

версия 10 выпуск 0  
Июнь 2016 г.

*Руководство по SOAP API для  
IBM Campaign*

**IBM**

**Примечание**

Перед тем как использовать данный документ и продукт, описанный в нем, прочтите сведения под заголовком “Замечания” на стр. 63.

Данное издание относится к версии 10, выпуску 0, модификации 0 продукта IBM Campaign и ко всем его последующим выпускам и модификациям, пока в новых изданиях не будет указано иное.

© Copyright IBM Corporation 1998, 2017.

# Содержание

## Глава 1. IBM Campaign Обзор API SOAP 1

Обзор разработки API SOAP . . . . .	1
Изменения API SOAP по версиям. . . . .	3
Существующие реализации, использующие файл JAR клиентского API. . . . .	4
Существующие реализации, непосредственно использующие WSDL . . . . .	5
Ссылки SOAP API. . . . .	7

## Глава 2. Использование IBM Campaign SOAP API . . . . . 9

Использование клиентского API jar для вызова служб Campaign . . . . .	9
OfferAPI.java . . . . .	9
Использование WSDL для вызова служб Campaign. . . . .	11
Замечания относительно производительности . . . . .	11

## Глава 3. Типы данных API SOAP . . . . . 13

WSReference . . . . .	13
WSVersion . . . . .	13
WSServiceInfo. . . . .	13
WSAttributeTypeEnum . . . . .	13
WSAttributeStatusEnum . . . . .	14
WSAccessTypeEnum . . . . .	14
WSSelectTypeEnum . . . . .	14
WSRunStatusEnum . . . . .	14
WSRunTypeEnum . . . . .	15
WSAttribute . . . . .	15
WSAttributeMetadata. . . . .	16
WSCampaignInfo . . . . .	18
WSComponentOrFolderInfo. . . . .	18
WSTargetCellInfo . . . . .	18
WSMetricsInfo. . . . .	18
WSRunResults. . . . .	19
WSOfferInfo . . . . .	19
WSOfferCodeOrName . . . . .	19
WSOfferValidationInfo . . . . .	20
WSOfferTemplateInfo . . . . .	20
WSBulkOfferInfo . . . . .	20
WSOfferInfoStatus . . . . .	20

## Глава 4. Методы API SOAP . . . . . 21

Методы API SOAP: служба . . . . .	21
getServiceInfo . . . . .	21
Методы API SOAP: Атрибуты . . . . .	21
getAttributesByName. . . . .	21
updateAttributes . . . . .	22
getAttributeMetadataByName . . . . .	23
createAttributeMetadata . . . . .	24
updateAttributeMetadata . . . . .	25
deleteAttributeMetadata . . . . .	26
Методика SOAP API: Кампании и потоковые диаграммы . . . . .	27
generateCampaignCode . . . . .	28
deleteCampaigns . . . . .	29

createCampaign . . . . .	29
listCampaignsByPage. . . . .	30
stopFlowchart . . . . .	31
Методы API SOAP: Ячейки назначения . . . . .	31
createTargetCell . . . . .	32
bulkCreateTargetCells . . . . .	33
listTargetCells . . . . .	34
bulkUpdateTargetCells . . . . .	35
getRunResultsByCell . . . . .	36
bulkDeleteTargetCells . . . . .	37
updateTemplateAttributes . . . . .	37
listBottomUpTargetCells . . . . .	38
Методы API SOAP: Анализ . . . . .	39
getCampaignMetrics . . . . .	39
Методы API SOAP: Предложения, списки предложений, шаблоны предложений . . . . .	39
listOffersAndFolders . . . . .	40
searchOffersBasic . . . . .	41
listOffersByPage . . . . .	42
createSmartOfferList . . . . .	43
createStaticOfferList . . . . .	44
getOffers . . . . .	44
validateOffers . . . . .	45
editOfferList . . . . .	46
createOffer . . . . .	47
retireOffers . . . . .	48
deleteOffers . . . . .	48
deleteOffersAndLists . . . . .	49
listOfferTemplates. . . . .	49
createTemplate. . . . .	50
getOfferTemplate . . . . .	50
retireOfferTemplates . . . . .	51
getOffersAndListsByPage . . . . .	51
bulkCreateOffers . . . . .	52
getOfferListDetails . . . . .	53
getOfferListMembers. . . . .	53
getOffersByQuery. . . . .	53
retireOfferLists. . . . .	54
createFolder . . . . .	55
editFolder . . . . .	55
getSubFoldersList . . . . .	56
moveFolders . . . . .	57
deleteFolders . . . . .	57

## Глава 5. Общие исключения API SOAP . . . . . 59

### Прежде чем обращаться в службу технической поддержки IBM . . . . . 61

### Замечания . . . . . 63

Товарные знаки . . . . .	65
Замечания относительно политики конфиденциальности и положений об использовании . . . . .	65



---

## Глава 1. IBM Campaign Обзор API SOAP

Спецификация API SOAP IBM® Campaign определяет версию 3.0 интерфейса прикладного программирования, который также называется Сервисный Campaign. Эта спецификация поставляется как часть Инструментария разработчика сервисного ПО Campaign (devkits), который устанавливается с IBM Campaign.

Каталог <CAMPAIGN\_HOME>/devkits/CampaignServicesAPI, который предоставляется средством инсталляции, включает примеры, полную сборку и текстовые сценарии, Javadoc для открытых классов и интерфейсов, а также сопроводительные записки.

Предназначение сервисного API SOAP Campaign:

- Обеспечение мелкомодульного доступа для создания, открытия, чтения и обновления к компонентам IBM Campaign с одновременной изоляцией клиентов от основных деталей реализации.
- Работа параллельно с существующим пользовательским интерфейсом IBM Campaign с минимальным воздействием.
- Гарантия правильности данных.
- Соответствие необходимым требованиям служб безопасности IBM Campaign.
- Поддержка SOAP промышленного стандарта (Простой протокол доступа к объектам), включая защищенную аутентификацию.

---

### Обзор разработки API SOAP

Campaign Services SOAP API представляет собой интерфейс для представления клиента при выполнении IBM экземпляра приложения Campaign. Представлено будет только подмножество возможностей IBM Campaign, но этого достаточно для управления ключевыми аспектами функции Campaign.

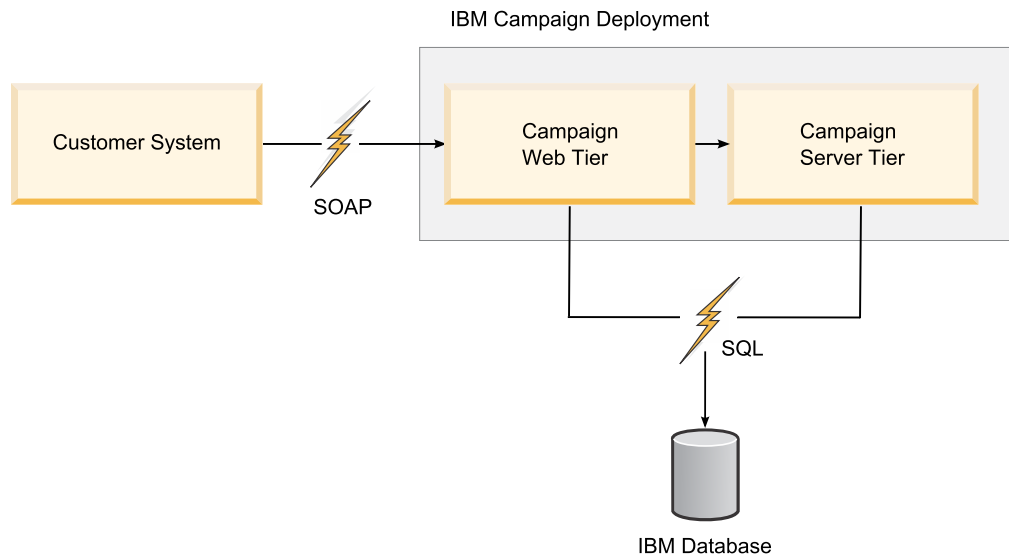
#### Функции и диаграмма

API используется одновременно с веб-пользователями IBM Campaign и другими потоками API.

Как правило, API поддерживает следующие типы операций на кампаниях, предложениях и компонентах целевой ячейки:

- Создание компонента
- Открытие компонента
- Удаление компонента
- Создание, проверка и изменение атрибута компонента и метаданных атрибута
- Выборка результатов потоковой диаграммы

На следующей диаграмме показан пример внедрения CampaignServices 3.0.



## Аутентификация пользователя

Аутентификация устанавливает идентификационные данные пользователя. Аутентификация пользователя - это ответственность клиентского приложения.

## Авторизация пользователей

Авторизация предусматривает собой полномочия, которые аутентифицируемый пользователь имеет относительно представленных API компонентов и операций.

Пользователь может аутентифицироваться успешно, но при этом не иметь достаточно полномочий для выполнения некоторых операций, таких как редактирование сводной информации кампании. В этом случае, метода API набрасывает `AuthorizationException`.

## Локаль

Запросы API предусматривают дополнительный параметр **requestedLocale** для определения локали для выполнения конкретного требования. Если значения по умолчанию API не определены для предпочтительной локали пользователя IBM. Для возврата сообщений и другого локализованного текста в требуемую локаль используется обычный алгоритм соответствия максимальных усилий Java™.

Этот параметр имеет класс типа `java.util.Locale`.

**Примечание:** в локали определившего текст пользователя находится некоторый определенный пользователем текст, например, описания кампании. IBM Campaign не пытается локализовать такие данные. В API локализованы только информационные сообщения, предупреждения и сообщения об ошибках.

## Управление состояниями

CampaignServices API не имеет состояния, что означает, что никакая информация о клиенте не будет сохранена API путем вызовов.

Очевидно, что определенные вызовы API могут изменить состояние базовых экземпляров компонента, которыми управляют, Campaign, и такие изменения

состояния могут быть сохранены в базе данных.

---

## Изменения API SOAP по версиям

Цель этой темы состоит в том, чтобы определить изменения в API SOAP IBM Campaign для клиентов, которые используются API SOAP в настоящее время. Если вы обновили из предыдущей версии, смотрите следующую информацию, чтобы определить необходимость внесения изменений в код приложения.

### Версии и совместимость с предшествующими версиями

Будущие версии CampaignServices API совместимы с более ранними версиями, со всеми незначительными и отладочными версиями, совместно использующими номер основной версии. Однако, IBM сохраняет за собой право повредить совместимость с более ранней версией для главных версий "точечного ноля" (x.0).

Если внесены какие-либо из следующих изменений, номер основной версии API будет увеличен:

- Изменилась интерпретация данных
- Изменилась бизнес-логика, то есть, функциональность метода службы
- Изменились параметры метода и/или типы возвращаемых значений

Если какое-либо из следующих изменений внесено, то номер младшей версии API будет увеличен.

- Добавлен новый метод
- Добавлен новый тип данных и их использование ограничено новыми методами
- Добавлен новый тип к перечислимому типу
- определена новая версия интерфейса

IBM продолжает поддерживать опубликованный WSDL, клиент SOAP и версию Apache Axis, использующуюся для реализации предложения SOAP, по крайней мере, следующей основной версии IBM. В сущности это достигается путем одновременной поддержки нескольких определенных для версии веб-служб. (IBM уже внутренне поддерживает несколько версий этой службы.)

### Изменения в v9.1.2

В версии 9.1.2 не было никаких изменений API.

**Примечание:** API REST был введен в v9.1.2, в дополнение к API SOAP, описанному в этом руководстве. The Описание REST API приводится отдельно.

### Изменения в v9.1.1

**stopFlowchart** включает в себя новый входной параметр, **RUNID**, для поддержки кластеризованной среды детектора.

### Изменения в v9.1

В версии 9.1 не было никаких изменений API.

### Изменения в v9.0

В версии 9.0 не было никаких изменений API.

## Изменения в v8.6

Реализованные в v8.6 изменения API:

- Механизм SOAP обновлен от версии 1.4.1 AXIS до AXIS2 1.5.2.
- WSDL был реструктурирован для решения проблем обработки необходимых и дополнительных параметров.
- Клиентский файл API .jar изменился из-за изменений WSDL. В результате произведенные заглушки и классы также изменились. Клиентские параметры метода API не были изменены, но для использования преобразователя AXIS2 WSDL2Java были изменены конструкторы поддержки объектов значения.
- URL веб-службы указывает на:

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

и соответствующий WSDL, полученный в:

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

Код приложения необходимо изменить, если вы выполняете обновление до IBM Campaign версии 8.6 или выше, и в настоящее время используете службу API Campaign.

В зависимости от использования клиентского API или WSDL, вы можете видеть следующие разделы для дополнительной информации:

- “Существующие реализации, использующие файл JAR клиентского API”
- “Существующие реализации, непосредственно использующие WSDL” на стр. 5

## Существующие реализации, использующие файл JAR клиентского API

Эта информация будет релевантной при обновлении до IBM Campaign версии 8.6 или выше и при использовании в настоящее время клиентского файла API JAR для взаимодействия с веб-программой Campaign.

### Клиентский файл API.jar

Ваше приложение Java должно использовать файл .jar, который:

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

Пример Java, показывающий создание нового предложения, можно найти в разделе “OfferAPI.java” на стр. 9. Тот же самый пример может быть найден в вашей установке Campaign в:

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

### Зависимые файлы .JAR

В результате обновления до версии 1.5.2 AXIS2, ваше приложение Java должно также обновиться для использования файлов распределения AXIS2 1.5.2 .jar, так как файл CampaignServicesClient30.jar зависит от этих файлов .jar. Все зависимые файлы .jar должны быть включены в путь класса Java вашего приложения и могут быть найдены в файле Campaign.war, который содержится в <CAMPAIGN\_HOME>/Campaign.war.



