

Version 10 Release 0
Juni 2016

*IBM Campaign
Handbuch Validierungs-PDK*

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 17 gelesen werden.

Diese Ausgabe bezieht sich auf Version 10, Release 0, Modifikation 0 von IBM Campaign und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage geändert wird.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM Campaign Validation PDK Guide, Version 10 Release 0,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1998, 2016

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
Juni 2016

© Copyright IBM Corporation 1998, 2016.

Inhaltsverzeichnis

Kapitel 1. Übersicht IBM Developer's Kit für das Validierungs-Plug-in (PDK) . . . 1

Inhalt des Validierungs-PDK	1
Zwei Möglichkeiten für die Verwendung der Validierungs-API	2
Erstellen eines Java-Klasse-Plug-ins, das in die Anwendung geladen wird	3
Aufrufen einer Anwendung zur Ausführung der Validierung	3
Validierung von Angeboten/Kampagnen	3
Beispielvalidatoren im Validierungs-PDK	4
Testharness für das Validierungs-PDK	4
Erstellungsscripts für das Validierungs-PDK	5

Kapitel 2. Entwickeln von Validierungs-Plug-ins für Campaign 7

Konfiguration der Umgebung für die Verwendung des Validierungs-PDK	7
Erstellen des Validators	8
Konfigurieren von Campaign für die Verwendung eines Validierungs-Plug-ins	8
validationClass	9

validationClasspath	9
validatorConfigString	10
Testen der Validator Konfiguration	11
Erstellen eines Validators	11
Beispielszenario für eine Validierung: Kampagnenbearbeitung verhindern	12

Kapitel 3. Aufruf einer Anwendung zur Durchführung der Validierung 13

Konfigurieren von Campaign für die Verwendung des ausführbaren Beispiel-Plug-ins	13
Erwartete Benutzeroberfläche für ausführbare Funktionen	13

Bevor Sie sich an den technischen Support von IBM wenden. 15

Bemerkungen. 17

Marken.	19
Hinweise zu Datenschutzrichtlinien und Nutzungsbedingungen	19

Kapitel 1. Übersicht IBM Developer's Kit für das Validierungs-Plug-in (PDK)

Mit dem IBM® Developer's Kit für das Validierungs-Plug-in (PDK) können Sie eine benutzerdefinierte Validierungslogik für die Verwendung in IBM Campaign entwickeln.

Sie können Plug-ins erstellen, um eine benutzerdefinierte Validierungslogik für Kampagnen und/oder Angebote auszuführen.

Mögliche Einsatzbereiche der Validierungslogik sind zum Beispiel:

- Überprüfung von erweiterten (benutzerdefinierten) Attributen
- Bereitstellung von Autorisierungsservices, die nicht von IBM Marketing Platform angeboten werden (zum Beispiel die Validierung, welche Benutzer welche erweiterten Attribute bearbeiten dürfen).

Das Validierungs-PDK ist eine Unterklasse der von IBM Campaign bereitgestellten allgemeinen Plug-ins.

Das Validierungs-PDK enthält Javadoc-Referenzinformationen sowohl für die Plug-in-API als auch für den Beispielcode. Öffnen Sie die folgende Datei in Ihrem Web-Browser, um die Dokumentation anzuzeigen:

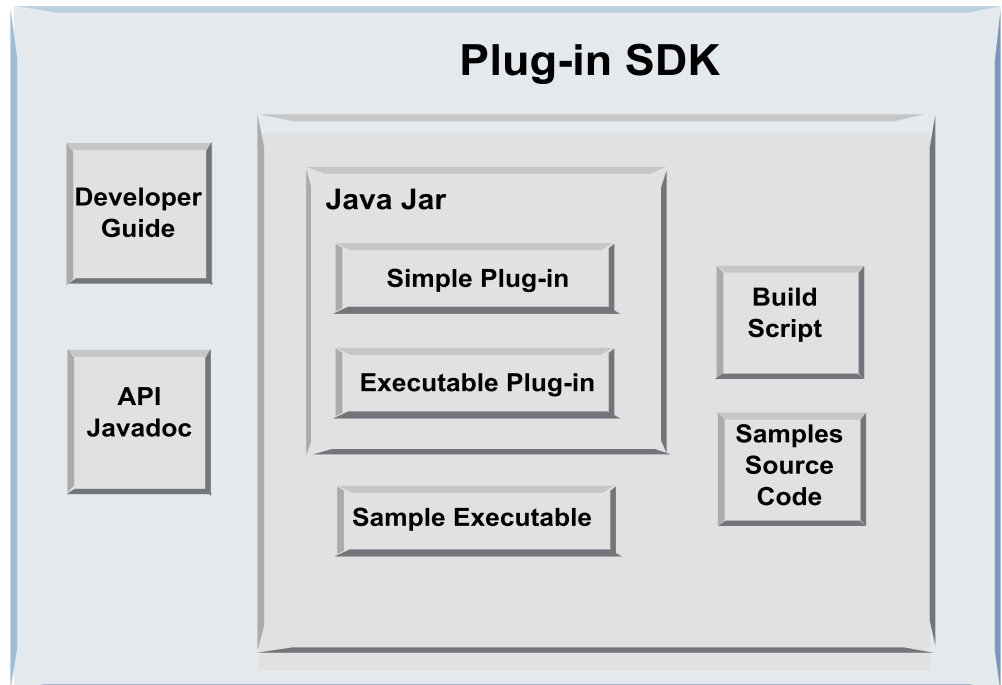
```
C:\IBM\IMS\Campaign_Home\devkits\validation\javadoc\index.html
```

