IBM Unica Campaign
Versión 9 Release 0
15 de enero de 2013

Especificación de la API Offer

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información de la sección “Avisos” en la página 49.

Esta edición se aplica a la versión 9, release 0, modificación 0 de IBM Unica Campaign y a todos los releases y modificaciones subsiguientes hasta que se diga lo contrario en nuevas ediciones.

Contenido

Capítulo 1. Introducción

Nota: Este documento define la parte de la oferta de la versión 3.0 de la interfaz de programación de aplicaciones de los servicios de IBM® Unica Campaign, también denominada CampaignServices. Solo los servicios de oferta que se describen en esta guía están soportados.

Nota: Si actualiza a IBM Unica Campaign versión 8.2 o superior y tiene implementados actualmente los servicios de IBM Unica Campaign, los cambios en la API necesitados para la actualización de AXIS versión 1.3. a AXIS2 1.4.1 requieren cambios en el código de la aplicación. Para obtener detalles, consulte "Nuevos archivos .jar dependientes" en la página 2.

En esta sección se incluyen los temas principales siguientes:

Resumen de alto nivel de los cambios

- Migración del motor de SOAP AXIS versión 1.3 a AXIS2 1.4.1.
- El WSDL se ha reestructurado para tratar aspectos de manejo de parámetros necesarios/opcionales.
- El archivo `.jar` de la API Client ha cambiado como resultado de cambios de WSDL y, por lo tanto, los apéndices y las clases generados han cambiado. Los parámetros del método de API Client no han cambiado, pero los constructores de los objetos de valores de soporte se han modificado debido al uso del convertidor AXIS2 WSDL2Java.
- El URL del servicio web se ha cambiado para que apunte a:

  ```
  http://<host>:<puerto>/Campaign/services/CampaignServices30Service
  ```

  y el WSDL correspondiente se puede recuperar en:

  ```
  http://<host>:<puerto>/Campaign/services/CampaignServices30Service?wsdl
  ```

Como resultado de estos cambios, si utiliza actualmente la API debe modificar el código de aplicación. En función de si utiliza la API de cliente o el WSDL, consulte las secciones siguientes para ver los detalles:

- "Implementaciones existentes mediante la API Client"
- "Implementaciones existentes mediante WSDL directamente" en la página 3

Implementaciones existentes mediante la API Client

Si utiliza el archivo `.jar` de la API Client para interactuar con la aplicación web de Campaign, en las secciones siguientes se detallan los cambios que debe tener en cuenta.

Nuevo archivo .jar de la API Client

Los cambios en la implementación interna de las clases de API Client significan que la aplicación Java™ debe utilizar el nuevo archivo `.jar` ubicado en:

```
<INICIO_CAMPAIGN>/devkits/CampaignServicesAPI/lib/
CampaignServicesClient30.jar
```

Para ver un ejemplo de Java que muestre la creación de la nueva oferta, consulte "Código OfferAPI.java" en la página 43. El mismo ejemplo se puede encontrar en la instalación de Campaign en:

```
<INICIO_CAMPAIGN>/devkits/CampaignServicesAPI/samples/OfferAPI.java
```

Nuevos archivos .jar dependientes

Como resultado de la actualización a AXIS2 versión 1.4.1, la aplicación Java también debe actualizarse para utilizar los archivos `.jar` de la distribución de AXIS2 1.4.1, ya que `CampaignServicesClient30.jar` depende de estos archivos `.jar`. Todos los archivos `.jar` dependientes se deben incluir en la ruta de clases Java de la aplicación y se pueden encontrar en el archivo `Campaign.war` ubicado en:

```
<INICIO_CAMPAIGN>/Campaign.war
```

Extraiga los archivos `.jar` de `Campaign.war`, e inclúyalos en la ruta de clases Java.

Cambios en el constructor de la API Client

Al construir el objeto de la API Client, cambie el URL del servicio web y la firma de excepción, según se muestra en este ejemplo.

```
try {
URL serviceURL = new URL(PROTOCOL, HOST, PORT,
"/Campaign/services/CampaignServices30Service");
CampaignServices30SoapClient client = new
CampaignServices30SoapClient(serviceURL, TIMEOUT);
} catch (RemoteException exception) {
exception.printStackTrace();
}
```

Cambios en constructores parametrizados de las clases de soporte

Con el motor AXIS2, las clases y los apéndices generados ya no tienen constructores parametrizados. En su lugar, estas clases solo tienen el constructor sin argumentos predeterminado con métodos set y get para los miembros.

Por ello:

```
WSReference wsRef = new WSReference(WSComponentTypeEnum typeEnum, Long id);
```

se cambia por:

```
WSReference wsRef = new WSReference();
wsRef.setComponentTypeEnum(typeEnum);
wsRef.setId(id);
```

Implementaciones existentes mediante WSDL directamente

El WSDL del servicio web Campaign se utiliza para generar apéndices del lado de cliente y clases de soporte utilizando una herramienta de convertidor de otra empresa. Si utiliza el WSDL para interactuar con la aplicación web de Campaign, en las siguientes secciones se detallan los cambios que debe tener en cuenta. Se proporcionan ejemplos de la utilización de la herramienta WSDL2Java de Apache AXIS2 1.4.1.

URL de servicio y ubicación de WSDL

El servicio web de Campaign para IBM Unica Campaign se despliega en:

```
http://host:puerto/Campaign/services/CampaignServices30Service
```

El WSDL correspondiente se puede recuperar de:

```
http://host:puerto/Campaign/services/CampaignServices30Service?wsdl
```

Generación de apéndices y clases

La herramienta WSDL2Java de Apache AXIS2 1.4.1 se puede utilizar para generar los apéndices y las clases Java de soporte del WSDL. Un ejemplo de tarea Ant se muestra a continuación.

La herramienta también se puede utilizar desde la línea de mandatos con un conjunto de argumentos similar. Los valores de los argumentos se pueden modificar para ajustarlos al entorno.

Nota: El vínculo ADB predeterminado se utiliza para el siguiente ejemplo de convertidor WSDL2Java.

```
<java classname="org.apache.axis2.wsdl.WSDL2Java" fork="true">
    <classpath refid="axis2.class.path"/> <!-Ruta de clases que tiene
bibliotecas AXIS2 -->
    <arg value="-uri"/>
    <arg file="CampaignServices30.wsdl"/> <!-Ubicación real de
WSDL -->
    <arg value="-s"/> <!-- Generar código de estilo de sincronización -->
    <arg value="-Euwc"/> <!-- Se encarga de la generación de tipos java de derivador
para elementos nillable = true. -->
    <arg value="-uw"/> <!-- Parámetros para anular recorte -->
    <arg value="-u"/> <!-- Clases de desempaquetado -->
    <arg value="-ns2p"/> <!-- Espacio de nombre para empaquetar correlación. El cliente
puede tener sus propios nombres de paquete. -->
    <arg value="http://webservices.unica.com/campaign/CampaignServices/
3.0=com.unica.publicapi.campaign.campaignservices.soap.v30"/>
    <arg value="-o"/> <!-- Directorio de salida -->
    <arg file="${autogen.java.dir}"/>
</java>
```

Utilización de apéndices y clases de soporte generados

El apéndice se puede utilizar de la siguiente manera:

```
CampaignServices30ServiceStub serviceStub = new
CampaignServices30ServiceStub(serviceURL);

serviceStub._getServiceClient().getOptions().setTimeOutInMilliSeconds
(webServiceTimeout); //Tiempo de espera en milisegundos.
```

La oferta se puede crear de la siguiente manera:

```
try{
    //Cambie el puerto y host para que coincida con su entorno.
    String serviceURL = "http://host:port/Campaign/services/CampaignServices30Service";
    CampaignServices30ServiceStub serviceStub = new CampaignServices30ServiceStub(serviceURL);
    long webServiceTimeout = 2*60*1000; // 2 minutos
    serviceStub._getServiceClient().getOptions().setTimeOutInMilliSeconds(webServiceTimeout); //Tiempo de espera en milisegundos.

    WSTextAttribute nameAttribute = new WSTextAttribute();
    nameAttirbute.setMetadata(null);
    nameAttirbute.setName("uacOfferDescription");
    nameAttirbute.setValues(new String[]{"description " + System.currentTimeMillis()});

    WSTextAttribute[] wsAttributes = {nameAttirbute};
    // convertir a WSAttributeArrays
    WSAttributeArrays obj = new WSAttributeArrays();
    obj.setTextAttributes(wsAttributes);
    //Cambie los valores de las variables siguientes para que coincidan con su entorno.
    String authorizationLoginName = "asm_admin"; //nombre de usuario de inicio de sesión
    String partitionName = "partition1"; //Usar su política de seguridad de Campaign
    String securityPolicyName = "Política global"; //Usar su política de seguridad de Campaign

    String offerName = "Primera oferta"; //Nombre de la oferta que se va a crear.
    String templateName = "Plantilla de oferta"; //Nombre de la plantilla de oferta existente.
    long folderID = 100; //ID real de la carpeta donde se creará esta oferta.
    //Para folderID <=0, la oferta se creará a nivel raíz.

    CreateOffer createOfferObject = new CreateOffer();
    createOfferObject.setAuthorizationLoginName(authorizationLoginName);
    createOfferObject.setPartitionName(partitionName);
    createOfferObject.setRequestedLocale(Locale.US.toString());
    createOfferObject.setSecurityPolicyName(securityPolicyName);
    createOfferObject.setName(offerName);
    createOfferObject.setFolderID(folderID);
    createOfferObject.setTemplateName(templateName);
    createOfferObject.setAttributes(obj);
    // realizar llamada a WebService de campaña
    WSCreateOfferResponse wsResponse = serviceStub.createOffer(createOfferObject);
    // estado de proceso
    WSRequestStatus status = wsResponse.getStatus();
    // fin
    WSOfferInfo offerInfo = wsResponse.getOfferInfo();    System.out.println("status = "+status.getStatusType());
    System.out.println("offerInfo = "+offerInfo.getName());
} catch (Exception exception) {
    //Manejar la excepción aquí.
    exception.printStackTrace();
}
```

En este ejemplo, createOffer() solo acepta un parámetro de tipo CreateOffer.

Con el motor AXIS2, las clases y los apéndices generados ya no tienen constructores parametrizados. Para obtener detalles, consulte “Cambios en constructores parametrizados de las clases de soporte” en la página 2.