Извлеките файлы .jar из Campaign.war, и включите их в путь к классу Java.

## Клиентский конструктор API

По мере создания клиентского объекта API, измените URL веб-службы и подпись Исключения, как показано в этом примере.

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

## Параметризованные конструкторы классов поддержки

Произведенные с помощью механизма AXIS2 классы и тупики не имеют параметризованных конструкторов. Вместо этого, эти классы имеют только конструктор без параметров по умолчанию с механизмами включения и методами чтения для элементов.

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

## Существующие реализации, непосредственно использующие WSDL

Эта информация будет релевантной при обновлении IBM Campaign версия 8.6 или выше и текущем использовании WSDL для взаимодействия с веб-приложением Campaign. WSDL представляет собой веб-службу Campaign для создания клиентских тупиков и классов поддержки при помощи любого стороннего инструмента преобразователя. Приведенные здесь примеры используют инструмент WSDL2Java от Apache AXIS2 1.5.2.

## Местоположение WSDL и служба URL

Веб-служба Campaign для IBM Campaign будет развернута в:

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

Соответствующий WSDL может быть получен в:

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

## Создание тупиков и классов

Инструмент WSDL2Java от Apache AXIS2 1.5.2 может использоваться для создания тупиков и классов поддержки Java в WSDL. Демонстрационная задача Ant показана здесь.

Инструмент можно также использовать из командной строки с аналогичным набором параметров. Значения аргументов можно изменить для соответствия вашей среде.

**Примечание:** Привязка ADB по умолчанию используется для следующего примера преобразователя WSDL2Java.

```

<java classname="org.apache.axis2.wsdl.WSDL2Java" fork="true">
  <classpath refid="axis2.class.path"/> <!--Class path having
  AXIS2 libraries -->
  <arg value="-uri"/>
  <arg file="CampaignServices30.wsdl"/> <!--Actual location of
  WSDL -->
  <arg value="-s"/> <!-- Generate sync style code -->
  <arg value="-Euwc"/> <!-- Takes care of generating Wrapper
  java types for nillable = true elements. -->
  <arg value="-uw"/> <!-- Unwrap params -->
  <arg value="-u"/> <!-- Unpack classes -->
  <arg value="-ns2p"/> <!-- Namespace to package mapping. Покупатель
  может иметь собственные имена пакетов. -->
  <arg value="http://webservices.unica.com/campaign/CampaignServices/
  3.0=com.unica.publicapi.campaign.campaignservices.soap.v30"/>
  <arg value="-o"/> <!-- Выходной каталог -->
  <arg file="\${autogen.java.dir}"/>
</java>

```

## Использование произведенных тупиков и классов поддержки

Тупик может использоваться следующим образом:

```

CampaignServices30ServiceStub serviceStub = new
CampaignServices30ServiceStub(serviceURL);

serviceStub.getServiceClient().getOptions().setTimeOutInMilliseconds
(webServiceTimeout); //Тайм-аут в миллисекундах.

```

Предложение может быть создано следующим образом:

```

try{
  //Измените хост и порт для соответствия среде.
  String serviceURL = "http://host:port/Campaign/services/CampaignServices30Service";
  CampaignServices30ServiceStub serviceStub = new CampaignServices30ServiceStub(serviceURL);
  long webServiceTimeout = 2*60*1000; // 2 минуты
  serviceStub.getServiceClient().getOptions().setTimeOutInMilliseconds(webServiceTimeout); //Таймаут в миллисекундах.

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

  WSTextAttribute[] wsAttributes = {nameAttribute};
  // преобразовать в WSAttributeArrays
  WSAttributeArrays obj = new WSAttributeArrays();
  obj.setTextAttributes(wsAttributes);
  //Измените значения следующих переменных для соответствия вашей среде.
  String authorizationLoginName = "asm_admin"; //имя зарегистрированного пользователя
  String partitionName = "partition1"; //Использование вашей политики безопасности Campaign
  String securityPolicyName = "Global Policy"; //Использование вашей политики безопасности Campaign

  String offerName = "1st Offer"; //Имя предложения, которое будет создано.
  String templateName = "Offer Template"; //Существующее имя шаблона предложений.
  long folderID = 100; //Фактический ID папки, где будет создано это предложение.
  //For folderID <=0, предложение будет создано на корневом уровне.

  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);
  // make campaign Webservice call
  WSCreateOfferResponse wsResponse = serviceStub.createOffer(createOfferObject);
  // состояние процесса
  WSRequestStatus status = wsResponse.getStatus();
  // готово
  WSOfferInfo offerInfo = wsResponse.getOfferInfo();
  System.out.println("status = "+status.getStatusType());
  System.out.println("offerInfo = "+offerInfo.getName());
}

```

```
} catch (Exception exception) {  
    //Обработайте исключение здесь.  
    exception.printStackTrace();  
}
```

В этом примере `createOffer()` теперь принимает только один параметр типа `CreateOffer`.

Произведенные с помощью механизма AXIS2 классы и тупики не имеют параметризованных конструкторов.

---

## Ссылки SOAP API

При подготовке IBM Campaign SOAP API использовались следующие ссылки.

- "Основная версия профиля 1.1", организация функциональной совместимости веб-службы (WS-I), 10 апреля 2006 г. (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-0310.html>)
- "SOAP 1.2 (проект)", рабочая группа W3C Soap, 24 июня 2003 г. (<http://www.w3.org/TR/soap/>)
- "JAX-RPC 1.1", Sun Microsystems, October 14, 2003 (<http://java.sun.com/webservices/jaxrpc/index.jsp>)
- Рабочая группа веб-служб Apache (<http://ws.apache.org/axis2>)



---

## Глава 2. Использование IBM Campaign SOAP API

Для использования веб-службы SOAP API IBM Campaign, вы можете использовать клиентский файл API .jar или напрямую использовать язык описания программных интерфейсов WSDL. Пример показывает, как использовать метод файла .jar для создания предложения.

---

### Использование клиентского API .jar для вызова служб Campaign

IBM Campaign предоставляет клиентский API, использующий веб-службы SOAP для взаимодействия с веб-приложением Campaign. Данный адаптер объединен в файл .jar, который клиентское приложение может использовать для вызова API Campaign.

Файл .jar можно найти по следующему пути:

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

Следующий пример демонстрирует создание нового предложения на уровне корневой папки Предложение в Campaign. Такой же пример можно найти в:

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

**Примечание:** В примере использованы некоторые фиктивные значения для параметров; ваши фактические значения могут отличаться.

Кроме того, URL для веб-служб Campaign - `http://host:port/Campaign/services/CampaignServices30Service`, где хост и порт относятся к названию ведущего приложения и номеру порта компьютера, где развернуто веб-приложение Campaign.

При использовании предоставленных примеров обязательно измените их в соответствии с вашей клиентской средой.

#### OfferAPI.java

Чтобы скомпилировать и выполнить выборку Java, всего зависимые файлы .jar необходимо включить в путь к классам Java. Файл CampaignServicesClient30.jar зависит от файлов .jar движка Apache AXIS2 SOAP или других обычных .jar Apache, которые можно найти в <ДОМАШНИЙ\_КАТАЛОГ\_CAMPAIGN>/Campaign.war. Извлеките файлы .jar из Campaign.war, и включите их в путь к классам 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;
```

```
/**
```

```
 * Это демонстрационный клиентский класс Java, показывающий использование API услуг SOAP  
 Campaign.
```

```
 * Эта выборка использует фасадный метод CampaignServices30SoapClient  
 для взаимодействия с веб-службой Campaign.
```

```

* Здесь показано создание Предложения. Смотрите руководство по API для
получения подробной информации.
*
* @author AGijare
*
*/
public class OfferAPI {

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

    private static void setup(){
        try {
            String protocol = "http"; //http or https
            String host = "localhost"; //Имя хоста внедренной Кампании.
            Используйте надлежащее имя вычислительного узла здесь.
            int port = 7001; //номер порта развернутой Campaign
            long timeOut = 2*60*1000; // 2 минуты
            String servicesURI = "/Campaign/services/CampaignServices30Service";
            CLIENT = new CampaignServices30SoapClient(
                новый URL (протокол, вычислительный узел, порт, servicesURI),
                timeOut);
        } catch (Exception exception) {
            exception.printStackTrace();
            System.exit(-1);
        }
    }

    public static void main(String[] args) {
        //Измените значения следующих переменных для соответствия вашей среде.
        String userName = "user_name"; //login user name
        String partitionName = "partition1"; //Использование надлежащее имя раздела
        Кампания
        Locale loc = Locale.US;
        String securityPolicy = "Global"; //Использование надлежащее имя раздела
        Кампания

        String offerName = "Offer1";
        String offerTemplate = "Offer Template"; // Шаблон, на основе которого
        будет создано предложение.
        long folderID = 1002; //фактический идентификатор папки, в которой будет создано
        это предложение. Для folderID <=0, предложение будет создано на корневом уровне.
        //Атрибуты предложения
        WSAttribute[] wsAttributes = {
            WSAttributeUtils.getWSTextAttribute(IAttributeMeta
            data.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME, null, new String[]{"description "
            + System.currentTimeMillis()})
        };

        setup();

