または全部を、産総研の文書による承諾なく、無断で複写、複製、転載することを禁じます。本書の情報の使用に対して特許責任は一切負いません。また、本書は細心の注意を払って作成されていますが、産総研は、本書に掲載されている情報の正確性、誤り、記載漏れについての責任を負うものではありません。また、本書に掲載されている情報は、将来予告なしに変更される場合があります。

MZ Platform をいかなる第三者に対しても、産総研の許可なく貸与、譲渡、その他の方法で使用させたり、担保に供したりしてはなりません。また産総研の許可なく、別のハードウェアに複製を作成することも禁じます。

本書中の社名、商品名などは、各社の登録商標または商標です。

# バーコード入力アプリケーション

このアプリケーションは、USB 接続のバーコードリーダからの入力を受け取り、その入力文字列から製品コードとロット番号を読み取ったのち、これに対応する製品名ならびにロット番号をテキストフィールドに設定するものです。本書は MZ Platform の基本的な操作ができる方を対象に記載しております。コンポーネントの接続方法や起動メソッドの設定方法等は「開発チュートリアル(基礎編)」または「アプリケーションビルダー操作説明書」をご参照ください。

## 目次

Lesson.1 バーコード入力アプリケーションの作成 .....	2
Step.1 概要 .....	2
Step.2 画面作成 .....	3
Step.3 アプリケーション開始/終了時の処理 .....	12
Step.4 製品名を設定する .....	15
Step.5 ロット番号を設定する .....	24
Step.6 [バーコード入力] テキストフィールドのクリアと入力フォーカスの設定 .....	29
Step.7 バーコードサンプル一覧の印刷 .....	32

# Lesson.1 バーコード入力アプリケーションの作成

## Step.1 概要

このアプリケーションは、USB 接続のバーコードリーダからの入力を受け取り、その入力文字列から製品コードとロット番号を読み取ったのち、これに対応する製品名ならびにロット番号をテキストフィールドに設定するものです。本書では USB 接続のバーコードリーダからの入力の受け取り方、および、受け取った文字列の扱い方の例を示します。

### 1) バーコードの仕様

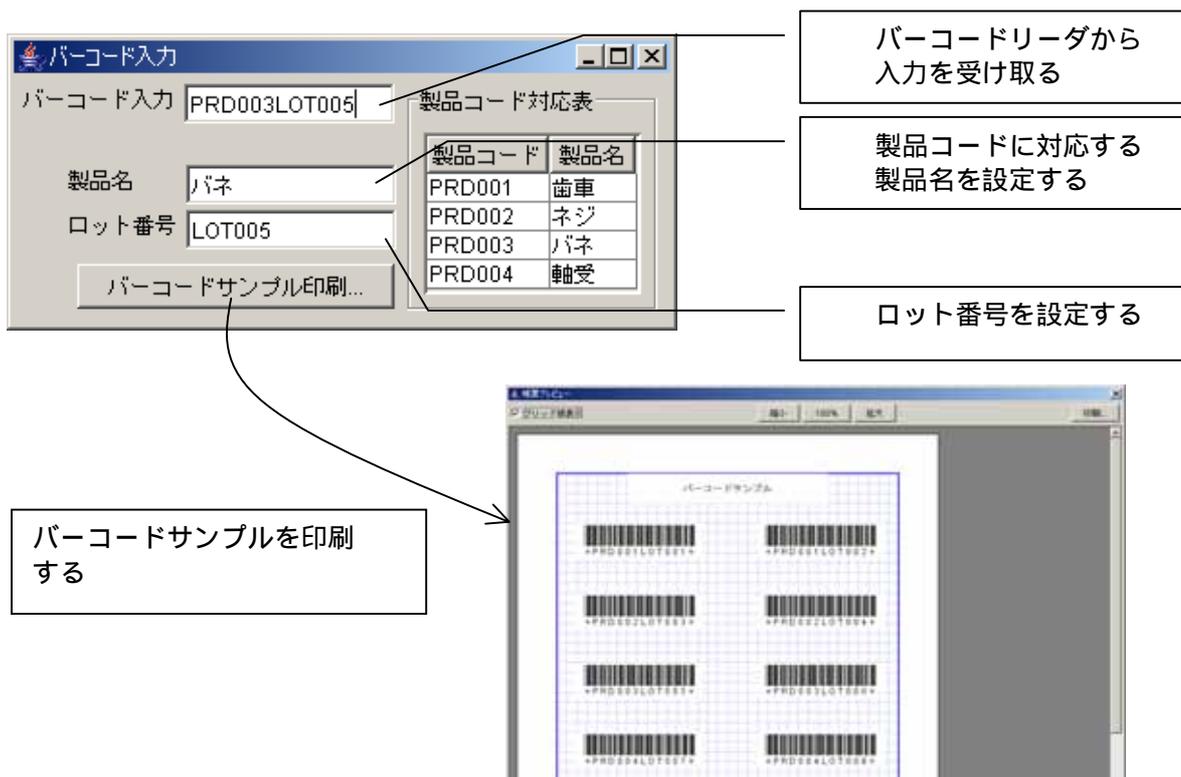
このアプリケーションではコード体系は CODE39 を採用しています。アプリケーションは、バーコードリーダをパソコンの USB ポートに接続し、コードを CODE39 に設定して使用します。なお、バーコードリーダの設定方法はそれぞれのバーコードリーダに付属のマニュアルをご覧ください。

コード体系	CODE39
書式	PRDxxxLOTyyy の 12 文字。 前半 6 文字が製品コードを示し、後半 6 文字がロット番号を示す。 例) <u>PRD003LOT005</u> 製品コード    ロット番号

### 2) USB 接続のバーコードリーダ

現在市販されている USB 接続のバーコードリーダからの入力は、キーボード入力として扱われます。したがって、バーコードを読み取る場合、そのときにフォーカス（カーソル）のある場所にバーコード入力文字列が出力されます。例えば、メモ帳でテキスト編集をしているときにバーコードの読み取りを行うと、その場所にバーコード文字列が書き込まれます。そこで、このアプリケーションは [バーコード入力] テキストフィールドでバーコード入力文字列を受け取るように作成しています。

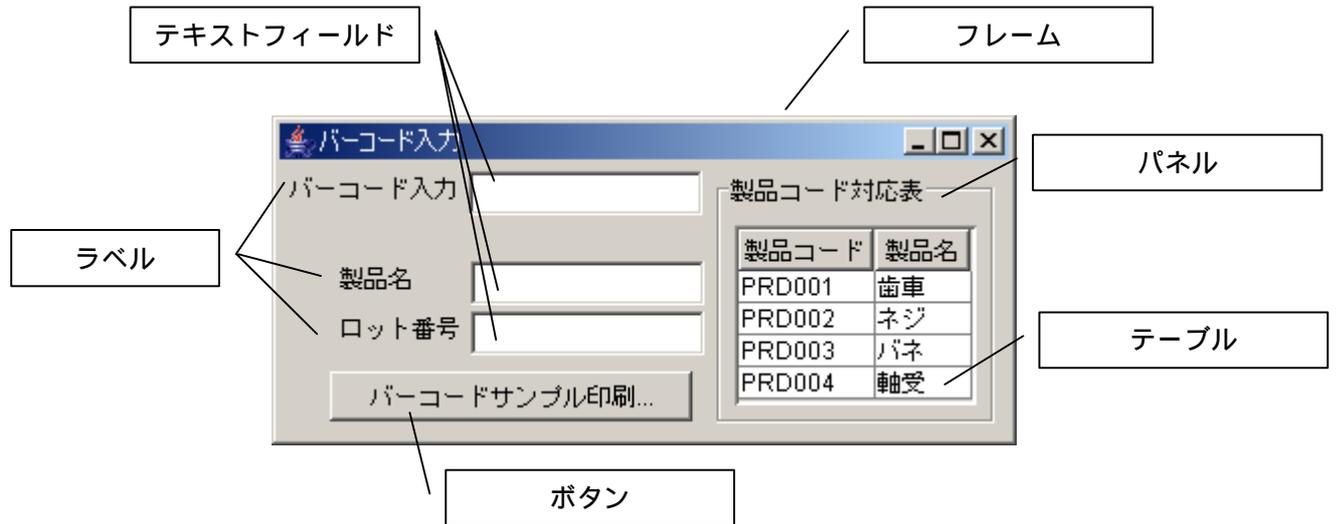
アプリケーションの機能概要は下図のとおりです。



## Step.2 画面作成

画面を構成する主なコンポーネントを追加し、画面を作成します。

### 完成図



### 準備

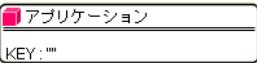
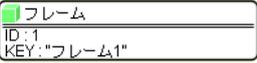
ここでは以下のコンポーネントを使用します。

アプリケーション KEY:""	テキストフィールド ID:6 KEY:"テキストフィールド6"
フレーム ID:1 KEY:"フレーム1"	テキストフィールド ID:7 KEY:"テキストフィールド7"
パネル ID:2 KEY:"パネル2"	テキストフィールド ID:8 KEY:"テキストフィールド8"
ラベル ID:3 KEY:"ラベル3"	ボタン ID:9 KEY:"ボタン9"
ラベル ID:4 KEY:"ラベル4"	テーブル ID:10 KEY:"テーブル10"
ラベル ID:5 KEY:"ラベル5"	

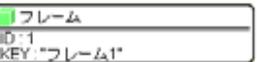
**接続確認**

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 KEY: ""
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	接続先 :  フレーム ID: 1 KEY: "フレーム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 KEY: "フレーム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	接続先 :  アプリケーション KEY: ""  起動メソッド : アプリケーションを終了する()

## 操作

必要なコンポーネントを追加し、画面を作成しましょう。

必要なコンポーネントを一括追加します。

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。

右側の領域に表示されるコンポーネントの分類を切り替え、次のように必要なコンポーネントの追加数を必要数分入力します。

[画面構成部品] パネル - [ウィンドウ] - [フレーム] の追加数に「1」と入力します。

[画面構成部品] パネル - [パネル] - [パネル] の追加数に「1」と入力します。

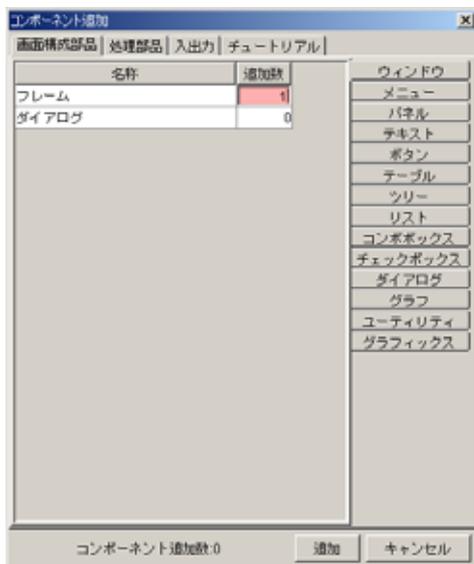
[画面構成部品] パネル - [テキスト] - [ラベル] の追加数に「3」と入力します。

[画面構成部品] パネル - [テキスト] - [テキストフィールド] の追加数に「3」と入力します。

[画面構成部品] パネル - [ボタン] - [ボタン] の追加数に「1」と入力します。

[画面構成部品] パネル - [テーブル] - [テーブル] の追加数に「1」と入力します。

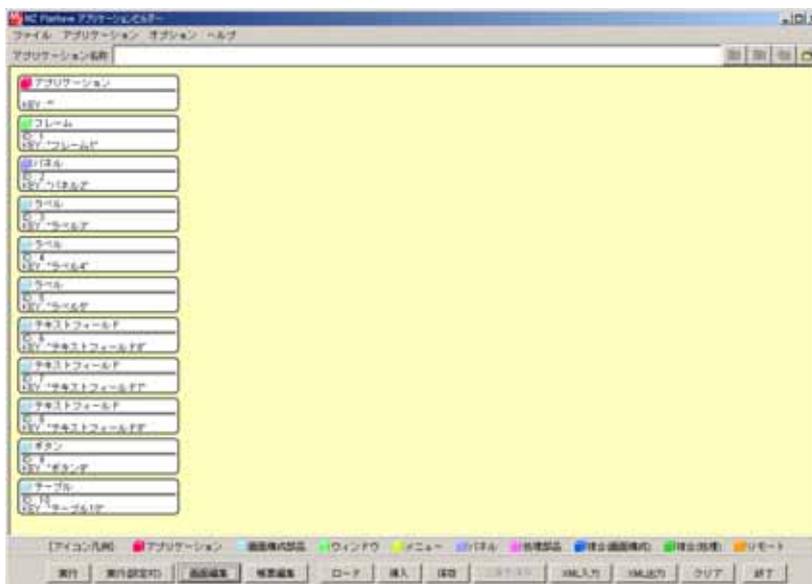
最後に [追加] ボタンをクリックします。



確認



コンポーネントが追加されます。



[パネル] コンポーネントのコンポーネントキーを変更します。

[パネル] コンポーネント上で右クリック - [属性情報設定...] とクリックします。

[BorderTitle] に「製品コード対応表」と入力し [設定] ボタンをクリックします。

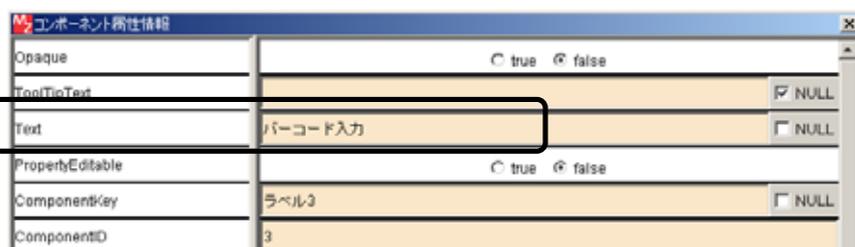


[ラベル ( ID:3)] コンポーネントのテキストを変更します。

[ラベル ( ID:3)] コンポーネント上で右クリック - [属性情報設定...] とクリックします。

[Text] に「バーコード入力」と入力し、[設定] ボタンをクリックします。

同様に、[ラベル ( ID:4)] のテキストを「製品名」、[ラベル ( ID:5)] のテキストを「ロット番号」に変更します。



[テキストフィールド (ID:6)] コンポーネントのコンポーネントキーを変更します。

[テキストフィールド (ID:6)] コンポーネント上で右クリック - [属性情報設定...] とクリックします。 [ComponentKey] に「バーコード入力」と入力し、[設定] ボタンをクリックします。

