

IBM Campaign
Versión 9 Release 0
11 de diciembre de 2012

Guía PDK de validación

IBM

Note

Before using this information and the product it supports, read the information in "Notices" en la página 13.

Contenido

IBM PDK (Plug-in Developer's Kit) de validación 1

Ayuda adicional de PDK de validación	1
Contenido del PDK de validación	1
Componentes del PDK de validación	2
Buscar material de referencia	2
Prestaciones del PDK de validación.	2
Cargar el plug-in en IBM Campaign	3
Validación de oferta versus campaña	4
Configuración del PDK de validación	4
validationClass	4
validationClasspath	5
validatorConfigString	5
Desarrollar un plug-in	6
Descripción general	6
Pasos principales para crear plug-ins	7
Configuración	7

Crear los validadores	7
Configurar IBM Campaign	8
Probar la configuración del validador	8
Crear un validador	8
Muestra ejecutable	9
Configurar Campaign para el plug-in ejecutable de muestra	9
Interfaz de uso de ejecutable esperado	10
Ejemplo	10
Impedir ediciones de campaña	10

Contacting IBM technical support . . . 11

Notices 13

Trademarks	15
Privacy Policy and Terms of Use Considerations . . .	15

IBM PDK (Plug-in Developer's Kit) de validación

El PDK (Plug-in Developer's Kit) de validación de IBM® le permite desarrollar lógica de validación personalizada en IBM Campaign. El PDK de validación es una subclase de una estructura de plug-in más genérica que se proporciona con IBM Campaign.

Las secciones clave de este capítulo incluyen:

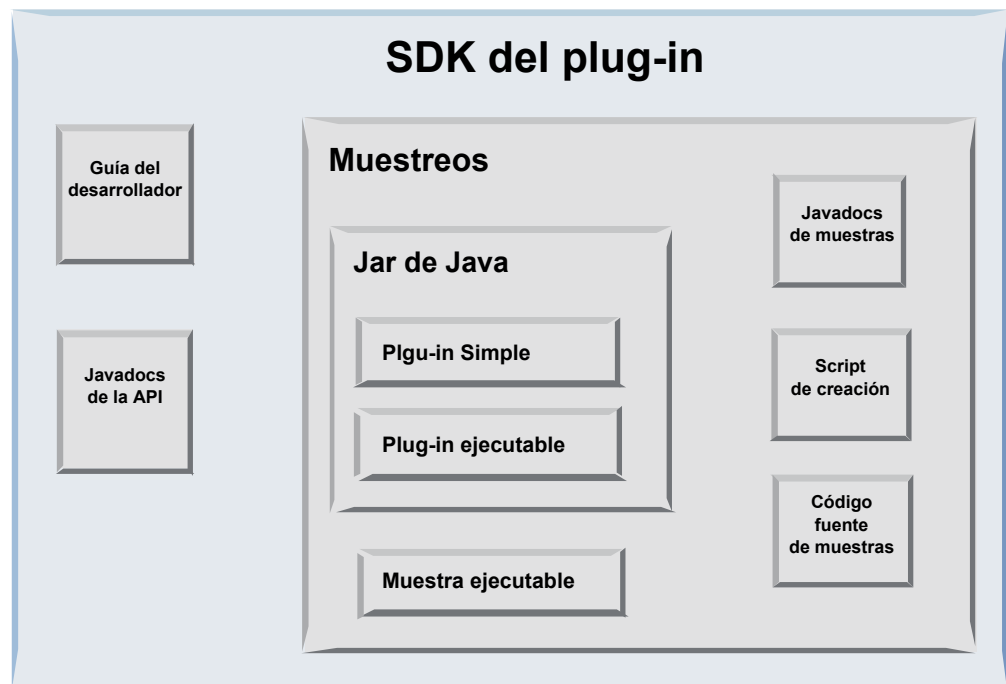
- “Contenido del PDK de validación”
- “Prestaciones del PDK de validación” en la página 2
- “Configuración del PDK de validación” en la página 4
- “Desarrollar un plug-in” en la página 6
- “Pasos principales para crear plug-ins” en la página 7
- “Muestra ejecutable” en la página 9
- “Ejemplo” en la página 10

Ayuda adicional de PDK de validación

Si necesita ayuda al utilizar el PDK (Plug-in Developer's Kit) de validación, póngase en contacto con el soporte técnico de IBM para efectuar sus preguntas. Para obtener detalles, consulte “Contacting IBM technical support” en la página 11.