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

```

---

## Использование WSDL для вызова служб Campaign

Службы Campaign можно вызвать при помощи файла веб-службы WSDL Campaign: CampaignServices30.wsdl.

Файл CampaignServices30.wsdl можно найти по адресу:

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

или в распределении Campaign в:

`<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/`

Клиентское приложение Java должно использовать классы и "рыбы", созданные при помощи WSDL с использованием любого стороннего конвертирующего инструмента WSDL-Java. IBM рекомендует использование Apache AXIS.

Документы Java, созданные на основании "рыбы" и классов, генерируемых WSDL при помощи Apache AXIS2, можно найти в:

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

**Примечание:** Все зависимые файлы .jar должны быть включены в путь к классам Java. Файл CampaignServicesClient30.jar зависит от файлов движка Apache AXIS2 SOAP.jar и других общих файлов Apache.jar, которые можно найти в файле Campaign.war, т.е. в `<CAMPAIGN_HOME>/Campaign.war`. Извлеките файлы JAR из Campaign.war, и включайте их в путь к классам Java

---

## Замечания относительно производительности

Текущий профиль выполнения CampaignServices API напоминает профиль выполнения приложения GUI. Некоторые API явно разработаны для производительности. В частности listCampaignsByPage () API разрешает относительно эффективное разбиение на страницы.

Интерфейс SOAP вводит задержку и ограничения, поскольку все данные преобразовываются в формат XML, который в некоторых случаях довольно словесный. Например, для простого возврата цикла SOAP может потребоваться 100 мс в типичной сети (в Java 1.4.x еще медленнее). API оптимизирован для типичного портала и других бизнес-сценариев использования клиентского приложения, таких как see listOffersByPage (), таким образом, производительность SOAP должна быть соответствующей.

При этом клиент не должен слишком сильно нагружать нормальное обслуживание CampaignServices запросов веб-пользователя. В целом, обработка пользователя API должна не превышать установленную для типичного веб-пользователя IBM Campaign.





---

## Глава 3. Типы данных API SOAP

Службы IBM Campaign API SOAP используют следующие общедоступные типы данных.

---

### WSReference

Простая оболочка вокруг идентификатора базы данных:

- **componentTypeEnum**: перечислимый тип, указывающий на тип компонента, которому соответствует идентификационный номер. Одно из следующих значений:
  - FOLDER
  - CAMPAIGN
  - FLOWCHART
  - TCS\_CELL
  - OFFER
  - OFFER\_LIST
  - OFFER\_TEMPLATE
- *id*: *Длинный*, определяет уникальный числовой идентификатор определенной базы данных для ссылки.

---

### WSVersion

Тип оболочки, записывающей различные компоненты версии, включая следующее:

- *полноценный*: целое значение, определяющее номер полноценной версии, такой как '8' в полной версии 8.1.2.3.
- *младший*: целое значение, определяющее номер младшей версии, такой как '1' в полной версии 8.1.2.3.
- *обслуживание*: необязательное целое значение, определяющее номер обслуживания версии, если он применяется, такой как '2' в полной версии 8.1.2.3. Никогда не предоставляется с версией API.
- *patch*: необязательное целое значение, определяющее номер выпуска patch, такой как '3' в полной версии 8.1.2.3. Никогда не предоставляется с версией API.

---

### WSServiceInfo

Тип простой оболочки вокруг информации о службе. Здесь содержатся следующие поля:

- *apiVersion*: экземпляр *WSVersion*, определяющий наиболее актуальную версию API, поддерживаемую службой. (*apiVersion* включает только информацию о полноценной и младшей версии)
- *campaignVersion*: экземпляр *WSVersion*, определяющий полноценную версию заложенного экземпляра IBM Campaign.
- *имя*: внутреннее название службы, такой как "CampaignServices30Service".

---

### WSAttributeTypeEnum

Перечислимый тип, определяющий все возможные типы атрибутов, один из:

- СТАНДАРТНЫЙ: стандартный или основной атрибут, определяемый Campaign.

- **ПОЛЬЗОВАТЕЛЬСКИЙ**: атрибут, определенный другим приложением IBM , покупателем или другим лицом.
- **ВХОДНОЙ ПАРАМЕТР**: входной параметр, такой как атрибут, используемый для запуска потоковой диаграммы IBM Campaign.
- **ВЫХОДНОЙ ПАРАМЕТР**: выходной параметр, такой как атрибут, значение которого заполнено в результате того, что потоковая диаграмма запущена в IBM Campaign.

---

## WSAttributeStatusEnum

Перечисление всех возможных кодов состояния атрибутов, один из:

- **АКТИВНЫЙ** - атрибут активен и может использоваться произвольно.
- **RETIRED**: атрибут удален из использования и не должен использоваться.

---

## WSAccessTypeEnum

Перечислимый тип, определяющий все возможные типы доступа значения атрибутов, один из:

- **READ\_ONLY**: значение атрибута может быть прочитано и показано, но не изменено.
- **READ\_WRITE**: значение атрибута может быть прочитано, показано и изменено.

Доступ к атрибуту является дополнением к правам доступа. Например, если политика безопасности для пользователя клиента отказывает в доступе к определенному атрибуту, то доступ атрибута не может переопределить данную настройку безопасности. На самом деле API никогда не возвратит атрибут клиенту.

---

## WSSelectTypeEnum

Определяет все возможные типы выбора для конкретного значения атрибута, одного из:

- **ОТСУТСТВУЕТ**: не выбран (*hasOptions* = false).
- **ОДИНОЧНЫЙ ВЫБОР**: на данный момент можно выбрать только одну опцию атрибута из списка возможных опций (допускается, если атрибут - *hasOptions*).
- **MULTIPLE\_SELECT**: подобно **SINGLE\_SELECT**, за исключением того, что одновременно можно выбрать одну или более опций.

---

## WSRunStatusEnum

Перечисленный тип всех возможных потоковых диаграмм, состояний запуска ячеек или ветвей, один из:

- **NOT\_STARTED**: запуск запланирован, но еще не начат.
- **ЗАПУСК**: запуск в процессе.
- **ОТМЕНА**: запуск отменен пользователем Campaign или данным API.
- **УСПЕШНО**: Выполнение завершилось успешно.
- **ОШИБКА**: ошибка запуска; детали ошибки указаны в отдельном отчете. (Смотрите “WSRunResults” на стр. 19.)

---

## WSRunTypeEnum

Перечисленный тип всех возможных типов запуска, одного из:

- NOT\_RUN
- TEST\_RUN
- PRODUCTION\_RUN
- RUN\_SKIPPED
- TEST\_FLOWCHART
- PRODUCTION\_FLOWCHART
- TEST\_BRANCH
- PRODUCTION\_BRANCH
- TEST\_PROCESS
- PRODUCTION\_PROCESS

---

## WSAttribute

Атрибуты обеспечивают простой, расширяемый механизм для присоединения произвольных данных к экземплярам компонента, доступным через API, любые стандартные данные, такие как кампания *Имя*, потоковая диаграмма запускают входные параметры как *пол*, или произвольные пользовательские данные, определенные другим приложением IBM или покупателем IBM .

**Примечание:** В данном API атрибуты используются для моделирования большинства данных компонента, не только пользовательских атрибутов Campaign.

Компоненты обычно имеют множество атрибутов, связанных с ними, и представленных CampaignServices API как специально введенная Карта, которую называют *AttributeMap*. Данные атрибута представлены как четко обозначенный введенный класс по всему API, такой как *WSDecimalAttribute*, для атрибутов, содержащих десятичные данные (число с двойной точностью).

Каждый атрибут включает следующее:

- *Имя*: уникальное имя атрибута. Данное имя служит ключом для доступа к атрибуту и его метаданным в рамках экземпляра компонента, который оно встречает. Формат имени не определен; в некоторых случаях он присваивается службой клиентом, или пользователем IBM Campaign.

В целом, данное имя не является именем для вывода на экран, которое бы представлялось Campaign или пользователю клиента. Оно может стандартизироваться с помощью API, такого как *uacDescription*, может присваиваться командой IBM Campaign при публикации потоковых диаграмм, или присваиваться приложением IBM или клиентом при определении пользовательских атрибутов. Во всех случаях, однако, имя гарантированно уникальное.

- *Метаданные*: (необязательные) информация о данных атрибута, таких как тип данных значения, имя для вывода на экран, описание, подсказки, значение по умолчанию, выбранный тип, длина (текст), точность (десятичные числа), опции (в случае с групповым или одиночным выбором). Смотрите раздел “WSAttributeMetadata” на стр. 16.
- *Значения*: массив объектов с введенными нулевыми или другими значениями. Поле значений предоставляется конкретным классом атрибутов; тип каждого значения должен быть тем же самым и согласиться с определением типа в поле метаданных атрибута. Однако, не все атрибуты поддерживают множественные значения.

Поддерживаются следующие конкретные типы атрибутов:

- **WSBooleanAttribute**: атрибут с логически выраженным значением, которое является *true* или *false*.
- **WSIntegerAttribute**: целое значение (*java.lang.Long*).
- **WSDecimalAttribute**: значение десятичного числа двойной точности (*java.lang.Double*).
- **WSCurrencyAttribute**: составное значение валюты, включая необязательный код валюты ISO 4217 значения валюты, такого как "USD" для доллара США и значения валюты, зафиксированные как *Двойные*. Если код валюты не предоставлен, используемое по умолчанию значение IBM Campaign является условным.  
Смотрите <http://www.xe.com/symbols.php> для информации о списке стран, символах денежной единицы и кодах. Локаль, используемая для значения валюты, может отличаться от предпочтительной локали пользователя.
- **WSCalendarAttribute**: значения которого являются календарными датами или датой/временем в некоторых часовых поясах и локали.
- **WSTextAttribute**: строка текста Unicode (возможно, пустая или отсутствующая).

**Примечание:** Список возможных атрибутов обычно отличается для каждого типа компонента, но списки могут накладываться.

---

## WSAttributeMetadata

WSAttributeMetadata указывает информацию о данных определенного введенного атрибута, такого как тип данных значения, локализованный текст (имя для вывода на экран, описание, подсказки), его значение по умолчанию, допустимый диапазон значений, выбранный тип, опции (в случае с групповым или одиночным выбором). Как и в случае с атрибутами, метаданные атрибута также вводятся. Например, WSDecimalAttribute *myNumber* должен иметь веб-службу *DecimalAttributeMetadata*, привязанную к нему, и все значения, включая значения атрибутов, значения метаданных по умолчанию и возможные значения опций, должны вводиться *Двойными*.

Описания, метки и другой текст метаданных атрибута локализованы; несмотря на это, определенный пользователем текст может только быть доступным, если он был введен пользователем. Каждый вызов API включает требуемую локаль, которую клиентский код может использовать для определения локали, в которой определенному пользователю потребовался бы вывод локализованных сообщений. Стандартные политики Java возврата локали используются для выполнения запроса.

WSAttributeMetadata включает следующие поля:

- **Имя:** Имя атрибута, стандартное или пользовательское; также имя, использованное атрибутом, которое привязывается к данным метаданным. Стандартные атрибуты определены системой и имеют стандартные имена в пространстве зарезервированных имен (то есть, они используют префикс "UAC"), пользовательские имена могут использовать любое другое соглашение о присвоении имен.

**Примечание:** Название атрибута должно быть уникальным, никогда не локализуется и имеет ограничения длины (которые зависят от символического содержания и базы данных). Имя не зависит от регистра и может быть составлено из любой комбинации букв Unicode или цифр плюс символ подчеркивания '\_', но не может начинаться с цифры.

- **описание:** необязательное описание атрибута. Подходит для подсказки или другого представления пользовательского интерфейса.

- Предикаты: различные предикаты, описывающие атрибут:
  - *isRequired*: true, если атрибут обязателен.
  - *isInternal*: true, если атрибут определен системой и только для внутреннего пользования (не должен быть представлен пользователю).
  - *isGenerated*: true, если значение или значения атрибута генерируются автоматически с помощью IBM Campaign при создании компонента, такого как код целевой ячейки. Обычно, *accessTypeEnum* будет доступным только ДЛЯ ЧТЕНИЯ для полученных значений.
  - *hasOptions*: true, если атрибут имеет опции. Подразумевает, что опции определяются для этих метаданных и что *selectTypeEnum* является или ОДИНОЧНЫМ ВЫБОРОМ или ГРУППОВЫМ ВЫБОРОМ.
- *typeEnum*: A *WSAttributeTypeEnum*, определяющие тип атрибута, такой как СТАНДАРТНЫЙ или ПОЛЬЗОВАТЕЛЬСКИЙ.
- *statusEnum*: A *WSAttributeStatusEnum*, определяющий состояние атрибута, например, АКТИВНЫЙ.
- *accessTypeEnum*: A *WSAccessTypeEnum*, определяющий тип доступа к значению атрибута, например, ТОЛЬКО ДЛЯ ЧТЕНИЯ.
- *selectTypeEnum*: a *WSAccessTypeEnum*, определяющий тип выбора, используемый для атрибута, например, ЕДИНИЧНЫЙ. Должен стоять на ОТСУТСТВУЕТ для кампании и атрибутов ячеек, или если других опций нет.
- *componentTypeEnum*: A *WSComponentTypeEnum* из всех возможных компонентов Campaign, представленных API, таких как КАМПАНИЯ, ПАПКА.
- *defaultValue* (только для потоковой диаграммы): необязательное вводимое значение по умолчанию для атрибута. Это значение предоставляется определенным классом метаданных атрибута, таким как значение по умолчанию *WSTextAttributeMetadata*, которое имеет тип Строка. (Обратитесь к описанию значений атрибута). Для компонентов кроме потоковых диаграмм значение по умолчанию не определено.
- *опции*: необязательный список опций для данного атрибута. Если берутся вместе, опции атрибута определяют точный набор допустимых значений для того атрибута; каждая опция вводится, чтобы, например, *WSTextAttributeMetadata* мог иметь только *WSTextAttributeOption*, привязанный к нему.

**Примечание:** Существует ограничение на опции; поддерживаются только текстовые атрибуты.

Каждая опция определяет следующее:

- *подсказка*: подсказка для опции, подходящая для выпадающих меню, таких как "Мужской" как опция атрибута "пол". В отличие от подсказки метаданных, показанные названия опции обычно не содержат пунктуации.
- *описание* локализованное описание опции, такое как "Человек твердых убеждений". Подходит для текста всплывающей подсказки.
- *ISDEFAULT*: true, если эта конкретная опция является значением по умолчанию. Для типов выбора MULTIPLE\_SELECT больше чем одна опция может быть отмечена как значение по умолчанию.
- *значение*: вводимое значение опции. Как и с метаданными атрибута *defaultValue*, данное значение предоставляется конкретным подклассом опций, таким как значение *WSDecimalAttributeOption* имеет Десятичное тип. (Обратитесь к описанию значений атрибута). В продолжение вышеуказанного примера с *полом*, значение может быть объявлено как строка "m" (*WSTextAttributeOption*) или как числовой код, 123 (*WSDecimalAttributeOption*).

---

## WSCampaignInfo

Тип простой оболочки вокруг данных атрибута кампании.

Здесь содержатся следующие поля:

- *ссылка*: ссылка кампании.
- *имя*: название кампаний(*uacName*); не гарантированно уникальное.
- *описание*: необязательное описание кампании (*uacDescription*).
- *campaignCode*: уникальный код кампании (*uacCampaignCode*), присвоенный клиентом или Campaign.

---

## WSComponentOrFolderInfo

Содержит комбинацию кампании в оболочке или данных атрибута папки, таких как имя для вывода на экран и ссылка на него.

Здесь содержатся следующие поля:

- *ссылка*: ссылка компонента или папки.
- *имя*: название компонента или папки (*uacName*); не гарантированно уникальные.
- *описание*: необязательный компонент или описание папки (*uacDescription*).
- *componentCode*: уникальный код компонента или, пустой символ - если папка.

---

## WSTargetCellInfo

Простая оболочка вокруг данных атрибута строки целевой ячейки.

Здесь содержатся следующие поля:

- *ссылка*: ссылка на ячейку.
- *имя*: название ячейки (*uacName*); уникальность не гарантируется.
- *описание*: необязательное описание ячейки (*uacDescription*).
- *cellCode*: код ячейки (*uacCellCode*); назначенный клиентом или Кампанией. Коды ячеек можно сделать уникальными, установив параметр конфигурации IBM Campaign *DuplicateCellCodesAllowed* на *false*.
- *flowchartName*: необязательное название потоковой диаграммы, к которой привязана ячейка.

---

## WSMetricsInfo

Тип простой оболочки вокруг аналитических данных кампании, включая число контактов. Здесь содержатся следующие поля:

- *totalContacts*: долгий подсчет общего количества контактов.
- *ответы*: введенный список экземпляров *WSMetricsResponse*, каждый экземпляр определяет контактную информацию для одного ответа:
  - *typeCode*: строка, определяющая код типа ответа, такой как *PHC* для контакта телефонного вызова.
  - *подсчет*: долгий подсчет количества случаев установления данного контакта.

---

## WSRunResults

Тип оболочки результатов потоковой диаграммы, окна процессов или запуска ячеек, который, возможно, все еще выполняется, включая состояние запуска, начало выполнения потоковой диаграммы и дата/время завершения, а также подсчет.

Он включает следующие поля:

- *sourceReference*: необязательная ссылка на источник результата запуска. В соответствии с контекстом того, что результаты запуска были выбраны, это может обратиться к потоковой диаграмме, окну процесса потоковой диаграммы или целевой ячейке. В любом случае оставшиеся данные результата выполнения ссылаются на этот источник.
- *FLOWCHARTNAME*: название выполненной потоковой диаграммы.
- *flowchartId*: идентификатор базы данных для потоковой диаграммы.
- *runId*: идентификатор баз данных для запуска.
- *typeEnum*: перечисляемый тип, определяющий то, какой запуск сгенерировал результаты, такие как ПРОЦЕСС ПРОИЗВОДСТВА (смотрите *WSRunTypeEnum*).
- *statusEnum*: перечисляемый тип, определяющий состояние запуска, например, ЗАПУСК (смотрите *WSRunStatusEnum*).
- *statusCode*: целое значение необязательного кода состояния.
- *statusMessage*: необязательное сообщение о состоянии.
- *startDate*: необязательные дата/время календаря, когда начался запуск. Пустой символ - если запуск не выполнялся.
- *ENDDATE*: как и с *startDate*, но дата/время, когда выполнение завершилось (успешно или неуспешно), пустой символ - если запуск не выполнялся или выполнение не завершено.