Referencias

Las referencias siguientes se han utilizado para preparar esta especificación:

- "Basic Profile Version 1.1", Web Service Interoperability Organization (WS-I), 10 de abril de 2006. (http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-0310.html)
- "SOAP 1.2 (draft)", W3C Soap working group, 24 de junio de 2003 (http://www.w3.org/TR/soap/)
- "JAX-RPC 1.1", Sun Microsystems, 14 de octubre de 2003 (http://java.sun.com/webservices/jaxrpc/index.jsp)
- Apache Web services working group (http://ws.apache.org/axis2)

Requisitos

En esta sección se resumen los requisitos necesarios para el diseño. La API CampaignServices debe:

- Proporcionar el acceso de creación, descubrimiento, lectura y actualización preciso para los componentes de IBM Unica Campaign, a la vez que se aísla a los clientes de los detalles de implementación subyacente
- Coexistir con los usuarios basados en la GUI de IBM Unica Campaign existentes y minimizar el efecto en ellos
- Garantizar la integridad de los datos
- Soportar la arquitectura de seguridad de IBM Unica Campaign
- Soportar SOAP estándar del sector, incluyendo la autenticación segura

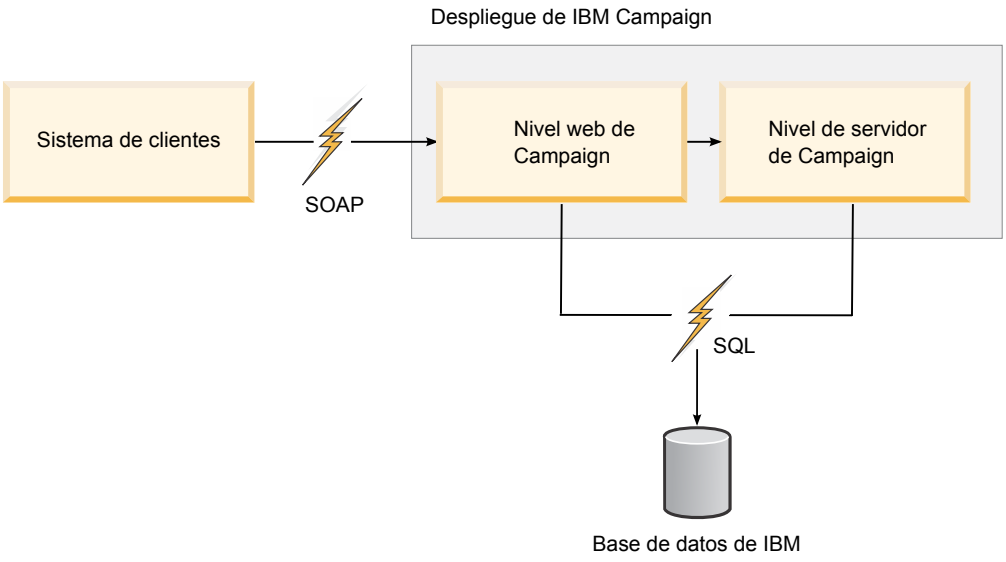
Descripción general del diseño

La API CampaignServices es una fachada que proporciona una vista de cliente de una instancia de aplicación IBM Unica Campaign en ejecución. Solo se expone un subconjunto de las prestaciones de IBM Unica Campaign, pero es suficiente para controlar los aspectos clave de la funcionalidad de Campaign. La API está diseñada para utilizarse simultáneamente con usuarios web de IBM Unica Campaign y otros subprocesos de API.

Generalmente, la API soporta los siguientes tipos de operaciones en componentes de campañas, ofertas y celdas objetivo:

- Creación de componentes
- Descubrimiento de componentes
- Supresión de componentes
- Creación, inspección y modificación de atributos de componentes y metadatos de atributos
- Captación de resultados de la ejecución de diagrama de flujo

El siguiente diagrama presenta una muestra del despliegue de CampaignServices 3.0.



Notas

En esta sección se explican los puntos concretos sobre el diseño.

Versiones y compatibilidad retroactiva

Por lo general, las versiones futuras de la API CampaignServices serán compatibles con las versiones anteriores con todos los releases de mantenimiento menores que comparten el mismo número de versión principal. Sin embargo, IBM Unica se reserva el derecho de interrumpir la compatibilidad con la versión anterior de los releases principales "punto cero" (x.0) si la empresa o cuestiones técnicas así lo justifican.

El número de versión principal de esta API se incrementará si se realiza alguno de los cambios siguientes:

- Se cambia la interpretación de datos.
- Se cambia la lógica empresarial, es decir, la funcionalidad de método de servicio.
- Se cambian los parámetros de método y/o tipos de retorno.

El número de versión menor de la API se incrementará si se realiza alguno de los cambios siguientes:

- Se añade un nuevo método.
- Se añade un nuevo tipo de datos y su uso se restringe a nuevos métodos.
- Se añade un nuevo tipo a un tipo enumerado.
- Se define una nueva versión de una interfaz.

En particular, IBM Unica continuará dando soporte al WSDL publicado, al cliente SOAP y a la versión de Apache Axis utilizada para implementar la oferta de SOAP como mínimo hasta el próximo release principal de IBM Unica . En la práctica, esto se realiza mediante el soporte simultáneo de varios servicios web específicos de la versión. (IBM Unica ya soporta varias versiones de este servicio internamente).

Autenticación de usuario

La autenticación se encarga de establecer una identidad de usuario.

Nota: Para este release, la autenticación de usuario es responsabilidad de la aplicación cliente.

Autorización de usuario

La autorización se encarga de los permisos que un usuario autenticado tiene en relación a los componentes y las operaciones expuestos por la API.

Es posible que un usuario se autentique satisfactoriamente, pero no tenga suficientes permisos para realizar algunas operaciones, como por ejemplo editar la información de resumen de una campaña. En este caso, el método de la API emitirá `AuthorizationException`.

Entorno local

Las solicitudes de API proporcionan un parámetro opcional **`requestedLocale`**, que define el entorno local que se debe utilizar para llevar a cabo esa solicitud en particular. Si no se ha definido, la API toma de forma predeterminada el entorno local preferido del usuario de IBM Unica . Se utiliza el algoritmo normal de máxima coincidencia Java para devolver mensajes y otros textos localizados en el entorno local solicitado.

Este parámetro es de tipo de clase `java.util.Locale`.

Nota: Algunos textos especificados por el usuario, por ejemplo, las descripciones de campaña estarán en el entorno local del usuario que ha especificado el texto. IBM Unica Campaign no intenta localizar estos datos. La API solo localizará los mensajes informativos, de aviso y de error.

Gestión de estado

La API CampaignServices es sin estado, lo que significa que la API no guarda información por cliente entre llamadas.

Obviamente, llamadas específicas a la API pueden cambiar el estado de instancias de componentes subyacentes gestionadas por Campaign, y estos cambios de estado pueden ser persistentes en la base de datos.

Capítulo 2. Tipos de datos

En esta sección se definen los tipos de datos públicos utilizados por la API CampaignServices.

WSReference

Un derivador simple alrededor de un ID de base de datos:

- **componentTypeEnum**: tipo enumerado que indica el tipo de componente para el que es el ID. Uno de los siguientes:
  - FOLDER
  - CAMPAIGN
  - FLOWCHART
  - TCS_CELL
  - OFFER
  - OFFER_LIST
  - OFFER_TEMPLATE
- *id*: valor *Largo*, que define un identificador numérico exclusivo específico de la base de datos para la referencia.

WSVersion

Un tipo de derivador que captura los diversos componentes de una versión, incluidos los siguientes:

- *major*: entero que define el número de versión principal, como '8' de la versión completa 8.1.2.3.
- *minor*: entero que define el número de versión menor, como '1' de la versión completa 8.1.2.3.
- *maintenance*: entero opcional que define el número de mantenimiento de la versión, si es aplicable, como '2' de la versión completa 8.1.2.3. Nunca se suministra con una versión de la API.
- *patch*: entero opcional que define el número de release del parche, si es aplicable, como '3' de la versión completa 8.1.2.3. Nunca se suministra con una versión de la API.

WSServiceInfo

Un tipo de derivador simple alrededor de información sobre el servicio. Contiene los campos siguientes:

- *apiVersion*: una instancia de *WSVersion* que define la versión más actual de la API soportada por el servicio. (Tenga en cuenta que *apiVersion* incluirá solo la información de versión principal y menor).
- *campaignVersion*: una instancia de *WSVersion* que define la versión de la instancia de IBM Unica Campaign subyacente.
- *name*: nombre interno del servicio, como "CampaignServices30Service".

WSAttributeTypeEnum

Un tipo enumerado que define todos los tipos de atributos posibles, uno de los siguientes:

- STANDARD: atributo estándar o base definido por Campaign.
- CUSTOM: atributo definido por otra aplicación de IBM Unica , el cliente o un tercero.
- INPUT_PARAMETER: parámetro de entrada, como un atributo que se utiliza para ejecutar un diagrama de flujo de IBM Unica Campaign.
- OUTPUT_PARAMETER: parámetro de salida, como por ejemplo un atributo cuyo valor se completa como resultado de una ejecución de diagrama de flujo en IBM Unica Campaign.

WSAttributeStatusEnum

Una enumeración de todos los posibles códigos de estado de atributo, uno de los siguientes:

- ACTIVE: el atributo está activo y se puede utilizar a voluntad.
- RETIRED: el atributo se ha eliminado del servicio y no debe utilizarse.

WSAccessTypeEnum

Un tipo enumerado que define todos los tipos de acceso posibles a los valores de atributos, uno de los siguientes:

- READ_ONLY: el valor del atributo se puede leer y visualizar, pero no modificar.
- READ_WRITE: el valor del atributo se puede leer, visualizar y modificar.

El acceso de atributo es aditivo a los permisos de seguridad, por lo que si por ejemplo, la política de seguridad para el usuario cliente deniega el acceso de lectura a un atributo en particular, el acceso de atributo no puede alterar temporalmente ese valor de seguridad. De hecho, la API nunca devolverá el atributo al cliente.

WSSelectTypeEnum

Define todos los tipos de selección posibles para un valor de atributo en particular, uno de los siguientes:

- NONE: ninguna selección (*hasOptions* es false).
- SINGLE_SELECT: solo se puede elegir una opción de atributo de la lista de opciones posibles cada vez (solo es válido si es un atributo *hasOptions*).
- MULTIPLE_SELECT: similar a SINGLE_SELECT, excepto que se puede seleccionar una o varias opciones a la vez.

WSRunStatusEnum

Un tipo enumerado de todos los estados posibles de ejecución de diagrama de flujo, rama o celda, uno de los siguientes:

- NOT_STARTED: la ejecución se ha planificado, pero todavía no se ha iniciado.
- RUNNING: ejecución en curso.
- CANCELLED: la ejecución se ha cancelado, por un usuario de Campaign o mediante esta API.
- SUCCEEDED: la ejecución se ha completado con éxito.

- FAILED: la ejecución ha fallado; los detalles del error se informan por separado. (Consulte "WSRunResults" en la página 15).

WSRunTypeEnum

Un tipo enumerado de todos los tipos de ejecución posibles, uno de los siguientes:

- NOT_RUN
- TEST_RUN
- PRODUCTION_RUN
- RUN_SKIPPED
- TEST_FLOWCHART
- PRODUCTION_FLOWCHART
- TEST_BRANCH
- PRODUCTION_BRANCH
- TEST_PROCESS
- PRODUCTION_PROCESS

WSAttribute

Los atributos proporcionan un mecanismo simple y ampliable para conectar datos arbitrarios a instancias de componentes accesibles a través de la API, ya sean datos estándar como el *nombre* para una campaña, los parámetros de entrada de ejecución de diagrama de flujo como el *género* o los datos personalizados arbitrarios especificados por otra aplicación de IBM Unica o cliente de IBM Unica .

Nota: En esta API, los atributos se utilizan para modelar la mayoría de los datos de componente, no solo los atributos personalizados de Campaign.

Por lo general, los componentes tendrán muchos atributos asociados, que la API CampaignServices expone como una correlación de tipo especial denominada *AttributeMap*. Los datos de atributo se representan como una clase concreta de tipo firme en toda la API, como *WSDecimalAttribute*, para los atributos que contienen datos decimales (numéricos de precisión doble).

Cada atributo incluye lo siguiente:

- *Nombre*: nombre exclusivo del atributo. Este nombre sirve como clave para acceder al atributo y sus metadatos dentro de la instancia de componente en la que aparece. El formato del nombre está sin definir; en algunos casos lo asigna el servicio, el cliente o un usuario de IBM Unica Campaign.

