

IBM Unica Campaign
バージョン 8 リリース 6
2012 年 4 月 30 日

検証 PDK ガイド

IBM

注

本書および本書で紹介する製品をご使用になる前に、15 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM Unica Campaign (製品番号 5725-D22) バージョン 8 リリース 6 モディフィケーション 0、および新しいエディションで明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

© Copyright IBM Corporation 1998, 2012.

目次

IBM Unica Validation Plug-in

Developer's Kit (PDK) 1

検証 PDK の追加のヘルプ 1

検証 PDK の内容 2

 検証 PDK のコンポーネント 2

 参照資料を見つける方法 3

検証 PDK の機能 3

 IBM Unica Campaign へのプラグインのロード 3

 オファーとキャンペーンの検証 4

検証 PDK の構成 5

 validationClass 5

 validationClasspath 6

 validatorConfigString 7

プラグインの開発 7

 概説 7

プラグイン作成の主な手順 8

 セットアップ 9

バリデーターのビルド 9

IBM Unica Campaign の構成 10

バリデーター構成のテスト 10

バリデーターの作成 10

実行可能サンプル 11

 サンプル実行可能プラグイン用に Campaign を構

 成する 11

 想定される実行可能プログラム使用インターフェ

 ース 12

例 12

 キャンペーン編集を防止する 12

IBM Unica 技術サポートへの連絡 13

特記事項 15

 商標 17

IBM Unica Validation Plug-in Developer's Kit (PDK)

IBM® Unica Validation Plug-in Developer's Kit (PDK) を使用すると、IBM Unica Campaign で使用するためのカスタム検証ロジックを開発できます。検証 PDK は、IBM Unica Campaign に付属している汎用性の高いプラグイン・フレームワークのサブクラスです。

この章の主なセクションは次のとおりです。

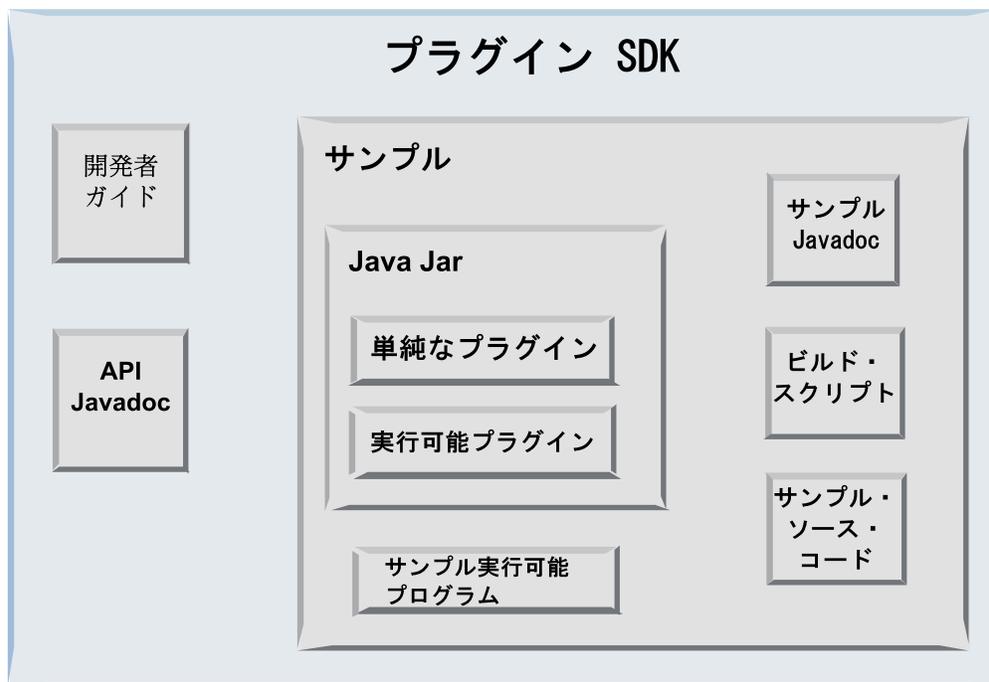
- 2 ページの『検証 PDK の内容』
- 3 ページの『検証 PDK の機能』
- 5 ページの『検証 PDK の構成』
- 7 ページの『プラグインの開発』
- 8 ページの『プラグイン作成の主な手順』
- 11 ページの『実行可能サンプル』
- 12 ページの『例』