- *подсчет*: необязательный общий подсчет количества контактов, выбранных во время запуска; может быть нулевым или пустым, если выполнение не завершено.

---

## WSOfferInfo

Тип простой оболочки вокруг предложения или данных атрибута списка предложений.

Здесь содержатся следующие поля:

- *ссылка*: предложение или ссылка на список предложений.
- *имя*: название предложения или списка предложений (*uacName*) с не гарантированной уникальностью.
- *описание*: необязательное описание (*uacDescription*).
- *offerCode*: код предложения (*uacOfferCode*) - если предложение, или пустой символ - если список предложений. (уникальность не гарантируется)

---

## WSOfferCodeOrName

Тип простой оболочки вокруг кодов предложения или данных имен списка предложений.

Здесь содержатся следующие поля:

- *isCode*: логическое значение, указывающее на то, является ли поле *codeOrName* предполагаемым полем предложения (true) или названием списка предложений (false).

- *codeOrName*: уникальный код предложения (*uacOfferCode*) или названия списка предложения.

---

## WSOfferValidationInfo

Тип простой оболочки вокруг информации о проверке предложения.

Здесь содержатся следующие поля:

- *errorCode*: если не пустой символ - определяет буквенно-цифровой валидационный код ошибки. Коды ошибок смотрите в классе *IStandardDefinitions*.
- *errorMessage*: необязательное локализованное сообщение, описывающее ошибку (если она случилась).
- *codeOrName*: подтвержденный код предложения или названия списка предложений.
- *ссылка*: ссылка на предложение или список предложений, если она действительна.

---

## WSOfferTemplateInfo

Тип простой оболочки вокруг данных шаблона предложения.

Здесь содержатся следующие поля:

- *ссылка*: ссылка на шаблон предложения.
- *имя*: название шаблона предложения, уникальность не гарантирована.
- *описание*: необязательное описание (*uacDescription*).
- *offerTemplateID*: уникальный идентификатор базы данных шаблонов предложений.

---

## WSBulkOfferInfo

Используются для массового создания предложений.

Здесь содержатся следующие поля:

- *offerName*: имя создаваемого предложения.
- *атрибуты*: массив типов *WSAttribute*, указывающий на атрибуты предложения.

---

## WSOfferInfoStatus

Тип возврата для метода API *bulkCreateOffers()*, который указывает на состояние массового создания предложений.

Здесь содержатся следующие поля:

- *имя*: имя предложения
- *код*: код предложения. Пустой символ - если создание предложения не завершено успешно.
- *описание*: описание предложения.
- *ссылка*: *WSReference* созданного предложения. Пустой символ - если создание предложения не завершено успешно.
- *состояние*: экземпляр *WSRequestStatus*, указывающий на состояние создания предложения.



---

## Глава 4. Методы API SOAP

Службы IBM Campaign API SOAP используют следующие методы.

---

### Методы API SOAP: служба

Методика IBM Campaign API SOAP обеспечивает способ получить информацию об идентификации службы.

#### getServiceInfo

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

Возвращает информацию о службе, такой как актуальнейшая поддерживаемая версия API, полная версия базового экземпляра IBM Campaign.

**Примечание:** Никакая информация о клиенте не требуется для этого вызова, и никакие права доступа не применимы.

#### Параметры

Нет.

#### Возвращает

Возвращает экземпляр *WSServiceInfo*.

#### Ошибки

Нет.

---

### Методы API SOAP: Атрибуты

Большинство данных экземпляра компонента могут быть представлены IBM Campaign API SOAP как атрибуты или метаданные атрибута.

В некоторых случаях определения метаданных атрибута являются глобальной переменной для IBM Campaign (например, пользовательские атрибуты кампании). В других случаях они ограничены особым компонентом, таким как пользовательские переменные потоковой диаграммы. Если не указано иное, все атрибуты могут быть считаны при наличии у клиента достаточных прав доступа.

**Примечание:** Для этих API будут представлены только те компоненты, которые активны и к которым у клиента есть доступ. Общественная поддержка ограничена подмножеством доступных API.

#### getAttributesByName

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

Выбирает именованные атрибуты, связанные с указанным экземпляром компонента (может быть пустым).

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: дополнительная локаль для использования с этим требованием; если не предусмотрена, будут использоваться IBM предпочтения пользовательской локали. По мере необходимости применяется алгоритм по умолчанию нормальной локали.

*partitionName*: необязательное имя используемого раздела кампании. Если не определен, то используется раздел по умолчанию.

*ссылка*: ссылка для экземпляра компонента, содержащего требуемые атрибуты. *InvalidComponentException* сбрасывается, если ссылка недействительная или компонент не существует.

*имена*: дополнительный массив имен атрибутов для выборки (не показываемые имена); если не предусмотрен, будут возвращены все атрибуты. Сбрасывает *AttributeNotFoundException*, если ни один из именованных атрибутов не существует.

## Возвращает

Введенная карта ноля или больше атрибутов; название атрибута представляет собой ключ ввода карты, а экземпляр атрибута представляет собой входное значение.

## Ошибки

*InvalidComponentException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

**Примечание:** Все эти исключения упакованы в *CampaignServicesException*.

## updateAttributes

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

Обновите один или несколько атрибутов экземпляра компонента, пользуясь предоставленными значениями атрибута.

## Обновите логику

Логика обновления следующая.

Для каждого атрибута, содержащегося в предоставленной карте атрибута:

1. Если название атрибута соответствует существующему атрибуту, попытайтесь перезаписать поле *Значения* предоставленным полем значений.
2. Создайте атрибут, если атрибут не существует, значение *allowCreate* является true, а также известны его метаданные. Это относится к глобальным метаданным атрибута, а также атрибутам экземпляра (кроме потоковых диаграмм).
3. Если тип значения или любой другой аспект определения метаданных атрибута не предусмотрен, или один или несколько предоставленных значений являются недействительными или вышедшими за пределы диапазона, *InvalidAttributeException* будет накапливаться.
4. В противном случае, сбросьте *AttributeNotFoundException*, если именованный атрибут не существует.

**Примечание:** В случае исключительной ситуации ни одно обновление не принимается.

Этот конкретный метод не поддерживает определение новых пользовательских атрибутов; для этой цели используйте метод `createAttributeMetadata()`.

Во всех случаях операция по обновлению атрибута зависит от обычных ограничений по защите и от проверок. Задача клиента - определить атрибуты, которые будут необходимы для экземпляра конкретного компонента, правильных типов, и т.д.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*ссылка*: ссылка на экземпляр компонента, содержащего обновляемые атрибуты.

*allowCreate*: указывает на то, должен ли создаваться новый атрибут, если его все еще не существует для компонента. (Смотрите “Обновите логику” на стр. 22.)

*атрибуты*: обновляемый массив атрибутов: название атрибута используется для расположения обновляемого атрибута и новые значения используются для обновления значения существующего атрибута как отдельного объекта правильного типа или массива, если это применимо. (Смотрите Глава 5, “Общие исключения API SOAP”, на стр. 59.)

## Возвращает

Нет.

## Ошибки

`InvalidComponentException`, `AttributeNotFoundException`, `InvalidAttributeException`

`AuthorizationException`, `DataException`

## getAttributeMetadataByName

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

Выбирает определения именованных метаданных атрибутов, которые привязаны к особому компоненту, шаблону или глобальному определению.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*ссылка*: дополнительная ссылка компонента или шаблона, содержащего метаданные требуемых атрибутов. Если предоставлен только `ComponentTypeEnum`, то выборка будет ограничена компонентами этого типа. При отсутствии ссылки выборка

возвращает все глобальные определения метаданных для всех типов компонента. Сбрасывает *InvalidComponentException*, если предоставленная ссылка недопустима.

*имена*: дополнительный массив имен метаданных атрибута для выборки. При отсутствии все метаданные для компонента будут глобально возвращены, если отсутствует соответствующая ссылка. Сбрасывает *AttributeNotFoundException*, если один или больше указанных определений метаданных атрибута не существует.

## Возвращает

Введенная карта ноля или больше определений метаданных атрибута; название атрибута представляет собой ключ ввода карты, а метаданные атрибута представляют собой входное значение.

## Ошибки

*InvalidComponentException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

## createAttributeMetadata

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

Создайте один или несколько новых определений метаданных атрибута и дополнительно привяжите их к особому компоненту или шаблону.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*ссылка*: дополнительная ссылка компонента или шаблона, к которому должны быть привязаны метаданные. Если не предусмотрена, созданное определение метаданных будет в виде глобальной переменной. Если ссылка предусмотрена, но недействительная, то *InvalidComponentException* будет сброшен.

*attributeMetadata*: массив определений метаданных атрибута для привязки. Если к компоненту привязаны одни или несколько указанных метаданных, то есть, если имя не уникально, сбросьте *AttributeExistsException*. Сбрасывает *InvalidAttributeException*, если существует проблема с одними или несколькими указанными метаданными, то есть, при наличии внутренней несовместимости.

## Возвращает

Нет.

## Ошибки

*InvalidComponentException*, *AttributeExistsException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## updateAttributeMetadata

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

Обновите один или несколько определений метаданных атрибутов для указанного компонента или шаблона, необязательно создавая новые определения метаданных в случае необходимости.

### Логика обновлений

Логика обновлений следующая:

Для каждого определения метаданных атрибута, содержащегося в предоставленном массиве:

1. Если название атрибута не соответствует существующим метаданным, связанным с компонентом, выполните следующие шаги на основании значения параметра *allowCreate*:
  - a. *True*: создайте новое определение метаданных. Функционально идентичный использованию запроса `createAttributeMetadata ()`
  - b. *False*: генерация исключения *AttributeNotFoundException*.
2. Если тип метаданных атрибута отличается, генерировать исключение *InvalidAttributeException*.
3. Попытка перезаписи существующего определения метаданных атрибута с вводом значений полей предоставленных метаданных, в ином случае генерировать исключение *InvalidAttributeException*. Поддерживаются только следующие обновления (в противном случае генерация исключений *InvalidAttributeException*):
  - a. *Имя*: нельзя изменить (имя - это ключ!).
  - b. *displayName*: принять новое значение.
  - c. *Описание*: принять новое значение.
  - d. *isRequired*: позволить изменения только с *true* на *false*.
  - e. *isInternal*: принять новое значение.
  - f. *isGenerated*: изменения запрещены.
  - g. *attributeTypeEnum*: изменения запрещены.
  - h. *accessTypeEnum*: принять новое значение.
  - i. *selectTypeEnum*: принять данные переходы при предоставлении опций:
    - 1) ВЫБРАТЬ ОТСУТСТВУЕТ ДО ОДНОГО или ВЫБРАТЬ НЕСКОЛЬКО
    - 2) ВЫБРАТЬ ОДНОГО до ВЫБРАТЬ НЕСКОЛЬКО
  - j. *опции*: опции можно добавить, но не удалять. Поддерживаются лишь следующие изменения опции (согласно подборке значений):
    - 1) *displayName*: принять новое значение.
    - 2) *Описание*: принять новое значение (без цепи переноса).
    - 3) *isDefault*: принять новое значение; несмотря на это, должно соответствовать *SelectTypeEnum*.
    - 4) *значение*: изменения запрещены (значение-ключ!).
  - k. *defaultValue* (только потоковые диаграммы): принять новое значение по умолчанию.
  - l. *maxLength* (только текст): принять новую длину, если она больше.

4. Если определение метаданных атрибута является внутренне противоречивым, генерировать исключение *InvalidAttributeException*.
5. Если необходимо, найти все экземпляры компонента, ссылающиеся на обновленные метаданные атрибута и обновить в соответствии с необходимостью.

**Примечание:** В случае исключительной ситуации ни одно обновление не принимается.

Во всех случаях операция по обновлению атрибута зависит от обычных ограничений по защите и от проверок.

Смотрите `createAttributeMetadata()`, `deleteAttributeMetadata()`

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*ссылка*: необязательная ссылка на экземпляр компонента, который содержит требуемые атрибуты. Если отсутствует, это значит, что обновления глобальных определений метаданных ограничены. Генерировать исключение *InvalidComponentException*, если предоставленная ссылка не работает.

*allowCreate*: если true, создаются определения метаданных, не существующих на текущий момент (функционально идентичен использованию метода `createAttributeMetadata()`).

*attributeMetadata*: массив определений метаданных атрибута, подлежащих обновлению (и добавлению, если флаг *allowCreate* = true). Название атрибута используется для расположения определения метаданных, подлежащих обновлению, а оставшиеся данные используются для обновления существующего определения. (Смотрите “Логика обновлений” на стр. 25.)

## Возвращает

Нет.

## Ошибки

`InvalidComponentException`, `InvalidAttributeException`

`AuthorizationException`, `DataException`

## deleteAttributeMetadata

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

Удаляет один или несколько определений именованных метаданных атрибутов из указанного компонента, шаблон (только метаданные пользовательских атрибутов), или глобальные определения метаданных атрибутов.

Метод выполняет поиск ссылающихся на удаленные метаданные компонентов как часть этой задачи и надлежащим образом их обновляет.

**Примечание:** Однако в случае исключительной ситуации ни одно удаление не принимается.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*ссылка*: дополнительная ссылка компонента или шаблона, содержащего атрибуты, которые подлежат удалению. Если не предоставлено, удаление ограничено определениями глобальных метаданных. Сбрасывает *InvalidComponentException*, если предоставленная ссылка недопустима.

**Примечание:** Если дополнительный массив имен метаданных атрибута не предоставлен, то этот метод удалит все метаданные пользовательского атрибута, связанные с компонентом или всеми глобальными определениями, при отсутствии ссылки.

*имена*: дополнительный массив имен метаданных атрибута для удаления. Сбрасывает *AttributeNotFoundException*, если один или больше именованных метаданных атрибута не существует. Сбрасывает *InvalidAttributeException*, если атрибут не мог быть удален.

## Возвращает

Нет.

## Ошибки

*InvalidComponentException*, *AttributeNotFoundException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

---

## Методика SOAP API: Кампании и потоковые диаграммы

Методика IBM Campaign API SOAP поддерживает следующие операции по кампаниям и потоковым диаграммам (подлежит правам доступа).

- Создание кампании
- открытие (перечисляет кампании по различным критериям)
- создание, считывание и обновление атрибута (через атрибуты API)
- прекращение выполнения потоковой диаграммы

Кампании имеют много стандартных атрибутов, связанных с ними, которые представлены API. Этот список может быть расширен по желанию клиентом путем добавления пользовательских атрибутов (смотрите API атрибутов).

Стандартные атрибуты кампании приводятся ниже:

- *uacName*: имя кампании (не гарантируемое уникальное).
- *uacDescription*: дополнительная строка с описанием кампании.
- *uacCampaignCode*: код строки для уникального определения кампании. Обычно генерируется Campaign, но может быть предоставлен клиентом.
- *uacCreateDate*: Календарь с указанием даты и времени создания сервером кампании.

- *uacUpdateDate*: Календарь с указанием даты и времени обновления сервером кампании.
- *uacInitiative*: дополнительная строка с определением инициативы кампании.
- *uacObjectives*: дополнительная строка с определением цели кампании.
- *uacStartDate*: дополнительный календарь с указанием даты и времени запуска кампании сервером, или запланированного запуска.
- *uacEndDate*: как и в случае с *uacStartDate*, определяет дату и время завершения кампании, или запланированного завершения. Должен быть после *uacStartDate*.
- *uacLastRunDate*: дополнительный календарь с указанием даты и времени привязки любой потоковой диаграммы к кампании при последнем выполнении (пустой указатель).
- *uacExternalLinkOwner*: дополнительная строка с определением имени владельца внешней ссылки (смотрите атрибут *uacExternalLinkReference*). IBM используйте только; должно быть одно из следующего:
  - "План" (теперь известный как IBM Marketing Operations)
  - "Сотрудничество" (теперь известно как IBM Distributed Marketing)
- *uacExternalLinkId*: дополнительный числовой идентификатор базы данных, присвоенный другим приложением IBM к привязанному с этой кампанией объекту. IBM используйте только: смотрите также атрибут *uacExternalLinkOwner*

## generateCampaignCode

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

Генерирование нового кода кампании.

Этот код гарантируется уникальным и отличающимся от значения, которое возвращается по предыдущему или будущему вызову по этому методу или методу `createCampaign()` или значению, сгенерированному для кампании, которая была создана в графическом интерфейсе пользователя IBM Campaign.

**Примечание:** Использование этого метода дополнительное, так как `createCampaign()` API производит код кампании для клиента при отсутствии такового.

Смотрите `createCampaign()`.

### Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании. Если существует только один раздел в установке Campaign, то этот параметр может быть пустым.

### Возвращает

Произведенный код кампании.

### Ошибки

AuthorizationException, DataException



## deleteCampaigns