  Generalmente, este nombre no es el nombre de visualización que se presentará a Campaign o a un usuario cliente. La API lo puede estandarizar, como *uacDescription*, IBM Unica Campaign lo puede asignar al publicar diagramas de flujo o bien la aplicación o cliente de IBM Unica lo pueden asignar al definir los atributos personalizados. Sin embargo, en todos los casos, se garantiza que el nombre será exclusivo.
- *Metadatos*: (opcional) información sobre los datos del atributo, como el tipo de datos del valor, el nombre de visualización, la descripción, las solicitudes, el valor predeterminado, el tipo de selección, la longitud (texto), la precisión (decimales), las opciones (de selección única o múltiple), etc. Consulte "WSAttributeMetadata" en la página 12.
- *Valores*: matriz de cero o más objetos de valor de tipo firme. La clase de atributo concreta suministra el campo de valores; el tipo de cada valor debe ser el mismo

y ajustarse a la definición de tipo en el campo de metadatos del atributo. Sin embargo, no todos los atributos soportan varios valores.

Están soportados los siguientes tipos de atributos concretos:

- **WSBooleanAttribute**: atributo cuyo valor es booleano, es decir, *true* o *false*.
- **WSIntegerAttribute**: valor entero (*java.lang.Long*).
- **WSDecimalAttribute**: valor de número decimal de precisión doble (*java.lang.Double*).
- **WSCurrencyAttribute**: valor compuesto de moneda, que incluye un valor opcional de código de moneda ISO 4217, como "*USD*" para dólar americano y los valores de moneda capturados como *Doble*. Si el código de moneda no se proporciona, se presupone el valor predeterminado utilizado por IBM Unica Campaign-

  Consulte `http://www.xe.com/symbols.php` para obtener una lista de países, símbolos de moneda y códigos. Tenga en cuenta que el entorno local utilizado para una moneda puede ser diferente del entorno local preferido del usuario.
- **WSCalendarAttribute**: cuyos valores son fechas de calendario, o fechas y horas, en algún huso horario y entorno local.
- **WSTextAttribute**: una cadena de texto Unicode (posiblemente nulo o vacío).

Nota: La lista de atributos posibles es generalmente diferente para cada tipo de componente, pero las listas se pueden solapar.

WSAttributeMetadata

WSAttributeMetadata define la información sobre los datos de un atributo de tipo en particular, como un tipo de datos de valor, texto localizado (nombre de visualización, descripción, solicitudes, etc.), su valor predeterminado, el rango de valores permitidos, el tipo de selección, las opciones (de selección única o múltiple), etc. Igual que para los atributos, los metadatos de atributos son de tipo firme. Por ejemplo, WSDecimalAttribute *miNúmero* debe disponer de un enlace *DecimalAttributeMetadata* de WS y todos los valores, incluidos los valores de atributo, el valor predeterminado de metadatos y los valores de opciones posibles, serán de tipo *Doble*.

Las descripciones, las etiquetas y otro texto de metadatos de atributos están generalmente localizados; sin embargo, el texto especificado por el usuario puede que solo esté disponible tal como lo escribió el usuario. Cada llamada a la API incluye un entorno local solicitado que el código de cliente puede utilizar para definir el entorno local en el que un usuario en particular desea que se visualicen los mensajes localizados. Se utilizan las políticas de reserva de entorno local de Java normales para satisfacer la solicitud.

WSAttributeMetadata contiene los campos siguientes:

- *name*: nombre del atributo, estándar o personalizado; también el nombre utilizado por el atributo que se vincula a estos metadatos. Los atributos estándar se definen por el sistema y tienen nombres estándar en un espacio de nombres reservados (es decir, utilizan un prefijo "*uac*"), los nombres personalizados pueden utilizar otro convenio de denominación.

  Nota: El nombre de atributo debe ser exclusivo, nunca se localiza, y tiene restricciones de longitud (que dependen del contenido de caracteres y base de datos). El nombre distingue entre mayúsculas y minúsculas, y puede estar compuesto de cualquier combinación de caracteres de letras o dígitos Unicode, más el carácter de subrayado '_', pero no puede empezar por un dígito.

- *description*: descripción opcional del atributo. Adecuado para una ayuda contextual u otra presentación de interfaz de usuario.
- Predicados: diversos predicados que describen el atributo:
  - *isRequired*: true si el atributo es obligatorio.
  - *isInternal*: true si el atributo lo define el sistema y es solo para uso interno (no se debe presentar a un usuario).
  - *isGenerated*: true si los valores del atributo se generan automáticamente por IBM Unica Campaign al crear el componente, como un código de celda objetivo. Normalmente, *accessTypeEnum* será READ_ONLY para los valores generados.
  - *hasOptions*: true si el atributo tiene opciones. Implica que hay opciones definidas para estos metadatos y que *selectTypeEnum* es SINGLE_SELECT o MULTIPLE_SELECT.
- *typeEnum*: *WSAttributeTypeEnum* que define el tipo de atributo, como STANDARD o CUSTOM.
- *statusEnum: WSAttributeStatusEnum* que define el estado del atributo, como ACTIVE.
- *accessTypeEnum*: *WSAccessTypeEnum* que define el tipo de acceso para el valor de atributo, como READ_ONLY.
- *selectTypeEnum*: *WSAccessTypeEnum* que define el tipo de selección utilizada para el atributo, como SINGLE. Debe ser NONE para los atributos de campaña y celda, o si no se proporcionan opciones.
- *componentTypeEnum*: *WSComponentTypeEnum* de todos los componentes posibles de Campaign expuestos por la API, como CAMPAIGN, FOLDER, etcétera.
- *defaultValue* (solo diagramas de flujo): valor predeterminado de tipo opcional para el atributo. Este valor lo proporciona la clase concreta de metadatos de atributo, como valor predeterminado de *WSTextAttributeMetadata* de tipo Cadena. (Consulte la descripción de los valores del atributo). Para otros componentes distintos de los diagramas de flujo, el valor predeterminado está sin definir.
- *options*: lista opcional de opciones para este atributo. Combinadas, las opciones de un atributo definen el conjunto exacto de valores permitidos para ese atributo; cada opción es de tipo firme, por lo que por ejemplo *WSTextAttributeMetadata* solo puede tener una *WSTextAttributeOption* vinculada.

Nota: Hay una restricción sobre las opciones; solo se soportan atributos de texto.

Cada opción define lo siguiente:

- *prompt*: solicitud de la opción adecuada para menús desplegables, como "Masculino", como opción de atributo de género. Tenga en cuenta que, a diferencia de la solicitud de metadatos, los nombres de visualización de la opción no suelen incluir puntuación.
- *description:* descripción localizada de la opción, como "Una persona de convicción masculina". Adecuado para el texto de ayuda contextual.
- *isDefault*: true si esta opción en particular es el valor predeterminado. Para los tipos de selección MULTIPLE_SELECT, se puede marcar más de una opción como valor predeterminado.
- *value*: valor de opción de tipo. Igual que en defaultValue de los metadatos de atributos, este valor lo proporciona la subclase de opción concreta, por ejemplo, un valor de *WSDecimalAttributeOption* es de tipo Decimal. (Consulte la descripción de los valores de atributo). Continuando con el ejemplo de *género*

anterior, el valor puede declararse como una cadena (*WSTextAttributeOption*) o como código numérico, 123 (*WSDecimalAttributeOption*).

WSCampaignInfo

Un tipo de derivador simple alrededor de los datos de atributos de la campaña.

Contiene los campos siguientes:

- *reference*: referencia de la campaña.
- *name*: nombre de la campaña (*uacName*); no se garantiza que sea exclusivo.
- *description*: descripción opcional de la campaña (*uacDescription*).
- *campaignCode*: código exclusivo de la campaña (*uacCampaignCode*); asignado por el cliente o Campaign.

WSComponentOrFolderInfo

Contiene una combinación de la campaña derivada o los datos de atributos de carpeta, como el nombre de visualización, su referencia, etc.

Contiene los campos siguientes:

- *reference*: referencia del componente o carpeta.
- *name*: nombre de componente o carpeta (*uacName*); no se garantiza que sea exclusivo.
- *description*: descripción opcional del componente o carpeta (*uacDescription*).
- *componentCode*: código exclusivo del componente, o nulo si es una carpeta.

WSTargetCellInfo

Un derivador simple alrededor de datos de atributos de filas de celda objetivo.

Contiene los campos siguientes:

- *reference*: referencia de celda.
- *name*: nombre de celda (*uacName*); no se garantiza que sea exclusivo.
- *description*: descripción opcional de celda (*uacDescription*).
- *cellCode*: código de celda (*uacCellCode*); asignando por el cliente o Campaign. Tenga en cuenta que se puede forzar que los códigos de celda sean exclusivos definiendo el parámetro de configuración IBM Unica Campaign *DuplicateCellCodesAllowed* como false.
- *flowchartName*: nombre opcional del diagrama de flujo al que está vinculada la celda.

WSMetricsInfo

Un tipo de derivador simple alrededor de los datos analíticos de la campaña, incluido el número de contactos. Contiene los campos siguientes:

- *totalContacts*: valor largo que proporciona el número total de contactos.
- *respuestas*: lista de tipo de instancias de *WSMetricsResponse*, cada una de las instancias define la información de contacto para una respuesta:
  - *typeCode*: cadena que define el código de tipo de respuesta, como *PHC* para un contacto telefónico.

– *count*: valor largo que proporciona el número de veces que ha ocurrido este contacto.

WSRunResults

Un tipo de derivador alrededor de los resultados de la ejecución de un diagrama de flujo, un cuadro de proceso o una celda, posiblemente todavía en curso, que incluye el estado de la ejecución, la fecha y la hora de inicio y finalización de la ejecución del diagrama de flujo y recuentos.

Contiene los campos siguientes:

- *referenciaOrigen*: referencia opcional del origen del resultado de la ejecución. Según el contexto en el que se capten los resultados de la ejecución, puede hacer referencia a un diagrama de flujo, a un cuadro de proceso de diagrama de flujo o a una celda objetivo. En cualquier caso, los datos del resultado de la ejecución restantes hacen referencia a este origen.
- *flowchartName*: nombre del diagrama de flujo que se ha ejecutado.
- *flowchartId*: ID de base de datos para el diagrama de flujo.
- *runId*: ID de base de datos de la ejecución.
- *typeEnum*: tipo enumerado que define la clase de ejecución que ha generado los resultados, como PRODUCTION_PROCESS (consulte *WSRunTypeEnum*).
- *statusEnum*: tipo enumerado que define el estado de ejecución, como RUNNING (consulte *WSRunStatusEnum*).
- *statusCode*: código de estado entero opcional.
- *statusMessage*: mensaje de estado opcional.
- *startDate*: fecha y hora opcional del calendario de cuando se ha iniciado la ejecución; será nulo si la ejecución no se ha iniciado.
- *endDate*: igual que *startDate*, pero la fecha y hora en que ha finalizado la ejecución (éxito o error); será nulo si la ejecución no se ha iniciado o aún no ha finalizado.
- *count*: recuento total opcional de contactos seleccionados por la ejecución; puede ser cero o nulo si la ejecución no se ha completado.

WSOfferInfo

Un tipo de derivador simple alrededor de los datos de atributos de ofertas o listas de ofertas.

Contiene los campos siguientes:

- *reference*: referencia de oferta o lista de ofertas.
- *name*: nombre de oferta o lista de ofertas (*uacName*); no se garantiza que sea exclusivo.
- *description*: descripción opcional (*uacDescription*).
- *offerCode*: código de oferta (*uacOfferCode*) si es una oferta, o nulo si es una lista de ofertas. (No se garantiza que sea exclusivo.)

WSOfferCodeOrName

Un tipo de derivador simple alrededor de los datos de códigos de oferta o nombres de listas de códigos de oferta.

Contiene los campos siguientes:

- *isCode:* booleano que indica si el campo *codeOrName* es un presunto código de oferta (true) o el nombre de una lista de ofertas (false).
- *codeOrName*: código de oferta exclusivo (*uacOfferCode*) si es una oferta, o el nombre de la lista de ofertas.

