

IBM Campaign
Versión 9 Release 1
Octubre de 2013

Guía PDK de validación

IBM

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información de la sección "Avisos" en la página 17.

Esta edición aplica a la versión 9, release 1, modificación 0 de IBM Campaign y a todos los releases y modificaciones subsiguientes mientras no se indique lo contrario en nuevas ediciones.

© Copyright IBM Corporation 1998, 2013.

Contenido

Capítulo 1. IBM Visión general del PDK (Plug-in Developer's Kit) de validación 1

Contenido del PDK de validación	1
Dos formas de utilizar la API de validación	2
Crear un plug-in de clase Java que se carga en la aplicación	2
Llamar a un aplicación para manejar la validación	3
Validación de oferta versus campaña	3
Validadores de muestra incluidos en el PDK de validación	4
Arnés de prueba para el PDK de validación	4
Scripts de creación para el PDK de validación	5

Capítulo 2. Desarrollo de plug-ins de validación para Campaign 7

Configuración del entorno para utilizar el PDK de validación	7
Compilación del validador.	8
Configuración de Campaign para utilizar un plug-in de validación	8

validationClass	9
validationClasspath	9
validatorConfigString	10
Comprobación de la configuración del validador	10
Creación de un validador.	11
Escenario de validación de ejemplo: Impedir ediciones de campaña	11

Capítulo 3. Invocación a un aplicación para manejar la validación 13

Configuración de Campaign para utilizar el plug-in ejecutable de muestra	13
Interfaz de uso de ejecutable esperado	13

Cómo contactar con el soporte técnico de IBM 15

Avisos 17

Marcas registradas	19
------------------------------	----

Capítulo 1. IBM Visión general del PDK (Plug-in Developer's Kit) de validación

Utilice el IBM PDK (Plug-in Developer's Kit) de validación para desarrollar la lógica de validación personalizada para que se utilice en IBM Campaign.

Puede crear plug-ins para que lleven a cabo una lógica de validación personalizada para campañas, ofertas o ambas cosas.

Algunas de las posibles utilizaciones de la lógica de validación son:

- Para comprobar los atributos ampliados (personalizados)
- Para 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).

El PDK de validación es una subclase de una estructura de plug-in más genérica que se proporciona con IBM Campaign.

El PDK de validación contiene información de referencia de Javadoc para la API de plug-In API y el código de muestra. Para ver la documentación, abra el archivo siguiente en cualquier navegador web:

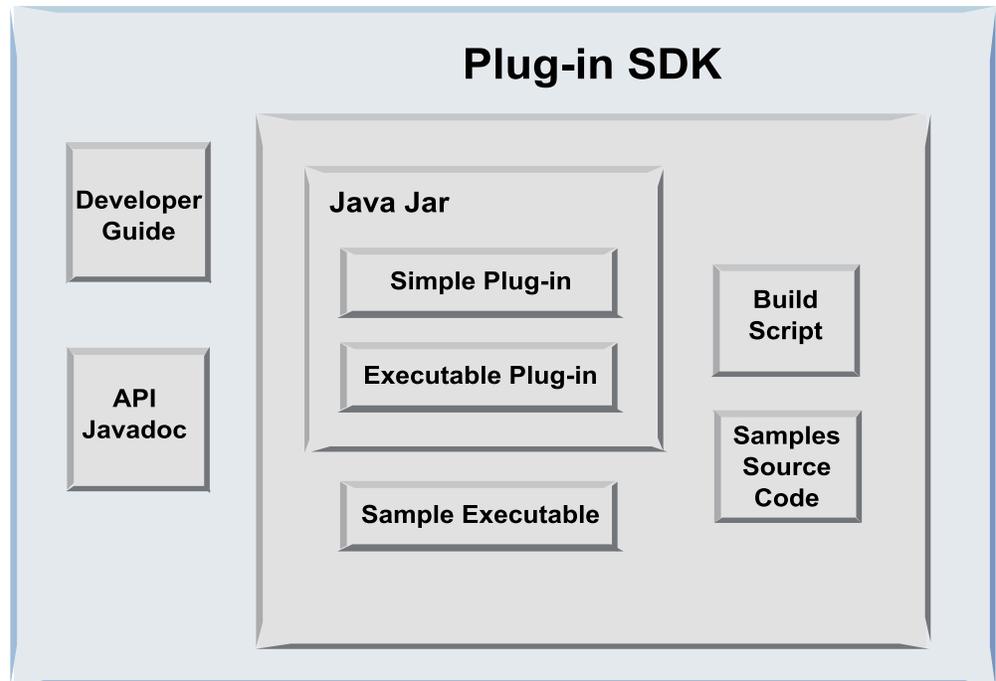
```
C:\Dir_Inicio_IBM_EMM\Dir_Inicio_Campaign\devkits\validation\javadoc\index.html
```