Contenido del PDK de validación

El PDK de validación contiene todo lo que necesita para desarrollar ejecutables de la línea de comandos Java™ para añadir validación a IBM Campaign.



Componentes del PDK de validación

El PDK (Plug-in Developer's Kit) de validación tiene los siguientes componentes.

Componente	Descripción
Guía PDK de validación	Este documento.
Muestras	Las muestras son ejemplos construibles y documentados de cómo utilizar el PDK.
Archivo .jar Java	Un archivo JAR de muestra que contiene los plug-ins de muestra. El jar contiene: <ul style="list-style-type: none">• Plug-in simple: un ejemplo de una clase de validador autónomo.• Plug-in ejecutable: un validador de ejemplo que ejecuta un ejecutable de la línea de comandos definido por el usuario para realizar la validación.
Muestra ejecutable	Un ejecutable de la línea de comandos que se puede utilizar con el plug-in ejecutable en UNIX.
Javadocs de muestra	Javadocs que describen las clases de validador de muestra.
Script de creación	Un script Ant que crea el código fuente incluido en plug-ins de validador utilizables.
Código fuente de muestras	El código fuente Java para el validador simple y el validador ejecutable.

Buscar material de referencia

El PDK de validación contiene información de referencia para la API de plug-in y el código de muestra. La información de referencia tiene formato de Javadocs. Para ver la documentación, abra el archivo siguiente:

```
C:\Dir_Inicio_IBM_EMM\Dir_Inicio_Campaign\devkits\validation\javadoc\index.html
```

Donde *Dir_Inicio_IBM_EMM* es la ruta al directorio de instalación raíz de IBM y *Dir_Inicio_Campaign* es la ruta al directorio de instalación de Campaign.

Por ejemplo, la ruta a la documentación puede ser similar al ejemplo siguiente:

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

Puede ver los Javadocs en cualquier navegador web.

Prestaciones del PDK de validación

Un plug-in creado con el PDK de validación puede ejecutar lógica de validación personalizada para campañas, ofertas o ambas cosas. Algunas de las posibles utilizaciones de la lógica de validación son:

- Comprobación de los atributos ampliados (personalizados)
- Proporcionar servicios que están fuera del ámbito de IBM Marketing Platform (por ejemplo, validar cuáles son los usuarios con permiso para editar y cuáles son los atributos ampliados).

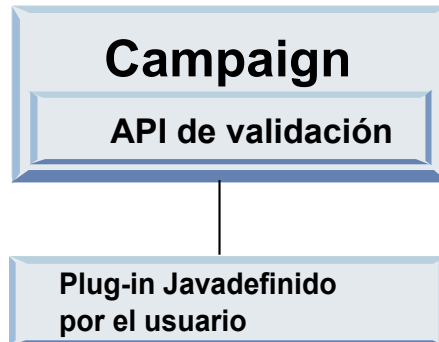
Cargar el plug-in en IBM Campaign

Existen dos formas de utilizar la API:

- Utilizarla para crear un plug-in de clase Java que se carga en la aplicación.
- Utilizar uno de los plug-ins incluidos para llamar a una aplicación ejecutable para manejar la validación.

