MZ Platform

<u>アプリケーションビルダー操作説明書</u>

= Application Builder Operating Manual =

Revision 3.5 [MZ Platform.3.5]

=目次=

1. MZ PLATFORM について	4
1.1. 概要	
1.2. コンポーネント	
1.3. コンポーネント間の接続	5
1.4. 提供範囲	6
1.5. インストール方法と動作環境	7
1.6. 動作環境の設定	7
2. 用語説明	8
3. アプリケーションの構築	12
3.1. アプリケーションビルダーの起動	
3.2. 構築作業の開始	
3.3. コンポーネントの追加/削除	
3.4. 複合コンポーネントの利用	
3.5. コンポーネント間の接続設定	
3.6. コンポーネントのコピー/切り取り/貼り付け	
3.7. アプリケーション開始処理/終了処理の設定	
3.8. 画面配置の設定	
3.9. コンポーネント属性の変更	
3.10. 実行	
3.11. デバッグ機能	
3.12. アプリケーションの保存/ロード	
3.13. アプリケーションのパスワードロック機能	
3.14. アプリケーション構築時のユーティリティ機能	
3.15. コメント機能	
3.16. その他機能	
4. 帳票の作成	84
4.1. 帳票のデータ構造	
4.1.1. 帳票コンポーネント	
4.1.2. 帳票構成要素 : ラベル要素	
4.1.3. 帳票構成要素 : テーブル要素	
4.1.4. 帳票構成要素 : バーコード要素	
4.1.5. 帳票構成要素 : QR コード要素	
4.1.6. 帳票構成要素:イメージ要素	
4.1.7. 帳票構成要素:画面イメージ要素	
4.2. 帳票作成/印刷の流れ	
4.3. 帳票作成の操作手順	
4.4. 帳票印刷手順	
4.4.1. 帳票印刷プレビュー	
4.4.2. 帳票印刷	

5. 複合コンポーネントの構築	
5.1. 複合コンポーネント	
5.2. GUI 複合コンポーネントの構築	
5.2.1. 構築作業の開始	
5.2.2. 画面表示の動作確認	
5.2.3. 外部公開メソッドの設定	110
5.2.4. 外部公開イベントの設定	115
5.3. 非 GUI 複合コンポーネントの構築	116
5.4. 複合コンポーネントの利用	117
5.5. 複合コンポーネントの外部参照化	118
5.5.1. 複合コンポーネント外部参照の考え方	118
5.5.2. 複合コンポーネントの外部参照ファイル	119
5.5.3. 外部参照設定方法	119
5.5.4. XML 出力機能におけるのパスワードロックと外部参照設定	119
5.5.5. 外部参照化されたアータファイル名	120
6. WEB アプリケーションの構築	121
6.1. WEB アプリケーション	
6.2. WEB アプリケーションの構築	
6.2.1. 構築作業の開始	
6.2.2. コンポーネントの追加	
6.2.3. コンポーネント間の接続	
6.2.4. 画面レイアウト設定	
6.2.5. 画面表示の動作確認	
6.2.6. Web アプリケーション動作確認	
7. リモートアプリケーションとの連携	126
7.1. データ連携機能	
8. アプリケーションの実行(アプリケーションローダー)	127
9. コンポーネント情報の編集	
9.1. メソッド情報の設定	
9.1.1. メソッドの公開設定	
9.1.2. メソッド引数の設定	
9.2. イベント情報の設定	
9.2.1. イベント番号の設定	
9.2.2. イベント番号の追加	
9.2.3. イベント番号の削除	133
9.2.4. イベント内包データの設定	
10. アプリケーションのライセンス管理	135
11. XML によるアプリケーション表現	136
11.1. XML 形式ドキュメント構造	
11.2. XML タグ	
11.3. データ表現形式	

11.4. XML 形式表現サンプル	(参考)	
--------------------	------	--

1. MZ Platform について

1.1. 概要

MZ Platform は、設計・製造支援アプリケーション用共通プラットフォームの研究開発プロジェクト の成果物です。MZ Platform は、ソフトウェアのコンポーネント化(部品化)によって、システム構築 や変更を利用者自身による組み立て作業によって実現することを目的としています。そのため、様々な システムで共通に使用される標準コンポーネントと、それらの部品を使用してシステムを構築/利用す るための環境を提供します。

MZ Platform ではソフトウェアをコンポーネント化することで、ソフトウェアの保守性を高めるだけ でなく、コンポーネントの接続/構成を容易に、かつ動的に行うことによって、アプリケーションシス テム全体をより拡張性のあるものにします。

MZ Platform 上のアプリケーションは機能単位に分割されたコンポーネントによって構成され、コン ポーネント間は互いに依存性の無い形で関係付けを行います。MZ Platform の基本アーキテクチャを下 図に示します。



図1MZ Platform 基本構造

1.2. コンポーネント

MZ Platform 上のアプリケーションは機能単位に分割されたコンポーネントの集合として表現され ます。これによって各コンポーネント単位での保守性/再利用性を確保します。

MZ Platform のコンポーネントはオブジェクト指向開発におけるオブジェクトの一種で、データと機能を一体として持ち、共通のインターフェイスを有しています。MZ Platform は Java で開発されていて、コンポーネントは JavaBeans の派生クラスとして実装されています。

1.3. コンポーネント間の接続

コンポーネント間の接続は転送イベントモデルを使用し、すべての連携はイベント発生をトリガーに した処理起動によって行われます。MZ Platform はコンポーネントの管理とコンポーネント間の接続を 実現する基幹機能(コンポーネントバス)を提供します。コンポーネントバスはあるコンポーネントから イベントを受け、他のコンポーネントの処理を起動します。コンポーネント接続を実現するための機能 として、プラットフォームは以下の機能を提供します。

1)接続関係の定義

コンポーネント間の接続関係を定義するためのツールとして、アプリケーションビルダーを提供しま す。このツールは画面操作や少数のキーボード入力によって、ソースコードを書かずにアプリケーシ ョンを構築することができる開発支援ツールです。

2)動的な処理起動

コンポーネント間の接続関係はプログラムソース内に埋め込まず、実行時のデータとして管理するこ とによって、動的に処理を変更することができるようにしています。これによって、コンポーネント の接続はアプリケーション実行中でも変更可能とし、アプリケーションの実行を止めることなく仕様 変更/動作確認が可能です。

1.4. 提供範囲

MZ Platform が提供する範囲は以下の通りです。(下図太枠が提供範囲)



図 2 MZ Platform 提供範囲

- 1)アプリケーション実行環境
 - ①コンポーネントバス

アプリケーションでのコンポーネント管理/コンポーネント接続を行うための、プラットフォー ム基幹機能です。

②アプリケーションローダー

アプリケーションビルダーによって構築/保存されたアプリケーションデータを、ファイルから ロードし、実行します。

- 2)アプリケーション構築支援ツール
 - ①アプリケーションビルダー

コンポーネントをアプリケーションとして組み立てる機能をもつユーティリティツールを提供し ます。このツール上ではコンポーネントの貼り付け/属性変更、画面レイアウト設定、コンポー ネント間の接続、印刷設定(帳票)が可能です。また、ここで作成したアプリケーションはロー カルファイルに保存し、再利用が可能です。

②標準コンポーネント

アプリケーションを構築する際に使用される頻度が高いコンポーネント群を標準ライブラリとし て含めています。基本的には標準コンポーネントの組み合わせで広い範囲のアプリケーションが 作成可能です。

- 3)コンポーネント開発支援ツール(詳細は「コンポーネント開発ガイド」参照)
 - ①コンポーネント開発フレームワーク

アプリケーション構築の際に標準コンポーネントでは機能が不足する場合に、独自のコンポーネ ントを開発することが可能です。そのような場合にコンポーネントを開発するための共通クラス /インターフェイス群を提供します。また、コンポーネント開発用に必須メソッドなどを記述し た、テンプレートソースを提供します。

②コンポーネントソースのサンプル

コンポーネント開発のサンプルとして、サンプルコンポーネントソースを提供します。

1.5. インストール方法と動作環境

インストーラを実行し、所定の設定操作を実行すると MZ Platform が利用可能になります。詳細は「インストールガイド」をご覧ください。

1.6. 動作環境の設定

1)クラスパスの設定

MZ Platform の実行時に使用するクラスパスを設定することが可能です。MZ Platform に対するクラ スパスの設定は、クラスパス設定ファイル(導入フォルダ¥etc¥PlatformClassPath.ini)を編集しま す。設定方法については、「詳細設定説明書」(導入フォルダ¥docs¥manual内)を参照してください。

2)実行パラメータの設定

プラットフォームの動作設定は、初期設定ファイル(導入フォルダ¥etc¥Platform.ini)にて行います。 初期設定ファイルのパラメータの内容、設定方法については、「詳細設定説明書」(導入フォルダ ¥docs¥manual 内)を参照してください。

2. 用語説明

1)アプリケーション

ある特定の仕事を行うためのソフトウェア。それ自身がある業務の機能をはたす。

■関連用語

<u>プログラミング言語</u>

アプリケーションを作成するために使用する言語。通常のアプリケーションはプログラミング言語に よって"プログラム"という単位のソフトウェアを作成し、小さなプログラムの集まりでアプリケー ションという1つの機能を果たすまとまりとなる。

2)コンポーネント

アプリケーションを構成するソフトウェア部品。コンポーネントはそれだけである機能を提供しており、それぞれの間に依存関係はない。MZ Platform では、アプリケーションを構築するための最小単位の部品をコンポーネントと呼び、すべてのアプリケーションはコンポーネントの組み合わせによって構成されている。なお、コンポーネント自体はプログラムによって作られている。

■関連用語

<u>GUI コンポーネント</u>

GUI はグラフィカル・ユーザ・インターフェイスの略。文字での表示/入力だけでなく、表やグラ フのようにより見やすく表示し、マウス操作などでの簡単な操作を提供するもの。

<u>コンテナコンポーネント</u>

GUI コンポーネントをまとめてグループとして扱うための部品。例えばウィンドウ(フレーム、ダ イアログ)も複数の GUI コンポーネントをまとめて一つの枠内で表示するためのコンテナコンポー ネントである。主なコンテナコンポーネントはウィンドウとパネル。

<u>ユーティリティコンポーネント</u>

数値演算、集計など、それ自身は画面に表示されずに裏で処理をおこなう部品。

<u>コンポーネント属性</u>

コンポーネントがもっている性質を定義する情報。属性はコンポーネント毎にそれぞれ提供されており、例えば GUI コンポーネントであれば、表示する色や大きさなどが属性となる。

3)イベント

コンポーネントの状態変更を外部に伝える機能。イベントには多くの種類があり、コンポーネントに よって発生するイベントが異なる。MZ Platform では、アプリケーションの動作はすべてこの"イベ ント"の発生をきっかけに行われており、アプリケーションの構築作業は、イベントが発生したとき の振る舞いを指定することで行う。イベントの種類については「コンポーネント開発ガイド」を、各 コンポーネントから発生するイベントについては「コンポーネントリファレンス」または Javadoc を 参照。

■関連用語

イベント発生元コンポーネント

イベントを発生させるコンポーネント。アプリケーション構築画面上のコンポーネント接続関係にお いて、左側に位置するコンポーネントであり、イベント処理追加の操作によってイベントに対する処 理を追加する。

接続先コンポーネント

イベント発生元コンポーネントから発生したイベントによって起動される処理をおこなうコンポー ネント。アプリケーション構築画面上のコンポーネント接続関係において、右側に位置するコンポー ネントである。 4)メソッド

コンポーネントで実行する処理を指示する方法であり、コンポーネントの機能によってさまざまなメ ソッドが提供されている。各コンポーネントが提供しているメソッドについては、「コンポーネント リファレンス」を参照。

■関連用語

<u>メソッド引数</u>

メソッドを起動する際に、外部から与える情報。例えば足し算を行うメソッドに対して、計算対象と して与えられる2つの数字が"引数"である。

<u>メソッド引数取得方法</u>

メソッド引数の値を指定する形式。引数取得方法には以下の6つがあり、引数となる値を持っている 対象によってどれかを選択する。

①固定値

メソッド引数に固定の数値や文字列を渡す形式。実行するときの状態に関係なく、常に同じ値が 渡される。

②メソッド戻り値

メソッド引数に他のコンポーネントのメソッド戻り値を指定する形式。引数として渡したい情報 を他のコンポーネントがもっている場合には、この形式によってコンポーネント内のデータを取 得して引数として渡す。

③コンポーネント

メソッド引数にコンポーネント自体を指定する形式。引数として渡したい情報がコンポーネント 自身の場合には、この形式によって任意のコンポーネントを引数として渡す。

④イベント内包

メソッド引数にイベントに含まれているデータを指定する形式。例えば"データ変更イベント" には、変更情報がイベントに含まれており、その変更情報をメソッドに渡す必要がある場合には、 この形式によってイベントに含まれている情報を引数として渡す。

⑤イベント

メソッド引数にイベントそのものを指定する形式。イベントそのものを引数として受け取ること ができるメソッドに対しては、発生したイベントを引数として渡す。

⑥メソッド処理結果

メソッド引数にすでに処理の終わっているメソッドの処理結果(戻り値)を指定する形式。処理 結果を次々に引き渡すような場合には、この形式によって処理結果データを引数として渡す。

5)アプリケーションローダー

MZ Platform が提供するアプリケーションを実行するためのツール。アプリケーションビルダーで構築されたアプリケーションを実行するためのもので、組み立てられたアプリケーションを変更することはできない。

6)アプリケーションビルダー

MZ Platform が提供するアプリケーション構築を行うためのツール。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する作業を、マウス操作などの画面操作によって行う。

■関連用語

<u>実行</u>

構築したアプリケーションを実際に動かす操作。アプリケーションビルダーから [実行] ボタンを押 下して起動。なお、アプリケーションの実行は、アプリケーションコンポーネントの『アプリケーシ ョン開始イベント』に接続されている処理から実施される。

<u>実行(設定可)</u>

実行しながら GUI コンポーネントの属性を編集できる実行形態。アプリケーションビルダーから[実行(設定可)] ボタンを押下して起動。属性設定方法は GUI 部品で準備されており、多くはポップア ップメニューによる属性設定を提供している。

<u>画面編集</u>

GUI コンポーネントの配置を行い、画面のレイアウトを設定する機能。アプリケーションビルダー から [画面編集] ボタンを押下して起動。画面配置の方法には以下の 5 つがあり、表示したいレイア ウトにあわせて好きな形式を選択する。

①手動配置

GUI 部品を自由な位置に配置できる形式。位置はマウスによって自由に移動することが可能。 ②横方向整列

GUI 部品を横方向に一列に並べる形式。表示範囲の横幅が決まっている場合、横いっぱいになったところで折り返す。

③縦方向整列

GUI 部品を縦方向に一列に並べる形式。表示範囲の縦幅が決まっている場合、縦いっぱいになったところで折り返す。

④領域配置

GUI 部品を四方(東/西/南/北)、中央の 5 方向に配置する形式。配置される GUI 部品は表示範囲全体の大きさにあわせて拡大/縮小される。

⑤矩形分割配置

表示範囲全体を N×M の矩形に分割し、その左上から順に配置する形式。配置される GUI 部品の大きさによって、分割される領域は調整される。

帳票編集

アプリケーションから出力する帳票のレイアウトを設定するための機能。アプリケーションビルダー から[帳票編集]ボタンを押下して起動。帳票の要素として指定できるものには以下の6つがある。 ①ラベル

文字列1つについて表示するための帳票要素。文字列長に制限はなく、描画領域にあわせて折り 返して描画される。また、文字列中に改行コードがある場合、その位置にて改行される。 ②テーブル

表形式で描画するための帳票要素。テーブルのセル内の表示要素は文字列、バーコード、QRコ ード¹、イメージのいずれかをカラム単位で選択する。文字列の場合には、すべて横幅にあわせ て折り返して描画される。また、文字列中に改行コードがある場合、その位置にて改行される。 バーコード、QRコード、イメージは原寸表示/枠幅に合わせた縮小表示の何れかで描画される。 ③バーコード

バーコードを表示するための帳票要素。文字列情報を入力とし、指定されたコード体系に変換し

¹ QR コードは(株)デンソーウェーブの登録商標です。

て出力する。バーコードイメージの描画は横幅にあわせて調整され、縦方向のサイズは縦横比率 を維持した状態で自動調整される。

④QR コード

QR コードを表示するための帳票要素。文字列情報を入力とし、指定されたバージョン、エラー 訂正レベル等に従って描画される。

⑤イメージ要素

イメージを表示するための帳票要素。Java の Image データを入力とし、出力する。イメージの 描画は横幅にあわせて調整され、縦方向のサイズは縦横比率を維持した状態で自動調整される。 ⑥画面イメージ要素

GUI 部品の画面のイメージをそのまま描画するための帳票要素。イメージの描画は横幅にあわせて調整され、縦方向のサイズは縦横比率を維持した状態で自動調整される。

<u>ロード</u>

保存されたアプリケーションまたは複合コンポーネントをアプリケーションビルダー上に読み込む 機能。アプリケーションビルダーから [ロード] ボタンを押下して起動。

<u>挿入</u>

保存されたアプリケーションまたは複合コンポーネントを現在編集中のアプリケーション階層に追 加する機能。アプリケーションビルダーから [挿入] ボタンを押下して起動。

<u>保存</u>

構築したアプリケーションまたは複合コンポーネントを外部ファイルに保存する機能。アプリケーションビルダーから [保存] ボタンを押下して起動。保存したデータはロード機能により再度アプリケーションビルダー上に読み込むことができる。

<u>上書き保存</u>

保存データをロードした状態、または一度保存した状態で、再度同じファイルにアプリケーションま たは複合コンポーネントを保存する機能。アプリケーションビルダーから [上書き保存] ボタンを押 下して起動。

<u>クリア</u>

構築したアプリケーションまたは複合コンポーネントをすべてクリアし、初期状態に戻す機能。アプ リケーションビルダーから [クリア] ボタンを押下して起動。

7)データ構造

アプリケーション上で取り扱うデータには、数値や文字列のように単純に1つの値で表現されるもの もあれば、複数の値を合わせて構造をもたせた形で表現されるものもある。MZ Platform が提供する 最も基本的なデータ構造は、リスト構造、テーブル構造、ツリー構造の3つである。

■関連用語

<u>リスト構造</u>

データが一列に並んだ構造。一次元配列。N 個のデータを扱う場合、このリスト構造の長さは N となる。

<u>テーブル構造</u>

データが N×M に並んだ構造。2 次元配列。

列: 表データ構造で項目を示すもの。通常の表記では横方向にならぶ単位が列となる。

行 : 表データ構造でデータ1件を示すもの。通常の表記では縦方向にならぶ単位が行となる

セル:表データ構造で1つの最小単位のデータ枠を示すもの。セルは〇行〇列と表現される

<u>ツリー構造</u>

親子関係(階層)をもったデータ構造。1つの親に N 個の子が関係付けられ、最上位階層は1つ。

3. アプリケーションの構築

3.1. アプリケーションビルダーの起動

スタートメニューからアプリケーションビルダーを起動すると、下のような画面が表示されます。も し、実行中にコンソールを表示させたい場合は、"アプリケーションビルダー (コンソール)"を実行し ます。

[スタート] -[(すべての) プログラム]-[MZ Platform 3.5]-[アプリケーションビルダー]



記号	名称	役割		
а	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。		
с	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。		
е	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。		

アプリケーションビルダー画面の中央にアプリケーションの情報を表示するための広い領域があり ます。この表示領域のなかでアプリケーションの組み立てを行います。また、アプリケーション構築作 業のための様々な機能は、上部にあるメニューや下部にあるボタンで操作していきます。通常の作業の ほとんどは、中央の表示領域と下部のボタンを使用して行われます。

画面のサイズは変更可能ですが、小さくすると画面内に表示がおさまらなくなり、ボタンが表示され なくなってしまいます。しかし、すべてのボタンの機能は画面上部にあるメニューバーからも起動でき ますので、もし画面上にボタンが表示されない場合には、メニューバーから操作してください。

3.2. 構築作業の開始

アプリケーションの構築は、以下の流れで行います。



任意のタイミングで作業状況を保存/再生





3.3. コンポーネントの追加/削除

1)コンポーネントの追加

画面	アプリケーションビルダー メイン画面						
手順	①背景にてマウスを右クリックし、コンポーネント追加メニューを表示 ②追加対象のコンポーネントを指定						
	■ アプリケーション KEY:***						
	コンポーネント追加 画面構成部品 ウィンドウ 🗐 フレーム						
	コンボーネントー括道加 処理部品 メニュー ■ ダイアログ 複合コンボーネント作成 入出力 パネル ・ 複合コンボーネント追加 クラス指定 テキスト ・						
	貼り付け ポタン・ テーブル・						
	□ コンホーネンド検索 GUIコンボーネント検索 コンソント行注加 コンソント行注加 ローコンソント行注加						
	コメント行検索 クラフ ロメント行検索 ユーティリティ 検索 グラフィックス						
	選択したコンポーネントが表示される						
	■ アブリケーション KEY:"" ■ フレーム						

2)クラス指定によるコンポーネント追加

画面	アプリケーションビルダー メイン画面
手順	①背景にてマウス右クリックしコンポーネント追加メニューの [クラス指定]を選択
	■ アブリケーション
	KEY: ""
	コンポーネント追加 🕴 画面構成部品 🕨
	コンポーネント一括追加 処理部品 🕨
	複合コンポーネント作成 ● 入出力 ●
	後音コンホーネンド追加 クラス指定
	貼り付け
	すべて展開
	すべて閉じる
	コンポーネント検索
	GUIコンポーネント検索
	コメント行追加
	コメント行検索
	検索
	コンポーネントID再設定

②追加するクラス名を指定する					
	አታ			×	
	道 で ジェン 1ア プリケーション SY:**** フレーム	動するコンポーネントのクラス4 p.go.aist.dmrc.platform.beans.gu 了解 取 入力したコンポーネント	名を入力してください。 ui.container.PFFrame 消し が表示される		
	:1 EY:"フレーム1"				

3)コンポーネントの一括追加



4)コンポーネントの削除



3.4. 複合コンポーネントの利用

複合コンポーネントの詳細については、5.複合コンポーネントの構築を参照してください。

画面	アプリケーションビルダー メイン画面					
手順	背景にてマウスを右クリックし、複合コンポーネント作成メニューを表示					
	(GUI コンポーネントか非 GUI コンポーネントかを選択)					
	■ アプリケーション					
	KEY: ""					
	コンボーネント通知					
	▲ 複合コンポーネント追加 ■ GUIコンポーネント					
	貼り付け					
	すべて展開					
	すべて閉じる					
	コンポーネント検索					
	GUIコンポーネント検索					
	コメント行検索					
	(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)					
	コンポーネントID再設定					
	複合コンポーネントが作成される					
	■ アブリケーション					

アプリケーションビルダー メイン画面 画面 ①背景にてマウスを右クリックし、複合コンポーネント追加メニューを表示 手順 *「外部参照複合コンポーネント保存先フォルダ」(デフォルトは導入フォルダ ¥AP_DATA_COMB) 内の内容が複合コンポーネント追加メニューの右に表示されます。 ②追加対象のコンポーネントを指定 🗐 アブリケーション KEY : "" コンポーネント追加 コンポーネントー括追加.. <u> 複合コンポーネント作成</u> 複合コンポーネント追加 🏼 🕨 Web機能 グラフ 貼り付け チュートリアル 🕨 ジャグリング (Lesson.10).mzcx すべて展開 テーブル フォルダ管理 (Lesson.7).mz L 住所録 (Lesson.6).mzcx すべて閉じる プロセス 合成モジュール 気象情報 (Lesson 8) mzcx コンポーネント検索... 稼働日時 都道府県情報 (Lesson.9).mzcx GUIコンポーネント検索... 金型履歴 コメント行追加... ロード... コメント行検索... 検索... コンポーネントID再設定 選択したコンポーネントが表示される 🗍 アプリケーション KEY : "" **||**住所録 (Lesson.6) ______ ID : 1 KEY : "住所錄 (Lesson.6)"

2)既存複合コンポーネントの追加

3)ファイル指定による複合コンポーネントの追加





4) 複合コンポーネントの編集



5)編集階層の変更と別ウィンドウ表示

画面	アプリケーションビルダー メイン画面
手順	画面右上の4つのボタンによって階層選択と別ウィンドウ表示を行う。
	[上位階層への移動]表示している状態から順に上の階層に移動
	[最上位階層への移動] 編集対象の最上位の階層に移動
	 [階層選択] 編集対象の階層構造を表示し、選択された階層に移動 ※移動したい階層を選択(下図参照)し、選択ボタン押下により移動。 [新規ウィンドウ] 編集対象の階層を別ウィンドウで表示する
	【アプリケーション階層選択画面】
	₩ アプリケーション階層選択
	 ■ デブリケーション (KEY:"") ■ 住所録 (Lesson.6) [ID:3] (KEY:"住所録 (Lesson.6)") ■ フィルタ [ID:4] (KEY:"")
	■ 新規ウィンドウで表示
	選択 キャンセル
	※[新規ウィンドウで表示]をチェックすると別ウィンドウで表示する

3.5. コンポーネント間の接続設定

1)イベント処理の追加



2)イベント接続先の追加





3)イベント接続先コンポーネントの指定

4)起動メソッドの指定

画面	アプリケーションビルダー メイン画面			
手順	①設定対象の起動メソッド名または接続先コンポーネント上でマウスを右クリックし、			
	メソッド設定メニューを表示			
	②起動メソッド設定メニューを選択する			
	※設定対象の起動メソッド名または接続先コンポーネント上での			
	マウス左ボタンダブルクリック操作でも同様			
	送 接続コンボーネント選択			
	持続コンボーネント選択			
	起動メソッド設定			
	イベント番号設定 ト			
	起動モード			
	記動メソッド追加 ▶			
	コピー			
	貼り付け ▶			
	切り取り			
	自己的论			
	コメント編集			
	接続コンポーネント宣言位置検索			
	ブレークポイント設定/解除			
	記動メソッド設定画面が表示される			
	2 記動メソッド情報			
	メソッド 🔹 全メソッド対象			
	NO型<			

③起動メソッドを選択する 手順 コンポーネントが提供するメソッドのうち、公開設定されているメソッドのみが 表示される。引数の並びを含めて対象のメソッドを選択する。 🛂 起動メソッド情報 X メソッド 📃 全メソッド対象
 テキストの水平方向の配置を設定する(int)

 テキストの水平方向配置を設定する(int)

 テキストを取得する()

 デキストを認定する(String)

 入力フォーカスを設定する(String)

 入力モードを日本語にする()

 入力モードを日本語にする()

 丸力・振力を設定する(boolean)

 編集両否を設定する(Color)
 NO 型 1×88 メソッド/値 了解 取消し 起動メソッドの引数情報が表示される 🛂 起動メソッド情報 x 🔹 💿 全メソッド対象 メソッド テキストを設定する(String) NO 型 説明 取得方法 コンポーネント メソッド/値 メソッド戻り値 0 String テキスト 了解 取消し 【補足】 起動するメソッドが表示されない場合、そのメソッドが非公開の設定になっている。 "全メソッド対象"のチェックボックスをONにすれば、 コンポーネントの全 public メソッドが表示され、選択可能となる。 また、必要に応じてメソッドの情報設定(後述)を行えば、常に表示される。 🛂 起動メソッド情報 ___X__) 📝 全メソッド対象 メソッド setSelectedTextColor(Color) NO 型 B setSelectionColor(Color) メソッド/値 setSelectionEnd(int) setSelectionStart(int) setSize(Dimension) setSize(int,int) setText(String) setToITipText(String) setTransferHandler(TransferHandler) setUI(TextUI) 了解 取消し setVerifyInputWhenFocusTarget(boolean) setVisible(boolean) show() show(boolean) size() toString() transferFocus() transferFocusBackward() transferFocusDownCycle() transferFocusUpCycle() ※注意:メソッドに引数がない場合は以降④~⑦の作業は不要

手順	 ④メソッド引 メソッド引 ・メソッ固定 ・メンガ ・イベン ・メソッ 	数の指定方法 数の指定には」 「してラル指す ッド戻り値 パーネント ット内包データ ットオブジェク ッド処理結果	を選択する 以下の 6 種類があり ^{全)} (イベントオブジェ ト(イベントオブジ	、対象の指定; クトのメソッ ェクト自身)	方法を選択する。 ド戻り値指定)
	🔛 起動メソッド情報				
		براج	ノッド テキストを設定する(String)	▼ 2メソッド対象	R
	NO型 0 String	説明 テキスト	取得方法 コンオ メソッド戻り値 マ	ドーネント	メソッド/値
			固定値 メワッド戻り値 コンボーネント イベント内包 イベント メソッド処理結果		
			選択した指定方法	が表示される	I MA
			ノッド テキストを設定する(String)		*
	NO 쀺	1.12月	取得方法 コンオ	ポーネント	
	0 String	テキスト	メソッド戻り値		
					7般 取消し