Por ejemplo:

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

Contenido del PDK de validación

El PDK de validación contiene componentes para desarrollar los plug-ins de Java™ o los ejecutables de línea de comandos para añadir la validación personalizada a IBM Campaign. El PDK contiene ejemplos documentados y compilables de cómo utilizar el PDK.



En la tabla siguiente se describe cada componente.

Tabla 1. Componentes del PDK de validación

Componente	Descripción
Guía del desarrollador	Un documento PDF denominado <i>IBM Campaign - Guía PDK de validación</i> .
API Javadoc	Información de referencia para la API del plug-in.
Archivo .jar Java	Un archivo JAR de muestra que contiene los plug-ins de muestra. El archivo 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 para realizar la validación.
Muestra ejecutable	Un ejecutable de la línea de comandos que se puede utilizar con el plug-in ejecutable en UNIX.
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.

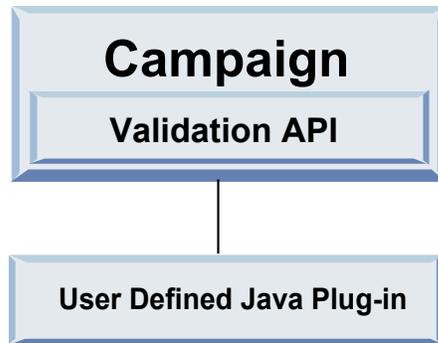
Dos formas de utilizar la API de validación

Hay dos formas de utilizar la API de validación.

- 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.

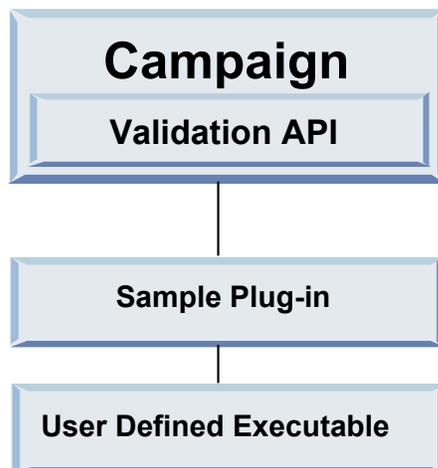
Crear un plug-in de clase Java que se carga en la aplicación

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

Puede utilizar uno de los plug-ins PDK de validación incluidos para llamar a una aplicación ejecutable para que maneje la validación.



El ejecutable puede estar escrito en cualquier lenguaje, pero debe residir en el servidor de IBM Campaign y se debe ejecutar 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 anteriores y posteriores 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.

Validación de oferta versus campaña

Un plug-in creado con el PDK de validación de Campaign puede ejecutar la lógica de validación personalizada para campañas, ofertas o ambas cosas.

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

- `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.

Validadores de muestra incluidos en el PDK de validación

En el PDK de validación de Campaign se incluyen dos validadores de muestra: `SimpleCampaignValidator` y `ExecutableCampaignValidator`.

- `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 antes de editarla para conservar la versión original por si la necesitara.