Beispiel:

```
C:\IBM\IMS\Campaign\devkits\validation\javadoc\index.html
```

Inhalt des Validierungs-PDK

Das Validierungs-PDK beinhaltet Komponenten für die Entwicklung von Java™ - Plug-ins oder über die Befehlszeile ausführbaren Funktionen, um eine benutzerdefinierte Validierung in IBM Campaign hinzuzufügen. Das PDK enthält dokumentierte Beispiele für die Verwendung des PDK, auf denen Sie aufbauen können.



Die einzelnen Komponenten werden in der folgenden Tabelle beschrieben.

Tabelle 1. Komponenten des Validierungs-PDK

Komponente	Beschreibung
Handbuch für Entwickler	PDF-Dokument mit dem Titel <i>IBM Campaign - Handbuch für das Validierungs-PDK</i> .
API-Javadoc	Referenzinformationen für die Plug-in-API.
Java-JAR-Datei	JAR-Beispieldatei mit Beispiel-Plug-ins. Die JAR-Datei enthält: <ul style="list-style-type: none"> Einfaches Plug-in: Beispiel einer eigenständigen Validator-Klasse. Ausführbares Plug-in: Beispiel-Validator, der für die Validierung eine benutzerdefinierte ausführbare Funktion ausführt, die über die Befehlszeile aufgerufen werden kann.
Beispiel einer ausführbaren Funktion	Über die Befehlszeile ausführbare Funktion, die zusammen mit dem ausführbaren Plug-in unter UNIX verwendet werden kann.
Erstellungsscript	Ant-Script, das aus dem enthaltenen Quellcode verwendbare Validator-Plug-ins erstellt.
Beispielquellcode	Java-Quellcode für den einfachen Validator und den ausführbaren Validator.

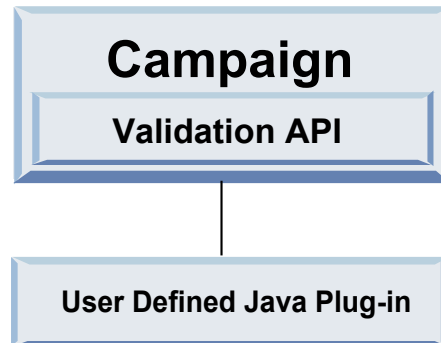
Zwei Möglichkeiten für die Verwendung der Validierungs-API

Es gibt zwei Möglichkeiten für die Verwendung der Validierungs-API.

- Verwendung der API für die Erstellung eines Java-Klasse-Plug-ins, das in die Anwendung geladen wird.
- Verwendung eines der enthaltenen Plug-ins für den Aufruf einer ausführbaren Anwendung zur Durchführung der Validierung.

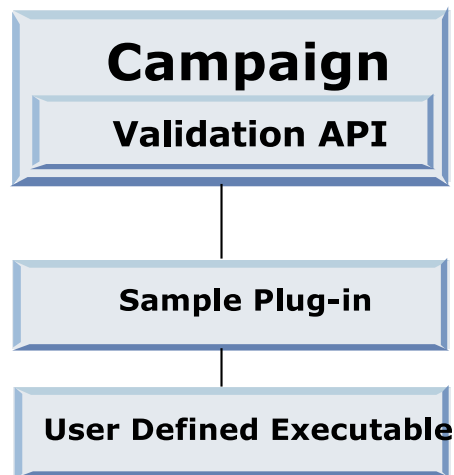
Erstellen eines Java-Klasse-Plug-ins, das in die Anwendung geladen wird

Das Validierungs-PDK bietet die Benutzeroberflächen, Helper-Klassen und Entwicklertools für die Entwicklung dieser Klassen.



Aufrufen einer Anwendung zur Ausführung der Validierung

Sie können eines der enthaltenen Plug-ins des Validierungs-PDK verwenden, um eine ausführbare Anwendung aufzurufen, mit der die Validierung durchgeführt werden soll.



