



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

Lesso	n.1 バーコード入力アプリケーションの作成	2
Stop 1	# 再	9
Step.1 Step.2	佩安	⊿ ع
Step.2 Step 3	2001-7次	
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ー覧の印刷	33

Lesson.1 バーコード入力アプリケーションの作成

<u>Step.1 概要</u>

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	PRDxxxL0Tyyy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) PRD003L0T005
	製品コード ロット番号

2) UBS 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。



<u>Step.2 画面作成</u>

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。



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]
■ パネル	1	[画面構成部品]-[パネル]-[パネル]
<mark>■</mark> ラベル	3	[画面構成部品]-[テキスト]-[ラベル]
■テキストフィールド	3	[画面構成部品]-[テキスト]-[テキストフィールド]
■ボタン	1	[画面構成部品]-[ボタン]-[ボタン]
<mark>■</mark> テーブル	1	[画面構成部品]-[テーブル]-[テーブル]

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	■アプリケーション
(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	■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)	
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()



必要なコンポーネントを追加し、画面を作成しましょう。

① 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ー[コンポーネントー括追加...]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]パネルー[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]パネルー[パネル]-[パネル]の追加数に「1」と入力します。 [画面構成部品]パネルー[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]パネルー[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]パネルー[ボタン]-[ボタン]の追加数に「1」と入力します。 [画面構成部品]パネルー[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。

MZ Platform アプリケーションビルダー	
ファイル アブリケーション オブション ヘルブ	
7プリケーション名称	🖹 🖹 🚔 🖄
D:1 KY:"20-41"	
10:2 KEY119年ル2*	
ID:3 KEY:"ラベル3"	
ID:4 KEY:"ラベル4"	
10:5 KEY: "ラベル5"	
テキストフィールド	
D:6 KEY:"テキストフィールド6"	
<u>ー テキストフィールド</u>	
ID:7 KEY: "テキストフィールド7"	
1 テキストフィールド	
D:8 KEY:"テキストフィールド8"	
ID:9 KEY:"ポタン9"	
- テーブル	
ID:10 KEY:"テーブル10"	
【アイコン凡例】 🛑 アブリケーション 👘 画面構成部品 📄 ウィンドウ 🤜 メニュー 🔤 パネル 🛑 処理部品 🗐 複合 価面構成) 🗐 複合	(処理) 🛑リモート
実行 実行(設定可) 画面編集 「「「栗編集」 ロード 「挿入」 保存 」上書法(保存 」 クリア 新	終了

② [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ー [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。

₩2コンポーネント属性情報		<u>×</u>
ComponentPublicName		🔽 NULL 🔺
AllowRemoteInvocation	C true ☉ false	
AllowPullTransfer	C true ☉ false	
AllowPushTransfer	C true ☉ false	
PropertyEditable	C true . ● false	
ComponentKeys	日本語: 英語: 英語:	T NULL
MultiLocaleToolTipText	日本語:	🗆 NULL
ContainerLayout	1	
AutoResize	🗨 true 🔿 false	
BorderTitle	製品コード対応表	
MultiLocaleBorderTitle	日本語: 英語:	
BorderTitleVisible	C true ⊙ false	
GridLayoutRows	0	
GridLayoutColumns	0	
ContainerOrderedFocusTraverse	C true .	
FocusTraverseByEnterEnabled	C true . ● false	
GridInterval	n	

③ [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ー [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。

₩コンポーネント属性情報		2
Opaque	C true	<u>_</u>
ToolTipText		NULL
Text	パーコード入力	🗖 NULL
ComponentKey	ラベル3	🗖 NULL
ComponentID	3	
ComponentPublicName		NULL

Opaque	C tru	ue 💿 false	<u>^</u>
ToolTipText			NULL NULL
Text	製品名		□ NULL
ComponentKey	ラベル4		I NULL
ComponentID	4		
ComponentPublicName			NULL
		-	

──□ンポーネント属性情報		×
Opaque	C true @ false	<u> </u>
ToolTipText		NULL .
Text	ロット番号	🗖 NULL
ComponentKey	ラベル5	🗖 NULL
ComponentID	5	
ComponentPublicName		NULL

④ [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	
ComponentID	6	
ComponentKey	製品名	T NULL
ComponentID	7	
	Component/D ComponentKey ComponentID	ComponentID 6 ComponentKey 製品名 ComponentID 7

⑤ [ボタン] コンポーネントのテキストを変更します。

ComponentID

[ボタン] コンポーネント上で右クリックー [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。

──□ンポーネント属性情報			×
 ToolTipText			🔽 NULL 📤
Text	バーコードサンプル印刷		NULL
 ComponentKey	ボタン9		
ComponentID	9		
ComponentPublicName			NULL