WSOfferValidationInfo

Un tipo de derivador simple alrededor de la información de validación de la oferta.

Contiene los campos siguientes:

- *errorCode*: si no es nulo, define el código de error de validación alfanumérica. Consulte la clase IStandardDefinitions para ver los códigos de error.
- *errorMessage*: mensaje localizado opcional que describe el error (si se ha producido uno).
- *codeOrName*: código de oferta o nombre de lista de ofertas validados.
- *reference*: referencia de oferta o lista de ofertas, si es válida.

WSOfferTemplateInfo

Un tipo de derivador simple alrededor de los datos de plantilla de oferta.

Contiene los campos siguientes:

- *reference*: referencia de plantilla de oferta.
- *name*: nombre de plantilla de oferta; se garantiza como exclusivo.
- *description*: descripción opcional (*uacDescription*).
- *offerTemplateID*: ID de base de datos exclusivo de plantillas de ofertas.

WSBulkOfferInfo

Se utiliza para crear ofertas colectivas.

Contiene los campos siguientes:

- *offerName*: nombre de la oferta que se está creando.
- *attributes*: matriz de tipos WSAttribute que indica los atributos de la oferta.

WSOfferInfoStatus

Un tipo de retorno para el método de API *bulkCreateOffers()* que indica el estado de la creación de ofertas colectivas.

Contiene los campos siguientes:

- *name*: nombre de la oferta.
- *code*: código de la oferta. Será nulo si falla la creación de la oferta.
- *description*: descripción de la oferta.
- *reference*: WSReference de la oferta creada. Será nulo si falla la creación de la oferta.
- *status*: una instancia de WSRequestStatus que indica el estado de la creación de la oferta.

Capítulo 3. Excepciones comunes

En esta sección se explican una serie de excepciones comunes emitidas por la API CampaignServices.

Todos los mensajes de excepciones localizados están traducidos según el entorno local, si está disponible, para IBM Unica Campaign. Se aplican las políticas de reserva del entorno local normal de Java.

RemoteException

Nota: Este elemento se aplica solo a la interfaz SOAP.

Todas las llamadas de SOAP pueden emitir una *RemoteException* si se encuentra un error a nivel de sistema, como un problema en la capa de proceso de ensobrar de SOAP (Axis), una restricción definida en el WSDL del servicio web que se ha infringido por alguna razón, etc.

Las excepciones comunes de API comprobadas y no comprobadas, como *DataException*, se devolverán como un estado de error, no como una RemoteException.

Consulte la sección de la interfaz SOAP para ver los detalles.

AuthenticationException

El usuario no se puede autenticar para la partición especificada de Campaign. Compruebe el rol de usuario configurado enMarketing Platform.

AuthorizationException

El usuario no está autorizado a realizar la operación solicitada. Cualquier método de API puede emitir esta excepción, por lo que no se declara (no se selecciona). Compruebe los permisos asignados al rol de usuario en IBM Unica Marketing Platform.

DataException

Se ha producido una excepción muy grave en la capa de base de datos subyacente en Campaign (deseleccionado).

Consulte los registros del diagrama de flujo y el escucha de Campaign para ver los detalles.

LockException

Una excepción temporal emitida cuando el cliente intenta actualizar un componente, como un diagrama de flujo, mientras lo edita otro usuario. Generalmente, esta excepción se puede recuperar esperando algún tiempo y reintentando la operación. Sin embargo, reintentar la lógica es responsabilidad del cliente.

InvalidComponentException

Se ha intentado hacer referencia a un componente no válido o desconocido (campaña, diagrama de flujo, celda objetivo, etc.). El método `getComponentReference()` de la excepción devuelve la referencia del componente problemático.

InvalidAttributeException

Una excepción emitida cuando el cliente proporciona o hace referencia a un atributo no válido, como cuando utiliza un tipo de datos incorrecto o una matriz de valores en la que no hay ninguno permitido, etc. El método `getAttributeName()` de excepción devuelve el nombre del atributo del problema, `getAttributeValue()` devuelve el valor y `getComponentReference()` identifica el componente (o índice colectivo).

AttributeNotFoundException

Se genera cuando el cliente intenta hacer referencia a un atributo desconocido (campaña, diagrama de flujo, celda objetivo, etc.). El método `getAttributeName()` devuelve el nombre del atributo no coincidente; `getComponentReference()` identifica el componente (o índice colectivo).

AttributeExistsException

Se emite cuando el cliente intenta definir metadatos de un atributo duplicado para un componente. El método `getAttributeName()` de la excepción devuelve el nombre del atributo duplicado; `getComponentReference()` identifica el componente (o índice colectivo).

CompositeException

Algunas API utilizan una *CompositeException* para informar de los diversos errores al llamante. Normalmente habrá más de una causa vinculada; todas las causas se capturan como una lista en el orden en que se producen. El método `getCauseList()` de la excepción devuelve esta lista, que se puede inspeccionar para obtener los detalles de cada error.

Nota: Generalmente, la API se completará satisfactoriamente o se retrotraerá su trabajo antes de emitir una *CompositeException*. Consulte, por ejemplo, las API de hoja de cálculo de celdas objetivo colectivas descritas en "Métodos de celda objetivo" en la página 28.

Capítulo 4. Métodos de la API CampaignServices

En esta sección se definen los métodos principales exportados por la API CampaignServices 3.0.

Métodos de servicio

La API proporciona una manera de determinar información de identificación sobre el propio servicio.

getServiceInfo

```
WSServiceInfo getServiceInfo()
    throws CampaignServicesException;
```

Devuelve la información sobre el servicio, como la versión de API más reciente que soporta, la versión completa de la instancia de IBM Unica Campaign subyacente, etc.

Nota: Esta llamada no necesita ninguna información de cliente y no se aplica ningún permiso de seguridad.

Parámetros

Ninguno.

Devoluciones

Devuelve una instancia de *WSServiceInfo*.

Errores

Ninguno.

Atributos

La API expone la mayoría de datos de instancia de componente como *atributos* o *metadatos* de atributos. En algunos casos las definiciones de metadatos de atributos son globales para IBM Unica Campaign (como los atributos personalizados de campañas), mientras que en otros se limitan a un componente en particular (como las variables de usuario de diagrama de flujo).

A menos que se indique lo contrario, todos los atributos se pueden leer si el cliente tiene suficientes permisos de seguridad para hacerlo.

Nota: Esta API solo expone los componentes que están activos y a los que el cliente tiene acceso. El soporte público se limita a atributos de oferta, plantilla de oferta y metadatos.

getAttributesByName

```
Map<String, WSAttribute>
    getAttributesByName(String userCredential, String partitionName,
        Locale requestedLocale,
        WSReference reference,
        String[] names)
    throws CampaignServicesException;
```

Captura los atributos con nombre asociados a la instancia de componente especificada (puede estar vacía).

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud; si no se indica, se utilizarán las preferencias de entorno local del usuario de IBM Unica . Se aplicará el algoritmo predeterminado del entorno local normal si es necesario.

partitionName: nombre opcional de la partición de campaña que desea utilizar. Si no se ha definido, se utiliza la partición predeterminada.

reference: *Reference* para la instancia del componente que contenga los atributos deseados. Se emite una *InvalidComponentException* si la referencia no es válida o el componente no existe.

names: matriz de nombres opcional de atributos que desea captar (no nombres de visualización): si no se suministra, se devuelven todos los atributos. Se emite *AttributeNotFoundException* si uno de los atributos denominados no existe.

Devoluciones

Una correlación de tipo de cero o más atributos; el nombre de atributo es la clave de entrada de correlación y la instancia de atributo es el valor de entrada.

Errores

InvalidComponentException, AttributeNotFoundException

AuthorizationException, DataException

Nota: Todas estas excepciones se recortan dentro de la CampaignServicesException.

updateAttributes

```
void updateAttributes(String userCredential, String partitionName,
        Locale requestedLocale, WSReference reference,
        boolean allowCreate,
        WSAttribute[] attributes)
    throws CampaignServicesException;
```

Actualice uno o varios atributos de la instancia de componente con los valores de atributos suministrados.

Lógica de la actualización

La lógica de la actualización es la siguiente.

Para cada atributo contenido en la correlación de atributos suministrada:

1. Si el nombre de atributo coincide con un atributo existente, intente sobrescribir su campo *values* con el campo de valores suministrado.
2. Si el atributo aún no existe, *allowCreate* es true y sus metadatos son conocidos, cree el atributo. Esto se aplica a los metadatos de atributos globales así como a los atributos de instancia (excepto a diagramas de flujo).
3. Si el tipo de valor o algún otro aspecto de la definición de metadatos del atributo no se cumple, o uno o varios de los valores suministrados no son válidos, están fuera de rango, etc., se emite *InvalidAttributeException*.

4. Si no, se emite *AttributeNotFoundException* si el atributo denominado no existe.

Nota: En caso de una excepción, no se confirmará ninguna de las actualizaciones.

Este método en particular no soporta la definición de nuevos atributos personalizados; utilice el método `createAttributeMetadata()` para esto.

En todos los casos, la operación de actualización de atributos está sujeta a las restricciones y la validación de seguridad normales. Es responsabilidad del cliente determinar qué atributos son necesarios para una instancia de componente en particular, los tipos correctos, etc.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

reference: referencia para la instancia de componente que contiene los atributos que desea actualizar.

allowCreate: indica si se debe crear un nuevo atributo si no existe aún para el componente. (Consulte "Lógica de la actualización" en la página 20).

attributes: matriz de atributos que se desea actualizar; el nombre de atributo se utiliza para localizar el atributo que se va a actualizar y los valores nuevos se utilizan para actualizar el valor del atributo existente como un único objeto del tipo apropiado o una matriz, si es aplicable. (Consulte Capítulo 3, "Excepciones comunes", en la página 17).

Devoluciones

Ninguna.

Errores

InvalidComponentException, AttributeNotFoundException, InvalidAttributeException

AuthorizationException, DataException

getAttributeMetadataByName

```
Map<String, WSAttributeMetadata>
    getAttributeMetadataByName(String userCredential,
        StringpartitionName, Locale requestedLocale,
        WSReference reference, String[] names)
    throws CampaignServicesException;
```

Capta las definiciones de metadatos de atributos con nombre vinculadas a un componente o una plantilla en particular, o definidas globalmente.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

reference: referencia opcional para el componente o plantilla que contiene los metadatos de atributos deseados. Si solo se proporciona ComponentTypeEnum, la captación se restringe a los componentes de ese tipo; si no se suministra la referencia, la captura devolverá todas las definiciones de metadatos globales, para todos los tipos de componentes. Se emite una *InvalidComponentException* si la referencia suministrada no es válida.

names: matriz opcional de nombres de metadatos de atributos que desea captar; si no se suministra, se devuelven todos los metadatos para el componente, o se define globalmente si no se proporciona ninguna referencia. Se emite *AttributeNotFoundException* si una o varias de las definiciones de metadatos de atributos especificadas no existen.

Devoluciones

Una correlación de tipo de cero o más definiciones de metadatos de atributo ; el nombre de atributo es la clave de entrada de correlación y los metadatos de atributos es el valor de entrada.

Errores

InvalidComponentException, AttributeNotFoundException

AuthorizationException, DataException

createAttributeMetadata

```
void createAttributeMetadata(String userCredential,
        String partitionName,
        Locale requestedLocale, WSReference reference,
        WSAttributeMetadata[] attributeMetadata)
    throws CampaignServicesException;
```

Cree una o varias definiciones de metadatos de atributos nuevas y, opcionalmente, enlácelas a un componente o plantilla en particular.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

reference: referencia opcional para el componente o la plantilla a los que desea vincular los metadatos. Si no se proporciona, la definición de metadatos creada será global. Si se proporciona la referencia, pero no es válida, se emite una *InvalidComponentException*.

attributeMetadata: matriz de definiciones de metadatos de atributos que desea vincular. Si uno o varios de los metadatos especificados ya están vinculados al componente, es decir, el nombre no es exclusivo, se emite una *AttributeExistsException*. Se emite una *InvalidAttributeException* si hay un problema con uno o varios de los metadatos especificados, es decir, es internamente incoherente.