検証 PDK の追加のヘルプ

Validation Plug-in Developer's Kit (PDK) の使用方法に関するヘルプ情報が必要な場合は、Unica テクニカル・サポートにお問い合わせください。詳しくは、13 ページの『IBM Unica 技術サポートへの連絡』を参照してください。

検証 PDK の内容

検証 PDK には、IBM Unica Campaign に検証を追加するための Java プラグインまたはコマンド行実行可能プログラムの開発に必要なすべてのものが含まれています。



検証 PDK のコンポーネント

Validation Plug-in Developer's Kit (PDK) には、以下のコンポーネントが含まれています。

コンポーネント	説明
検証 PDK ガイド	この資料です。
サンプル	サンプルは、PDK の使用方法を示すドキュメント化された、ビルド可能な例です。
Java .jar ファイル	サンプル・プラグインが入っているサンプル JAR ファイル。この jar には以下のものが含まれています。 <ul style="list-style-type: none">• 単純なプラグイン: 自己完結型バリデーター・クラスの例。• 実行可能プラグイン: 検証のためにユーザー定義のコマンド行実行可能プログラムを実行するバリデーターの例。
サンプル実行可能プログラム	UNIX で実行可能プラグインと共に使用できるコマンド行実行可能プログラム。
サンプル Javadoc	サンプル・バリデーター・クラスについて説明する Javadoc。

コンポーネント	説明
ビルド・スクリプト	組み込みソース・コードをビルドして使用可能なバリデーター・プラグインにする Ant スクリプト。
サンプル・ソース・コード	単純なバリデーターおよび実行可能バリデーター用の Java ソース・コード。

参照資料を見つける方法

検証 PDK には、プラグイン API およびサンプル・コードに関する参照情報が含まれています。参照情報は Javadoc 形式です。資料を表示するには、以下のファイルを開いてください。

```
C:¥Unica_Home¥Campaign_Home¥devkits¥validation¥javadoc¥index.html
```

ここで *Unica_Home* は IBM Unica ルート・インストール・ディレクトリーのパス、*Campaign_Home* は Campaign インストール・ディレクトリーのパスです。

例えば、資料のパスは次のようになります。

```
C:¥IBM¥Unica¥Campaign¥devkits¥validation¥javadoc¥index.html
```

任意の Web ブラウザーで Javadoc を表示できます。

検証 PDK の機能

検証 PDK を使って作成されたプラグインは、キャンペーン、オファー、またはその両方に関するカスタム検証ロジックを実行することができます。検証ロジックを次のように使用できます。

- 拡張属性の検査 (例えばオプション・フィールドまたは関連する従属フィールドに対する妥当性検査)
- IBM Unica Marketing Platform のスコープ外にある許可サービスの提供 (例えば、どのユーザーがどの拡張属性の編集を許可されるかの検証)

IBM Unica Campaign へのプラグインのロード

以下の 2 つの方法で API を使用できます。

- これを使用して、アプリケーションにロードされる Java クラス・プラグインを作成します。
- いずれかの組み込みプラグインを使用して、検証を扱う実行可能アプリケーションを呼び出します。