⑥ [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ー [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。

₩コンポーネント属性情報		
ToolTipText		V NULL -
ComponentKey	製品コード対応表	
ComponentID	10	
ComponentPublicName		VULL

⑦ [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。

₩_起					
		メソッド フレームな	を表示する()	💌 🗌 全メソッド対象	
NO	型	説明	取得方法	コンポーネント	メソッド/値
·					閉じる

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

⑧ [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

⑨ 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-[コンポーネントー括追加] とクリックします。

 ① [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。



⑪ 設定ボタンをクリックします。

② 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。



③ [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。

────────────────────────────		
□	·コード入力")	バネル [ID:2] (KEY:"製品コード対応表")
	コンポーネント追加 🕨 🕨	■ テーブル [ID:10] (KEY:"製品コード対応表")
	コンポーネントー括追加	
	新規コンポーネント追加 🕨 🕨	
	新規コンポーネント→括追加	
	コンポーネント挿入 🔹 🕨	
📕 🛛 🗐 ボタン [ID:9] (KEY:"バーコ	コンポーネント削除	(455,405)
	子コンポーネントー括削除	(155,135)
	属性情報設定	

⑭ [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。



- (1) 閉じるボタンをクリックし、ビルダー画面に戻ります。
- 16 実行(設定可)ボタンをクリックし実行します。

① パネルのタイトルを表示します。
 パネル上で右クリックー [枠] - [タイトル表示有無] - [表示する] とクリックします。

確認	▲ バーコード入力 製品名 ロット番号 バーコードサご	ンブル印刷		 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	ズ ・ ノト ・ Pウト ・ 回整列 ・ -カス移動 ・	タイトル タイトル表示 種類	1無 ♪ 表示する → ◆表示しな	
	パネルにタイト	ルが表示されま	す。					
		▲ バーコード入力		製品コード	対応表	<u>×</u>		
		製品名						

18 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ー [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。

テーブル作	成	×
?	空のテーブルを作ります。 行数	
	4	
	列数	
	2	
	デフォルトの列型	
	java.lang.String	-
	適用 取消	

19 次のように列名およびデータを入力します。

ロット番号

バーコードサンブル印刷...

<u>م</u>		
バーコード入力	「製品コード対	応表
	製品コード	製品名
製品名	PRD001	歯車
ロット番号	PRD002	ネジ
	PRD003	バネ
バーコードサンブル印刷	PRD004	軸受

2 確認後、ウィンドウを閉じます。

ここまで進めると次のようになります。

■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	
KEY:"			ID:1 КЕҮ:"フレーム1"
	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
ID : 1 KEY : "フレーム1"			KEY:"
「 パネル			
IU:2 KEY:"製品コード対応表"			
ラベル			
ID:13 KEY:"バーコード入力"			
KEY:"製品名"			
ラベル			
KEY:"ロット番号"			
<u> </u>			
KEY:"バーコード入力"			
<u> テキストフィールド</u> D:7			
KEY:"製品名"			
<u> テキストフィールド</u> D:8			
KEY:"ロット番号"			
ホタン ID:9 KEY:"バーコードサンブル印刷"			
□ テーブル			
ID:10 KEY:"製品コード対応表"			

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	■アプリケーション
(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	■テキストフィールド(ID:6, KEY:バーコード入力)
起動メソッド	入力フォーカスを設定する()

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。
- ② 接続したコンポーネントの処理を選びます。

[入力フォーカスを設定する()]をクリックします。

^	こ記動メソッド情報				×
		メソッド 入力:	フォーカスを設定する() 🔽 🗌 全メソッド対象	
1	10 型		取得方法	コンポーネント	メソッド/値
Г			·		
					閉じる

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●

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❶	■テキストフィールド(ID:6, KEY:バーコード入力)
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:固定值
	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❷	■テキストフィールド(ID:7, KEY:製品名)
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:固定值
	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

接続先コンポーネント❸	■テキストフィールド(ID:8, KEY:ロット番号)
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:固定值
	メソッド/値:なし

<mark>∼</mark> ₽起	動メソッド情報				×	
	メソッド テキストを設定する(String) 📃 🗖 全メソッド対象					
NO	型	[兑 ¹ 月	取得方法	コンポーネント	メソッド/値	
0	String	テキスト	固定値	-		
					閉じる	