同様に、[テキストフィールド (ID:7)] のコンポーネントキーを「製品名」、[テキストフィールド (ID:8)] のコンポーネントキーを「ロット番号」に変更します。

ComponentKey	バーコード入力	<input type="checkbox"/> NULL
ComponentID	6	

ComponentKey	製品名	<input type="checkbox"/> NULL
ComponentID	7	

ComponentKey	ロット番号	<input type="checkbox"/> NULL
ComponentID	8	

[ボタン] コンポーネントのテキストを変更します。

[ボタン] コンポーネント上で右クリック - [属性情報設定...] とクリックします。

[Text] に「バーコードサンプル印刷...」と入力し、[設定] ボタンをクリックします。

ToolTipText		<input type="checkbox"/> NULL
Text	バーコードサンプル印刷...	<input type="checkbox"/> NULL
PropertyEditable	<input checked="" type="radio"/> true <input type="radio"/> false	
ComponentKey	バーコードサンプル印刷...	<input type="checkbox"/> NULL
ComponentID	9	

[テーブル] コンポーネントのコンポーネントキーを変更します。

[テーブル] コンポーネント上で右クリック - [属性情報設定...] とクリック

します。 [ComponentKey] に「製品コード対応表」と入力し、[設定] ボタンをクリックします。

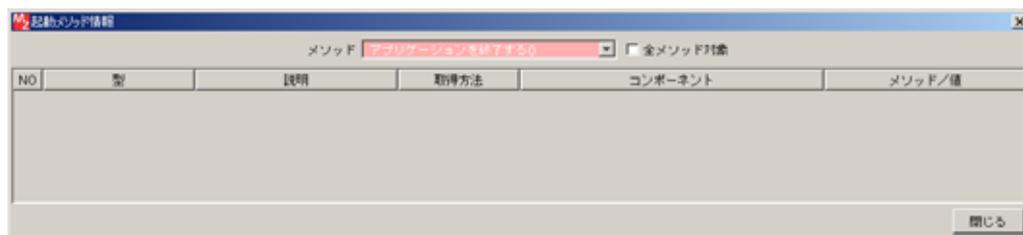
ToolTipText		<input type="checkbox"/> NULL
PropertyEditable	<input type="radio"/> true <input checked="" type="radio"/> false	
ComponentKey	製品コード対応表	<input type="checkbox"/> NULL
ComponentID	10	

[アプリケーション] コンポーネントから [フレーム] コンポーネントに接続します。

NO	型	説明	取得方法	コンポーネント	メソッド/値

接続先コンポーネント ([フレーム] コンポーネント) の起動メソッドの設定画面

[フレーム] コンポーネントから [アプリケーション] コンポーネントに接続します。



### 接続先コンポーネント ([アプリケーション] コンポーネント) の起動メソッドの設定画面

画面を作成します。

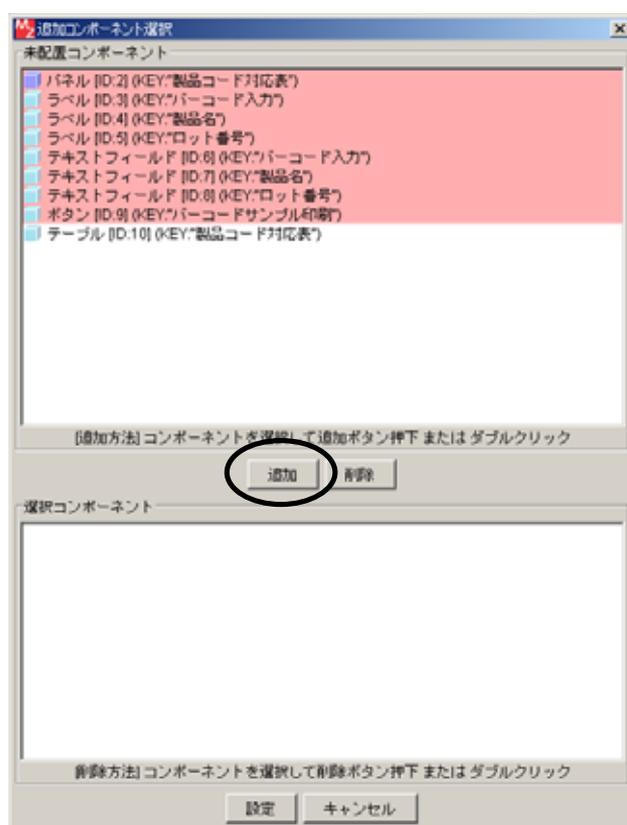
[画面編集] をクリックします。

[パネル] コンポーネント、[ラベル] コンポーネント、[テキストフィールド] コンポーネント、  
[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 - [コンポーネント一括追加] とクリックします。

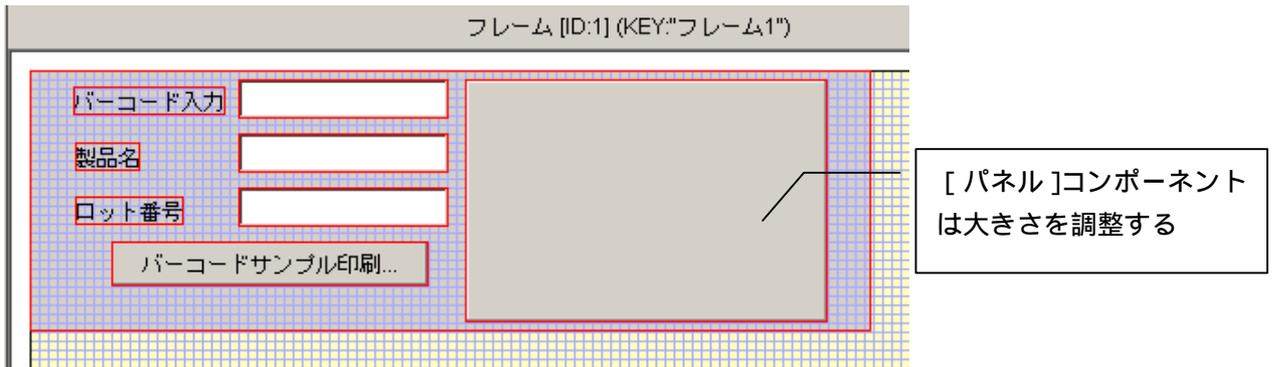
[テーブル] コンポーネントを除く [未配置コンポーネント] を全て [選択コンポーネント] に追加します。

[パネル] コンポーネントをクリックし、[ボタン] コンポーネントを【Shift】キーを押しながらクリックします。[追加] ボタンをクリックします。

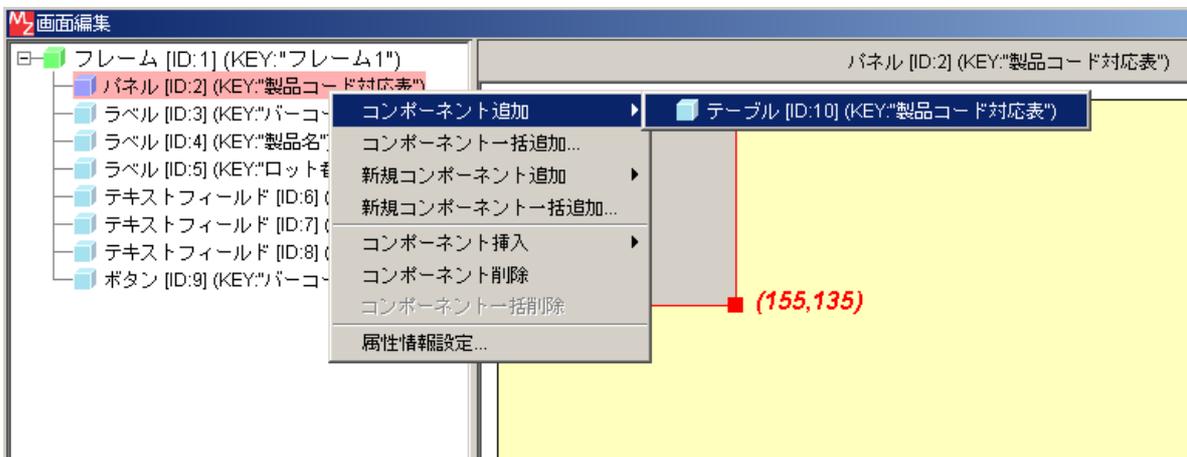


[設定] ボタンをクリックします。

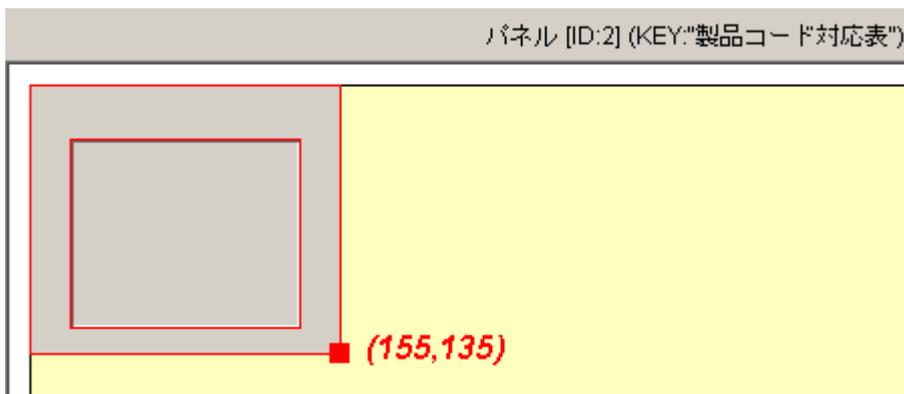
コンポーネントを配置し直します。配置を [ 手動配置 ] に切り替えます。  
次のように配置します。



[ テーブル ] コンポーネントをパネル内に追加します。  
[ パネル ] コンポーネント上で右クリック - [ コンポーネント追加 ] - [ テーブル ] とクリックします。



[ パネル ] コンポーネント内に [ テーブル ] コンポーネントを配置し直します。

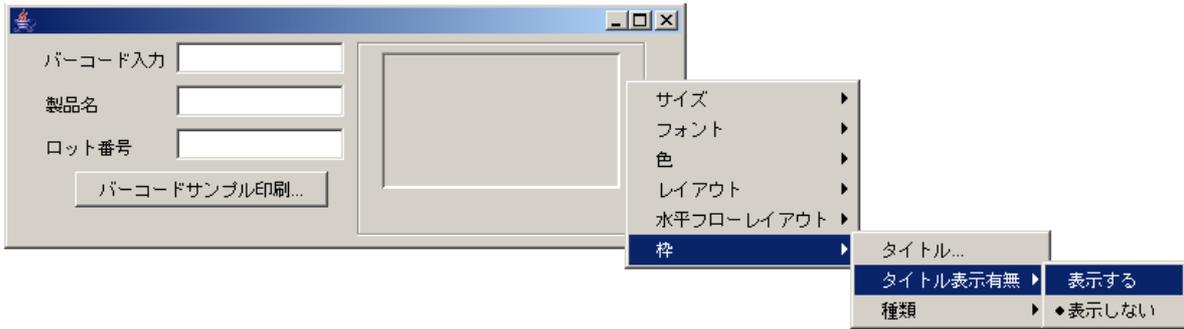


[ 閉じる ] ボタンをクリックし、ビルダー画面に戻ります。

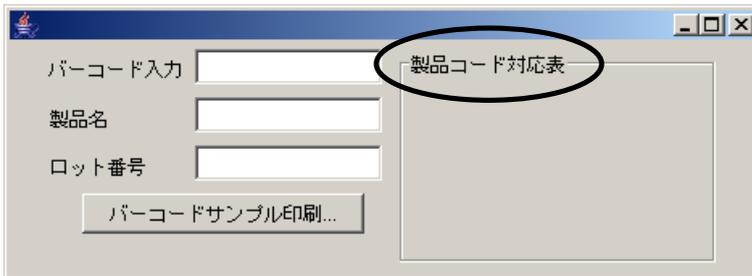
[ 実行 (設定可) ] ボタンをクリックし実行します。

パネルのタイトルを表示します。

パネル上で右クリック - [ 枠 ] - [ タイトル表示有無 ] - [ 表示する ] とクリックします。

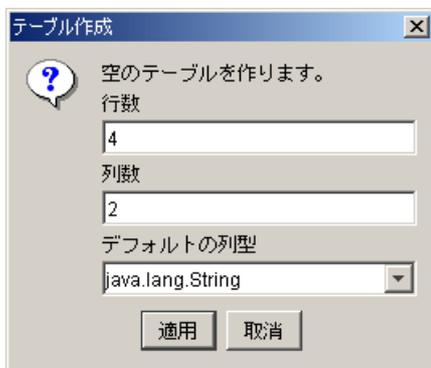


パネルにタイトルが表示されます。



テーブルを作成します。

[ テーブル ] コンポーネント上で右クリック - [ テーブル ] - [ テーブル作成... ] とクリックします。  
行数を「4」、列数を「2」とし、[ 適用 ] ボタンをクリックします。