```
public WSDeleteCampaignsResponse deleteCampaigns(String userCredential,  
String partitionName, Locale requestedLocale, WSReference[] wsReferences)  
throws CampaignServicesException
```

Удаляет указанные кампании из системы.

### Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*wsReference*: идентификатор кампаний для удаления.

### Возвращает

Возвращает объект типа `WSDeleteCampaignsResponse`.

### Ошибки

Сбрасывает `CampaignServicesException`, если кампания не существует, или ссылка недействительная или не существует.

## createCampaign

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

Создайте новую кампанию для клиента, раздела и `securityPolicyName`, применив указанные атрибуты. Все кампании, созданные этим API, находятся под корневой папкой.

### Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*securityPolicyName*: дополнительное имя политики безопасности кампании для создания кампании. Все последующие операции в этой кампании используют эту политику. Если не определено, то используется Глобальная политика.

*Имя*: имя для присвоения нового экземпляра предложения (атрибут *uacName*).

*attributes*: массив атрибутов инициализации; любые предоставленные атрибуты перезаписывают значения предложения по умолчанию; другие остаются без

изменений. Например, если предусмотрен атрибут *uacCampaignCode*, он используется вместо автоматически созданного атрибута. Клиент сам определяет, какие атрибуты требуются кампанией, их типы, и т.д.

Сбрасывает *AttributeNotFoundException* если один или больше именованных атрибутов не существуют, или *InvalidAttributeException* если значение атрибута недействительное (например, неправильный тип данных).

## Возвращает

Единственный экземпляр *CampaignInfo* для созданной кампании.

## Ошибки

*InvalidAttributeException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

## listCampaignsByPage

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

Пронумеруйте "страницу" кампаний, соответствующих дополнительным значениям атрибута, начиная с указанного смещения на странице. Папки будут проигнорированы.

После извлечения каждое возвращение *CampaignInfo* может быть использовано как есть, например, для просмотра сводного списка, или методы атрибута могут быть использованы для выбора или обновления атрибутов предложения.

Никакое состояние не обслуживается этим API, таким образом, возможно выполнять вызовы к нему в любом порядке.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*атрибуты*: дополнительный массив атрибутов для соответствия; имя атрибута, тип данных и значения используются для определения соответствия; если атрибут поддерживает массивы, то все определенные значения должны соответствовать. Подразумеваемый оператор соответствий - ОСУ, таким образом, будут возвращены все компоненты, соответствующие любому из предоставленных значений атрибута.

Сбрасывает *AttributeNotFoundException*, если один или больше именованных атрибутов не существуют, или если значение атрибута недействительное (например, неправильный тип данных).

*pageOffset*: стартовое смещение всех возможных компонентов для начала перечисления (нулевое значение). Например, если перечисление соответствует 1 000 предложений, и это значение установлено на 10, то страница запустится с 11-го компонента. *RangeException* сбрасывается, если предоставленное смещение вне диапазона.

*pageSize*: максимальное количество подобранных компонентов для возврата страницы (не может превышать 500).

### **Возвращает**

Возвращает введенный Список с нулевыми или положительными экземплярами надстройки данных *Предложения*, по одному для каждой совпавшей кампании на странице.

### **Ошибки**

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

## **stopFlowchart**

stopFlowchart(int pid, int runid)

Этот API останавливает рабочую потоковую диаграмму. Для единственной конфигурации детектора выполненная потоковая диаграмма может быть однозначно определена по PID, связанным с выполненной потоковой диаграммой. PID указывает на идентификатор процесса unica\_acsvr. При конфигурировании IBM Campaign множественных детекторов необходимо включить идентификацию прогона, связанную со потоковой диаграммой, и PID.

### **Параметры**

*pid*: ID процесса unica\_acsvr - процесс, связанный с выполненной потоковой диаграммой.

*runid*: идентификация прогона, связанная с выполненной потоковой диаграммой. Обязательный параметр для кластеризованной конфигурации приемников. Необязательный параметр, если сконфигурирован единственный приемник.

### **Возвращает**

Отсутствует

### **Ошибки**

Отсутствует

---

## **Методы API SOAP: Ячейки назначения**

Ячейки назначения являются абстрактным понятием для обозначения определенных известных подмножеств результатов кампании, которыми управляет IBM Campaign в виде электронной таблицы ячеек назначения (TCS). Ячейки назначения могут быть глобальными по отношению к кампании или могут быть связаны с особой потоковой диаграммой кампании.

API SOAP IBM Campaign поддерживает следующие операции с ячейками назначения:

- создание одной или нескольких новых глобальных ячеек назначения
- массовое обновление одной или нескольких существующих ячеек назначения
- открытие (перечисление ячеек назначения)
- создание, чтение и обновление атрибутов (через API атрибута)
- удаление существующей ячейки назначения

- выбор результатов выполнения, связанных с одной или несколькими ячейками

Ячейки назначения имеют много стандартных атрибутов, связанных с ними, которые представлены API. Клиент может расширить этот список по желанию путем добавления определений метаданных пользовательских атрибутов (смотрите API атрибутов). Метаданные каждого атрибута могут считаться столбцом в TCS; макет электронной таблицы определяется клиентом.

Стандартные атрибуты ячейки назначения:

- *uacName*: имя ячейки
- *uacDescription*: дополнительная строка, описывающая потоковую диаграмму.
- *uacCellCode*: строка кода, однозначно идентифицирующая ячейку. Обычно она генерируется автоматически в Campaign, но может быть предоставлена клиентом.
- *uacCreateDate*: экземпляр Календаря, который предоставляет дату & и время создания ячейки сервером.
- *uacUpdateDate*: Экземпляр Календаря, определяющий, когда ячейка была обновлена сервером в прошлый раз.
- *uacIsControl*: логическая переменная, указывающая, является ли эта ячейка контрольной ячейкой (true) или нет (false). Другие ячейки могут ссылаться на эту ячейку как контрольную ячейку (смотрите *uacControlCell*).
- *uacControlCell*: дополнительная ссылка контрольной ячейки (ссылка на контрольную ячейку не допускается). Смотрите атрибут *uacIsControl*.
- *uacIsApproved*: логическая переменная, указывающая, одобрена ячейка (true) или нет (false).
- *uacIsReadOnly*: логическая переменная, указывающая, доступна ли ячейка только для чтения (true) или нет (false).
- *uacDisplayOrder*: целое число, определяющее порядок этой ячейки (строки) относительно других элементов в электронной таблице ячеек назначения.
- *uacIsTopDown*: логическая переменная, указывающая, является ли ячейка нисходящей.
- *uacAssignedOffers*: дополнительный массив из одной или нескольких ссылок предложений или списков предложений, присвоенный этой ячейке (не допускается для контрольной ячейки).
- *uacFlowchartName*: дополнительное имя потоковой диаграммы, с которой соединена эта ячейка (атрибут "только для чтения" необходимо установить с помощью ГИП IBM Campaign; не допускается для контрольной ячейки).
- *uacFlowchartId*: дополнительный идентификатор базы данных для потоковой диаграммы, определяющий то, с которым соединена эта ячейка (только для чтения, так же как и выше).

## createTargetCell

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

Создайте новую определенную для кампании строку целевой ячейки, применив указанные атрибуты ячеек и информацию о пользователе.

Указанные атрибуты могут быть стандартными или пользовательскими; при использовании пользовательских атрибутов должны существовать соответствующие определения глобальных метаданных атрибута.

Когда целевая ячейка создана, значения атрибута можно изменить при помощи API атрибутов.

Смотрите `listTargetCells()`, `bulkCreateTargetCells()`.

Смотрите информацию о `createAttributeMetadata()`, `listAttributeMetadata()`, `getAttributesByName()`

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*campaignReference*: ссылка кампании, которая содержит электронную таблицу целевой ячейки, которая будет обновлена. Накапливается *InvalidComponentException*, если кампания не существует или ссылка недействительная.

*атрибуты*: дополнительный массив TCS определяет характеристики новой ячейки. Каждый предоставленный элемент атрибута перезаписывает значения по умолчанию соответствующего атрибута ячеек; другие без изменений. Клиент вправе на свое усмотрение определить требуемые для ячеек, их типов и т.д атрибуты. Накапливает *InvalidAttributeException*, если существует проблема с указанным атрибутом.

Если накоплены какие-либо исключения, этот метод сбрасывает *CompositeException*, а все созданные элементы будут отменены. Список исключения причин включает в себя исключение для каждого атрибута, вызвавшего ошибку, а также числовой индекс вместо *ссылки*, имя атрибута и обычно недействительное значение. Список причин имеет очередность согласно входу *attributeList*.

## Возвращает

А *TargetCellInfo* надстройка данных для созданной ячейки TCS.

## Ошибки

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## bulkCreateTargetCells

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

Создайте множество новых определенных для кампании строк целевой ячейки одновременно, применив указанные атрибуты ячеек и информацию о пользователе.

Указанные атрибуты могут быть стандартными или пользовательскими; при использовании пользовательских атрибутов должны существовать соответствующие определения глобальных метаданных атрибута.

Когда целевая ячейка создана, значения атрибута можно изменить при помощи API атрибутов.

Смотрите `listTargetCells()`.

Смотрите информацию о `createAttributeMetadata()`, `listAttributeMetadata()`, `getAttributesByName()`

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*campaignReference*: ссылка кампании, которая содержит электронную таблицу целевой ячейки, которая подлежит обновлению. Накапливает *InvalidComponentException*, если кампания не существует или ссылка недействительная.

*attributeList*: дополнительный список массивов на атрибут ячеек, по одному на каждую строку целевой ячейки, которая будет создана. Любые предоставленные атрибуты особого элемента списка перезаписывают значения по умолчанию соответствующего атрибута ячеек; другие остаются без изменений. Клиент вправе на свое усмотрение определить требуемые для ячеек, их типов и т.д атрибуты. Накапливает *InvalidAttributeException*, если существует проблема с указанным атрибутом.

Если накоплены какие-либо исключения, этот метод сбрасывает *CompositeException*, а все созданные элементы будут отменены. Список исключения причин включает в себя исключение для каждого атрибута, вызвавшего ошибку, а также числовой индекс вместо *ссылки*, имя атрибута и т.д. Список причин появляется с входом *attributeList*.

## Возвращает

Список надстроек данных *TargetCellInfo* создается для каждого созданного экземпляра по очередности элемента параметра входа *attributesList*.

## Ошибки

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## listTargetCells

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

Приводит перечень сведений обо всех в настоящее время существующих целевых ячейках, которые соответствуют указанным атрибутам, или об указанной кампании, или глобально, если кампания не определена.

Смотрите `getAttributeMetadata()`, `getAttributesByName()`.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*campaignReference*: ссылка на родительскую кампанию. Сбрасывает *InvalidComponentException*, если кампания не существует или ссылка недействительная.

*атрибуты*: дополнительный массив атрибутов для соответствия. Подразумеваемый оператор соответствий - ОСУ, таким образом, будут возвращены все компоненты, соответствующие любому из предоставленных значений атрибута.

Сбрасывает *InvalidAttributeException*, если один или больше указанных атрибутов недействительны.

## Возвращает

Возвращает список с нулевым или положительным значением экземпляров *TargetCellInfo* для совпавших ячеек.

## Ошибки

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## bulkUpdateTargetCells

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

Обновите атрибуты одной или более целевых ячеек.

Логика обновления следующая.

Для каждого элемента в предоставленном *attributesMap*, ключ записи представляет собой ссылку целевой ячейки для обновления, а входное значение представляет собой массив атрибутов обновления для такой ячейки. Если целевая ячейка не существует, *InvalidComponentException* будет накапливаться.

Когда целевая ячейка найдена для каждого заданного атрибута, сделайте следующее:

1. Если имя атрибута соответствует существующему атрибуту, попытайтесь перезаписать ее поле значений с предоставленным полем значений.
2. Если тип значения или любой другой аспект определения метаданных атрибута не предусмотрен, или один или несколько предоставленных значений являются недействительными или вышедшими за пределы диапазона, *InvalidAttributeException* будет накапливаться.
3. В противном случае будет накапливаться атрибут *AttributeNotFoundExpection*, если именованный атрибут не существует.

При накоплении каких-либо исключений, этот метод сбросит *CompositeException*, а все обновления будут отменены. Список причин исключительной ситуации включает упомянутые выше исключительные ситуации. Для каждого атрибута, вызвавшего ошибку, записывается и ссылка, и имя атрибута.

Во всех случаях операция по обновлению атрибута зависит от обычных ограничений по защите и от проверок. Задача клиента - определить атрибуты, которые будут необходимы для экземпляра конкретного компонента, правильных типов, и т.д.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*attributesMap*: ключ записи представляет собой ссылку целевой ячейки для обновления, а входное значение представляет собой массив атрибутов обновления для такой ячейки. Имя атрибута используется для расположения атрибута для обновления, а новые значения атрибута используются для обновления значения существующего атрибута как отдельного объекта надлежащего типа или массива, по мере возможности. Смотрите исключения выше.

## Возвращает

Нет.

## Ошибки

ComponentException

AuthorizationException, DataException

## getRunResultsByCell

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

Получите результаты выполнения одного или нескольких целевых ячеек для потоковой диаграммы, которые не были запущены или еще в процессе выполнения.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*cellReferences*: массив ссылок целевых ячеек, результаты выполнения которых являются желаемыми. Сбрасывает *InvalidComponentException*, если одна или несколько ссылок на ячейки являются недействительными или имеется несуществующая ячейка.

## Возвращает

Возвращает список с типами выполненных результатов для именованных ячеек, упорядоченных согласно входному массиву ссылок.



Каждое состояние выполнения РАБОТАЕТ, если базовое поле процесса потоковой диаграммы по-прежнему работает, а также ТЕРПИТ НЕУДАЧУ, если выполнение потерпело неудачу по некоторым причинам, или выдает состояние NOT\_STARTED, если выполненное поле процесса не запустилось. Также приводится подробная информация о состоянии.

## Ошибки

InvalidComponentException

AuthorizationException, DataException

## bulkDeleteTargetCells

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

Удаляет одну или несколько существующих целевых ячеек и все их зависимые компоненты (связь потоковой диаграммы, атрибуты).

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*cellReferences*: массив или одна или несколько ссылок ячеек, которые подлежат удалению. *InvalidComponentException* будет накоплен, если существует проблема с одной из указанных ссылок, или при отсутствии ячейки.

При накоплении каких-либо исключений этот метод сбрасывает *CompositeException* и отменяет все элементы для удаления. Список причин исключительной ситуации включает упомянутые выше исключительные ситуации. Для каждой ячейки, вызвавшей ошибку, записывается ссылка.

## Возвращает

Нет.

## Ошибки

CompositeException

AuthorizationException, DataException

## updateTemplateAttributes