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

ここまで進めると次のようになります。

■ 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	
KEY:"			ID:1 КЕҮ:"フレーム1"
		入力フォーカスを設定する	□ テキストフィールド
			ID:6 KEY:"バーコード入力"
	アプリケーション終了イベント	テキストを設定する	テキストフィールド
			ID:6 KEY:"バーコード入力"
		テキストを設定する	□ テキストフィールド □
	-		ID:7 KEY:"製品名"
		テキストを設定する	□ テキストフィールド
	L		ID:8 KEY:"ロット番号"
[]フレ−ム	アクションイベント	アプリケーションを終てする	🗐 アプリケーション
ID:1 KEY:"フレーム1"			KEY:"
ID:2 KEY:"製品コード対応表"			
1 ラベル			
ID:3 KEY:"バーコード入力"			
ID:4 KEY:"製品名"			
ID.s KEY:"ロット番号"			
□ テキストフィールド			
ID:10 KEY:"バーコード入力"			
<u> </u>			
ID:7 KEY:"製品名"			
□ テキストフィールド			
ID:8 KEY:"ロット番号"			
ボタン			
ID : 9 KEY : "バーコードサンブル印刷"			
<u>■</u> テーブル			
ID : 10 KEY : "製品コード対応表"			

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

≜ バーコード入力	<u>_ </u>
バーコード入力 PRD003L0T005	製品コード対応表
	製品コード製品名
製品名 パネ	PRD001 歯車
ロット番号	PRD002 ネジ
	PRD003 パネ
バーコードサンブル印刷	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。

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]
■サブルーチン	1	[処理部品]-[サブルーチン]-[サブルーチン]
■ファンクション	3	[処理部品]-[サブルーチン]-[ファンクション]
■リスト格納変数	3	[処理部品]-[変数]-[リスト格納変数]

操作

必要な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[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] 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。
- ② [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。

_	₩ コンポーネント属性情報			
	ComponentKey	製品名設定		L
	ComponentID	12		
	ComponentPublicName		VUL	L

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	■文字列格納変数(ID:11)
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
	コンポーネント : テキストフィールド(ID:6)
	メソッド/値:テキストを取得する

<mark>^∕</mark> 2₫	己動メソッド情報				×
		メソッド 文字列を設	定する(String)	🔽 🗌 全メソッド対象	
NO	型		取得方法	コンポーネント	メソッド/値
	String	文字列	メソッド戻り値	テキストフィールド [ID:6] (KEY:"バーコード入力")	テキストを取得する
					閉じる

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を呼び出す

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)	■文字列格納変数(ID: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	■サブルーチン(ID:12, KEY:製品名設定)
起動メソッド	処理を呼び出す()

₩2起動メソッド情報	

"と起	目動メソッド情報				×
		メソッド 処理を	呼び出す()	▼ □ 全メソッド対象	
NO	型	説明	取得方法	コンポーネント	メソッド/値
					閉じる

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

接続項目	接続関係
接続元コンポーネント	■ サブルーチン (ID:12 KEV:制品名設定)
(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❶	■文字列格納変数(ID:11)
起動メソッド	substring(int, int)
<引数0>	取得方法:固定值
	メソッド/値:0
<引数1>	取得方法:固定值
	メソッド/値:6

₩2起動メソッド情報

	メソッド substring(int,int) 🔽 🗖 全メソッド対象						
NO	型	記印	取得方法	コンポーネント	メソッド/値		
0	int		固定値	-	0		
1	int		固定値	-	6		
	B1*2						

lase 1

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

製品コード(前半6文字)を取得する●

ファンクションを呼び出す2

接続先コンポーネント❷	■ファンクション(ID:13)
起動メソッド	ファンクションの呼び出し(1 引数)(Object)
<引数>	説明:引数
	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)

製品名を設定する₿

接続先コンポーネント€	■テキストフィールド(ID:7, KEY:製品名)
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:ファンクションの呼び出し(1引数)
	(ファンクション)

<mark>™</mark> ∠∄						
		メソッド ファンクション	の呼び出し(1引数)	(Object) 🔽 🗌 全メソッド対象		
NO	型		取得方法	コンボーネント	メソッド/値	
0	Object	弓 数	メソッド処理結果	-	substring (文字列格納変数 [ID:11])	
					閉じる	

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

₩2起						
	メソッド 🔽 テキストを設定する(String) 🔽 🗖 全メソッド対象					
NO	型		取得方法	コンポーネント	メソッド/値	
0	String	テキスト	メソッド処理結果	-	ファンクションの呼び出し(1引…	
					閉じる	