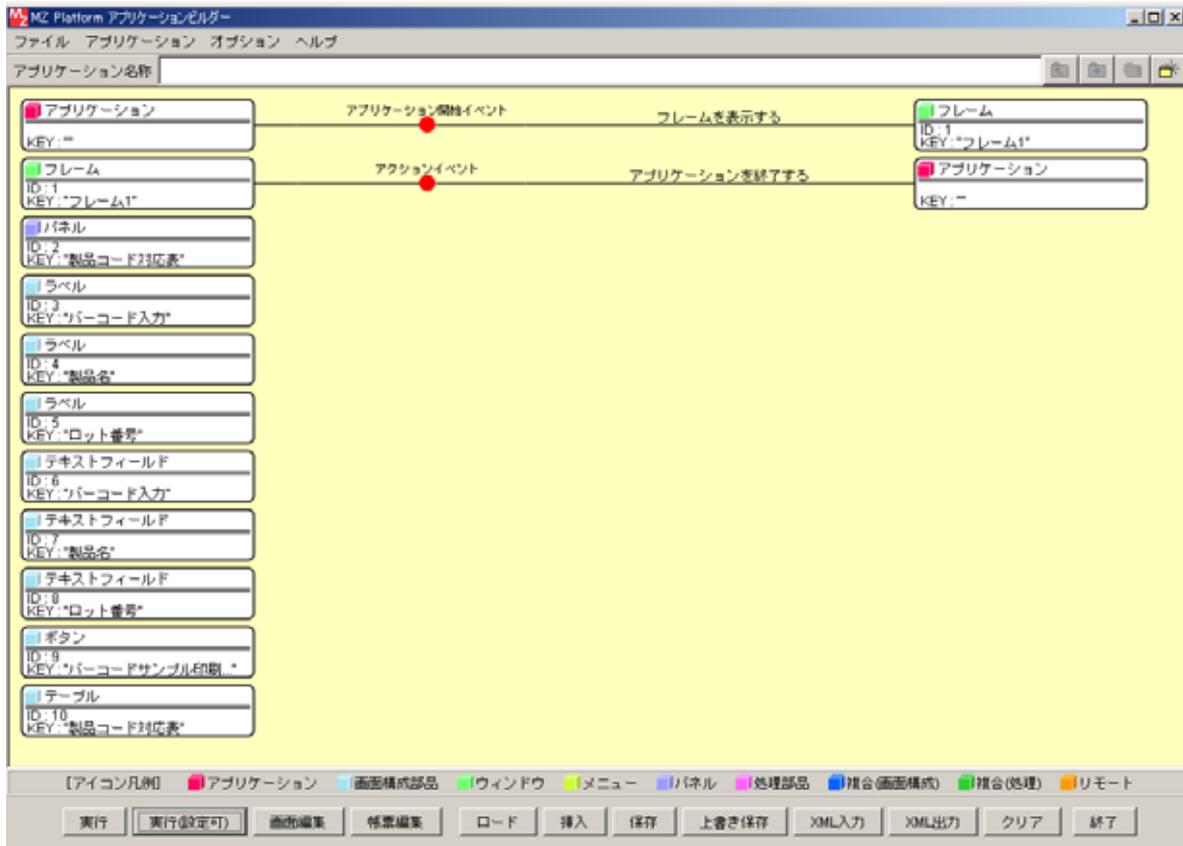
次のように列名およびデータを入力します。



確認後、ウィンドウを閉じます。

## まとめ:

ここまで進めると次のようになります。



### Step.3 アプリケーション開始/終了時の処理

アプリケーションの起動時には、入力フォーカス（カーソル）が [ バーコード入力 ] テキストフィールドに設定されるようにします。アプリケーション終了時には、[ バーコード入力 ] テキストフィールド、[ 製品名 ] テキストフィールド、[ ロット番号 ] テキストフィールドのデータを削除するようにします。

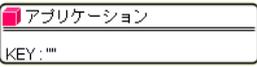
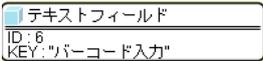
#### ■■■注意事項■■■

バーコードを読み込む際に、入力フォーカスが [ バーコード入力 ] テキストフィールドにないとアプリケーションは動作しません。ここで行う処理により、アプリケーション起動時には入力フォーカスが [ バーコード入力 ] テキストフィールドに自動的に設定されるようになりますが、その後テーブルや他のテキストフィールド等をクリックした場合には、改めて [ バーコード入力 ] テキストフィールドをクリックし、入力フォーカスを設定する必要があります。

#### 接続確認

コンポーネント同士の接続を確認します。

アプリケーション開始時に [ バーコード入力 ] テキストフィールドに入力フォーカスを設定する

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 )	
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	接続先 :  起動メソッド : requestFocus()

#### 操作

コンポーネント同士を接続しましょう。

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 ] 上でダブルクリックし、接続先コンポーネントとして [ テキストフィールド ( ID:6 ) ] コンポーネントを割り当てます。

接続したコンポーネントの処理を選びます。

日本語化されていないメソッドのため [ 全メソッド対象 ] チェックボックスをクリックし、一覧から [ requestFocus() ] をクリックします。

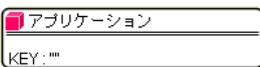
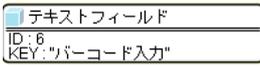


接続先コンポーネント ([ テキストフィールド ] コンポーネント) の起動メソッドの設定画面

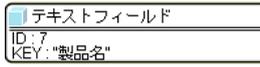
## 接続操作

コンポーネント同士を接続しましょう。

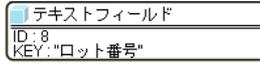
アプリケーション終了時に [ バーコード入力 ] テキストフィールドをクリアする①

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 )	
発生イベント	アプリケーション終了イベント
接続先コンポーネント①	接続先 :  起動メソッド : テキストを設定する(String) < 引数 > 説明 : テキスト 取得方法 : 固定値 メソッド / 値 : なし

アプリケーション終了時に [ 製品名 ] テキストフィールドをクリアする②

接続先コンポーネント②	接続先 :  起動メソッド : テキストを設定する(String) < 引数 > 説明 : テキスト 取得方法 : 固定値 メソッド / 値 : なし
-------------	--

アプリケーション終了時に [ ロット番号 ] テキストフィールドをクリアする③

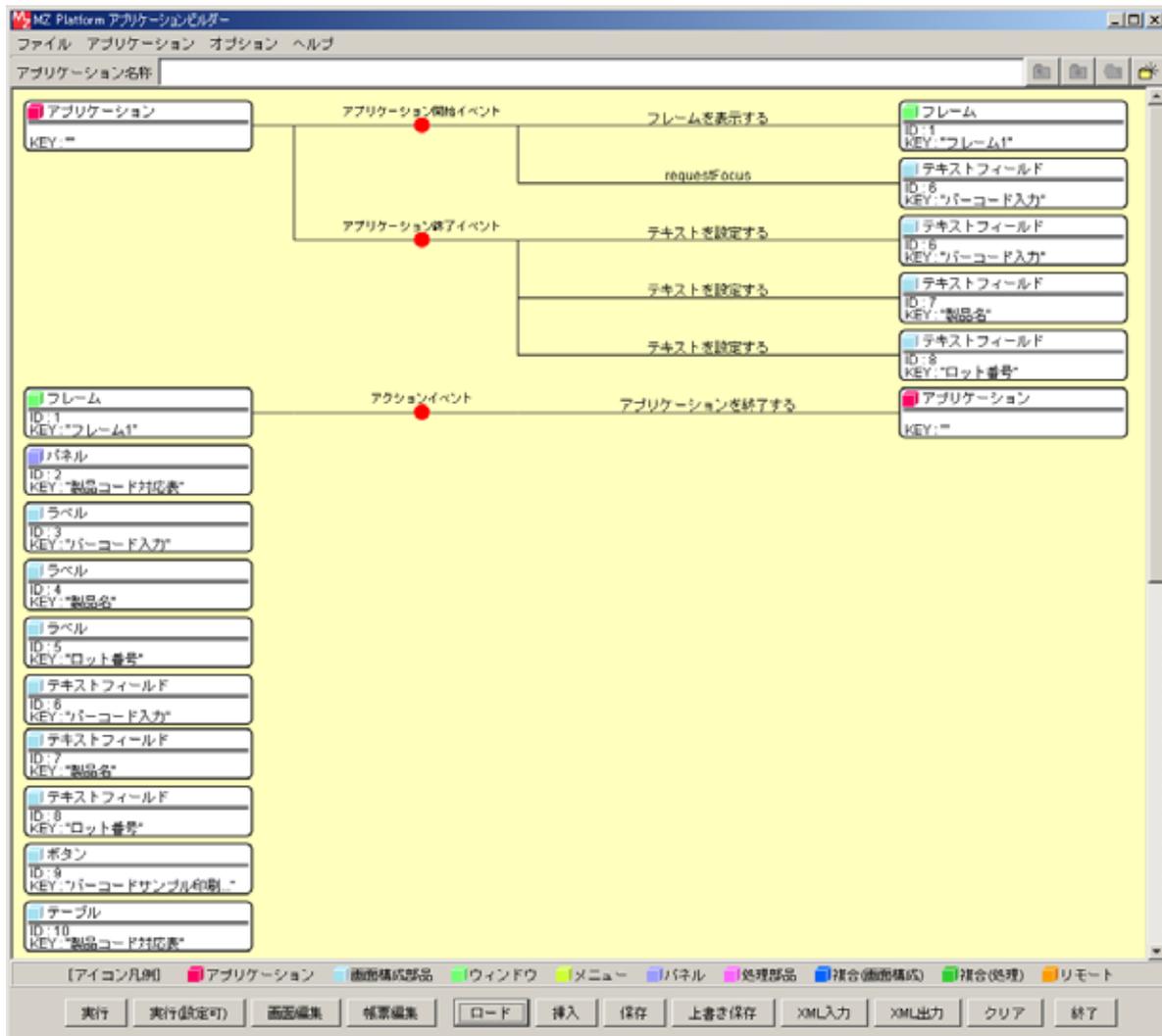
接続先コンポーネント③	接続先 :  起動メソッド : テキストを設定する(String) < 引数 > 説明 : テキスト 取得方法 : 固定値 メソッド / 値 : なし
-------------	---



接続先コンポーネント ([ テキストフィールド ] コンポーネント①②③) の起動メソッドの設定画面

## まとめ:

ここまで進めると次のようになります。



## Step.4 製品名を設定する

バーコードリーダからの入力文字列をもとに、製品名を設定します。

バーコードの書式は「PRDxxxLOTyyy」で、前半の6文字（PRD003等）が製品コード、後半の6文字（LOT005等）がロット番号を表すものとします。

ここでは、前半の6文字で表現される製品コードを[製品コード対応表]テーブルと照合し、対応する製品名を[製品名]テキストフィールドに設定します。

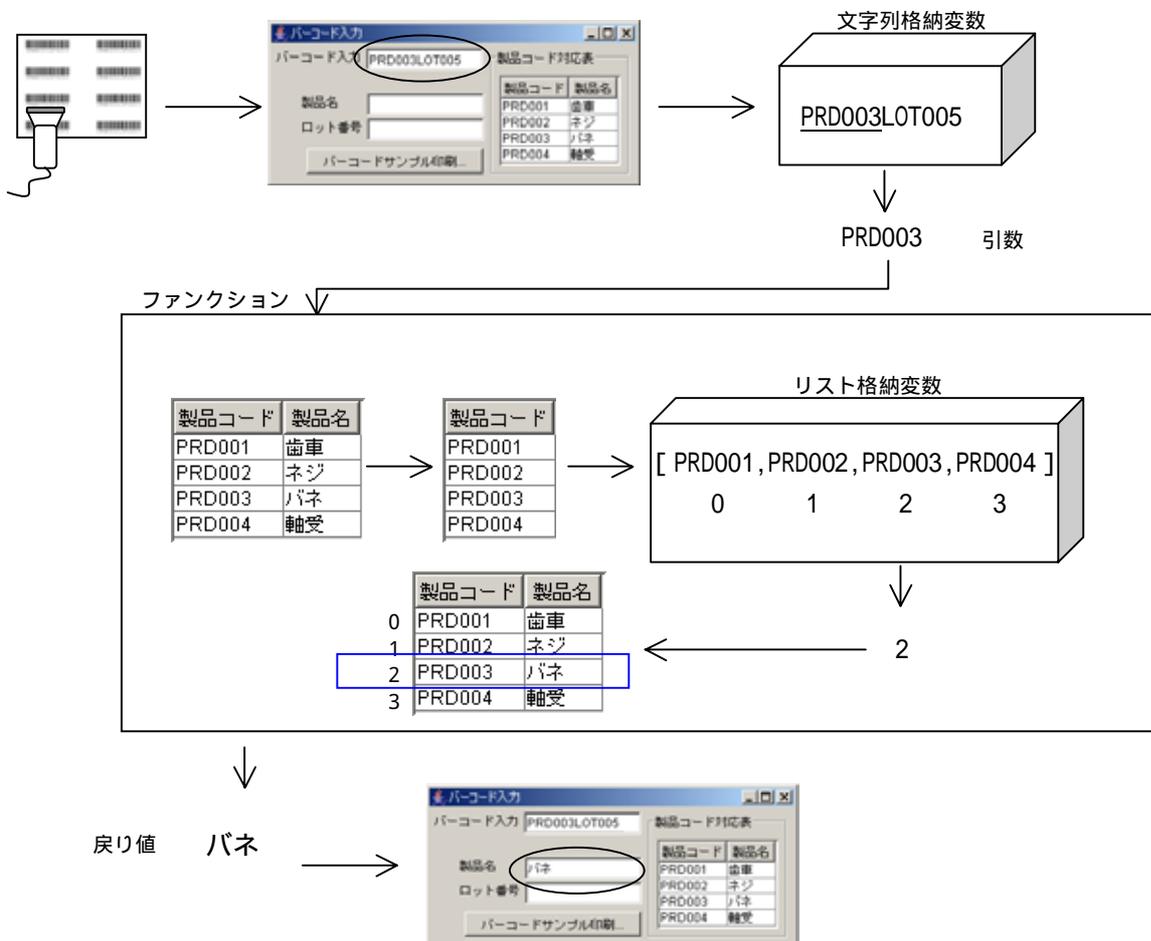
### 1) サブルーチンとファンクション

MZ Platform には処理をまとめるためのコンポーネントとして、『サブルーチン』『ファンクション』等があります。

サブルーチンとは、頻繁に利用する小さな処理を個別に作成し、必要に応じて呼び出すという手法です。[サブルーチン]コンポーネントでは、メソッドが呼び出された際に、接続先のコンポーネントの処理が実行されます。