```
updateTemplateAttributes  
(String userCredential, String partitionName, Locale requestedLocale,  
WSReference wsReference, boolean allowCreate,  
boolean clearExisting, WSAttribute[] wsStaticAttributes,  
WSAttribute[] wsHiddenAttributes, WSAttribute[] wsParametricAttributes)  
    генерирует исключение CampaignServicesException
```

Обновляет атрибуты указанных шаблонов

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*wsCampaignReference*: ссылка на родительскую кампанию.

*allowCreate*: на данный момент не используется.

*clearExisting*: если флаг установлен на true, то все предыдущие значения шаблона, которые не были направлены в запросе, удаляются.

*wsStaticAttributes*: список статических атрибутов шаблона.

*wsHiddenAttributes*: список скрытых атрибутов шаблона.

*wsParametricAttributes*: список параметрических атрибутов шаблона.

## Возвращает

Нет.

## Ошибки

Генерирует исключение CampaignServicesException, если шаблона предложения не существует или ссылка не работает или не была предоставлена.

## listBottomUpTargetCells

```
public List <WSTargetCellDetails>  
listBottomUpTargetCells(String userCredential, String partitionName,  
Locale requestedLocale, WSReference wsCampaignReference)  
throws CampaignServicesException
```

Приводит список сведений обо всех целевых ячейках снизу вверх, которые имеются в настоящее время для указанной кампании.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*wsCampaignReference*: ссылка на родительскую кампанию.

## Возвращает

Возвращает список с нулевым или положительным значением экземпляров WSTargetCellDetails для совпавших ячеек

## Ошибки

Генерирует CampaignServicesException, если кампания не существует или ссылка недопустима.

---

## Методы API SOAP: Анализ

Опция IBM Campaign API SOAP поддерживает извлечение простых метрик из IBM Campaign.

### getCampaignMetrics

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

Выберите метрики для указанной кампании.

#### Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*campaignReference*: ссылка родительской кампании. Сбрасывает *InvalidComponentException* при наличии проблемы со ссылкой кампании, или если кампания не существует.

#### Возвращает

Возвращает экземпляр *MetricsInfo* для кампании.

#### Ошибки

InvalidComponentException

AuthorizationException, DataException

---

## Методы API SOAP: Предложения, списки предложений, шаблоны предложений

API SOAP IBM Campaign поддерживает следующие операции, связанные с предложениями.

- Обнаружение: перечисление по папкам (предложения, списки предложений и подпапки), атрибут (предложения и шаблоны предложений), или значение поиска (предложения)
- проверка
- информационный поиск (получить атрибуты для определенного предложения или шаблона предложений)
- создание, редактирование, выведение из использования и удаление предложений

Предложения имеют много стандартных атрибутов, связанных с ними. Этот список может быть расширен клиентом путем добавления определений метаданных пользовательского атрибута (смотрите API атрибутов).

Стандартные атрибуты предложения:

- *uacName*: имя предложения
- *uacDescription*: дополнительная строка с описанием предложения.
- *uacOfferCode*: строка кода, однозначно определяющая предложение. Обычно она генерируется в IBM Campaign, но может быть предоставлена клиентом.
- *uacCreateDate*: экземпляр Календаря, в котором указаны дата и время создания предложения сервером.
- *uacUpdateDate*: экземпляр Календаря, в котором указано время последнего обновления предложения сервером.

У шаблонов предложений также есть стандартные и пользовательские атрибуты.

Атрибуты стандартного шаблона предложения:

- *uacName*: имя шаблона предложения
- *uacDescription*: дополнительная строка с описанием шаблона предложения.
- *uacCreateDate*: экземпляр Календаря, в котором указаны дата и время создания шаблона предложения сервером.
- *uacUpdateDate*: экземпляр Календаря, в котором указано время последнего обновления шаблона предложения сервером.

## listOffersAndFolders

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

Перечислите все предложения, списки предложений и папки под необязательной родительской папкой.

После извлечения каждое возвращение *WSComponentOrFolderInfo* может быть использовано как есть, например, чтобы показать сводный список, или методы атрибута могут быть использованы для выбора или обновления атрибутов предложения.

### Параметр

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*parentReference*: дополнительная ссылка родительской папки для списка. Только непосредственные дочерние предложения, списки предложения и папки этой родительской папки будут пронумерованы, таким образом, последовательные вызовы для этой API будут необходимыми для навигации по всей иерархии папок (обычно очень поверхностно). Если никакой родитель не представлен, то возвращаются все компоненты и папки под корнем.

Сбрасывает *InvalidFolderException* при наличии проблемы с указанной ссылкой родительской папки.

Введенный *Список* с нулевыми или положительными экземплярами надстройки данных *WSComponentOrFolderInfo*, по одному для каждого совпавшего компонента или папки.

## Ошибки

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;
```

Пронумеруйте "страницу" предложений, содержащих данные критерии поиска по полям имени, описания, createBy или кода предложения, начиная с указанного смещения на странице. Поиск основан на дополнительном вводе для Папки. (Если идентификатор папки установлен на 0, по умолчанию используется корневая папка предложения). Совпадения возвращаются на основе "содержащих" совпадений для строки поиска.

После извлечения каждое возвращение *WSOfferInfo* может быть использовано как есть, например, для просмотра сводного списка, или методы атрибута могут быть использованы для выбора или обновления атрибутов предложения.

Никакое состояние не обслуживается этим API, таким образом, возможно выполнять вызовы к нему в любом порядке.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*folderID*: идентификатор папки Предложение, который должен быть найден; если идентификатор папки равен 0, будет выполняться поиск корневой папки.

*searchCriteria*: поисковая фраза.

*includeRetired*: булево значение, определяющее, включают ли результаты поиска удаленные предложения. Допустимые значения - TRUE и FALSE, где первое указывает на включение сброшенных предложений, а второе - на отсутствие включения предложений FALSE.

*pageOffset*: стартовое смещение всех возможных компонентов для начала перечисления (нулевое значение). Например, если перечисление соответствует 1 000 предложений, и это значение установлено на 10, то страница запустится с 11-го компонента. Если предоставленное смещение вне диапазона, то генерируется RangeException.

*pageSize*: максимальное количество подобранных компонентов для возврата страницы (не может превышать 500).

## Возвращает

Возвращает введенный Список с нулевыми или положительными экземплярами надстройки данных *Предложения*, по одному для каждого возвращенного предложения на странице.

## Ошибки

RangeException

## listOffersByPage

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

Пронумеруйте "страницу" предложений, соответствующих дополнительным значениям атрибута, начиная с указанного смещения на странице. Папки будут проигнорированы. Совпадения будут возвращены на основе "подобного" совпадения для строк (где совпадение считается достаточным, если строка содержит запрошенное значение), а также на основе точного совпадения для дат и чисел.

После извлечения каждое возвращение *OfferInfo* может быть использовано как есть, например, чтобы показать сводный список, или методы атрибута могут быть использованы для выбора или обновления атрибутов предложения.

Никакое состояние не обслуживается этим API, таким образом, возможно выполнять вызовы к нему в любом порядке.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*атрибуты*: дополнительный массив атрибутов для соответствия; имя атрибута, тип данных и значения используются для определения соответствия; если атрибут поддерживает массивы, то все определенные значения должны соответствовать. Подразумеваемый оператор соответствий - ОСУ, таким образом, будут возвращены все компоненты, соответствующие любому из предоставленных значений атрибута.

Сбрасывает *AttributeNotFoundException*, если один или больше именованных атрибутов не существуют, или если значение атрибута недействительное (например, неправильный тип данных).

*pageOffset*: стартовое смещение всех возможных компонентов для начала перечисления (нулевое значение). Например, если перечисление соответствует 1 000 предложений, и это значение установлено на 10, то страница запустится с 11-го компонента. *RangeException* сбрасывается, если предоставленное смещение вне диапазона.

*pageSize*: максимальное количество подобранных компонентов для возврата страницы (не может превышать 500).

## Возвращает

Введенный *Список* с нулевыми или положительными экземплярами надстройки данных *OfferInfo*, по одному для каждого совпавшего компонента на странице.

## Ошибки

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

## createSmartOfferList

```
public WSCreateSmartOfferListResponse createSmartOfferList
(String userCredential, String partitionName, Locale requestedLocale,
Имя строки, описание строки, policyName строки,
WSReference parentFolder, WSSmartListInfo offerListInfo,
WSApplicationTypeEnum createdBy, long creatorObjectId)
сбрасывает CampaignServicesException {
```

Создает новый интеллектуальный список предложений.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*имя*: имя папки для создания.

*описание*: описание для новой папки.

*securityPolicyName*: имя политики безопасности для использования.

*parentFolder*: идентификатор папки, в которой должен быть создан список предложений.

*offerListInfo*: объект типа WSSmartListInfo.

*createdBy*: (дополнительный) объект типа указывает WSApplicationTypeEnum на приложение для создания папки. Возможные значения - Campaign/Plan/Collaborate/eMessage. Если не определено, то используется Campaign.

*creatorObjectId*: (дополнительный) используется планом (Маркетинговые операции IBM) для установления соединения с папкой в IBM Campaign с папкой в Маркетинговых операциях.

## Возвращает

Возвращает объект типа WSCreateSmartOfferListResponse.

## Ошибки

Генерирует CampaignServicesException, если ID parentFolder недопустим или не предоставлена offerListInfo.

Генерирует CampaignServicesException, если имя списка - дубликат.

## createStaticOfferList

```
public WSCreateStaticOfferListResponse createStaticOfferList
(String userCredential, String partitionName, Locale requestedLocale, String name,
Описание строки, policyName строки, WSReference parentFolder, WSReference []
listMembers, WSApplicationTypeEnum createdBy, long creatorObjectId)
throws CampaignServicesException
```

Создает новый статический список предложений.

### Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*имя*: имя папки для создания.

*описание*: описание для новой папки.

*securityPolicyName*: имя политики безопасности для использования.

*parentFolder*: идентификатор папки, в которой должен быть создан список предложений.

*listMembers*: ссылки на предложения, которые будут включены в список предложений.

*createdBy*: (дополнительный) объект типа указывает WSApplicationTypeEnum на приложение для создания папки. Возможные значения - Campaign/Plan/Collaborate/eMessage. Если не определено, то используется Campaign.

*creatorObjectId*: (дополнительный) используется планом (Маркетинговые операции IBM) для установления соединения с папкой в IBM Campaign с папкой в Маркетинговых операциях.

### Возвращает

Возвращает объект типа WSCreateStaticOfferListResponse.

### Ошибки

Генерирует CampaignServicesException, если ID parentFolder или listMembers недопустимы.

Генерирует CampaignServicesException, если имя списка - дубликат.

## getOffers

```
public WSGetOffersResponse getOffers
(String userCredential, String partitionName,
Locale requestedLocale, WSReference[] wsReferences)
throws CampaignServicesException
```

Приводит список подробностей о предложениях, как определено в запросе.



## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*wsCampaignReference*: ссылка на родительскую кампанию.

## Возвращает

Возвращает объект типа `WSGetOffersResponse`.

## Ошибки

Сбрасывает `CampaignServicesException`, если предложение не существует, или ссылка недоступна или недействительная.

## validateOffers

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

Проверьте предоставленные коды предложения или имена списков предложений и информацию о проверке возврата для каждого. "Подтверждение валидности" состоит из проверки факта существования лишь одного существования подходящего предложения или списка предложений в базе данных.

Объект `OfferValidationInfo` содержит сообщение с ошибкой вместо информации о предложении, если не было найдено совпадений каких-либо предложений или списков предложений с указанными кодом или названием. Ошибка также возвращается вместо совпадения, если указанные код или названия соответствуют нескольким предложениям или спискам предложений. Список возвращается в том же порядке, в каком он был предоставлен. Коды предложений и имена списков предложений проверяются на основе на точного совпадения с предложениями.

## Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*codeOrNames*: массив всех кодов предложения или названий списков предложений, валидность которых проверяется.

**Примечание:** Никакие исключения данным методом не генерируются; вместо этого информация о валидации возвращается для всех предоставленных кодов или названий.

## Возвращает

Введенный Список несуществующих или нескольких `OfferValidationInfo` экземпляров оболочки данных.

## Ошибки

Нет.

## editOfferList

```
public WSEditOfferListResponse editOfferList(String userCredential,  
String partitionName, Locale requestedLocale, WSReference listReference,  
boolean isSmartList, String name, String description,  
WSReference[] listMembers, WSSmartListInfo offerListInfo,  
Long creatorObjectId, boolean clearExisting)  
throws CampaignServicesException
```

Обновляет сведения об интеллектуальных и статических списках предложений.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*listReference*: ссылка на список предложений.

*isSmartList*: флаг указывает на интеллектуальный или статичный список.

*имя*: имя папки для создания.

*описание*: описание для новой папки.

*listMembers*: ссылки на предложения, которые будут включены в список предложений.

*offerListInfo*: объект типа WSSmartListInfo.

*creatorObjectId*: (дополнительный) используется планом (Маркетинговые операции IBM) для установления соединения с папкой в IBM Campaign с папкой в Маркетинговых операциях.

*clearExisting*: флаг указывает на необходимость очистки существующей информации. Если это так, существующие элементы списка будут очищены перед добавлением новых элементов. Если false, то новые элементы будут присоединяться к существующим элементам.

## Возвращает

Возвращает объект типа WSEditOfferListResponse.

## Ошибки

Сбрасывает CampaignServicesException, если недействителен идентификатор parentFolder или listMembers, или отсутствует идентификатор offerListInfo.

Генерирует CampaignServicesException, если имя списка - дубликат.

## 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;
```

Создайте новое предложение для клиента, применив указанные атрибуты.

### Параметры

*authorizationLoginName*: Имя пользователя пользователя, создающего предложение. Чтобы использовать этот метод, пользователи должны получить разрешение на Добавление предложений.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*securityPolicyName*: дополнительное имя политики безопасности кампании для создания предложения. Все последующие операции в этом предложении используют эту политику. Если не определено, то используется *Глобальная* политика.

*Имя*: имя для присвоения нового экземпляра предложения (атрибут *uacName*).

*folderID*: идентификатор папки Предложение, в которой создано предложение. Этот ID проверяется на правильность, и если ID недопустим, создается исключительная ситуация.

*templateName*: необходимое (уникальное) имя существующего шаблона предложения, который должен использоваться для нового предложения.

*wsAttributes*: массив атрибутов инициализации; любые предоставленные атрибуты перезаписывают значения предложения по умолчанию; другие остаются без изменений. Например, если предусмотрен атрибут *uacOfferCode*, он используется вместо автоматически созданного атрибута. Клиент сам определяет, какие атрибуты требуются предложением, их типы, и т.д.

Сбрасывает *CampaignServicesException*, если происходит одно из следующих условий:

- Параметр *folderID* недопустим (не существует или не соответствует предложению типа).
- У пользователя нет полномочий для выполнения этой операции.
- В *wsAttributes* представлены недопустимые атрибуты.
- Другие исключения происходят на этапе выполнения.

### Возвращает

Единственный экземпляр *OfferInfo* для созданного предложения.

### Ошибки

*CampaignServicesException*

## retireOffers

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

Удаление одного или нескольких существующих предложений.

### Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*ссылки*: массив ссылок предложений, которые подлежат удалению.

*InvalidComponentException* будет сброшен, если существует проблема с конкретной ссылкой, или предложение не существует.

### Возвращает

Нет.

### Ошибки

InvalidComponentException

AuthorizationException, DataException

## deleteOffers

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

Удаляет одно или несколько существующих предложений.

### Параметры

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

*ссылка*: массив ссылок предложений, которые подлежат удалению.

*InvalidComponentException* будет сброшен, если существует проблема с ссылкой, или предложение не существует.

### Возвращает

Нет.

### Ошибки

InvalidComponentException

AuthorizationException, DataException

## deleteOffersAndLists

```
public WSDeleteOffersAndListsResponse deleteOffersAndLists  
(String userCredential, String partitionName, Locale requestedLocale,  
WSReference[] offers)  
throws CampaignServicesException
```

Удаляет указанные предложения и списки.

### Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*предложения*: массив ссылок или предложения или списка предложения.

### Возвращает

Возвращает объект типа WSGetOfferListMembersResponse.

### Ошибки

Генерирует CampaignServicesException, если ID предложения или ID списка предложений недопустимы.

## listOfferTemplates

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

Перечислите все шаблоны предложений, на которые у пользователя есть полномочия на просмотр.

После извлечения каждое возвращение *WSOfferTemplateInfo* может быть использовано как есть, например, чтобы показать сводный список, или методы атрибута могут быть использованы для выбора или обновления атрибутов предложения.

### Параметр

*userCredential*: идентификационные данные клиента.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*partitionName*: необязательное имя используемого раздела кампании.

### Возвращает

Введенный Список с нулевыми или положительными экземплярами надстройки данных *WSOfferTemplateInfo*, по одному для каждого возвращенного шаблона.

### Ошибки

InvalidExecutionContextException, AuthorizationException

DataException

## createTemplate

```
createTemplate(String userCredential, String partitionName, Locale requestedLocale,  
String name, String securityPolicyName, WSAttribute[]  
wsStaticAttributes, WSAttribute[] wsHiddenAttributes,  
WSAttribute[] wsParametricAttributes)  
throws CampaignServicesException
```

Создает новый шаблон предложения.

### Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*имя*: имя недавно созданного шаблона предложения.

*securityPolicyName*: имя политики безопасности для использования.

*wsStaticAttributes*: список статических атрибутов в шаблоне.

*wsHiddenAttributes*: список скрытых атрибутов в шаблоне.

*wsParametricAttributes*: список параметрических атрибутов в шаблоне.

### Возвращает

Возвращает объект типа WSCreateTemplateResponse.

### Ошибки

Сбрасывает CampaignServicesException, если шаблон предложения не существует или ссылка недоступна или недействительная.

## getOfferTemplate

```
public WSGetOfferTemplateResponse getOfferTemplate(String userCredential,  
String partitionName, Locale requestedLocale, WSReference[] wsReferences)  
throws CampaignServicesException
```

Приводит список подробностей о шаблонах предложений, как определено в ссылках.

### Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*wsCampaignReference*: ссылка на родительскую кампанию.

### Возвращает

Возвращает объект типа WSGetOfferTemplateResponse.

## Ошибки

Сбрасывает CampaignServicesException, если шаблон предложения не существует или ссылка недоступна или недействительная.

## retireOfferTemplates

```
public WSGenerateOfferCodeResponse generateOfferCodes  
(String userCredential, String partitionName, Locale requestedLocale,  
Строка шаблонов offerName, WSReference)  
throws CampaignServicesException
```

Выводится из использования один или несколько указанных шаблонов предложений.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*wsCampaignReference*: ссылка на родительскую кампанию.

## Возвращает

Возвращает объект типа WSRetireOfferTemplatesResponse.

## Ошибки

Сбрасывает CampaignServicesException, если шаблон предложения не существует или ссылка недоступна или недействительная.

## getOffersAndListsByPage

```
public WSGetOffersAndListsByPageResponse getOffersAndListsByPage  
(String userCredential, String partitionName, Locale requestedLocale,  
WSCoponentTypeEnum type,  
int pageSize, int pageOffset)  
throws CampaignServicesException
```

Список предложений или список предложений по страницам.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*ведите*: тип, определяющий необходимость запроса предложений или списков предложений.

*pageSize*: максимальное количество подобранных компонентов для возврата страницы.

*pageOffset*: стартовое смещение всех возможных компонентов для начала перечисления (нулевое значение). Например, если перечисление соответствует 1 000 предложений, и это значение установлено на 10, то страница запустится с 11-го компонента. Если предоставленное смещение вне диапазона, то генерируется RangeException.

## Возвращает

Возвращает объект типа WSGetOffersAndListsByPageResponse.

## Ошибки

Нет.

## bulkCreateOffers

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

Создает предложения большими партиями с атрибутами для каждого предложения, определенных в параметре *предложения*. Все предложения будут созданы под указанным *FOLDERID* при помощи указанного параметра *TemplateName*.

## Параметр

*authorizationLoginName*: учетные данные пользователя клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*securityPolicyName*: дополнительное имя политики безопасности кампании для создания предложения. Если не определено, то используется Глобальная политика.

*templateName*: имя существующего шаблона предложения в системе. Все предложения созданы с помощью этого шаблона.

*folderID*: идентификатор папки Предложение, в которой создано предложение. Этот ID проверяется, и если ID недопустим, создается исключительная ситуация.

*предложения*: массив объектов *WSBulkOfferInfo*, который определяет имя предложения и атрибуты. Дополнительную информацию смотрите в типе данных *WSBulkOfferInfo*.

## Возвращает

Массив экземпляров *WSOfferInfoStatus* для каждого предложения. Содержит информацию о предложении и состоянии. Состояние указывает, создано предложение успешно или нет.

## Ошибки

CampaignServicesException



## getOfferListDetails

```
public WSGetOfferListDetailsResponse getOfferListDetails(String userCredential,
String partitionName, Locale requestedLocale, WSReference listReference)
throws CampaignServicesException {
```

Перечислите сведения указанного списка предложений.

### Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*listReference*: ссылка на список предложений.

### Возвращает

Возвращает объект типа WSGetOfferListDetailsResponse.

### Ошибки

Генерирует CampaignServicesException, если ссылки списка недопустимы.

## getOfferListMembers

```
public WSGetOfferListMembersResponse getOfferListMembers
(String userCredential, String partitionName, Locale requestedLocale,
WSReference listReference)
throws CampaignServicesException {
```

Перечисляется информация о предложениях в указанном списке предложения.

### Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*listReference*: ссылка на список предложений.

### Возвращает

Возвращает объект типа WSDeleteOffersAndListsResponse.

### Ошибки

Генерирует CampaignServicesException, если ID списка предложений недопустим.

## getOffersByQuery

```
public WSGetOffersByQueryResponse getOffersByQuery(String user_credential,
String partition_name, Locale locale, String query, Integer maxSize,
Boolean includeSubFolder, WSReference[] scopeFolders)
throws CampaignServicesException
```

Списки предложений, которые соответствуют предоставленным предложениям.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*запрос*: запрос на поиск предложений. Формат запроса - тот же, что и в интеллектуальном списке предложений.

*maxSize*: максимальное количество записей в списке.

*includeSubFolder*: флаг, указывающий на необходимость включение вложенной папки в поиск.

*scopeFolders*: список ссылок папки, для которых будет выполнен поиск для предложений.

## Возвращает

Возвращает объект типа `WSGetOffersByQueryResponse`.

## Ошибки

Генерирует `CampaignServicesException`, если ссылки папки недопустимы.

## retireOfferLists

```
public void retireOfferLists(String user_credential, String partition_name,  
Locale locale, WSReference[] wsReferences)  
throws CampaignServicesException
```

Выводится из использования один или несколько указанных списков предложений.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*wsReferences*: ссылка для списков предложений.

## Возвращает

Нет.

## Ошибки

Генерирует `CampaignServicesException`, если ссылки списка недопустимы.

## createFolder

