

# 詳細設定説明書

## 目次

1. 概要.....	2
2. クラスパスの設定について.....	2
3. 実行パラメタ設定.....	2
3.1. MZ PLATFORM パラメタ.....	2
3.1.1. <i>LogLevel</i> .....	2
3.1.2. <i>LookAndFeel</i> .....	3
3.1.3. <i>RuntimeLocale</i> .....	3
3.1.4. <i>ComponentListFile_XX</i> .....	3
3.1.5. <i>ComponentListFile</i> .....	3
3.1.6. <i>ComponentInformationFolder</i> .....	4
3.1.7. <i>BinaryDataAutoSave</i> .....	4
3.1.8. <i>ExpandAllMethods</i> .....	4
3.1.9. <i>UseDataCooperation</i> .....	4
3.1.10. <i>BrokerAddress</i> .....	4
3.1.11. <i>PlatformName</i> .....	5
3.1.12. <i>LocalhostAddress</i> .....	5
3.1.13. <i>java.rmi.server.codebase</i> .....	5
3.1.14. <i>java.security.policy</i> .....	5
3.1.15. <i>brokermonitor.broker_access_log</i> .....	5
3.1.16. <i>brokermonitor.broker_debug_log</i> .....	6
3.1.17. <i>UseOnlyLightDataCooperation</i> .....	6
3.1.18. <i>RMIPort</i> .....	6
3.1.19. <i>CombinativeComponentsFolder</i> .....	6
3.1.20. <i>AccessPolicyFile</i> .....	7
3.2. JAVA パラメタ.....	7
3.3. 各コンポーネント設定.....	7
4. JNI ライブラリの使用について.....	7

# 1. 概要

MZ Platform では実行時の設定をカスタマイズすることができます。この文書では MZ Platform でカスタマイズ可能な各種設定について説明します。なお、この文書で扱う内容は高度な設定であり、通常の使用ではこれらを行う必要はありません。

## 2. クラスパスの設定について

MZ Platform に対するクラスパスの設定は、"(インストール先)\etc"フォルダにあるクラスパス設定ファイル "PlatformClassPath.ini"を編集します。このファイルには、1行につき1つのパスを設定し、パスにはローカルマシンの jar/zip ファイル、およびフォルダを指定することができます。クラスパスの優先順は記述されている順となります。また、行の初めが # である場合、その行はコメントと見なされます。

例)

```
# MZ Platform ClassPath
jars¥mzplatform.jar
# MZ Checker ClassPath
jars¥MZChecker¥mzplatform.jar
```

## 3. 実行パラメタ設定

MZ Platform の動作を外部から設定するには、"(インストール先)\etc"フォルダにある初期化ファイル "Platform.ini" を編集します。各パラメタの意味は以下の通りです。

### 3.1. MZ Platform パラメタ

#### 3.1.1. LogLevel

実行時のエラー情報やトレース情報などのログ出力のレベルを設定します。

[ 設定値 ]

"0" : 出力なし

"1" : エラーログのみ出力

"2" : 重要ログのみ出力

"3" : すべて出力

[ デフォルト(省略時) ]

重要ログのみ出力

### 3.1.2. LookAndFeel

実行時の Look&Feel を設定します。

[ 設定値 ]

"Windows" : Windows ユーザーインターフェイス

"Motif" : Motif ユーザーインターフェイス

"Java" : Java(Metal)ユーザーインターフェイス

[ デフォルト(省略時) ]

自動設定

### 3.1.3. RuntimeLocale

実行時のロケールを設定します。

[ 設定値 ]

言語コードを表す英字 2 文字 (以下例)

"ja" : 日本語

"en" : 英語

[ デフォルト(省略時) ]

実行環境にあわせて動作

### 3.1.4. ComponentListFile\_XX

コンポーネント追加時に使用するコンポーネント一覧ファイル名を設定します。パスを指定する場合の区切り文字は「¥」ではなく「¥¥」もしくは「/」となります。

[ 名称 ]

"XX"は言語コードを表す英字 2 文字 (以下例)

日本語 : PlatformComponents\_ja

英語 : PlatformComponents\_en

[ 設定値 ]