アプリケーションにロードされる Java クラス・プラグインの作成

検証 PDK は、インターフェース、ヘルパー・クラス、およびこれらのクラスを開発するための開発者ツールを備えています。



検証を処理するためのアプリケーションの呼び出し

検証 PDK の 2 番目の使用方法は、いずれかの組み込みプラグインを使用して、検証を扱う実行可能アプリケーションを呼び出すことです。



任意の言語でこの実行可能プログラムを作成できますが、IBM Unica Campaign サーバー上にこれを格納して、サーバー上で実行する必要があります。実行可能プログラムを呼び出すプラグインは、検証される情報を含む XML ファイルを送信します (例えばオブジェクトを編集しているユーザー、そのオブジェクトのすべての標準/拡張属性の変更前/後の値)。IBM Unica Campaign は、XML ファイル形式で結果情報が戻されることを想定しています。詳しくは、7 ページの『プラグインの開発』を参照してください。

オファーとキャンペーンの検証

検証 PDK ではオファーとキャンペーンを検証することができます。オファーまたはキャンペーンのオブジェクトが保存されるたびに、検証プラグインが定義されている場合はそれが IBM Unica Campaign によって自動的に呼び出されます。IBM Unica Campaign は、プラグインの検証メソッドを呼び出すときにフラグを設定します。IBM Unica Campaign は以下のフラグを渡します。

- ValidationInputData.CAMPAIGN_VALIDATION (キャンペーンを追加または変更するとき)

または

- ValidationInputData.OFFER_VALIDATION (オファーを追加または編集するとき)

その後、これらのフラグを使用して、オファーやキャンペーンに適用される検証規則を構成することができます。

検証 PDK の構成

検証 PDK で使われる構成パラメーターは、使用すべきプラグイン・クラスを見つける方法を Campaign に指示します。また、これらのプラグインにいくつかの構成情報を渡すこともできます。

すべての設定は、以下の場所にある構成ページに含まれています。

Campaign > partitions > partition[n] > validation

注: 検証は複数のパーティションで機能します。 partition[n] を任意のパーティション名に変更して、それらのパーティションの検証ルーチンを提供することもできます。

このセクションには、以下の設定が含まれています。

- 『validationClass』
- 6 ページの 『validationClasspath』
- 7 ページの 『validatorConfigString』

validationClass

プロパティ	説明
説明	これは、検証に使用するクラスの名前です。 validationClasspath プロパティの値は、このクラスの格納場所を示しています。
詳細	クラスは、パッケージ名で完全修飾する必要があります。このプロパティが設定されていない場合、Campaign はカスタム検証を実行しません。
例	com.unica.campaign.core.validation.samples.SimpleCampaignValidator これは、validationClass をサンプル・コードの SimpleCampaignValidator クラスに設定します。
デフォルト	デフォルトでは、パスが設定されていません。 <property name="validationClass" />

validationClasspath

プロパティ	説明
説明	これは、カスタム検証で使われるクラスのパスです。絶対パスまたは相対パスが可能です。
詳細	<p>パスの末尾がスラッシュ (UNIX ではスラッシュ /、Windows では円記号 ¥) になっている場合、Campaign では、使用すべき Java プラグイン・クラスが含まれるディレクトリーのパスと見なされます。</p> <p>パスの末尾がスラッシュでない場合、Campaign では、Java クラスを含む .jar ファイルの名前と見なされます。例えば、次の値の場合、</p> <pre data-bbox="948 737 1403 789">/opt/Unica/Campaign/devkits/validation/lib/validator.jar?</pre> <p>これは、プラグイン開発者キットに付属のすぐに使用可能な JAR ファイルを指す、UNIX プラットフォーム上のパスです。</p> <p>相対パスである場合、Campaign を実行しているアプリケーション・サーバーによって動作が異なります。WebLogic ではドメイン作業ディレクトリーのパスが使用され、このパスはデフォルトでは次のとおりです。</p> <pre data-bbox="935 1136 1377 1163">c:%bea%user_projects%domains%mydomain</pre> <p>設定値にストリングが含まれない場合、Campaign はプラグインのロードを試行しません。</p>
例	<pre data-bbox="935 1308 1390 1360">/opt/Unica/Campaign/devkits/validation/lib/validator.jar?</pre> <p>これは、プラグイン開発者キットと共にパッケージ化されている JAR ファイルを指す、UNIX プラットフォーム上のパスです。</p>
デフォルト	<p>デフォルトでは、パスが設定されていません。</p> <pre data-bbox="935 1587 1398 1614"><property name="validationClasspath" /></pre>
関連項目	使用するクラスの指定については、5 ページの『validationClass』の情報を参照してください。