ファンクションとは「関数」であり、入力値(引数)に対し、それに対応した出力(戻り値)の関係を示すものです。[ファンクション]コンポーネントでは、オブジェクトとして引数を指定してメソッドが呼び出された際に、接続先のコンポーネントの処理結果を得ることができます。また、接続先のコンポーネントのメソッドを実行する際に、[ファンクション]コンポーネントのメソッドの引数として指定されたオブジェクトを取得することができます。接続先コンポーネントのうち最後に接続されているコンポーネントのメソッドからの戻り値が処理結果として取得できます。

ここでは、サブルーチンとファンクションを使用し、製品名を設定する処理をまとめておきます。処理の流れは次のようになります。



## 完成図

製品コード	製品名
PRD001	歯車
PRD002	ネジ
PRD003	バネ
PRD004	軸受

**準備** ここでは以下のコンポーネントを追加します。

文字列格納変数 ID:11 KEY:"文字列格納変数11"	ファンクション ID:13 KEY:"ファンクション13"
サブルーチン ID:12 KEY:"サブルーチン12"	リスト格納変数 ID:14 KEY:"リスト格納変数14"

**操作** 必要なコンポーネントを追加しましょう。

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] - [文字列格納変数] とクリックします。

作業領域で右クリック - [コンポーネント追加] - [処理部品] - [サブルーチン] - [サブルーチン] とクリックします。

作業領域で右クリック - [コンポーネント追加] - [処理部品] - [サブルーチン] - [ファンクション] とクリックします。

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] - [リスト格納変数] とクリックします。

[サブルーチン] コンポーネントのコンポーネントキーを変更します。

[サブルーチン] コンポーネント上で右クリック - [属性情報設定...] とクリックします。

[ComponentKey] に「製品名設定」と入力し、[設定] ボタンをクリックします。

ComponentKey	製品名設定	<input type="checkbox"/> NULL
ComponentID	12	
ComponentKeys	日本語:      英語:	<input type="checkbox"/> NULL

設定

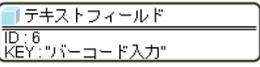
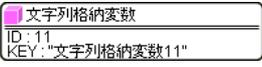
## 2) 製品名設定サブルーチン

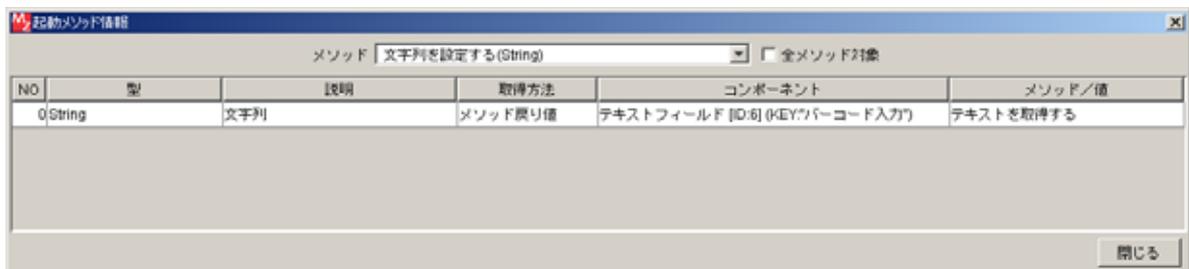
製品コードを [製品コード対応表] テーブルと照合し、製品名を [製品名] テキストフィールドに設定します。この処理を製品名設定サブルーチンとしてまとめておきます。

### 接続操作

コンポーネント同士を接続しましょう。

バーコード入力文字列を文字列格納変数に格納する

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)	
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	接続先 :  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 : メソッド戻り値 メソッド / 値 : テキストを取得する



接続先コンポーネント ([文字列格納変数] コンポーネント) の起動メソッドの設定画面

製品名設定サブルーチン呼び出す

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 )	<div style="border: 1px solid black; padding: 2px;"> <span style="background-color: #e0e0e0;">文字列格納変数</span>                      ID: 11                      KEY: "文字列格納変数11"                 </div>
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	接続先 : <div style="border: 1px solid black; padding: 2px;"> <span style="background-color: #e0e0e0;">サブルーチン</span>                      ID: 12                      KEY: "製品名設定"                 </div> 起動メソッド : 処理を呼び出す()



接続先コンポーネント ([ サブルーチン ] コンポーネント) の起動メソッドの設定画面

製品コード (前半 6 文字) を取得する ①

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 )	<div style="border: 1px solid black; padding: 2px;"> <span style="background-color: #e0e0e0;">サブルーチン</span>                      ID: 12                      KEY: "製品名設定"                 </div>
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 ①	接続先 : <div style="border: 1px solid black; padding: 2px;"> <span style="background-color: #e0e0e0;">文字列格納変数</span>                      ID: 11                      KEY: "文字列格納変数11"                 </div> 起動メソッド : substring(int, int) < 引数 0 > 取得方法 : 固定値 メソッド / 値 : 0 < 引数 1 > 取得方法 : 固定値 メソッド / 値 : 6



接続先コンポーネント ([ 文字列格納変数 ] コンポーネント) ①の起動メソッドの設定画面

ファンクションを呼び出す②

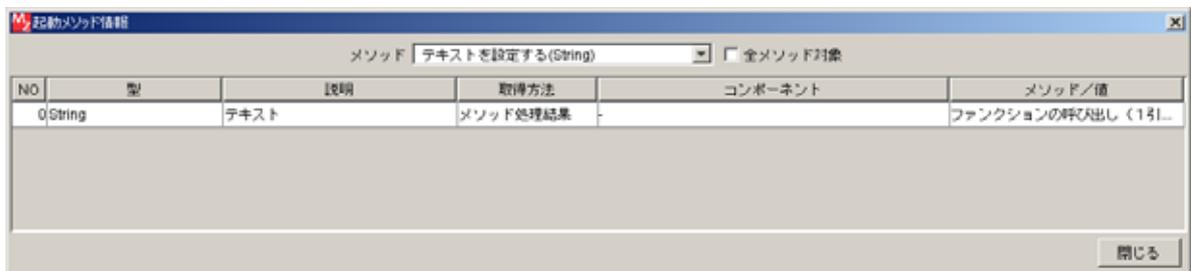
<p>接続先コンポーネント②</p>	<p>接続先：</p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <span style="background-color: #e0e0e0;">ファンクション</span>              ID: 13              KEY: "ファンクション13"         </div> <p>起動メソッド：</p> <p>ファンクションの呼び出し(1引数)(Object)</p> <p>&lt;引数&gt;</p> <p>説明：引数</p> <p>取得方法：メソッド処理結果</p> <p>メソッド/値：substring (文字列格納変数)</p>
--------------------	---

製品名を設定する③

<p>接続先コンポーネント③</p>	<p>接続先：</p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <span style="background-color: #e0e0e0;">テキストフィールド</span>              ID: 7              KEY: "製品名"         </div> <p>起動メソッド：</p> <p>テキストを設定する(String)</p> <p>&lt;引数&gt;</p> <p>説明：テキスト</p> <p>取得方法：メソッド処理結果</p> <p>メソッド/値：ファンクションの呼び出し(1引数) (ファンクション)</p>
--------------------	--



接続先コンポーネント ([ ファンクション ] コンポーネント②) の起動メソッドの設定画面



接続先コンポーネント ([ テキストフィールド ] コンポーネント③) の起動メソッドの設定画面

### 3) ファンクション

引数として与えられた製品コードを [製品コード対応表] テーブルと照合し、対応する製品名を取得します。

#### 接続操作

コンポーネント同士を接続しましょう。

製品コード対応表から製品コードの列 (第0列) を取得する①

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)	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <span style="color: purple;">■</span> ファンクション                      ID: 13                      KEY: "ファンクション13"                 </div>
発生イベント	処理要求イベント
接続先コンポーネント①	接続先 : <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <span style="color: blue;">■</span> テーブル                      ID: 10                      KEY: "製品コード対応表"                 </div> 起動メソッド : getColumnData(int) <引数> 取得方法 : 固定値 メソッド / 値 : 0

製品コードをリスト格納変数に格納する②

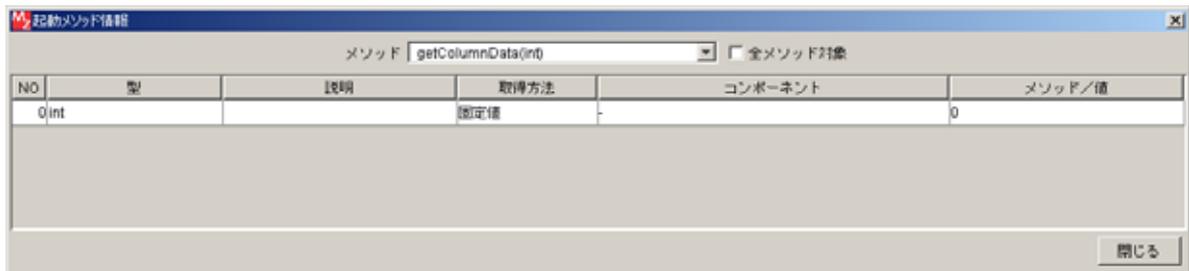
接続先コンポーネント②	接続先 : <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <span style="color: purple;">■</span> リスト格納変数                      ID: 14                      KEY: "リスト格納変数14"                 </div> 起動メソッド : リストを設定する (PFObjectList) <引数> 説明 : リスト 取得方法 : メソッド処理結果 メソッド / 値 : getColumnData (テーブル)
-------------	--

与えられた製品コードのインデックスを取得する③

接続先コンポーネント③	接続先 : <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;"> <span style="color: purple;">■</span> リスト格納変数                      ID: 14                      KEY: "リスト格納変数14"                 </div> 起動メソッド : indexOf(Object) <引数> 取得方法 : メソッド戻り値 コンポーネント : ファンクション メソッド / 値 : 第1引数の取得
-------------	---

製品名を取得する④

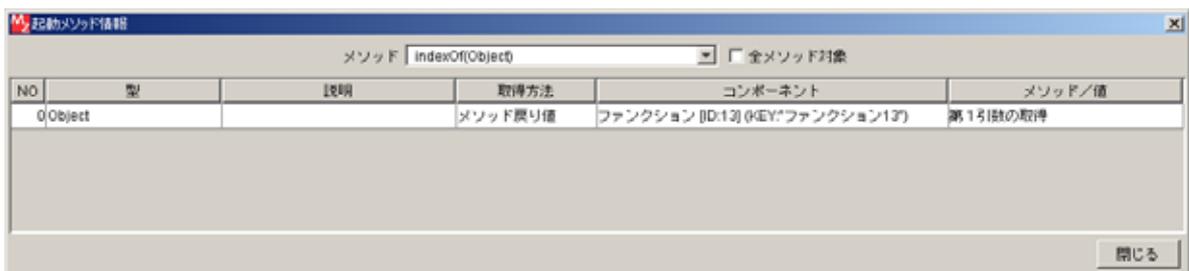
<p>接続先コンポーネント④</p>	<p>接続先：</p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 10px;"> <p>テーブル ID: 10 KEY: "製品コード対応表"</p> </div> <p>起動メソッド：</p> <p>getValueAt(int, int)</p> <p>&lt; 引数 0 &gt;</p> <p style="padding-left: 20px;">取得方法：メソッド処理結果</p> <p style="padding-left: 20px;">メソッド / 値：indexOf リスト格納変数)</p> <p>&lt; 引数 1 &gt;</p> <p style="padding-left: 20px;">取得方法：固定値</p> <p style="padding-left: 20px;">メソッド / 値：1</p>
--------------------	--



