Versión 9 Release 1.2
Septiembre de 2015

IBM Campaign - Guía de PDK de validación

Note
Before using this information and the product it supports, read the information in “Notices” en la página 17.

Esta edición se aplica a la versión 9, release 1, modificación 2 de IBMCampaign y a todos los releases y modificaciones subsiguientes hasta que no se indique lo contrario en nuevas ediciones.

Contenido

Capítulo 1. IBMVisión 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:

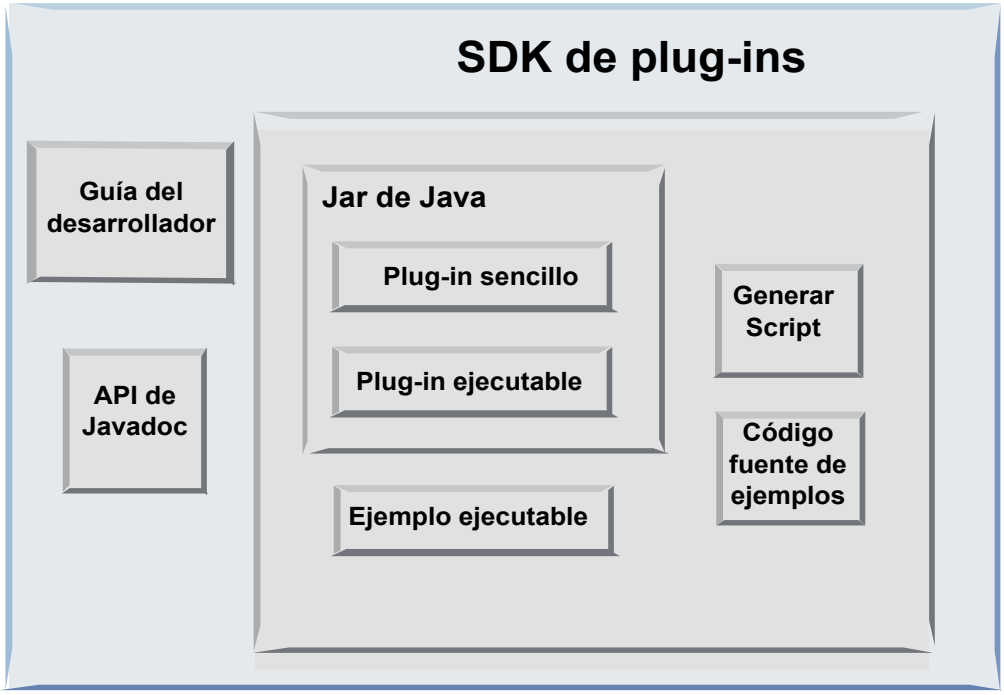
C:\Dir_Inicio_IBM_EMM\Dir_Inicio_Campaign\devkits\validation\javadoc\index.html

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 *IBM Campaign - Guía PDK de validación*.
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: • 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.

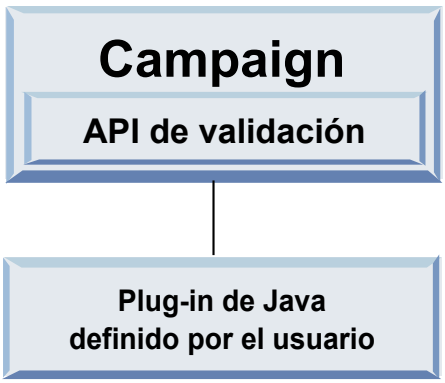
Dos formas de utilizar la API de validación

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.

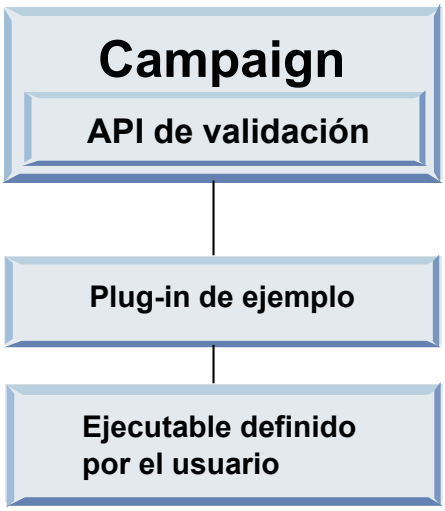
Crear un plug-in de clase Java que se carga en la aplicación

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.
   - ParaWebLogic 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 `validationClasspath` 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	`com.unica.campaign.core.validation.samples.SimpleCampaignValidator` Este ejemplo establece `validationClass` en la clase `SimpleCampaignValidator` del código de muestra.
Valor predeterminado	De forma predeterminada, no se establece ninguna ruta: `<property name="validationClass" />`

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 `c:\bea\user_projects\domains\midominio`. 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	`/<CAMPAIGN_HOME>/devkits/validation/lib/validator.jar` 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:

  `<Campaign`*`inicio_`*`>\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.

Before you contact 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 these guidelines 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.

Nota: Technical Support does not write or create API scripts. For assistance in implementing our API offerings, contact IBM Professional Services.

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 that 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, check for a `version.txt` file that is located under the installation directory for your application.

Contact information for IBM technical support

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

Nota: To enter a support request, you must log in with an IBM account. This account must be linked to your IBM customer number. To learn more about associating your account with your IBM customer number, see **Support Resources > Entitled Software Support** on the Support Portal.

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 on 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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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
B1WA LKG1
550 King Street
Littleton, MA 01460-1250
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 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