

IBM Unica Kampagne  
Version 8 Release 6  
15. Mai 2012

*Handbuch Validierungs-PDK*

**IBM**

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 15 gelesen werden.

Diese Ausgabe bezieht sich auf Version 8, Release 6, Modifikation 0 von IBM Unica Kampagne (Produktnummer 5725-D22) und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs  
*IBM Unica Kampagne Version 8 Release 6.0, Validation PDK Guide*,  
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1998, 2012

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  
Mai 2012

---

# Inhaltsverzeichnis

## **IBM Unica Developer's Kit für das Validierungs-Plug-in (PDK) . . . . . 1**

Ergänzende Hilfe zum Validierungs-PDK . . . . .	1
Inhalt des Validierungs-PDK . . . . .	1
Komponenten des Validierungs-PDK . . . . .	2
Bereitstellungsort von Referenzmaterial . . . . .	2
Funktionalität des Validierungs-PDK . . . . .	2
Laden eines Plug-ins in IBM Unica Kampagne . . . . .	3
Angebots- gegenüber Kampagnenvalidierung . . . . .	4
Konfigurieren des Validierungs-PDK . . . . .	4
validationClass . . . . .	4
validationClasspath . . . . .	5
validatorConfigString . . . . .	6
Entwicklung eines Plug-ins . . . . .	6
Übersicht . . . . .	6
Hauptschritte bei der Erstellung von Plug-ins . . . . .	7
Einrichtung . . . . .	8
Erstellen der Validatoren . . . . .	8

Konfigurieren von IBM Unica Kampagne . . . . .	9
Die Validatorconfiguration testen . . . . .	9
Erstellen eines Validators . . . . .	9
Ausführbare Beispielanwendung . . . . .	10
Campaign für das Beispiexemplar des ausführbaren Plug-ins konfigurieren. . . . .	10
Erwartete Benutzerschnittstelle für ausführbare Anwendungen . . . . .	11
Beispiel . . . . .	11
Verhindern von Kampagnenbearbeitungen . . . . .	11

## **Kontakt zum technischen Support von IBM Unica . . . . . 13**

## **Bemerkungen. . . . . 15**

Marken. . . . .	17
-----------------	----



---

## IBM Unica Developer's Kit für das Validierungs-Plug-in (PDK)

Mit dem IBM® Unica Developer's Kit für das Validierungs-Plug-in (PDK) können Sie an Ihre individuellen Anforderungen angepasste Validierungslogiken für IBM Unica Kampagne entwickeln. Das Validierungs-PDK ist eine Unterklasse eines im Rahmen von IBM Unica Kampagne bereitgestellten allgemeineren Plug-in-Gerüsts.

Hauptabschnitte dieses Kapitels sind:

- „Inhalt des Validierungs-PDK“
- „Funktionalität des Validierungs-PDK“ auf Seite 2
- „Konfigurieren des Validierungs-PDK“ auf Seite 4
- „Entwicklung eines Plug-ins“ auf Seite 6
- „Hauptschritte bei der Erstellung von Plug-ins“ auf Seite 7
- „Ausführbare Beispielanwendung“ auf Seite 10
- „Beispiel“ auf Seite 11