Devoluciones

Ninguna.

Errores

InvalidComponentException, AttributeExistsException, InvalidAttributeException

AuthorizationException, DataException

updateAttributeMetadata

```
void updateAttributeMetadata(String userCredential,
        String partitionName,
        Locale requestedLocale, WSReference reference,
        boolean allowCreate,
        WSAttributeMetadata[] attributeMetadata)
    throws CampaignServicesException;
```

Actualice una o varias definiciones de metadatos de atributos del componente o plantilla especificados, creando opcionalmente nuevas definiciones de metadatos si es necesario.

Lógica de la actualización

La lógica de la actualización es la siguiente.

Para cada definición de metadatos de atributos contenida en la matriz suministrada:

1. Si el nombre de atributo no coincide con metadatos existentes vinculados al componente, haga lo siguiente según el valor del parámetro *allowCreate*:
   a. *True*: cree una nueva definición de metadatos. Es funcionalmente idéntico a utilizar la solicitud `createAttributeMetadata()`.
   b. *False*: emita *AttributeNotFoundException*.
2. Si el tipo de datos de metadatos de atributos es diferente, se emite *InvalidAttributeException*.
3. Intente sobrescribir la definición de metadatos de atributos existente con los valores de campo de los metadatos suministrados, si no se emite *InvalidAttributeException*. Solo están soportadas las siguientes actualizaciones (si no se emite *InvalidAttributeException*):
   a. *name*: no se puede cambiar (el nombre es la clave).
   b. *displayName*: aceptar el nuevo valor.
   c. *description*: aceptar el nuevo valor.
   d. *isRequired*: solo permite cambiar de *true* a *false*.
   e. *isInternal*: aceptar el nuevo valor.
   f. *isGenerated*: no se permite ningún cambio.
   g. *attributeTypeEnum*: no se permite ningún cambio.
   h. *accessTypeEnum*: aceptar el nuevo valor.
   i. *selectTypeEnum*: aceptar estas transiciones si se proporcionan opciones:
      1) NONE a SINGLE_SELECT o MULTIPLE_SELECT
      2) SINGLE_SELECT a MULTIPLE_SELECT
   j. *options*: se pueden añadir opciones, pero no suprimir. Solo los siguientes cambios de opciones están soportados (según la coincidencia de valores):
      1) *displayName*: aceptar el nuevo valor (sin ajuste)
      2) *description*: aceptar el nuevo valor (sin ajuste)
      3) *isDefault*: aceptar el nuevo valor; aunque debe coincidir con SelectTypeEnum.
      4) *value*: no se permite ningún cambio (el valor es la clave).

k. *defaultValue* (solo diagramas de flujo): aceptar el nuevo valor predeterminado.
l. *maximumLength* (solo texto): aceptar la nueva longitud si es mayor.

4. Si la definición de metadatos de atributo no es coherente internamente, se emite *InvalidAttributeException*.
5. Si es necesario, busque todas las instancias de componentes que hagan referencia a los metadatos de atributos actualizados y actualícelas según proceda.

Nota: En caso de una excepción, no se confirmará ninguna de las actualizaciones.

En todos los casos, la operación de actualización de atributos está sujeta a las restricciones y la validación de seguridad normales.

See `createAttributeMetadata()`, `deleteAttributeMetadata()`

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

reference: referencia opcional para la instancia de componente que contiene los atributos deseados. Si no se suministra, la actualización se restringirá a las definiciones de metadatos globales. Se emite una *InvalidComponentException* si la referencia suministrada no es válida.

allowCreate: si es true, se crearán las definiciones de metadatos que no existen actualmente (es funcionalmente equivalente a utilizar el método `createAttributeMetadata()`).

attributeMetadata: una matriz de definiciones de metadatos de atributos que se debe actualizar (y añadir si el indicador *allowCreate* es true); el nombre de atributo se utiliza para localizar la definición de metadatos que desea actualizar y los datos restantes se utilizan para actualizar la definición existente. (Consulte "Lógica de la actualización" en la página 23).

Devoluciones

Ninguna.

Errores

InvalidComponentException, InvalidAttributeException

AuthorizationException, DataException

deleteAttributeMetadata

```
void deleteAttributeMetadata(String userCredential,
        String partitionName,
        Locale requestedLocale, WSReference reference,
        String[] names)
    throws CampaignServicesException;
```

Suprime una o varias definiciones de metadatos de atributos con nombre del componente especificado, la plantilla (solo metadatos de atributos personalizados) o las definiciones de metadatos de atributos.

Como parte de esta tarea, el método encontrará todos los componentes que hacen referencia a los metadatos suprimidos y los actualizará según sea apropiado.

Nota: Sin embargo, en el caso de una excepción, ninguna de las supresiones se confirmará.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

reference: referencia opcional del componente o plantilla que contiene los atributos que desea suprimir. Si no se suministra, la supresión estará restringida a la definiciones de metadatos globales. Se emite una *InvalidComponentException* si la referencia suministrada no es válida.

Nota: Si la matriz de nombres opcional de los metadatos de atributo no se suministra, este método intentará suprimir todos los metadatos de atributo personalizados asociados con el componente, o todas las definiciones globales si no se ha proporcionado la referencia.

names: matriz de nombres opcional de los metadatos de atributo que desea suprimir. Se emite *AttributeNotFoundException* si uno o más de los metadatos de atributos no existen. Se emite una *InvalidAttributeException* si un atributo no se ha podido eliminar.

Devoluciones

Ninguna.

Errores

InvalidComponentException, AttributeNotFoundException, InvalidAttributeException

AuthorizationException, DataException

Campañas

La API soporta las siguientes operaciones en campañas (sujetas a permisos de seguridad):

- Creación de una nueva campaña
- Descubrimiento (listado de campañas por diversos criterios)
- Creación, lectura y actualización de atributos (mediante las API de atributos).

Las campañas tienen un número de atributos estándar asociados que las API exponen. El cliente puede ampliar esta lista a voluntad añadiendo atributos personalizados (consulte las API de atributos).

Los atributos de campaña estándar son:

- *uacName*: nombre de campaña (no se garantiza que sea exclusivo).
- *uacDescription*: cadena opcional que describe la campaña.
- *uacCampaignCode*: código de cadena que identifica de forma exclusiva la campaña. Normalmente la campaña lo genera de forma automática, pero también lo puede proporcionar el cliente.

- *uacCreateDate*: calendario que define la fecha y la hora en que el servidor ha creado la campaña.
- *uacUpdateDate*: calendario que indica la fecha y la hora en que el servidor ha actualizado la campaña por última vez.
- *uacInitiative*: cadena opcional que define la iniciativa de la campaña.
- *uacObjectives*: cadena opcional que identifica los objetivos de la campaña.
- *uacStartDate*: calendario opcional que proporciona la fecha y la hora en que el servidor ha iniciado la campaña, o que está planificado que se inicie.
- *uacEndDate*: como *uacStartDate*, pero define la fecha y la hora en que se ha completado la campaña o que está planificado que se complete. Debe estar después de *uacStartDate*.
- *uacLastRunDate*: calendario opcional que proporciona la fecha y la hora en que el diagrama de flujo vinculado a la campaña se ha ejecutado por última vez (de lo contrario es nulo).
- *uacExternalLinkOwner*: cadena opcional que define el nombre del propietario de un enlace externo (consulte el atributo uacExternalLinkReference). Solo para uso de IBM Unica ; debe ser uno de los siguientes:
  - "Plan" (ahora denominado IBM Unica Marketing Operations)
  - "Collaborate" (ahora denominado IBM Unica Distributed Marketing)
- *uacExternalLinkId*: identificador de base de datos numérico opcional asignado por otra aplicación de IBM Unica a un objeto enlazado a esta campaña. Solo para uso de IBM Unica : consulte también el atributo uacExternalLinkOwner.

generateCampaignCode

```
String generateCampaignCode(String userCredential,
    String partitionName,
    Locale requestedLocale);
```

Genere un nuevo código de campaña.

Se garantiza que este código es exclusivo y diferente del valor devuelto por una llamada anterior o futura a este método, el método `createCampaign()` o el valor generado para una campaña creada mediante la GUI de IBM Unica Campaign.

Nota: El uso de este método es opcional, ya que la API `createCampaign()` generará un código de campaña para el cliente si no se suministra uno.

Consulte `createCampaign()`.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar. Si solo hay una partición en la instalación de Campaign, el argumento puede ser nulo.

Devoluciones

El código de campaña generado.

Errores

AuthorizationException, DataException

createCampaign

```
CampaignInfo createCampaign(String userCredential,
        String partitionName,
        Locale requestedLocale,
        String securityPolicyName,
        String name, Attribute[] attributes)
    throws InvalidFolderException, AttributeNotFoundException,
        InvalidAttributeException;
```

Cree una nueva campaña para el cliente, la partición y securityPolicyName, aplicando los atributos especificados. Todas las campañas creadas por esta API estarán ubicadas en la carpeta raíz.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

securityPolicyName: nombre opcional de la política de seguridad de la campaña que desea utilizar para crear la campaña. Todas las operaciones subsiguientes de esta campaña utilizarán esta política. Si no se ha definido, se utilizará la política global.

name: nombre que desea asignar a la nueva instancia de campaña (su atributo "*uacName*").

attributes: una matriz opcional de atributos de inicialización; los atributos suministrados sobrescribirán los valores predeterminados de la campaña, otros se dejarán intactos. Por ejemplo, si se proporciona un atributo *uacCampaignCode*, se utilizará en lugar de otro generado automáticamente. Es responsabilidad del cliente determinar los atributos necesarios por la campaña, sus tipos, etc.

Se emite una *AttributeNotFoundException* si uno o varios de los atributos denominados no existen o *InvalidAttributeException* si un valor de atributo no es válido (como un tipo de datos incorrecto).

Devoluciones

Una única instancia de CampaignInfo para la campaña creada.

Errores

InvalidAttributeException, AttributeNotFoundException

AuthorizationException, DataException

listCampaignsByPage

```
List<CampaignInfo>
    listCampaignsByPage(String userCredential, String partitionName,
        Locale requestedLocale, Attribute[] attributes,
        long pageOffset, int pageSize)
    throws AttributeNotFoundException, InvalidAttributeException,
        RangeException;
```

Enumere una "página" de campañas que coincida con los valores de atributos opcionales, empezando por el desplazamiento de página especificado. Las carpetas se ignoran.

Una vez recuperadas, las *CampaignInfo* devueltas se pueden utilizar tal cual, por ejemplo, para visualizar una lista de resumen, o los métodos de atributos se pueden utilizar para captar o actualizar los atributos de la campaña.

Esta API no mantiene ningún estado, por lo que se puede llamar en cualquier orden.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

attributes: matriz de atributos opcionales que deben coincidir; el nombre del atributo, el tipo de datos y los valores se utilizan para determinar la coincidencia; si el atributo soporta matrices, todos los valores especificados deben coincidir. El operador de coincidencia implícito es Y, por lo que solo se devolverán las campañas que coincidan con todos los valores de atributos suministrados.