手順	⑤設定するコンポーネントを選択する
J //R	● 成之 ゲ ビーン・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	011とリムー アフラド戻り値 ヘトカ・メソッドお動が免ったポーネントな一覧から選択(場佐け下記)
	◇ハガ・ハブラトに知対家コブルーネンドを一見から送扒(保1Fは下記) ◇ 】 カチェック・たし
	<u>0旧に刀広ー コンハーハント</u> Δ 1 カ · 引物 トたてついポーク、トた一覧から 翌日 (場体は下記)
	◇ハガ:別数となるコンホーネントを一見かり迭折(採TFは下記) ◇ユカエーック、コンポーカントの刑が引数の刑トエュズいてわエーック
	◇人力チェックミコンホーネントの空かり数の空とあっているがチェック
	人刀个罢(人刀个可)
	して、「こう」であった。 「こう」、「こう」、「こう」、「こう」、「こう」、「こう」、「こう」、「こう」、
	ンホーネントー覧を表示する
	20 コンポーネント選択
	コンポーネント一覧
	■ [ID:0] アブリケーション (KEY:"")
	■ [ID:2] ボタン (KEY: "ボタン2")
	■ [ID:3] テキストフィールド (KEY:"テキストフィールド3")
	ソート 🔲 名称 キー 💿 昇順 🔿 降順
	画面表示 選択 キャンセル
	対象のコンポーネントを選択
	メソッド テキストを設定する(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
	O[String テキスト メソッド戻り値 ボタン [ID:2] (KEY:*ボタン2")
	7解 取消し
	·





5)起動対象イベント番号の設定

コンポーネントから発生するイベントには、そのイベントの内容をしめす"イベント番号"がついて います。アプリケーションの動作は、このイベント番号によって処理を分岐する必要がありますので、 起動メソッドそれぞれに対して、対象とするイベントの番号を設定することができます。対象のイベン ト番号を設定しない場合は、イベント番号に関わらずすべてのイベントに対して処理が行われます。

画面	アプリケーションビルダー メイン画面
手順	①設定対象の起動メソッド名または接続先コンポーネント上でマウスを右クリックし、
	『イベント番号設定』メニューから『イベント番号設定』を選択
	(常に起動したい場合は『定常起動』を、起動しない場合は『起動しない』を選択)
	テキストを設定する テキストフィールド
	接続コンポーネント選択
	接続コンポーネント選択 ◆
	<u> 近朝しんない</u> <u> 起動メソッド追加</u> →
	コピー
	貼り付け ▶
	切り取り 削除
	コメント編集
	接続コンボーネント宣言位置検索
	ブレークポイント設定/解除
	発生するイベント番号の一覧が表示される
	▲ イベント番号設定
	「 た常記載 イベント番号追加
	NO イベント発生 起動
	0 ボタンが押されたとき 日本動する
	設定 キャンセル

5)起動対象イベント番号の設定 続き

手順	②『定常起動』のチェックを外し、表示されたイベント番号情報を参考に、起動するイ ベント番号を設定する
	W イベント番号設定
	ご 定常起動 イベント番号追加
	NO イベント発生 起動
	0ボタンが押されたとき 図 2 起動する
	設定 キャンセル
	設定された番号が表示される(『定常起動』は非表示)
	■フラリケーション KEY:"
	10:11 KEY:*フレーム1*
	ボタン アクションイベント テキストを設定する テキストフィールド ID:2 アクションイベント アイント 10:3 10:3 10:3 10:5 10:
	M) イベント番号= "0"を設定した場合
	※補足説明 ましイベント番号設定画面上に 指定したいイベント番号が表示されたい場合
	右上の[イベント番号追加]ボタンを押して、番号を入力します。
	また、コンポーネント情報の登録により、表示情報を設定することもできます。
	入力
	() 追加するイベント番号を入力してください。

6) 起動対象の Finally 化設定

イベント処理のメソッド起動は通常設定された順に逐次実行されますが、途中でエラーが発生した場合は続く処理は実行されません。これに対して、途中でエラーが発生しても必ず実行する必要のあるメ ソッドについては"Finally 起動"の設定をすることで、エラーの発生に関係なく必ず実行されます。



7)起動対象の ErrorOnly 化設定

イベント処理のメソッド起動は通常設定された順に逐次実行されますが、途中でエラーが発生した場合にのみ処理を実行したい場合、"ErrorOnly 起動"の設定をします。



8)起動メ	ソ	ッ	ドの削除
-------	---	---	------

アプリケーションビルダー メイン画面
削除対象の起動メソッド名または接続先コンポーネント上でマウスを右クリックし、 『削除』を指示
アキストフィールド 単株コンボーネント選択・ 単株コンボーネント選択・ 単数サンツ・ド遊加・ コピー 貼り付け コピー 貼り付け コピー 貼り付け コピー 脱り付け コピー 比切切 ロリロロ アント編集 接続コンボーネント宣言位置検索 ブレークボイント設定/解除 対象の起動メソッドが削除される
アブリケーション レーム D:1 ボタン アクションイベント D:2 ボタン アクションイベント D:3 KEY:"ラキストフィールド3"

9)起動メソッドのコピー

画面	アプリケーションビルダー メイン画面
手順	コピー対象の起動メソッド名または接続先コンポーネント上でマウスを右クリックし、 『コピー』を指示。Shift キーを押しながら起動メソッド上でクリックし複数の起動メソ ッドを選択し、一括してコピーすることも可能。 → 対象の起動メソッドが貼り付け対象となる
	テキストを設定する デキストフィールド 接続コンボーネント選択・ 接続コンボーネント選択・ 接続コンボーネント選択・ 起動メリッド設定・ 起動メリッド設定 イベント番号設定 レンボーキン・ド追加・ レンボーキン・ レンガーキン・ レンボーキン・ レンガード レンボーキン・ レンガーキン・ レンボーキン・ レンガーキン・ レンガーキン・ レンガーキン・ レンガーキン・ レンガーキン・ ロンド レンガーキン・ ロンド レンガーキン・ ロンド レンガーキン・ ロンド レンガーキン・ ロント

10)起動メソッドのコピー (イベント指示)

画面	アプリケーションビルダー メイン画面
手順	コピー対象のイベント上でマウスを右クリックし、『コピー』を指示 → 対象イベントから起動される一連の処理メソッドが貼り付け対象となる
	アクションイベント テキストを設定する ラキストフィールド D:3 ロビー ロビー 貼り付け 切り取り 削除 コメント編集 コメント編集

11)起動メソッドの切り取り

画面	アプリケーションビルダー メイン画面
<u>画面</u> 手順	アプリケーションビルダー メイン画面 切り取り対象の起動メソッド名または接続先コンポーネント上でマウスを右クリックし、 『切り取り』を指示。Shift キーを押しながら起動メソッド上でクリックし複数の起動メ ソッドを選択し、一括して切り取ることも可能。 → 対象の起動メソッドが貼り付け対象となる
	ゴレークポイント設定/解除 14000000000000000000000000000000000000

12)起動メソッドの切り取り(イベント指示)

画面	アプリケーションビルダー メイン画面
手順	切り取り対象のイベント上でマウスを右クリックし、『切り取り』を指示 → 対象イベントから起動される一連の処理メソッドが貼り付け対象となる
	アクションイベント テキストを設定する ラキストフィールド 起動メワッド追加 ID:3 KEY:"テキストフィールド3" コピー 貼り付け 切り取り 削除 コメント編集 コメント編集


13)起動メソッドの貼り付け

14)起動メソッドの貼り付け(イベント指示)



15)イベント処理の削除



3.6. コンポーネントのコピー/切り取り/貼り付け

1)コンポーネントのコピー

手順 コピー対象のコンポーネント上でマウスを右クリックし	ノ、『コピー』を	
指示。Shift キーを押しながらコンポーネント上でクリ	ックし複数のコン	ポーネントを選
択し、一括してコピーすることも可能。		
→ 対象のコンポーネントが貼り付け対象となる		
■ アプリケーション		
KEY : ""		
ID:1 KEY:"フレーム1"		
	アクションイベント	
ID:2 KEY: <i>"ボタ</i> イベント処理追加 ▶		
U:3 KEY:"テキー 切り取り		
肖邶余		
属性情報設定		
コンポーネント情報表示…		
メソッド起動位置検索		
インデント ト		
15557		

2)コンポーネントの切り取り

画面	アプリケーションビルダー メイン画面
手順	コピー対象のコンポーネント上でマウスを右クリックし、『切り取り』を 指示。Shift キーを押しながらコンポーネント上でクリックし複数のコンポーネントを選 択し、一括して切り取ることも可能。 → 対象のコンポーネントが貼り付け対象となる
	アブリケーション KEY:"" フレーム D:1 KEY:"フレーム1" アクションイベント アクションイベント Fキスト レー D:3 原性情報認定 コンボーネント情報表示 メソッド起動位置検索 インデント
	対象のコンポーネントが切り取られる
	■ アプリケーション KEY:= ■ フレーム ■ フレーム ■ デキストフィールド ■ デキストフィールド3*

3)コンポーネントの貼り付け

画面	アプリケーションビ	ルダー メイン画面		
手順	背景にてマウスを右 ※事前にコピーまた	クリックし、『貼り付け は切り取り処理を行っ	t』を指示 ておくこと	
	アブリケーション レビン レビン	יעאינעפעד	テキストを設定する	<u> テキストフィールド 10:3</u> KEY:"テキストフィールド3*
		コピーしたコンポ	ーネントが追加される	
	アブリケーション とEY:"" フレーム レ:1 レ:2:"70レーム1" ボタン レ:2:"ボタン2" テキストフィールド レ:3:"5+ストフィールド3" ボタン レ:4:" レ:5:"ボタン4" 注意事項 コピー元のコンポー	709ョンイベント 709ョンイベント ネントに設定されてい	_{テキストを設定する} _{テキストを設定する} るイベント処理もすべて	「テキストフィールド D:3 KEY:"テキストフィールド3" 「テキストフィールド D:3: デキストフィールド3" Cコピーされます
	D:2 デキストフィールド D:3 ビビY: "ボタン2* 「テキストフィールド3* 「ボタン D:4 ビビY: "ボタン4* 注意事項 コピー元のコンポー	^{アクションイベント} ネントに設定されてい	- TAT SEAL 2 	しい。 ビビバ・デキストラ Dist ビビバ・デチス てコピーされ

3.7. アプリケーション開始処理/終了処理の設定

アプリケーションを実行するには、開始時の処理、および終了時の処理を設定する必要があります。 開始処理が何もない場合、アプリケーションは起動されません。アプリケーションの開始処理/終了処 理の制御はプラットフォーム基幹機能であるコンポーネントバス(ビルダー上ではアプリケーションコ ンポーネントとして表示)が行います。

1)アプリケーション開始処理の設定

画面	アプリケーションビルダー メイン画面				
手順	画面最上部に表示されているコンポーネント"アプリケーション"から、 通常のコンポーネント間接続と同様に、アプリケーション開始イベント接続を行う。				
	アブリケーション アブリケーション(株イベント フレームを表示する フレーム した(*) レーン レーン レーン レーン レーン レーン レーン レーン				
	例)フレームの表示(メソッド:フレームを表示する)を設定				

2)アプリケーション終了処理の設定

画面	アプリケーションビルダー メイン画面
手順	①アプリケーション終了の設定
	任意のコンポーネントから発生するアプリケーションを終了するイベントを、アプリ
	ケーションの終了処理(メソッド:アプリケーションを終了する)に接続する。終了処
	理ではプロセスを終了させるため、全ての画面や処理が強制的に終了される。
	※終了処理を省略するとアプリケーションが終了しないため、必ず実施すること
	■アブリケーション アブリケーション アブリケーション アブリケーション アブリケーション アブリケーション アブリケーション アブリケーション アブリケーション
	кеу: =
	■ フレーム アクションイベント アブリケーションを終了する ■ アブリケーション 「D_11」 レビン・***
	「ボタン アクションイベント テキストを設定する 「テキストフィールド
	D:3 KEY:"ボタン2" (KEY:"テキストフィールド3"
	(NET: デキストワイール13) 例) フレームのクローズ時(アクションイベント)にアプリケーション終了
	②アプリケーション終了イベントの設定
	◎ アプリケーション終了時に行いたい処理があれば、画面最上部に表示されているアプ
	リケーションコンポーネントから、アプリケーション終了イベントの接続を行う。
	アブリケーション アブリケーション アブリケーション フレームを表示する
	кеү:""
	アプリケーション終了イベント
	■フレーム アクションイベント アブリケーションを終了する ■アブリケーション
	D:1 КЕҮ:"
	ボタン アクションイベント テキストを設定する ゴテキストフィールド ID:2 ID:3 ID:3 ID:3
	10:3 KEY:"テキストフィールド3"

3.8. 画面配置の設定

画面を伴うアプリケーションを構築する場合、ダイアログなどのウィンドウコンポーネントを使用して画面のベースとなるウィンドウを作成し、その上にボタンなどの画面構成部品コンポーネントを配置します。MZ Platform では画面の配置を以下のようにコンポーネントを区分し、これらの組み合わせを 階層化することによって画面を構成します。



1)画面の構成

画面	アプリケーションビルダー 画面編集ダイアログ
手順	①画面編集ダイアログの表示
	『アプリケーションビルダー メイン画面』の【画面編集】ボタンを押下し、
	『画面編集ダイアログ』を表示する
	▲ 画面編集
	- 1 フレーム [ID:1] (KEY:"フレーム1") フレーム [ID:1] (KEY:"フレーム1")
	BLAL (見力回加定21) * (*) 日朝リイスaRAE シワックTrailyra 3 0 10 20 30 40 50 閉じる

1) 画面の構成 続き

<u>コンホーホンド通加</u> ,	■ ボタン [ID:2] (KEY:"ボタン2")
コンボーネントー括道加 新規コンポーネント追加 ▶ 新規コンポーネントー括追加 コンポーネント貼り付け	■ デキストフィールド [ID:3] (KEY:"テキストフィールド3")
子コンポーネント一括削除	
届性情報設定	
元に戻す	

1)画面の構成 続き

手順	②コンポーネントの追加
	b)複数コンポーネントー括追加
	以下のどちらかの操作でメニューを表示し、一括追加を選択する
	・画面左側の階層ツリーのコンテナコンポーネント
	(ダイアログ/フレーム/パネル)を右クリックしてメニューを表示
	・画面右側の画面レイアウトの背景を右クリックしてメニューを表示
	コンポーネント追加
	コンボーネントー括追加
	新規コンホーキンド道加
	コンボーネント貼り付け
	子コンポーネントー 狂笛服金
	、西位生情辛加苦贫乏…
	元に戻す コンポーネント削除
	やり直し
	↓コンボーネント選択画面が表示される
	★ BLIG コンホーホンド ■ ボタン (ID:2) (KEY:"ボタン2")
	□ テキストフィールド [ID:3] (KEY:"テキストフィールド3")
	(ethot注1 ー 2. ゼーウ2. ト本溜垣) てきかぜん2. 地下 キャリナ ボビルクリック
	「「「「「「「」」」」」「「「「」」」」「「」」」「「」」」「「」」」」「「」」」」
	這加」
	選択コンポーネント
	[削除方法] コンボーネントを選択して削除ボタン押下 または ダブルクリック
	設定 キャンセル
	この回面上で配直するコンホーネントを選択する。選択万法は以下の2つ。 しいかと対象コンピュネントを選択する。選択万法は以下の2つ。
	・ 上段から対象コンホーネントを選択し(複数可)、[追加]ホタン押下
	・上校の対象コンホーネントをダブルクリック











千ा石	⑦ 按
于順	
	部品の移動は背景のクリット線にあわせて行われる。クリット線の間隔を調整するこ
	とにより、部品の配置位置を自由に揃えることが可能となる。
	グリッド線間隔の調整は、画面下のスライダーによって設定する。
	編集
	■ フレーム [D:1] (KEY:"フレーム1") ■ #8: 2 (ID:2) (KEY:"プレーム1") ■ #8: 2 (ID:2) (KEY:"プレーム1")
	- デキストフィールド [ID:3] (KEY:"テキストフ. - 「テキストフィールド [ID:3] (KEY:"テキストフ.
	配置 手動配置 ▼ 図 自動サイズ設定 グリッド間隔22 0 10 20 30 40 50 周にあ
特記事項	①画面配置モードの切り替え
	画面配置モードを『手動配置』以外に切り替えると、画面配置は再設定され以前に設
	定した配置情報はすべてクリアされる
	②配置方法の選択について
	◇手動配置
	「「「「」」」) 「「」」」)「」」」) 「」」」)」
	ハーベントは核動するノブットフォーム(USZ WINdow システム)にようし衣小サ ノブノカウリノブなどどあたてたた。彼社広博で取開たにこて計取開ては注意が以
	イスノス子サイスなどか変わるため、絶対座標で配直を行う手動配直では注意か必
	要となる。開発環境/実行環境が同じブラットフォームである場合には、問題なく
	手動配置を行うことができる。
	◇自動配置(横方向/縦方向/領域/矩形分割)
	要件にあわせた配置を行うには、パネルを使用して配置を階層的に構築する必要が
	ある、細かな設定を行うと面面レイアウト設定が指述にたるが、相対的な配置設定
	C(T) = C(T) = C(T) + C(T)
	用的な画面レイアウト設定が可能である。マルチフラットフォームアプリケーショ
	ンの構築時には、自動配置が望ましい。

2) 画面レイアウトの手動設定 続き

3) 画面レイアウトの自動設定

画面レイアウトの設定には以下の4つの自動配置モードが提供されています。作成する画面設計にあ わせて、適切な設定モードを選択してください。

①横方向整列

設定された範囲内で、コンポーネントを左から右に並べます。並びきらない場合は次の行(下段) の左端から順に並べます。

②縦方向配置

設定された範囲内で、コンポーネントを上から下に並べます。並びきらない場合は次の列(右側) の上から順に並べます。

③領域配置

■泊加卡汁

配置領域を上側/下側/右側/左側/中央の5つに分け、コンポーネントの配置を設定します。コ ンポーネント配置時には、上側/下側/右側/左側/中央のいずれかを指示します。このとき、全 体の配置領域にあわせてコンポーネントの表示サイズも自動的に調整されます。

■ ^{20加力広} 領域配置モードの場合、コンオ	ポーネント追加メニューから、配置位置を指定	
コンポーネント追加	■ ボタン [ID:2] (KEY:"ボタン2")	North
コンポーネントー括追加 り	🧊 テキストフィールド [ID:3] (KEY:"テキストフィールド3")	South
新規コンポーネント追加 🕨		East
新規コンポーネント→括追加 ▶		West
コンボーネント貼り付け		Center
子コンポーネントー括削除		
属性情報設定		
元に戻す		
やり直し		

④矩形分割配置

縦横それぞれの表示数を指定することで、表示領域を M×N の領域に分割します。 追加された GUI コンポーネントは左上から右に順に配置され、分割された領域の表示サイズは配置される GUI コ ンポーネントのサイズにあわせて自動調節されます。 4)GUI コンポーネントの順番変更

画面レイアウトが横方向配置などの自動設定になっている場合、GUI コンポーネントの表示順は画面 左側の画面階層ツリーの表示順(登録順)によって決まります。順序を変更するにはツリー上で、その 順序を変更します。

画面	アプリケーションビルダー 画面編集ダイアログ
手順	①GUI コンポーネントの順序変更
	ツリー上のコンポーネントをマウスでドラッグし、任意の位置に移動する。
	□ フレーム [ID:1] (KEY: フレーム1") □ オタン [D2] (KEY: オタン2")
	配置 横方向控列 ▼ 図 自動サイズ設定 グリッド開発5 0 10 20 30 40 50 開日:***
	離した位置で表示順が決定される
	¥ ₩ ⊑क⊮+
	■ フレーム [ID:1] (KEY:"フレーム1") フレーム [ID:1] (KEY:"フレーム1") フレーム [ID:1] (KEY:"フレーム1")
	■ #タン (DD2) (KEY:*#タン2*)
	配置 横方向整列 → ▽自動サイズ設定 グリッド間隔.5
	・ 0 10 20 30 40 50 閉じる

5)GUI コンポーネントのサイズ設定





5)GUI コンポーネントのサイズ設定 続き

3.9. コンポーネント属性の変更

1)アプリケーションビルダー メイン画面からの設定

画面	アプリケーションビルダー	- メイン画面	
手順	対象コンポーネントを右ク	フリックしてメニューを表示し、『属性情報設定』を選	択
	※設定対象コンポーネント	- にて左ボタンダブルクリックでも同様	
	<u>■ アプリケ</u>	アーション アプリケーション開始イベント	
	<u> 同フレーム</u> ID:1		
		<u>y-41"</u>	
	<u>- ボタン</u> ID:2		
	KEY:"#2	()2" イベント処理追加 ▶	
		メソッド起動位置検索 	
		インデント・	
		属性情報設定画面が表示される	
	Text	(ボタン>	
	PropertyEditable	True @ false	
	ComponentPublicName	Vide Video	E
	ComponentID	2	
	ComponentKey		
	ComponentKeys	日本語: 英語: 回 NULL	
	AllowRemoteInvocation	True	
	AllowPullTransfer	⑦ true	
	AllowPushTransfer	─ true	
	MultiLocaleToolTipText	日本語: 英語: I NULL	
	MultiLocaleText	日本語: 英語: III NULL	
	DefaultCapable	● true ◎ false	
	Enabled	💿 true 💿 false	
	FocusPainted	le true 💿 false	
	HorizontalAlignment	0	
	HorizontalTextPosition		•

画面	アプリケーションビルダー [画面編集ダイアログ
手順	設定対象のコンポーネントを	た右クリックしてメニューを表示し、[属性情報設定]を選択。
	(ツリーのノートまたはノレ	ノビュー内の表示)
	コンポーネ コンポーネ 属性情報設定	ント挿入 ▶ ント削除 記 属性情報設定画面が表示される
	↓ 「 「 「 「 「 「 」 、 、 、 ト	¥
	ToolTipText	NULL ^
	Text	<#9>> I NULL
	PropertyEditable	○ true
	ComponentPublicName	V NULL
	ComponentID	2
	ComponentKey	ボタン2 INULL
	ComponentKeys	日本語: 英語: NULL
	AllowRemoteInvocation	─ true
	AllowPullTransfer	─ true
	AllowPushTransfer	⊘ true ● false
	MultiLocaleToolTipText	日本語: 英語: I NULL
	MultiLocaleText	日本語:
	DefaultCapable	í true i ⊘ false
	Enabled	í true ⊂ false
	FocusPainted	í true ⊂ false
	HorizontalAlignment	0
	HorizontalTextPosition	

2) 画面編集ダイアログからの設定

3)属性情報の設定

画面	アプリケーションビルダ-	- 属性編集ダイアログ	
手順	設定対象の属性をキーボ-	ード/マウスから変更する。	
	□ヽ.ポーマヽ.ト同性性相		X
	Toot		NOLL NULL
		97770×37	NOLL
		© true ⊚ false	
			V NOLL
			NULL NULL
	ComponentKeys	日本語: 央語:	NULL
	AllowRemoteInvocation	💿 true 💿 false	
		💿 true 💿 false	
	AllowPushTransfer	True in false	
	MultiLocaleToolTipText	<u>日本語:英語:</u>	NULL
	MultiLocaleText	日本語: サンプルボタン 英語:	NULL NULL
	DefaultCapable	💿 true 💿 false	
	Enabled	💿 true 🔘 false	
	FocusPainted	● true	
	HorizontalAlignment	0	
	¥		×
	□	フレーム [ID:1] (KEY:"フレーム1")	
	<		o 30 40 50 開じる

4)ポップアップメニューからの設定

画面	アプリケーションビルダー 画面編集ダイアログ
手順	以下の5つのコンポーネントに限っては、画面編集ダイアログのポップアップメニューから直接属性を設定することもできる。
	・コンボボックス ・リスト ・チェックボックスグループ
	・ラジオボタングループ ・テーブル
	これらのコンポーネントに限り、画面右の画面レイアウトの設定対象コンポーネント上で 右クリックした場合に表示されるメニューに、[コンポーネント内容設定]が追加されて いるので、それを選択。
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
	コンホーネント的味 デフォルトサイズ設定
	コンボーネント内容設定 道加 末尾に追加
	選択モート 可視行数… レイアウト方法 レ
	[コンポーネント内容設定]のサブメニューは、各コンポーネントが提供する属性設定機 能である。これは、3.10で示す編集可能モードでの実行において表示されるメニューと同
	じである。
	この方法では、設定が終わっても画面レイアウトは自動的に再描画されない。設定を反映 させるには、一度画面レイアウトを左ボタンクリックする必要がある。

3.10. 実行

画面	アプリケーションビルダー メイン画面
手順	①編集可能モードでの実行
	[実行(設定可)]ボタンを押下し、構築したアプリケーションを実行する。対象とするコ
	ンポーネントを右クリックすることで、各コンポーネントが提供する属性設定機能を
	使用して属性の設定を行う。
	サンプルボタン
	<u>────</u>
	サイズ ト
	フォント ト
	色 ▶
	有効/無効 ▶
	配置方法 ▶
	テキスト位置 ▶
	アイコン・
	ニーモニック 🕨
	全白
	②通営エードでの実行
	②屈市 こ 「ていた」 「宇行]ボタンを畑下」 横筑」たアプリケーションを宇行する ちクリック」でも属性
	に、日本のであたい。 設定機能は使用できない。
	サンプルボタン

3.11. デバッグ機能

アプリケーションビルダー上でアプリケーション構築を行う過程において、効率よく作業を進めるた めにデバッグ機能を提供します。ここで提供するデバッガは、アプリケーションの実行を任意の位置で 停止させる「ブレークポイント設定機能」や、動作を確認しながら一つずつ処理を進める「ステップ実 行機能」、実行中のデータの状態などを見る「トレース機能」を備えています。

本機能の詳細な使用方法については、「デバッガ操作説明書」をご覧下さい。

画面	アプリケーションビルダー メイン画面
手順	①デバッガの起動
	メニューバーから [アプリケーション]-[デバッグ] を選択し、
	デバッグ画面を表示する。
	🛃 A. A. K. 📲 🔾
	マメソッドコールスタック
	- 引数と属性

3.12. アプリケーションの保存/ロード

作成したアプリケーションをファイルに保存することで、ロード機能を使用して再利用が可能になりま す。アプリケーションの再編集を行ったり、作成したアプリケーションを別のプラットフォームで使用 したりすることが可能になります。アプリケーションデータ及び複合コンポーネントデータのロード・ 保存標準形式は XML 形式(拡張子:.mzax)です。また、バイナリデータ形式(拡張子:.mzas)でも保存がで きます。複合コンポーネント内で[保存]ボタンを押した時には、複合コンポーネントのみをアプリケー ションとは別に保存することができます。

デフォルトではアプリケーションデータを XML 形式で保存する際にバックアップとして同時にシリア ライズデータを自動保存するようになっています。

1)アプリケーションの保存

画面	アプリケーションビルダー メイン画面
<u>画面</u> 手順	アプリケーションビルダー メイン画面 ①[保存] ボタンをクリック ②保存先ファイル名を指定する ③ファイルタイプを選択する ● Catacooperation ● MetalMold ● Datacooperation ● MetalMold ● Production ● SoffGenerator デスクトップ ■ Tutorial ■ SoffGenerator ■ SoffGenerator ■ SoffGenerator
	コンピュー ター ファイル名: sampleApp.mzax (保存 ネットワーク ファイルタイブ: MZ アプリケーションXML (.mzax) ・ 取消し

2) 複合コンポーネントの保存



3)バイナリデータ自動保存設定/解除



4)アプリケーションのロード

画面	アプリケーションビルダー メイン画面
手順	 ①[ロード] ボタンをクリック 現在のアプリケーションがクリアされることが警告される。 必要であれば現状を保存してから、再度[ロード] ボタンをクリックする。 ②ファイルタイプを選択する ③ロードファイル名を指定する
	₩ 開<
	参照: 🔐 AP_DATA 🔹 🔊 🌮 🖽 📟
	シレビュー ター ファイル名: Sample.mzax
	ネットワーク ファイルタイプ: すべての読込可能なファイル (.mzax, .mzcx, .mzas, .mzcs, .xml, .apl, .cmp) ▼ 取消し