---

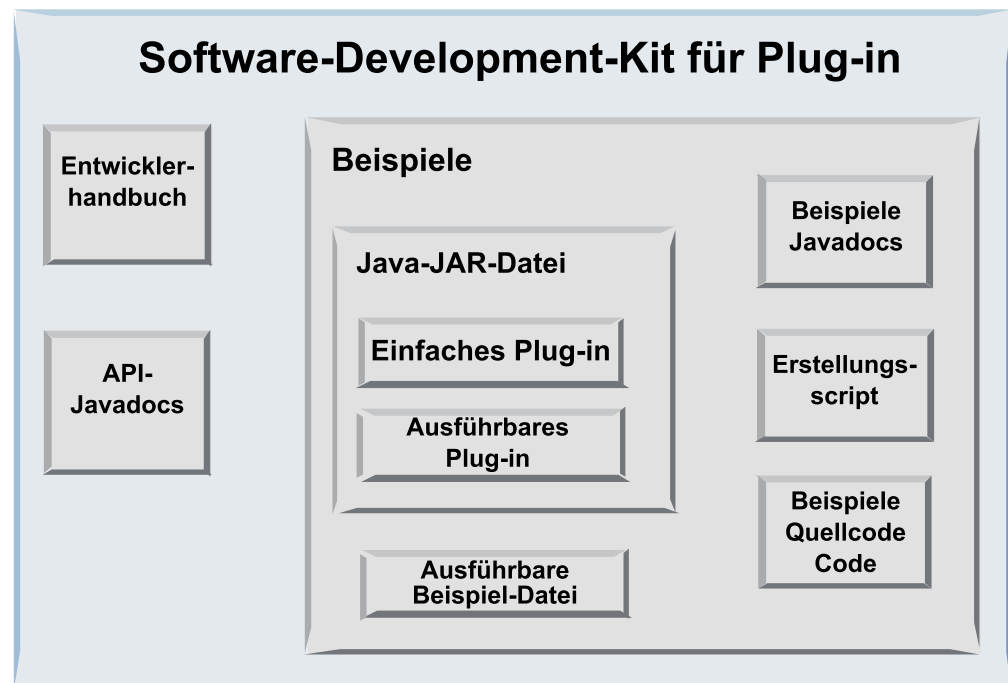
### Ergänzende Hilfe zum Validierungs-PDK

Falls Sie Hilfe beim Einsatz des Developer's Kit für das Validierungs-Plug-in (PDK) benötigen, wenden Sie sich bitte mit Ihren Fragen an den technischen Support von Unica. Weitere Einzelheiten hierzu finden Sie unter „Kontakt zum technischen Support von IBM Unica “ auf Seite 13.

---

### Inhalt des Validierungs-PDK

Das Validierungs-PDK beinhaltet alles, was Sie für die Entwicklung von Java-Plug-ins oder per Befehlszeile aufrufbaren ausführbaren Anwendungen benötigen, um Validierungsfunktionen in IBM Unica Kampagne zu realisieren.



## Komponenten des Validierungs-PDK

Das Developer's Kit für das Validierungs-Plug-in (PDK) enthält die folgenden Komponenten.

Komponente	Beschreibung
Handbuch Validierungs-PDK	Dieses Dokument.
Beispiele	Dies sind dokumentierte, erstellbare Beispiele für die Verwendung des PDK.
Java-jar-Datei	Eine Beispiel-JAR-Datei mit Beispiel-Plug-ins. Die JAR-Datei enthält: <ul style="list-style-type: none"><li>• <b>Einfaches Plug-in:</b> ein Beispiel einer eigenständigen Validator-Klasse.</li><li>• <b>Ausführbares Plug-in:</b> ein Beispielvalidator, der eine benutzerdefinierte, per Befehlszeile aufrufbare Anwendung ausführt.</li></ul>
Beispielexemplar einer ausführbaren Anwendung	Eine per Befehlszeile aufrufbare ausführbare Anwendung, die zusammen mit dem ausführbaren Plug-in unter UNIX verwendet werden kann.
Beispiel-Javadocs	Javadocs mit Beschreibungen der Beispielvalidator-Klassen.
Erstellungsscript	Ein Ant-Script, das aus dem enthaltenen Quellcode verwendbare Validatorscripts erstellt.
Beispielquellcode	Der Java-Quellcode für den einfachen Validator und den ausführbaren Validator.

## Bereitstellungsort von Referenzmaterial

Das Validierungs-PDK enthält Referenzinformationen sowohl für das Plug-in-API als auch für den Beispielcode. Die Referenzinformationen liegen in der Form von Javadocs vor. Sie finden die Dokumentation in der folgenden Datei:

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

Unter *Unica\_Home* befindet sich Ihr Installationsstammverzeichnis für IBM und unter *Campaign\_Home* Ihr Installationsverzeichnis für Kampagne.