validatorConfigString

プロパティ	説明
説明	これは、Campaign によってロードされるときにバリデーター・プラグインに渡される文字列です。
詳細	プラグインがこの文字列をどのように使用するかは、設計によって異なります。システムがプラグインをロードするときに構成文字列をプラグインに送るために、これを使用できます。 例えば、(PDK に含まれるサンプル実行可能プラグインの) ExecutableCampaignValidator は、実行すべき実行可能プログラムを示すためにこのプロパティを使用します。
例	検証スクリプトとしてサンプル Bourne シェル・スクリプトを実行するには、 validatorConfigString を次のように設定します。 <pre>/opt/unica/campaign/devkits/validation/ src/com/unica/campaign /core/validation/samples/validate.sh</pre>
デフォルト	デフォルトでは、パスが設定されていません。 <pre><property name="validatorConfigString" /></pre>

プラグインの開発

概説

プラグインは、起動時にロードされ、キャンペーンまたはオファーが検証されるたびに呼び出される Java クラスです。ユーザーがキャンペーンを保存するたびに、この検証が発生します。検証 PDK に備わっているツールを使用すると、独自の Java プラグインを作成できます。サンプル・プラグインのソース・コードと、プラグインのコンパイルに使用する Ant ファイル (Java ベースのビルド・ツールである Apache Ant) がこれに含まれています。以下のセクションでは、プラグインの開発環境をセットアップする方法を示した後、独自のプラグインの作成について説明します。

サンプル・バリデーター

サンプル・バリデーターは Campaign 標準インストールに含まれています。

- 自己完結型のプラグイン SimpleCampaignValidator は、カスタム許可、許容されるキャンペーン名の検証などを行う方法を示します。これは、以下のパスにあります。

```
devkits/validation/src/com/unica/campaign/core/validation/
samples/SimpleCampaignValidator.java
```

これを操作するときには、直接編集する代わりにクラスのコピーを作成することをお勧めします。

- ExecutableCampaignValidator は、検証を実行するために実行可能アプリケーションを呼び出す Java プラグインです。ExecutableCampaignValidator のソース・コードは SimpleCampaignValidator と同じディレクトリーに入っています。

ただし、この例の本当の目的は、検証用のコマンド行実行可能プログラムとして使用することです。このファイルは次のパスにあります。

```
devkits/validation/src/com/unica/campaign/
core/validation/samples/validate.sh
```

このファイルはサンプル・ループバック実行可能プログラムであり、一般的な種類の検証作業を示しています。

テスト・ハーネス

エクストリーム・プログラミングその他のアジャイル手法が広く使われ、役立っています。これらの方法の重要な一面は、これらがテスト中心であることです。これらの方法を使用するお客様は、単体テストを広範に使用します。検証 PDK は、Campaign の外部でプラグインを実行するためのテスト・ハーネスを備えることにより、これらの方法をサポートします。IBM Unica Campaign の中にコードを配置しなくてもテストできるため、プラグイン開発プロセスがより迅速になります。

テスト・ハーネスを使用するには、次のようにします。

1. プラグイン内の検証ロジックを反映するよう、単体テスト・ケースを変更します。
2. ビルド・スクリプトを次のように実行します。
 - 単体テストを実行せずにプラグインを作成するには、"ant jar" コマンドを使ってビルド・スクリプトを実行します。
 - プラグインを作成するとともに単体テストを実行するには、"ant run-test" コマンドを使ってビルド・スクリプトを実行します。