- `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`:

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

Sin embargo, la finalidad real de este ejemplo su utilización como un ejecutable de línea de comandos para la validación. Este archivo se halla 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 para el PDK de validación

El hecho de poder probar el código sin colocarlo en IBM Campaign acelera el proceso del desarrollador de plug-in.

Los clientes que utilizan la programación extrema y otras metodologías ágiles utilizan la realización de pruebas ampliamente. El PDK de validación da soporte a estas metodologías ofreciendo un arnés de pruebas para ejecutar un plug-in fuera de Campaign.

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 para el PDK de validación

Los scripts de creación en el PDK de validación compilan todas las clases en un directorio y las colocan en un archivo JAR que resulta adecuado para utilizarlo en IBM Campaign.

El script de creación proporcionado utiliza el directorio siguiente:

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

Capítulo 2. Desarrollo de plug-ins de validación para Campaign

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.

La validación se produce cada vez que un usuario guarda una campaña. Puede crear sus propios plug-ins de Java con las herramientas que se proporcionan en el PDK de validación. El PDK 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.

En los pasos siguientes se explica cómo configurar el entorno para desarrollar un plug-in y, a continuación, se le guía por la creación de su propio plug-in.

1. “Configuración del entorno para utilizar el PDK de validación”
2. “Compilación del validador” en la página 8
3. “Configuración de Campaign para utilizar un plug-in de validación” en la página 8
4. “Comprobación de la configuración del validador” en la página 10
5. “Creación de un validador” en la página 11

Configuración del entorno para utilizar el PDK de validación

Para utilizar el PDK de validación con Campaign, debe modificar la ruta y establecer la variable de entorno `JAVA_HOME`.

El PDK de validación se puede instalar en cualquier máquina, pero los plug-ins que cree con él deben estar en la máquina donde se esté ejecutando IBM Campaign. Se recomienda instalar el PDK en la máquina donde se están realizando las pruebas de los plug-ins.

El PDK requiere que disponga de un kit de desarrollador Apache Ant y Java de Sun en la máquina para crear los plug-ins de Java. Para garantizar la compatibilidad, utilice los paquetes de Ant y JDK que se proporcionan con el servidor de aplicaciones.

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 11gR1 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`

Compilación del validador

El PDK de validación 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 Javadoc y ejecutar pruebas en los validadores para garantizar 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_EM\Campaigninicio_>\devkits\validation\build`
Verá el script Ant, `build.xml`, en este directorio.
2. Ejecute el jar Ant en la línea de comandos.
 - Para crear el plug-in sin realizar ninguna prueba de unidad, utilice el comando "ant jar".
 - Para crear el plug-in y también realizar pruebas de unidad, utilice el comando "ant run-test".

Ant ejecuta el script y produce un archivo JAR denominado `validator.jar` en el directorio:

```
<Dir_Inicio_IBM_EMM\Campaigninicio_>\devkits\validation\build\lib
```

Ahora dispone de un validador personalizado que se puede utilizar en IBM Campaign. El próximo paso es configurar Campaign para que utilice este validador.

Configuración de Campaign para utilizar un plug-in de validación

Para configurar Campaign para que utilice un plug-in de validación, utilice los valores de configuración en Campaign > particiones > partición[n] > validación.

Las propiedades de configuración indican a Campaign cómo buscar la clase de plug-in y representan una forma de pasar información de configuración a los plug-ins.

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.

Puede ajustar los siguientes valores de configuración de validación:

- "validationClass" en la página 9
- "validationClasspath" en la página 9
- "validatorConfigString" en la página 10

Para utilizar `SimpleCampaignValidator`, establezca las propiedades tal como se indica a continuación:

- `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.

Para utilizar `ExecutableCampaignValidator`, establezca las propiedades tal como se indica a continuación:

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

validationClass

El valor de `validationClass` indica a Campaign el nombre de la clase que se debe utilizar para la validación con un plug-in PDK de validación.

