

IBM Campaign
バージョン9 リリース0
2012 年 12 月 11 日

検証 PDK ガイド

IBM

お願い

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 Campaign バージョン 9、リリース 0、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。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM Campaign
Version 9 Release 0
December 11, 2012
Validation PDK Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.12

© Copyright IBM Corporation 1998, 2012.

目次

IBM 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 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	5
validatorConfigString	6
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	8

バリデーターのビルド	9
IBM Campaign の構成	9
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	11
例	12
キャンペーン編集を防止する	12

IBM 技術サポートへの連絡 13

特記事項 15

商標	17
プライバシー・ポリシーおよび利用条件の考慮事項	17

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

IBM® Validation Plug-in Developer's Kit (PDK) を使用すると、IBM 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 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。

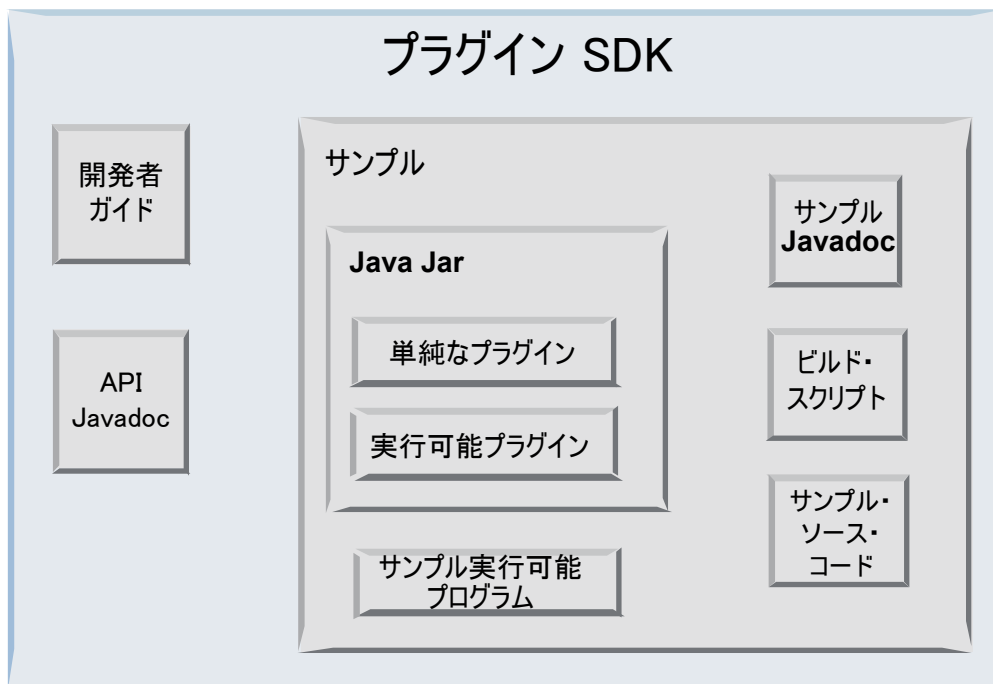
- 2 ページの『検証 PDK の内容』
- 3 ページの『検証 PDK の機能』
- 5 ページの『検証 PDK の構成』
- 7 ページの『プラグインの開発』
- 8 ページの『プラグイン作成の主な手順』
- 11 ページの『実行可能サンプル』
- 12 ページの『例』

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、IBM 技術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 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 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・ライン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 の使用方法を示す文書化された、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・ライン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・ライン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:¥IBM_EMM_Home¥Campaign_Home¥devkits¥validation¥javadoc¥index.html
```

ここで *IBM_EMM_Home* は IBM 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¥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 (カスタム) 属性の検査
- IBM 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)

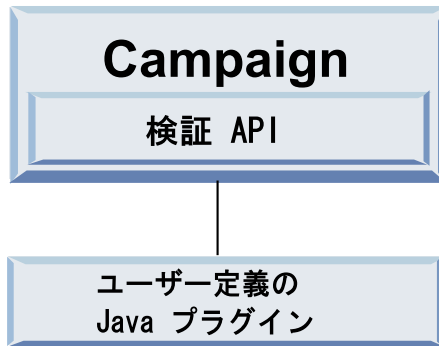
IBM Campaign へのプラグインのロード

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。

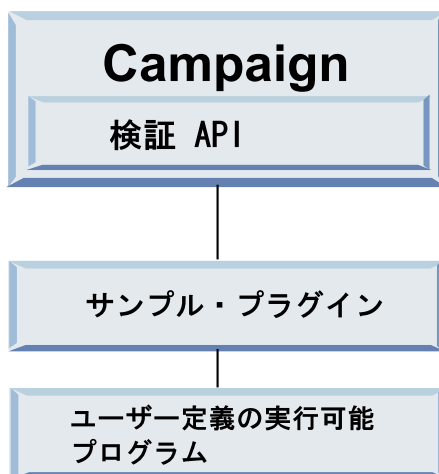
アプリケーションにロードされる Java クラス・プラグインの作成

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 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 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。検証プラグインを定義すると、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、その検証プラグインが IBM Campaign によって自動的に呼び出されます。IBM 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 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場合)

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 は、構成パラメーターを使用することにより、Campaign に対して使用するべきプラグイン・クラスを見つける方法を示します。また、これらのプラグインにいくつかの構成情報を渡すための手段も提供します。

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使用できます。つまり、partition[n] を任意のパーティション名に変更して、複数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』
- 『validationClasspath』
- 6 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。

プロパティ	説明
詳細	<p>絶対パスまたは相対パスを使用します。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>c:%bea%user_projects%domains%mydomain</pre> <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・クラスの位置を指しています。</p> <p>パスの末尾がスラッシュでない場合、Campaign では、下の例に示されているように Java クラスを含む .jar ファイルの名前と見なされます。</p> <p>パスが設定されていない場合、Campaign はプラグインのロードを試行しません。</p>
例	<pre>/<CAMPAIGN_HOME>/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><property name="validationClasspath" /></pre>
関連項目	<p>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。</p>

validatorConfigString

プロパティ	説明
説明	<p>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。</p>
詳細	<p>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。</p> <p>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。</p>

プロパティ	説明
例	<p>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、</p> <p>validatorConfigString を次のように設定します。</p> <pre>/opt/unica/campaign/devkits/validation/ src/com/unica/campaign /core/validation/samples/validate.sh</pre>
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><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。このプラグインには、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 (Apache Ant は Java ベースのビルド・ツール) が含まれます。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

Campaign 標準インストールには、2 つのサンプル・バリデーターが含まれています。

- SimpleCampaignValidator は、カスタム許可や許容されるキャンペーン名の検証などの方法を示す、自己完結型のプラグインです。これは、以下のパスにあります。