ビルド・スクリプト

PDK のビルド・スクリプトはディレクトリー内のすべてのクラスをコンパイルして、IBM Unica Campaign での使用に適した jar の中にそれらを格納します。提供されているビルド・スクリプトで使用されるディレクトリーは、次のとおりです。

```
devkits/validation/src/com/unica/campaign/core/validation/samples/
```

プラグイン作成の主な手順

プラグインを作成するための主な手順は次のとおりです。

- 9 ページの『セットアップ』
- 9 ページの『バリデーターのビルド』
- 10 ページの『IBM Unica Campaign の構成』
- 10 ページの『バリデーター構成のテスト』

- 10 ページの『バリデーターの作成』

以下のセクションでは、.jar ファイルを作成するためのソースをビルドする方法について示します。

セットアップ

任意のマシンに検証 PDK をインストールできますが、これを使って作成されたプラグインは IBM Unica Campaign を実行しているマシン上に配置される必要があります。プラグインのテストに使われるマシン上に PDK をインストールすることをお勧めします。

PDK で Java プラグインを作成するには、マシン上に Jakarta Ant および Sun Java Developer Kit が必要です。互換性を確実に得るために、アプリケーション・サーバーに付属の Ant および JDK パッケージを使用することをお勧めします。

注: 別の JDK を使用することも可能です。ただし、アプリケーション・サーバー付属の JDK を使用すると、作成されるプラグインはコンパイル時の互換性検査に合格します。例えば (実行用に 1.4 JDK を使う) WebLogic 8.1 を使用する場合、Campaign がプラグインの使用を試みたときに、1.5 JDK 固有のクラスがあれば「クラスが見つからない」というエラーが発生します。コンパイルで 1.4 JDK を使用すると、1.5 JDK 固有のクラスの使用が防止され、Campaign でのプラグイン互換性が確実に得られます。

検証 PDK の使用のために環境をセットアップするには、次のようにします。

1. Ant 実行可能プログラムが入っているフォルダーをパスに追加します。例えば Windows 上のデフォルト・ディレクトリーに WebLogic 8.1 がインストールされている場合、以下をパスに追加します。

```
c:\bea\weblogic81\server\bin
```

2. JAVA_HOME 環境変数を、JDK の bin および lib ディレクトリーが入っているディレクトリーに設定します。例えば Windows 上の WebLogic 8.1 の場合、JAVA_HOME を次のように設定します。

```
c:\bea\jdk141_03
```

バリデーターのビルド

PDK に備わっている Ant スクリプトは、サンプル・ファイル内のすべてのコードをビルドすることができます。このスクリプトは、デフォルト動作として、検証クラスを含む jar を作成します。また、実動でのプラグイン使用を試みる前に Campaign で機能することを確認するために、バリデーターに対する Javadoc および run test をオプションで作成することもできます。

バリデーターをビルドするには、次のようにします。

1. PDK ディレクトリーに移動します。

```
<Unica_Home\Campaign_Home>%devkits%\validation\build
```

このディレクトリー内に Ant スクリプト build.xml があります。

2. コマンド行で Ant jar を実行します。

Ant はスクリプトを実行して、`validator.jar` という JAR ファイルを次のディレクトリー内に生成します。

```
<Unica_Home%Campaign_Home>%devkits%validation%build%lib
```

これで、IBM Unica Campaign で使用可能なカスタム・バリデーターが生成されました。次のセクションでは、このバリデーターを使用するよう Campaign を構成する方法について説明します。

IBM Unica Campaign の構成

バリデーター・プラグインを作成した後、それを使用可能にするには、プラグインの場所を IBM Unica Campaign に指示する必要があります。

`SimpleCampaignValidator` を使用するには、5 ページの『検証 PDK の構成』で説明されているプロパティを次のように設定します。

- `validationClasspath`: `Unica%campaign%devkits%validation%lib%validator.jar`
- `validationClass`:
`com.unica.campaign.core.validation.samples.SimpleCampaignValidator`
- 構成ストリングは使われないため、`SimpleCampaignValidator` の使用のために `validatorConfigString` を設定する必要はありません。