Propiedad	Descripción
Descripción	El nombre de la clase que se utilizará para la validación. El valor de la propiedad <code>validationClasspath</code> indica la ubicación de esta clase.
Detalles	La clase debe estar completamente calificada con su nombre de paquete. Si no se ha establecido esta propiedad, Campaign no realiza ninguna validación personalizada.
Ejemplo	<code>com.unica.campaign.core.validation.samples.SimpleCampaignValidator</code> Este ejemplo 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

El valor de `validationClasspath` indica a Campaign la ubicación de la clase que se debe utilizar para la validación con un plug-in PDK de validación.

Propiedad	Descripción
Descripción	La ruta a la clase que se ha utilizado para la validación personalizada.
Detalles	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 <code>c:\bea\user_projects\domains\midominio</code> . Si la ruta finaliza con una barra inclinada (/ para UNIX o \ para Windows), Campaign supone que apunta a la ubicación de la clase del plug-in Java que se debe utilizar. Si la ruta no finaliza con una barra inclinada, Campaign supone que es el nombre de un archivo .jar que contiene la clase Java, tal como se muestra en el ejemplo siguiente. La la ruta no se establece, Campaign o intenta cargar un plug-in.
Ejemplo	<code>/<CAMPAIGN_HOME>/devkits/validation/lib/validator.jar</code> Es la ruta en una plataforma UNIX que apunta al archivo JAR que se empaqueta con el kit del desarrollador de plug-in.

Propiedad	Descripción
Valor predeterminado	De forma predeterminada, no se establece ninguna ruta: <property name="validationClasspath" />
Consulte también	Consulte "validationClass" en la página 9 para obtener información sobre cómo designar la clase que se utilizará.

validatorConfigString

El valor validatorConfigString se pasa al plug-in de validador cuando Campaign lo carga.

Propiedad	Descripción
Descripción	Una cadena que se pasa al plug-in de validador cuando lo carga Campaign.
Detalles	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. Por ejemplo, ExecutableCampaignValidator (del plug-in ejecutable de muestra incluido con el PDK) utiliza esta propiedad para indicar el ejecutable que se ejecutará.
Ejemplo	Para ejecutar el script de shell Bourne de muestra como script de validación, establezca validatorConfigString en /opt/unica/campaign/devkits/validation/src/com/unica/campaign/core/validation/samples/validate.sh
Valor predeterminado	De forma predeterminada, no se establece ninguna ruta: <property name="validatorConfigString" />

Comprobación de la configuración del validador

Tras crear el archivo validator.jar que contiene la clase SimpleCampaignValidator y realizar los cambios necesarios en la configuración, puede probar y utilizar el plug-in.

El siguiente ejemplo de plug-in impide que los usuarios de Campaign 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 instrucciones, consulte la documentación del servidor.
2. Inicie la 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. Si recibe un mensaje de error del validador, sabrá que está funcionando correctamente.

Creación de un validador

Estas instrucciones explican cómo 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 "Compilación del validador" en la página 8.

Nota: Si el servidor de aplicaciones bloquea el archivo `validate.jar` mientras se está utilizando, deberá 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 "Comprobación de la configuración del validador" en la página 10.

No debería poder guardar las campañas denominadas "badCampaign2."

Escenario de validación de ejemplo: Impedir ediciones de campaña

En este ejemplo se explica cómo utilizar la validación para impedir ediciones específicas en una 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 garantiza que la siguiente comprobación se realiza cuando se guarda la campaña:

```
código_campaña_nuevo == código_campaña_anterior
```

Para manejar el caso en que la campaña se crea por primera vez, pase a la rutina un indicador que informa de si la campaña que se está validando es nueva (creación) o existente (edición). Si este indicador muestra **edición**, compare 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.

Capítulo 3. Invocación a un aplicación para manejar la validación

El PDK de validación incluye un validador de muestra, `ExecutableCampaignValidator`, que ejecuta un archivo ejecutable `validate.sh`, desde la línea de comandos, para realizar la validación.