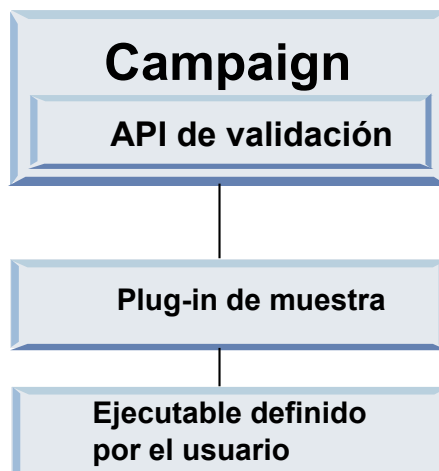
Crear un plug-in de clase Java que se carga en la aplicación

El PDK de validación proporciona las interfaces, las clases de asistente y herramientas del desarrollador para desarrollar estas clases.



Llamar a un aplicación para manejar la validación

El segundo modo de utilizar el PDK de validación consiste en utilizar uno de los plug-ins incluidos para llamar a una aplicación ejecutable para manejar la validación:



El ejecutable puede estar escrito en cualquier lenguaje, pero debe residir en el servidor de IBM Campaign y ejecutarse en el servidor. El plug-in que llama al ejecutable envía un archivo XML que contiene la información que debe validarse; por ejemplo, el usuario que edita el objeto y los valores before/after para todos los atributos estándar y ampliados de dicho objeto). IBM Campaign espera que la información de los resultados se devuelva con el formato de un archivo XML. Para obtener más información, consulte el apartado "Desarrollar un plug-in" en la página 6.

Validación de oferta versus campaña

El PDK de validación puede validar ofertas y campañas. Si hay definido un plug-in de validación, lo llama automáticamente IBM Campaign cada vez que se guarda un objeto de oferta o campaña. IBM Campaign establece un indicador cuando llama al método de validación del plug-in. IBM Campaign pasa los siguientes indicadores:

- `ValidationInputData.CAMPAIGN_VALIDATION`, cuando se añade o cambia una campaña o bien
- `ValidationInputData.OFFER_VALIDATION`, cuando se añade o edita una oferta.

A continuación, puede utilizar estos indicadores para construir reglas de validación aplicables a ofertas y campañas.

Configuración del PDK de validación

El PDK de validación utiliza parámetros de configuración que indican a Campaign cómo encontrar la clase de plug-in que debe utilizarse y ofrece un modo de pasar información de configuración a estos plug-ins.

Todos los valores se encuentran en la página de configuración:

Campaign > particiones > partición[n] > validación

Nota: La validación funciona con varias particiones; `partición[n]` se puede cambiar por cualquier nombre de partición para proporcionar rutinas de validación también para estas particiones.

Esta sección contiene los siguientes valores:

- “`validationClass`”
- “`validationClasspath`” en la página 5
- “`validatorConfigString`” en la página 5

validationClass

Propiedad	Descripción
Descripción	Es el nombre de la clase que se utilizará para la validación. El valor de la propiedad <code>validationClasspath</code> indica dónde reside esta clase.
Detalles	La clase debe estar completamente calificada con su nombre de paquete. Si no se ha establecido correctamente, Campaign no realiza ninguna validación personalizada.
Ejemplo	<code>com.unica.campaign.core.validation.samples.SimpleCampaignValidator</code> Esto establece <code>validationClass</code> en la clase <code>SimpleCampaignValidator</code> del código de muestra.
Valor predeterminado	De forma predeterminada, no se establece ninguna ruta: <code><property name="validationClass" /></code>