該当するロケール時に使用するコンポーネント一覧ファイル名。リリース時の設定は以下。

ComponentListFile\_ja=etc¥¥PlatformComponents\_ja.ini

ComponentListFile\_en=etc¥¥PlatformComponents\_en.ini

[ デフォルト(省略時) ]

実行ロケールが省略されている場合、デフォルト設定"ComponentListFile"を使用(3.1.5 参照)。

### 3.1.5. ComponentListFile

コンポーネント一覧ファイル名のデフォルトを設定します。パスを指定する場合の区切り文字は「¥」ではなく「¥¥」もしくは「/」となります。

[ 設定値 ]

実行時ロケールのコンポーネント一覧ファイル名が指定されていない場合に使用されるコンポーネント一覧ファイル名。リリース時の設定は以下。

ComponentListFile=etc¥¥PlatformComponents\_en.ini

[ デフォルト(省略時) ]

実行時ロケールのコンポーネント一覧ファイル名が指定されていない場合は省略不可。

### 3.1.6. ComponentInformationFolder

コンポーネント情報の格納フォルダを設定します。パスを指定する場合の区切り文字は「¥」ではなく「¥¥」もしくは「/」となります。

[ 設定値 ]

コンポーネント情報が格納されているフォルダ名。絶対パス、または実行時フォルダからの相対パスで指定。リリース時の設定は以下。

ComponentInformationFolder=components

[ デフォルト(省略時) ]

省略不可。

### 3.1.7. BinaryDataAutoSave

アプリケーションや複合コンポーネントを XML 形式(拡張子 mmax/mzcx/xml)で保存する場合に、未知の不具合によるデータの損失を避けるためにシリアライズ形式のデータ(拡張子 mzas/mzcs)を同時に保存する機能を提供しています。この機能の有効・無効を設定します。

[ 設定値 ]

false を指定すると自動保存が解除され、それ以外は true に設定される。リリース時の設定は以下。

BinaryDataAutoSave=true

[ デフォルト(省略時) ]

true (自動保存有効)

### 3.1.8. ExpandAllMethods

アプリケーションビルダー上でのメソッド起動の折りたたみ表示のデフォルト状態を設定します。

[ 設定値 ]

展開表示する場合は"true"、折りたたみ表示する場合は"false"を指定する。

ExpandAllMethods=true

[ デフォルト(省略時) ]

true (すべて展開)

### 3.1.9. UseDataCooperation

データ連携機能を使用するかどうかを設定します。

[ 設定値 ]

データ連携機能を使用する場合は"true" (大文字/小文字区別なし) を指定。

それ以外の設定については、使用しないと判断される。

[ デフォルト(省略時) ]

データ連携機能は使用しない。

### 3.1.10. BrokerAddress

データ連携を行うネットワーク環境上で、プラットフォームが旧データ連携機能で接続するブローカのア

ドレスを指定します。データ連携機能を使用するように設定(3.1.9 参照)されていることが前提です。

[ 設定値 ]

接続先ブローカが稼働するコンピュータ名。

[ デフォルト(省略時) ]

データ連携機能を使用する場合、省略不可。

### 3.1.11. PlatformName

データ連携を行うネットワーク環境上で、プラットフォーム稼働環境を識別するためのプラットフォーム名を指定します。データ連携機能を使用するように設定(3.1.9 参照)されていることが前提です。

[ 設定値 ]

連携するネットワーク環境上で、一意に識別可能な任意の名称。

[ デフォルト(省略時) ]

データ連携機能を使用する場合、省略不可。

### 3.1.12. LocalhostAddress

データ連携を使用する場合に、プラットフォームを起動するマシンのアドレスを記述します。

[ 設定値 ]

マシンのホストアドレス (もしくは IP アドレス)。

[ デフォルト(省略時) ]

データ連携を使用する場合、省略不可。

### 3.1.13. java.rmi.server.codebase

プラットフォームが旧データ連携機能で接続するブローカの物理アドレスを指定します。データ連携機能を使用するように設定(3.1.9 参照)されていることが前提です。

[ 設定値 ]

接続先ブローカの接続先 URL。

記述例) `http://hostname:80/broker/rmi/`

[ デフォルト(省略時) ]