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●

接続項目	接続関係
接続元コンポーネント	\neg \neg \neg γ
(イベント発生コンポーネント)	
発生イベント	処理要求イベント
接続先コンポーネント❶	■テーブル (ID:10, KEY:製品コード対応表)
起動メソッド	指定列の列データをリスト形式で取得する(int)
<引数>	取得方法:固定值
	メソッド/値:0

製品コードをリスト格納変数に格納する❷

接続先コンポーネント❷	■リスト格納変数(ID:14)
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:指定列の列データをリスト形式で取得する
	(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❸

接続先コンポーネント❸	■リスト格納変数(ID:14)
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❹

接続先コンポーネント❹	■テーブル (ID:10, KEY:製品コード対応表)
起動メソッド	指定セルの値を取得する(int, int)
<引数0>	取得方法:メソッド処理結果
	メソッド/値:index0f(リスト格納変数)
<引数1>	取得方法:固定值
	メソッド/値:1

		导する(int) 🔽 🗖 全メソッド対象	ータをリスト形式で取	メソッド 指定列の列デ		
	メソッド/値	コンポーネント	取得方法	記印	型	NO
	0	-	固定値	指定列	int	0
- z	BI					
			-			

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

<mark>~</mark> ∠₽						
	メソッド リストを設定する(PFObjectList) 🔽 🗖 全メソッド対象					
NO	型		取得方法	コンポーネント	メソッド/値	
0	PFObjectList	リスト	メソッド処理結果	-	指定列の列データをリスト形式で…	
					閉じる	

接続先コンポーネント([リスト格納変数] コンポーネント②)の起動メソッドの設定画面

19 記動メソッド情報					
	メワッド indexe	Df(Object)	💌 🗖 全メソッド対象		
NO 型		取得方法	コンポーネント	メワッド/値	
0 Object		メソッド戻り値	ファンクション [ID:13] (KEY:"ファンクション13")	第1引数の取得	
				閉じる	

<mark>™</mark> z趙						
メンッド 指定セルの値を取得する(int,int)						
NO	型		取得方法	コンボーネント	メソッド/値	
0	int	指定行	メソッド処理結果	-	indexOf (リスト格納変数 [ID:9])	
1	int	指定列	固定値	-	1	
,					閉じる	

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

確認

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。
- ② [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、「PRD003L0T005」と入力し【Enter】キーを押します。

≜ バーコード入力	
バーコード入力 PRD003L0T005	製品コード対応表
	製品コード製品名
製品名 バネ	PRD001 歯車
ロット番号	PRD002 ネジ
	PRD003 バネ
バーコードサンプル印刷	PRD004 軸受

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

≜ バーコード入力	
バーコード入力 PRD003L0T005	製品コード対応表
製品名 バネ	製品コード 製品名 PRD001 歯車
ロット番号 LOT005	PRD002 ネジ PRD003 バネ
バーコードサンブル印刷	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。

コンポーネント名	必要数	
■サブルーチン	1	[処理部品]-[サブルーチン]-[サブルーチン]

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ー[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。
- ② [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。

10ポーネント属性情報		x
ComponentKey	ロット番号設定	ULL
ComponentID	15	
ComponentPublicName		ULL

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を呼び出す

接続項目	接続関係
接続元コンポーネント	■ 立字 副 换 纳 亦 粉 (ID:11)
(イベント発生コンポーネント)	
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	■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()

<mark>^∕</mark> ∠₽	動メソッド情報				×
		メソッド 処理を	呼び出す()	▼ □ 全メソッド対象	
NO	型	記印	取得方法	コンボーネント	メソッド/値
1					
					閉じる

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●

接続項目	接続関係
接続元コンポーネント	■サブルーチン (ID:15 KEV:ロット来号設宁)
(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❶	■文字列格納変数(ID:11)
起動メソッド	substring(int, int)
<引数0>	取得方法:固定值
	メソッド/値:6
<引数1>	取得方法:固定值
	メソッド/値:12

ロット番号を設定する❷

接続先コンポーネント❷	■テキストフィールド(ID:8, KEY:ロット番号)	
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)	

<mark>^∕</mark> ₂起	動メソッド情報				X
		メソッド substring(i	nt,int)	▼ □ 全メソッド対象	
NO	型	説明	取得方法	コンポーネント	メソッド/値
0	int		固定値	-	6
1	int		固定値	-	12
					閉じる

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

№起	勧メソッド情報				×
		メソッド テキス	ストを設定する(String)	💌 🗆 全メソッド対象	
NO	型	記問	取得方法	コンポーネント	メソッド/値
0	String	テキスト	メソッド処理結果	-	substring (文字列格納変数 [ID:11])
					閉じる