validationClasspath

Propiedad	Descripción
Descripción	Es la ruta a la clase que se utiliza para la validación personalizada. Puede ser una ruta completa o una ruta relativa.
Detalles	<p>Utilice una ruta completa o una ruta relativa. Si la ruta es relativa, el comportamiento depende del servidor de aplicaciones que ejecuta Campaign. WebLogic utiliza la ruta al directorio de trabajo del dominio que, de forma predeterminada, es</p> <p><code>c:\bea\user_projects\domains\mydomain</code></p> <p>Si la ruta finaliza con una barra inclinada (una barra inclinada / para UNIX o una barra inclinada invertida \ para Windows), Campaign supone que apunta a la ubicación de la clase de plug-in Java plug-in que se debe utilizar.</p> <p>Si la ruta no finaliza con una barra inclinada, Campaign supone que es el nombre de un archivo .jar que contiene la clase Java, como se muestra en el ejemplo siguiente.</p> <p>La la ruta no se establece, Campaign o intenta cargar un plug-in.</p>
Ejemplo	<p><code>/<CAMPAIGN_HOME>/devkits/validation/lib/validator.jar</code></p> <p>Es la ruta en una plataforma UNIX que apunta al archivo JAR que viene empaquetado con el kit del desarrollador de plug-in.</p>
Valor predeterminado	<p>De forma predeterminada, no se establece ninguna ruta:</p> <pre><property name="validationClasspath" /></pre>
Consulte también	Consulte "validationClass" en la página 4 para obtener información sobre cómo designar la clase que se utilizará.

validatorConfigString

Propiedad	Descripción
Descripción	Es una cadena que se pasa al plug-in de validador cuando Campaign lo carga.
Detalles	<p>La forma en que el plug-in utiliza esta cadena depende del diseñador. Puede utilizarlo para enviar una cadena de configuración al plug-in cuando el sistema lo carga.</p> <p>Por ejemplo, ExecutableCampaignValidator (del plug-in ejecutable de muestra incluido con el PDK) utiliza esta propiedad para indicar el ejecutable que se ejecutará.</p>
Ejemplo	<p>Para ejecutar el script de shell Bourne de muestra como script de validación, establezca</p> <p><code>validatorConfigString</code> en</p> <pre>/opt/unica/campaign/devkits/validation/ src/com/unica/campaign /core/validation/samples/validate.sh</pre>

Propiedad	Descripción
Valor predeterminado	De forma predeterminada, no se establece ninguna ruta: <property name="validatorConfigString" />

Desarrollar un plug-in

Descripción general

Un plug-in es una clase Java que se carga durante el tiempo de inicio y se llama cada vez que se valida una campaña o una oferta. Esta validación se produce cada vez que un usuario guarda una campaña. Puede crear sus propios plug-ins Java utilizando las herramientas que proporciona el PDK de validación. Contiene código fuente para los plug-ins de muestra y un archivo Ant (Apache Ant es una herramienta de creación basada en Java) que se utiliza para compilar plug-ins. Las siguientes secciones le muestran cómo configurar el entorno para desarrollar un plug-in i, a continuación, le guía por la creación de su propio plug-in.

Validadores de muestra

En la instalación estándar de Campaign se incluyen dos validadores de muestra.

- SimpleCampaignValidator es un plug-in autónomo que muestra cómo realizar acciones como la autorización personalizada y la validación de nombres de campaña permitidos. Puede encontrarlo en la ruta siguiente:

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

Es recomendable realizar una copia de la clase mientras se trabaja con ella, en lugar de editarla directamente.

- ExecutableCampaignValidator es un plug-in Java que llama a una aplicación ejecutable para realizar la validación. El código fuente para ExecutableCampaignValidator se incluye en el mismo directorio que SimpleCampaignValidator.

Sin embargo, la finalidad real de este ejemplo su utilización como un ejecutable de línea de comandos para la validación. Este archivo se encuentra en la ruta siguiente:

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

Este archivo es un ejecutable de bucle de retorno de muestra, que ilustra los tipos comunes de trabajo de validación.

Arnés de prueba

La programación extrema y otras metodologías ágiles son útiles y populares. Un aspecto importante de estas metodologías es que son de prueba central. Los clientes que utilizan estas metodologías utilizan ampliamente la prueba de unidad. El PDK de validación da soporte a estas metodologías ofreciendo un arnés de prueba para ejecutar un plug-in fuera de Campaign. El hecho de poder probar el código sin colocarlo en IBM Campaign acelera el proceso del desarrollador de plug-in.