```
devkits¥validation¥src¥com¥unica¥campaign¥core¥validation¥
samples¥SimpleCampaignValidator.Java
```

このプラグインを使用するときは、直接編集するのではなく、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、このサンプルの本当の目的は、検証用のコマンド・ライン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、およびその他のアジャイル手法は広く使われており、役立っています。これらの手法の重要な一面は、これらがテスト中心であることです。これらの手法を使用するお客様は、単体テストを広範囲に使用します。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、これらの手法をサポートします。IBM 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 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。

- 『セットアップ』
- 9 ページの『バリデーターのビルド』
- 9 ページの『IBM 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 ファイルをビルド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 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。

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。2 つの例を示します。
 - Windows 上のデフォルト・ディレクトリーに WebLogic 10gR3 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:¥Oracle¥Middleware¥wls_server_10.3¥common¥bin
 - Windows 上のデフォルト・ディレクトリーに WebSphere® 7.0 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:¥IBM¥WebSphere¥AppServer1¥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。2 つの例を示します。
 - Windows 上の WebLogic 11gR1 の場合、JAVA_HOME を
C:¥Oracle¥Middleware¥jdk160_18 に設定します。
 - Windows 上の WebSphere 7.0 の場合、JAVA_HOME を
C:¥IBM¥WebSphere¥AppServer1¥java¥jre に設定します。

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、実稼働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。

```
<IBM_EMM_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・ラインで 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。

```
<IBM_EMM_Home¥Campaign_Home>¥devkits¥validation¥build¥lib
```

これで、IBM 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 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 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 ファイルをビルドし、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 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、その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制します。

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。

```
<IBM_EMM_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ビルドします。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、バリデーターをビルド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・ライン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 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`」を持つすべての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・ライン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・ライン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 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。このルーチンは、キャンペーンを保存するときに以下の検査が行うことによって、キャンペーンを編集するユーザーによるキャンペーン・コードの変更が行われないようにします。

```
new_campaign_code == old_campaign_code
```

キャンペーンを初めて作成する場合に対処するため、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 **編集** 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` オブジェクトに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 技術サポートへの連絡

文書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 技術サポートに電話することができます。このセク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 管理者にお問い合わせください。

収集する情報

IBM 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関する情報。

システム情報

IBM 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 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 アプリケーションについても、そのインストール・ディレクトリーの下にある `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 技術サポートのコンタクト情報

IBM 技術サポートとの連絡を取る方法については、IBM 製品技術サポートの Web サイト (http://www-947.ibm.com/support/entry/portal/open_service_request) 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 Corporation 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。

プライバシー・ポリシーおよび利用条件の考慮事項

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品（「ソフトウェア・オファリング」）では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。Cookie とは Web サイトからお客様のブラウザに送信できるデータで、お客様のコンピューターを識別するタグとしてそのコンピューターに保存されることがあります。多くの場合、これらの Cookie により個人情報が収集されることはありません。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的な事項を確認ください。

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理、お客様の利便性の向上、または利用の追跡または機能上の目的のために、それぞれのお客様のユーザー名、およびその他の個人情報を、セッションごとの Cookie および持続的な Cookie を使用して収集する場合があります。これらの Cookie は無効にできますが、その場合、これらを有効にした場合の機能を活用することはできません。

Cookie およびこれに類するテクノロジーによる個人情報の収集は、各国の適用法令等による制限を受けます。この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人情報を収集する機能を提供する場合、お客様は、個人情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意取得の要求も含まれますがそれらには限られません。

お客様は、IBM の使用にあたり、(1) IBM およびお客様のデータ収集と使用に関する方針へのリンクを含む、お客様の Web サイト利用条件（例えば、プライバシー・ポリシー）への明確なリンクを提供すること、(2) IBM がお客様に代わり閲覧者のコンピューターに、Cookie およびクリア GIF または Web ビーコンを配置することを通知すること、ならびにこれらのテクノロジーの目的について説明すること、

および (3) 法律で求められる範囲において、お客様または IBM が Web サイトへの閲覧者の装置に Cookie およびクリア GIF または Web ビーコンを配置する前に、閲覧者から合意を取り付けること、とします。

このような目的での Cookie を含むさまざまなテクノロジーの使用については、IBM の『IBM オンラインでのプライバシー・ステートメント』(<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』を参照してください。



Printed in Japan