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)❷の起動メソッドの設定画面



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。
- ② [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、「PRD003L0T005」と入力し【Enter】キーを押します。

🌲 バーコード入力	
バーコード入力 PRD003L0T005	製品コード対応表
	製品コード製品名
製品名 バネ	PRD001 歯車
ロット番号 1 07005	PRD002 ネジ
	PRD003 バネ
バーコードサンプル印刷	PRD004 軸受

まとめ:

ここまで進めると次のようになります。

アブリケーション	アプリケーション開始イベント	フレームを表示する	
KEY:"			
		入力フォーカスを設定する	■ テキストフィールド
			ID:6 KEY:"バーコード入力"
	アプリケーション終了イベント	テナフトを設定する	「 テキストフィールド
	• • •	J TAT CAREY J	ID:6 KEY:"パーコード入力"
		テキストを設定する	1D:7
		テキストを設定する	
			KEY:"ロット番号"
	アクションイベント	アプリケーションを終了する	アブリケーション
	. · · · · · · · · · · · · · · · · · · ·		KEY:"
「 パネル			
ID:2 KEY:"製品コード対応表"			
∫ ラベル			
TD:3 KEY:"バーコード入力"			
1D:4 KEY·"制星之"			
		文字列を設定する	1D:11
KEY:"バーコード入力"			KEY:"文字列格納変数11"
□ テキストフィールド □D: 7			
KEY:"製品名"			
<u>■テキストフィールド</u>			
IU.:8 KEY: "ロット番号"			
ボタン			
TD : 9 KEY : "バーコードサンプル印刷"			
(■) テーブル			
TD:10 KEY:"製品コード対応表"			
1 文字列格納変数	データ設定イベント	ᅇᅖᅕᄧᅸᅎᆙᄔᆣ	サブルーチン
ID:11 KEY:"文字列格纳变数11"	• •••••••••••••••••••••••••••••••••••	242070413	ID:12 KEY:"製品名號定"
()			
		処理を呼び出す	ID:15 ID:15
	マクションズベント		
ID:12		substring	
(KEY:"製品名設定"			KEY:"文字列格納変数11"
		ファンクションの呼び出し(1引数)	<u> ファンクション</u> D:13
			(REY:"ファンクション13"
		テキストを設定する	→ テキストフィールド → → → → → → → → → → → → → → → → → → →
			KEY:"製品名"
ファンクション	処理要求イベント	指定列の列データをリスト形式で取得する	<u> </u>
ID:13 KEY:"ファンクション13"			ID:10 KEY:"製品コード対応表"
		リストを設定する	リスト格納変数
			ID:14 KEY:"リスト格納変数14"
		indexOf	ULL 100000000000000000000000000000000000
		Indexor	ID:14 ID:14 /EX:"リフト救納恋謝14"
			(= =
		指定セルの値を取得する	
		指定セルの値を取得する	□ テーブル ID:10 KEY:"製品コード対応表"
 リスト格納変数 ID:14 		指定セルの値を取得する	■ テーブル ID:10 KEY:"製品コード対応表"
Uスト格納変数 ID:14 KEY:"リスト格納変数14"		指定セルの値を取得する	■ テーブル ID:10 KEY:"製品ユード対応表"
Uスト格納変数 ID:14 KEY:"リスト格納変数14" UD:15 ID:15 ID:	アクションイベント	指定セルの値を取得する substring	 テーブル ID:10 ID:10 KEY:"製品コード対応表" 文字列格納変数 ID:11 ID:11
<mark>リスト格納変数 ID:14 KEY:リスト格納変数14" リサブルーチン ID:15 ID:15 KEY:"ロット番号設定"</mark>	アクションイベント	指定セルの値を取得する substring	 テーブル ID:10
<mark>リスト格納変数</mark> ID:14 KEY:"リスト格納変数14" リサブルーチン ID:15 KEY:"ロット番号設定"	アダションイベント	指定セルの値を取得する substring テキストを設定する	 テーブル □ テーブル □D:10 KEY: "製品コード対応表" ↓文字列格納変数 □D:11 KEY: "文字列格納変数11" □ テキストフィールド

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

≜ バーコード入力	
バーコード入力	製品コード対応表
製品名 パネ	製品コード 製品名 PRD001 歯車
ロット番号 LOT005	PRD002 ネジ PRD003 バネ
バーコードサンブル印刷	PRD004 軸受

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●

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❶	■テキストフィールド(ID:6, KEY:バーコード入力)
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:固定值
	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