Para utilizar el arnés de prueba:

1. Modifique el caso de prueba de unidad para reflejar la lógica de validación en el plug-in.

2. Ejecute el script de creación:
 - Para crear el plug-in sin realizar pruebas de unidad, ejecute los scripts de creación utilizando el comando "ant jar".
 - Para crear el plug-in y además realizar la prueba de unidad, ejecute los scripts de creación utilizando el comando "ant run-test".

Scripts de creación

Los scripts de creación del PDK compilan todas las clases en un directorio y las colocan en un jar adecuado para utilizarlo en IBM Campaign. El directorio que utiliza el script de creación proporcionado es:

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

Pasos principales para crear plug-ins

Estos son los pasos principales para crear plug-ins:

- "Configuración"
- "Crear los validadores"
- "Configurar IBM Campaign" en la página 8
- "Probar la configuración del validador" en la página 8
- "Crear un validador" en la página 8

Las siguientes secciones muestran cómo crear el origen para crear el archivo .jar.

Configuración

El PDK de validación se puede instalar en cualquier máquina, pero los plug-ins que crea con él deben estar en la máquina en que se ejecuta IBM Campaign. Se recomienda instalar el PDK en la máquina en la que prueba los plug-ins.

El PDK requiere que disponga de un kit de desarrollador Java de Sun y Jakarta Ant en la máquina plug-ins Java plug-ins. Se recomienda utilizar los paquetes de Ant y JDK que se proporcionan con el servidor de aplicaciones para garantizar la compatibilidad.

Para configurar el entorno para utilizar el PDK de validación:

1. Añada la carpeta que contiene el ejecutable Ant a la ruta. Se proporcionan dos ejemplos.
 - Para WebLogic 10gR3 instalado en el directorio predeterminado en Windows, añada lo siguiente a la ruta: C:\Oracle\Middleware\wlserver_10.3\common\bin
 - Para WebSphere 7.0 instalado en el directorio predeterminado en Windows, añada lo siguiente a la ruta: C:\IBM\WebSphere\AppServer1\bin
2. Establezca la variable de entorno JAVA_HOME en el directorio que contiene los directorios bin y lib del JDK. Se proporcionan dos ejemplos.
 - Para WebLogic 11gR1 en Windows, establezca JAVA_HOME en C:\Oracle\Middleware\jdk160_18
 - Para WebSphere 7.0 en Windows, establezca JAVA_HOME en C:\IBM\WebSphere\AppServer1\java\jre

Crear los validadores

El PDK proporciona un script Ant que puede crear todo el código en los archivos de muestra. El comportamiento predeterminado del script es crear un jar que

contenga las clases de validación. Opcionalmente, también puede crear Javadocs y ejecutar pruebas en los validadores para asegurarse de que funcionan en Campaign antes de intentar utilizar el plug-in en producción.

Para crear el validador:

1. Cambie al directorio de PDK,
`<Dir_Inicio_IBM_EMM\Dir_Inicio_Campaign>\devkits\validation\build`
Verá el script Ant, `build.xml`, en este directorio.
2. Ejecute el jar Ant en la línea de comandos.
Ant ejecuta el script y produce un archivo JAR denominado `validator.jar` en el directorio:
`<Dir_Inicio_IBM_EMM\Dir_Inicio_Campaign>\devkits\validation\build\lib`

Ahora dispone de un validador personalizado que se puede utilizar en IBM Campaign. En el apartado siguiente se explica cómo configurar Campaign para que utilice este validador.

Configurar IBM Campaign

Después de crear un plug-in de validador, debe indicar su ubicación a IBM Campaign para poder utilizarlo.

Para utilizar `SimpleCampaignValidator`, establezca las propiedades que se describen en “Configuración del PDK de validación” en la página 4 del modo siguiente:

- `validationClasspath`: `Unica\campaign\devkits\validation\lib\validator.jar`
- `validationClass`:
`com.unica.campaign.core.validation.samples.SimpleCampaignValidator`
- No es necesario establecer `validatorConfigString` para utilizar `SimpleCampaignValidator` porque no utiliza una cadena de configuración.