5)アプリケーションの挿入

保存されたアプリケーションを、現在編集中のアプリケーションに挿入する機能です。アプリケーシ ョンのファイル情報を読み込み、その中のコンポーネントやイベント接続など、すべての情報を編集中 のアプリケーションに追加します。ただし、挿入対象アプリケーションの以下の情報については、挿入 時に復元されませんので、注意してください。

<挿入対象がアプリケーションの場合>

・アプリケーションコンポーネントからのイベント接続情報

・コンポーネントからアプリケーションコンポーネントへの接続情報

<挿入対象が複合コンポーネントの場合>

・最上位複合コンポーネントからのイベント接続情報

・コンポーネントから最上位複合コンポーネントへの接続情報

・最上位 GUI 複合コンポーネントの画面レイアウト情報



6)アプリケーションのクリア

画面	アプリケーションビルダー メイン画面
手順	[クリア] ボタンをクリック
	現在のアプリケーションがクリアされることが警告される。
	必要であれば現状を保存してから、再度[クリア] ボタンをクリックする。
	確認
	編集中のすべてのアプリケーション情報がクリアされます。よろしいですか?
	はい(Y) しいいえ(N)

3.13. アプリケーションのパスワードロック機能

コンポーネントの構成で構築したアプリケーションは、内部構造を見たり、編集したりすることが容易にできます。しかし、アプリケーションを実運用するためには、アプリケーション編集を不可にし、 内容を隠したい場合があります。そのため、アプリケーション、および複合コンポーネントにパスワー ドによってロックをかけることができます。

構築したアプリケーションの内容を公開したくない場合、パスワードロック機能を使用してください。 パスワードロックされたアプリケーション、および複合コンポーネントは、実行は通常どおり可能です が、ビルダー上のロードや複合コンポーネントへの階層移動などについてはパスワードの入力が必要と なり、外部への情報漏洩を防ぐことができます。

- ※注意:パスワードロック機能は、バイナファイルとして保存されたアプリケーションおよび複合コン ポーネントに対してのみ有効です。XML として保存されたファイルには、パスワードロックはかか りません。
- 1)アプリケーション/複合コンポーネントへのパスワード設定

画面	アプリケーションビルダー メイン画面
手順	①最上位コンポーネント上にてマウスを右クリックし、[パスワード設定]を指定
	アプリケーション アプリケーション 開始イベント アプリケーション 開始イベント
	KEY:"" イベント処理追加
	KEY: "フレーム バスワード解除
	「ボタン」 コンポーネント情報表示 アクションイベント
	ID:2 KEY: "サンプル メソッド起動位置検索
	■ テキストフィールド
	D:3 KEY:"テキストフィールド3"
	パスワード入力画面が表示される
	パスワード設定
	バスワードの設定
	バスワードの確認
	設定 キャンセル

2)アプリケーション/複合コンポーネントのパスワード解除



3)パスワードロックされたアプリケーション/複合コンポーネントの利用

パスワードロックされているアプリケーションや複合コンポーネントのロード時/挿入時、ロックさ れた複合コンポーネントへの階層移動などの時に、以下のパスワード入力画面が表示され、正しいパス ワード入力時にのみ利用が可能となります。

パスワー	
?	このアプリケーションはバスワードで保護されています。 バスワードを入力してください。 OK キャンセル

3.14. アプリケーション構築時のユーティリティ機能

アプリケーションを構築する際、コンポーネントの数が増えると編集作業が大変になります。そこで、 構築作業を効率化するために、いくつかのユーティリティ機能を提供しています。

1)コンポーネントの検索



2)GUI コンポーネントの検索



3)コンポーネント ID 再設定

コンポーネントの移動や削除で、コンポーネント ID がバラバラになってしまった場合、表示の並び 順に ID の振りなおしを行う機能です。

画面	アプリケーションビルダー メイン画面
手順	背景にてマウスを右クリックし、[コンポーネント ID 再設定]を指定
	コンボーネント追加 ト
	コンボーネントー括2回加 複合コンボーネント作成 ▶
	複合コンボーネント追加 ▶
	見占り付けナ
	すべて展開
	すべて閉じる
	コンボーネント検索
	GUIコンポーネント検索
	コメント行追加
	コメント行復茶
	検索
	コンボーネントID再設定
	₩
	コンポーネントの表示順に ID が振りなおされる

4)マルチウインドウ

アプリケーションの編集中に別の箇所や異なる階層を参照したり、複数箇所を見比べながら編集で きるように、ビルダー画面を複数開くことのできる機能です。

画面	アプリケーションビルダー メイン画面
手順	画面右上のボタンによって新規ウィンドウを開く。
	ドウが、新規で開く。
	MZ Platform アプリケーションビルダー
	ファイル 編集 アブリケーション オブション ヘルブ コメント行検索.
	アプリケーション名称 🗎 📾 📾 🔮 🖄
	●アプリケーション アプリケーション機械イベント フレームを表示する ●フレーム
	プレーム アクションイベット アクリケーションを終了する アクリケーション マション マション
	D:2 レンジングングング レンジングング レンジングング レンジングング レンジング レング レンジング レング
	5#2.F7~///
	UC3: "7=+2, + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2
	【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 ■リネル ■独名画面構成 ■雑名画面構成 ■雑名(感面構成)
	実行 実行(設定寸) 画面編集 「條票編集 ロード 挿入 「保存 上書き保存 クリア 統了
	※注意事項
	初して用いたフィントワにはエ部メーユーと下部小ダン群はついていません

5)接続コンポーネント宣言位置検索

対象とする起動メソッドを呼び出されているコンポーネントの宣言位置を検索して表示する機能 です。

画面	アプリケーションビルダー メイン画面
手順	 ①設定対象の起動メソッド名または接続先コンポーネント上でマウスを右クリックして [接続コンポーネント宣言位置検索]を指定。
	テキストフィールド 接続コンボーネント選択 接続コンボーネント選択 起動メソッド設定 イベント番号設定 起動メソッド追加 起動メソッド追加 コピー 貼り付けす 切り取り 削除 コメント編集
	確認ダイアログが表示される
	確認
	接続コンポーネントの宣言位置を表示しますか? Iまい(Y) いいえ(N)
	②確認の上で[はい]ボタンを押下すると、コンポーネントの宣言位置が表示される。

6)メソッド起動位置検索

対象とするコンポーネントのメソッドを起動している位置を検索して表示する機能です。



7)コンポーネントインデント表示

対象とするコンポーネントをインデント表示する機能です。


8)メソッド折りたたみ表示機能

ビルダーで規模の大きいアプリケーションを構築する際、膨大な数の起動メソッドでビルダーの右 側が埋まってしまうことがあります。メソッド折りたたみ機能は、複数の起動メソッドを折りたたんで 表示をコンパクトにまとめ、アプリケーションの全体像の把握を容易にするための機能です。

閉じる(折りたたむ)

画面	アプリケーションビルダー メイン画面				
手順	①メソッド起動の折りたたみマーク日をクリックします。				
	■ アブリケーション				
	KEY:"月別平均気温表示" KEY:"ラーブル2" テーブルデータを設定する 折れ線グラフ				
	レンジャンシンのためのです。 (D:4 KEY:*平均気温*				
	フレームを表示する フレームを表示する 10:1				
	メソッド起動が折りたたみ状態になる				
	アブリケーション アブリケーション(開始イベント テーブルデータを設定する ゴテーブル KEY:*月別平均気温表示* ロ ロ レ				
	※折りたたみ状態では、最上部のメソッドのみが表示されます。				

画面	アプリケーションビルダー メイン画面				
手順	<u>(</u>)メソッド起動の折りたたみマーク田をクリックします。				
	■ アブリケーション アブリケーション KEY: "月別平均気温表示" アブリケーション開始4 (シト アブリケーション開始4 (シト アブリケーション開始4 (シト アブリケーション開始4 (シト アーブルデータを設定する 「アーブル 「ロ・フ KEY: "テーブル2"				
	メソッド起動が展開される				
	アブリケーション アブリケーション(開始イベント テーブルデータを設定する リテーブル レビン: レビン: レビン: レビン: ケーブルデータを設定する ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・				



初期状態設定

画面	アプリケーションビルダー メイン画面				
手順	①[メニュー]-[オプション]-[起動メソッド群の初期表示]をクリックし、「すべて展開」 または「すべて閉じる」を選択する				
	MZ Platform アプリケーションビルダー				
	ファイル 編集 アラリケーション (オノション) ヘルフ アブリケーション名称 コンポーネント情報編集				
	■ アプリケーション Look&Feel ト KEY:*** ロケール ト				
	 ✓ バイナリデータ自動保存 コンポーネントコピー接続先確認 				
	メモリ使用量表示 メモリ整理(ガベージコレクタ実行)				
	☆~~で選択した状能け、ビルダー終了時にデフォルト値に良ります。デフォルト値				
	へここで選択した状態は、ビルターボリードにアフォルト値に戻ります。アフォルト値 の設定は etc フォルダ内の Platform.ini 内に設定します。詳細は「詳細設定説明書」(導 入フォルダ¥docs¥manual 内)をご覧ください。				

3.15. コメント機能

アプリケーションが大規模になってくると、それぞれの部分が何の処理を行っているのかがわかりに くくなります。そこで、アプリケーションビルダー上にコメントを記入する機能を提供しています。

1)全体コメント(コメント行)の記入



2)全体コメント (コメント行)の検索機能

・ドロップダウンリストからの検索

画面	アプリケーションビルダー メイン画面					
手順	①右上の[コメント行検索]をクリックする。					
	W 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 オガション ヘルブ コメント行後未… ・ ・					
	アブリケーション名称 データベースアクセスサンブル拡張版					
	■ アブリケーション アブリケーション周齢イベント 処理を呼び出す ■ サブルーチン ↑					
	データベース接続フレーム					
	データベース撮続フレーム アクションイベント アブリケーションを終了する アブリケーション ID:1 KEY:"*** [NO:0] [NO:0] [NO:0] [NO:0]					
	テータペー人に接続する (NO.1) (25.5) -					
	<u> </u>					
	<u> していたいでは、 しいたいでは、 いいたいでは、 いいた</u>					
	リウルーチン アクションスペノト シッ ヒシンののパリッション 10:2 レビYッリー生成・設定************************************					
	データベースのツリーを取得する					
	<u>ッリーデータを設定する</u> レーデータを設定する レーズース表示フレーム レーズー ドログーー					
	[アイコン凡州] ■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 ■視会(画面構成) ■視会(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					
	コメント行の一覧が表示される					
	コメント行検索 <					
	データベース接続フレーム					
	データベースツリー作成					
	<u>処理を呼び出す</u> データベース切断処理 データベース表示フレーム					
	拡張データベースアクセス					
	<u>処理を呼び出す</u> 施行処理 終了処理					



・ポップアップメニューからの検索

画面	アプリケーションビルダー メイン画面
手順	①背景にてマウスを右クリックし、[コメント行検索]を指定
于順	① 目気に C く ワス を 石 ク リ ッ ク し、 「コ メ ク ト 行換 案 」 を 指定 コンボーネント 追加 コンボーネント 作成 被合 コンボーネント作成 は 合 コンボーネント 修泰 ロ ンボーネント 検索 コンド 行検索 マント 行換 案 ロ スント 行選 択画 面が表示 される
	データベース切断処理 データベース伝統フレーム データベース表示フレーム 拡張データベースアクセス 終了処理 開始処理
	 ● 昇順 ● 降順 ○ 確据
	※操作方法 ・昇順/降順のソート表示が可能 ②検索対象のコメント行を選択し「選択]ボタンを押下するト 選択したコメントに移動
	②(☆糸対象のコメント1)を迭折し、「迭折」小ダンを押下9 ると、迭折したコメントに移動する する
	※対象コメント行をダブルクリックしても同様

3)イベントコメントの記入

画面	アプリケーションビルダー メイン画面				
手順	①設定対象のイベント上でマウスを右クリックし、[コメント編集]を指定				
	「ボタン アクションイベント ID:2 起動メソッド追加 ドEY:"サンブルボタン" 起動メソッド追加 コピー BLり付け ID:3 ドEY:"テキストフィールド3" ドEY:"ナキストフィールド3" コメント編集				
	コメント入力画面が表示される				
	イベントコメントを入力してください。				
	「一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一				
	※操作方法 ・キーボードよりコメントの内容を入力				
手順	②了解ボタンをクリック コメントが追加される				
	「ボタン このイベントで重要な処理を行う」 ID:2 アクションイベント KEY: "サンプルボタン" ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・				
	□ テキストフィールド □D:3 KEY: "テキストフィールド3" KEY: "テキストフィールド3"				

画面 手順 ①設定対象の起動メソッド上でマウスを右クリックし、[コメント編集...]を指定 テキストフィールド テキストを設定する 接続コンポーネント選択... 接続コンポーネント選択 起動メソッド設定… イベント番号設定 起動モード 起動メソッド追加 コピー 貼り付け 切り取り 削除 コメント編集... 接続コンポーネント宣言位置検索 ブレークポイント設定/解除 コメント入力画面が表示される x コメント入力 起動メソッドコメントを入力してください。 了解 取消し ※操作方法 ・キーボードよりコメントの内容を入力 手順 ②了解ボタンをクリック コメントが追加される この処理はとても重要 テキストフィールド テキストを設定する "テキストフィールド3"

4)起動メソッドコメントの記入

コメントのコピー/切り取り/貼り付け/削除についても同様に、メニューから項目を選択することに より実行可能です。

3.16. その他機能

1)メニューバー

アプリケーションビルダー画面にはメニューバーが提供されており、以下の機能を呼び出すことがで きます。

メニュー	アイテム	説明	
ファイル	新規作成	以下のプログラムの新規作成を行う	
		①アプリケーション	
		②複合コンポーネント	
		③複合 GUI コンポーネント	
	ロード	アプリケーションのロードを行う([ロード]ボタンと同等)	
	挿入	アプリケーションの挿入を行う([挿入]ボタンと同等)	
	保存	アプリケーションの保存を行う([保存]ボタンと同等)	
	上書き保存	アプリケーションの上書き保存を行う	
		(「上書き保存」ボタンと同等)	
	画像出力	ビルダー画面を画像ファイルとして保存する	
	クリア	アブリケーションのクリアを行う([クリア]ボタンと同等)	
	終了	アブリケーションビルダーを終了する	
/ - //	/	(「終了」ボタンと同等)	
編集	元に戻す	操作を行った状態を操作前の状態に戻す	
	やり直し	[元に戻す]により元に戻された操作を冉実行する	
		選択された要素をコピーする	
	貼り付け	コビーまたは切り取られた要素を貼り付ける	
	切り取り	要素を切り取る	
	前际	要素を削除する	
	インテント	コンホーネントの表示位直を調整する	
アフリケーション		アプリケーションを実行する([実行]ホダンと同寺)	
	美行(設定可)	設定可能モート (アノリケーンヨンを美行する)	
	面面短度	(【天1](設たり川小ダンと回寺) 両面しノマウトた短集する(「両面短集】ギタント回答)	
		回回レイプリトを補朱9る([回回補朱]小グノと回寺) 「「「「「」」」「「「「「「」」」」」」	
	「「「「「「「「」」」	版示レイプリトを補未する([版示補未]小グノと回守) デバッグ機能を記動する	
オプション	コンポーネント情報編集	フンポーネント信報を編集する	
7 7 7 3 7		コンホーホンド情報を編末する LookをFool を以下から設定する	
	LOOK	()Windows	
		2)Motif	
		(3)Java (Metal)	
	ロケール	ロケールを以下から設定する	
		①日本語	
		②英語	
	ログ出力レベル	ログ出力レベルを以下から設定する	
		①出力なし	
		②エラーログのみ出力	
		③重要ログのみ出力	
		④全て出力	
	起動メソッド群の初期表	起動メソッドの折りたたみ表示の初期状態を設定する	
	示	①すべて展開	
		②すべて閉じる	
	ツールチッブ表示 	コンボーネント説明および起動メソッド説明等のツール	
		ナツノ表示有無を設定する	
	ハイナリテータ目動保存 	アノリケーンヨンテータを XML 形式で保存する場合にパ	
		ツクチツノとし(同時にンリチフイ人ナータを目動保仔 オスかじらかた訳字オス	
		9 るかとつかと設足9 る	

	メモリ使用量表示	以下のメモリサイズを別ダイアログで表示する
		①確保しているメモリサイズ
		②使用しているメモリサイズ
	メモリ整理	ガベージコレクタ実行
ヘルプ	プラットフォーム	プラットフォームのライセンス情報を表示する
	ライセンス情報	
	アプリケーション	登録されているアプリケーションライセンス情報の表示、
	ライセンス情報	および更新を行う
	バージョン情報	実行中の MZ Platform と Java のバージョンを表示する

2)ダブルクリックによるショートカット

アプリケーション構築画面には、マウスの左ダブルクリックによるショートカット操作があります。

マウス位置	状態	動作
背景	—	コンポーネントー括追加
コンポーネント(左側矩形)	複合コンポーネント	コンポーネント編集(下位階層編集)
	上記以外	コンポーネント属性編集
イベント(赤円)	-	起動メソッド追加
起動メソッド名(文字列)	-	起動メソッド情報編集
接続先コンポーネント(右側矩形)	接続先未設定時	接続先コンポーネント選択
	接続先既設定時	起動メソッド情報編集

4. 帳票の作成

プラットフォーム上で扱われているデータや画面を帳票出力するために、以下の機能を提供します。 1)帳票レイアウト設計機能

アプリケーション構築機能として、帳票レイアウト設計機能を提供します。帳票レイアウト設計 機能は、アプリケーションビルダーの機能として開発者向けに提供します。

2) 帳票印刷機能

帳票の印刷、プレビュー表示、印刷設定の機能を、帳票コンポーネントとして提供します。

4.1. 帳票のデータ構造

帳票は以下のような構造で構成されています。帳票を作成する作業は、以下の構造を構築することに なります。



4.1.1. 帳票コンポーネント

- ◆機能
 - ・印刷機能
 - ・印刷プロパティ(プリンタ選択/用紙設定/枚数設定など)設定
 - ・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)
- ◆ビルダー上での操作
 - ・属性の設定

4.1.2. 帳票構成要素: ラベル要素

◆概要

文字列1つについて表示するための帳票要素。JavaのStringクラスに対応し、文字列長に制限はなく、サイズにあわせて折り返して描画される。また、文字列中に改行コードがある場合、その位置にて改行される。

[イメージサンプル]

テキスト

- ◆属性
 - ・文字フォント
 - ・文字色
 - ・下線有無
 - · 文字表示位置
 - ・テキスト余白(縦方向/横方向)
 - ・行間
 - ・背景色
 - ・罫線色
 - ・罫線幅
 - ・繰り返し印刷モード
- ◆データ設定方法
 - ①テキスト入力

固定値として文字列を入力する。

②メソッド戻り値

通常描画時はコンポーネントのメソッドの戻り値で文字列を設定する。繰り返し印刷指定時はメ ソッドの戻り値で PFObjectTable とデータ取得列インデックスとを設定する。ただし、指定でき るメソッドは引数の無いものに制限する。

- ◆画面操作
 - ・属性の変更
 - ・描画位置の設定
 - ・描画サイズの設定
 - ・テキストの入力
 - ・データ取得メソッドの設定

4.1.3. 帳票構成要素:テーブル要素

◆概要

表形式のデータを描画するための帳票要素。プラットフォームが提供するデータ構造 PFObjectTable クラスに対応する。各セルの描画要素は文字列、バーコード、QR コード、イメー ジの何れかをカラム単位で選択できる。文字列はすべて横幅にあわせて折り返して描画される。ま た、文字列中に改行コードがある場合、その位置にて改行される。バーコード、QR コードはデー タの文字列表現から、イメージはファイルパスの文字列表現から生成され、原寸表示/枠幅に合わ せた縮小表示を選択できる。縦幅については、デフォルトではテキストまたはイメージの描画に必 要な領域にあわせて自動調整され、それより小さな値には設定できない(大きな値に設定すること は可能)。なお、描画属性については、表全体の設定、ヘッダ行に対する設定、カラム毎にデータ 行に対する設定ができる。

[イメージサンプル]

<i>列</i> A	<i>列 B</i>	<i>列</i> C
データ A1	データ B1	データ C1
データ A2	データ B2	データ C2

◆属性

- ①テーブル全体属性
 - ・文字フォント/文字色/下線有無/文字表示位置
 - ・テキスト余白(縦方向/横方向)/行間
 - ・背景色/罫線色/罫線幅(外枠/ヘッダ区切/縦方向/横方向)
 - ・テーブル縦幅の自動調整モード
 - ・繰り返し印刷モード
- ②ヘッダ行属性
 - ・ヘッダ行描画有無
 - ・テーブル全体属性の引継ぎ
 - ・文字フォント/文字色/下線有無/文字表示位置
 - ・テキスト余白(縦方向/横方向)/行間/背景色

③カラム別データ行属性 ※各カラム単位で設定

- ・テーブル全体属性の引継ぎ
- ・描画タイプ(文字列/バーコード/QR コード/イメージ)
- ·余白(縦方向/横方向)/表示位置/背景色
- ・文字フォント/文字色/下線有無/行間
- ・テキスト表示パターン(対象データ型:数値/論理値/日付)
- ・バーコード体系/データ表示/チェックディジット/表示サイズ
- ・QR コードバージョン/倍率/エラー訂正レベル
- ◆データ設定方法

コンポーネントのメソッドの戻り値で PFObjectTable を設定する。ただし、指定できるメソッドは 引数の無いものに制限する。

- ◆画面操作
 - ・属性の変更(テーブル全体/ヘッダ行/カラム別データ行)
 - ・描画位置の設定
 - ・描画サイズの設定(横方向のみ)

MZ Platform アプリケーションビルダー操作説明書

- ・カラム幅の設定
- ・行高さの設定
- ・縦方向行高さの自動調整
- ・データ取得メソッドの設定

4.1.4. 帳票構成要素: バーコード要素

◆概要

バーコード表示するための帳票要素。文字列情報を入力とし、指定されたコード体系に変換して出 力する。バーコードイメージは拡大/縮小が可能。なお、バーコードの描画はイメージに落とすこ とはしないため、拡大や縮小によって文字情報や線情報が崩れてしまうことはない。

[イメージサンプル]



◆属性

- ・コード体系
- ・データ文字列表示有無
- ・チェックディジット有無
- ·罫線色
- ・罫線幅
- ・繰り返し印刷モード
- ◆データ設定方法
 - ①テキスト入力

固定値として文字列を入力する。

②メソッド戻り値

通常描画時はコンポーネントのメソッドの戻り値で文字列を設定する。繰り返し印刷指定時はメ ソッドの戻り値で PFObjectTable とデータ取得列インデックスとを設定する。ただし、指定でき るメソッドは引数の無いものに制限する。

- ◆画面操作
 - ・属性の変更
 - ・描画位置の設定
 - ・描画サイズの設定(横幅のみ/縦幅は自動調整)
 - ・テキストの入力
 - ・データ取得メソッドの設定

4.1.5. 帳票構成要素: QR コード要素

◆概要

QR コードを表示するための帳票要素。文字列情報を入力とし、指定されたバージョン、エラー修 正レベル等に従って描画される。

[イメージサンプル]



- ◆属性
 - ・倍率
 - ・バージョン
 - ・エラー訂正レベル
 - ・表示サイズ
 - ・罫線色
 - ・罫線幅
 - ・繰り返し印刷モード
- ◆データ設定方法
 - ①テキスト入力

固定値として文字列を入力する。

②メソッド戻り値

通常描画時はコンポーネントのメソッドの戻り値で文字列を設定する。繰り返し印刷指定時はメ ソッドの戻り値で PFObjectTable とデータ取得列インデックスとを設定する。ただし、指定でき るメソッドは引数の無いものに制限する。

- ◆画面操作
 - ・属性の変更
 - ・描画位置の設定
 - ・描画サイズの設定(横幅のみ/縦幅は自動調整)
 - ・テキストの入力
 - ・データ取得メソッドの設定

4.1.6. 帳票構成要素:イメージ要素

◆概要

イメージを表示するための帳票要素。Java の Image データを入力とし、出力する。イメージの描 画は横幅にあわせて調整され、縦方向のサイズは縦横比率を維持した状態で自動調整される。

- ◆属性
 - ・罫線色
 - ・罫線幅
 - ・繰り返し印刷モード
- ◆データ設定方法
 - ①ファイルからロード
 - イメージファイルを指定し、イメージ情報をロードする。
 - ②メソッド戻り値

コンポーネントのメソッドの戻り値でイメージ(Image)を設定する。繰り返し印刷指定時はメ ソッドの戻り値で PFObjectTable とデータ取得列インデックスとを設定する。ただし、指定でき るメソッドは引数の無いものに制限する。

- ◆画面操作
 - ・属性の変更
 - ・描画位置の設定
 - ・描画サイズの設定(横幅のみ/縦幅は自動調整)
 - ・イメージファイルからのロード
 - ・データ取得メソッドの設定

4.1.7. 帳票構成要素:画面イメージ要素

◆概要

GUI 画面のイメージをそのまま描画するための帳票要素。プラットフォームが提供する GUI コン ポーネントである PFGUIComponent を指定する。イメージの描画は横幅にあわせて調整され、縦 方向のサイズは縦横比率を維持した状態で自動調整される。なお、GUI コンポーネントの描画はイ メージに落とすことはせず、GUI コンポーネントの描画ロジックを使用するため、拡大や縮小によ って文字情報や線情報が崩れてしまうことはない。

- ◆属性
 - ・罫線色
 - ・罫線幅
- ◆データ設定方法

アプリケーション上に存在する GUI コンポーネントを設定する。

- ◆画面操作
 - ・属性の変更
 - ・描画位置の設定
 - ・描画サイズの設定(横幅のみ/縦幅は自動調整)
 - ・対象 GUI コンポーネントの設定

4.2. 帳票作成/印刷の流れ





4.3. 帳票作成の操作手順

1)帳票コンポーネントの追加

帳票印刷機能を提供する『帳票コンポーネント』をアプリケーションに追加します。帳票コンポーネントは、GUIや汎用ユーティリティと同じように標準コンポーネントとして提供され、追加操作は通常のコンポーネント追加操作によって行います。