Der Pfad zur Dokumentation kann also beispielsweise folgendermaßen aussehen:

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

Die Javadocs lassen sich in einem beliebigen Web-Browser anzeigen.

---

## Funktionalität des Validierungs-PDK

Ein mit dem Validierungs-PDK erstelltes Plug-in kann benutzerdefinierte Validierungslogiken für Kampagnen, Angebote oder beides ausführen. Mögliche Anwendungsbereiche der Validierungslogik sind zum Beispiel:

- Die Prüfung erweiterter Attribute (zum Beispiel gültig gegenüber optional oder zugehörige abhängige Felder)

- Bereitstellung von Berechtigungsservices, die nicht von IBM Unica Marketing Platform angeboten werden (zum Beispiel die Validierung, welche Benutzer welche erweiterten Attribute bearbeiten dürfen).

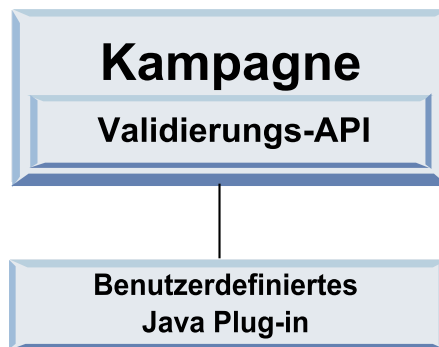
## Laden eines Plug-ins in IBM Unica Kampagne

Es gibt zwei Anwendungsmöglichkeiten für das API:

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

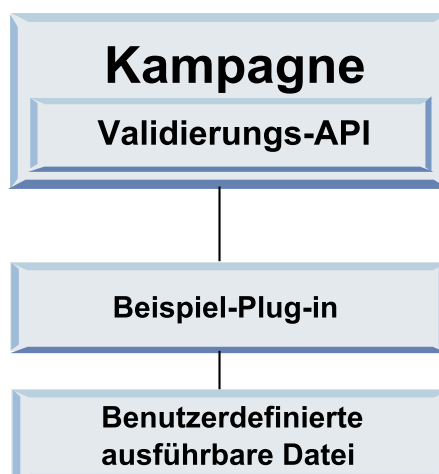
### Erstellen eines Plug-ins mit einer Java -Klasse, das in die Anwendung geladen wird

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



### Aufruf einer Anwendung zur Durchführung der Validierung

Die zweite Einsatzmöglichkeit des Validierungs-PDK ist die Verwendung eines der enthaltenen Plug-ins für den Aufruf einer ausführbaren Anwendung zur Durchführung der Validierung:



Diese ausführbare Anwendung kann in einer beliebigen Sprache geschrieben sein, muss sich aber auf dem IBM Unica Kampagne-Server befinden und dort ausgeführt werden. Das die ausführbare Anwendung aufrufende Plug-in lädt eine XML-Datei mit den zu validierenden Informationen. Solche Informationen sind zum Bei-

spiel der Bearbeiter des Objekts oder die Vorher-/Nachher-Werte für alle Standard- und erweiterten Attribute dieses Objekts. IBM Unica Kampagne erwartet die Rückgabe von Ergebnisinformationen in der Form einer XML-Datei. Weitere Informationen finden Sie unter „Entwicklung eines Plug-ins“ auf Seite 6.

## Angebots- gegenüber Kampagnenvalidierung

Das Validierungs-PDK wird zur Validierung von Angeboten und Kampagnen verwendet. Wenn ein Validierungs-Plug-in definiert wurde, wird es bei jedem Abspeichern eines Angebots- oder Kampagnenobjekts automatisch durch IBM Unica Kampagne aufgerufen. IBM Unica Kampagne setzt ein Flag, wenn es die Validierungsmethode des Plug-ins aufruft. IBM Unica Kampagne 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.