En las secciones siguientes se explica cómo hacerlo:

- Configure Campaign para ejecutar el plug-in ejecutable de muestra y
- Cree un plug-in ejecutable que sea compatible con la utilización de la interfaz de uso de ejecutables.

Configuración de Campaign para utilizar el plug-in ejecutable de muestra

Para utilizar `ExecutableCampaignValidator`, ajuste los valores de configuración en Campaign > particiones > partición[n] > validación.

Establezca las propiedades tal como se indica a continuación:

- `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 de validación 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, por ejemplo, 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

El plug-in `ExecutableCampaignValidator` llama a un archivo ejecutable con una línea de comandos que contiene los argumentos siguientes.

- `nombre_ejecutable`: la cadena establecida en `validatorConfigString` en IBM Marketing Platform.
- `nombre_archivo_datos`: el nombre del archivo que lee el ejecutable como entrada. Los datos de entrada deben tener formato XML.
- `nombre_archivo_resultados_esperados`: 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.

Cómo contactar con el soporte técnico de IBM

Si encuentra un problema que no puede resolver consultando la documentación, la persona responsable del contacto con el servicio de soporte técnico en su empresa puede realizar una llamada al soporte técnico de IBM . Utilice la información de este apartado para garantizar que su problema se resuelva de forma eficaz y satisfactoria.

Si usted no es una de las personas responsables del contacto con el servicio de soporte técnico en su empresa, póngase en contacto con su administrador de IBM para obtener información.

Información que debe recopilar

Antes de ponerse en contacto con el soporte técnico de IBM , debe recopilar la información siguiente:

- Una breve descripción del tipo de problema.
- Mensajes de error detallados que aparezcan cuando se produce el problema.
- Pasos detallados para reproducir el problema.
- Archivos de registro relacionados, archivos de sesión, archivos de configuración y archivos de datos.
- Información sobre el producto y el entorno del sistema, que se puede obtener tal como se describe en "Información del sistema."

Información del sistema

Cuando llama al soporte técnico de IBM , es posible que le soliciten que proporcione información sobre el entorno.

Si el problema que tiene no le impide iniciar sesión, la mayoría de la información estará disponible en la Página Acerca de, que proporciona información sobre las aplicaciones de IBM que tiene instaladas.

Puede acceder a la página Acerca de, seleccionando **Ayuda > Acerca de**. Si no la página Acerca de no está accesible, puede obtener el número de versión de cualquier aplicación de IBM del archivo `version.txt` que se encuentra en el directorio de instalación de cada aplicación.

Información de contacto para el soporte técnico de IBM

Para saber las formas de contacto con el soporte técnico de IBM , consulte el sitio web del soporte técnico del producto IBM : (<http://www.unica.com/about/product-technical-support.htm>).

Avisos

Esta información se ha desarrollado para productos y servicios que se ofrecen en los EE.UU.

Es posible que IBM no ofrezca los productos, servicios o características que se tratan en este documento en otros países. Consulte al representante local de IBM para obtener información de los productos y servicios disponibles actualmente en su zona. Las referencias a programas, productos o servicios de IBM no pretenden establecer ni implicar que sólo puedan utilizarse dichos productos, programas o servicios de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran la materia descrita en este documento. La entrega de este documento no le otorga ninguna licencia sobre dichas patentes. Puede enviar consultas acerca de licencias, por escrito, a la dirección siguiente:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
EE.UU.

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM de su país o envíe sus consultas, por escrito, a:

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

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde estas disposiciones sean incompatibles con la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍAS DE NINGÚN TIPO, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUYENDO PERO NO LIMITÁNDOSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO. Algunas legislaciones no contemplan la declaración de limitación de responsabilidad, ni implícita ni explícita, en determinadas transacciones, por lo que cabe la posibilidad de que esta declaración no sea aplicable en su caso.

Esta información puede contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en la información que aquí se presenta; estos cambios se incorporarán en las nuevas ediciones de la publicación. IBM puede realizar en cualquier momento mejoras o cambios en los productos o programas descritos en esta publicación sin previo aviso.