2) 帳票の属性設定

帳票が持つ属性を設定します。設定できる属性は先に列挙したもので、これらを設定するための設定 画面を提供します。



3)帳票レイアウトの設定

作成する帳票にあわせてページや帳票要素を追加し、それぞれについて設定を行うことで、対象となる帳票のレイアウトを設定します。

a)ページの追加

ページを追加します。

画面	アプリケーションビルダー 帳票レイアウト設定画面			
手順	①帳票階層ツリーの帳票ページノードからマウス右ボタンクリックでメニューを表示			
	※帳票レイアウト設定画面の[ページ追加]	ページ追加 🕴 前	Ī	
	ボタンでも同様	ページ削除 後	ŝ	
		帳票要素追加 ▶		
		帳票要素貼り付け		
	②追加したい位置を選択(前/後)→ ページが追加される			

b)帳票要素の追加

帳票要素を追加します。プラットフォームから提供されている帳票要素から、目的に適した要素を 選択します。

画面	アプリケーションビルダー 帳票レイアウト設定画面
手順	①帳票階層ツリーの帳票ノードまたはページノードからマウス右ボタンクリックでメニ
	ューを表示
	※帳票イメージの余白部分でのマウス
	「「「「「」」」」」「「」」」」」「「」」」」「「」」」」」」「「」」」」」」
	帳票要素追加 ラベル要素 ラベル要素
	帳票要素貼り付け テーブル要素
	②追加したい帳票要素種類を選択
	→ 指定した帳票要素が帳票階層ツリーとレイアウト画面に追加される
	Wiem Image: Comparison of
	編集 日 日 帳票 [D:13] (KEY."データ帳票") 縮小 100% 拡大 前ページ 次ページ ページ酒除 1/1
	□ - F - F - F - F - F - F - F - F - F -
	□-P.1 - ▲ ラベル (月別平均気温)
	 ▲ ラヘル (■水温 クラノ) ■ 画面イメージ (KEY:平均気温 TID:5))
	0 10 20 30 40 50

c)帳票要素のデータ設定

追加した帳票要素について、データの設定を行います。データ設定方法は帳票要素によって異なり ます。

画面	アプリケーションビルダー 帳票レイアウト設定画面
手順	①帳票階層ツリーの要素ノードからマウス右ボタンクリックでメニューを表示
	※帳票イメージの各要素でのマウス右ボタンクリックでも同様
	②データ設定を行う
	() 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、
	「指定方法A]テキスト入力
	描画したいテキストを入力する。改行を含んだ文字列も設定可能。
	テキストデータを入力してください。
	[指定方法B] コンポーネントのメソッド戻り値設定
	・通常描画時
	コンポーネントとメソッドを選択することにより、データの取得方法を設定。
	指定できるメソッドは、戻り値が void でなく、引数がないものに限定。
	🙀 データ取得設定
	取得元コンポーネント 変更…
	データ取得メソッド
	□ 全メソッド
	設定 キャンセル
	・繰り返し印刷指定時
	コンポーネント、メソッド、データ取得列インデックスを選択することにより、
	データの取得方法を設定。指定できるメソッドは、戻り値が PFObjectTable に
	友快可能で、 「奴かない ひのに 阪足。
	🕎 データ取得設定
	取得元コンボーネント 変更
	データ取得メソッド
	□ 全メソッド
	データ取得列インデックス 0 😌
	[指定方法 C] ページ番号
	自動的にページ番号が設定されます。

c)帳票要素のデータ設定 続き

手順	◇テーブル形式の場合
	コンポーネントとメソッドを選択することにより、データの取得方法を設定。
	指定できるメソッドは、戻り値が PFObjectTable に変換可能で、引数がないものに
	限定。
	🔛 データ取得設定
	取得元コンポーネント 変更…
	データ取得メソッド
	🗌 全メソッド
	設定 キャンセル
	◇バーコード形式、QR コード形式の場合
	[指定方法A] テキスト入力
	バーコード化または QR コード化したいデータを人力する。
	テキストデータを入力してください。
	テキストデータを入力してください。
	バーコードの場合 QR コードの場合
	[指定方法B] コンポーネントのメソッド戻り値設定
	・通常描画時
	コンポーネントとメソッドを選択することにより、データの取得方法を設定。
	指定できるメソッドは、戻り値が void でなく、引数がないものに限定。
	■ 取得元コンポーネント 変更
	データ取得メソッド
	設定キャンセル
	• 繰り返し印刷指定時
	「ホッムし日初月日に町 コンポーネント メソッド データ取得列インデックスを選択することにとい
	データの取得方法を設定 指定できるメソッドは 豆は値が DEObjactTable に
	変換可能で、引数かないものに限定。
	₩ データ取得設定
	取得元コンポーネント 変更
	データ取得メソッド
	□ 全メソッド
	設定 キャンセル

c)帳票要素のデータ設定 続き

貝	
	[指定方法A]イメージファイルからのロード
	イメーシンティルを迭代する。
	■ mz_logo_long.gif 最近使った項
	マイドキュ
	1日 と じ こ る ス ノ ツ ト は 、 戻 ツ 旭 か 1111 age に 変 換 可 肥 じ 、 51 数 か な い も の に 限 データ取得設定
	データ取得メソッド
	データ取得メソッド 🗌 全メソッド
	データ取得メソッド
	データ取得メソッド
	データ取得メソッド □ 全メソッド ↓ 設定 キャンセル
	データ取得メソッド 金メソッド 設定 キャンセル ・繰り返し印刷指定時
	 データ取得メソッド ・繰り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ
	データ取得メソッド ・繰り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ データの取得方法を設定。指定できるメソッドは、戻り値が PF0b jectTable
	データ取得メソッド ・ 繰り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ データの取得方法を設定。指定できるメソッドは、戻り値が PF0b jectTable 変換可能で、引数がないものに限定。
	 データ取得メソッド ・ 金メソッド ・ 繰り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ データの取得方法を設定。指定できるメソッドは、戻り値が PFObjectTable 変換可能で、引数がないものに限定。
	 データ取得メソッド 金メソッド ・金メソッド ・金メソッド ・・・・・・・・・・・・・・・・・・・・・
	データ取得メソッド ・金メソッド ・線り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ データの取得方法を設定。指定できるメソッドは、戻り値が PF0b jectTable 変換可能で、引数がないものに限定。 アーク取得設定 取得元コンポーネント
	データ取得メソッド ・ ・ 繰り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ データの取得方法を設定。指定できるメソッドは、戻り値が PFObjectTable 変換可能で、引数がないものに限定。 ジデータ取得設定 アータ取得設定 アータ取得ションボーネント
	データ取得メソッド ・繰り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ データの取得方法を設定。指定できるメソッドは、戻り値が PF0b jectTable 変換可能で、引数がないものに限定。 アータ取得設定 取得元コンポーネント コンポーネント、メソッド、データ取得列インデックスを選択することによ データ取得方法を設定。指定できるメソッドは、戻り値が PF0b jectTable 変換可能で、引数がないものに限定。
	データ取得メソッド ・ 金メソッド ・ 繰り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ コンポーネント、メソッド、データ取得列インデックスを選択することによ データの取得方法を設定。指定できるメソッドは、戻り値が PF0bjectTable 変換可能で、引数がないものに限定。 アータ取得設定 アータ取得設定 アータ取得設定 アータ取得知インデックス ・ アータ取得列インデックス
	データ取得メソッド ・ 全メソッド ・ 繰り返し印刷指定時 コンポーネント、メソッド、データ取得列インデックスを選択することによ データの取得方法を設定。指定できるメソッドは、戻り値が PFObjectTable 変換可能で、引数がないものに限定。 アーク取得設定

c)帳票要素のデータ設定 続き



d)帳票要素の配置/サイズの設定

画面	アプリケーションビルダー 帳票レイアウト設定画面
手順	①帳票イメージの各要素をドラッグすることで位置設定が可能(全要素共通)
	②帳票イメージの各要素の縁をドラッグすることで描画サイズの設定が可能
	ラベル :横サイズ[右端]/縦サイズ[下端]/全体サイズ[右下隅]
	テーブル :カラム幅[カラム右端]/行高さ[行下端]
	バーコード : 横方向[右端] ※縦方向は自動リサイズ
	QR コード : 横方向[右端] ※縦方向は自動リサイズ
	イメージ :横方向[右端] ※縦方向は自動リサイズ
	画面イメージ :横方向[右端] ※縦方向は自動リサイズ

e)帳票要素の属性設定

追加した帳票要素について、属性設定を行います。属性設定方法は帳票要素によって異なります。

画面	アプリケーションビルダー 帳票レイアウト設定画面							
手順	①帳票階層ツリーの要素ノードからマウス右ボタンクリックでメニュー表示							
	※帳票イメージの各要素でのマウス右ボタンクリックでも同様							
	②属性設定を選択							
	③属性設定画面上で入力							
	◇ラベル形式の場合							
	文字フォント 変更 ブレビュー							
	文字色 変更 重あぁアァアァ Sample 123							
	下線 🔲 表示する							
	縦余白(Point) 5€							
	横余白(Point) 5€							
	行間源(Point) 0 🔦							
	文字位置 左上 ▼							
	指景色 <u>変更</u>							
	罫線色 変更							
	野線太さ (Point) 0.5 🔦							
	■ 繰り返し印刷							
	設定 キャンセル							
	· · · · · · · · · · · · · · · · · · ·							
	イテーブルデータの指定列のセルの値を順に設定します。最後の要素においてテ							
	ーブルデータの最終行主で到達していたい提合にはその要素の方在するページ							
	カーシュージョンをつうして、ショントを言うして							

e)帳票要素の属性設定 続き

手順	◇テーブル形式の場合				
	「テーブル描画設定				×
	テーブル設定	罫線設定		ヘッダ行設定	
	文字フォント 変更	罫線色	変更	📝 ヘッダ行を表示す	3
	文字色 変更			📝 テーブル属性利用	
	下線 🗌 表示する	外枠(Point)	1	文字フォント	変更
	縦余白(Point) 5€	ヘッダ線(Point)	1	文字色	変更
	横余白(Point) 5 €	描约(Point)	0.5	下線	□ 表示する
	行間隔(Point) 0 🚔	1708(FOILT)		縦余白(Point)	5
	文字位置 左上 •	縦線(Point)	0.5	横余白(Point)	5 🛓
				行間隔(Point)	0
	表示行数 10 —			文字位置	左上 👻
	▼ テーブル高さ自動調整			背景色	変更
	テーブルコンポーネント属性利用			テーブルコンポーネ	ント属性利用
	□フォント □文字色 □背景色			□フォント	
	プレビュー				
	列-A	列-B	列-C 列-D		
	データ-AD	データ-B0	データ・CO デー	夕-D0	
	データ-A1	データ-B1	データ-01 デー	9-D1	
	データ-A2	データ-B2	データ・02 デー	9-D2	
		設定	キャンセル		
		した場合、設定	定したテーブ	ルデータが指定	こした行数で収ま
	りきらない場合、同様	に設定した次の	のテーブル要素	素がある場合に	こはそのテーブル
	に残りのデータを設定	します。最後の	のテーブル要素	素に収まりきら	ない場合にはそ
	のテーブルの存在する	ページを印刷	時に繰り返し	印刷します。	

e)帳票要素の属性設定 続き



e)帳票要素の属性設定 続き



f)テーブルカラム描画設定

テーブル形式要素の場合、各テーブルカラム毎に属性を設定できます。

画面	アプリケーションビルダー 帳票レイアウト設定画面
手順	①帳票階層ツリーのテーブル帳票要素ノードからマウス右ボタンクリックで
	メニュー表示
	※帳票イメージのテーブル要素でのマウス右ボタンクリックでも同様
	②カラム描画設定メニューから対象のカラムを選択
	③テーブルカラム描画設定画面上で描画タイプ(文字列、バーコード、QR コード、イメ
	ージ)を選択し、タイプに応じて属性を設定 ヘカウマル・カウル剤の場合
	テーフルカラム描画設定
	 ◎ 文字列 ◎ バーコード ◎ QRコード ◎ イメージ
	文字フォント 変更
	文字色 変更
	下線 💿 表示する
	縦余白(Point) 5 条
	横余白(Point) 5 点
	行間源(Point) 0 一
	文字位置
	背景色 変更
	設定 キャンセル
	◇文字列:数値型、日付型の場合
	テーブルカラム描画設定
	 ● 文字列 ○ バーコード ○ QRコード ○ イメージ
	文字フォント 変更
	文字色 変更
	下線 🔲 表示する
	縦余白(Point) 5 📮
	横余白(Point) 5 ♣
	行間隔(Point) 0 🔔
	文字位置 左上 マ テキスト形式
	[表示パターン例:#,##0円]
	3桁ごとにカンマ表示/1の位は必ず表示/小数点以下なし/単位は"円"
	数 個 空: java.text.DecimalFormat のパターン 人子列を使用
	」 「「「」」」「」」」「」」」「」」「」」「」」「「」」」「「」」」」「「」」」」

f)テーブルカラム描画設定 続き

>文字列:論理型の場合	•	
テーブルカラム描画設定		
◎ 文字列 💿 バーコード	◎QR⊐ード ◎イメージ	プレビュー
文字フォント	変更	
文字色	変更	7 -9
下線	 一 表示する	
縦余白(Point)	5	
横余白(Point)	5	
行間隔(Point)	0	テキスト形式
文字位置	左上 ▼	真 TRUE
背景色	変更	偽 FALSE
	設定(キャン	ンセル
バーコードの場合		
テーブルカラム描画設定		_
◎ 文字列 : ◙ バーコード	◎QR⊐ード ◎イメージ	プレビュー
コード体系	CODE39 -	
データ表示	🔲 表示する	
チェックディジット	🔲 付加する	
表示サイズ	📝 原寸表示する	
縦余白(Point)	5	
横余白(Point)	5	
表示位置	左上 ▼	
背景色	変更	
	設定(キャン	ンセル
>QR コードの場合		
テーブルカラム描画設定		
◎ 文字列 ◎ バーコード	◉QR⊐ード ◎イメージ	プレビュー
倍率	2	
バージョン	0	
エラー訂正レベル	L •	
表示サイズ	🔽 原寸表示する	
縦余白(Point)	5	
横余白(Point)	5	
Nortel (Found)		
表示位置	中央 •	
表示位置	中央 ▼	

f)テーブルカラム描画設定 続き

手順	\diamond	イメージの場合			
		テーブルカラム描画設定			
		◎ 文字列 ◎ バーコード	◎ QR⊐ード ◎イメージ	プレビュー	
		表示サイズ	🔽 原寸表示する		
		縦余白(Point)	5		1
		横余白(Point)	5		
		表示位置	左上 ▼	₹\$	
		背景色	変更	6	
			設定(キャン	ンセル	

4.4. 帳票印刷手順

4.4.1. 帳票印刷プレビュー

帳票コンポーネントは印刷イメージを確認するために、プレビュー画面表示メソッドを提供します。 このメソッドの引数は、パラメータ設定ダイアログの親コンポーネントを指定するためのものです。プ レビュー画面は 50~250%の倍率で表示ができ、アプリケーション構築時と同様に、帳票および帳票要 素の描画属性を編集することも可能です。

	アプリケ-	ーション	帳票印刷に	プレビュー	-画面				
手順	 ①表示の 	拡大/縮小	N N						
	拡大	:[拡大]オ	ドタンの押	ጉ					
	縮小	:[縮小]オ	ドタンの押	ጉ					
	等倍表	表示 · [10	0%]ボタン	の押下					
	のページオ	以引·····		•••					
		いおかい	「盐 _の 、、	<u>`</u>]+` / \ /	~변도				
	則べ-	ーンを到こ		/] 小ダン(ᇬᄪᆕ				
	火へ-	ーン移動:	こし 火ヘーン	/]ホタン(の押下				
	(3) 配置/1	サイズなど	この設定						
	・帳票・	イメージの	D各要素を	ドラッグ	することで	位置設定	が可能(全	雲素共通)	
	・帳票~	イメージの	の各要素の	縁をドラ	ッグするこ	とで描画	サイズの設	定が可能	
	④属性設況	定							
	帳票イン	メージの名	各要素での	マウス右フ	ボタンクリ	ックでメ	ニューから)、	
	属性設定	定画面を表	長示して設	定する。					
	● 帳票プレビ	2-							×
	▼ グリッド網			阁小 100%	1 拡大 前	ページ 次ペー	-ジ 1/1		ED局J
									<u>^</u>
				月 月	別平均気	,温			Î
		このデータ	は気象データ	<u>月</u> ベースとして	別平均気	<u>温</u> る過去の気候	: 情報から、 [~] #	L幌″、″東	
		このデータ 京 ^で 、 那覇	は気象データ 第の3都市につ	<u>月</u> ベースとして ついて月別平	<u>別平均気</u> 公開されてい 均気温を抜粋	<u> 二 温 る 過 去 の 気 候 し た 資 料 で す</u>	ff報から、	L幌″、″東	
		このデータ 京 [″] 、 [″] 那覇	は気象データ 第の3都市につ 気温(4.4.)	<u>月</u> ベースとして ついて月別平 ^{隣水量(礼幌)}	<u>別平均気</u> 公開されてい 均気温を抜粋	<u>温</u> る過去の気候 した資料です ^{降水量(東京)}	ff報から、 [《] 村 o 気温(那覇) β	L幌″、″東	
		このデータ 京 [″] 、 [″] 那覇 月 1月	は気象データ ずの3都市につ <u>気温(礼幌)</u> -4.3 で	<u>月</u> ペースとして ついて月別平 降水量(礼幌) 107.6 mm	<u>別平均気</u> 公開されてい 均気温を抜粋 <u>& & (東京)</u> 5.4 で	<u> る過去の気候</u> した資料です ^{内水量(東京)} 45.1 mm	情報から、"木 。 気温(那覇) P 16.3 で	L幌"、"東 等大量(邪覇) 113.0 mm	E
		このデータ 京 ^で 、 [*] 那覇 月 1月 2月	は気象データ ずの3都市につ <u>気温(礼嶋)</u> -4.3 で -3.7 で -3.7 で	月 ベースとして ついて月別平 107.6 mm 94.1 mm 94.1 mm	<u>別平均気</u> 公開されてい 均気温を抜粋 <u>& 4 で</u> <u>6.4 で</u> <u>6.8 で</u>	<u> </u>	た 情報から、 が た 。 、 、 、 、 、 、 、 、 、 、 、 、 、	L幌″、″東 \$水量(部額) 113.0 mm 106.0 mm	E
		このデータ 京、、「那覇 月 1月 2月 3月 4月	は気象データ ずの3都市につ <u>気温(礼嶋)</u> -4.3 で -3.7 で 0.0 で 6.6 で	月 ベースとして ついて月別平 107.6 mm 94.1 mm 81.8 mm 62.3 mm	<u>別平均気</u> 公開されてい 均気温を抜粋 <u>& 4 で</u> <u>6.4 で</u> <u>6.8 で</u> <u>8.7 で</u> 14.2 で	こ 二 る過去の気候 した資料です	た情報から、"木 。 気温(那颗) P 16.3 ℃ 16.4 ℃ 16.4 ℃ 21.2 ℃	L幌"、"東 +大量(振数) 113.0 mm 106.0 mm 162.0 mm 152.0 mm	E
		このデータ 京 ^{で、} 『那覇 月 1月 2月 3月 4月 5月	は気象データ デの3都市につ <u>気温(礼嶋)</u> -4.3で -3.7で 0.0で 6.6で 12.1で	ベースとして ついて月別平 107.6 mm 94.1 mm 81.8 mm 62.3 mm 54.8 mm	<u>別平均気</u> 公開されてい 均気温を抜粋 <u>& 4 で</u> <u>6.4 で</u> <u>6.8 で</u> 8.7 で 14.2 で 18.7 で		大学校 16.3 で 3.3 で 16.4 で 16.4 で 18.3 で 21.2 で 23.8 で	L幌"、"東 \$大量(那颗) 113.0 mm 106.0 mm 162.0 mm 152.0 mm 243.2 mm	E
		このデータ 京、、「那覇 月 1月 2月 3月 4月 5月 6月	は気象データ デの3都市につ <u>気温(礼嶋)</u> -4.3で -3.7で 0.0で 6.6で 12.1で 16.2で	ベースとして ついて月別平 107.6 mm 94.1 mm 81.8 mm 62.3 mm 54.8 mm 54.8 mm 66.4 mm	<u>別平均気</u> 公開されてい 均気温を抜粋 <u>気温(東京)</u> <u>6.4 ℃</u> <u>6.8 ℃</u> 8.7 ℃ 14.2 ℃ 14.2 ℃ 18.7 ℃ 21.7 ℃		大学校 16.3 で 3.3 で 16.3 で 16.4 で 18.3 で 21.2 で 23.8 で 26.4 で	L幌"、"東 4大量(那颗) 113.0 mm 106.0 mm 162.0 mm 243.2 mm 243.2 mm 252.7 mm	E
		このデータ 京、、『那覇 月 1月 2月 3月 4月 5月 6月 7月	は気象データ デの3都市につ	月 ベースとして ついて月別平 107.6 mm 94.1 mm 62.3 mm 62.3 mm 64.8 mm 66.4 mm 68.7 mm	<u>別平均気</u> 公開されてい 均気温を抜粋 <u>& 4 で</u> 5.4 で 5.8 で 8.7 で 14.2 で 14.2 で 21.7 で 22.5 3 で	このでは、 このででは、 このでは、 こ	大学校 16.3 で 16.3 で 16.4 で 16.4 で 21.2 で 23.8 で 26.4 で 28.4 で 28.5 で 29.5 で 29.5 で 29.5 で 29.5 で 20.5 で	L幌"、"東 4大量(部額) 113.0 m 106.0 m 162.0 m 162.0 m 243.2 m 252.7 m 195.0 m	E
		このデータ 京、、「那覇 月 1月 2月 3月 4月 5月 6月 7月 8月 9月	は気象データ ずの3都市につ く温(北線) -4.3で -3.7で 0.0で 6.6で 12.1で 16.2で 20.3で 21.7で 17.4で	メースとして ついて月別平 時水量(14,紀) 107.6 mm 94.1 mm 62.3 mm 62.3 mm 62.3 mm 64.8 mm 66.4 mm 66.7 mm 142.0 mm 137.7 mm	<u>別平均気</u> 公開されてい 均気温を抜粋 <u>5.4 で</u> 5.4 で 5.8 で 8.7 で 14.2 で 18.7 で 21.7 で 22.3 で 22.3 で 23.2 で	このでは、 この	大学校 16.3 で 3.3 で 16.3 で 16.4 で 16.4 で 21.2 で 23.8 で 26.4 で 28.4 で 28.4 で 27.3 で	Lter.", "p +x±(###)) 113.0 um 106.0 um 162.0 um 162.0 um 243.2 um 252.7 um 190.2 um 258.9 um 168.0 um	E
		このデータ 京 ^{「、「} 那覇 月 1月 2月 3月 4月 5月 6月 7月 8月 9月 10月	は気象データ ずの3都市につ く温(北線) -4.3で -3.7で 0.0で 6.6で 12.1で 16.2で 20.3で 21.7で 17.4で 11.0で	メースとして ウンマ月別平 降水量(14,42) 107.6 mm 94.1 mm 81.8 mm 62.3 mm 54.8 mm 66.4 mm 66.4 mm 142.0 mm 142.0 mm 142.7 mm 115.6 mm	分開されてい 均気温を抜粋 <u> 気温(東京)</u> 5.4 で 5.8 で 14.2 で 14.2 で 18.7 で 21.7 で 22.3 で 22.3 で 23.2 で 17.8 で	このでは、またでは、またでは、またでは、またでは、またでは、またでは、またでは、また	大学校会社(新歌)の時代でのでは、「本学校ののでは、「本学校ののでは、「本学校ののでは、」、「本学校のでのでは、「本学校のの」、「本学校ののでは、「本学校ののでは、「本学校ののでは、「本学校ののでは、「本学校のの」、「本学校ののでは、「本学校のの」、「本学校ののでは、「本学校のの」、「本学校ののでは、「本学校のの」、「本学校ののでは、「本学校のの」、「本学校ののでは、「本学校のの」、本学体の、本学体の、本学体の、本学体の、本学体の、本学体の、本学体の、本学体の	L.快."、"東 4大量(部語) 113.0 mm 106.0 mm 102.0 mm 162.0 mm 162.0 mm 243.2 mm 252.7 mm 190.2 mm 258.9 mm 168.0 mm 168.0 mm	E
		このデータ 京、、『那覇 月 1月 2月 3月 4月 5月 6月 7月 8月 9月 10月 11月	は気象データ デの3都市につ	メースとして ウンマ月別明平 降水量(14,42) 107.6 mm 94.1 mm 81.8 mm 62.3 mm 64.8 mm 64.8 mm 64.8 mm 64.7 mm 142.0 mm 143.0 mm	分開されてい 均気温を抜粋 <u> 気温(東京)</u> 5.4 で 5.8 で 8.7 で 14.2 で 14.2 で 18.7 で 21.7 で 22.3 で 23.2 で 17.8 で 12.8 で	 ・ ・ ・	大学校会社(新歌)のよいでは、「本学校会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社	L 快 "、" 東 4大量(部語) 113.0 m 106.0 m 162.0 m 162.0 m 162.0 m 162.0 m 162.0 m 162.0 m 162.0 m 162.0 m 163.0 m 164.0 m 100.2 m	E
		このデータ 京、、『那覇 月 1月 2月 3月 6月 7月 6月 7月 8月 9月 10月 11月 12月	は気象データ デの3都市につ	メースとして ウンマ月別男子 降水量(14,42) 107.6 mm 94.1 mm 81.8 mm 62.3 mm 54.8 mm 66.4 mm 66.4 mm 142.0 mm 142.0 mm 137.7 mm 115.6 mm 98.5 mm 100.1 mm	め の の の の の の の の の の の の の	 ・ 	大学校会社(新歌)のよいでは、「本学校会社の法律」では、「本学校会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社	Lter.", " p +x±(###) 113.0 m 106.0 m 106.0 m 162.0 m	E
		このデータ 京、、『那覇 月 1月 2月 3月 6月 7月 6月 7月 8月 9月 10月 11月 12月	は気象データ デの3都市につ	メースとして ついて月別サギ 107.6 mm 94.1 mm 81.8 mm 62.3 mm 54.8 mm 66.4 mm 142.0 mm 142.0 mm 142.0 mm 137.7 mm 115.6 mm 98.5 mm 100.1 mm	別平均気 公開されてい 均気温を抜粋 5.4 で 5.8 で 6.8 で 14.2 で 14.2 で 18.7 で 21.7 で 22.3 で 21.7 で 21.7 で 22.3 で 17.8 で 12.8 で 8.1 で	 ・ 	大学校会社(新歌)のよいでは、「本学校会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社会社	Lter.", " +x±(#100) 113.0 mm 106.0 mm 108.0 mm 162.0 mm 243.2 mm 252.7 mm 190.2 mm 258.9 mm 168.0 mm 168.0 mm 169.0 mm 169.0 mm 169.0 mm 123.0 mm	E
		このデータ 京 [*] 、 [*] 那覇 1月 2月 3月 4月 5月 6月 7月 8月 9月 10月 11月 12月	は気象データ のの3都市につ 、22(4線) -4.3で -4.3で -3.7で -3.7で 0.0で 6.6で 12.1で 12.1で 12.2で 20.3で 21.7で 17.4で 11.0で 4.5で -1.2で	月 ベースとして ついて月別平 時水量(札幌) 107.6 mm 94.1 mm 94	別平均気 公開されてい 均気温を抜粋 <u> 気温(東京)</u> 5.4 ℃ 5.8 ℃ 6.4 ℃ 5.8 ℃ 14.2 ℃ 14.2 ℃ 14.2 ℃ 14.7 ℃ 22.3 ℃ 27.1 ℃ 23.2 ℃ 17.8 ℃ 12.8 ℃ 8.1 ℃	このでは、またでは、またでは、またでは、またでは、またでは、またでは、またでは、また	第二(那歌)ののので、「おいの」のので、「おいの」ので、「おいの」ので、「おいの」ので、「おいの」で、「おいっの」で、「は、「は、「、」、「、」、「、」、「、」、「、」、「、」、「、」、「、」、「	上紀、、、、、 中 本大重(邪祭) 113.0 mm 106.0 mm 102.0 mm 162.0 mm 163.0 mm 173.0 mm	E
		このデータ 京 [*] 、 [*] 那覇 1月 2月 3月 4月 5月 6月 7月 8月 9月 10月 11月 12月	は気象データ の3都市につ 、22(北線) -4.3で -3.7で 0.0で 6.6で 12.1で 16.2で 20.3で 21.7で 17.4で 11.0で 4.5で -1.2で	月 ペースとして ついて月別平	別平均気 公開されてい 均気温を抜粋 <u> 気温で</u> 5.4 で 5.8 で 8.7 で 14.2 で 14.2 で 11.7 で 20.3 で 21.7 で 22.3 で 12.8 で 12.8 で 8.1 で 8.1 で	このでは、このでは、のでは、のでは、のでは、のでは、のでは、のでは、のでは、のでは、のでは、	第二(那動)の 第二(那動)の 第二(那動)の 16.3 ℃ 16.3 ℃ 16.4 ℃ 16.4 ℃ 18.3 ℃ 26.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.4 ℃ 28.5 ℃ 18.3 ℃ 18.3 ℃	Lter.", " *X±(##5) 113.0 mm 106.0 mm 162.0 mm 162.0 mm 162.0 mm 162.0 mm 162.0 mm 162.0 mm 162.0 mm 162.0 mm 163.0 mm 168.0 mm 168.0 mm 168.0 mm 169.0 mm 169.	
		このデータ 京 [「] 、 [「] 那覇 1月 2月 3月 4月 5月 6月 7月 8月 9月 10月 10月 11月 12月	は気象データ 7 の3都市につ 5 (14年) -4.3 で -3.7 で 0.0 で 12.1 で 16.2 で 20.3 で 21.7 で 11.4 で 11.0 で 4.5 で -1.2 で	月 ペースとして ついて月別平 時水量(札幌) 107.6 mm 94.1 mm 94.1 mm 66.3 mm 66.3 mm 142.0 mm 142.0 mm 142.0 mm 115.6 mm 115.6 mm 100.1 mm 100.1 mm	B) 平均気 CX開されてい 均気温を抜粋 S.4 で S.8 で S.7 で 14.2 で 25.3 で 27.7 で 25.3 で 3.7 で 25.3 で 3.7 で 27.7 で 25.3 で 3.7 で 27.7 で 25.3 で 3.7 で 4.1 で 5.1 で	このでは、またでは、またでは、またでは、またでは、またでは、またでは、またでは、また	第二(那新)のら、"木 の" (16.3 ℃ 16.3 ℃ 16.4 ℃ 18.3 ℃ 21.2 ℃ 226.4 ℃	Lter.", "p */***********************************	E B B U C S