Die ausführbare Funktion kann in einer beliebigen Sprache geschrieben sein, muss sich aber auf dem IBM Campaign-Server befinden und dort ausgeführt werden. Das Plug-in, mit dem die ausführbare Funktion aufgerufen wird, sendet eine XML-Datei mit den zu validierenden Informationen. Das sind zum Beispiel der Bearbeiter des Objekts oder die Vorher-/Nachher-Werte für alle Standardattribute und erweiterten Attribute des betreffenden Objekts. IBM Campaign erwartet die Rückgabe von Ergebnisinformationen in Form einer XML-Datei.

Validierung von Angeboten/Kampagnen

Ein mit dem Validierungs-PDK von Campaign erstelltes Plug-in kann eine benutzerdefinierte Validierungslogik für Kampagnen und/oder Angebote ausführen.

Das Validierungs-PDK wird zur Validierung von Angeboten und Kampagnen verwendet. Wenn ein Validierungs-Plug-in definiert wurde, wird es bei jedem Spei-

chern eines Angebots- oder Kampagnenobjekts automatisch durch IBM Campaign aufgerufen. IBM Campaign setzt ein Flag, wenn es die Validierungsmethode des Plug-ins aufruft. IBM Campaign übergibt die folgenden Flags:

- `ValidationInputData.CAMPAIGN_VALIDATION` beim Hinzufügen oder Ändern einer Kampagne
oder
- `ValidationInputData.OFFER_VALIDATION` beim Hinzufügen oder Bearbeiten eines Angebots.

Mit diesen Flags können Validierungsregeln für Angebote und Kampagnen aufgestellt werden.

Beispielvalidatoren im Validierungs-PDK

Das Validierungs-PDK von Campaign enthält die folgenden beiden Beispielvalidatoren: `SimpleCampaignValidator` und `ExecutableCampaignValidator`.

- `SimpleCampaignValidator` ist ein eigenständiges Plug-in beispielsweise für die benutzerdefinierte Autorisierung oder für die Validierung zulässiger Kampagnennamen. Es befindet sich im folgenden Pfad:

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

Es wird empfohlen, vor der Bearbeitung der Klasse eine Kopie anzulegen, damit die ursprüngliche Version bei Bedarf zur Verfügung steht.

- `ExecutableCampaignValidator` ist ein Java-Plug-in, das eine ausführbare Anwendung aufruft, um die Validierung auszuführen. Der Quellcode für `ExecutableCampaignValidator` befindet sich im selben Verzeichnis wie der Quellcode von `SimpleCampaignValidator`:

```
devkits\validation\src\com\unica\campaign\core\validation\  
samples\ExecutableCampaignValidator.java
```

Der eigentliche Zweck dieses Beispiels ist jedoch die Verwendung als in der Befehlszeile aufrufbare Validierungsfunktion. Diese Datei befindet sich in folgendem Pfad:

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

Bei dieser Datei handelt es sich um ein Beispiel für eine ausführbare Loopback-datei, mit der häufige Validierungsaktionen demonstriert werden.

Testharness für das Validierungs-PDK

Durch die Möglichkeit, den Gültigkeitscode testen zu können, ohne dass dieser in IBM Campaign geladen werden muss, lässt sich der Entwicklungsprozess für das Plug-in beschleunigen.

Kunden, die Extreme Programming und andere agile Methodiken nutzen, führen in der Regel sehr viele Komponententests durch. Das Validierungs-PDK unterstützt diese Methodiken mit dem bereitgestellten Testharness für die Ausführung eines Plug-ins außerhalb von Campaign.

Vorgehensweise für die Anwendung des Testharness:

1. Ändern Sie den Komponententest so ab, dass die Validierungslogik im Plug-in berücksichtigt wird.
2. Führen Sie das Erstellungsscript aus:

- Wenn Sie das Plug-in ohne Komponententests erstellen möchten, führen Sie die Erstellungsscripts mit dem Befehl "ant jar" aus.
- Wenn Sie das Plug-in erstellen und auch Komponententests durchführen möchten, führen Sie die Erstellungsscripts mit dem Befehl "ant run-test" aus.

Erstellungsscripts für das Validierungs-PDK

Mit den Erstellungsscripts im Validierungs-PDK werden alle Klassen in einem Verzeichnis kompiliert und in einer JAR-Datei abgelegt, die sich zur Verwendung in IBM Campaign eignet.

Das bereitgestellte Erstellungsscript befindet sich in folgendem Verzeichnis:

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

Kapitel 2. Entwickeln von Validierungs-Plug-ins für Campaign

Ein Plug-in ist eine Java-Klasse, die beim Start geladen und bei jeder Validierung einer Kampagne oder eines Angebots aufgerufen wird.

Die Validierung erfolgt jedes Mal, wenn ein Benutzer eine Kampagne speichert. Sie können Ihre eigenen Java-Plug-ins mit den Tools erstellen, die mit dem Validierungs-PDK bereitgestellt werden. Das PDK enthält Quellcode für Beispiel-Plug-ins und eine Ant-Datei (Apache Ant ist ein Erstellungstool auf Java-Basis) für die Kompilierung von Plug-ins.