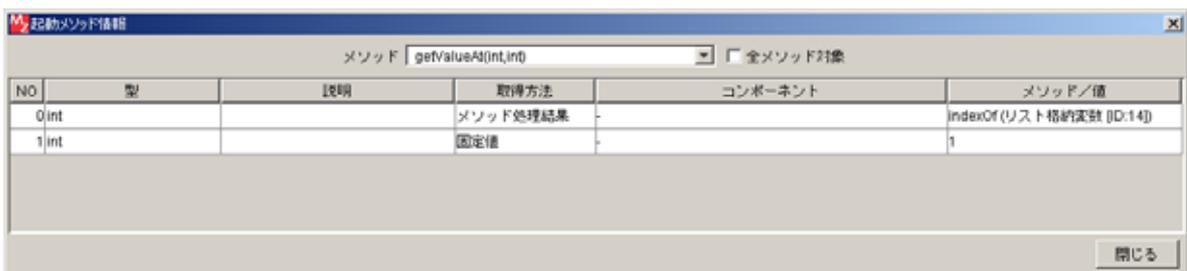
接続先コンポーネント ([ テーブル ] コンポーネント④) の起動メソッドの設定画面



接続先コンポーネント ([ リスト格納変数 ] コンポーネント⑤) の起動メソッドの設定画面



接続先コンポーネント ([ リスト格納変数 ] コンポーネント⑥) の起動メソッドの設定画面



接続先コンポーネント ([ テーブル ] コンポーネント④) の起動メソッドの設定画面

## 操作

製品名が設定される処理の動作を確認しましょう。

アプリケーションを実行します。

[実行] ボタンをクリックします。

[バーコード入力] テキストフィールドにキーボードから製品名とロット番号を入力します。

ここでは、「PRD003LOT005」と入力し【Enter】キーを押します。

確認

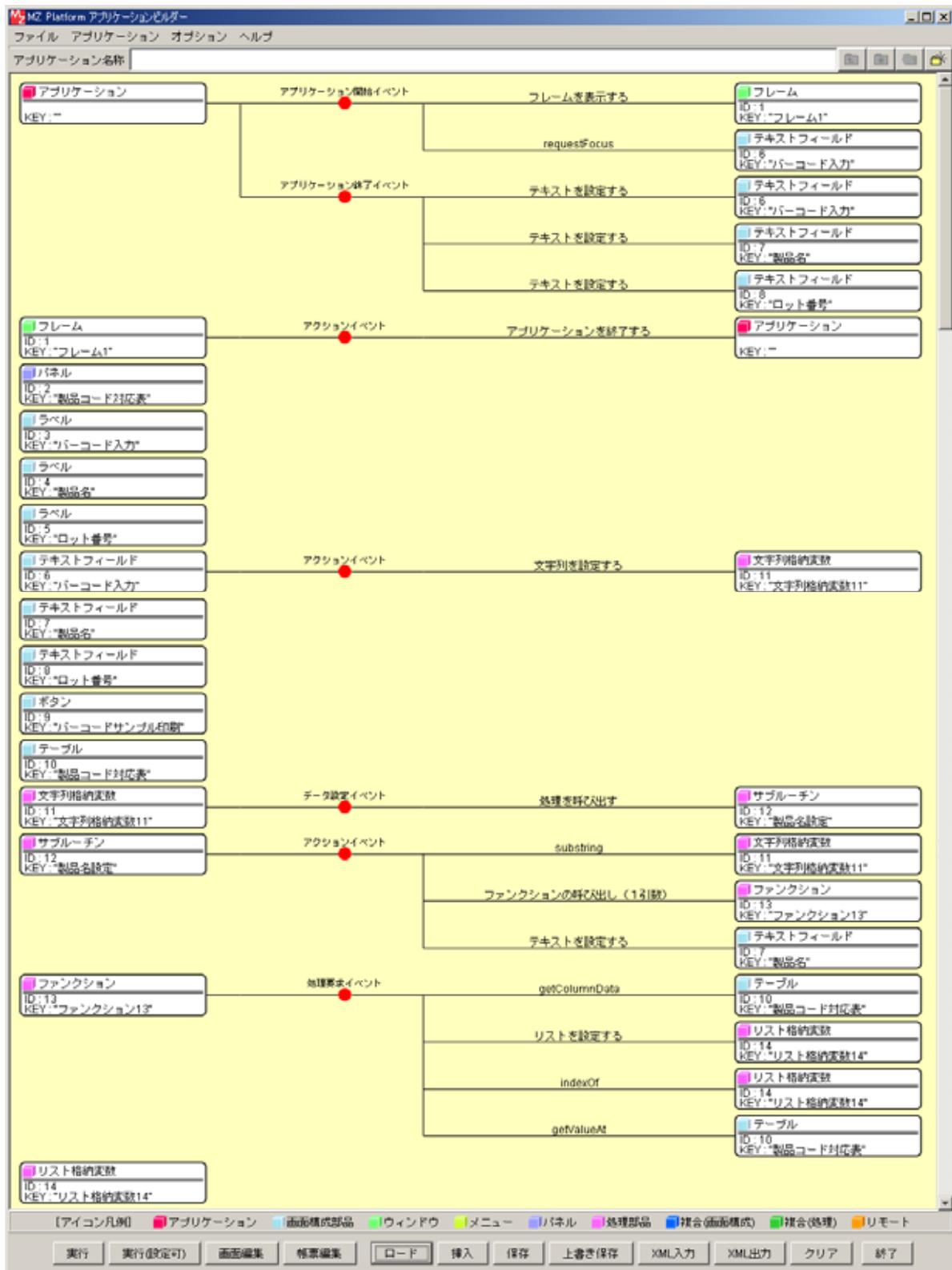


[製品名] テキストフィールドに「バネ」と表示されます。

製品コード	製品名
PRD001	歯車
PRD002	ネジ
PRD003	バネ
PRD004	軸受

## まとめ:

ここまで進めると次のようになります。



## Step.5 ロット番号を設定する

バーコードリーダからの入力文字列をもとに、製品名を設定します。  
バーコードの書式は「PRDxxxLOTyyy」で、前半の6文字（PRD003等）が製品コード、後半の6文字（LOT005等）がロット番号を表すものとします。

ここでは、後半の6文字で表現されるロット番号を[ロット番号]テキストフィールドに設定します。  
この処理をロット番号設定サブルーチンとしてまとめておきます。

### 完成図

製品コード	製品名
PRD001	歯車
PRD002	ネジ
PRD003	バネ
PRD004	軸受

### 準備

ここでは以下のコンポーネントを追加します。

サブルーチン
ID:15
KEY:"サブルーチン15"

### 操作

必要なコンポーネントを追加しましょう。

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] - [サブルーチン] - [サブルーチン] とクリックします。

[サブルーチン]コンポーネントのコンポーネントキーを変更します。

[サブルーチン]コンポーネント上で右クリック - [属性情報設定...] とクリックします。

[ComponentKey]に「ロット番号設定」と入力し、[設定]ボタンをクリックします。

ComponentKey	ロット番号設定	<input type="checkbox"/> NULL
ComponentID	15	
ComponentKeys	日本語: <input type="checkbox"/> NULL	英語: <input type="checkbox"/> NULL

## 接続操作

コンポーネント同士を接続しましょう。

ロット番号設定サブルーチン呼び出す

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)	<input type="text" value="文字列格納変数"/> ID: 11 KEY: "文字列格納変数11"
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	接続先 : <input type="text" value="サブルーチン"/> ID: 15 KEY: "ロット番号設定"  起動メソッド : 処理を呼び出す()



接続先コンポーネント ([ サブルーチン ] コンポーネント) の起動メソッドの設定画面

製品コード（後半6文字）を取得する①

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)	<div style="border: 1px solid black; padding: 2px;"> <span style="background-color: #e0e0e0;">サブルーチン</span>                      ID:15                      KEY:"ロット番号設定"                 </div>
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①	接続先： <div style="border: 1px solid black; padding: 2px;"> <span style="background-color: #e0e0e0;">文字列格納変数</span>                      ID:11                      KEY:"文字列格納変数11"                 </div> 起動メソッド： substring(int,int) <引数0> 取得方法：固定値 メソッド/値：6 <引数1> 取得方法：固定値 メソッド/値：12

ロット番号を設定する②

接続先コンポーネント②	接続先： <div style="border: 1px solid black; padding: 2px;"> <span style="background-color: #e0e0e0;">テキストフィールド</span>                      ID:8                      KEY:"ロット番号"                 </div> 起動メソッド： テキストを設定する(String) <引数> 説明：テキスト 取得方法：メソッド処理結果 メソッド/値：substring(文字列格納変数)
-------------	--



接続先コンポーネント ([文字列格納変数]コンポーネント)①の起動メソッドの設定画面



接続先コンポーネント (テキストフィールド)コンポーネント)②の起動メソッドの設定画面

## 操作

ロット番号が設定される処理の動作を確認しましょう。

アプリケーションを実行します。

[実行] ボタンをクリックします。

[バーコード入力] テキストフィールドにキーボードから製品名とロット番号を入力します。  
ここでは、「PRD003LOT005」と入力し【Enter】キーを押します。

確認

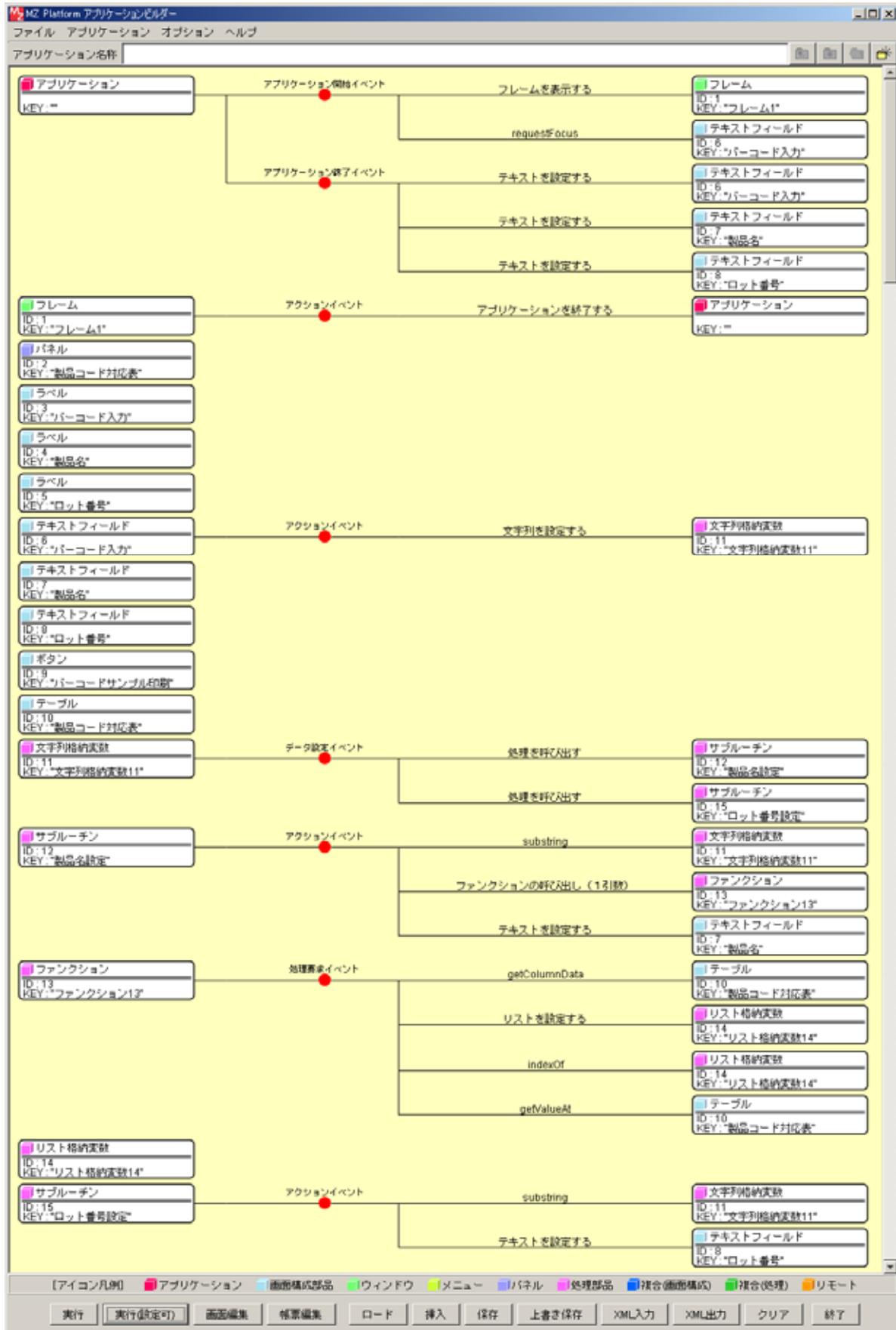


[ロット番号] テキストフィールドに「LOT005」と表示されます。

製品コード	製品名
PRD001	歯車
PRD002	ネジ
PRD003	バネ
PRD004	軸受

## まとめ:

ここまで進めると次のようになります。



## Step.6 [バーコード入力]テキストフィールドのクリアと入力フォーカスの設定

バーコードリーダからの入力文字列である製品名とロット番号を、対応する製品名ならびにロット番号テキストフィールドに設定した後は、次の入力に備え [バーコード入力] テキストフィールドのデータをクリアするように設定し、入力フォーカスをテキストフィールドに設定します。