---

## Konfigurieren des Validierungs-PDK

Das Validierungs-PDK verwendet Konfigurationsparameter, um Campaign mitzuteilen, wo sich die zu verwendende Plug-in-Klasse befindet, und bietet eine Möglichkeit, um einige Konfigurationsoptionen in diese Plug-ins zu übernehmen.

Alle Einstellungen befinden sich auf der Konfigurationsseite unter:

Campaign > partitions > partition[n] > validation

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

Dieser Abschnitt behandelt die folgenden Einstellungen:

- „validationClass“
- „validationClasspath“ auf Seite 5
- „validatorConfigString“ auf Seite 6

### validationClass

Eigenschaft	Beschreibung
Beschreibung	Dies ist der Name der für die Validierung zu verwendenden Klasse. Der Wert der Eigenschaft <code>validationClasspath</code> gibt an, wo sich diese Klasse befindet.
Details	Die Klasse muss mit dem Paketnamen vollständig qualifiziert sein. Wenn diese Eigenschaft nicht festgelegt ist, führt Campaign keine benutzerdefinierte Validierung durch.



Eigenschaft	Beschreibung
Beispiel	<p>com.unica.campaign.core.validation.samples.SimpleCampaignValidator</p> <p>Hiermit wird validationClass auf die Klasse SimpleCampaignValidator aus dem Beispielcode festgelegt.</p>
Standard	<p>Standardmäßig wird kein Pfad festgelegt:</p> <pre>&lt;property name="validationClass" /&gt;</pre>

## validationClasspath

Eigenschaft	Beschreibung
Beschreibung	<p>Dies ist der Pfad zu der für die individuell definierte Validierung verwendeten Klasse. Es kann sich dabei um einen vollständigen Pfad oder um einen relativen Pfad handeln.</p>
Details	<p>Wenn der Pfad mit einem Schrägstrich endet (Vorwärtsschrägstrich / bei UNIX oder Rückwärtsschrägstrich \ bei Windows), geht Campaign davon aus, dass dies der Pfad zu einem Verzeichnis ist, das die zu verwendende Java-Plug-in-Klasse enthält.</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. Zum Beispiel ist der Wert</p> <pre>/opt/Unica/Campaign/devkits/validation/lib/validator.jar?</pre> <p>auf einer UNIX-Plattform der Pfad zu der Datei, die im Developer's Kit für Plug-ins enthalten ist.</p> <p>Bei einem relativen Pfad ist das Verhalten von dem Anwendungsserver abhängig, auf dem Campaign ausgeführt wird. WebLogic verwendet den Pfad zum Domänenarbeitsverzeichnis, das standardmäßig folgendermaßen lautet:</p> <pre>c:\bea\user_projects\domains\mydomain</pre> <p>Wenn die Einstellung keine Zeichenfolge enthält, versucht Campaign nicht, ein Plug-in zu laden.</p>
Beispiel	<pre>/opt/Unica/Campaign/devkits/validation/lib/validator.jar?</pre> <p>Dies ist auf einer UNIX-Plattform der Pfad zu der Datei, die im Developer's Kit für Plug-ins enthalten ist.</p>
Standard	<p>Standardmäßig wird kein Pfad festgelegt:</p> <pre>&lt;property name="validationClasspath" /&gt;</pre>

Eigenschaft	Beschreibung
Siehe auch	Siehe „validationClass“ auf Seite 4; hier finden Sie Informationen zur Benennung der zu verwendenden Klasse.