接続先コンポーネント❷	■テキストフィールド(ID:6, KEY:バーコード入力)
起動メソッド	入力フォーカスを設定する()

<mark>∾</mark> と起	動メソッド情報				×
		メソッド テキス	ストを設定する(String)	🖃 🗖 全メソッド対象	
NO	型	説明	取得方法	コンポーネント	メソッド/値
0	String	テキスト	固定値	-	
					閉じる

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

№起意	カメンッド情報				×
		メソッド 入力:	フォーカスを設定する(👤 🗌 全メソッド対象	
NO	型		取得方法	コンポーネント	メソッド/値
J					
					閉じる

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❷)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。
- ② [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、「PRD003L0T005」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。

≜ バーコード入力	
バーコード入力	製品コード対応表
	製品コード製品名
製品名 バネ	PRD001 歯車
ロット番号 0,07005	PRD002 ネジ
	PRD003 バネ
バーコードサンプル印刷	PRD004 軸受

ここまで進めると次のようになります。

アプリケーション	アプリケーション開始イベント	フレーノ ギキニオス	[]7V-4
	•		ID:1 KEY: "バーコード入力"
		パーコード入力用のテキストフィールドに入力フォーカスを	「設定します。」
		入力フォーカスを設定する	■テキストフィールド
			ID:6 KEY:"バーコード入力"
	アプリケーション終了イベント	- テキストを設定する	デキストフィールド
	t		ID:6 KEY:"バーコード入力"
		テキストを設定する	テキストフィールド
			TD:7 KEY:"製品名"
		テキストを設定する	
			ID:8 KEY:"ロット番号"
1 70-4	アクションイベント	アゴリケーション方級ですろ	(■アプリケーション
ID:1 KEY:"バーコード入力"	•	7777747CN(178	KEY: "バーコード入力"
	<u> </u>		
バーコードリーダからの入力文字列を バーコードの書式は「PRDXXXLOTY ここでは、前半の6文字で表現される また、後半の6文字で表現されるロッ	『もとに、製品名とロット番号を設定します。 YY」で、前半の6文字(PRD001など)が壊)製品コードを[製品コード対応表]テーブ (ト番号を、[ロット番号]テキストフィー	。 製品コード、後半の6文字(LOT001など)がロット番号を表 ルと照合し、対応する製品名を[製品名]テキストフィール! ルドに設定します。	すものとします。 ℃に設定します。
	2 		
KEY:"製品名"			
1D:5			
KEY:"ロット番号"		「パーコード入力文字列を設定」ます。	
			(一)文字列格納変数
	, E		ID:11 KEY:"バーコード入力文字列"
		次の入力用に、テキストフィールドをクリアしておきます。	
		テキストを設定する	<u> </u>
			ID1:0 KEY:"バーコード入力"
		バーコード入力用のテキストフィールドに入力フォーカスを	
		入力フォーカスを設定する	<u> テキストフィールド</u> D:6
C-171			(KEY:"バーコード入力"
<u> テキストフィールド</u> D:7			
KEY:"製品名"			
テキストフィールド D:8 /EV: ロト#号			
「ボタン			
ID:9 KEY:"バーコードサンブル印刷"			
<u>■ テーブル</u>			
ID:10 KEY:"製品コード対応表"			
文字列格納変数 ID:11	データ設定イベント	製品コードを「製品コード対応表」テーブルと照合し、製品コードを「製品コードを」	品名を [製品名] テキストフィールド
KEY:"バーコード入力文字列"		処理を呼び出す	■サブルーチン
	<u>-</u>		ID:12 KEY:"製品名設定"
		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	रव.
		処理を呼び出す	■サブルーチン
			ID.15 KEY:"ロット番号設定"

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]テー	キストフィールドに設定します。	
	製品コード(前半6文字)を取得します。	
ID:12 KEY:"製品名設定"	substring	● 文字列格納変数
		ID:11 KEY:"バーコード入力文字列"
	【製品コードを【製品コード対応表】テーブルと照合し、製品	名を取得します。
	ファンクションの呼び出し(1引数)	ファンクション
		ID:13 KEY:"コード照合&製品名取得"
	製品名を設定します。	
	テキストを設定する	<u> テキストフィールド</u> D:7
		KEY:"製品名"
	対応する製品名を取得します。	
「ファンクション」の理要求イベント	製品コードリスト(第0列)の取得	
		■ テーブル
		TD:10 KEY:"製品コード対応表"
	製品コードリストをリスト格納変数に設定	
	リストを設定する	リスト格納変数
		ID:14 KEY:"製品コードリスト"
	与えられた製品コードのインデックスを取得	
	indexOf	 リスト格納変数 リスト格納変数
		ID.14 KEY:"製品コードリスト"
	製品名の取得	
	指定セルの値を取得する	■ テーブル ID:10
		KEY:"製品コード対応表"
<mark>リスト格納変数</mark> ID:14 KEY:"製品コードリスト"		
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。		
	ロット番号(後半6文字)を取得します。	
 	substring	文字列格納変数
	The second secon	ID:11 KEY:"バーコード入力文字列"
	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。
	テキストを設定する	■ テキストフィールド
		ID。 KEY:"ロット番号"