4.4.2. 帳票印刷

帳票コンポーネントは印刷を行うために、以下の2つのメソッドを提供します。

①印刷 (printPaper() ※引数なし)

②印刷 (printPaper(boolean) ※引数あり)

メソッドの引数は、印刷時のパラメータ設定を行うためのダイアログを表示するかどうかを指定する もので、"printPaperO"は "printPaper(false)"と同じです。印刷パラメータ設定ダイアログでは、プ リンタの選択や拡大/縮小、印刷部数などの設定が可能であり、これを表示しない場合はデフォルトの プリンタから、デフォルトの設定で印刷されます。

▲ 印刷	x
一般(G) ページ設定(<u>S</u>) 外観(<u>A</u>)	
印刷サービス	
名前(<u>N</u>): Microsoft XPS Document Writer	 プロパティ(R)
状態: ジョブを受け付けています。	
タイプ:	
「青幸服」	🗌 ファイルに出力(E)
白泉車	白刷音降数
 すべて(L) 	印刷倍降数(<u>0</u>): 1 🚔
◎ ページ(E) 1 印刷範囲 1	□ 丁合い(<u>C</u>)
印刷取消し	

5. 複合コンポーネントの構築

5.1. 複合コンポーネント

複合コンポーネントとは、いくつかのコンポーネントをグループ化して1つのコンポーネントのよう に扱うもので、複数のコンポーネントから新たにコンポーネントを構築する仕組みです。具体的にはイ ベント伝播によって関係付けられた複数のコンポーネント群を新たに1つのコンポーネントとして切り 出し、その複合コンポーネントについて、外部に公開するメソッドや、発生させるイベントなどを定義 します。通常のプログラミングのモジュール分割や共通ライブラリ作成のような考え方を、コンポーネ ント指向的に実装したものです。

下図で示すとおり、複合コンポーネントでは内部構造と外部のインターフェイスを定義します。

1)コンポーネント構造

複合コンポーネント内のコンポーネント構造を管理します。通常のアプリケーション同様、コン ポーネント間はイベントによるメソッド起動によって接続されます。

2)外部インターフェイス

①外部公開メソッド

複合コンポーネントが外部に公開するメソッドを設定します。複合コンポーネント内にあるす べてのコンポーネントのメソッドを公開するとアプリケーション構築作業が非効率的になる ため、外部に公開するメソッドを選択する機能を提供します。また、メソッド名の重複を回避 するために、メソッド名の別名機能を提供します。

②発生イベント

複合コンポーネントから外部に伝播させるイベントを設定します。複合コンポーネント内にあるコンポーネントから発生するイベントのうち、外部に伝播させるイベントのみを選択します。 このとき、複数の内部コンポーネントから発生するイベントを識別させるために、イベント番 号を設定する機能を提供します。



複合コンポーネントはその用途によって、以下の2種類があります。

1)GUI 複合コンポーネント

複合コンポーネント自身が GUI コンポーネントとして、他のウィンドウやパネルに貼り付け可能なコンポーネントです。構築時に画面構成も設定します。

2)非 GUI 複合コンポーネント

複合コンポーネント自身は GUI コンポーネントではなく、他のウィンドウやパネルに貼り付け られないコンポーネントです。ただし、非 GUI 複合コンポーネントから別ウィンドウを表示す ることは可能です。
5.2. GUI 複合コンポーネントの構築

GUI 複合コンポーネントの構築は、以下の流れで行います。基本的な操作はアプリケーション構築と ほぼ同じですので、以降、操作の異なる操作(下図の網掛け部分)のみ説明します。それ以外の操作に ついては、前述の操作手順を参照してください。



任意のタイミングで作業状況を保存/再生



5.2.1. 構築作業の開始



5.2.2. 画面表示の動作確認

画面レイアウトの確認はアプリケーションビルダーの画面確認のウィンドウ上で行います。

画面	アプリケーションビルダー メイン画面
手順	①「画面確認」ボタンをクリックし、画面確認ダイアログを表示する
	WMZ Platform アブリケーションビルダー ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	ファイル 編集 アブリケーション オブション ヘルプ コメント行検索・・・
	コンポーネント名称 <no name=""></no>
	No Name>
	¥
	₩ 複合コンポーネントプレビュー
	親っシテナレイアウト設定「手動配置」 ■ 「再構画」 ― ――――――――――
	「プレビューーーーーーーーーーーーーーーーーーーーーーーーーー

5.2.3. 外部公開メソッドの設定

1)公開メソッドの選択



1)公開メソッドの選択 続き

	手順	②公開メソッド設定画面上で、公開するメソッドをもつコンポーネントからマウス右ク
		リックでメニューを表示し、[公開メソッド設定]を選択
		■ ケンフルコンホーネント [U:0] (KEY:"") ー 「テキストフィーサード 1000 (KEY:"")
		ボタン [[D:0-2] 公開メソッド設定
		□□ ファイル進伏[ID.0-3] (KEY. ファイル通(K3")
		閉じる
		┃ メソッド選択画面が表示される
		2015 メソッド選択
		□ 全メソッド対象
		メソッド一覧
		テキストの水平方向の自住置を設定する(int) テキストの水平方向配置を取得する()
		テキストを取得する()
		テキストを設定する(String) 入力フォーカフを設定する()
		入力モードを日本語にする()
		入力モードを英語にする() たか、無効を約つする()
		編集可否を取得する()
		編集可否を設定する(boolean) 非界界を認定する(coolean)
		目文巴で設定 9 る(COIDF)
		公開メソッド一覧
		設定 キャンセル
		▶□□へ↓ コンポーネントタトでダブルクロックオスコンにとはメソッド選択両面たキテキ+42
		コンホーホンドロエビアンルノリンフォロビビムタグノンド送扒回回で衣小させる ニレキ可能
		ここひり形。 扫動オスメソッドが主テキャたい提合 エのメソッドが北八明の乳ウにたっている
		に到するイノフトがな小C1Uない場口、てのイノフトが非公開の改正になつている。 "会ノリッド計算"のエーックギックフカ ON にナヤビ
		- エクノット対象 のテェック小ツクスを UNIに9 化は、
		コノハーイノトの王 public メノツトか衣示され、迭択可能となる。
		」 また、必要に応してメノットの情報設定(後述)を行えば常に表示されるようになる。
1		

手順 ③メソッド選択画面上で、公開するメソッドを"公開メソッド一覧"に追加する ("メソッド一覧"でメソッドを選択し、[追加] ボタン押下) × 🛂 メソッド選択 📃 全メソッド対象 メソッド一覧 テキストの水平方向の配置を設定する(int) テキストの水平方向配置を取得する() 入力フォーカスを設定する() 入力モードを日本語にする() 入力モードを英語にする() 有効・無効を設定する(boolean) 編集可否を取得する() 編集可否を設定する(boolean) 背景色を設定する(Color) 追加(公開化) 削除(非公開化) 公開メソッド一覧 テキストを取得する() テキストを設定する(String) 設定 キャンセル [設定] ボタンで反映される 🛂 公開メソッド設定 X ■ サンブルコンポーネント [ID:0] (KEY:"") ー■ テキストフィールド [ID:0-1] (KEY:"テキストフィールド1") ●テキストを取得する() テキストを設定する(String) ● サイスト シィールト [ID:0-1] (KEY: サイストクソーー・● getText() --> テキストを取得する()
 ● setText(String) --> テキストを設定する(String)
 ■ ボタン [ID:0-2] (KEY: "ボタン2")
 ■ ファイル選択 [ID:0-3] (KEY: "ファイル選択3") 閉じる

1)公開メソッドの選択 続き

2)公開メソッドの別名設定



3)公開メソッドの削除

画面	公開メソッド設定画面
手順	公開設定したメソッドからマウス右ボタンでメニューを表示し、
	[公開メソッド削除]を選択する
	・ フィールド[ID:0-1] (KEY:"テキストフィールド1") フィールドICテキストを設定する(String)
	←● getText()> テキストを取得する() ● setText(String)> フィール 「「「「」」 「「「」」 「「」」 「」」 「」」 「」」 「」」 「」」
	- ■ ボタン [D:0-2] (KEY:"ボタン2 - ■ ファイル選択 [D:0-3] (KEY:" - ■ ファイル選択 [D:0-3] (KEY:"
	開じる
	削除実行確認後、削除が反映される
	■ サンブルコンボーネント [ID:0] (KEY:"") テキストを取得する()
	ー ラテキストフィールド [ID:0-1] (KEY:"テキストフィールド1") └● getText()> テキストを取得する()
	── ボタン [ID:0-2] (KEY:"ボタン2") ── ファイル選択 [ID:0-3] (KEY:"ファイル選択3")

5.2.4. 外部公開イベントの設定

外部公開イベントは、内部で発生したイベントを複合コンポーネントのイベント外部伝播メソッドを 呼び出すことで設定します。設定方法は通常のコンポーネント間接続と同様です。

- イベント外部伝播メソッドには、2つの引数形式があります。
 - ①イベントを伝播させる (PFEvent) [notifyEvent(PFEvent)]

受け取ったイベントをそのまま外部に伝えます。

②イベント番号を指定してイベントを伝播させる(PFEvent, int) [notifyEvent(PFEvent, int)] 受け取ったイベントのイベント番号を第2引数に置き換えて外部に伝えます。このとき、イベン トオブジェクトは複製してから伝播しますので、このメソッド後に続くメソッド処理においては、 もとのイベント番号のままで処理が行われます。

画面	アプリケーショ	ョンビルダー メ	イン画面		
手順	通常の手順で、 "イベントをf ※引数の取行	伝播させたい 云播させる(PFF 导方法には"イ	イベントから lvent)"に接 ベント"を記	>複合コンポーネントの 続する 8定する)メソッド
		メソッド イベ	ントを伝播させる(PFEveni	● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	tậ,
	NO 型 0/PFEvent	説明 対象イベント	取得方法 イベント	コンボーネント -	メソッド/値
	 ● サンブルコンボーネント ● テキストフィールド ● テキストフィールド ● バール・パー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	· · · · · · · · · · · · · · · · · · ·	イベント伝 29ョンイベント - 9連携イベント	:播が設定される ^{単数Open用ファイル選択ダイアログを表示す} イベントを伝播させる	tる ファイル選択 D.0-3 KEY:"ファイル選択3" サンブルコンボーネント D.0- KEY:"

5.3. 非 GUI 複合コンポーネントの構築

非 GUI 複合コンポーネントの構築は、以下の流れで行います。操作は GUI 複合コンポーネント構築 と同じです。





5.4. 複合コンポーネントの利用

複合コンポーネントは、アプリケーションからコンポーネントとして使用可能です。以下の点で通常 のコンポーネントと異なります。

- 1)コンポーネント追加方法
 - 複合コンポーネントの追加は、メイン画面の背景メニューから三つの追加操作が可能です。
 - ①新規に複合コンポーネント(非 GUI)を追加する
 - [複合コンポーネント作成] [コンポーネント]を選択すると、通常のコンポーネントと同様に新規複合コンポーネントが追加されます。
 - ②新規に複合コンポーネント (GUI) を追加する
 - [複合コンポーネント作成] [GUI コンポーネント]を選択すると、通常のコンポーネント と同様に新規複合コンポーネントが追加されます。
 - ③既存の複合コンポーネントを追加する

[複合コンポーネント追加] - [ロード...]を選択したときに表示されるファイル選択画面またはファイル名一覧から、ファイルを選択します。

2) 複合コンポーネントの編集

アプリケーションに貼り付けた複合コンポーネントは、再編集が可能です。

- 3) 複合コンポーネントのメソッド
 - 他のコンポーネントからの接続を受けるメソッドは、各複合コンポーネントで設定されている外部 公開メソッドのみです。それ以外のメソッドは接続時に表示されません。また、別名が設定されて いる場合、別名が表示されます。
- 4) 複合コンポーネントから発生するイベント
 - 複合コンポーネントから発生するイベントは、各複合コンポーネントで設定されている外部公開イ ベントのみです。具体的には、複合コンポーネント内でイベント伝播メソッドによって外部伝播設 定されているイベントのみ発生します。

5) 複合コンポーネントの属性

複合コンポーネントの属性編集画面に表示される属性は、コンポーネント ID/コンポーネント Key の基本属性と、外部公開メソッドによって設定/取得可能なもの(setXxxx0/getXxxx0が公開され ているもの)のみです。外部で属性として扱いたい場合、外部公開メソッドとして設定しておく必要があります。

5.5. 複合コンポーネントの外部参照化

通常、複合コンポーネントはアプリケーション内に含まれてしまうため、複合コンポーネントをアプ リケーションと独立した一つのコンポーネントとして扱うことが困難です。それを回避するために、複 合コンポーネントの外部参照化、という機能を提供します。

5.5.1. 複合コンポーネント外部参照の考え方

アプリケーション内における複合コンポーネントの保存形式の形態として以下の2形態を提供し、その使い分けについては複合コンポーネントの属性を操作することによって利用者が設定可能です。

1)アプリケーションの一部として扱う

従来のプラットフォームの保存形式。複合コンポーネントを含むアプリケーションを保存した場 合、アプリケーションデータに内包される形で保存される。この場合、複合コンポーネントがア プリケーション内に取り込まれてしまうため、アプリケーションと複合コンポーネントを並行し て開発するといった独立性がなくなってしまう。ただし、複合コンポーネントがそのアプリケー ション固有のものであれば、この形式でも構わない。



2) 複合コンポーネントを独立したファイルで扱う

複合コンポーネントを含むアプリケーションを保存した場合、アプリケーションデータ内にはそ の複合コンポーネントが保存されているデータへの参照情報(ファイル名)が保存され、その実 体は保存されない。これによって、ある複合コンポーネントを複数のアプリケーションで使用す る場合に複合コンポーネントのみを変更すればその変更がすべてに反映される、といった共有の ための仕組みが実現できる。



5.5.2. 複合コンポーネントの外部参照ファイル

外部参照化された複合コンポーネントの保管場所は初期設定ファイル(Platform.ini)に設定するものとし、ロード時にはそのフォルダから複合コンポーネントをロードします。ただし、このフォルダへのデータファイル保管はアプリケーション保存時に自動で行われるわけではないため、対象となる複合コンポーネントを単独でこのフォルダに保存する操作が必要となります。

◇Platform.ini

CombinativeComponentsFolder=<外部参照複合コンポーネント保存先フォルダ名>

5.5.3. 外部参照設定方法

複合コンポーネントの外部参照化設定は、以下の手順で行います。

窟)
×
NULL
NULL
NULL

5.5.4. XML 出力機能におけるのパスワードロックと外部参照設定

パスワードロックされた複合コンポーネントは、XML 出力することで内部情報が容易に見えてしま うため、パスワードロックされた複合コンポーネントを含むアプリケーションについては、XML 出力 できないようにガードされています。

ただし、パスワードロックされている複合コンポーネントが外部参照設定されている場合は、内部情報が見られないため、XML 出力が可能となります。内部情報を隠すためにパスワードロックしたアプリケーションの XML 出力には、外部参照設定機能を利用してください。

5.5.5. 外部参照化されたデータファイル名

外部参照化された複合コンポーネントのデータファイル保管場所は一つであるため、データファイル 名はできるだけ他と重複しない識別しやすいものとしてください。また、複合コンポーネントの外部参 照名を設定する際、存在しないファイルや間違ったものを指定するといった不注意があると、アプリケ ーションデータが正しく復元できなくなってしまいます。特に、外部参照先が自分自身になっているな どといった参照のループが設定されてしまった場合、ロード処理が終わらなくなってしまいます。設定 時にファイル名をよく確認するとともに、間違えにくいファイル名にするようにご注意ください。

6. Web アプリケーションの構築

6.1. Web アプリケーション

Web アプリケーションとは、インターネット等のネットワークを介して Web ブラウザ上で実行する アプリケーションソフトウェアです。MZ Platform で作成したアプリケーションは、基本的にはコンピ ュータ単体での運用またはセキュリティが確保された一事業所内でのデータベース等を介した運用を 前提としておりました。MZ Platform 3.0 より Web アプリケーション構築機能が追加され、Web ブラ ウザを用いて多地点からアクセス可能なアプリケーションを構築することが可能となりました。

MZ Platform における Web アプリケーション実行環境の主な構成要素は MZ サーブレット、MZ API、 MZ アプリケーション (アプリケーションビルダーで作成した Web アプリケーション)です。基本的な 動作は次のようになっております。



- Web ブラウザからの指定アドレスへのアクセスに対し、MZ アプリケーションから画面情報を生成し、Web ブラウザにアプリケーション画面を表示する
- ② Web ブラウザに表示されたアプリケーション画面に向かって、ユーザが操作を行う
- ③ MZ サーブレットがブラウザからのリクエストを受け取り、MZ API を呼び出す
- ④ MZ API を利用して MZ アプリケーションを外部から起動し、指定された処理を行う
- ⑤ MZ アプリケーションの処理結果を、MZ サーブレットが MZ API 経由で取得し、Web ブラウザ 画面に反映する

以下では、アプリケーションビルダーを用いた MZ アプリケーション(以下では、Web アプリケーションと呼ぶ)の構築法について述べます。

6.2. Web アプリケーションの構築

Web アプリケーションの構築は、以下の流れで行います。基本的な操作はアプリケーション構築とほ ぼ同じですので、以降、操作の異なる操作(下図の網掛け部分)のみ説明します。それ以外の操作につ いては、前述の操作手順を参照してください。



任意のタイミングで作業状況を保存/再生



6.2.1. 構築作業の開始



6.2.2. コンポーネントの追加

コンポーネントの追加手順は通常のアプリケーションと同様です。すべてのコンポーネントが追加可 能ですが、Webアプリケーションとして画面表示が可能なコンポーネントは次のコンポーネントに限ら れています。ただし、画面表示ができないコンポーネントについても処理を行う部品として使用するこ とは可能です。

- ・ラベル
- ・ボタン
- ・テキストフィールド
- ・パスワード入力フィールド
- ・数値入力フィールド
- ・日付入力フィールド
- ・テキストエリア
- ・コンボボックス
- ・チェックボックス
- ・ラジオボタングループ
- ・テーブル
- ・リスト
- ・イメージビューワ
- ・各種グラフ
- ・パネル
- ・メッセージダイアログ
- ・確認ダイアログ
- ・GUI 複合コンポーネント
- ・Web 画面(Web アプリケーション専用、複数画面アプリケーション構築用)
- ・ファイルアップロード(Webアプリケーション専用)
- ・ファイルダウンロードボタン(Webアプリケーション専用)
- ・ファイルダウンロードリンク(Webアプリケーション専用)
- ・ハイパーリンクボタン(Webアプリケーション専用)
- ・ハイパーリンクテキスト(Webアプリケーション専用)

6.2.3. コンポーネント間の接続

コンポーネント間の接続手順は通常のアプリケーションと同様ですが、Webアプリケーションとして 発生するイベントがコンポーネントの種類に応じて次のように限定されています。発生しないイベント はグレイで表示されます。また、使用可能なメソッドについても限定されており、使用不可能なメソッ ドは自動的に「起動しない」が設定され、表示がグレイになり、実行されません。

・ボタン :アクションイベント :アクションイベント、データ設定イベント ・テキストフィールド ・パスワード入力フィールド :アクションイベント、データ設定イベント ・数値入力フィールド :アクションイベント、データ設定イベント :アクションイベント、データ設定イベント ・日付入力フィールド ・テキストエリア :データ設定イベント ・コンボボックス :データ選択イベント ・チェックボックス :データ更新イベント ・ラジオボタングループ :データ選択イベント ・Webアプリケーション :アプリケーション終了イベント

₩ 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 🔻
Webアプリケーション名称			(in
Webアブリケーション KEY:"" デキストフィールド レ1. KEY:"** 確認タイアログ D:2 KEY:"確認タイアログ2"	799904401 799904401	OK・キャンセルボタン付きダイアログを書 テキストを設定する 入力フォーカスを設定する	にする 「端記ダイアログ D.12 KFY: 「端記ダイアログ2 「テキストフィールド P.14 (NO.11) レ:4 (KY: デキストフィールド4* 変動しない) に、デーラストフィールド4*
□ デキストフィールド ID:4: (KEY): デキストフィールド4* □ ボタン ID:3: KEY: ボタン3*	70932442F	OK・キャンセルボタン付きダイアログをま OK・キャンセルボタン付きダイアログをま	「確認ダイアログ D:2 (株式): 確認ダイアログ (株式): 確認ダイアログ (株式): 確認ダイアログ (株式): 確認ダイアログ (株式): 確認ダイアログ (株式): 確認ダイアログ
「アイコン月後川 ・ ■ マゴリケーション ・ 三 浦	面積成部品。 副内マンドウ	・ マーコー ・ パンズ・ルン・ のい時のに見、	11年今(病毒) 「11年~ト
【アイコン凡例】 ■ アブリケーション ■ ●	▲構成部品 ■ ウィンドウ	メニューバネル処理部品 _	▶ 複合(画面構成) ■ 複合(処理) ■ リモート
画面確認	集 帳票編集 ロー	・ド 挿入 保存 上書き保存	クリア 終了

6.2.4. 画面レイアウト設定

通常のアプリケーションと同様に、画面レイアウトの設定についてはアプリケーションビルダーの画 面編集のウィンドウ上で行います。コンポーネントの配置については、Web アプリケーションとして表 示可能なコンポーネントのみ配置可能です。

6.2.5. 画面表示の動作確認

画面レイアウトの確認はアプリケーションビルダーの画面確認のウィンドウ上で行います。



6.2.6. Web アプリケーション動作確認

サーバにおける MZ サーブレットの指定フォルダに、作成したアプリケーションファイルを配置します。Web ブラウザで該当 URL にアクセスし、動作を確認します。

MZ サーブレットの動作とアプリケーションファイルの配置に関しては、配布物に含まれている MZ Platform Servlet フォルダ内の ReadMe.txt をご覧ください。

7. リモートアプリケーションとの連携

7.1. データ連携機能

データ連携とは、複数のプラットフォームが互いに連携して処理を行う機能であり、具体的には以下 の2つの機能があります。

[コンポーネント連携] 他のプラットフォーム上のコンポーネントを遠隔利用する [コンポーネント転送] 他のプラットフォーム上のコンポーネントを転送

コンポーネント連携は、他のプラットフォーム上にあるコンポーネントを、あたかもローカルのプラ ットフォーム上にあるかのように扱える機能です。アプリケーション上は、他のプラットフォーム上に 存在するコンポーネントをリモートコンポーネントとして扱います。

連携先のコンポーネントの処理結果はリモートコンポーネントに返され、その情報(処理の終了ステ イタスやデータなど)はリモートコンポーネントから発生するコンポーネント連携処理結果通知イベン トによって取得可能です。



コンポーネント転送は、通常のコンポーネントと同様にコンポーネントを追加し、メソッドの呼び出 しによって処理を設定します。

<注意事項>

データ連携機能を利用するには、そのための設定が必要です。詳細は、"データ連携導入手順書.pdf" をご覧ください。 8. アプリケーションの実行(アプリケーションローダー)

構築されたアプリケーションは、アプリケーションローダーから実行します。アプリケーションの実 行はアプリケーションビルダーからもできますが、アプリケーションを変更する必要がない場合、また は変更させたくない場合、アプリケーションローダーを使用して実行します。

スタートメニューからアプリケーションローダーを起動します。

[スタート] -[(すべての) プログラム]-[MZ Platform 3.5]-[アプリケーションローダー]

また、実行中にコンソールを表示させたい場合は、"アプリケーションローダー (コンソール)"を実行します。下のようなファイル選択画面が表示されますので、アプリケーションを選択します。

── 開く			x
参照:	🐌 AP_DATA	- 👔 🔊	
ようしていた。 最近使った項 目 デスクトップ マイドキュ メント コンピュー	Datacooperat MetalMold MZChecker Production Sample SoftGenerator Tutorial	ion	
	ファイル名:		開く
ネットワーク	ファイルタイプ:	「すべての読込可能なアプリケーションファイル (.mzax, .mzas, .xml, .apl) ▼	取消し

アプリケーションローダーの実体は、"導入フォルダ¥3.5¥PFLoader.exe"です。このプログラムに は実行時引数が指定でき、アプリケーションファイル名を指定することで、ファイル選択の操作を省略 することができます。

◇引数なし

ファイル選択ダイアログが表示され、そこで指定されたファイルをロードして起動します。 ◇引数あり

引数で指定したファイルからアプリケーション情報をロードして、起動します。

実行形式: PFLoader <アプリケーションファイル名>

9. コンポーネント情報の編集

アプリケーション構築時の操作支援として、アプリケーションビルダー上にコンポーネントのメソッド情報やイベント情報が表示されます。コンポーネント情報は、XML テキストファイル形式で実行環境内に提供されます。この情報は利用環境で変更が可能ですので、必要に応じて編集してください。

コンポーネント情報の編集操作は、アプリケーションビルダーから起動されるコンポーネント情報編 集ツールによって行います。

画面	アプリケーションビルダー メイン画面	
手順	 ①メニューバーから [オプション] ー [コンポーネント情報編集] を選択 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 	
	アブリケーション名称 コンボーネント情報編集 アブリケーション ● アブリケーション Look&Feel ロケール ロケール ログール ログール ログール コンボーネント説明ツールチップ表示 ●理部品 マ コンボーネント説明ツールチップ表示 人出力 マ コンボーネントコピー接続先確認 メモリ使用量表示… メモリ健理(ガベージコレクタ実行) マンボーネントの選択により、編集画面が表示される	

9.1. メソッド情報の設定

アプリケーション構築時に表示されるコンポーネントのメソッド情報は、すべてこのメソッド情報設 定にて登録された情報です。不要なメソッドを表示させないようにしたり、メソッドの処理内容を設定 したりすることで、利用者独自の実行環境構築が可能です。