Se emite *AttributeNotFoundException* si un nombre de atributo no existe o *InvalidAttributeException* si uno o varios de los atributos suministrados no son válidos.

pageOffset: desplazamiento inicial de todas las campañas posibles en el que debe empezar la enumeración (con valor cero). Por ejemplo, si la enumeración coincide con 1000 campañas y este valor se define como 10, la página empezaría en el componente número 11. Se emite una *RangeException* si el desplazamiento suministrado está fuera de rango.

pageSize: número máximo de campañas coincidentes que se deben devolver para la página (no puede exceder de 500).

Devoluciones

Una lista de tipo de cero o más instancias de derivador de datos *CampaignInfo*, una por cada campaña con coincidencia en la página.

Errores

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

Métodos de celda objetivo

Las celdas objetivo son una abstracción para algunos subconjuntos conocidos de resultados de campañas que gestiona IBM Unica Campaign como una hoja de cálculo de celdas objetivo (TCS). Las celdas objetivo pueden ser globales para una campaña o estar asociadas a un diagrama de flujo de campaña en particular.

La API soporta las siguientes operaciones en celdas objetivo:

- Creación de una o más celdas objetivo globales nuevas
- Actualización colectiva de una o varias celdas objetivo existentes
- Descubrimiento (listado de celdas objetivo)
- Creación, lectura y actualización de atributos (mediante las API de atributos)
- Supresión de una celda objetivo existente

- Captación de resultados de ejecución asociados a una o varias celdas

Las celdas objetivo tienen un número de atributos estándar asociados que la API expone. El cliente puede ampliar esta lista a voluntad añadiendo definiciones de metadatos de atributos personalizados (consulte las API Attributes). Todos los metadatos de atributo se puede considerar como una columna de la TCS; el diseño de la hoja de cálculo depende del cliente.

Los atributos de celda objetivo estándar son:

- *uacName*: nombre de celda.
- *uacDescription*: cadena opcional que describe el diagrama de flujo.
- *uacCellCode*: cadena de código que identifica de forma exclusiva la celda. Normalmente Campaign lo genera automáticamente, pero también lo puede proporcionar el cliente.
- *uacCreateDate*: instancia de calendario que proporciona la fecha y la hora en que el servidor ha creado la celda.
- *uacUpdateDate*: instancia de calendario que define la última vez que el servidor ha actualizado la celda.
- *uacIsControl*: booleano que indica si es una celda de control (true) o no (false). Otras celdas pueden hacer referencia a esta celda como celda de control (consulte *uacControlCell*).
- *uacControlCell*: referencia opcional de la celda de control (no permitido si es una celda de control). Consulte el atributo *uaclsControl*.
- *uacIsApproved*: booleano que indica si la celda se ha aprobado (true) o no (false).
- *uacIsReadOnly*: booleano que indica si la celda es de solo lectura (true) o no (false).
- *uacDisplayOrder*: entero que proporciona el orden de esta celda (fila) en relación a otras de la hoja de cálculo de celdas objetivo.
- *uacIsTopDown*: booleano que indica si la celda es de arriba a abajo.
- *uacAssignedOffers*: matriz opcional de una o más referencias de ofertas o listas de ofertas asignadas a esta celda (no permitido si es una celda de control).
- *uacFlowchartName*: nombre opcional de diagrama de flujo al que está enlazada esta celda (de solo lectura, debe establecerse mediante la GUI de IBM Unica Campaign; no está permitido si es una celda de control).
- *uacFlowchartId*: identificador de base de datos opcional para el diagrama de flujo al que está enlazada esta celda (de solo lectura como antes).

createTargetCell

```
TargetCellInfo
    createTargetCell(String userCredential, String partitionName,
        Locale requestedLocale,
        Reference campaignReference,
        Attribute[] attributes)
    throws InvalidComponentException, CompositeException;
```

Cree una nueva fila de celdas objetivo específica de la campaña, aplicando los atributos por celdas especificados y la información de usuario.

Los atributos especificados pueden ser estándar o personalizados; sin embargo, si son personalizados, las definiciones de metadatos de atributos globales correspondientes ya deben existir.

Una vez creada la celda objetivo, los valores de atributos se pueden cambiar utilizando las API de atributos.

Consulte `listTargetCells()`, `bulkCreateTargetCells()`.

Consulte `createAttributeMetadata()`, `listAttributeMetata()`, `getAttributesByName()`

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

campaignReference: referencia de la campaña que contiene la hoja de cálculo de celdas objetivo que desea actualizar. Se acumula una *InvalidComponentException* si la campaña no existe o si la referencia no es válida.

attributes: matriz opcional de atributos de TCS para la nueva celda. Cada elemento de atributo proporcionado sobrescribirá los valores predeterminados de atributos de celda correspondientes; otros se dejarán intactos. Es responsabilidad del cliente determinar qué atributos necesita la celda, sus tipos, etc. Se acumula una *InvalidAttributeException* si hay un problema con un atributo especificado.

Si se han acumulado excepciones, este método emitirá una *CompositeException* y se desharán todas las creaciones. La lista de causas de excepción incluirá una excepción para cada atributo que ha causado el error y contendrá un índice numérico en lugar de *reference*, el nombre de atributo y normalmente el valor problemático. La lista de causas estará ordenada como en la entrada *attributeList*.

Devoluciones

Un derivador de datos *TargetCellInfo* para la celda de TCS creada.

Errores

InvalidComponentException, CompositeException

AuthorizationException, DataException

bulkCreateTargetCells

```
List<TargetCellInfo>
        bulkCreateTargetCells(String userCredential,
        String partitionName,
        Locale requestedLocale,
        Reference campaignReference,
        List<Attribute[]> attributesList)
    throws InvalidComponentException, CompositeException;
```

Cree muchas filas de celdas objetivo específicas de la campaña a la vez, aplicando los atributos por celda especificados y la información de usuario.

Los atributos especificados pueden ser estándar o personalizados; sin embargo, si son personalizados, las definiciones de metadatos de atributos globales correspondientes ya deben existir.

Una vez creada la celda objetivo, los valores de atributos se pueden cambiar utilizando las API de atributos.

Consulte `listTargetCells()`.

Consulte `createAttributeMetadata()`, `listAttributeMetata()`, `getAttributesByName()`

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

campaignReference: referencia de la campaña que contiene la hoja de cálculo de celdas objetivo que desea actualizar. Se acumula una *InvalidComponentException* si la campaña no existe o si la referencia no es válida.

attributeList: lista opcional de matrices de atributos por celda, una para cada fila de celdas objetivo que desea crear. Los atributos suministrados para un determinado elemento de la lista sobrescribirán los valores predeterminados de atributos de celda correspondientes; otros se dejarán intactos. Es responsabilidad del cliente determinar qué atributos necesita la celda, sus tipos, etc. Se acumula una *InvalidAttributeException* si hay un problema con un atributo especificado.

Si se han acumulado excepciones, este método emitirá una *CompositeException* y se desharán todas las creaciones. La lista de causas de la excepción incluirá una excepción para cada atributo que ha causado el error e incluirá un índice numérico en lugar de *reference* y el nombre del atributo, etc. La lista de causas se ordenará como con la entrada *attributeList*.

Devoluciones

Una lista de derivadores de datos de *TargetCellInfo*, uno por cara instancia creada, ordenados según el orden de elementos del parámetro *attributesList* de entrada.

Errores

InvalidComponentException, CompositeException

AuthorizationException, DataException

listTargetCells

```
List<TargetCellInfo>
        listTargetCells(String userCredential,
        Reference campaignReference, Locale requestedLocale,
        Attribute[] attributes)
    throws InvalidComponentException, InvalidAttributeException;
```

Lista información acerca de todas las celdas objetivo que existen actualmente que coinciden con los atributos especificados, ya sea para la campaña especificada o bien globalmente si no se especifica ninguna campaña.

Consulte `getAttributeMetadata()`, `getAttributesByName()`.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

campaignReference: referencia de la campaña padre. Se emite *InvalidComponentException* si la campaña no existe o la referencia no es válida.

attributes: matriz opcional de atributos que deben coincidir. El operador de coincidencia implícito es Y, por lo que solo se devolverán las celdas que coincidan con todos los valores de atributos suministrados.

Se emite *InvalidAttributeException* si uno o varios de los atributos especificados no es válido.

Devoluciones

Devuelve una lista de cero o más instancias de *TargetCellInfo* para las celdas con coincidencia.

Errores

InvalidComponentException, InvalidAttributeException

AuthorizationException, DataException

bulkUpdateTargetCells

```
void bulkUpdateTargetCells(String userCredential,
        String partitionName,
        Locale requestedLocale,
        Map<Reference, Attribute[]> attributesMap)
    throws CompositeException;
```

Actualiza los atributos de una o varias celdas objetivo.

La lógica de la actualización es la siguiente.

Para cada elemento proporcionado en *attributesMap*, la clave de entrada es la referencia de la celda objetivo que se va a actualizar y el valor de entrada es una matriz de atributos de actualización para esa celda. Si la celda objetivo no existe, se acumula una *InvalidComponentException*.

Una vez localizada una celda objetivo, haga lo siguiente para cada atributo especificado:

1. Si el nombre de atributo coincide con un atributo existente, intente sobrescribir su campo de valores con el campo de valores suministrado.
2. Si el tipo de valor o algún otro aspecto de la definición de metadatos del atributo no se cumple, o uno o más de los valores suministrados no son válidos, están fuera de rango, etc., se acumula una *InvalidAttributeException*.
3. Si el atributo denominado no existe, se acumula una *AttributeNotFoundException*.

Si se han acumulado excepciones, este método emitirá una *CompositeException* y se desharán todas las actualizaciones. La lista de causas de la excepción incluirá las excepciones listadas más arriba. Para cada atributo que ha causado el error, se registrarán la referencia y el nombre de atributo.

En todos los casos, la operación de actualización de atributos está sujeta a las restricciones y la validación de seguridad normales. Es responsabilidad del cliente determinar qué atributos son necesarios para una instancia de componente determinada, los tipos correctos, etc.

Parámetros
userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

attributesMap: correlación de celdas objetivo que desea actualizar; la clave de entrada es la referencia de la celda que se va a actualizar y el valor de entrada es una matriz de atributos de actualización. El nombre de atributo se utiliza para localizar el atributo que se va a actualizar y los nuevos valores se utilizan para actualizar el valor del atributo existente como un único objeto del tipo apropiado o una matriz, si es aplicable. Consulte las excepciones anteriores.

Devoluciones
Ninguna.

Errores
ComponentException

AuthorizationException, DataException

getRunResultsByCell

```
List<RunResults>
    getRunResultsByCell(String userCredential, String partitionName,
        Locale requestedLocale,
        Reference[] cellReferences)
    throws InvalidComponentException;
```

Obtenga los resultados de la ejecución de una a o varias celdas objetivo, posiblemente para un diagrama de flujo que no se ha iniciado nunca o que todavía está en curso.

Parámetros
userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

cellReferences: matriz de referencias de las celdas objetivo cuyos resultados de ejecución se desean. Se emite una *InvalidComponentException* si una o varias referencias de celda no son válidas o hacen referencia a una celda que no existe.

Devoluciones
Devuelve una lista de tipo de resultados de ejecución para las celdas denominadas, ordenadas según la matriz de referencias de entrada.

Cada estado de ejecución será RUNNING si el cuadro de proceso de diagrama de flujo sigue en ejecución, FAILED si la ejecución ha fallado por alguna razón o NOT_STARTED si el cuadro de proceso no se ha iniciado. También se proporcionan los detalles de estado.

Errores
InvalidComponentException

AuthorizationException, DataException

bulkDeleteTargetCells

```
void bulkDeleteTargetCells(String userCredential,
        String partitionName,
        Locale requestedLocale,
        Reference[] cellReferences)
    throws CompositeException;
```

Suprime una o varias celdas objetivo existentes y todos los componentes dependientes (es decir, enlace de diagrama de flujo, atributos, etc.).