Probar la configuración del validador

Después de crear el archivo `validator.jar` que contiene la clase `SimpleCampaignValidator` y realizar los cambios necesarios en la configuración, está listo para probar y utilizar el plug-in. Este plug-in impide que los usuarios guarden una campaña denominada “`badCampaign`.”

Para probar la configuración:

1. Vuelva a desplegar el servidor de aplicaciones para que los cambios entren en vigor. Para obtener detalles sobre cómo volver a desplegar el servidor de aplicaciones, consulte la documentación del servidor.
2. Inicie sesión en IBM Campaign y vaya a la página de creación de campaña.
3. Cree una campaña con el nombre **badCampaign** e intente guardarla.

Si se ha configurado correctamente todo, no podrá guardar la campaña nueva. Debería recibir un mensaje de error del validador.

Crear un validador

En esta sección, se crea un plug-in de validación que se parece bastante a `SimpleCampaignValidator`, pero se impide la creación de campañas denominadas “`badCampaign2`.”

Para crear un validador:

1. Realice una copia del validador de muestra SimpleCampaignValidator.java, que se encuentra en
`<Dir_Inicio_IBM_EMM\Dir_Inicio_Campaign>\devkits\validation\src\com\unica\campaign\core\validation\samples`
2. Denomine la copia MyCampaignValidator.java y déjela en el mismo directorio que el origen.
3. Abra MyCampaignValidator.java en un editor. Busque la palabra "badCampaign" en el documento y sustitúyala por la palabra "badCampaign2."
4. Guarde el archivo y cierre el editor.
5. Vuelva a crear los validadores. Para obtener más detalles, consulte "Crear los validadores" en la página 7.

Nota: Si el servidor de aplicaciones bloquea el archivo validate.jar mientras se utiliza, debe detener el servidor antes de crear los validadores.

6. Vuelva a configurar campaign_config.xml para utilizar la nueva clase:

```
<property name="validationClass"
value="com.unica.campaign.core.validation.samples.MyCampaignValidator">
```
7. Pruebe el validador. Para obtener más detalles, consulte "Probar la configuración del validador" en la página 8.

No debería poder guardar las campañas denominadas "badCampaign2."

Muestra ejecutable

Los validadores de muestra también incluyen un validador, ExecutableCampaignValidator, que puede ejecutar un ejecutable desde la línea de comandos.

Esta sección:

- Muestra cómo configurar IBM Campaign para ejecutar el plug-in ejecutable de muestra y
- Describe cómo crear los propios plug-ins ejecutables que son compatibles con el uso de la interfaz de uso de ejecutable.

Configurar Campaign para el plug-in ejecutable de muestra

Para utilizar ExecutableCampaignValidator, establezca la propiedades que se describen en "Configuración del PDK de validación" en la página 4 del modo siguiente:

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

El script de muestra que se proporciona con el PDK es un script de shell Bourne para UNIX. Niega la creación de la campaña a cualquiera que tenga el nombre de usuario "badUser." Puede ver el código para el ejecutable en el directorio siguiente:

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

Debe desarrollar su propio script que realice la validación pertinente para su implementación. Los lenguajes de script como PERL y Python son buenos candidatos para los scripts de proceso de texto como este; sin embargo, cualquier lenguaje que se pueda ejecutar desde la línea de comandos es aceptable.