In den folgenden Schritten wird erläutert, wie Sie Ihre Umgebung für die Entwicklung eines Plug-ins einrichten können. Anschließend werden Sie durch die Erstellung eines eigenen Plug-ins geführt.

1. „Konfiguration der Umgebung für die Verwendung des Validierungs-PDK“
2. „Erstellen des Validators“ auf Seite 8
3. „Konfigurieren von Campaign für die Verwendung eines Validierungs-Plug-ins“ auf Seite 8
4. „Testen der Validator-Konfiguration“ auf Seite 11
5. „Erstellen eines Validators“ auf Seite 11

Konfiguration der Umgebung für die Verwendung des Validierungs-PDK

Damit Sie das Validierungs-PDK mit Campaign verwenden können, müssen Sie Ihren Pfad ändern und die Umgebungsvariable `JAVA_HOME` festlegen.

Das Validierungs-PDK kann in jedem beliebigen System installiert werden, die damit erstellten Plug-ins müssen sich jedoch in dem System befinden, in dem IBM Campaign ausgeführt wird. Es wird empfohlen, das PDK in dem System zu installieren, in dem Sie Ihre Plug-ins testen.

Damit das PDK Java-Plug-ins erstellen kann, müssen in dem System Apache Ant und ein Sun Java Developer Kit installiert sein. Um die Kompatibilität sicherzustellen, sollten Sie die Ant- und JDK-Pakete verwenden, die mit Ihrem Anwendungsserver ausgeliefert wurden.

Konfiguration Ihrer Umgebung für das Validierungs-PDK:

1. Stellen Sie den Ordner mit der ausführbaren Ant-Datei in Ihren Pfad. Beispiele:
 - Bei einer Installation von WebLogic 11gR1 im Standardverzeichnis unter Windows fügen Sie Ihrem Pfad Folgendes hinzu: `C:\Oracle\Middleware\wlserver_10.3\common\bin`
 - Bei einer Installation von WebSphere 7.0 im Standardverzeichnis unter Windows fügen Sie Ihrem Pfad Folgendes hinzu: `C:\IBM\WebSphere\AppServer1\bin`.
2. Geben Sie für die Umgebungsvariable `JAVA_HOME` das Verzeichnis mit den `bin`- und `lib`-Verzeichnissen Ihres JDK an. Beispiele:
 - Bei WebLogic 11gR1 unter Windows legen Sie `JAVA_HOME` auf `C:\Oracle\Middleware\jdk160_18` fest.

- Bei WebSphere 7.0 unter Windows legen Sie JAVA_HOME auf C:\IBM\WebSphere\AppServer1\java\jre fest.

Erstellen des Validators

Das IBM Campaign-Validierungs-PDK stellt ein Ant-Script bereit, mit dem sich der gesamte Code in den Beispieldateien erstellen lässt.

Standardverhalten des Scripts ist die Erstellung einer JAR-Datei mit den Validierungsklassen. Optional kann Javadocs erstellt und Tests der Validatoren ausgeführt werden, um vor der Verwendung des Plug-ins in der Produktionsumgebung sicherzustellen, dass die Validatoren in Campaign funktionsfähig sind.

Vorgehensweise zum Erstellen des Validators:

1. Wechseln Sie in das PDK-Verzeichnis <IBM_IMS_Home\Campaign_Home>\devkits\validation\build
Beispiel: C:\IBM\IMS\Campaign\devkits\validation\build
Dieses Verzeichnis enthält das Ant-Script build.xml.
2. Führen Sie die Ant-JAR-Datei in der Befehlszeile aus.
 - Zur Erstellung des Plug-ins ohne Komponententests verwenden Sie den Befehl ant jar.
 - Zur Erstellung des Plug-ins mit Komponententests verwenden Sie den Befehl ant run-test.

Das Ant-Script wird ausgeführt und erstellt eine JAR-Datei mit dem Namen validator.jar im Unterverzeichnis lib. Beispiel:

C:\IBM\IMS\Campaign\devkits\validation\build\lib

Sie haben nun einen benutzerdefinierten Validator für die Verwendung in IBM Campaign erstellt. Im nächsten Schritt müssen Sie Campaign für die Verwendung dieses Validators konfigurieren.

Konfigurieren von Campaign für die Verwendung eines Validierungs-Plug-ins

Für die Konfiguration von Campaign zur Verwendung eines Validierungs-Plug-ins müssen Sie die Konfigurationseinstellungen unter Campaign > partitions > partition[n] > validation verwenden.

Mit den Konfigurationseigenschaften wird Campaign informiert, wie die Plug-in-Klasse gefunden werden kann, und sie bieten eine Möglichkeit zur Übergabe der Konfigurationsinformationen an die Plug-ins.