## validatorConfigString

Eigenschaft	Beschreibung
Beschreibung	Dies ist eine Zeichenfolge, die an das Validator-Plug-in übergeben wird, wenn es von Campaign geladen wird.
Details	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.  Zum Beispiel verwendet der ExecutableCampaignValidator (aus dem im PDK enthaltenen Beispiexemplar eines ausführbaren Plug-in) diese Eigenschaft, um das ausführbare Programm anzugeben, das aufgerufen werden soll.
Beispiel	Um das Beispiexemplar einer Bourne-Shell als Validierungsscript auszuführen, legen Sie validatorConfigString auf  /opt/unica/campaign/devkits/validation/src/com/unica/campaign/core/validation/samples/validate.sh fest.
Standard	Standardmäßig wird kein Pfad festgelegt:  <property name="validatorConfigString" />

---

## Entwicklung eines Plug-ins

### Übersicht

Ein Plug-in ist eine Java-Klasse, die beim Start geladen und bei jeder Validierung einer Kampagne oder eines Angebots aufgerufen wird. Diese Validierung erfolgt bei jedem Speichern einer Kampagne. Sie können Ihre eigenen Java-Plug-ins erstellen, indem Sie die Tools verwenden, die durch das Validierungs-PDK bereitgestellt werden. Es enthält Quellcode für Beispiel-Plug-ins und eine Ant-Datei (Apache Ant ist ein Erstellungstool auf Java-Basis) für das Kompilieren von Plug-ins. Die folgenden Abschnitte zeigen, wie Sie Ihre Umgebung für die Entwicklung eines Plug-ins konfigurieren können und anschließend werden Sie durch den Vorgang zur Erstellung Ihres eigenen Plug-ins geführt.

### Beispielvalidatoren

Die Standardinstallation von Campaign enthält zwei Beispielvalidatoren.

- SimpleCampaignValidator ist ein eigenständiges Plug-in, mit dem Vorgehensweisen etwa für eine benutzerdefinierte Autorisierung oder für die Validierung zulässiger Kampagnennamen gezeigt werden. Es befindet sich im folgenden Pfad:

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

Um mit ihm zu arbeiten, empfehlen wir das Anlegen einer Kopie der Klasse, anstatt es direkt zu bearbeiten.