Interfaz de uso de ejecutable esperado

ExecutableCampaignValidator llama a un ejecutable con un línea de comandos que contiene los argumentos siguientes:

- nombre_ejecutable: es la cadena establecida en validatorConfigString en IBM Marketing Platform.
- nombre_archivo_datos: es el nombre del archivo que el ejecutable lee como entrada. Los datos de entrada deben tener formato XML.
- nombre_archivo_resultados_esperados: es el nombre del archivo que el ejecutable debe enviar como salida. Los resultados esperados deben tener el formato datos XXX.xml donde XXX es un número.
 - A continuación se proporciona un ejemplo de cómo se envían datos satisfactorios:

```
<ValidationResult result="0" generalFailureMessage="" />
```
 - A continuación se proporciona un ejemplo de cómo se envían datos anómalos:

```
<ValidationResult result="1" generalFailureMessage="">
  <AttributeError attributeName="someAttribute" errorMessage="something" />
  <AttributeError attributeName="someAttribute2" errorMessage="something2" />
</ValidationResult>
```
 - El texto del archivo XML debe codificarse con caracteres ASCII normales o UTF-8.

Nota: Es muy recomendable proporcionar mensajes de error fáciles de comprender para que puedan corregir el problema antes de volver a intentar otra operación de guardar.

Ejemplo

Esta sección contiene un ejemplo de un escenario de validación.

Impedir ediciones de campaña

Si intenta impedir que alguien que edita una campaña pueda cambiar el código de campaña, puede utilizar una rutina de validación de campaña personalizada. La rutina garantizará que cuando se guarde la campaña se realice la siguiente comprobación:

```
código_campaña_nuevo == código_campaña_anterior
```

Para manejar el caso en que la campaña se crea por primera vez, pase al indicador de rutina indicando si la campaña que se valida es nueva (creación) o existente (edición). Si este indicador indica **edición**, realice la comparación de los códigos de campaña.

La aplicación Campaign establece este indicador en el objeto InputValidationData que a continuación pasa al plug-in. El plug-in lee el indicador cuando determina si la validación es para una campaña nueva o modificada.

Contacting IBM technical support

If you encounter a problem that you cannot resolve by consulting the documentation, your company's designated support contact can log a call with IBM technical support. Use the information in this section to ensure that your problem is resolved efficiently and successfully.

If you are not a designated support contact at your company, contact your IBM administrator for information.

Information to gather

Before you contact IBM technical support, gather the following information:

- A brief description of the nature of your issue.
- Detailed error messages you see when the issue occurs.
- Detailed steps to reproduce the issue.
- Related log files, session files, configuration files, and data files.
- Information about your product and system environment, which you can obtain as described in "System information."

System information

When you call IBM technical support, you might be asked to provide information about your environment.

If your problem does not prevent you from logging in, much of this information is available on the About page, which provides information about your installed IBM applications.

You can access the About page by selecting **Help > About**. If the About page is not accessible, you can obtain the version number of any IBM application by viewing the `version.txt` file located under the installation directory for each application.

Contact information for IBM technical support

For ways to contact IBM technical support, see the IBM Product Technical Support website: (http://www-947.ibm.com/support/entry/portal/open_service_request).

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

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

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Privacy Policy and Terms of Use Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. A cookie is a piece of data that a web site can send to your browser, which may then be stored on your computer as a tag that identifies your computer. In many cases, no personal information is collected by these cookies. If a Software Offering you are using enables you to collect personal information through cookies and similar technologies, we inform you about the specifics below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name, and other personal information for purposes of session management, enhanced user usability, or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

Various jurisdictions regulate the collection of personal information through cookies and similar technologies. If the configurations deployed for this Software Offering provide you as customer the ability to collect personal information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for providing notice and consent where appropriate.

IBM requires that Clients (1) provide a clear and conspicuous link to Customer's website terms of use (e.g. privacy policy) which includes a link to IBM's and Client's data collection and use practices, (2) notify that cookies and clear gifs/web beacons are being placed on the visitor's computer by IBM on the Client's behalf along with an explanation of the purpose of such technology, and (3) to the extent required by law, obtain consent from website visitors prior to the placement of cookies and clear gifs/web beacons placed by Client or IBM on Client's behalf on website visitor's devices

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Online Privacy Statement at: <http://www.ibm.com/privacy/details/us/en> section entitled "Cookies, Web Beacons and Other Technologies."



Impreso en España