Anmerkung: Die Validierung kann mit verschiedenen Partitionen arbeiten; partition[n] kann auf einen beliebigen Partitionsnamen geändert werden, um auch für diese Partitionen Validierungsroutinen bereitzustellen.

Sie können die folgenden Konfigurationseinstellungen für die Validierung anpassen:

- „validationClass“ auf Seite 9
- „validationClasspath“ auf Seite 9
- „validatorConfigString“ auf Seite 10

Legen Sie die Eigenschaften zur Verwendung von SimpleCampaignValidator wie folgt fest:

- validationClasspath: *Unica*\campaign\devkits\validation\lib\validator.jar
- validationClass:
com.unica.campaign.core.validation.samples.SimpleCampaignValidator
- validatorConfigString muss für die Verwendung von SimpleCampaignValidator nicht festgelegt werden, da dieser keine Konfigurationszeichenfolge verwendet.

Legen Sie die Eigenschaften zur Verwendung von ExecutableCampaignValidator wie folgt fest:

- validationClasspath: <Campaign_home>\devkits\validation\lib\validator.jar
- validationClass:
com.unica.campaign.core.validation.samples.ExecutableCampaignValidator
- validatorConfigString: <Campaign_home>\pdk\bin\validate.sh

validationClass

Mit der Eigenschaft validationClass wird Campaign der Name der Klasse übermittelt, die für die Validierung mit einem Plug-in des Validierungs-PDK verwendet werden muss.

Eigenschaft	Beschreibung
Beschreibung	Name der Klasse, die für die Validierung verwendet werden soll. Der Wert der Eigenschaft validationClasspath gibt die Position dieser Klasse an.
Details	Die Klasse muss mit dem Paketnamen vollständig qualifiziert sein. Wenn diese Eigenschaft nicht festgelegt ist, führt Campaign keine benutzerdefinierte Validierung durch.
Beispiel	com.unica.campaign.core.validation.samples.SimpleCampaignValidator In diesem Beispiel wird validationClass auf die Klasse SimpleCampaignValidator aus dem Beispielcode gesetzt.
Standard	Standardmäßig wird kein Pfad festgelegt: <property name="validationClass" />

validationClasspath

Mit der Eigenschaft validationClasspath wird Campaign die Position der Klasse übermittelt, die für die Validierung mit einem Plug-in des Validierungs-PDK verwendet werden muss.

Eigenschaft	Beschreibung
Beschreibung	Pfad der Klasse, die für die benutzerdefinierte Validierung verwendet wird.

Eigenschaft	Beschreibung
Details	<p>Verwenden Sie entweder einen vollständigen Pfad oder einen relativen Pfad. Bei einem relativen Pfad ist das Verhalten von dem Anwendungsserver abhängig, auf dem Campaign ausgeführt wird. WebLogic verwendet den Pfad zum Domänenarbeitsverzeichnis, das Standardverzeichnis ist:</p> <p><code>c:\bea\user_projects\domains\mydomain.</code></p> <p>Wenn der Pfad mit einem Schrägstrich endet (normaler Schrägstrich (/) bei UNIX bzw. umgekehrter Schrägstrich (\) bei Windows), geht Campaign davon aus, dass er auf die Position der zu verwendenden Java-Plug-in-Klasse zeigt.</p> <p>Wenn der Pfad nicht mit einem Schrägstrich endet, geht Campaign davon aus, dass dies der Name einer JAR-Datei ist, welche die Java-Klasse enthält (siehe folgendes Beispiel).</p> <p>Wenn kein Pfad festgelegt ist, versucht Campaign nicht, ein Plug-in zu laden.</p>
Beispiel	<p><code>/<CAMPAIGN_HOME>/devkits/validation/lib/validator.jar</code></p> <p>Dies ist auf einer UNIX-Plattform der Pfad zu der JAR-Datei, die im Paket des Developer's Kit für Plug-ins enthalten ist.</p>
Standard	<p>Standardmäßig wird kein Pfad festgelegt:</p> <pre><property name="validationClasspath" /></pre>
Siehe auch	<p>Siehe „validationClass“ auf Seite 9; hier finden Sie Informationen zur Benennung der zu verwendenden Klasse.</p>

validatorConfigString

Die Eigenschaft `validatorConfigString` wird an das Validator-Plug-in übergeben, wenn es von Campaign geladen wird.