バリデーター構成のテスト

`SimpleCampaignValidator` クラスを含む `validator.jar` ファイルを作成し、必要に応じて構成を変更した後、プラグインをテストして、使用することができます。このプラグインは、"badCampaign" という名前のキャンペーンがユーザーによって保存されるのを防ぎます。

構成をテストするには、次のようにします。

1. 変更を有効にするために、アプリケーション・サーバーを再デプロイします。アプリケーション・サーバーの再デプロイについて、詳しくはサーバーの資料を参照してください。
2. IBM Unica Campaign にログインして、キャンペーン作成ページまでナビゲートします。
3. **badCampaign** という名前のキャンペーンを作成して、それを保存するよう試みます。

すべての構成が正しい場合、新しいキャンペーンを保存することはできません。バリデーターからエラー・メッセージを受け取ります。

バリデーターの作成

このセクションで作成する検証プラグインは `SimpleCampaignValidator` によく似ていますが、"badCampaign2" というキャンペーンの作成を抑止します。

バリデーターを作成するには、次のようにします。

1. 以下の場所にあるサンプル・バリデーター `SimpleCampaignValidator.java` のコピーを作成します。

```
<Unica_Home%Campaign_Home>%devkits%validation%src%com
%unica%campaign%core%validation%samples
```

2. そのコピーに MyCampaignValidator.java という名前を付けて、ソースと同じディレクトリーにそれを配置します。
3. エディターで MyCampaignValidator.java を開きます。文書内の "badCampaign" という語を見つけて、それを "badCampaign2" という語に置き換えます。
4. ファイルを保存して、エディターを閉じます。
5. バリデーターを再び作成します。詳しくは、9 ページの『バリデーターのビルド』を参照してください。

注: 使用中にアプリケーション・サーバーによって validate.jar ファイルがロックされる場合、バリデーターを作成する前にサーバーを停止する必要があります。

6. 新しいクラスを使用するよう、campaign_config.xml を次のように再構成します。

```
<property name="validationClass"
value="com.unica.campaign.core.validation.samples.MyCampaignValidator">
```

7. バリデーターをテストします。詳しくは、10 ページの『バリデーター構成のテスト』を参照してください。

これで、"badCampaign2" という名前のキャンペーンを保存できなくなります。

実行可能サンプル

サンプル・バリデーターには、コマンド行から実行可能プログラムを実行できる ExecutableCampaignValidator というバリデーターも含まれています。

このセクションでは、

- サンプル実行可能プラグインの実行用に IBM Unica Campaign をセットアップする方法を示します。さらに
- 実行可能プログラム使用インターフェースに適合する独自の実行可能プラグインを作成する方法について説明します。

サンプル実行可能プラグイン用に Campaign を構成する

ExecutableCampaignValidator を使用するには、5 ページの『検証 PDK の構成』で説明されているプロパティーを次のように設定してください。

- validationClasspath:

```
<Campaign_home>%devkits%validation%lib%validator.jar
```

- validationClass:

```
com.unica.campaign.core.validation.samples.ExecutableCampaignValidator
```

- validatorConfigString:

```
<Campaign_home>%pdk%bin%validate.sh
```

PDK に付属のサンプル・スクリプトは、UNIX 用の Bourne シェル・スクリプトです。これは、ユーザー名 "badUser" を持つユーザーによるキャンペーン作成を拒否します。この実行可能プログラムのコードは、以下のディレクトリーにあります。

```
devkits¥validation¥src¥com¥unica¥campaign¥core¥
validation¥samples¥validate.sh
```

実装に合わせて適切な検証を実行する、独自のスクリプトを開発する必要があります。このようなテキスト処理スクリプトには、PERL、Python などのスクリプト言語はもちろん、コマンド行から実行可能な任意の言語を使用できます。