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

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

```
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 Validierungstätigkeiten demonstriert werden.

## Testharness

Extreme Programming und andere agile Methodiken sind nützlich und beliebt. Ein wichtiger Aspekt dieser Methodiken ist, dass sie testorientiert sind. Kunden, die diese Methodiken einsetzen, führen in umfangreichem Maße Komponententests durch. Das Validierungs-PDK unterstützt diese Methodiken, indem es ein Testharness für die Ausführung eines Plug-ins außerhalb von Campaign bietet. Den Code testen zu können, ohne ihn in IBM Unica Kampagne laden zu müssen, beschleunigt den Entwicklungsprozess für das Plug-in.

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:
  - Zur Erstellung des Plug-ins ohne Komponententests führen Sie die Erstellungsscripts mit dem Befehl "ant jar" aus.
  - Zur Erstellung des Plug-ins mit Komponententests führen Sie die Erstellungsscripts mit dem Befehl "ant run-test" aus.

## Erstellungsscripts

Die Erstellungsscripts im PDK kompilieren alle Klassen in einem Verzeichnis und legen sie in einer JAR-Datei ab, die sich zur Verwendung in IBM Unica Kampagne eignet. Das bereitgestellte Erstellungsscript verwendet das Verzeichnis:

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

---

## Hauptschritte bei der Erstellung von Plug-ins

Folgendes sind die Hauptschritte bei der Erstellung von Plug-ins:

- „Einrichtung“ auf Seite 8
- „Erstellen der Validatoren“ auf Seite 8
- „Konfigurieren von IBM Unica Kampagne“ auf Seite 9
- „Die Validator Konfiguration testen“ auf Seite 9
- „Erstellen eines Validators“ auf Seite 9

In den folgenden Abschnitten erfahren Sie, wie Sie die Quelle für die Erzeugung der JAR-Datei erstellen.

## Einrichtung

Das Validierungs-PDK kann in jedem beliebigen System installiert werden, wobei aber die mit ihm erstellten Plug-ins in dem System abgestellt werden müssen, in dem IBM Unica Kampagne läuft. Wir empfehlen die Installation des PDK in dem System, mit welchem Sie Ihre Plug-ins testen.

Damit Java Plug-ins erstellt werden können, setzt das PDK voraus, dass in Ihrem System Jakarta Ant und ein Sun Java Developer Kit installiert sind. Zur Sicherstellung der Kompatibilität empfehlen wir die Verwendung der JDK-Pakete, die mit Ihrem Anwendungsserver ausgeliefert wurden.

**Anmerkung:** Sie können auch ein anderes JDK verwenden. Wenn Sie jedoch das auf Ihrem Anwendungsserver mitgelieferte JDK verwenden, werden für die von Ihnen erstellten Plug-ins während des Kompilierens Kompatibilitätsprüfungen durchgeführt. Wenn Sie zum Beispiel WebLogic 8.1 einsetzen, wofür das JDK 1.4 verwendet wird, verursachen alle JDK-1.5-spezifischen Klassen "class-not-found"-Fehler, wenn Campaign versucht, das Plug-in zu verwenden. Ein JDK 1.4 verhindert die Verwendung von JDK-1.5-spezifischen Klassen und stellt die Campaign-Kompatibilität des Plug-ins sicher.

Vorgehensweise für die Konfigurierung Ihrer Umgebung für das Validierungs-PDK:

1. Stellen Sie den Ordner mit der ausführbaren Ant-Datei in Ihren Pfad. Zum Beispiel müssen Sie für ein im Standardverzeichnis unter Windows installiertes WebLogic 8.1 Folgendes in Ihren Pfad stellen:

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

2. Geben Sie für die Umgebungsvariable JAVA\_HOME das Verzeichnis mit den bin- und lib- Verzeichnissen Ihres JDK an. Beispielsweise legen Sie für WebLogic 8.1 unter Windows JAVA\_HOME fest auf:

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

## Erstellen der Validatoren

Das PDK stellt ein Ant-Script bereit, mit dem sich der gesamte Code in den Beispieldateien erstellen lässt. Standardmäßig erstellt das Script eine JAR-Datei mit den Validierungsklassen. Optional kann es außerdem Javadocs erzeugen und Tests der Validatoren ausführen, um vor der Verwendung des Plug-ins im Produktivbetrieb sicherzustellen, dass die Validatoren in Campaign funktionsfähig sind.

Vorgehensweise für das Erstellen des Validators:

1. Wechseln Sie in das PDK-Verzeichnis

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

In diesem Verzeichnis finden Sie das Ant-Script build.xml.

2. Führen Sie die Ant-JAR-Datei in der Befehlszeile aus.

Ant führt das Script aus und erstellt eine JAR-Datei mit dem Namen validator.jar im Verzeichnis:

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

Sie haben nun einen benutzerdefinierten Validator für die Verwendung in IBM Unica Kampagne. Im nächsten Abschnitt wird erläutert, wie Sie Campaign für die Verwendung dieses Validators konfigurieren können.

## Konfigurieren von IBM Unica Kampagne

Nach dem Erstellen eines Validator-Plug-ins müssen Sie, um es nutzen zu können, IBM Unica Kampagne mitteilen, wo es sich befindet.

Zur Verwendung des SimpleCampaignValidator legen Sie die in „Konfigurieren des Validierungs-PDK“ auf Seite 4 beschriebenen Eigenschaften wie folgt fest:

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

## Die Validator-Konfiguration testen

Nach dem Erstellen der Datei validator.jar, welche die Klasse SimpleCampaignValidator enthält, und nach den erforderlichen Konfigurationsänderungen können Sie das Plug-in testen und einsetzen. Dieses Plug-in verhindert das Speichern einer Kampagne mit dem Namen "badCampaign".

Vorgehensweise für das Testen Ihrer Konfiguration:

1. Implementieren Sie Ihren Anwendungsserver neu, damit die Änderungen wirksam werden. Details zur Neuimplementierung Ihres Anwendungsservers finden Sie in der Dokumentation zu Ihrem Server.
2. Melden Sie sich bei IBM Unica Kampagne an und navigieren Sie zur Seite für die Kampagnenerstellung.
3. Erstellen Sie eine Kampagne mit dem Namen **badCampaign** und versuchen Sie, diese zu speichern.

Wenn alles richtig konfiguriert ist, lässt sich die neue Kampagne nicht speichern. Sie müssten stattdessen eine Fehlermeldung vom Validator erhalten.

## Erstellen eines Validators

In diesem Abschnitt erstellen Sie ein Validierungs-Plug-in ganz ähnlich dem SimpleCampaignValidator, das aber die Erstellung von Kampagnen mit dem Namen "badCampaign2" verhindert

Um einen Validator zu erstellen, gehen Sie wie folgt vor:

1. Erstellen Sie eine Kopie des Beispielvalidators SimpleCampaignValidator.java im Verzeichnis  