### 完成図

### 接続操作

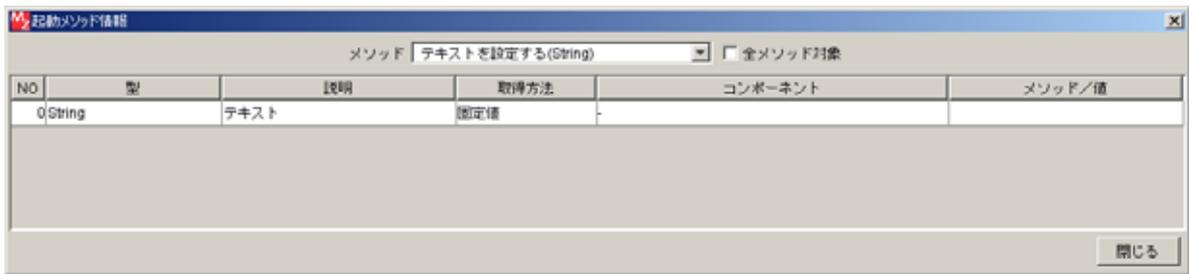
コンポーネント同士を接続しましょう。

#### [バーコード入力] テキストフィールドをクリアする①

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)	
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①	接続先 :  起動メソッド : テキストを設定する (String) <引数> 説明 : テキスト 取得方法 : 固定値 メソッド / 値 : なし

#### [バーコード入力] テキストフィールドに入力フォーカスを設定する②

接続先コンポーネント②	接続先 :  起動メソッド : requestFocus()
-------------	---



接続先コンポーネント ([テキストフィールド]コンポーネント①)の起動メソッドの設定画面



接続先コンポーネント ([テキストフィールド]コンポーネント②)の起動メソッドの設定画面

### 操作

バーコード入力テキストフィールドのデータがクリアされ、入力フォーカスが設定される処理の動作を確認しましょう。

アプリケーションを実行します。

[実行] ボタンをクリックします。

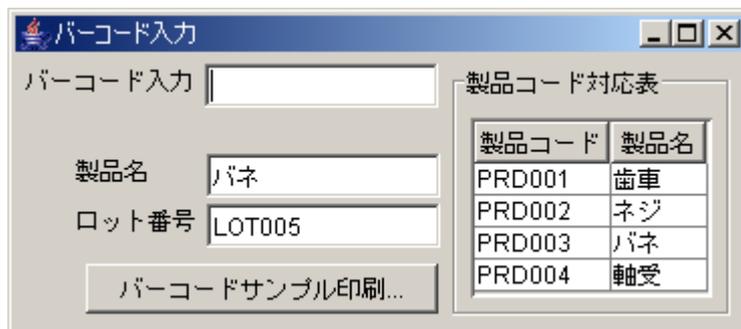
[バーコード入力] テキストフィールドにキーボードから製品名とロット番号を入力します。

ここでは、「PRD003LOT005」と入力し【Enter】キーを押します。

確認

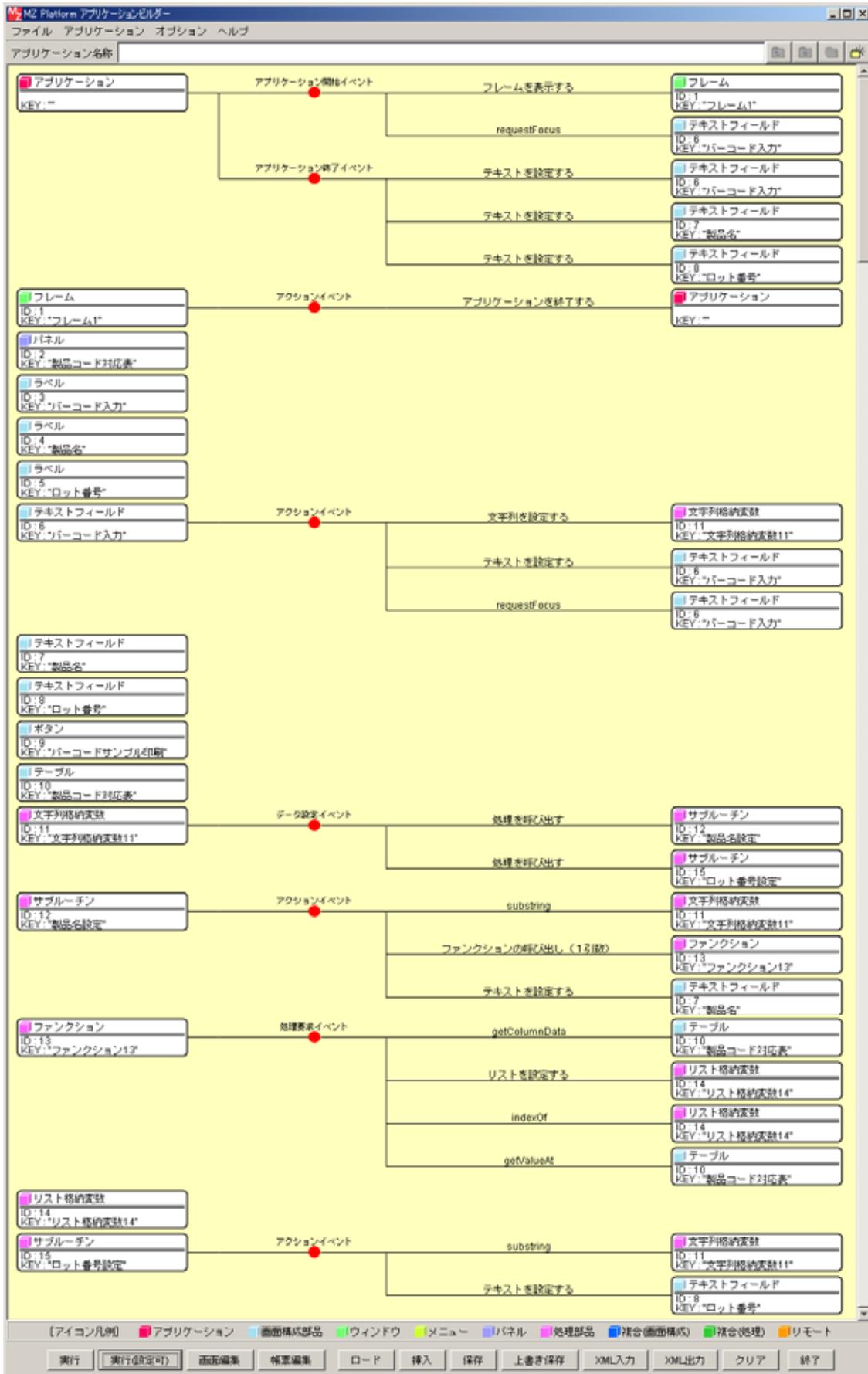


[バーコード入力] テキストフィールドのデータがクリアされ、入力フォーカスが表示されます。



## まとめ:

ここまで進めると次のようになります。



## Step.7 バーコードサンプル一覧の印刷

### 1) バーコード用サンプルデータの準備

バーコードのサンプル一覧を作成します。ここでは次の8つのバーコードを作成します。

PRD001LOT001、PRD001LOT002、PRD002LOT003、PRD002LOT004

PRD003LOT005、PRD003LOT006、PRD004LOT007、PRD004LOT008

#### 準備

ここでは以下のコンポーネントを追加します。

ラベル ID:16 KEY:"ラベル16"	ラベル ID:20 KEY:"ラベル20"
ラベル ID:17 KEY:"ラベル17"	ラベル ID:21 KEY:"ラベル21"
ラベル ID:18 KEY:"ラベル18"	ラベル ID:22 KEY:"ラベル22"
ラベル ID:19 KEY:"ラベル19"	ラベル ID:23 KEY:"ラベル23"

#### 操作

必要なコンポーネントを追加しましょう。

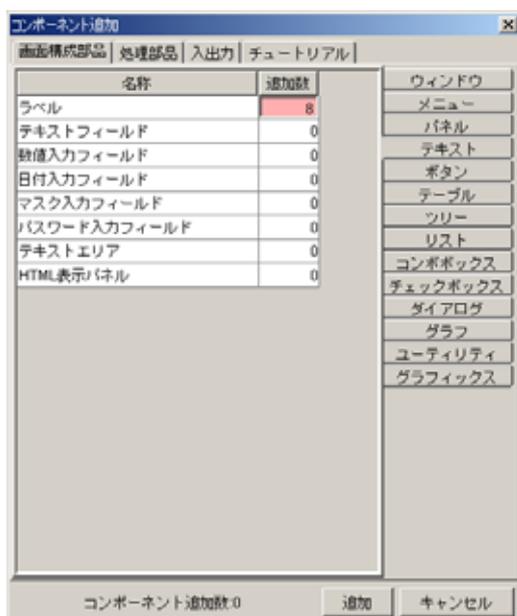
必要なコンポーネントを一括追加します。

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。

右側の領域に表示されるコンポーネントの分類を切り替え、次のように必要なコンポーネントの追加数を必要数分入力します。

[画面構成部品] パネル - [テキスト] - [ラベル] の追加数に「8」と入力します。

[追加] ボタンをクリックします。



[ラベル ( ID:16 )] コンポーネントのテキストを変更します。

[ラベル ( ID:16 )] コンポーネント上で右クリック - [属性情報設定...] とクリックします。

[Text ] に「PRD001LOT001」と入力し、[設定] ボタンをクリックします。

同様に、次のように [ラベル] コンポーネントのテキストを変更します。

コンポーネント ID	テキスト
ID:17	PRD001LOT002
ID:18	PRD002LOT003
ID:19	PRD002LOT004
ID:20	PRD003LOT005
ID:21	PRD003LOT006
ID:22	PRD004LOT007
ID:23	PRD004LOT008



## 2) [ 帳票 ] コンポーネント

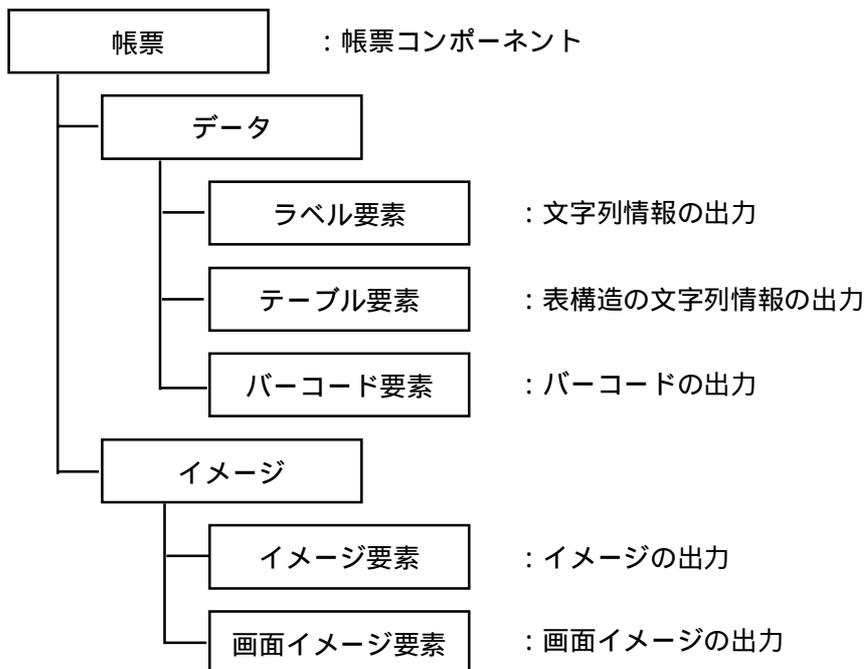
[ 帳票 ] コンポーネントは、帳票を作成して印刷する場合や、アプリケーションで扱う情報や画面そのものを印刷する場合に用います。

[ 帳票 ] コンポーネントには次の機能や属性があります。

機能	・印刷機能 ・印刷プロパティ（プリンタ選択 / 用紙設定 / 枚数設定等） ・印刷プレビュー機能（印刷イメージ表示 / 帳票レイアウト設定）
属性	・帳票サイズ ・帳票方向 ・帳票余白（上下左右）

## 3) 帳票のデータ構造

1枚の帳票は、次のような構造で構成されています。



#### 4) 帳票イメージの作成と印刷

帳票を印刷するには、印刷プレビュー画面を表示し、印刷プレビュー内の [印刷] ボタンをクリックして印刷する方法があります。[印刷] ボタンをクリックすると印刷設定ダイアログが表示されるので、プリンタの設定や印刷の設定を行った後、印刷が実行できます。