9.1.1. メソッドの公開設定

①ひてにたいす	いいドの "ひ		エェックボックファエー	いカたしかて
①公開したいメ	フットの 公	用9 る	テェックホックスにナェ	ックを入れる
→ 該当メソッ	, ド行が黄色て	で表示され	13	
🛛 🖳 コンポーネント情	報編集			
	コンポーネント名称	テキストフィー	ルド	
	コンポーネント概要	1行小立字列表	みわすることが可能なつ?」ポーマ?」と	
	コノホーネノト恢要	「17の文子列を	ハノ」することが可能なコンホーネント	
	コンポーネントクラス	jp.go.aist.dmrc.	platform.beans.gui.text.PFTextField	
マメソッド				
メソッド			別名	公開
setSelectionColor(Color)			🔲 公開する
setSelectionEnd(int)			📃 公開する
setSelectionStart(in	t)			📃 公開する
setSize(Dimension)			🔲 公開する
setSize(int,int)				📃 公開する
setText(String)			テキストを設定する	✓ 公開する
setToolTipText(Strin	g)			
setTransferHandler	(TransferHandler)			□ 公開する
setUI(TextUI)				<u></u> 公開する
setVerifyInputWhen	-ocusTarget(boolean)			<u> </u>] 公開する
setVisible(boolean)				□公開する
show()				□公開する
show(boolean)				□公開する
d of the				
イベント			NO イベント発生	
アクションイベント			0Enterキーが押されたとき	
フォーカスイベント			0フォーカスの消失	
フォーガスイベント			1 フォーガスの取得	
	1			
2 メソッドの表 (カリ たがゴ	▶ 示文字列を"	〔 別名"セ スニレズ	 0マウスクリック佐ボタン以外) 設定 ++ンセル ルに設定する カコキ() 	
②メソッドの表 (セルをダブ	ト 示文字列を" ルクリックす	「 別名"セ ることで	 	
②メソッドの表 (セルをダブ	[▶] 示文字列を" ルクリックす	〔 別名"セ ることで	 0マウスクリック佐ボタン以外) 酸定 ++ンセル ルに設定する 入力可能) 	
〕メソッドの表 (セルをダブ	ト 示文字列を" ルクリックす ^{報編集} ⊐ンポーキント名称	[別名"セ ることで ^{[テキストフィ〜)}	 0マウスクリック佐ボタン以外) 設定 ++ンセル ルに設定する 入力可能) ルド 	
〕メソッドの表 (セルをダブ	ト 示文字列を" ルクリックす ^{戦編集} コンポーネント格率 コンポーネント概要	別名"セ ることで ^{テキストフィー,} 1行の文字列を	 	
②メソッドの表 (セルをダブ	ト 示文字列を" ルクリックす ^{鍛編集} □ンポーネント名称 □ンポーネント概要	別名"セ ることで ^{テキストフィー、} 1行の文字列を、	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	
②メソッドの表 (セルをダブ	ト 示文字列を" ルクリックす ^{縦編集} コンポーネント名称 コンポーネント病要 コンポーネントの要	別名"セ ることで ^{テキストフィー、} 1行の文字列を、 (pgo.aist.dmrc.)		
2)メソッドの表 (セルをダブ ^{(メリッド}	ト 示文字列を" ルクリックす ^{総編集} コンポーネント名称 コンポーネント概要 コンポーネントの要	別名"セ ることで ^{テキストフィー、} 1行の文字列を、 [pgo.aist.dmrc.	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	
メソッドの表 (セルをダブ) メソッド メソッド メソッド	ト 示文字列を" ルクリックす ^{戦編集} コンポーネント名称 コンポーネント概要 コンポーネントクラス	別名"セ ることで ^{テキストフィー、} 1行の文字列を、 [pg0.aist.dmrc.]	 ・マウスクリック(左ボタン以外) ・ ・ ・	
②メソッドの表 (セルをダブ) ^{図 コンポーネント#} ^{メソッド} ^{メソッド} ^{メソッド} ^{setSelectionColor((}	ト 示文字列を" ルクリックす ^(戦編集) コンボーネント名称 コンボーネント概要 コンボーネントの要 コンボーネントの要	別名"セ ることで ^{テキストフィー,} 1行の文字列を、 p.go.aist.dmrc.	 ロマウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド スカすることが可能なコンボーネント platform.beans.guitext.PFTextField 別名 	
)メソッドの表 (セルをダブ ジ ^{ンボーネント#} ^{メソッド} setSelectionColor(setSelectionEnd(im	ト 示文字列を" ルクリックす 	別名"セ ることで ^{テキストフィー,} 1行の文字列を、 p.go.aistdmrc.	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド 入力することが可能なコンボーネント platform.beans.gui.text.PFTextField 別名	
)メソッドの表 (セルをダブ) ^{メソッド} ^{メソッド} ^{メソッド} ^{setSelectionColor(^{setSelectionEnd(int)} ^{setSelectionStart(int)}}	ト 示文字列を " ルクリックす ¹ ¹⁰ ¹⁰ ¹⁰ ¹⁰ ¹⁰ ¹⁰ ¹⁰ ¹⁰ ¹⁰	別名"セ ることで ^{テキストフィー} 、 1行の文字列を、 ()p.go.aist.dmrc,	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド 入力することが可能なコンボーネント platform.beans.gui.text.PFTextField	
)メソッドの表 (セルをダブ) ジョンボーネント メソッド setSelectonColor(setSelectonStat(in setSelectonStat(in setSize(Dimension	ト 示文字列を " ルクリックす ^{報編集} コンポーネント名称 コンポーネントクラス Color))	 別名"セ ることで 	・マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ・レド スカすることが可能なコンボーネント platform.beans.guitext.PFTextField 別名	
シメソッドの表 (セルをダブ) メソッド メソッド メソッド setSelectionEnd(int setSelectionStart(int setSize(Dimensiont setSize(int, int)	ト 示文字列を" ルクリックす ^{報編集} コンポーネント橋栗 コンポーネント/橋栗 コンポーネント/荷要 コンポーネント/クラス Solor)	別名"セ ることで ^{テキストフィー,} 1行の文字列を、 1行の文字列を、 1行の文字列を、 1	0マウスクリック(左ボタン以外) 酸定 キャンセル ルに設定する 入力可能) ルド 入力することが可能なコンボーネント platform.beans.guitext.PFTextField 別名	 公開 公開する 公開する 公開する 公開する 公開する 公開する 公開する
)メソッドの表 (セルをダブ) 	ト 示文字列を" ルクリックす ^{報編集} コンポーネント想要 コンポーネントクラス Solor))	別名"セ ることで ^{テキストフィー,} 1行の文字列を、 (p.go.aist.dmrc,)	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド 入力することが可能なコンボーネント platform.beans.guitext.PFTextField 別名 テキストを設定する	
>メソッドの表 (セルをダブ) メソッド メソッド メソッド setSelectionColor((setSelectionStart(in setSize(Dimension setSize(intint) setSize(intint) setText(String) setToolTipText(String)	ト 示文字列を" ルクリックす ^{総編集} コンポーネント名称 コンポーネントの要 コンポーネントクラス Color)) Color))	 別名"セ ることで ^{テキストフィー,} 1行の文字列を、 [p.go.aist.dmrc.	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	公開 公開する
②メソッドの表 (セルをダブ) メソッド メソッド メソッド setSelectionColor(setSelectionStar(in setSize(Dimension setSize(Dimension setSize(Dimension setSize(Dimension setSize(Dimension setSize(Dimension setSize(Dimension setTransferHandler = attransferHandler	ト 示文字列を" ルクリックす ^{翻編集} コンポーネントの称 コンポーネントの要 コンポーネントクラス Color)) (o) (TransferHandler)	別名"セ ることで ^{テキストフィー、} 1行の文字列を、 [pg0.aist.dmrc.]	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド スカすることが可能なコンボーネント platform.beans.guit.text.PFTextField アキストを説定する	公開する 公 公 () () () () () () () () () () () () () (
②メソッドの表 (セルをダブ) メソッド メソッド setSelectionEnd(int setSelectionEnd(int) setSelectio	ト 示文字列を" ルクリックす ^{観編集} コンボーネント名称 コンボーネントの概要 コンボーネントクラス Color)) () (TransferHandler)	別名"セ ることで ^{テキストフィー,} 1行の文字列を [] p go.aist.dmrc	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	公開 公開する ○公開する ○ ○ ○ ○ ○ ○
②メソッドの表 (セルをダブ)	ト 示文字列を " ルクリックす 部編集 コンポーネント名称 コンポーネントの要 コンポーネントクラス Color)) (IransferHandler) FocusTarget(boolean)	別名"セ ることで ^{テキストフィー,} 1行の文字列を () jp.go.aist.dmrc.	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	
②メソッドの表 (セルをダブ) ②メソッドの表 (セルをダブ)	ト 示文字列を " ルクリックす ¹ ¹⁰ 	 別名"セ ることで テキストフィー、 1行の文字列を、 p.go.aist.dmrc. 	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	
②メソッドの表 (セルをダブ) ②メソッドの表 (セルをダブ) ジョンポーネンド SetSelectonColor(setSelectonColor(setSelectonStat(in setSelectonStat(in setSelectonStat(in setSze(Dimension set	ト 示文字列を " ルクリックす ¹ ¹	[別名"セ ることで ^{テキストフィー,} 1行の文字列を、 ()p go.aist.dmrc.	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド 入力することが可能なコンボーネント plaform.beans.guitext.PFTextField 男り名 テキストを設定する	
②メソッドの表 (セルをダブ)	ト 示文字列を" ルクリックす ^{報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラス Color)) (TransferHandler) FocusTarget(boolean)	「 別名"セ ることで ^{テキストフィー、} 1行の文字列を、 [p.go.aist.dmrc.]	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド 入力することが可能なコンボーネント platform.beans.guitext.PFTextField 5+4ストを設定する ワームストを設定する	公開 公開する ○公開する ○公開する ○公開する ○公開する ○公開する
②メソッドの表 (セルをダブ)	ト 示文字列を" ルクリックす ^{酸編集} コンポーネント名称 コンポーネント想要 コンポーネントクラス Color)) () (TransferHandler) =ocusTarget(boolean)	別名"セ ることで ^{テキストフィー、} 1行の文字列を、 [pgo.aistdmrc.]	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド スカすることが可能なコンボーネント platform.beans.guitext.PFTextField アキストを設定する ワークストンを設定する	公開 公開する 公開する
②メソッドの表 (セルをダブ)	ト 示文字列を " ルクリックす ^{観編集} コンポーネント名称 コンポーネントの概要 コンポーネントクラス Color)) () (TransferHandler) =ocusTarget(boolean)	別名"セ ることで ^{テキストフィー、} 1行の文字列を、 [pgo.aistdmrc]	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド スカすることが可能なコンボーネント platform.beans.guitext.PFTextField アキストを設定する デキストを設定する	公開 公開する 公開する
2)メソッドの表 (セルをダブ) メソッド メソッド メソッド setSelectionColor((setSelectionEnd(in setSize(intin) setSize(intin) setToolTipText(Strir setTertatString) setToolTipText(Strir setTert(String) setToolTipText(Strir setTert(String) setVertfyinputWhen setVisible(boolean) show() show(boolean) eixa/ イベント イベント	ト 示文字列を " ルクリックす 副編集 コンポーネント名称 コンポーネントの概要 コンポーネントのの (o) (TransferHandler) = Coloration (TransferHandler)	[別名"セ ることで ^{テキストフィー,} 1行の文字列を () jp.go.aistdmrc,	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド設定する 入力可能) ルド 入力することが可能なコンボーネント platform beans guitext PFTextField 別比名 デキストを設定する ウーマント発生	
A ソッドの表 (セルをダブ メソッド メソッド メソッド setSelectionColor(setSelectionEnd(ini setSize(initi) setSize(initi) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setSize(initini) setVisible(boolean) siza(1	ト 示文字列を " ルクリックす ¹ ¹	別名"セ ることで ^{テキストフィー,} 1行の文字列を、 p.go.aistdmrc.	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド 入力することが可能なコンボーネント platform.beans.guitext.PFTextField アキストを設定する デキストを設定する 0 イペント発生 0Enterキーが押されたとき	
	ト 示文字列を " ルクリックす ¹ ¹	別名"セ ることで ^{テキストフィー,} 1行の文字列を、 ip go.aistdmrc.	 ●マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルに設定する ルド スカすることが可能なコンボーネント platform.beans.guitext.PFTextField 第川名 デキストを設定する デキストを設定する ハ0 イベント発生 0 マオーカスの消失 マオーカスの消失 	
2)メソッドの表 (セルをダブ) メリッド メリッド メリッド setSelectionColor(setSelectionColor(setSelectionStart(in setSize(Dimension setSize(inLint) setTransferHandler setUr(TextU) setVerffyInputWhen setVisible(boolean) show() show(boolean) eiza∆ イペント アクションイペント フォーカスイペント フォーカスイペント	ト 示文字列を " ルクリックす ¹ ^{叙編集} コンポーネント名称 コンポーネントの要 コンポーネントクラス Color)) t) cousTarget(boolean) = = - - - - - - - - - - - - -	[別名"セ ることで ¹ 行の文字列を、 1 (p.go.aist.dmrc.)	0マウスクリック(左ボタン以外) 設定 キャンセル ルに設定する 入力可能) ルド設定する 入力可能) ルド スカすることが可能なコンボーネント platform.beans.guitext.PFTextField デキストを設定する デキストを設定する ロビー レー ロビー ロー ロー	 公開する

画面	コンポーネント情報編集画面
手順	①該当メソッドのメソッド名をマウスで左クリック
	→ メソッド情報編集画面が表示される
	メソッド情報編集 ×
	メソッド名 setText(String)
	メソッド引数
	NO データ型 説明
	0 String テキスト
	(2)谷引数に対する記明文字列を"記明"セルに設定する
	(セルをダブルグリックすることで人力可能)
	🔀 メソッド情報編集
	メソッド名 setText(String)
	Color N R Z 陸h
	NO データ型 説明
	0 String テキスト
	③右上の[×]ホタンで終了(確定)

9.1.2. メソッド引数の設定

9.2. イベント情報の設定

アプリケーション構築時に表示されるイベント情報は、すべてこのイベント情報設定にて登録された 情報です。イベント番号やイベント内包データの説明が設定可能です。

9.2.1.	イベン	ト番号の設定
--------	-----	--------

①設守オス	イベント来旦た	"NO" +1	に入力する	
	1 ハノト留方を、	NU UN		
(セルをダフ	ブルクリックする	ことで入力	可能)	
🏧 コンポージ	ネント情報編集			
	コンポーネント名称	色選択		
	4940 4091468			
	コンポーネント概要	色を選択させるダイア	ログを表示するコンポーネント	
	コンボーネントクラス	र jp.go.aist.dmrc.platfor	m.beans.gui.chooser.PFColorChooser	
-メソッド―				
States R			別名	八明
addPEData	Selectl istener/PEDataSelectl istens	er)	<i>7</i> 720	「別」
equals(Obis	act)			□ ム則する
equals(Obje	500)			□ ム用する
getColor()				□ ム用する
getCompon	entID()			「公開する」
getCompon	entKev()			「公開する」
getCompon	entKey()			「公開する」
getCompon	entName()			「公開する」
getCompon	entPublicName()			二 公開する
getCompon	entString()			□ ム(用)する □ 小開する
netPED.ataS	electListenerList()			<u> </u>
hashCode())			□ ム(用)する □ 小開すス
isAllowPullT	, Fransfer()			
is Allow Push	Transfer()			
ーイベントー				
Correct b		NO	イベント発生	
-1 -1 -1 2210	/ -=-> - I	NO		
) = 3 進が				
2イベントの	の発生トリガーの	 説明を、"イ	」 ++>セル ベント発生"セルに入	カする
②イベント0 (セルをダご	D発生トリガーの ブルクリックする	↓ 説明を、"ィ ことで入力 [・]	」 ^{++ンセル} 「ベント発生"セルに入 可能)	カする
②イベント0 (セルをダご	D発生トリガーの ブルクリックする ^{ネント情報編集}	⇒ 説明を、"ィ ことで入力	」 ++ンセル 「ベント発生"セルに入 可能)	カする
②イベント0 (セルをダご	D発生トリガーの ブルクリックする ^{ネンド情報編集} コンポーネント名称	説明を、"イ ことで入力 ^{色潮沢}	 「ベント発生"セルに入 可能)	カする
②イベント0 (セルをダご ₩□>ポージ	D発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネント編集	議定 説明を、"イ ことで入力 ^{色選択}	 「ベント発生"セルに入 可能)	カする
②イベント0 (セルをダご ₩□>ポージ	D発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネント概要	 説明を、"イ ことで入力」 色選択 	<u>キ+ンセル</u> ベント発生"セルに入 可能)	カする
2)イベント0 (セルをダご) (セルをダご)	D発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラフ	設 説明を、"イ ことで入力 色 選択 色を選択させるダイア 、 jp.go.aist.dmrc.platfor	キャンセル ベント発生"セルに入 可能) ログを表示するコンポーネント m.beans.gui.chooser.PFColorChooser	カする
2)イベント0 (セルをダご) コンポー ×ソッド 	D発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネント概要 コンポーネントクラフ	設 説明を、"イ ことで入力" 色選択 色を選択させるダイア ス [p.go.aist.dmrc.platfor	キャンセル ベント発生"セルに入 可能) ログを表示するコンボーネント m. beans gui chooser .PFColorChooser	力する
)イベント0 (セルをダご) メンッド メンッド オンッド	D発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネント初要 コンポーネントクラフ Select istener(PFDataSelect istener	設定 説明を、"イ ことで入力" 色選択 色を選択させるダイア ス jp.go.aist.dmrc.platfor	キャンセル デベント発生"セルに入 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser	力する _
)イベント0 (セルをダご) メワッド メワッド guids(Ohie equals(Ohie	D 発生トリガーの ブルクリックする ^{オント情報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListene	談明を、"イ ことで入力" 色変源なせるダイア ア Pigo aist dmrc platfor er)	キャンセル ベント発生"セルに入 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名	力する
0イベント0 (セルをダご) メワッド メワッド addPFData3 equals(Obje oelClass()	D 発生トリガーの ブルクリックする ^{x ント情報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListene act)	議定 説明を、"イ ことで入力」 色溜沢 色を溜択させるダイア マ 、 p go.aist.dmrc.platfor er)	キャンセル ボント発生"セルに入 可能)	力する 公開 公開する 公開する 公開する 公開する の公開する
Oイベントの (セルをダご) メソッド addPFData equals(Ob)(petCalor() PetCalor()	D 発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネント将要 コンポーネントクラス SelectListener(PFDataSelectListene	設定 説明を、"イ ことで入力 色選択 色変獄ださせるダイア マ jp.go.aist.dmrc.platfor er)	キャンセル ベント発生"セルに入 可能) ログを表示するコンポーネント m.beans.gui.chooser.PFColorChooser	カする 公開 公開する 公開する 公開する の公開する の分開する
)イベントの (セルをダご) メワッド メワッド メワッド addPFData equals(Obje getCalsor) getColor() getColor()	D 発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネント概要 コンポーネントの要 コンポーネントクラス SelectListener(PFDataSelectListene ect)	設定 説明を、"イ ことで入力 色選択 色を選択させるダイア マ p.go.aist.dmrc.platfor	キャンセル 「ベント発生"セルに入 可能) ログを表示するコンポーネント m.beans.gui.chooser.PFColorChooser	カする 公開 公開する 公開する 公開する 公開する 公開する 公開する 公開する の 公開する
Oイベントの (セルをダご) メワッド メワッド メワッド メワッド タワッド タワッド ロース マース マース マース マース マース マース マース マ	D発生トリガーの ブルクリックする ^{オント情報編集} コンポーネント名称 コンポーネント概要 コンポーネントの要 コンポーネントクラス SelectListener(PFDataSelectListene act)	設定 説明を、"イ ことで入力 色選択 色を選択させるダイア マ p.go.aist.dmrc.platfor er)	★+ンセル ボベント発生"セルに入 可能) □グを表示するコンポーネント m. beans gui chooser PFColorChooser 別名	カする よ間 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する の 公開する の の の の の の の の の の の の の
Aベントの (セルをダご) ジョンボー メソッド メソッド addPFData equals(Obj getClass() getCompon getCompon getCompon getCompon	D 発生トリガーの ブルクリックする ^{オント情報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListene cd)	説明を、"イ ことで入力 色選択 色を選択させるダイア マ マ)	キャンセル ボベント発生"セルに入っ 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名	カする 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する (公開する) (公開する) (公開する) (公開する)
Aベントの (セルをダご) ジョンボー メソッド メソッド addPFData getCalso() getCompon getCompon getCompon getCompon	D 発生トリガーの ブルクリックする ^x ンド情報編集 コンポーネンド名称 コンポーネンドの要 コンポーネンドクラフ SelectListener(PFDataSelectListene act) entID() entKey() entKey() entKey()	談明を、"イ ことで入力" 色灌釈 色を灌釈させるダイア マ マ)	キャンセル ボント発生"セルに入っ 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名	カする 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する (公開する)
Aベントの (セルをダご) メワッド メワッド addPFData equals(asb) getCompon getCompon getCompon getCompon getCompon	D 発生トリガーの ブルクリックする ^{オント情報編集} コンポーネント名称 コンポーネント内要 コンポーネントクラフ SelectListener(PFDataSelectListene cct) enttD() ientKey() ientKey() ientKeys() ienttKey()	議定 説明を、"イ ことで入力 色選択 色変獄Rさせるダイア マ jp.go.aist.dmrc.platfor er)	★+>ンセル 「ベント発生"セルに入。 可能) ログを表示するコンポーネント m.beans.gul.chooser.PFColorChooser 別名	カする 公開 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する
2)イベントの (セルをダご) メソッド メソッド メソッド addPFData3 equals(Obje getCalss() getColor() getCompon getCompon getCompon getCompon	D 発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネント内要 コンポーネントクラス SelectListener(PFDataSelectListene ect) entID() entKeys() entKeys() entKeys() entPublicName() entPublicName()	説明を、"イ ことで入力 色選択 色変現れさせるダイア マ 」pgoaist dmrcplatfor er)	★+ンセル ボント発生"セルに入 可能) ログを表示するコンポーネント m.beans.gui.chooser.PFColorChooser 別名	力する
2)イベントの (セルをダご) メンッド メンッド メンッド オンッド ロddPFData equals(Obje getCalso() getCompon getCompon getCompon getCompon getCompon	D 発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラス SelectListener(PFDataSelectListene ect) entID() entKey() entKeys() entName() entPublicName() entString() biothiotectionectict	設定 説明を、"イ ことで入力 色選択 色を選択させるダイア ス p.go.aist.dmrc.platfor er)	★+ンセル ボベント発生"セルに入 可能) □グを表示するコンポーネント m. beans.gui.chooser PFColorChooser 別格	力する
2)イベントの (セルをダご) メンッド メンッド メンッド マンッド マンッド マンッド マンッド マンッド マンッド マンッド マ	D 発生トリガーの ブルクリックする ^{オント情報編集} コンポーネント名称 コンポーネント例要 コンポーネントクラス SelectListener(PFDataSelectListener act) entED() ientKey() ientKey() ientKeys() ientFublicName() ientString() ientString() ieletListenerList()	設定 説明を、"イ ことで入力 色選択 色を選択させるダイア ス jp.go.aist.dmrc.platfor er)	★+ンセル ボベント発生"セルに入。 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名	カする 人間 公開する (公開する (公開する) (公
2)イベントの (セルをダご) メンッド メンッド addPFData equals(Obj getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon	D 発生トリガーの ブルクリックする ^{オンド情報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListener act) entID() entKey() entKey() entName() entPublicName() entPublicName() entString() SelectListenerList()	設定 説明を、"イ ことで入力" 色選択 色を選択させるダイア ス [p go.aist.dmrc.platfor er)	キャンセル ボベント発生"セルに入、 可能) ログを表示するコンボーネント m.beans.gul.chooser.PFColorChooser 別名 ロークロークロークロークロークロークロークロークロークロークロークロークロークロ	カする 次間 公開する (公開する (公開する) (公
2)イベントの (セルをダご) メンッド メンッド addPFData3 equals(Obj getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon	D 発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListene ect) enttD() entKey() enttName() entFvDilcName() entString() SelectListenerList()) fransfer() - Dransfer()	議定 説明を、"イ ことで入力 色選択 色変源にさせるダイア マ p.go.aist.dmrc.platfor	キャンセル ボント発生"セルに入。 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名	カする 公開する (公開する (公開する) (公
2)イベントの (セルをダご メソッド メソッド addPFDatat equals(Obje getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon getCompon	D 発生トリガーの ブルクリックする ^{オント情報編集} コンポーネント名称 コンポーネント内要 コンポーネントクラフ SelectListener(PFDataSelectListene ect) sentD() ientKey() ientKey() ientKey() ientFublicName() ientFublicName() ientFublicName() ientString() bieletListenerList()	議定 説明を、"イ ことで入力・ 色選択 色変派なせるダイア マ マ)	★+ンセル 「ベント発生"セルに入、 可能) ログを表示するコンポーネント m. beans.gui.chooser.PFColorChooser B)名 □	カする 公開 公開する (公開する (公開する) (公別) (公開する) (公別) (公) (公) (公) (公) (公) (公) (公) (公
2)イベントの (セルをダご) メワッド メワッド メワッド オワッド ローーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーーー	D 発生トリガーの ブルクリックする ネンド情報編集 コンポーネント名称 コンポーネント相要 コンポーネントクラス SelectListener(PFDataSelectListene ect) entID() entKeys() entKeys() entFublicName() entFublicName() entString() SelectListenerList()) fransfer() atransfer()		キャンセル ボント発生"セルに入、 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名	カする 公開する (公開する (公開する) (公用する) (公開する) (公開する) (公知) (公開する) (公開する) (公開する) (公開する) (公開する) (公開する) (公開する) (公開する) (公開する) (公知) (公用する) (公知) (公用する) (公開する) (公知) (公用する) (公知) ((公知) ((公知) (()) ()
PACADO A A A A A A A A A A A A A A	D 発生トリガーの ブルクリックする ^{ネント情報編集} コンポーネント名称 コンポーネント相要 コンポーネントの要 コンポーネントクラス SelectListener(PFDataSelectListene act) entID() entKey() entKey() entKeys() entFublicName() entFublicName() entFublicName() entString() SelectListenerList()) fransfer()	設定 説明を、"イ ことで入力" 色選択 色変現にさせるダイア セを選択させるダイア () p. go. aist.dmrc.platfor er)	キャンセル ボント発生"セルに入、 可能) ログを表示するコンボーネント m. beans.guil chooser PFColorChooser 別格 ログを表示するコンボーネント ホート パート イベント発生	力する
Aベントの (セルをダス) メソッド メソッド メソッド オソッド addPFDatas equals(Obje getCass() getCompon getComp	D 発生トリガーの ブルクリックする ネンド情報編集 コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListener edd) entID() entKey() entNue() entPublicName() entPublicName() entString() SelectListenerList()) Fransfer() aTransfer()	設定 設明を、"イ ことで入力" 色速訳 色を選択させるダイア マ p.go.aist.dmrc.platfor er)	★+ンセル ボベント発生"セルに入、 可能) ログを表示するコンボーネント m beans gui chooser PFColorChooser 別名 □// ハント発生 イベント発生 1	カする 小する 公開する
Aベントの (セルをダご) メンッド メンッド オンッド オンッド オンッド オンッド オンッド オンッド オンッド オ	D 発生トリガーの ブルクリックする ネンド情報編集 コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListener entID() ientKey() ientName() ientPublicName() ientPublicName() ientPublicName() ientString() SelectListenerList()) Transfer() atransfer() atransfer() (ベント (ベント	i説明を、"イ ことで入力」 色選択 色変艱況させるダイア ary jp go aist dmrc platfor er)	キャンセル ボベント発生"セルに入、 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名 「ハント発生 イベント発生 イベント発生 イベント発生 10Kボタンが押されたとき	カする 公開する
2)イベントの (セルをダご) メソッド メソッド オソッド オソッド オソッド オソッド オソッド オソッド オソッド オ	D 発生トリガーの ブルクリックする ネント情報編集 コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListene edt) enttD() tentKey() tentFublicName() tentString() belectListenerList()) fransfer() atransfer() atransfer()		キャンセル ベント発生"セルに入、 可能) ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名 ロークローク パーント発生 イペント発生 イペント発生 イペント発生 イペント発生 レーク レーク <td>カする 公開 公開する</td>	カする 公開 公開する
2)イベントの (セルをダご メソッド メソッド オソッド addPFDatat equals(Obje getCanspon getCompon	D 発生トリガーの ブルクリックする ^{x > ト} 情報編集 コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListener cd) selectListener(PFDataSelectListener cd) enttD() ientKey() ientKey() ientString() selectListenerList()) fransfer() aTransfer() aTransfer()		キャンセル ベント発生"セルに入、 ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名 ログロシント発生 イベント発生 イベント発生 キャンセルまたは閉じるボタンが押されたとき OKボタンが押されたとき	カする 公開する
A ベントの (セルをダン メソッド メソッド メソッド メソッド メソッド メソッド メソッド メリッド オリッド オリッド オリッド オリッド マーロン マーロ マー マー マーロ マーロ マー	D 発生トリガーの ブルクリックする ^{オント情報編集} コンポーネント名称 コンポーネントの要 コンポーネントクラフ SelectListener(PFDataSelectListene ect) SelectListener(PFDataSelectListene ect) entFublicName() entFublicName() entFublicName() entString() ielectListenerList()) fransfer() aTransfer()		キャンセル ベント発生"セルに入、 ログを表示するコンボーネント m.beans.gui.chooser.PFColorChooser 別名 パペント発生 イペント発生 キャンセルまたは閉じるボタンが押されたとき りKボタンが押されたとき キャンセル	カする 公開する