Parámetros
userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

cellReferences: matriz o una o más referencias de celdas que desea suprimir. Se acumula una *InvalidComponentException* si hay un problema con una de las referencias especificadas, o una celda no existe.

Si se han acumulado excepciones, este método emitirá una *CompositeException* y se desharán todas las supresiones. La lista de causas de la excepción incluirá las excepciones listadas más arriba. Se registrará la referencia para cada celda que ha causado el error.

Devoluciones
Ninguna.

Errores
CompositeException

AuthorizationException, DataException

Analytics

La API soporta la recuperación de métricas simples de IBM Unica Campaign.

getCampaignMetrics

```
MetricsInfo getCampaignMetrics(String userCredential,
        String partitionName,
        Locale requestedLocale,
        Reference campaignReference)
    throws InvalidComponentException;
```

Captura las métricas de la campaña especificada.

Parámetros
userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

campaignReference: referencia de la campaña padre. Se emite *InvalidComponentException* si hay un problema con la referencia de campaña o la campaña no existe.

Devoluciones

Devuelve una instancia de *MetricsInfo* para la campaña.

Errores

InvalidComponentException

AuthorizationException, DataException

Métodos de oferta, lista de ofertas y plantilla de oferta

La API soporta las siguientes operaciones en las ofertas:

- Descubrimiento (listado por carpeta (ofertas, listas de ofertas y subcarpetas), atributo (ofertas y plantillas de ofertas) o valor de búsqueda (ofertas))
- Validación
- Recuperación de información (recuperación de atributos para una oferta específica o plantilla de oferta)
- Creación, edición, retiro y supresión de ofertas

Las ofertas tienen un número de atributos estándar asociados. El cliente puede ampliar esta lista a voluntad añadiendo definiciones de metadatos de atributos personalizados (consulte las API Attributes).

Los atributos de ofertas estándar son:

- *uacName*: nombre de oferta.
- *uacDescription*: cadena opcional que describe la oferta.
- *uacOfferCode*: cadena de código que identifica de forma exclusiva la oferta. Normalmente IBM Unica Campaign lo genera automáticamente, pero también lo puede proporcionar el cliente.
- *uacCreateDate*: instancia de calendario que proporciona la fecha y la hora en que el servidor ha creado la oferta.
- *uacUpdateDate*: instancia de calendario que define cuándo el servidor ha actualizado la oferta por última vez.

Las plantillas de ofertas también tienen atributos estándar y personalizados. Los atributos de plantilla de oferta estándar son:

- *uacName*: nombre de plantilla de oferta.
- *uacDescription*: cadena opcional que describe la plantilla de oferta.
- *uacCreateDate*: instancia de calendario que proporciona la fecha y la hora en que el servidor ha creado la plantilla de oferta.
- *uacUpdateDate*: instancia de calendario que define cuándo el servidor ha actualizado la plantilla de oferta por última vez.

listOffersAndFolders

```
List<WSComponentOrFolderInfo>
    listOffersAndFolders(String userCredential, String partitionName,
        Locale requestedLocale,
        WSReference parentReference)
    throws CampaignServicesException;
```

Liste todas las ofertas, las listas de ofertas y las carpetas bajo la carpeta padre opcional.

Una vez recuperadas, las instancias de *WSComponentOrFolderInfo* devueltas se pueden utilizar tal cual, por ejemplo, para visualizar el siguiente nivel de la jerarquía de carpetas, las API de atributo se pueden utilizar para captar o actualizar cualquier oferta que contenga.

Parámetro

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

parentReference: referencia opcional de la carpeta padre que desea listar. Solo se enumerarán las ofertas hijo, las listas de ofertas y las carpetas inmediatas de esta carpeta padre, por lo que se necesitarán llamadas sucesivas a esta API para navegar por toda la jerarquía de la carpeta (sin embargo, normalmente será muy poco profunda). Si no se suministra ningún padre, se devuelven todos los componentes y las carpetas bajo la raíz.

Se emite *InvalidFolderException* si hay un problema con la referencia de carpeta padre especificada.

Una *Lista* de tipo de cero o más instancias de derivador de datos *WSComponentOrFolderInfo*, una por cada componente o carpeta con coincidencia.

Errores

InvalidFolderException

InvalidExecutionContextException, AuthorizationException

searchOffersBasic

```
List<WSOfferInfo>
    searchOffersBasic(String userCredential, Locale requestedLocale,
        String partitionName, long folderID,
        String searchCriteria, boolean includeRetired,
        int pageOffset, int pageSize)
    throws CampaignServicesException;
```

Enumere una "página" de ofertas que contengan los criterios de búsqueda proporcionados en los campos de nombre, descripción, creada por o código de oferta, empezando por el desplazamiento de página especificado. La búsqueda se basa en la entrada de carpeta opcional. (Si se proporciona un folderID de 0, la carpeta de ofertas raíz se utiliza de forma predeterminada). Se devuelven las coincidencias en función de una coincidencia "contiene" para la cadena de búsqueda.

Una vez recuperadas, las *WSOfferInfo* devueltas se pueden utilizar tal cual, por ejemplo para visualizar una lista de resumen, o los métodos de atributos se pueden utilizar para captar o actualizar los atributos de la oferta.

Esta API no mantiene ningún estado, por lo que se puede llamar en cualquier orden.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

folderID: ID de la carpeta de ofertas en la que se debe buscar; si se especifica un folderID de 0, se buscará en la carpeta raíz.

searchCriteria: frase de búsqueda.

includeRetired: valor booleano que especifica si los resultados de la búsqueda incluirán las ofertas retiradas. Los valores válidos son TRUE y FALSE, TRUE indica que se incluyen las ofertas retiradas y FALSE indica que las ofertas retiradas no se incluyen.

pageOffset: desplazamiento inicial de todos los componentes posibles en el que debe empezar la enumeración (con valor cero). Por ejemplo, si la enumeración coincide con 1000 ofertas y este valor se establece en 10, la página empezaría en el componente número 11. Se emite una RangeException si el desplazamiento suministrado está fuera de rango.

pageSize: número máximo de componentes coincidentes que se deben devolver para la página (no puede exceder de 500).

Devoluciones

Devuelve una lista de tipo de cero o más instancias de derivador de datos *Offer*, uno para cada oferta devuelta en la página.

Errores

RangeException

listOffersByPage

```
List<OfferInfo>
    listOffersByPage(String userCredential, String partitionName,
        Locale requestedLocale, Attribute[] attributes,
        long pageOffset, int pageSize)
    throws AttributeNotFoundException, InvalidAttributeException,
        RangeException;
```

Enumere una "página" de ofertas que coincidan con los valores de atributos opcionales, empezando por el desplazamiento de página especificado. Las carpetas se ignoran. Las coincidencias se devuelven en función de una coincidencia “similar” para las cadenas (donde la coincidencia se considera suficiente si una cadena contiene el valor consultado), y una coincidencia exacta para las fechas y números.

Una vez recuperadas, las *OfferInfo* devueltas se pueden utilizar tal cual, por ejemplo, para visualizar una lista de resumen, o los métodos de atributo se pueden utilizar para captar o actualizar los atributos de la oferta.

Esta API no mantiene ningún estado, por lo que se puede llamar en cualquier orden.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

attributes: matriz de atributos opcionales que deben coincidir; el nombre del atributo, el tipo de datos y los valores se utilizan para determinar la coincidencia; si el atributo soporta matrices, todos los valores especificados deben coincidir. El operador de coincidencia implícito es O, por lo que se devolverán todos los componentes que coincidan con alguno de los valores de atributos suministrados.

Emite una *AttributeNotFoundException* si un nombre de atributo no existe o InvalidAttributeException si uno o varios de los atributos suministrados no es válido.

pageOffset: el desplazamiento inicial de todos los componentes posibles en que debe empezar la enumeración (con valor cero). Por ejemplo, si la enumeración coincide con 1000 ofertas y este valor se establece en 10, la página empezaría en el componente número 11. Se emite una *RangeException* si el desplazamiento suministrado está fuera de rango.

pageSize: número máximo de componentes coincidentes que se deben devolver para la página (no puede exceder de 500).

Devoluciones

Una lista de tipo de una o más instancias de derivador de datos *OfferInfo*, una por cada componente con coincidencia en la página.

Errores

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

validateOffers

```
List<OfferValidationInfo>
    validateOffers(String userCredential, String partitionName,
    Locale requestedLocale,
    OfferCodeOrName[] codeOrNames);
```

Valide los códigos de ofertas o los nombres de la lista de ofertas suministrados y la información de validación de retorno para cada uno de ellos. La “validación” consiste en la comprobación de si existe una y solo una oferta o lista de ofertas que coincide en la base de datos.

El objeto OfferValidationInfo contiene un mensaje de error en lugar de la información de oferta si se encuentran cero ofertas o listas de ofertas que coinciden con el código o nombre correspondiente. También se devuelve un error en lugar de

una coincidencia si el código o el nombre dado coincide con varias ofertas y listas de ofertas. La lista se devuelve en el mismo orden dado. Los códigos de oferta y los nombres de la lista de ofertas se validan en función de la coincidencia exacta con ofertas.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

codeOrNames: matriz de todos los códigos de oferta o nombres de la lista de ofertas que desea validar.

Nota: No se emiten excepciones por este método; en su lugar se devuelve información de validación para todos los códigos o nombres suministrados.

Devoluciones

Una lista de tipo de cero o más instancias de derivador de datos *OfferValidationInfo*.

Errores

Ninguno.

createOffer

```
OfferInfo createOffer(String userCredential, String partitionName,
        Locale requestedLocale,
        String securityPolicyName,
        String name, String templateName,
        Attribute[] attributes)
    throws InvalidFolderException, AttributeNotFoundException,
        InvalidAttributeException;

public WSOfferInfo createOffer(String authorizationLoginName, String
        partitionName, Locale requestedLocale, String
        securityPolicyName, String name, long folderID,
        String templateName, WSAttribute[] wsAttributes)
    throws CampaignServicesException;
```

Cree una nueva oferta para el cliente, aplicando los atributos especificados.

Parámetros

authorizationLoginName: nombre de usuario que crea la oferta. Se debe otorgar a los usuarios el permiso Añadir ofertas para utilizar este método.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

securityPolicyName: nombre opcional de la política de seguridad de la campaña que desea utilizar para crear la oferta. Todas las operaciones subsiguientes de esta oferta utilizarán esta política. Si no se ha definido, se utilizará la política *Global*.

name: nombre que desea asignar a la nueva instancia de oferta (su atributo *uacName*).

folderID: ID de la carpeta de ofertas donde se creará la oferta. Se validará si este ID es correcto y se emitirá una excepción si el ID no es válido.

templateName: nombre necesario (exclusivo) de una plantilla de oferta existente que desea utilizar para la nueva oferta.

wsAttributes: una matriz de atributos de inicialización; los atributos suministrados sobrescribirán los valores predeterminados de la oferta; otros se dejarán intactos. Por ejemplo, si se proporciona un atributo *uacOfferCode*, se utilizará en lugar de uno generado automáticamente. Es responsabilidad del cliente determinar los atributos necesarios por la oferta, sus tipos, etc.

Se emite una *CampaignServicesException* si se produce una de las condiciones siguientes:

- El parámetro folderID no es válido (no existe o no es del tipo de oferta).
- El usuario no está autorizado a ejecutar esta operación.
- Se suministran atributos no válidos en wsAttributes.
- Se producen otras excepciones de tiempo de ejecución.

Devoluciones

Una única instancia de *OfferInfo* para la oferta creada.