コメントは適宜記入してください。

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。

PRD001L0T001、PRD001L0T002、PRD002L0T003、PRD002L0T004 PRD003L0T005、PRD003L0T006、PRD004L0T007、PRD004L0T008

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	
■ラベル	8	[画面構成部品]-[テキスト]-[ラベル]

操作

必要な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一括追加します。
 作業領域で右クリック- [コンポーネントー括追加...] 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] パネルー [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。



② [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ー [属性情報設定...] とクリックします。
 [Text] に「PRD001L0T001」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	PRD001L0T002
ID:18	PRD002L0T003
ID:19	PRD002L0T004
ID:20	PRD003L0T005
ID:21	PRD003L0T006
ID:22	PRD004L0T007
ID:23	PRD004L0T008

	2
C true © false	
PRD001LOT001	
	VULL
PRD001L0T001	
16	
	VULL
	PRD001L0T001 PRD001L0T001 16

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。



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	
■帳票	1	[入出力]-[帳票]-[帳票]

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。

② 画面を作成します。

帳票編集ボタンをクリックします。

MZ Platform アブリケーションビルダー	
ファイル アブリケーション オブション ヘルブ	
アブリケーション名称 8 11	📾 🖄
マクションイベント Substring 文字列格納実数11 10 175 アクションイベント Substring D:11 10:15 E F F アキストを設定する 17年ストフィールド D:0.10	
<u> ビビソ、ロット番号</u>	
「ラベル ラベル ラベル ラベル ラベル	
D:19 KEY:"PRD002L0T004" 「ラベル D:20 KEY:"PRD003L0T005"	
ID: 21 KEY: "PRD003L0T006" ID: 72 KEY: "PRD04L0T007"	
【アイコン凡例】 🛑 アブリケーション 🏾 (画面構成部品 😑 <u>ヴィンド</u> ウ 😑 メニュー 🔤 パネル 📄 処理部品 💼 (積合(画面構成)) 💼 (積合(処理)) 🧧	 」リモート
実行 実行(設定可) 画面編集 「「熊栗編集」」ロード 挿入 保存 上書き保存 クリア 終了	

③ 帳票要素をレイアウト画面に追加します。

確認

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。



④ [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。



⑤ 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。

テキストデータを入力してください。
1. 2 104445
ハーコートザンラル
了解 取消し

⑥ [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。



⑦ [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。



⑧ [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。

	×
取得元コンボーネント ラベル [ID:16] (KEY:"PRD001L0T001")	変更
- データ取得メソッド	□ 全メソッド
ラベルのテキスト文字列を取得する()	•
設定 キャンセル	

⑨ ⑥~⑧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。



① 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]上で右クリックー[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。



- 12 ⑩~⑪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。
- (1) [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。



(4) 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●

接続項目	接続関係
接続元コンポーネント	- ギタン (ID:0 KEV:バーコードサンプル印刷)
(イベント発生コンポーネント)	「「バダン(ID:9, KEI:ハーコードリンフル山洞)
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	■帳票(ID:24)
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:固定值
	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❷

接続先コンポーネント❷	■テキストフィールド(ID:6, KEY:バーコード入力)
起動メソッド	入力フォーカスを設定する()

×

接続先コンポーネント([帳票] コンポーネント●)の起動メソッドの設定画面

№起	動メソッド情報				×
		メソッド 入力:	フォーカスを設定する(👤 🗖 全メソッド対象	
NO	型		取得方法	コンポーネント	メソッド/値
1					881.11.7