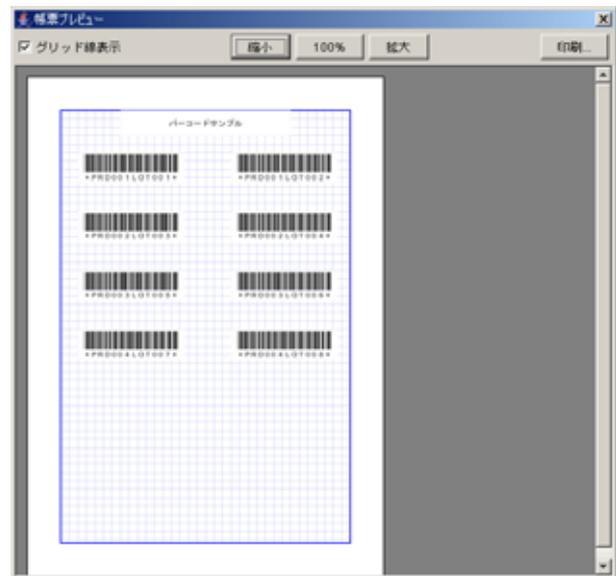
帳票のレイアウトは [帳票編集] 画面で作成します。

#### 完成図



印刷プレビュー画面表示後は [バーコード入力] テキストフィールドに入力フォーカス (カーソル) を表示する

クリックすると印刷プレビュー画面を表示する



#### 準備

ここでは以下のコンポーネントを追加します。



#### 操作

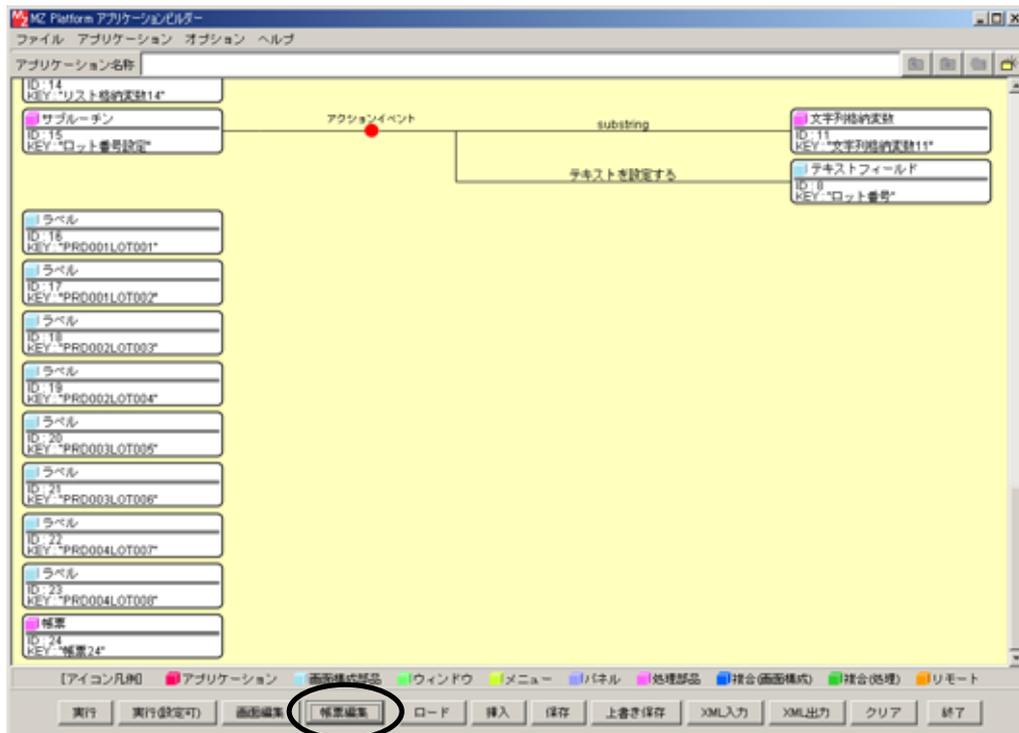
必要なコンポーネントを追加し、帳票イメージを作成しましょう。

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] - [入出力] - [帳票] - [帳票] とクリックします。

画面を作成します。

[ 帳票編集 ] ボタンをクリックします。

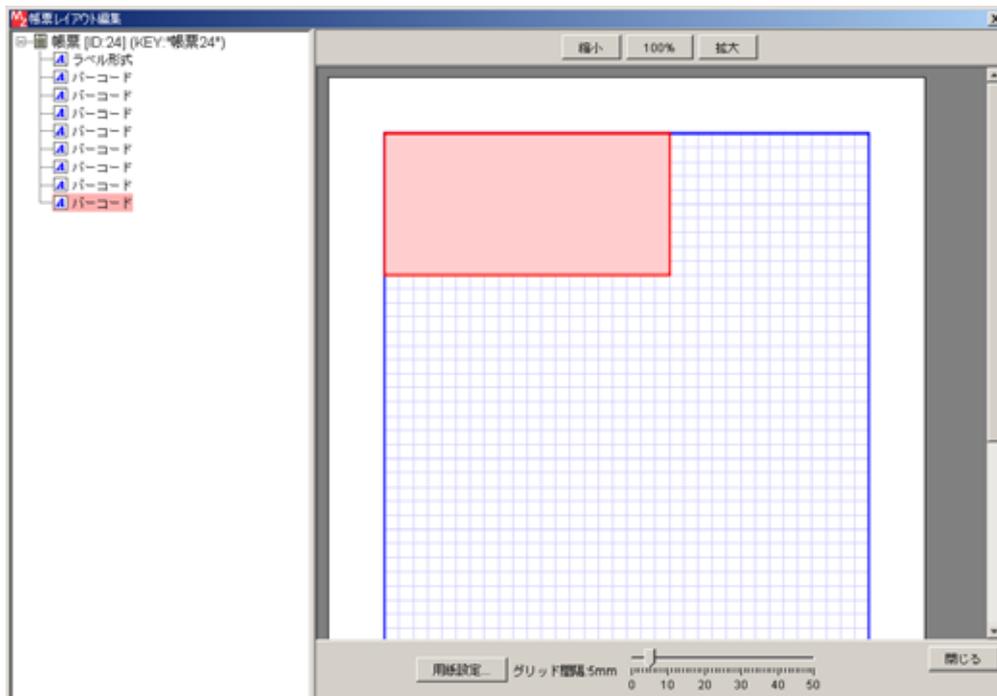


帳票要素をレイアウト画面に追加します。

ここでは、[ ラベル要素 ] を1つ、[ バーコード要素 ] を8つ追加します。

レイアウト画面上で右クリック - [ 帳票要素追加 ] - [ ラベル要素 ]