Eigenschaft	Beschreibung
Beschreibung	<p>Zeichenfolge, die an das Validator-Plug-in übergeben wird, wenn es von Campaign geladen wird.</p>
Details	<p>Wie das Plug-in diese Zeichenfolge verwendet, wird durch den Entwickler festgelegt. Sie können zum Beispiel festlegen, dass eine Zeichenfolge an Ihr Plug-in übergeben wird, wenn es vom System geladen wird.</p> <p>Zum Beispiel verwendet der <code>ExecutableCampaignValidator</code> (im PDK enthaltenes Beispiel-Plug-in für eine ausführbare Funktion) diese Eigenschaft, um die ausführbare Funktion anzugeben, die aufgerufen werden soll.</p>
Beispiel	<p>Um das Beispiel-Bourne-Shell-Script als Validierungsscript auszuführen, legen Sie</p> <pre>validatorConfigString auf</pre> <pre>/opt/unica/campaign/devkits/validation/src/com/unica/campaign/core/validation/samples/validate.sh fest.</pre>
Standard	<p>Standardmäßig wird kein Pfad festgelegt:</p> <pre><property name="validatorConfigString" /></pre>

Testen der Validatorkonfiguration

Nach Erstellen der Datei `validator.jar`, welche die Klasse `SimpleCampaignValidator` enthält, und nach den erforderlichen Konfigurationsänderungen können Sie das Plug-in testen und verwenden.

Mit dem folgenden Beispiel-Plug-in wird verhindert, dass Benutzer von Campaign eine Kampagne namens "badCampaign" speichern.

Vorgehensweise zum Testen Ihrer Konfiguration:

1. Implementieren Sie Ihren Anwendungsserver erneut, damit die Änderungen wirksam werden. Sie finden entsprechende Anweisungen in der Dokumentation Ihres Servers.
2. Melden Sie sich bei IBM Campaign an und rufen Sie die Seite für die Kampagnenerstellung auf.
3. Erstellen Sie eine Kampagne mit dem Namen **badCampaign** und versuchen Sie, diese zu speichern.

Wenn alles richtig konfiguriert ist, können Sie die neue Kampagne nicht speichern. Die Konfiguration funktioniert ordnungsgemäß, wenn Sie eine Fehlermeldung vom Validator erhalten.

Erstellen eines Validators

Befolgen Sie diese Anweisungen, um ein Validierungs-Plug-in ähnlich `SimpleCampaignValidator` zu erstellen, mit dem die Erstellung von Kampagnen namens "badCampaign2" verhindert wird.

1. Erstellen Sie eine Kopie des Beispielsvalidators `SimpleCampaignValidator.java` in `<IBM_IMS_Home\Campaign_Home>\devkits\validation\src\com\unica\campaign\core\validation\samples`. Geben Sie der Kopie den Namen `MyCampaignValidator.java` und belassen Sie sie im selben Verzeichnis wie die Quelle. Beispiel:
`C:\IBM\IMS\Campaign\devkits\validation\src\com\unica\campaign\core\validation\samples\MyCampaignValidator.java`
2. Öffnen Sie `MyCampaignValidator.java` in einem Editor. Suchen Sie das Wort "badCampaign" im Dokument und ersetzen Sie es durch das Wort "badCampaign2".
3. Speichern Sie die Datei und schließen Sie den Editor.
4. Erstellen Sie die Validatoren erneut. Weitere Informationen finden Sie unter „Erstellen des Validators“ auf Seite 8. Wenn Ihr Anwendungsserver die Datei `validate.jar` sperrt, während sie verwendet wird, müssen Sie den Server vor der Erstellung der Validatoren stoppen.
5. Rekonfigurieren Sie `campaign_config.xml` für die Verwendung Ihrer neuen Klasse: `<property name="validationClass" value="com.unica.campaign.core.validation.samples.MyCampaignValidator">`
6. Testen Sie den Validator. Weitere Informationen finden Sie im Abschnitt „Testen der Validatorkonfiguration“.

Vergewissern Sie sich, dass der Validator funktioniert: Es darf nicht möglich sein, Kampagnen mit dem Namen "badCampaign2" zu speichern.

Beispielszenario für eine Validierung: Kampagnenbearbeitung verhindern

In diesem Beispiel wird erläutert, wie eine bestimmte Bearbeitung einer Kampagne mit der Validierung verhindert werden kann.

Wenn Sie verhindern möchten, dass ein Bearbeiter den Kampagnencode ändert, können Sie eine benutzerdefinierte Routine für die Kampagnenvalidierung verwenden. Durch die Routine wird beim Speichern der Kampagne die folgende Prüfung vorgenommen:

```
neuer_Kampagnencode == alter_Kampagnencode
```

Damit auch neu erstellte Kampagnen berücksichtigt werden, müssen Sie ein Flag an die Routine übergeben, das anzeigt, ob die zu validierende Kampagne neu ist (creation) oder ob sie bereits existiert (edit). Zeigt dieses Flag **edit** an, werden die Kampagnencodes verglichen.

Die Campaign-Anwendung setzt dieses Flag im Objekt `InputValidationData`, von dem es dann an das Plug-in übergeben wird. Das Plug-in liest das Flag bei der Ermittlung, ob die Validierung für eine neue oder für eine geänderte Kampagne vorgenommen werden soll.

Kapitel 3. Aufruf einer Anwendung zur Durchführung der Validierung

Das Validierungs-PDK enthält den Beispielvalidator `ExecutableCampaignValidator`. Dieser führt für die Validierung über die Befehlszeile die ausführbare Datei `validate.sh` aus.