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❷)の起動メソッドの設定画面

ここまで進めると次のようになります。

■ アプリケーション	アプリケーション開始イベント	フレールをまデオス	■ フレーム
			ID:1 KEY:"バーコード入力"
		パーコード入力用のテキストフィールドに入力フォーカスを	設定します。
		入力フォーカスを設定する	■ テキストフィールド
	アプリケーション終了イベント	テキストを設定する	<u>■ テキストフィールド</u>
		T	KEY:"バーコード入力"
		テキストを設定する	<u> テキストフィールド</u>
			ID:7 KEY:"製品名"
		テキストを設定する	<u> </u>
			ID:8 KEY:"ロット番号"
	アクションイベント	アブリケーションを終了する	アプリケーション
ID:1 KEY:"バーコード入力"			KEY : "バーコード入力"
バーコードリーダからの入力文字列格 バーコードの書式は「PRDXXXLOTY ここでは、前半の6文字で表現されるロッ 「バネル [D:2 KEY:"製品コード対応表" 「ラベル [D:3 「ラベル [D:4 KEY:"コット番号" 「ラベル [D:5 KEY:"ロット番号" 「テキストフィールド [D:6 [D:6 [D:6] [D:7]	(もとに、製品名とロット番号を設定します、 Y) で、前半の6文字(PRD001など) が 製品コードを「製品コード対応表) テーブ ト番号を、「ロット番号」テキストフィー、 下着号を、「ロット番号」テキストフィー、 アクションイベント E	。 &品コード、後半の6文字(LOT001など)がロット番号を表す 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 ルドに設定します。 バーコード入力文字列を設定します。 文字列を設定する	ものとします。 に設定します。
 □テキストフィールド □D:7 ■ テキストフィールド □ テキストフィールド □D:8 		次の入力用に、テキストフィールドをクリアしておきます。 テキストを設定する パーコード入力用のテキストフィールドに入力フォーカスを 入力フォーカスを設定する	<pre>(KEY:"パーコード入力文字列") (テキストフィールド) :6 (テキストフィールド) () () () () () () () () () () () () (</pre>
KEY:"ロット番号"			
		日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日	U:24
【KEY:"バーコードサンブル印刷"			【KEY:"バーコードサンブル印刷" 】
			設定します。 「テキストフィールド
		ハバノオーガスで設定する	ID:6 KFY:"バーコード入力"
 ラーブル 10:10 KEY:*製品コード対応表" 文字列烙納支数 10:11 10:11 KEY:*りヾーコード入力文字列* 	データ設定イベント	「製品コードを [製品コード対応表] テーブルと照合し、製品 しに設定します。 」 04環を呼び2011す	名を [製品名] テキストフィールド
	E		ID:12 KEY:"製品名設定"
		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	Ţ.
		処理を呼び出す	サ ブルーチン
			ID:15 KEY:"ロット番号語定"

	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。	
サブルーチン	アクションイベント	製品コード(前半6文字)を取得します。	
	•		■ 文字列格納変数 10:44
			KEY:"バーコード入力文字列"
		製品コードを「製品コード対応表」テーブルと照合し、	製品名を取得します。
		ファンクションの呼び出し(15)数)	ID:13 ID:13 ID:13 ID:14
		製品名を設定します。	
			■ テキストフィールド
			LU:7 KEY:"製品名"
	「製品コード対応表」テーブルと昭合し。	対応する製品名を取得します。	
	処理要求イベント		
【KEY:"コード照合&製品名取得"】		自 指定列の列ナーダをリスト形式で収得する	
		製品コードリストをリスト格納変数に設定	
		リストを設定する	リスト格納変数
			ND-14 KEY:"製品コードリスト"
		[与えられた製品コードのインデックスを取得]	
		indexOf	
		製品名の取得	
		指定セルの値を取得する	■ テーブル
			ID ::10 KEY : "製品コード"対応表"
リスト格納変数			
(ID:14 (KEY:"製品コードリスト"			
 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。		
	アクションイベント	ロット番号(後半6文字)を取得します。	
D:15 /FV:"ロット#是語字"		L substring	■ 文字列格納変数
(KEIL HYPESER			ID:11 KEY:"バーコード入力文字列"
		ロット番号を [ロット番号] テキストフィールドに設え	
		テキストを設定する	
			(KEY:"山ット番号"
バーコードのサンブル一覧を印刷しま	す。		
• PRD001LOT001	30		
PRD001L0T002 PRD002L0T003			
• PRD002L0T004			
PRD003L0T005 PRD003L0T006			
• PRD004LOT007			
PRD004L0T008			
● ラベル			
ID:16 KEY:"PRD001L0T001"			
1 ラベル			
KEY: "PRD001LOT002"			
□ ラベル □) 18 □			
KEY:"PRD002LOT003"			
<u>ラベル</u> ID:19			
KEY: "PRD002LOT004"			

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.mzax」にはコメントが記述されています。 コメントは適宜追加してください。