Errores

CampaignServicesException

retireOffers

```
void retireOffers(String userCredential, String partitionName,
        Locale requestedLocale, WSReference[] references)
    throws CampaignServicesException;
```

Retira una o varias ofertas existentes.

Parámetros

userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

references: matriz de referencias de las ofertas que desea retirar. Se emite *InvalidComponentException* si hay un problema con una referencia en particular o una oferta no existe.

Devoluciones

Ninguna.

Errores

InvalidComponentException

AuthorizationException, DataException

deleteOffers

```
void deleteOffers(String userCredential, String partitionName,
        Locale requestedLocale, WSReference[] references)
    throws CampaignServicesException;
```

Suprime una o varias ofertas existentes.

Parámetros
userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

referencia: una matriz de referencias de las ofertas que desea suprimir. Se emite una *InvalidComponentException* si hay un problema con una referencia especificada, o si una oferta no existe.

Devoluciones
Ninguna.

Errores
InvalidComponentException

AuthorizationException, DataException

listOfferTemplates

```
List<WSOfferTemplateInfo>
    listOfferTemplates(String userCredential, String partitionName,
        Locale requestedLocale)
    throws CampaignServicesException;
```

Lista de todas las plantillas de ofertas que el usuario tiene permisos para ver.

Una vez recuperadas, las instancias de *WSOfferTemplateInfo* devueltas se pueden utilizar tal cual, o se puede utilizar una o varias API de atributos para captar o actualizar cualquier plantilla listada.

Parámetro
userCredential: credencial de usuario cliente.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

Devoluciones
Una lista de tipo de cero o varias instancias de derivador de datos *WSOfferTemplateInfo*, una para cada plantilla devuelta.

Errores
InvalidExecutionContextException, AuthorizationException

DataException

bulkCreateOffers

```
WSOfferInfoStatus[] bulkCreateOffers(String authorizationLoginName,
        String partitionName, Locale requestedLocale,
        String securityPolicyName, String templateName, long folderID,
        WSBulkOfferInfo[] offers)
    throws CampaignServicesException;
```

Crea ofertas colectivas y los atributos para cada oferta se especifican en el parámetro *offers*. Todas las ofertas se crean bajo el parámetro *folderID* especificado mediante el parámetro *templateName* especificado.

Parámetro

authorizationLoginName: credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que desea utilizar.

requestedLocale: entorno local opcional que desea utilizar para esta solicitud.

securityPolicyName: nombre opcional de la política de seguridad de la campaña que desea utilizar para crear la oferta. Si no se ha definido, se utilizará la política global.

templateName: nombre de la plantilla de oferta existente en el sistema. Todas las ofertas se crearán utilizando esta plantilla.

folderID: ID de la carpeta de ofertas donde se crearán las ofertas. Este ID se validará y se emitirá una excepción si el ID no es válido.

offers: matriz de objetos *WSBulkOfferInfo* que define el nombre y los atributos de la oferta. Consulte el tipo de datos WSBulkOfferInfo para obtener más detalles.

Devoluciones

Una matriz de instancias de *WSOfferInfoStatus* para cada oferta. Contiene el estado y la información de la oferta. El estado indica si la creación de la oferta ha sido satisfactoria o no.

Errores

CampaignServicesException

Capítulo 5. Utilización de la API

En esta sección se explica cómo utilizar la API de servicios web de Campaign. También se muestra un ejemplo de utilización del servicio de la API de Campaign para crear un oferta en Campaign.

Existen dos enfoques para utilizar la API de servicios de Campaign:

- "Utilización de .jar de la API Client"
- "Utilización de WSDL" en la página 45

Utilización de .jar de la API Client

IBM Unica Campaign proporciona una API de cliente que utiliza los servicios web de SOAP para interactuar con la aplicación web de Campaign. Este derivador está empaquetado en un archivo `.jar` que la aplicación cliente puede utilizar para llamar a la API de Campaign.

Este archivo `.jar` se puede encontrar en:

```
<INICIO_CAMPAIGN>/devkits/CampaignServicesAPI/lib/
CampaignServicesClient30.jar
```

El siguiente ejemplo muestra la creación de una nueva oferta en el nivel de carpeta de oferta raíz en Campaign. La misma muestra se puede encontrar en:

```
<INICIO_CAMPAIGN>/devkits/CampaignServicesAPI/samples/OfferAPI.java
```

Nota: El ejemplo utiliza algunos valores ficticios para los parámetros; los valores reales pueden ser diferentes.

Además, tenga en cuenta que el URL para los servicios web de Campaign es `http://host:puerto/Campaign/services/CampaignServices30Service`, donde host y puerto hacen referencia al nombre de host y al número de puerto de la máquina donde se despliega la aplicación web de Campaign.

Si utiliza una muestra que le haya sido proporcionada, asegúrese de modificarla para ajustarla a su entorno de cliente.

Código OfferAPI.java

```
import java.net.URL;
import java.util.Locale;
import com.unica.publicapi.campaign.campaignservices.CampaignServicesException;
import com.unica.publicapi.campaign.campaignservices.attribute.metadata.
    IAttribute Metadata;
import com.unica.publicapi.campaign.campaignservices.soap.v30.
    CampaignServices30SoapClient;
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSAttribute;
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSOfferInfo;
import com.unica.publicapi.campaign.campaignservices.utils.WSAttributeUtils;

/**
 * Esta es la clase de cliente Java de muestra que muestra el uso de la API de servicios SOAP
de Campaign.
 * Esta muestra utiliza la fachada de CampaignServices30SoapClient para interactuar
```

```
con el servicio web de Campaign.
 * Se muestra la creación de ofertas. Consulte la guía de la API para
obtener más detalles.
 *
 * @author AGijare
 *
 */
public class OfferAPI {

    /**
     * @param args
     */
    protected static CampaignServices30SoapClient CLIENT = null;

    private static void setup(){
        try {
           String protocol = "http"; //http o https
           String host = "hostlocal"; //Nombre de host de la campaña desplegada.
Utilizar el nombre de host apropiado.
           int port = 7001; //número de puerto de la campaña desplegada
           long timeOut = 2*60*1000; // 2 minutos
           String servicesURI = "/Campaign/services/CampaignServices30Service";
           CLIENT = new CampaignServices30SoapClient(
               new URL(protocol, host, port, servicesURI),
               timeOut);
      } catch (Exception exception) {
          exception.printStackTrace();
          System.exit(-1);
      }
    }

    public static void main(String[] args) {
        //Cambie los valores de las variables siguientes para que coincidan con el
entorno.
        String userName = "nombre_usuario"; // Nombre de usuario de inicio de sesión
        String partitionName = "partición1"; //Usar el nombre de partición adecuado de
la campaña
        Locale loc = Locale.US;
        String securityPolicy = "Global"; //Usar la política de seguridad de
la campaña

        String offerName = "Oferta1";
        String offerTemplate = "Plantilla de oferta"; // Plantilla a partir de la cual
se creará la oferta.
        long folderID = 1002; //ID real de la carpeta donde esta oferta
se creará. Cuando folderID <=0, la oferta se creará a nivel raíz.
        //Atributos de oferta
        WSAttribute[] wsAttributes = {
            WSAttributeUtils.getWSTextAttribute(IAttributeMeta
data.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME, null, new String[]{"descripción "
+ System.currentTimeMillis()})
        };

        setup();

        try {
            WSOfferInfo wsOfferInfo = CLIENT.createOffer(userName,
partitionName, loc, securityPolicy,
                offerName, folderID, offerTemplate, wsAttributes);
            System.out.println("Oferta creada: " + wsOfferInfo.getName());
        } catch (CampaignServicesException e) {
            e.printStackTrace();
        }
    }
}
```

Nota: Para compilar y ejecutar la muestra de Java mostrado más arriba, debe incluir todos los archivos dependientes `.jar` en la ruta de clases Java. El archivo `CampaignServicesClient30.jar` depende de los archivos `.jar` del motor SOAP de Apache AXIS2 y otros archivos `.jar` de Apache, que se pueden encontrar en el archivo `Campaign.war` ubicado en `<INICIO_CAMPAIGN>/Campaign.war`. Extraiga los archivos `.jar` de `Campaign.war`, e inclúyalos en la ruta de clases Java.

Utilización de WSDL

Los servicios de Campaign también se pueden llamar utilizando el archivo WSDL de servicios web de Campaign:

```
CampaignServices30.wsdl
```

que se puede encontrar en:

```
http://host:puerto/Campaign/services/CampaignServices30Service?wsdl
```

o en la distribución de Campaign en:

```
<INICIO_CAMPAIGN>/devkits/CampaignServicesAPI/lib/
```

La aplicación Java cliente necesita utilizar las clases y los apéndices generados de WSDL utilizando cualquier herramienta de conversión de WSDL a Java de terceros. IBM Unica recomienda el uso de Apache AXIS.

El javadocs creado a partir de apéndices y clases generados de WSDL utilizando Apache AXIS2 se puede encontrar en:

```
<INICIO_CAMPAIGN>/devkits/CampaignServicesAPI/javadocs/index.html
```

Nota: Todos los archivos `.jar` dependientes deben incluirse en la ruta de clases Java. El archivo `CampaignServicesClient30.jar` depende de los archivos `.jar` del motor SOAP de Apache AXIS2 y otros archivos `.jar` comunes de Apache, que se pueden encontrar en el archivo `Campaign.war` ubicado en `<INICIO_CAMPAIGN>/Campaign.war`. Extraiga los archivos `.jar` de `Campaign.war`, e inclúyalos en la ruta de clases Java.

Consideraciones sobre el rendimiento

El perfil actual de rendimiento de la implementación de la API CampaignServices es similar al que la aplicación ha experimentado a través de la GUI. Algunas API están diseñadas explícitamente para el rendimiento. En particular, la API `listCampaignsByPage()` permite la paginación relativamente eficiente.

La interfaz SOAP, debido a su naturaleza, introduce latencia y sobrecarga porque todos los datos se convierten al formato XML, que en algunos casos es bastante verboso. Por ejemplo, una simple llamada de SOAP de bucle de retorno puede tardar 100 ms en una red típica (Java 1.4.x era aún más lenta). La API se ha optimizado para los casos de uso del portal típico y otras empresas de aplicación cliente, como see `listOffersByPage()`, por lo que el rendimiento de SOAP debería ser adecuado.

Sin embargo, el cliente debe tener cuidado de no poner demasiada carga en los servicios normales que CampaignServices presta a las solicitudes de usuario web.

En general, se espera que las necesidades del proceso de un usuario de API no excedan de las de un usuario web típico de IBM Unica Campaign.

Empaquetado

Esta especificación se entrega como parte de CampaignServices Software Developer's Tookit (devkits) instalado con IBM Unica Campaign.

El directorio de devkits establecido por el instalador incluye ejemplos, scripts de construcción y de texto, javadoc para clases públicas e interfaces, notas del release, etc.

Cómo contactar con el soporte técnico de IBM Unica

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 Unica . 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 Unica para obtener información.

Información a recopilar

Antes de ponerse en contacto con el soporte técnico de IBM Unica , recopile 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 su producto y el entorno del sistema, que podrá obtener según se describe en "información del sistema".

Información del sistema

Cuando llame al soporte técnico de IBM Unica , es posible que se le pida información sobre su 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 Unica 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 Unica del archivo `version.txt` que se encuentra en el directorio de instalación de cada aplicación. Si el archivo `version.txt` no está disponible, utilice el mandato `NetTracker admin -buildinfo` para obtener la información de la versión.

Información de contacto para el soporte técnico de IBM Unica

Para saber las formas de contacto con el soporte técnico de IBM Unica , consulte el sitio web del soporte técnico del producto IBM Unica : (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