```
<Unica_Home\Campaign_Home>\devkits\validation\src\com  
\unica\campaign\core\validation\samples
```
2. Geben Sie der Kopie den Namen MyCampaignValidator.java und belassen Sie sie im selben Verzeichnis wie die Quelle.
3. Öffnen Sie MyCampaignValidator.java in einem Editor. Suchen Sie das Wort "badCampaign" im Dokument und ersetzen Sie es durch das Wort "badCampaign2".
4. Speichern Sie die Datei und schließen Sie den Editor.
5. Erstellen Sie die Validatoren erneut. Details hierzu finden Sie unter „Erstellen der Validatoren“ auf Seite 8.

**Anmerkung:** Falls Ihr Anwendungsserver die Datei `validate.jar` sperrt, während sie in Verwendung ist, müssen Sie den Server vor der Erstellung der Validatoren stoppen.

6. Konfigurieren Sie `campaign_config.xml` für die Verwendung Ihrer neuen Klasse neu:

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

7. Testen Sie den Validator. Details hierzu finden Sie unter „Die Validatorkonfiguration testen“ auf Seite 9.

Es darf nicht möglich sein, Kampagnen mit dem Namen "badCampaign2" zu speichern.

---

## Ausführbare Beispielanwendung

Die Beispielvalidatoren enthalten auch einen `ExecutableCampaignValidator`, der eine ausführbare Anwendung von der Befehlszeile aus ausführen kann.

Dieser Abschnitt:

- zeigt wie IBM Unica Kampagne für die Ausführung des Beispiel-Plug-ins für ausführbare Anwendungen zu definieren ist und
- beschreibt, wie Sie eigene Plug-ins für ausführbare Anwendungen erstellen können, die der Benutzerschnittstelle für ausführbare Anwendungen entsprechen.

## Campaign für das Beispiexemplar des ausführbaren Plug-ins konfigurieren

Zur Verwendung des `ExecutableCampaignValidator` legen Sie die in „Konfigurieren des Validierungs-PDK“ auf Seite 4 beschriebenen Eigenschaften wie folgt fest:

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

Das mit dem PDK ausgelieferte Beispielscript ist ein Bourne-Shell-Script für UNIX. Es verweigert allen Benutzern mit dem Benutzernamen "badUser" die Erstellung von Kampagnen. Der Code für diese ausführbare Anwendung steht im Verzeichnis:

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

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

## Erwartete Benutzerschnittstelle für ausführbare Anwendungen

ExecutableCampaignValidator ruft eine Befehlszeile mit den folgenden Argumenten auf:

- executable\_name: Dies ist die Zeichenfolge, die im validatorConfigString in der Marketing-Plattform IBM Unica festgelegt ist.
- data\_filename: Dies ist der Name der von der ausführbaren Anwendung eingelesenen Eingabedatei. Die Eingabedaten müssen in XML formatiert sein.
- expected\_result\_filename: Dies ist der Name der von der ausführbaren Anwendung zu sendenden Ausgabedatei. Die erwarteten Ergebnisse haben die Form Daten XXX.xml, wobei XXX eine Zahl ist.
  - Hier sehen Sie ein Beispiel für das Senden erfolgreicher Daten:

```
<ValidationResult result="0" generalFailureMessage="" />
```
  - Hier sehen Sie ein Beispiel für das Senden nicht erfolgreicher Daten:

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

**Anmerkung:** Es wird dringend empfohlen, gut verständliche Fehlermeldungen zu definieren, damit die Benutzer das Problem beheben können, bevor sie versuchen, eine weitere Sicherungsoperation durchzuführen.

---

### Beispiel

Dieser Abschnitt enthält ein Beispiel für ein Validierungsszenario.

### Verhindern von Kampagnenbearbeitungen

Wenn Sie verhindern möchten, dass ein Bearbeiter einer Kampagne den Kampagnencode ändert, können Sie eine individuell definierte Kampagnenvalidierungsroutine verwenden. Die Routine sorgt dafür, dass beim Speichern der Kampagne die folgende Prüfung vorgenommen wird:

```
neuer_Kampagnencode == alter_Kampagnencode
```

Damit auch Neuerstellungen von Kampagnen berücksichtigt werden können, wird mit einem Flag angezeigt, ob die zu validierende Kampagne neu ist (creation) oder ob sie bereits existiert (edit). Zeigt dieses Flag den **edit**-Zustand an, führen Sie den Vergleich von Kampagnencodes durch.

Die Campaign-Anwendung setzt dieses Flag im Objekt InputValidationData, welches sie dann an das Plug-in übergibt. Das Plug-in liest das Flag bei der Bestimmung, ob die Validierung für eine neue oder eine geänderte Kampagne vorgenommen werden soll.





---

## Kontakt zum technischen Support von IBM Unica

Sollte sich ein Problem nicht mithilfe der Dokumentation beheben lassen, können sich die für den Kundendienst zuständigen Kontaktpersonen Ihres Unternehmens telefonisch an den technischen Support von IBM Unica wenden. Damit wir Ihnen möglichst schnell helfen können, beachten Sie dabei bitte die Informationen in diesem Abschnitt.

Wenn Sie wissen möchten, wer die zuständige Kontaktperson Ihres Unternehmens ist, wenden Sie sich an Ihren IBM Unica -Administrator.

### Informationen, die Sie bereithalten sollten

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

- Kurze Beschreibung der Art Ihres Problems
- Detaillierte Fehlermeldungen, die beim Auftreten des Problems angezeigt werden
- Schritte zum Reproduzieren des Problems
- Entsprechende Protokolldateien, Session-Dateien, Konfigurationsdateien und Daten
- Informationen über Ihre Produkt- und Systemumgebung, die Sie, wie nachfolgend in „Systeminformationen“ beschrieben, erhalten können.

### Systeminformationen

Bei Ihrem Anruf beim technischen Support von IBM Unica 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 zu der installierten IBM Unica -Anwendung.

Sie können über **Hilfe > Info** (Help > About) auf die Info-Seite zugreifen. Wenn Sie nicht auf die Info-Seite zugreifen können, finden Sie die Versionsnummer der IBM Unica -Anwendung in der Datei `version.txt` im Installationsverzeichnis der Anwendung.

### Kontaktinformationen für den technischen Support von IBM Unica

Wenn Sie sich an den technischen Support von IBM Unica wenden möchten, finden Sie weitere Informationen auf der Website des technischen Supports für IBM Unica -Produkte (<http://www.unica.com/about/product-technical-support.htm>).



---

## 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. An Stelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene 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  
170 Tracer Lane  
Waltham, MA 02451  
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 Beispielprogramme entstehen.

---

## Marken

IBM, das IBM Logo und [ibm.com](http://ibm.com) sind Marken oder eingetragene Marken der International Business Machines Corporation. Weitere Produkt- und Servicenamen 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](http://www.ibm.com/legal/copytrade.shtml).