In den folgenden Abschnitten werden diese Vorgänge erläutert:

- Konfiguration von Campaign für die Ausführung des ausführbaren Beispiel-Plug-ins und
- Erstellung Ihres eigenen ausführbaren Plug-ins, das sich für die Verwendung der Benutzeroberfläche für ausführbare Funktionen eignet.

Konfigurieren von Campaign für die Verwendung des ausführbaren Beispiel-Plug-ins

Damit `ExecutableCampaignValidator` verwendet werden kann, müssen Sie die Konfigurationseinstellungen unter `Campaign > partitions > partition[n] > validation` entsprechend anpassen.

Legen Sie die Eigenschaften wie folgt fest:

- `validationClasspath`:
`<Campaign_home>\devkits\validation\lib\validator.jar`
- `validationClass`:
`com.unica.campaign.core.validation.samples.ExecutableCampaignValidator`
- `validatorConfigString`:
`<Campaign_home>\pdk\bin\validate.sh`

Das mit dem Validierungs-PDK bereitgestellte Beispielscript ist ein Bourne-Shell-Script für UNIX. Damit wird allen Benutzern mit dem Benutzernamen "badUser" die Erstellung von Kampagnen verweigert. Sie können den Code für diese ausführbare Funktion im folgenden Verzeichnis anzeigen:

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

Sie müssen Ihr eigenes Script entwickeln, das die Ihrer Implementierung entsprechende Validierung durchführt. Für Textverarbeitungsscripts wie dieses eignen sich scriptbasierte Sprachen wie PERL und Python gut, wobei aber alle Sprachen akzeptabel sind, die über die Befehlszeile ausgeführt werden können.

Erwartete Benutzeroberfläche für ausführbare Funktionen

Das Plug-in `ExecutableCampaignValidator` ruft über eine Befehlszeile mit den unten genannten Argumenten eine ausführbare Datei auf.

- `executable_name`: Zeichenfolge, die unter `validatorConfigString` in IBM Marketing Platform festgelegt ist.
- `data_filename`: Name der Datei, die von der ausführbaren Funktion als Eingabe gelesen wird. Die Eingabedaten müssen in XML formatiert sein.

- `expected_result_filename`: Name der Datei, die von der ausführbaren Funktion als Ausgabe gesendet werden soll. Die erwarteten Ergebnisse haben die Form Daten XXX.xml, wobei XXX eine Zahl ist.
 - Beispiel für erfolgreich gesendete Daten:

```
<ValidationResult result="0" generalFailureMessage="" />
```
 - Beispiel für nicht erfolgreich gesendete Daten:

```
<ValidationResult result="1" generalFailureMessage="">  
  <AttributeError attributeName="someAttribute" errorMessage="something" />  
  <AttributeError attributeName="someAttribute2" errorMessage="something2" />  
</ValidationResult>
```
 - Der Text in der XML-Datei muss mit normalen ASCII-Zeichen oder mit UTF-8 codiert sein.

Anmerkung: Es wird dringend empfohlen, leicht verständliche Fehlernachrichten zu definieren, damit Benutzer das Problem beheben können, bevor sie erneut versuchen, eine Sicherungsoperation durchzuführen.

Bevor Sie sich an den technischen Support von IBM wenden

Sollte sich ein Problem nicht mithilfe der Dokumentation beheben lassen, können sich die für den Support zuständigen Kontaktpersonen Ihres Unternehmens telefonisch an den technischen Support von IBM wenden. Verwenden Sie diese Anleitungen, damit das Problem effizient und erfolgreich gelöst wird.

Wenn Sie wissen möchten, wer die für den Support zuständige Kontaktperson Ihres Unternehmens ist, wenden Sie sich an Ihren IBM Administrator.

Anmerkung: Der technische Support schreibt bzw. erstellt keine API-Scripts. Wenden Sie sich zur Unterstützung bei der Implementierung unserer API-Angebote an IBM Professional Services.

Zusammenzustellende Informationen

Halten Sie folgende Informationen bereit, wenn Sie sich an den technischen Support von IBM wenden:

- Kurze Beschreibung der Art Ihres Problems
- Detaillierte Fehlermeldungen, die beim Auftreten des Problems angezeigt werden.
- Schritte zum Reproduzieren des Problems
- Entsprechende Protokolldateien, Sitzungsdateien, Konfigurationsdateien und Daten
- Informationen zu Ihrer -Produkt- und -Systemumgebung, die Sie entsprechend der Beschreibung unter "Systeminformationen" abrufen können.

Systeminformationen

Bei Ihrem Anruf beim technischen Support von IBM werden Sie um verschiedene Informationen gebeten.

Sofern das Problem Sie nicht an der Anmeldung hindert, finden Sie einen Großteil der benötigten Daten auf der Info-Seite. Dort erhalten Sie Informationen zur installierten IBM Anwendung.