(A) 14 -	「「イント」「和福米回回		
①追7	<u>加するイベントの"NO"セ</u>	ル上でマウス右クリックし <u>、</u>	[イベント追加]を
	20 コンポーネント情報編集		
	コンポーネント名称 色選択	र	
	コンポーネント概要 色を透	試 れさせるダイアログを表示するコンポーネント	
	コンポーネントクラス jp.go.	aist.dmrc.platform.beans.gui.chooser.PFColorChooser	
	メソッド		
	メソッド	別名	公開
	addPFDataSelectListener(PFDataSelectListener)		 公開する
	equals(Object)		 公開する (二) 公開する
	getColor()		 二 公開する
	getComponentID()		□ 公開する
	getComponentKey()		□ 公開する
	getComponentKeys()		 公開する ご 公開する
	getComponentPublicName()		 ご 公開する
	getComponentString()		📃 公開する
	getPFDataSelectListenerList()		 公開する
	nasnCode() isAllowPullTransfer()		□ 公開する
	ieAllow/PuehTranefer()		<u>□ ム朝</u> 9つ □ 小朋士ス
	イベント		
	イベント	NO イベント発生	
	データ選択イベント	イベント追加 カリー	とき
		イベント削除	
പ്പ	加したがで、シー語ウセバル		
		追加するイベント番号を入力してください。 2	
		追加するイベント番号を入力してください。 2 7解 取消し	
		道加するイベント番号を入力してください。 2 7解 取消し イベント番号が追加される	
	② ③	^{適加するイベント番号を入力してください。} 2 7解 取消し イベント番号が追加される	
	 図 コンポーネント情報編集 □ンポーネント体報編集 	^{追加するイベント番号を入力してください。} 2 7解 取消し イベント番号が追加される	
	 	道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される	
	 図 コンポーネント情報編集 ■ンポーネント情報編集 ■ンポーネント名称 色遊び コンポーネント概要 色を返 コンポーネントクラス [p.go. 	道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R 翻させるダイアログを表示するコンポーネント aist dmrc.platform.beans.gui.chooser.PFColorChooser	
	 ✓ ✓<td>道加するイベント番号を入力してください。 2 7解 取消し イベント番号が追加される R 額させるダイアログを表示するコンボーネント aist dmrc.platform.beans.gui.chooser.PFColorChooser</td><td></td>	道加するイベント番号を入力してください。 2 7解 取消し イベント番号が追加される R 額させるダイアログを表示するコンボーネント aist dmrc.platform.beans.gui.chooser.PFColorChooser	
	② ③ ③ ③ □ンボーネント情報編集 □ンボーネント名称 ④変初 □ンボーネントの概要 ●を支払 □ンボーネントクラス [p.go. メソッド メソッド メソッド ⊲dBPD at Solard Linear CPD block of it To block □	道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R ERCさせるダイアログを表示するコンボーネント aist dmrc.platform.beans.gui.chooser.PFColorChooser	
	② ③ ③ ③ ③ ③ ③ ③ ③ □ンボーネント情報編集 □ンボーネント名称 ④変ガ □ンボーネントの概要 ⑤を近 □ンボーネントクラス [p.go. メソッド メソッド メソッド ぶひッド ぶひッド □ンボーネントクラス [p.go. コンボーネントクラス [p.go. コンボーネントウラス [p.go. コンボーネントクラス [p.go. コンボーネントク [p.go. コンボーネントク コンボーネントク コンボーネントク コンボーネントク [p.go. コンボーネントク コンボーネントク [p.go. コンボ	道加するイベント番号を入力してください。 2 7解 取消し イベント番号が追加される R E駅させるダイアログを表示するコンボーネント aist dmrc platform beans gui.chooser.PFColorChooser 別名	
	② コンポーネント情報編集 □ンポーネント名称 色選 □ンポーネント名称 色選 □ンポーネントの要 色を近 □ンポーネントの要 色を近 □ンポーネントクラス jp.go. メソッド メソッド メリッド addPFDataSelectListener(PFDataSelectListener) equals(Object) getClass()	道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R BRさせるダイアログを表示するコンボーネント aist dmrc.platform.beans.gui.chooser.PFColorChooser	
	② コンポーネント情報編集 □ンポーネント名称 色選 □ンポーネント名称 色選 □ンポーネントの表更 色を送 □ンポーネントの表更 色を送 □ンポーネントクラス jp.go. メソッド メソッド メリッド addPFDataSelectListener(PFDataSelectListener) equals(Object) getCass() getColor()	道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R BRさせるダイアログを表示するコンボーネント aist.dmrc.platform.beans.gui.chooser.PFColorChooser	公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する 公開する
	W コンポーネント情報編集 コンポーネント名称 色識 コンポーネント名称 色識 コンポーネントの構要 色を送 コンポーネントクラス jp.go. メソッド メソッド メリッド addPFDataSelectListener(PFDataSelectListener) equals(Object) getClass() getClor() getComponentID()	道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R BRさせるダイアログを表示するコンボーネント aist.dmrc.plafform.beans.gui.chooser.PFColorChooser	
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R 群Rさせるダイアログを表示するコンポーネント alst dmrc.plafform.beans.gui.chooser.PFColorChooser 別名	公開 公開する 公 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
	✓ コンポーネント情報編集	道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R BlRさせるダイアログを表示するコンポーネント alist dmrc.platform.beans.gui.chooser.PFColorChooser	公開する 公問する
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R BlRさせるダイアログを表示するコンポーネント alst dmrc.platform.beans.gui.chooser.PFColorChooser	
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される R 都なせるダイアログを表示するコンポーネント aist dmrc.platform.beans.gui.chooser.PFColorChooser 別名	
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される ポロント番号が追加される	
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される ポロント番号が追加される	
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される ポポント番号が追加される	公開する 公開する
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される パ 都はせるダイアログを表示するコンポーネント aist dmrc platform.beans.gui.chooser.PFColorChooser 別名	
	W コンポーネント情報編集 コンポーネントス部 色遊耕 コンポーネント内募要 色を説 コンポーネント内募又 使を説 コンポーネントクラス jp.go. メソッド addPFDataSelectListener(PFDataSelectListener) equals(Object) getComponentKey() getC	道加するイベント番号を入力してください。 2 ア解 取消し イベント番号が追加される れる れない、ト番号が追加される 別名 別名 り り り り り し し し し し	公開する く公開する
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される れる れない。 7 取消し の マーグを表示するコンボーネント コンボーネント の マーグを表示するコンボーネント コンボーネント コンボーネント の マーグを表示するコンボーネント コンボーネント の マーグを表示するコンボーネント コンボーネント の マーグを表示するコンボーネント コンボーネント コンボーネント の マーグを表示するコンボーネント コンボーネント の マーグを表示するコンボーネント コンボーネント の マーグを表示するコンボーネント コンボーネント の マーグを表示するコンボーネント コンボーネント の マーグ マーグを表示するコンボーネント コンボーネント の マーグ マーグを表示するコンボーネント コンボーネント の マーグ マーグを表示するコンボーネント コンボーネント の マーグ マーグ マーグを表示するコンボーネント コンボーネント の マーグ マーグ マーグを表示するコンボーネント の マーグ マーグ マーグ マーグを表示するコンボーネント の マーグ マーグ マーグ マーグ マーグ マーグ マーグ マーグ	
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される れる れる れる パ ない。 の パ の イベント番号が追加される	
		道加するイベント番号を入力してください。 2 了解 取消し イベント番号が追加される れる れる れる れてント番号が追加される	

削隊	余するイベント	番号の "NO	0"セル_	上でマウス右ク	<u>リックし、[-</u>	イベント削除]
	🌄 コンポーネント情報	着集				
		コンボーネント名称	色選択			
		コンポーネント概要	色を選択させる	ダイアログを表示するコンポー	ネント	
		コンポーネントクラス	ip.go.aist.dmrc.	platform.beans.gui.chooser.PF	ColorChooser	
	マメンッド					
	121 m K			別名		公開
	addPEDataSelectLister	er/PEDataSelectListene	er)			二公開する
	equals(Object)					 この目的
	getClass()					🔲 公開する
	getColor()					📃 公開する
	getComponentID()					📃 公開する
	getComponentKey()					📃 公開する
	getComponentKeys()					公開する ジョンロック ロー ロー
	getComponentName()	0				 公開する 〇 公開する
	getComponentPubliciv	ame()				<u> 二 公開</u> する 一 公開する
	getPFDataSelectListen	erList()				
	hashCode()					 ご 公開する
	isAllowPullTransfer()					 公開する
	isAllowPushTransfer()					「小朋ナス」
	イベント					
	イベント			NO イベント発生		
	データ選択イベント			0キャンセルまたは閉じ・	るボタンが押されたとき	
	データ選択イベント			10Kボタンが押されたと	:き	
	データ選択イベント			イベント追加		
				イベント削除		
			[↓ 1 ^	イベント 調整 イベント 部時 マート 新学 フレ ・ マート 部時 ・ マート 部時 ・ マート 部時 ・ マート 部時 ・ マート の に ・ ・ 、 、 、 、 、 、 、 、 、 、 、 、 、	除される	
	🔛 コンポーネント情報		(↓ 1 ^	イベント 酸定 キャンセル ドント番号が削	除される	
	🎇 コンポーネント情報	集 コンボーネント名称	↓ イ へ 色選択	イベント 酸 、 ント番号が削	除される	
	₩ コンポーネント情報	編集 コンボーネント名称 コンボーネント概要	↓ イ へ ●選択 色を選択させる	イベント	除される	•
	₩ コンポーネント情報	■集 コンボーネント名称 コンボーネント概要 コンボーネントクラス	 ↓ 1 へ ●選択 ●を選択させる。 (jp.go.aist.dmrc.) 	イベント ^{制除} 設定 キャンセル ドント番号が削 ダイアログを表示するコンポー platform.beans.gui.chooser.PF	除される ^{ネント}	
	₩ コンポーネント情報	■ ユンボーネント名称 コンボーネント概要 コンボーネントクラス	↓ ↓ 1へ 色選択 色選択させる。 3 3 1 p go.aistdmrc.	イベント ^{書解除} 設定 キャンセル ドント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF	除される ^{ネント} [≂] ColorChooser	
	メンッド メンッド メンッド	■集 コンボーネント名称 コンボーネント概要 コンボーネントクラス	↓ ↓ 1 へ 色選択 色選択 きを選択させる。 3 3 3 1 9 go.aistdmrc.	イベント部除 設定 キャンセル ドント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ^{ネント} ⁻ ColorChooser	
	メソッド メソッド メソッド addPFDataSelectLister	■集 コンボーネント名称 コンボーネント構要 コンボーネントクラス her(PFDataSelectListene	↓ ↓ 1 へ 色選択 色選択 色選択させる。 3 ()p.go.aist.dmrc.) ar)	イベント部除 設定 キャノセル ドント番号が削 ダイアログを表示するコンポー platform.beans.gui.chooser.PF 別名	除される ネント ⁻ ColorChooser	 公開 公開する
	メソッド メソッド メソッド addPFDataSelectLister equals(Object)	編集 コンポーネント名称 コンポーネント頻要 コンポーネントクラス	(● 選択 色 選択 させる。 ス (p.go.aist.dmrc.) er)	イベント部除 設定 キャンセル ドント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ネント ⁻ ColorChooser	
	メソッド メソッド メソッド addPFDataSelectLister equals(Object) getClass()	編集 コンポーネント名称 コンポーネント概要 コンポーネントクラス	(● 選択 色 選択 色 差 選択させる。 ま d (jp.go.aist.dmrc) ar)	イベント開除 設定 キャンセル ドント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ^{ネント} ⁻ ColorChooser	
	メリッド メリッド メリッド addPFDataSelectLister equals(Object) getClass() getColor()	■ コンポーネント名称 コンポーネント概要 コンポーネントクラス her(PFDataSelectListene	↓ 1 ●選択 色変現れさせる。 ()p.go.aist.dmrc.(ar)	イベント制除 設定 キャンセル キャンセル ジント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF	除される ^{ネント} ^{-ColorChooser}	
	メリッド メリッド メリッド addPFDataSelectLister equals(Object) getClass() getColor() getCompenent()()	■集 コンボーネント名称 コンボーネント概要 コンボーネントクラス her(PFDataSelectListene	↓ イへ 色選択 色を選択させる ペ ()p.go.aist.dmrc.() ar)	イベント制除 設定 キャンセル キャンセル メント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ^{ネント} ^{-ColorChooser}	公開 公開する
	メソッド メソッド メソッド メソッド addPFDataSelectLister equals(Object) getCass() getComponentkey() getComponentkey()	■ ユンボーネント名称 コンボーネント病要 コンボーネントクラス her(PFDataSelectListene	し 違 沢 色を選択させる。 R (p.go.aistdmrc, ar)	イベント制除 設定 キャンセル キャンセル メント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ネント -ColorChooser	
	メワッド メワッド メワッド addPFDataSelectLister equals(Object) getClass() getComponentKey() getComponentKey() getComponentKey() getComponentKey()	■集 コンポーネント名称 コンポーネント概要 コンポーネントのラス her(PFDataSelectListene	(色選択 色を選択させる。 R (jp go aist dmrc) ar)	イベント制除 設定 キャンセル キャンセル ジイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ネント ColorChooser	公開する ○公開する ○公開する ○公開する ○公開する ○公開する
	メソッド メソッド メソッド addPFDataSelectLister equals(Object) getClass() getComponentD() getComponentKey() getComponentKey() getComponentKey() getComponentKey() getComponentKey()	製業 コンポーネント名称 コンポーネント概要 コンポーネントクラス her(PFDataSelectListene ame()	↓ 1 ●選択 色を選択させる。 K [p go.aist.dmrc; ar)	イベント制除 支定 キャンセル キャンセル ジイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ネント ColorChooser	公開 公開する
	メンッド メンッド メンッド addPFDataSelectLister equals(Object) getColars() getColor() getComponentKey() getComponentKey() getComponentKey() getComponentName() getComponentName() getComponentName() getComponentName() getComponentName()	また コンポーネント名称 コンポーネント概要 コンポーネントクラス her(PFDataSelectListene ame()	↓ 1 へ 色選択 色変選択させる。 R jp go.aist.dmrc; ar)	イベント制除 設定 キャンセル キャンセル ジイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ^{ネント} ⁻ ColorChooser	公開 公開する
	メンッド メンッド メンッド addPFDataSelectLister equals(Object) getClass() getComponentID() getComponentKey() getComponentKey() getComponentName() getComponentName() getComponentNate() getPFDataSelectLister	幕集 コンボーネント名称 コンボーネント概要 コンボーネントクラス er(PFDataSelectListene	色 澄沢 色 澄沢 させる な	イベント削除 設定 キャクセル キャクセル シント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名 □ □ □ □ □ □ □	除される ^{ネント} ^{-ColorChooser}	公開 公開する
	メソッド メソッド メソッド addPFDataSelectLister equals(Object) getClass() getComponentKey() getComponentKey() getComponentKeys() getComponentName() getComponentName() getComponentName() getComponentName() getComponentName() getComponentName() getComponentName() getComponentName() getFDataSelectListen hashCode()	^載 集 コンポーネント名称 コンポーネント概要 コンポーネントクラス er(PFDataSelectListene	↓ イへ 色選択 色選択させる は p.go.aistdmrc; ar)	イベント部除 設定 キャノセル キャノセル ジント番号が削 ダイアログを表示するコンポー platform. beans.gui.chooser.PF 別名 日 日 日 日 日 日 日	除される ^{ネント} ^{-ColorChooser}	公開 公開する
	メソッド メソッド メソッド addPFDataSelectLister equals(Object) getColor() getComponentKey() getComponentKeys() getComponentWalme() getComponentName() getComponentName() getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN	■ コンボーネント名称 コンボーネント想要 コンボーネントクラス err(PFDataSelectListene	↓ イへ 色選択 色変現れさせる ↓ p.go.aistdmrc.	イベント部除 設定 キャノセル キャノセル ジイアログを表示するコンボー platform.beans.gui.chooser.PF 別名	除される ^{ネント} ⁻ ColorChooser	公開 公開する
	メソッド メソッド メソッド addPFDataSelectLister equals(Object) getColor() getComponentKey() getComponentKeys() getComponentKeys() getComponentName() getComponentName() getComponentName() getComponentString() getPFDataSelectLister hashCode() isAllowPultTransfer() isAllowPultTransfer()	■ コンボーネント名称 コンボーネント概要 コンボーネントクラス err(PFDataSelectListene	レ 全選択 色選択 色変選択させる () jp.go.aistdmrc.	イベント部除 設定 キャノセル キャノセル シント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名 日 ロー	除される ネント ⁻ ColorChooser	
	メソッド メソッド メソッド addPFDataSelectLister equals(Object) getClass() getComponentID() getComponentKey() getComponentKey() getComponentValme() getComponentValme() getComponentValme() getComponentValme() getComponentString() getComponentString() getComponentString() getFDataSelectLister hashCode() isAllowPultTransfer() isAllowPultTransfer()	■ コンボーネント名称 コンボーネント想要 コンボーネントクラス avxi er(PFDataSelectListene ame() erList()	色選択 色選択 色変選択させる () jp.go.aistdmrc.	イベント部除 設定 キャノセル キャノセル キャノセル シト番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名 日 日 日 日 日 日	除される ネント ⁻ ColorChooser	
	メンッド メンッド メンッド addPFDataSelectLister equals(Object) getClass() getComponentID() getComponentKey() getComponentName() getComponentName() getComponentPublicN getComp	集集 コンボーネント名称 コンボーネントの表 コンボーネントのラス コンボーネントクラス arer(PFDataSelectListene arme() erList()	(回復) 「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	パペント部除 ジェート番号が削 ジィアログを表示するコンボー platform.beans.gui.chooser.PF 別名 コーニー リント	除される ネント ⁻ ColorChooser	
	メソッド メソッド メソッド addPFDataSelectLister equals(Object) getClass() getComponentiD() getComponentiNey() getComponentKey() getComponentKey() getComponentString	■ コンボーネント名称 コンボーネント想要 コンボーネントクラス arrer(PFDataSelectListene arrer()	(回転用)	イベント部時 設定 キャンセル Sント番号が削 ダイアログを表示するコンボー platform.beans.gui.chooser.PF 別名 リ名 ローク・ログを表示するコンボー ○ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	除される ネント	
	メソッド メソッド メリッド addPFDataSelectLister equals(Object) getComponentBC() getComponentKey() getComponentKey() getComponentName() getComponentName() getComponentName() getComponentName() getComponentName() getComponentName() getFDataSelectListen hashCode() isAllowPullTransfer() isAllow	■ コンボーネント名称 コンボーネント概要 コンボーネントの可要 コンボーネントクラス aver() erList()	(色選択 色選択させる。 () go.aist.dmrc.	イベント部時 ジェート番号が削 ジィアログを表示するコンボー platform beans gui chooser PF 別名 コーシー ロークジを表示するコンボー platform beans gui chooser PF 別名 ロークジェー ロークジェー	除される ネント ⁻ ColorChooser 	公開 く、間 く、間する (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、)、) (、) (、)、) (、) (、)、) (、) (、)、) (、) (、)、) (、) (、) (、)、) (、) (、) (、) (、) (、) (、) ((((((((((
	メソッド メソッド メソッド addPFDataSelectLister equals(Object) getClass() getComponentID() getComponentKey() getComponentKey() getComponentName() getComponentName() getComponentName() getComponentString() getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentPublicN getComponentString() getPFDataSelectLister hashCode() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublitCansfer() isAllowPublicN isAllowPublicN isAllowPublicN isAllowPublicN isAllowPublicN isAllowPublitCansfer() isAllowPublicN isAllowPublitCansfer() isAllowPublicN isAl	単単 コンボーネント名称 コンボーネント名称 コンボーネントの フラス ロンボーネントクラス ロ ロンボーネントクラス ロ ロンボーネントクラス ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ ロ	 ((<td>イベント部除 数定 キャノセル キャノセル キャノセル シント番号が削 リー り名 り名 リー リー</td><td>除される ネント ⁻ColorChooser </td><td></td>	イベント部除 数定 キャノセル キャノセル キャノセル シント番号が削 リー り名 り名 リー	除される ネント ⁻ ColorChooser 	

9.2.3. イベント番号の削除

画面	コンポーネント情報編集画面						
手順	①設定するイベント番号の"イベント"セル上でマウスを左クリックする						
	→ イベント情報編集画面が表示される						
	※イベント情報/派梁昌岡/2 ながらいる						
	ペイ・シード 情報が 豆 蘇 C れ C い ない 場合、 下 因 に 小 9 記 切が ア フォル 下 C し C 衣 小 C						
	1600						
	イベン	ト名 データ選択イベント	イベント番号 0				
	内包データ						
	取得メソッド	データ名	訂兑 印月				
	getEventNo	イベント番号	イベント番号				
	getSelectedData	選択データ	選択データ				
	getSource	イベント発生元コンポーネント	イベント発生元コンポーネント				
	getSourceData	イベント対象データ	イベント対象データ				
	②イベント内包デーク	タの説明を、"説明"セルに	入力する				
	デフォルトでは -	データ名と同じ文字列が入っ	っている				
	(セルを左ダブル	クリックすることで入力可能					
	────────────────────────────────────		×	n			
	イベン	ト名 データ選択イベント	イベント番号 0				
	内包データ						
	取得メソッド	データ名	記印				
	getEventNo	イベント番号	イベント番号				
	getSelectedData	選択データ	選択データ				
	getSource	イベント発生元コンポーネント	イベント発生元コンボーネント				
	getSourceData	イベント対象データ	イベント対象データ				
	$\bigcirc + \lfloor \phi[v] + b v =$	(なっ)					
	③右上の[×]ボタンで終了(確定)						

9.2.4. イベント内包データの設定

10. アプリケーションのライセンス管理

MZ Platform 上で構築されたアプリケーションには、有償アプリケーションや期間限定体験版などラ イセンスを必要とするアプリケーションが提供されることがあります。これらはアプリケーションデー タとともに、ライセンスキー文字列が提供されます。

1)アプリケーションライセンスの登録

ライセンスが必要なアプリケーションを使用する際には、アプリケーションビルダーでもアプリケー ションローダーでも初回使用時に以下のライセンスキー入力画面が表示されます。アプリケーション提 供側から提示されたライセンスキー文字列を入力してください。一度入力したライセンスは保存され、 2回目以降の使用時にはライセンスキー入力は不要となります。



2)アプリケーションライセンスの確認/更新

現在登録されているアプリケーションライセンスは、アプリケーションビルダー上で確認することが 可能です。また、有効期間を延長した新たなライセンスキーを入手した場合、ライセンスの更新を行う ことが可能です。

画面	アプリケーションビルダー メイン画面
手順	①メニューバーから [ヘルプ]-[アプリケーションライセンス情報] を指定する
	→ 登録済みのライセンスが有効期限とともに表示される
	₩ アプリケーションライセンス確認
	アブリケーションライセンス一覧
	◆ [ID:SAMPLE] sample.mzas (有効期限:2014/1/1)
	[ID:MZ-001] MZChecker.mzas (有効期限:2014/1/1)
	◆ [ID:MZ-002] Lesson05(電卓).mzas (有効期限:2017/1/1)
	▼ [ID:MZ-003] Lesson09(人口分布).mzas (有効期限:2012/1/1)
	②更新対象のアプリケーションをダブルクリックするか、選択後に[更新]ボタンを押下
	→ 新たなライセンスキーを入力する画面が表示される
	ライセンスキー入力
	🧑
	了解 取消1.

11. XML によるアプリケーション表現

MZ Platform 2.0 から、アプリケーションデータの標準の保存形式が XML 形式(拡張子 mzax、mzcx) になりました。従来のシリアライズ形式(拡張子 apl、cmp、mzas、mzcs)では、Java のバージョンによ ってアプリケーションデータの互換性がありませんでしたが、XML 形式に移行することで異なる Java のバージョンで MZ Platform を起動しても同一のアプリケーションデータを使用することができます。 ここでは、アプリケーションを表現する XML 形式について紹介します。MZ Platform が提供する XML 入力機能を使用すれば、アプリケーションの構造や動作を XML ファイルとして記述することでア プリケーションを構築でき、XML 出力機能を使用すれば他のアプリケーションでそのアプリケーショ ンデータを利用することができます。これによって、テキストエディタや XML エディタでアプリケー ションを編集できるほか、他のプログラムでアプリケーションの作成や利用が可能です。

11.1. XML 形式ドキュメント構造



※1:ベースコンポーネント

ベースコンポーネントは構築対象を示す特別なコンポーネントで、アプリケーションビルダー 画面の最上段に表示されるコンポーネントのこと。具体的には"アプリケーション"か"複合 コンポーネント"のどちらか。

※2:コンポーネント

コンポーネントは通常のコンポーネントと複合コンポーネントの両方を示す。 複合コンポーネント情報にはその内部構造や外部インターフェイス情報も含まれる。

11.2. XML タグ

アプリケーション情報			
タグ	説明	属性	
<information></information>	情報開始	—	
—	ベースコンポーネント情報	—	
—	コンポーネント情報	—	
—	コンポーネント間接続情報	—	
—	コンポーネント画面配置情報	—	
	情報終了	—	

	ベースコンポーネント情報	
タグ	説明	属性
<base/>	ベースコンポーネント情報開始	—
1	コンポーネントバス or 複合コンポーネント	—
	ベースコンポーネント情報終了	—

	コンポーネント情報	
タグ	説明	属性
<components></components>	コンポーネント情報開始	—
—	(コンポーネント or 複合コンポーネント)×N	—
	コンポーネント情報終了	—

	コンポーネント	
タグ	説明	属性
<component></component>	コンポーネント開始	class:クラス名
—	コンポーネント属性情報	—
	コンポーネント終了	—

コンポーネント属性情報			
タグ	説明	属性	
<properties></properties>	コンポーネント属性情報開始	—	
-	コンポーネント属性×N	—	
	コンポーネント属性情報終了	—	

拡張コンポーネント属性情報			
タグ	説明	属性	
<additionalproperties></additionalproperties>	拡張コンポーネント属性情報開始	—	
	コンポーネント属性 × N	—	
	拡張コンポーネント属性情報終了	_	

コンポーネント属性			
タグ	説明	属性	
<property></property>	コンポーネント属性開始	name:属性名	
<type></type>	属性データ型開始	—	
テキスト要素	属性データ型(データ型名文字列)	-	
	属性データ型終了	—	
<value></value>	属性值開始	-	
テキスト要素	属性値(データ値文字列表現)	—	
	属性値終了	—	
	コンポーネント属性終了	—	

	コンポーネント間接続情報	
タグ	説明	属性
<invocations></invocations>	コンポーネント間接続情報開始	—
-	コンポーネント間接続×N	_
	コンポーネント間接続情報終了	_

コンポーネント間接続		
タグ	説明	属性
<invocation></invocation>	コンポーネント間接続開始	—
—	接続元コンポーネント情報	—
-	接続先情報	_
	コンポーネント間接続終了	—

接続元コンポーネント情報		
タグ	説明	属性
<source/>	接続元コンポーネント情報開始	component : コンポーネント ID ※ベースの場合は"base"
<event></event>	発生イベント開始	—
テキスト要素	発生イベント(以下のいずれかを指定)	_
	PFApplicationStart	
	PFApplicationTerminate	
	PFProcessRequest	
	PFProcessTerminate	
	PFAction	
	PFMouseButton	
	PFMouseMotion	
	PFMouseWheel	
	PFKey	
	PFScroll	
	PFScroll2D	
	PFViewPick	
	PFViewLocate	
	PFViewUpdate	
	PFDataDrop	
	PFDataCreate	
	PFDataSet	
	PFDataUpdate	
	PFDataSelect	
	$\operatorname{PFComponentCooperationResult}$	
	PFPullComponentTransferResult	
	$\operatorname{PFPushComponentTransferResult}$	
	PFPushComponentTransferReceive	
	発生イベント終了	—
	接続元コンポーネント情報終了	—

接続先情報		
タグ	説明	属性
<targets></targets>	接続先情報開始	—
-	接続先×N	_
	接続先情報終了	—

接続先		
タグ	説明	属性
<target></target>	接続先情報開始	component:コンポーネント ID ※ベースの場合は"base" method:接続メソッド名 event-no:イベント番号 mode:起動モード finally:Finally 起動 error:ErrorOnly 起動
_	メソッド引数×N	_
	接続先情報終了	_

メソッド引数		
タグ	説明	属性
<argument></argument>	メソッド引数開始	kind:引数取得方法
		METHOD_RETURN:メソッド値
		STATIC_VALUE:固定值
		COMPONENT:コンポーネント
		EVENT_VALUE:イベント内包
		EVENT_OBJECT:イベント
		METHOD_RESULT:メソッド結果
<type></type>	引数データ型開始 ※必須項目	—
テキスト要素	引数データ型(データ型名文字列)	—
	引数データ型終了	—
<value></value>	引数値開始 ※kind=STATIC_VALUE の場合	—
テキスト要素	引数値(引数値文字列表現)	—
	引数值終了	_
<component></component>	引数コンポーネント開始	—
	※kind=METHOD_RETURN/COMPONENTの場合	
テキスト要素	引数コンポーネント ID	—
	引数コンポーネント開始終了	—
<method></method>	引数取得メソッド開始	—
	※kind=METHOD_RETURN/EVENT_VALUE の場合	
テキスト要素	引数取得メソッド名	—
	引数取得メソッド開始終了	—
	メソッド引数終了	—

コンポーネント画面配置情報		
タグ	説明	属性
<displays></displays>	コンポーネント画面配置情報開始	—
—	コンポーネント画面配置×N	—
	コンポーネント画面配置情報終了	—