想定される実行可能プログラム使用インターフェース

ExecutableCampaignValidator は、以下の引数を含むコマンド行を使って実行可能プログラムを呼び出します。

- `executable_name`: これは、IBM Unica Marketing Platform の `validatorConfigString` で設定されるストリングです。
- `data_filename`: これは、実行可能プログラムが入力として読み取るファイルの名前です。入力データは XML としてフォーマット設定される必要があります。
- `expected_result_filename`: これは、実行可能プログラムが出力として送信するファイルの名前です。想定される結果は `data XXX.xml` という形式になります (XXX は数値)。
 - 成功したデータは、例えば次のように送信されます。

```
<ValidationResult result="0" generalFailureMessage="" />
```
 - 失敗したデータは、例えば次のように送信されます。

```
<ValidationResult result="1" generalFailureMessage="">
  <AttributeError attributeName="someAttribute" errorMessage="something" />
  <AttributeError attributeName="someAttribute2" errorMessage="something2" />
</ValidationResult>
```
 - XML ファイル内のテキストは通常の ASCII 文字または UTF-8 でエンコードされる必要があります。

注: 保存操作を再び試行する前にユーザーが問題を修正できるよう、理解しやすいエラー・メッセージを提供することを強くお勧めします。

例

このセクションでは、検証シナリオの例を示します。

キャンペーン編集を防止する

キャンペーンを編集するユーザーによってキャンペーン・コードが変更されるのを防ぐには、カスタム・キャンペーン検証ルーチンを使用できます。キャンペーンを保存するとき、ルーチンによって以下のチェックが確実に行われます。

```
new_campaign_code == old_campaign_code
```

キャンペーンが初めて作成されるケースを扱うために、検証対象のキャンペーンが新規 (作成) または既存 (編集) のどちらであるかを示すフラグをルーチンに渡します。このフラグが `edit` を示している場合、キャンペーン・コードが比較されます。

Campaign アプリケーションは `InputValidationData` オブジェクトの中でこのフラグを設定した後、プラグインにこのオブジェクトを渡します。プラグインは、新規キャンペーンまたはキャンペーン変更のどちらの検証であるかを判別するときに、このフラグを読み取ります。

IBM Unica 技術サポートへの連絡

ドキュメンテーションを参照しても解決できない問題があるなら、指定されているサポート窓口を通じて IBM Unica 技術サポートに電話することができます。このセッションの情報を使用するなら、首尾よく効率的に問題を解決することができます。

サポート窓口が指定されていない場合は、IBM Unica 管理者にお問い合わせください。

収集する必要がある情報

IBM Unica 技術サポートに連絡する前に、以下の情報を収集しておいてください。

- 問題の性質の要旨。
- 問題発生時に表示されるエラー・メッセージの詳細な記録。
- 問題を再現するための詳しい手順。
- 関連するログ・ファイル、セッション・ファイル、構成ファイル、およびデータ・ファイル。
- 「システム情報」の説明に従って入手した製品およびシステム環境についての情報。

システム情報

IBM Unica 技術サポートに電話すると、実際の環境に関する情報について尋ねられることがあります。

問題が発生してもログインは可能である場合、情報の大部分は「バージョン情報」ページで入手できます。そのページには、インストールされている IBM Unica のアプリケーションに関する情報が表示されます。

「バージョン情報」ページは、「ヘルプ」>「バージョン情報」を選択することにより表示できます。「バージョン情報」ページを表示できない場合、どの IBM Unica アプリケーションについても、そのインストール・ディレクトリの下にある `version.txt` ファイルを表示することにより、各アプリケーションのバージョン番号を入手できます。

IBM Unica 技術サポートの連絡先情報

IBM Unica 技術サポートとの連絡を取る方法については、IBM Unica 製品技術サポートの Web サイト (<http://www.unica.com/about/product-technical-support.htm>) を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
170 Tracer Lane
Waltham, MA 02451
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

できます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴ、および ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、『www.ibm.com/legal/copytrade.shtml』をご覧ください。



Printed in Japan