データ連携機能を使用する場合、省略不可。

### 3.1.14. java.security.policy

データ連携機能を使用する場合の、java セキュリティ設定ファイル名を指定します。データ連携機能を使用するように設定(3.1.9 参照)されていることが前提です。パスを指定する場合の区切り文字は「¥」ではなく「¥¥」もしくは「/」となります。

[ 設定値 ]

java セキュリティ設定ファイル名。

記述例) `etc¥¥java.policy`

[ デフォルト(省略時) ]

データ連携機能を使用する場合、省略不可。

### 3.1.15. brokermonitor.broker\_access\_log

プラットフォームが旧データ連携機能で接続するブローカの出力するアクセスログファイル名を指定しま

す。このファイルはブローカモニタが稼働状況を監視するための、入力ファイルとして使用します。

[ 設定値 ]

ブローカのアクセスログファイル名

[ デフォルト(省略時) ]

ブローカモニタを使用する場合、省略不可。

※省略されている場合は、ログ情報非表示。

### 3.1.16. brokermonitor.broker\_debug\_log

プラットフォームが旧データ連携機能で接続するブローカの出力するアプリケーションログファイル名を指定します。このファイルはブローカモニタが稼働状況を監視するための、入力ファイルとして使用します。

[ 設定値 ]

ブローカのアプリケーションログファイル名。

[ デフォルト(省略時) ]

ブローカモニタを使用する場合、省略不可。

※省略されている場合は、ログ情報非表示。

### 3.1.17. UseOnlyLightDataCooperation

旧データ連携機能を使用せず、現行版のみを用いるかどうかの指定を行います。

[ 設定値 ]

旧データ連携機能を使用せず現行版のみを用いる指定(true/false)。

[ デフォルト(省略時) ]

true (現行版のみ使用)

### 3.1.18. RMIPort

データ連携機能を使用する場合に、RMI レジストリが使用するポートを指定します。リモートオブジェクトのエクスポートにもこのポート番号を使用します。

[ 設定値 ]

RMI レジストリが使用するポート番号。

[ デフォルト(省略時) ]

1099

### 3.1.19. CombinativeComponentsFolder

複合コンポーネントの外部参照機能を使用する場合に、外部参照先のフォルダ名を指定します。複合コンポーネント追加時のメニュー一覧にはこのフォルダ内容が表示されます。

フォルダ名は MZ Platform インストールフォルダからの相対パスもしくは絶対パスで記述します。パスを指定する場合の区切り文字は「¥」ではなく「¥¥」もしくは「/」となります(「3.1.4 ComponentListFile\_XX」参照)。

[ 設定値 ]

外部参照先のフォルダ名(インストールフォルダからの相対パスもしくは絶対パス)

[ デフォルト(省略時) ]

外部参照機能を使用する場合、省略不可。

### 3.1.20. AccessPolicyFile

データ連携機能を使用する場合に、連携先プラットフォームに対するアクセスポリシーを記述したプロパティファイル(通常 AccessPolicy.ini)のファイル名を指定します。ファイル名は MZ Platform インストールフォルダからの相対パスもしくは絶対パスで記述します。パスを指定する場合の区切り文字は「¥」ではなく「¥¥」もしくは「/」となります。

[ 設定値 ]

アクセスポリシーファイルのファイル名 (インストールフォルダからの相対パスもしくは絶対パス)

[ デフォルト(省略時) ]

etc¥¥AccessPolicy.ini

## 3.2. Java パラメタ

Java 実行時のシステムパラメタの設定を行うことができます。以下はその例です。

1)プロキシ関連

http.proxyHost

http.proxyPort

http.nonProxyHosts

2)描画系

sun.java2d.noddraw

sun.java2d.d3d

## 3.3. 各コンポーネント設定

"Platform.ini"で設定したパラメタは、実行時のシステムプロパティとして参照できます。コンポーネント作成時などに、外部パラメタとして実行時に渡したい情報がある場合、このファイルに記述しておくことで、コンポーネント内から参照ができます。

## 4. JNI ライブラリの使用について

JNI を利用する場合は、以下のいずれかを行ってください。

- ・ライブラリファイルをインストールフォルダの lib フォルダに格納する
- ・ライブラリファイルを PATH が通っているフォルダに格納する
- ・ライブラリファイルが存在するフォルダを PATH に追加する