レイアウト画面上で右クリック - [ 帳票要素追加 ] - [ バーコード要素 ] とクリックします。



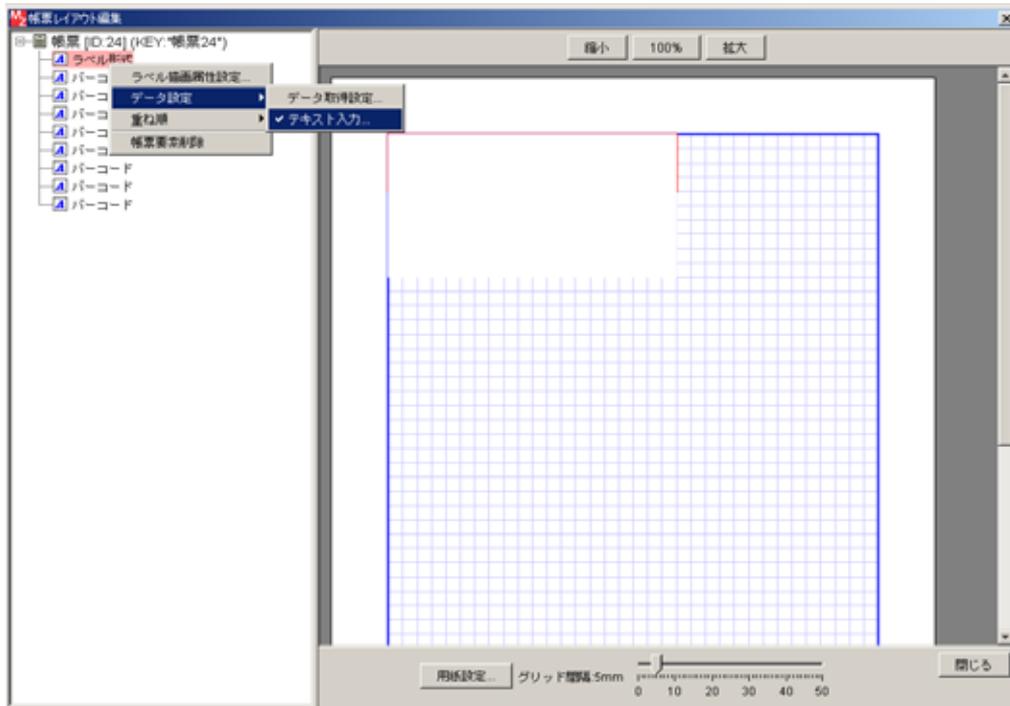
確認



レイアウト画面に [ ラベル要素 ] と [ バーコード要素 ] が追加され、重なって表示されます。

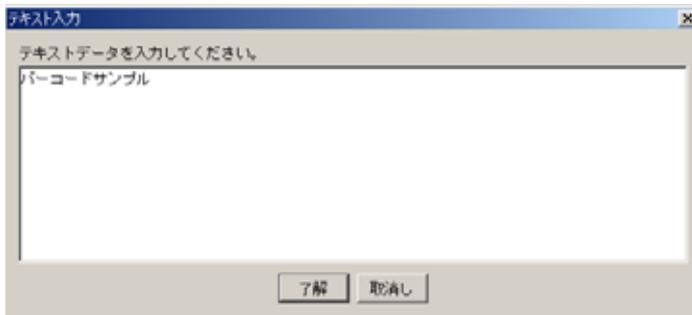
[ラベル要素]のデータを設定します。

左側の一覧の[ラベル要素]上で右クリック - [データ設定] - [テキスト入力...]とクリックします。



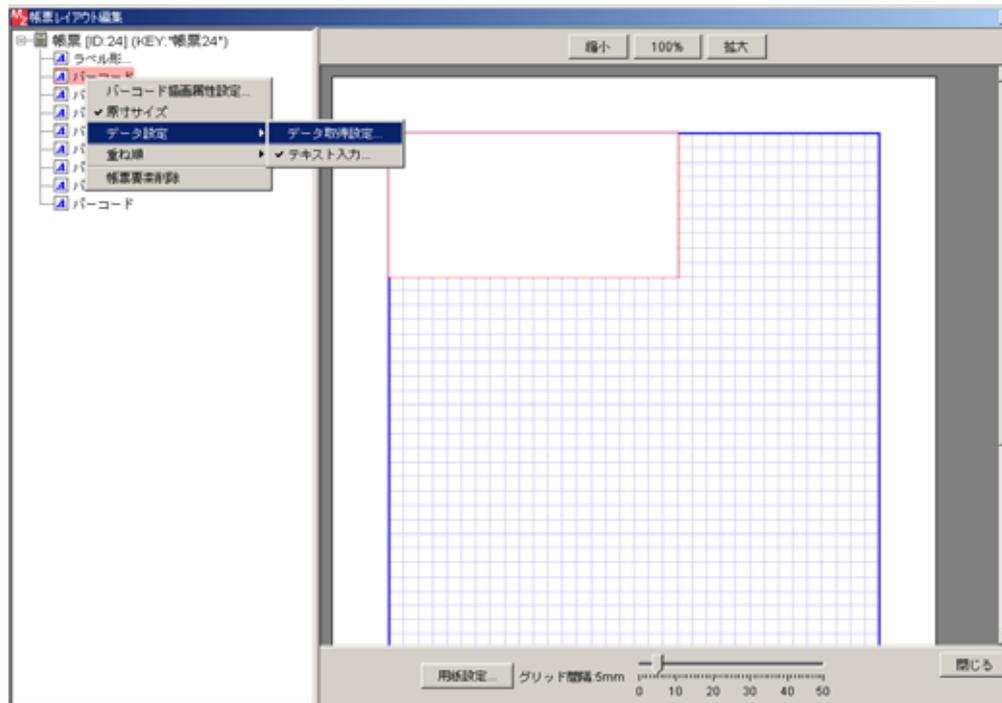
描画したい文字列を入力し、[了解]ボタンをクリックします。

ここでは、「バーコードサンプル」と入力します。

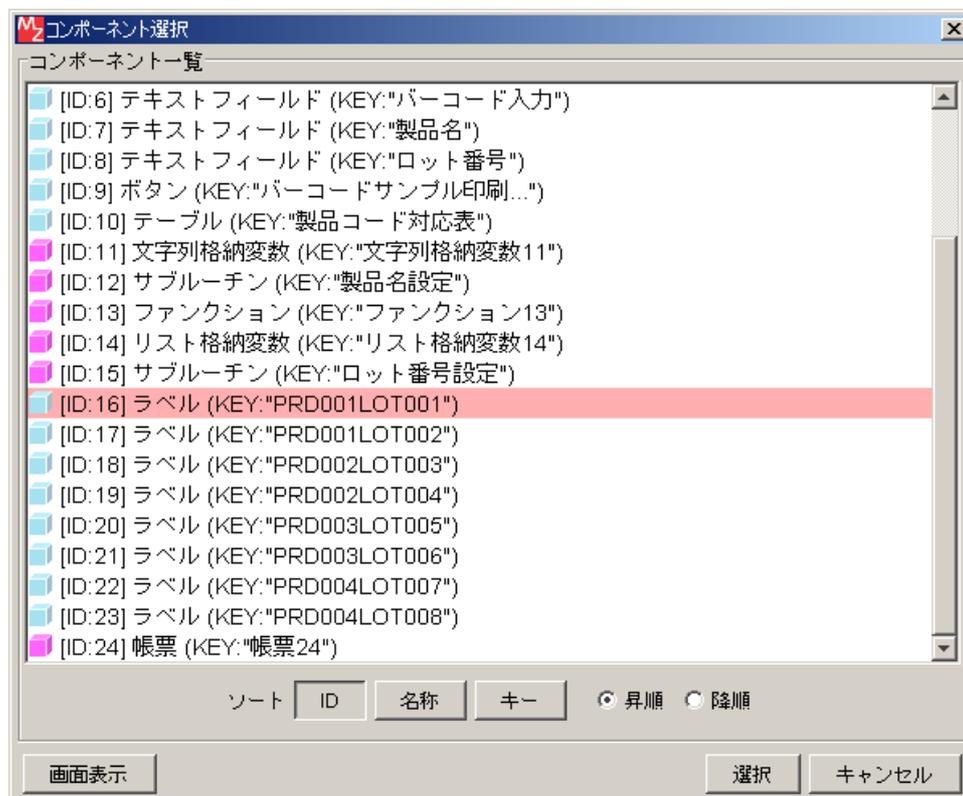


[バーコード要素]のデータを設定します。

左側の一覧の1番目の[バーコード要素]上で右クリック - [データ設定] - [データ取得設定...]とクリックします。



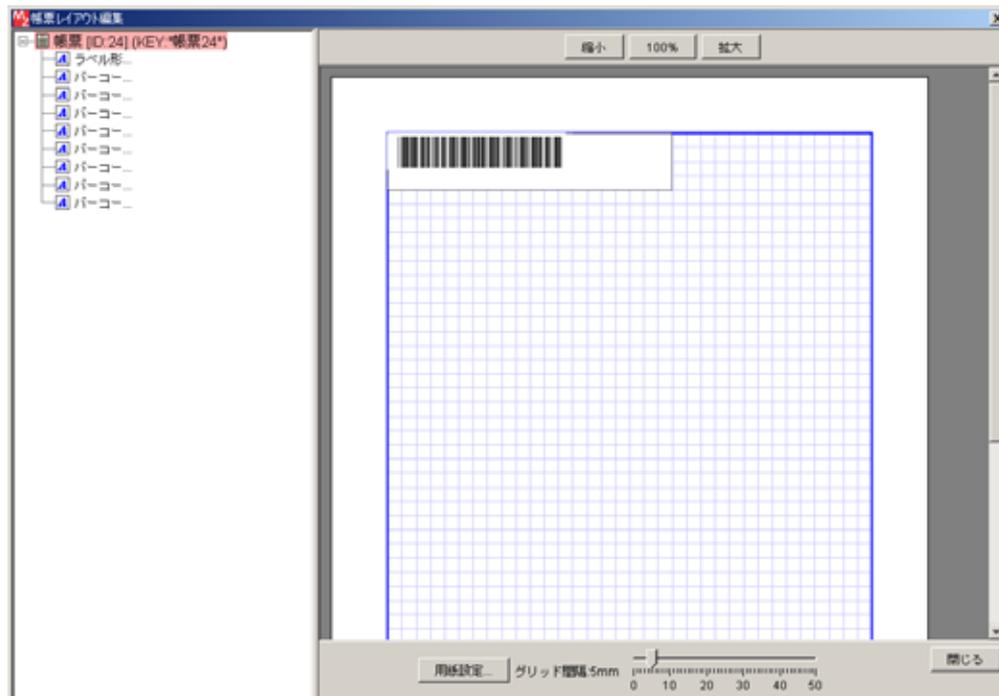
[取得元コンポーネント]を選択します。[変更...]ボタンをクリックし、[ラベル (ID:16)]コンポーネントを選択し[選択]ボタンをクリックします。



[ データ取得メソッド ] を選択します。[ ラベルのテキスト文字列を取得する( ) ] を選択し [ 設定 ] ボタンをクリックします。



～ を繰り返し、残りの [ バーコード要素 ] にそれぞれのデータを設定します。



バーコードのデータが表示されるように設定します。

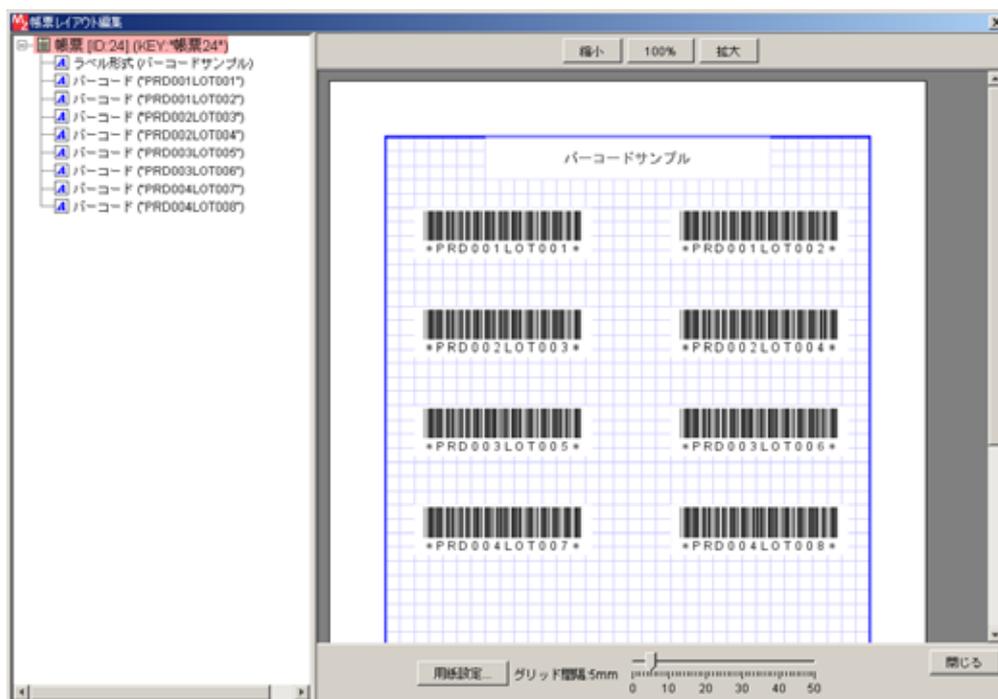
左側の一覧の1番目の [ バーコード要素 ] 上で右クリック - [ バーコード描画属性設定... ] とクリックします。

[ データ表示 ] の [ 表示する ] にチェックボックスにチェックをつけ [ 設定 ] ボタンをクリックします。



～ を繰り返し、残りの [ バーコード要素 ] にそれぞれのデータが表示されるように設定します。

[ ラベル要素 ] と [ バーコード要素 ] を見やすく配置し直します。

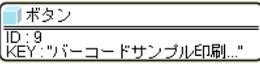
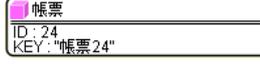


[ 閉じる ] ボタンをクリックし、ビルダー画面に戻ります。

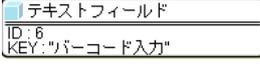
## 接続操作

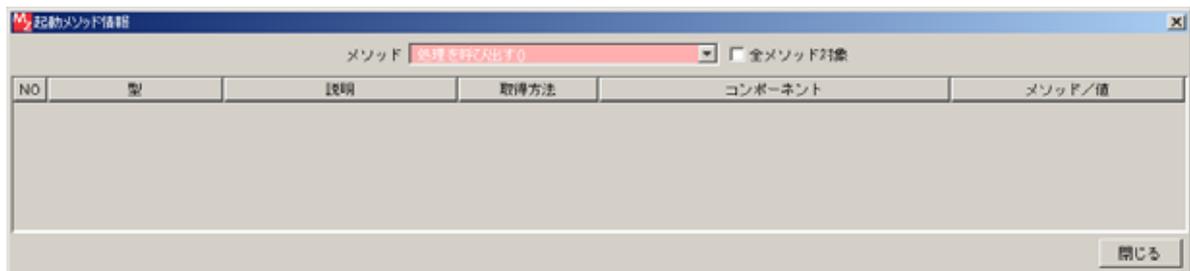
コンポーネント同士を接続しましょう。

ボタンをクリックしたら印刷プレビュー画面を表示する①

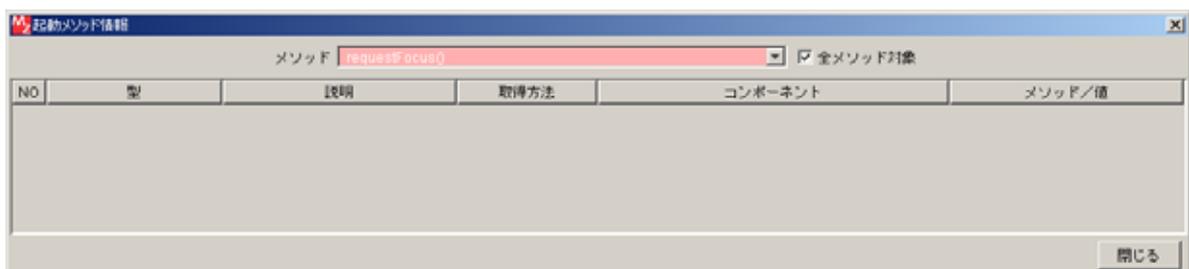
接続項目	接続関係
接続元コンポーネント (イベント発生コンポーネント)	
発生イベント	アクションイベント
接続先コンポーネント①	接続先 :   起動メソッド : 印刷プレビュー画面を表示する (Component) < 引数 > 説明 : 親コンポーネント 取得方法 : 固定値 メソッド / 値 : なし

[バーコード入力] テキストフィールドに入力フォーカスを設定する②

接続先コンポーネント②	接続先 :   起動メソッド : requestFocus()
-------------	--

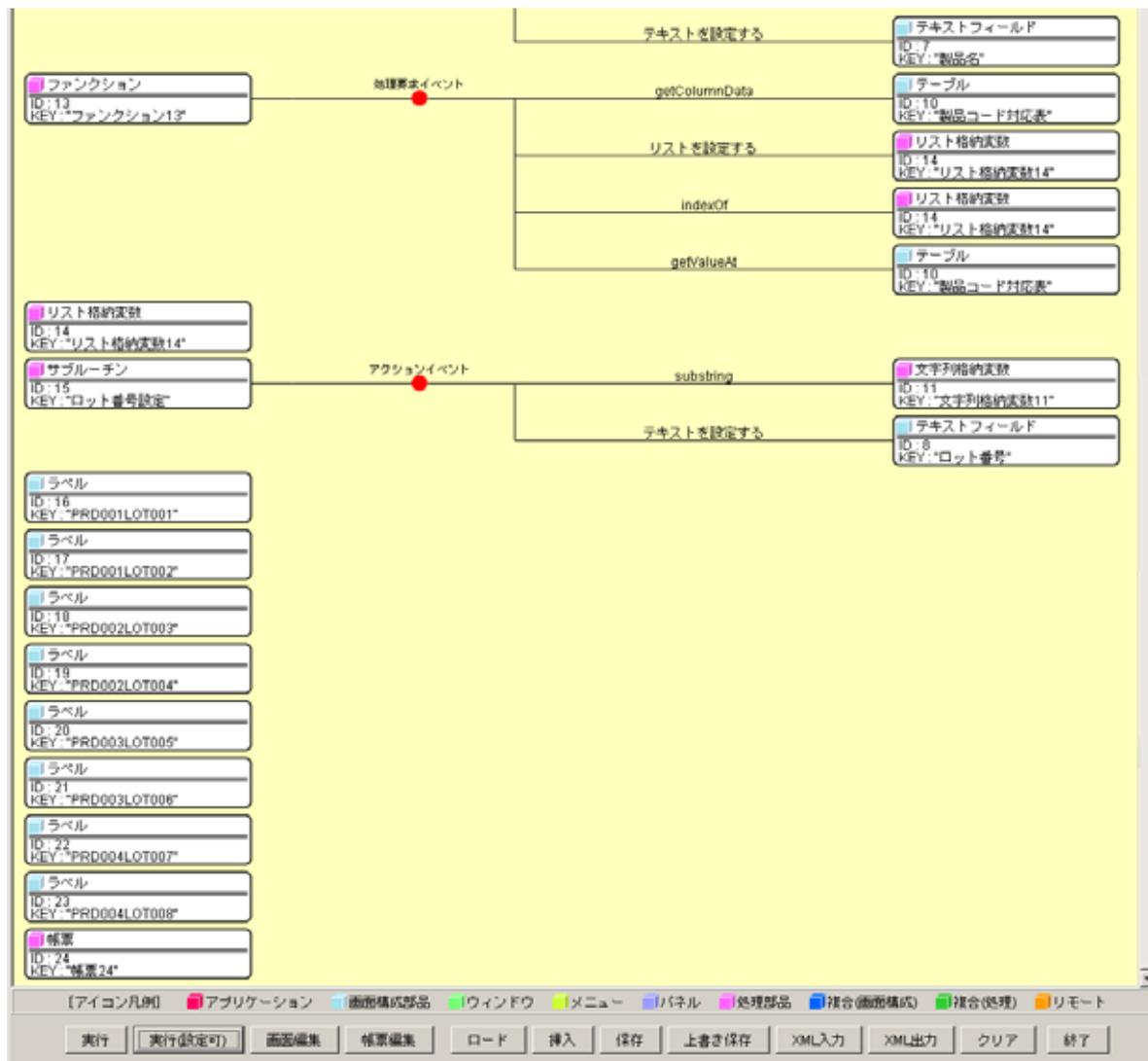


接続先コンポーネント ([ 帳票 ] コンポーネント①) の起動メソッドの設定画面



接続先コンポーネント ([ テキストフィールド ] コンポーネント②) の起動メソッドの設定画面





サンプルファイルとして収録されている「バーコード入力導入.apl」にはコメントが記述されています。