Sie können über **Hilfe > Info** (Help > About) auf die Info-Seite zugreifen. Wenn Sie nicht auf die Info-Seite zugreifen können, prüfen Sie, ob im Installationsverzeichnis Ihrer Anwendung die Datei `version.txt` vorhanden ist.

Kontaktinformationen für den technischen Support von IBM

Wenn Sie sich an den technischen Support von IBM wenden möchten, finden Sie weitere Informationen auf der Website des technischen Supports für IBM Produkte (http://www.ibm.com/support/entry/portal/open_service_request).

Anmerkung: Um eine Supportanforderung einzugeben, müssen Sie sich mit einem IBM Account anmelden. Dieser Account muss mit Ihrer IBM Kundennummer verknüpft sein. Weitere Informationen zum Zuordnen Ihres Accounts zu Ihrer IBM Kundennummer erhalten Sie unter **Unterstützungsressourcen > Gültige Softwareunterstützung** im Support Portal.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für die in diesem Handbuch beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East + Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation
B1WA LKG1

550 King Street
Littleton, MA 01460-1250
USA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können unter Umständen von den hier genannten Preisen abweichen.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis)

und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielpprogramme entstehen.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter www.ibm.com/legal/copytrade.shtml.

Hinweise zu Datenschutzrichtlinien und Nutzungsbedingungen

IBM Softwareprodukte, einschließlich Software as a Service-Lösungen ("Softwareangebote"), können Cookies oder andere Technologien verwenden, um Informationen zur Produktnutzung zu erfassen, die Endbenutzererfahrung zu verbessern und Interaktionen mit dem Endbenutzer anzupassen oder zu anderen Zwecken. Ein Cookie ist ein Datenelement, das von einer Website an Ihren Browser gesendet wird und dann als Tag auf Ihrem Computer gespeichert werden kann, mit dem Ihr Computer identifiziert wird. In vielen Fällen werden von diesen Cookies keine personenbezogenen Daten erfasst. Wenn ein Softwareangebot, das von Ihnen verwendet wird, die Erfassung personenbezogener Daten anhand von Cookies und ähnlichen Technologien ermöglicht, werden Sie im Folgenden über die hierbei geltenden Besonderheiten informiert.

Abhängig von den bereitgestellten Konfigurationen kann dieses Softwareangebot Sitzungscookies und permanente Cookies verwenden, mit denen der Benutzername des Benutzers und andere personenbezogene Daten zum Zwecke des Sitzungsmanagements, zur Verbesserung der Benutzerfreundlichkeit und zu anderen funktionsbezogenen Zwecken sowie zur Nutzungsüberwachung erfasst werden. Diese Cookies können deaktiviert werden. Durch die Deaktivierung kann jedoch auch die von ihnen bereitgestellte Funktionalität nicht mehr genutzt werden.

Die Erfassung personenbezogener Daten mithilfe von Cookies und ähnlichen Technologien wird durch verschiedene rechtliche Bestimmungen geregelt. Wenn die für dieses Softwareangebot implementierten Konfigurationen Ihnen als Kunde die Möglichkeit bieten, personenbezogene Daten von Endbenutzern über Cookies und andere Technologien zu erfassen, dann sollten Sie ggf. juristische Beratung zu den geltenden Gesetzen für eine solche Datenerfassung in Anspruch nehmen. Dies gilt auch in Bezug auf die Anforderungen, die vom Gesetzgeber in Bezug auf Hinweise und die Einholung von Einwilligungen vorgeschrieben werden.

IBM setzt voraus, dass Kunden folgende Bedingungen erfüllen: (1) Sie stellen einen klar erkennbaren und auffälligen Link zu den Nutzungsbedingungen der Kundenwebsite (z. B. Datenschutzerklärung) bereit. Dieser Link muss wiederum einen Link zu der Vorgehensweise von IBM und des Kunden bei der Datenerhebung und Datennutzung umfassen. (2) Sie weisen darauf hin, dass Cookies und Clear GIFs/Web-Bacons von IBM im Auftrag des Kunden auf dem Computer des Besuchers platziert werden. Dieser Hinweis muss eine Erläuterung hinsichtlich des Zwecks dieser Technologie umfassen. (3) Sie müssen in dem gesetzlich vorgeschriebenen Umfang die Einwilligung von Websitebesuchern einholen, bevor Cookies und Clear GIFs/Web-Bacons vom Kunden oder von IBM im Auftrag des Kunden auf den Geräten der Websitebesucher platziert werden.

Weitere Informationen zur Verwendung verschiedener Technologien einschließlich der Verwendung von Cookies zu diesen Zwecken finden Sie im IBM Online Privacy Statement unter der Webadresse <http://www.ibm.com/privacy/details/us/en> im Abschnitt mit dem Titel "Cookies, Web Beacons and Other Technologies".