Cualquier referencia en este documento a sitios web que no son de IBM se proporciona únicamente para su comodidad y no significa en modo alguno que se recomiende dichos sitios web. El material de estos sitios web no forma parte del material correspondiente a este producto IBM y el uso de estos sitios web es a cuenta y riesgo del usuario.

IBM puede utilizar o distribuir cualquier información que se le proporcione en la forma que considere adecuada, sin incurrir por ello en ninguna obligación para con el remitente.

Los usuarios con licencia de este programa que deseen obtener información sobre éste con el propósito de habilitar: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido este) y (ii) el uso mutuo de la información que se ha intercambiado, deben ponerse en contacto con:

IBM Corporation
170 Tracer Lane
Waltham, MA 02451
EE.UU.

Esta información puede estar disponible, sujeta a los términos y condiciones adecuados, incluido en algunos casos, el pago de una tasa.

El programa bajo licencia que se describe en este documento y todo el material bajo licencia disponible los proporciona IBM bajo los términos de las Condiciones Generales de IBM, Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre las partes.

Los datos sobre rendimiento aquí incluidos se han determinado en un entorno controlado. Por tanto, los resultados obtenidos en otros entornos operativos pueden variar de forma significativa. Algunas mediciones pueden haberse realizado en sistemas en nivel de desarrollo y no existe garantía alguna de que estas mediciones sean iguales en los sistemas de disponibilidad general. Además, es posible que algunas mediciones se hayan calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deberían verificar los datos aplicables en sus entornos específicos.

La información relacionada con los productos que no son de IBM se ha obtenido de los proveedores de dichos productos, sus anuncios publicados u otras fuentes de disponibilidad pública. IBM no ha comprobado estos productos y no puede confirmar la precisión de su rendimiento, compatibilidad ni contemplar ninguna otra reclamación relacionada con los productos que no son de IBM. Las preguntas relacionadas con las prestaciones de los productos que no son de IBM deberán dirigirse a los proveedores de estos productos.

Todas las declaraciones relativas a la dirección o intención futura de IBM están sujetas a ser cambiadas o retiradas sin aviso y representan sólo propósitos y objetivos.

Todos los precios de IBM que se muestran son precios actuales recomendados por IBM de venta al público y están sujetos a cambios sin notificación previa. Los precios de los distribuidores pueden variar.

Esta información contiene ejemplos de datos e informes utilizados en operaciones empresariales cotidianas. Para mostrarlos de la forma más completa posible, los

ejemplos incluyen nombres de personas, de empresas, de marcas y de productos. Todos estos nombres son ficticios y cualquier similitud a los nombres y direcciones que haya utilizado una empresa real es pura coincidencia.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de muestra en lenguaje fuente, que ilustran técnicas de programación en las distintas plataformas operativas. Puede copiar, modificar y distribuir los programas de muestra de cualquier forma, sin tener que pagar a IBM, con intención de desarrollar, utilizar, comercializar o distribuir programas de aplicación que estén en conformidad con la interfaz de programación de aplicaciones (API) de la plataforma operativa para la que están escritos los programas de muestra. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni dar por supuesta la fiabilidad, la capacidad de servicio ni la funcionalidad de estos programas. Los programas de muestra se proporcionan "TAL CUAL", sin garantía de ningún tipo. IBM no será responsable de los daños que surjan por el uso de los programas de muestra.

Si está visualizando esta información en copia software, es posible que no aparezcan las fotografías y las ilustraciones en color.

Marcas registradas

IBM, el logotipo de IBM e ibm.com son marcas registradas o marcas comerciales de International Business Machines Corp., registradas en muchas jurisdicciones en todo el mundo. Otros nombres de servicios y productos podrían ser marcas registradas de IBM u otras compañías. Hay disponible una lista actual de marcas registradas de IBM en el apartado "Información de marca registrada y copyright" en el sitio web www.ibm.com/legal/copytrade.shtml.



Impreso en España