コンポーネント画面配置		
タグ	説明	属性
<display></display>	コンポーネント画面配置開始	container : コンテナコンポーネント ID
		※ベースの場合は"base"
-	配置コンポーネント×N	—
	コンポーネント画面配置終了	—

配置コンポーネント		
タグ	説明	属性
<component></component>	配置コンポーネント開始	position:配置位置 ※領域配置の場合のみ North South East West Center
テキスト要素	コンポーネント ID	—
	配置コンポーネント終了	—

帳票構成情報		
タグ	説明	属性
<papers></papers>	帳票構成情報開始	—
1	帳票要素構成×N	—
	帳票構成情報終了	_

帳票要素構成		
タグ	説明	属性
<paper></paper>	帳票要素構成開始	id:帳票コンポーネント ID
	帳票要素×N	—
	帳票要素構成終了	—

ラベル帳票要素		
タグ	説明	属性
<label></label>	ラベル帳票要素開始	x, y:配置座標 width, height:サイズ textfont:フォント textcolor:文字色 underline:下線有無 verticalmargin:縦余白 horizontalmargin:横余白 linemargin:行間隔 textposition:文字位置 backgroundcolor:背景色 hordercolor:罪線魚
		borderwidth:罫線太さ
_	データ設定(文字列固定 or メソッド取得形式)	—
	ラベル帳票要素終了	—

テーブル帳票要素		
タグ	説明	属性
	テーブル帳票要素開始	x, y:配置座標
		width, height : サイズ
		textfont:フォント
		textcolor:文字色
		underline:下線有無
		verticalmargin:縱余白
		horizontalmargin:横余白
		linemargin:行間隔
		textposition:文字位置
		backgroundcolor:背景色
		autoresize:テーブル高さ自動調整
		bordercolor:罫線色
		linewidth:外枠線太さ
-	テーブルヘッダ行設定	—
—	テーブル列設定情報	—
—	テーブル行設定情報	—
_	データ設定(メソッド取得形式)	_
	テーブル帳票要素終了	—

ノンルトリメリ改と	
タグ 説明	属性
<header> テーブルヘッダ行設定開始 visible : linewidth usetable textfont textcolo underline verticaln horizont linemarg textposi</header>	表示有無 ホ: ヘッダ線太さ attribute : 属性継承 : フォント r : 文字色 e : 下線有無 nargin : 縦余白 almargin : 横余白 in : 行間隔 tion : 文字位置

テーブル列設定情報		
タグ	説明	属性
<columns></columns>	テーブル列設定情報開始	linewidth:縦線太さ
—	テーブル列設定×N	
	テーブル列設定情報終了	_

テーブル列設定		
タグ	説明	属性
<column></column>	テーブル列設定開始	index : 列インデックス width : 列幅 usetableattribute : 属性継承 textfont : フォント textcolor : 文字色 underline : 下線有無 verticalmargin : 縦余白 horizontalmargin : 横余白 linemargin : 行間隔 textposition : 文字位置 backgroundcolor : 背景色
—	表示パターン設定情報	—
	テーブル列設定終了	—

表示パターン設定情報		
タグ	説明	属性
<pattern></pattern>	表示パターン設定情報開始	—
—	表示パターン設定(数値 or 日付 or 論理)	—
	表示パターン設定情報終了	—

数値型表示パターン設定		
タグ	説明	属性
<number></number>	数値型表示パターン設定開始	—
テキスト要素	フォーマット文字列	_
	数値型表示パターン設定終了	—

日付型表示パターン設定		
タグ	説明	属性
<date></date>	日付型表示パターン設定開始	—
テキスト要素	フォーマット文字列	—
	日付型表示パターン設定終了	—

論理型表示パターン設定		
タグ	説明	属性
<boolean></boolean>	論理型表示パターン設定開始	—
1	真値表示パターン設定	—
—	偽値表示パターン設定	_
	論理型表示パターン設定終了	—

真値表示パターン設定		
タグ	説明	属性
<t></t>	真値表示パターン設定開始	—
テキスト要素	表示文字列	—
	真値表示パターン設定終了	_

偽値表示パターン設定		
タグ	説明	属性
⟨f⟩	偽値表示パターン設定開始	—
テキスト要素	表示文字列	—
	偽値表示パターン設定終了	—

テーブル行設定情報		
タグ	説明	属性
<rows></rows>	テーブル行設定情報開始	linewidth:横線太さ
	テーブル行設定×N	—
	テーブル行設定情報終了	—

テーブル行設定		
タグ	説明	属性
<row></row>	テーブル行設定	index:行インデックス
		height:行高さ

バーコード帳票要素			
タグ	説明	属性	
<barcode></barcode>	バーコード帳票要素開始	x, y:配置座標 width, height:サイズ codetype:コード体系 displaystringflag:データ表示有無 addcheckdigitflag: チェックディジット付加 originalsize:原寸表示 bordercolor:罫線色 borderwidht:罫線太さ	
—	データ設定(文字列固定 or メソッド取得形式)	_	
	バーコード帳票要素終了	—	
イメージ帳票要素			
----------	---------------------------	-------------------	--
タグ	説明	属性	
<image/>	イメージ帳票要素開始	x, y:配置座標	
		width, height:サイズ	
		originalsize:原寸表示	
		bordercolor:罫線色	
		borderwidth:罫線太さ	
	データ設定(イメージ情報 or メソッド取得形式)	—	
	イメージ帳票要素終了	—	

画面イメージ帳票要素		
タグ	説明	属性
<display></display>	画面イメージ帳票要素開始	x, y:配置座標
		width, height:サイズ
		originalsize:原寸表示
		bordercolor:罫線色
		borderwidth:罫線太さ
—	データ設定(コンポーネント指定形式)	—
	画面イメージ帳票要素終了	—

データ設定		
タグ	説明	属性
<data></data>	データ設定開始	type:データ形式 string:文字列指定 method:メソッド取得 image:イメージデータ component:コンポーネント指定
テキスト要素	出力文字列(文字列固定の場合のみ)	_
テキスト要素	イメージのバイト列(イメージ情報の場合のみ)	—
—	データ設定(メソッド取得 or コンポーネント)	—
	データ設定終了	—

メソッド取得設定		
タグ	説明	属性
<component></component>	取得元コンポーネント情報	id : コンポーネント ID
<method></method>	取得元メソッド情報	name:メソッド名
<combinative></combinative>	下位階層取得元コンポーネント情報	id : コンポーネント ID
	(取得元コンポーネントが下位階層の場合)	※階層情報含む(例:"1-2-2")

コンポーネント指定設定			
タグ 説明 属性			
<component></component>	取得元コンポーネント情報	id : コンポーネント ID	
<combinative></combinative>	下位階層取得元コンポーネント情報	id:コンポーネント ID	
	(取得元コンポーネントが下位階層の場合)	※階層情報含む(例:"1-2-2")	

複合コンポーネント		
タグ	説明	属性
<component></component>	複合コンポーネント開始	reference:外部参照ファイル名
—	コンポーネント属性情報	—
_	コンポーネント情報	-
—	コンポーネント間接続情報	—
-	コンポーネント画面配置情報	—
-	外部公開メソッド情報	—
	複合コンポーネント終了	—

外部公開メソッド情報		
タグ	説明	属性
<methods></methods>	外部公開メソッド情報開始	_
-	外部公開メソッド×N	-
	外部公開メソッド情報終了	_

外部公開メソッド		
タグ	説明	属性
<method></method>	外部公開メソッド	name:メソッド名 component:元コンポーネント ID alias:公開メソッド名 (別名)

11.3. データ表現形式

XML 形式テキストベースでの表現には制限がありますので、あらゆる情報を出力できるわけではありません。例えば数値や文字列のようなテキスト表現可能なものは XML 出力可能ですが、独自オブジェクトなどについては出力できません。

1)出力対象

①コンポーネント属性情報

属性情報として XML 形式に表現できるデータは以下に限定されます。

- ・プリミティブ型(boolean, byte, char, short, int, short, long, float, double)
- ・文字列型 (java.lang.String)
- ・サイズ情報 (java.awt.Dimension)
- ・点情報(java.awt.Point)

・枠情報(javax.swing.border.Border) ※EmptyBorder, EtchedBorder, LineBorder, BevelBorder のみ

- ・アイコン情報(javax.swing.ImageIcon) ※イメージ情報のみ
- ・イメージ情報 (java.awt.Image)
- ・色情報 (java.awt.Color)
- ・フォント情報 (java.awt.Font)
- ・マルチロケール文字列(jp.go.aist.dmrc.platform.util.PFMultiLocaleString)
- ・オブジェクトリスト (jp.go.aist.dmrc.platform.util.PFObjectList)
- ・オブジェクトテーブル (jp.go.aist.dmrc.platform.util.PFObjectTable)
- ・オブジェクトツリー (jp.go.aist.dmrc.platform.util.PFObjectTree)
- ・シリアライズ可能なオブジェクト (java.io.Serializable)

なお、オブジェクト型属性(上記データ型も含める)に null 値を設定する場合には、"<value>" 要素を記述しないようにします。また、階層をもったオブジェクト(リスト/テーブル/ツリー) については、下位階層までたどって出力します。

②メソッド引数情報

メッド引数の固定値として XML 形式に表現できるデータは以下に限定されます。

- ・プリミティブ型(boolean, byte, char, short, int, long, float, double)
- ・プリミティブラッパークラス型(Boolean, Byte, Char, Short, Integer, Long, Float, Double)
- ・数値クラス型(java.math.BigInteger, java.math.BigDecimal)
- ・文字列型 (java.lang.String)
- ・クラス型 (java.lang.Class)

なお、オブジェクト型属性(上記データ型も含める)に null 値を設定する場合には、"<value>" 要素を記述しないようにします。 2)テキスト表現形式

データ型	表現形式	表現例
プリミティブ型 (ラッパークラス含む)	toString()の表現形式	true false 100 1.5 A
サイズ(Dimension)	width,height	100,100
点(Point)	x,y	100,100
枠(Border)	種別番号(右参照)	0:空枠(EmptyBorder)
		1:エッジング(EtchedBorder)
		2:ライン(LineBorder)
		3:浮き出し斜影(BevelBorder:RAISED)
		4:くぼみ斜影(BevelBorder:LOWERED)
アイコン(ImageIcon)	アイコンイメージ情報	10,10,[0000000000000000]
	width,height,[imege-data]	
イメージ情報(Image)	width,height,[imege-data]	10,10,[0000000000000000]
色情報(Color)	red,green,blue,alpha	255,255,255,255
フォント(Font)	name,style,size	Dialog,3,20
	i in the style and the style in the style is the style i	Arial,0,30
	0:PLAIN	
	1:BOLD	
	2:ITALIC	
	3:BOLD&ITALIC	
マルチロケール文字列	ロケール毎に対応する	<local></local>
(PFMultiLocaleString)	文字列を記述	<lang>ja</lang>
		<string>日本語</string>
		<local></local>
		<lang>en</lang>
		<string>English</string>
オブジェクトリスト	リスト要素の文字列表	<element></element>
(PFObjectList)	現を順に記述	<type>java.lang.String</type>
		<value>AAAAA</value>
		<element></element>
		<type>java.lang.String</type>
		<value>BBBBB</value>

データ型	表現形式	表現例
オブジェクトテーブル	テーブル要素の文字列	<row></row>
(PFObjectTable)	表現を一行単位に記述	<element></element>
		<type>java.lang.String</type>
		<value>AAAAA</value>
		<element></element>
		<type>java.lang.Integer</type>
		<value>10000</value>
		<row></row>
		<row></row>
		<element></element>
		<type>java.lang.String</type>
		<value>BBBBB<!--/value--></value>
		<element></element>
		<type>java.lang.Integer</type>
		<value>20000</value>
		<row></row>
オブジェクトツリー	ツリー要素の文字列表	<element></element>
(PFObjectTree)	現をツリー構造で記述	<type>java.lang.String</type>
		<value>ROOT</value>
		<child></child>
		<element></element>
		<type>java.lang.String</type>
		<value>LEAF=1</value>
		<child></child>
		<element></element>
		<type>java.lang.String</type>
		<value>LEAF-2</value>
		<child></child>
		<element></element>
シリアライズ可能	シリアライズデータの	0000000000000000
オブジェクト	バイト列	

11.4. XML 形式表現サンプル(参考)

```
<?xml version="1.0" encoding="Shift_JIS"?>
<information>
    <!-- Base Component -->
    <base>
         <!―サンプルアプリケーション -->
        <component class="jp.go.aist.dmrc.platform.base.PFComponentBus">
             <properties>
                 <property name="ComponentID">
                      <type>int</type>
                      <value>0</value>
                 </property>
                 <property name="ExecuteMode">
                      <type>boolean</type>
                      <value>false</value>
                 </property>
                 <property name="ApplicationName">
                      <type>java.lang.String</type>
                      <value></value>
                 </property>
             </properties>
         </component>
    </base>
    <!-- Component Declarations -->
    <components>
        <!-- 月別平均気温 -->
         <component class="jp.go.aist.dmrc.platform.beans.gui.container.PFFrame">
             <properties>
                 <property name="ComponentID">
                      <type>int</type>
                      <value>1</value>
                 </property>
                 <property name="ComponentKey">
                      <type>java.lang.String</type>
                      <value>月別平均気温</value>
                 </property>
                 <property name="Title">
                      <type>java.lang.String</type>
                      <value>月別平均気温</value>
                 </property>
                 <property name="MultiLocaleTitle">
                      <type>jp.go.aist.dmrc.platform.util.PFMultiLocaleString</type>
                      <value>
                          <locale>
                               <lang>ja</lang>
                               <string>月別平均気温</string>
                          </locale>
                      </value>
                 </property>
                 <property name="Size">
                      <type>java.awt.Dimension</type>
                      <value>480,510</value>
                 </property>
             </properties>
         </component>
        <!-- テーブル 2 -->
        <component class="jp.go.aist.dmrc.platform.beans.gui.table.PFTable">
             <properties>
                 <property name="ComponentID">
                      <type>int</type>
                      <value>2</value>
                 </property>
                 <property name="ComponentKey">
                      <type>java.lang.String</type>
```

```
<value>テーブル 2</value>
</property>
<property name="ObjectTable">
    <type>jp.go.aist.dmrc.platform.util.PFObjectTable</type>
    <value>
         <row>
             <element>
                  <type>java.lang.String</type>
                  <value>1 月</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>-4.3</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>5.4</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>16.3</value>
             </element>
         </row>
         <row>
             <element>
                  <type>java.lang.String</type>
                  <value>2 月</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>-3.7</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>5.8</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>16.4</value>
             </element>
         </row>
         <row>
             <element>
                  <type>java.lang.String</type>
                  <value>3 月</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>0.0</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>8.7</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
                  <value>18.3</value>
             </element>
         </row>
         <row>
             <element>
                  <type>java.lang.String</type>
                  <value>4 月</value>
             </element>
             <element>
                  <type>java.lang.Double</type>
```

```
<value>6.6</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>14.2</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>21.2</value>
    </element>
</row>
<row>
    <element>
         <type>java.lang.String</type>
         <value>5 月</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>12.1</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>18.7</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>23.8</value>
    </element>
</row>
<row>
    <element>
         <type>java.lang.String</type>
         <value>6 月</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>16.2</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>21.7</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>26.4</value>
    </element>
</row>
<row>
    <element>
         <type>java.lang.String</type>
         <value>7 月</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>20.3</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>25.3</value>
    </element>
    <element>
         <type>java.lang.Double</type>
         <value>28.4</value>
    </element>
</row>
<row>
```

<element> <type>java.lang.String</type> <value>8 月</value> </element> <element> <type>java.lang.Double</type> <value>21.7</value> </element> <element> <type>java.lang.Double</type> <value>27.1</value> </element> <element> <type>java.lang.Double</type> <value>28.2</value> </element> </row> <row> <element> <type>java.lang.String</type> <value>9 月</value> </element> <element> <type>java.lang.Double</type> <value>17.4</value> </element> <element> <type>java.lang.Double</type> <value>23.2</value> </element> <element> <type>java.lang.Double</type> <value>27.3</value> </element> </row> <row> <element> <type>java.lang.String</type> <value>10 月</value> </element> <element> <type>java.lang.Double</type> <value>11.0</value> </element> <element> <type>java.lang.Double</type> <value>17.8</value> </element> <element> <type>java.lang.Double</type> <value>24.6</value> </element> </row> <row> <element> <type>java.lang.String</type> <value>11 月</value> </element> <element> <type>java.lang.Double</type> <value>4.5</value> </element> <element> <type>java.lang.Double</type> <value>12.8</value> </element>

```
<element>
                       <type>java.lang.Double</type>
                       <value>21.6</value>
                  </element>
             </row>
             <row>
                  <element>
                       <type>java.lang.String</type>
                       <value>12 月</value>
                  </element>
                  <element>
                       <type>java.lang.Double</type>
                       <value>-1.2</value>
                  </element>
                  <element>
                       <type>java.lang.Double</type>
                       <value>8.1</value>
                  </element>
                  <element>
                       <type>java.lang.Double</type>
                       <value>18.3</value>
                  </element>
             </row>
         </value>
    </property>
    <property name="ColumnWidthList">
         <type>jp.go.aist.dmrc.platform.util.PFObjectList</type>
         <value>
             <element>
                  <type>java.lang.Integer</type>
                  <value>65</value>
             </element>
             <element>
                  <type>java.lang.Integer</type>
                  <value>139</value>
             </element>
             <element>
                  <type>java.lang.Integer</type>
                  <value>148</value>
             </element>
             <element>
                  <type>java.lang.Integer</type>
                  <value>142</value>
             </element>
         </value>
    </property>
    <property name="Font">
         <type>java.awt.Font</type>
         <value>Dialog,0,12</value>
    </property>
    <property name="Size">
         <type>java.awt.Dimension</type>
         <value>470,220</value>
    </property>
</properties>
<additionalproperties>
    <property name="ColumnNames">
         <type>jp.go.aist.dmrc.platform.util.PFObjectList</type>
         <value>
             <element>
                  <type>java.lang.String</type>
                  <value>月</value>
             </element>
             <element>
                  <type>java.lang.String</type>
                  <value>気温(札幌)</value>
```

```
</element>
                  <element>
                      <type>java.lang.String</type>
                      <value>気温(東京)</value>
                  </element>
                  <element>
                      <type>java.lang.String</type>
                      <value>気温(那覇)</value>
                  </element>
             </value>
         </property>
         <property name="ColumnTypes">
             <type>jp.go.aist.dmrc.platform.util.PFObjectList</type>
             <value>
                  <element>
                      <type>java.lang.Class</type>
                      <value>class java.lang.String</value>
                  </element>
                  <element>
                      <type>java.lang.Class</type>
                      <value>class java.math.BigDecimal</value>
                  </element>
                  <element>
                      <type>java.lang.Class</type>
                      <value>class java.math.BigDecimal</value>
                  </element>
                  <element>
                      <type>java.lang.Class</type>
                      <value>class java.math.BigDecimal</value>
                  </element>
             </value>
         </property>
    </additionalproperties>
</component>
<!-- 気象情報 (Lesson.8) -->
<component class="jp.go.aist.dmrc.platform.beans.tutorial.PFWeatherInformation">
    <properties>
         <property name="ComponentID">
             <type>int</type>
             <value>3</value>
         </property>
         <property name="ComponentKey">
             <type>java.lang.String</type>
             <value>気象情報(Lesson.8)</value>
         </property>
    </properties>
</component>
   - 平均気温
<!-
              -->
<component class="jp.go.aist.dmrc.platform.beans.gui.chart.PFLineChart">
    <properties>
         <property name="ComponentID">
             <type>int</type>
             <value>4</value>
         </property>
         <property name="ComponentKey">
             <type>java.lang.String</type>
             <value>平均気温</value>
         </property>
         <property name="Background">
             <type>java.awt.Color</type>
             <value>255,255,255,255</value>
         </property>
         <property name="Font">
             <type>java.awt.Font</type>
             <value>Dialog,0,12</value>
         </property>
```

```
<property name="HeaderTitle">
             <type>java.lang.String</type>
             <value>平均気温</value>
         </property>
         <property name="Size">
             <type>java.awt.Dimension</type>
             <value>470,245</value>
         </property>
    </properties>
</component>
<!-- 印刷 -->
<component class="jp.go.aist.dmrc.platform.beans.gui.button.PFButton">
    <properties>
         <property name="ComponentID">
             <type>int</type>
             <value>5</value>
         </property>
         <property name="ComponentKev">
             <type>java.lang.String</type>
             <value>印刷</value>
         </property>
         <property name="Text">
             <type>java.lang.String</type>
             <value>印刷</value>
         </property>
         <property name="PropertyEditable">
             <type>boolean</type>
             <value>true</value>
         </property>
    </properties>
</component>
<!-- 帳票 --->
<component class="jp.go.aist.dmrc.platform.beans.system.print.PFPaper">
    <properties>
         <property name="ComponentID">
             <type>int</type>
             <value>6</value>
         </property>
         <property name="ComponentKey">
             <type>java.lang.String</type>
             <value>帳票</value>
         </property>
         <property name="PaperWidth">
             <type>float</type>
             <value>210.0</value>
         </property>
         <property name="PaperHeight">
             <type>float</type>
             <value>297.0</value>
         </property>
         <property name="PrintTopMargin">
             <type>float</type>
             <value>20.0</value>
         </property>
         <property name="PrintBottomMargin">
             <type>float</type>
             <value>20.0</value>
         </property>
         <property name="PrintLeftMargin">
             <type>float</type>
             <value>20.0</value>
         </property>
         <property name="PrintRightMargin">
             <type>float</type>
             <value>20.0</value>
         </property>
```

```
</properties>
        <additionalproperties>
            <property name="PaperSize">
                <type>int</type>
                <value>14</value>
            </property>
        </additionalproperties>
    </component>
</components>
<!-- Invocation Declarations -->
<invocations>
    <invocation>
        <!--->
        <source component="base">
            <event>PFApplicationStart</event>
        </source>
        <targets>
            <!-- テーブル 2 -->
            <target component="2" mode="NORMAL" method="setObjectTable">
                <argument kind="METHOD_RETURN">
                    <type>jp.go.aist.dmrc.platform.util.PFObjectTable</type>
                    <component>3</component>
                    <method>getTemperatureData</method>
                </argument>
            </target>
            <!-- 平均気温 -->
            <target component="4" mode="NORMAL" method="setObjectTable">
                <argument kind="METHOD_RETURN">
                    <type>jp.go.aist.dmrc.platform.util.PFObjectTable</type>
                    <component>3</component>
                    <method>getTemperatureData</method>
                </argument>
            </target>
            <!-- 月別平均気温 -->
            <target component="1" mode="NORMAL" method="show"/>
        </targets>
    </invocation>
    <invocation>
        <!-- 月別平均気温 -->
        <source component="1">
            <event>PFAction</event>
        </source>
        <targets>
            <!--->
            <target component="base" mode="NORMAL" method="terminateApplication"/>
        </targets>
    </invocation>
    <invocation>
        <!-- テーブル 2 -->
        <source component="2">
            <event>PFDataUpdate</event>
        </source>
        <targets>
            <!-- 平均気温 -->
            <type>jp.go.aist.dmrc.platform.util.PFObjectTable</type>
                    <method>getSourceData</method>
                </argument>
            </target>
        </targets>
    </invocation>
    <invocation>
        <!-- 平均気温 -->
        <source component="4">
```

```
<event>PFDataUpdate</event>
        </source>
        <targets>
            <!-- テーブル 2 -->
            <target component="2" mode="NORMAL" method="setObjectTable">
                 <argument kind="EVENT_VALUE">
                     <type>jp.go.aist.dmrc.platform.util.PFObjectTable</type>
                     <method>getSourceData</method>
                 </argument>
            </target>
            <!-- テーブル 2 -->
            <target component="2" mode="NORMAL" method="setColumnName">
                 <argument kind="STATIC_VALUE">
                     <type>java.lang.String</type>
                     <value>月</value>
                 </argument>
                 <argument kind="STATIC_VALUE">
                     <type>int</type>
                     <value>0</value>
                 </argument>
            </target>
        </targets>
    </invocation>
    <invocation>
        <!-- 印刷 --->
        <source component="5">
            <event>PFAction</event>
        </source>
        <targets>
            <!-- 帳票 -->
            <target component="6" mode="NORMAL" method="previewPaper">
                 <argument kind="COMPONENT">
                     <type>java.awt.Component</type>
                     <component>1</component>
                 </argument>
            </target>
        </targets>
    </invocation>
</invocations>
<!-- Display Declarations -->
<displays>
    <!-- 月別平均気温 -->
    <display container="1">
        <!-- テーブル 2 -->
        <component position="Center">2</component>
        <!-- 平均気温 -->
        <component position="Center">4</component>
        <!-- 印刷 --->
        <component position="Center">5</component>
    </display>
</displays>
```

```
<!-- Paper Declarations -->
<papers>
    <paper id="6">
         label backgroundcolor="255,255,255,255" textcolor="0,0,0,255"
          width="130.0" height="15.0" underline="false" textposition="22" linemargin="0.0" horizontalmargin="5.0" bordercolor="0,0,0,255" bordermargin="0.5"
          x="20.0" y="10.0" textfont="Dialog,0,24" verticalmargin="5.0"
              <data type="string">月別平均気温</data>
         </label>
         <table backgroundcolor="255,255,255,255" textcolor="0,0,0,255" autoresize="true"
          width="150.0" underline="false" textposition="11" linemargin="0.0"
          horizontalmargin="5.0" bordercolor="0,0,0,255" linewidth="2.0"
          x="10.0" y="30.0" textfont="Dialog,0,14" height="120.29724" verticalmargin="5.0">
              <header backgroundcolor="255,255,153,255" textcolor="0,0,0,255" underline="false"</pre>
               textposition="22" linemargin="0.0" visible="true" horizontalmargin="5.0" linewidth="2.0"
               usetableattribute="false" textfont="Dialog,0,14" verticalmargin="5.0"/>
              <columns linewidth="0.5">
                   <column backgroundcolor="255,255,255,255" textcolor="0.0.0,255" width="37.5"
                    underline="false" textposition="22" linemargin="0.0" horizontalmargin="5.0"
                    index="0" usetableattribute="false" textfont="Dialog,0,12" verticalmargin="5.0">
                   </column>
                   <column backgroundcolor="255,255,255,255" textcolor="0,0,0,255" width="37.5"
                    underline="false" textposition="32" linemargin="0.0" horizontalmargin="5.0"
                   index="1" usetableattribute="false" textfont="Dialog,0,12" verticalmargin="5.0">
                       <pattern>
                            <number>#,##0.### °C</number>
                            <date>yyyy/MM/dd HH:mm:ss</date>
                            <boolean>
                                 <t>TRUE</t>
                                 <f>FALSE</f>
                            </boolean>
                       </pattern>
                   </column>
                   <column backgroundcolor="255,255,255,255" textcolor="0,0,0,255" width="37.5"
                    underline="false" textposition="32" linemargin="0.0" horizontalmargin="5.0"
                   index="2" usetableattribute="false" textfont="Dialog,0,12" verticalmargin="5.0">
                       <pattern>
                            <number>#,##0.### °C</number>
                            <date>yyyy/MM/dd HH:mm:ss</date>
                            <boolean>
                                 <t>TRUE</t>
                                 <f>FALSE</f>
                            </boolean>
                       </pattern>
                   </column>
                   <column backgroundcolor="255,255,255,255" textcolor="0,0,0,255" width="37.5"
                    underline="false" textposition="32" linemargin="0.0" horizontalmargin="5.0"
                   index="3" usetableattribute="false" textfont="Dialog,0,12" verticalmargin="5.0">
                       <pattern>
                            <number>#,##0.### °C</number>
                            <date>yyyy/MM/dd HH:mm:ss</date>
                            <boolean>
                                 <t>TRUE</t>
                                 <f>FALSE</f>
                            </boolean>
                       </pattern>
                   </column>
              </columns>
              <rows linewidth="0.5">
                   <row height="10.230556" index="0"/>
                   <row height="9.172222" index="1"/>
                  <row height="9.172222" index="1"/>
<row height="9.172222" index="2"/>
                  <row height="9.172222" index="3"/>
                   <row height="9.172222" index="4"/>
                   <row height="9.172222" index="5"/>
                   <row height="9.172222" index="6"/>
```



</information>

※本サンプルは一部省略している部分があり、このままでは使用できませんのでご注意ください