```
public WSCreateFolderResponse createFolder(String userCredential,  
String partitionName, Locale requestedLocale,String name,  
Описание строки, securityPolicyName строки,  
long parentFolderId,WSFolderTypeEnum folderType,  
WSApplicationTypeEnum createdBy,long creatorObjectId)  
throws CampaignServicesException
```

Создает новую папку типа кампании/предложения/сеансов/сегментов.

### Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*имя*: имя папки для создания.

*описание*: описание для новой папки.

*securityPolicyName*: имя политики безопасности для использования.

*ParentFolderID*: (дополнительный) идентификатор родительской папки. Если не определено, то папка создается как корневая.

*folderType*: тип folder-Offer/session/campaign/segment.

*createdBy*: (дополнительный) объект типа указывает WSApplicationTypeEnum на приложение для создания папки. Возможные значения: Campaign, План, Сотрудничество и eMessage. Если не определено, то используется Campaign.

**Примечание:** Plan=Маркетинговые операции IBM. Collaborate=Распределенный маркетинг IBM.

*creatorObjectId*: (дополнительный) используется планом (Маркетинговые операции IBM) для установления соединения с папкой в IBM Campaign с папкой в Маркетинговых операциях.

### Возвращает

Возвращает объект типа WSCreateFolderResponse.

### Ошибки

Генерирует CampaignServicesException, если тип папки недопустим или это дубликат.

## editFolder

```
public WSEditFolderResponse editFolder(String userCredential,  
String partitionName, Locale requestedLocale, long folderId,  
String name,String description, WSFolderTypeEnum folderType,  
Long creatorObjectId, boolean clearExisting)  
сбрасывает CampaignServicesException {
```

Обновляет указанную папку.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*id*: идентификатор папки для обновления.

*имя*: имя папки для создания.

*описание*: описание для новой папки.

*folderType*: тип folder-Offer/session/campaign/segment.

*creatorObjectId*: (дополнительный) используется планом (Маркетинговые операции IBM) для установления соединения с папкой в IBM Campaign с папкой в Маркетинговых операциях.

*clearExisting*: не используется в настоящее время.

## Возвращает

Возвращает объект типа WSEditFolderResponse.

## Ошибки

Генерирует CampaignServicesException, если тип папки недопустим или это дубликат или если ID папки недопустим.

## getSubFoldersList

```
public WSGetSubFolderListResponse getSubFoldersList(String user_credential,
String partition_name, Locale locale, WSReference parentFolder,
WSFolderTypeEnum folderType)
throws CampaignServicesException
```

Здесь перечислены все подпапки в указанной папке.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*parentFolder*: ссылка папки, к которой запрашиваются все вложенные папки.

*folderType*: тип папки.

## Возвращает

Возвращает объект типа WSGetSubFolderListResponse.

## Ошибки

Генерирует CampaignServicesException, если ссылка папки недопустима.

## moveFolders

```
public WSMoveFolderResponse moveFolders(String userCredential,  
String partitionName, Locale requestedLocale, Long[] folderId,  
long parentFolder, long destinationFolder,  
WSFolderTypeEnum folderType)  
сбрасывает CampaignServicesException
```

Перемещает указанную папку в другую родительскую папку.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*folderId*: идентификатор папки для удаления.

*parentFolder*: идентификатор родительской папки.

*destinationFolder*: идентификатор папки назначения, в который перемещена указанная папка.

*folderType*: тип folder-Offer/session/campaign/segment.

## Возвращает

Возвращает объект типа WSMoveFolderResponse.

## Ошибки

Сбрасывает CampaignServicesException, если тип папки или идентификатор родительской папки недействительны или если идентификатор папки недействителен.

## deleteFolders

```
public WSDeleteFolderResponse deleteFolders(String userCredential,  
String partitionName, Locale requestedLocale, Long[] folderId,  
long parentFolder, boolean deleteChilds,  
WSFolderTypeEnum folderType)  
сбрасывает CampaignServicesException {
```

Удаляет указанные папки в системе наряду со всеми элементами в папке.

## Параметры

*userCredential*: идентификационные данные клиента.

*partitionName*: необязательное имя используемого раздела кампании.

*requestedLocale*: необязательная локаль для использования с этим требованием.

*folderId*: идентификатор папки для удаления.

*parentFolder*: идентификатор родительской папки.

*deleteChilds*: флаг указывает на необходимость удаления всех зависимостей папки. Если false, никакая зависимость не будет удалена.

*folderType*: тип folder-Offer/session/campaign/segment.

## **Возвращает**

Возвращает объект типа WSDepleteFolderResponse.

## **Ошибки**

Генерирует CampaignServicesException, если тип папки недопустим или если ID папки недопустим.

---

## Глава 5. Общие исключения API SOAP

API SOAP IBM Campaign может выдать следующие общие исключения. Локализованные сообщения всего исключения находятся в требуемой локали при наличии в IBM Campaign. Применяется обычная политика нейтрализации неисправности локали Java.

### RemoteException

Этот элемент применим только к интерфейсу SOAP.

Все вызовы SOAP к API могут сбросить RemoteException при наличии ошибки системного уровня, например, проблемы в слое обработки конверта SOAP (Axis), или нарушение ограничения, которое определено в WSDL веб-службы.

Заурядные зарегистрированные и незарегистрированные исключения API, такие как DataException, возвращаются в виде ошибочного состояния, а не в виде RemoteException.

Подробную информацию смотрите в разделе по интерфейсу SOAP.

### AuthenticationException

Пользователь не мог аутентифицироваться для указанного раздела Campaign. Проверьте пользовательскую роль, определенную в IBM Marketing Platform.

### Исключение авторизации

У пользователя нет полномочий для выполнения затребованной операции. Эта исключительная ситуация может быть выдана любым методом API, поэтому она не объявляется (не контролируется). Проверьте присвоенные для пользовательской роли полномочия в IBM Marketing Platform.

### Исключительное состояние данных

Критическое исключение произошло в базовом слое базы данных в Campaign (незарегистрированный).

Проверьте IBM Campaign журналы потоковой диаграммы и детектора для деталей.

### Исключение блокировки

Временное исключение, которое сбрасывается при попытке клиента обновить компонент, такой как потоковую диаграмму, в процессе ее редактирования другим пользователем. Обычно эту исключительную ситуацию можно исправить, подождя и повторив операцию. Алгоритм повторных попыток - ответственность клиента.

### InvalidComponentException

Была сделана попытка ссылки на недопустимый или неизвестный компонент (кампания, потоковая диаграмма, целевая ячейка). Метод исключения GetComponentReference () возвращает недействительную ссылку компонента.

## InvalidAttributeException

Исключение сбрасывается, когда клиент обеспечивает или ссылается на недействительный атрибут, как если бы он использовал неправильный тип данных или неразрешенный массив значений. Метод исключения `getAttributeName()` возвращает имя проблемного атрибута, а метод `getAttributeValue()` возвращает значение, в то время как метод `getComponentReference()` определяет компонент (или объемный индекс).

## AttributeExistsException

Генерируется, когда клиент пытается определить дубликат метаданных атрибута для компонента. Метод исключения `getAttributeName()` возвращает имя двойного атрибута; `getComponentReference()` определяет компонент (или объемный индекс).

## AttributeNotFoundException

Сбрасывается каждый раз при попытке клиента сослаться на неизвестный атрибут (кампанию, потоковую диаграмму, целевую ячейку и т.д.). Метод исключения `getAttributeName()` возвращает имя непревзойденного атрибута; `getComponentReference()` определяет компонент (или объемный индекс).

## CompositeException

`CompositeException` используется некоторыми API для сообщения вызывающей стороне о нескольких ошибках. С ним обычно не связана более чем одна причина; все причины будут оформлены в виде списка. Метод исключения `getCauseList()` возвращает этот список, который может быть проверен в дальнейшем для получения подробностей о каждой ошибке.

**Примечание:** Обычно API или завершается успешно, или откатывает свою работу перед выдачей составного исключения. Смотрите, например, объемные API электронной таблицы целевой ячейки, описанные в “Методы API SOAP: Ячейки назначения” на стр. 31.



---

## Прежде чем обращаться в службу технической поддержки IBM

Если вы столкнетесь с проблемой, которую невозможно разрешить при помощи документации, тот, кто назначен для вашей компании в качестве контактного лица для обращения в службу поддержки, может записать вызов в службу технической поддержки IBM. Используйте эти рекомендации, чтобы убедиться, что ваша проблема будет разрешена эффективно и успешно.

Если вы не отвечаете за поддержку в вашей компании, то за информацией обратитесь к своему администратору IBM.

**Примечание:** Служба технической поддержки не записывает и не создает сценарии API. За помощью по реализации наших предложений API обращайтесь в профессиональную службу IBM (IBM Professional Services).

### Какую информацию нужно собрать

Перед тем как обратиться в службу поддержки IBM, соберите следующие сведения:

- Краткое описание характера проблемы.
- Подробно: сообщения об ошибках, появляющиеся при возникновении проблемы.
- Подробное описание шагов по воспроизведению проблемы.
- Связанные файлы журналов, файлы сеансов, файлы конфигурации и файлы данных.
- Информацию о среде продукта и системы, которую можно получить, как рассказывается в разделе "Информация о системе".

### Информация о системе

При обращении в службу технической поддержки IBM вас могут попросить предоставить информацию о среде вашей системы.

Если проблема не мешает вам войти в систему, большая часть этой информации находится на странице О программе, где представлена информация об установленных приложениях IBM.

Доступ к странице О программе можно получить, выбрав **Справка > О программе**. Если страница О программе недоступна, смотрите файл `version.txt`, который находится в каталоге установки вашего приложения.

### Контактная информация для службы технической поддержки IBM

Как обратиться в службу технической поддержки IBM, можно узнать на веб-сайте технической поддержки продукта IBM: ([http://www.ibm.com/support/entry/portal/open\\_service\\_request](http://www.ibm.com/support/entry/portal/open_service_request)).

**Примечание:** Чтобы ввести запрос на поддержку, вы должны зарегистрироваться с учетной записью IBM. Эта учетная запись должна быть связана с вашим номером заказчика IBM. Чтобы подробнее узнать о связывании вашей учетной записи с вашим

номером заказчика IBM, смотрите **Support Resources>Entitled Software Support**  
(Ресурсы поддержки - Предоставляемая поддержка программ) в портале поддержки.

---

## Замечания

Эта информация относится к продуктам и услугам, предоставляемым в США.

IBM может предлагать описанные продукты, услуги и возможности не во всех странах. Сведения о продуктах и услугах, доступных в настоящее время в вашей стране, можно получить в местном представительстве IBM. Любые ссылки на продукты, программы или услуги IBM не означают явным или неявным образом, что можно использовать только продукты, программы или услуги IBM. Разрешается использовать любые функционально эквивалентные продукты, программы или услуги, если при этом не нарушаются права IBM на интеллектуальную собственность. Однако при этом ответственность за оценку и проверку работы всех продуктов, программ или услуг не-IBM возлагается на пользователя.

IBM может располагать патентами или рассматриваемыми заявками на патенты, относящимися к предмету данной публикации. Получение этого документа не означает предоставления каких-либо лицензий на эти патенты. Запросы по поводу лицензий следует направлять в письменной форме по адресу:

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

По поводу лицензий, связанных с использованием наборов двухбайтных символов (DBCS), обращайтесь в отдел интеллектуальной собственности IBM в вашей стране или направьте запрос в письменной форме по адресу:

Intellectual Property Licensing  
лицензированию интеллектуальной собственности  
IBM Japan, Ltd.  
19-21, Nihonbashi-Nakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

Следующий абзац не применяется в Великобритании или в любой другой стране, где подобные заявления противоречат местным законам: INTERNATIONAL BUSINESS MACHINES CORPORATION ПРЕДСТАВЛЯЕТ ДАННУЮ ПУБЛИКАЦИЮ "КАК ЕСТЬ" БЕЗО ВСЯКИХ ГАРАНТИЙ, КАК ЯВНЫХ, ТАК И ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВЫМИ, ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ СОБЛЮДЕНИЯ ЧЬИХ-ЛИБО АВТОРСКИХ ПРАВ, ВОЗМОЖНОСТИ КОММЕРЧЕСКОГО ИСПОЛЬЗОВАНИЯ ИЛИ ПРИГОДНОСТИ ДЛЯ КАКИХ-ЛИБО ЦЕЛЕЙ. В некоторых странах для определенных сделок подобные оговорки не допускаются, таким образом, это утверждение может не относиться к вам.

Данная информация может содержать технические неточности и типографские опечатки. Периодически в информацию вносятся изменения, они будут включены в новые издания этой публикации. IBM может в любое время без уведомления вносить изменения и усовершенствования в продукты и программы, описанные в этой публикации.

Любые ссылки в данной информации на сайты, не принадлежащие IBM, приводятся только для удобства и никоим образом не означают поддержки IBM этих сайтов. Материалы на таких веб-сайтах не являются составной частью материалов по данному продукту IBM, и вся ответственность за пользование такими веб-сайтами лежит на вас.

IBM может использовать или распространять информацию так, как сочтет нужным, без каких-либо обязательств с ее стороны.

Если обладателю лицензии на данную программу понадобятся сведения о возможности: (i) обмена данными между независимо разработанными программами и другими программами (включая данную) и (ii) совместного использования таких данных, он может обратиться по адресу:

IBM Corporation  
B1WA LKG1  
550 King Street  
Littleton, MA 01460-1250  
U.S.A.

Такая информация может быть предоставлена на определенных условиях (в некоторых случаях к таким условиям может относиться оплата).

Лицензированная программа, описанная в данном документе, и все прилагаемые к ней материалы предоставляются IBM на условиях IBM Customer Agreement (Соглашения IBM с заказчиком), Международного соглашения о лицензиях на программы IBM или эквивалентного соглашения.

Приведенные данные о производительности измерены в контролируемой среде. Таким образом, результаты, полученные в других операционных средах, могут существенно отличаться от них. Некоторые показатели измерены получены в системах разработки и нет никаких гарантий, что в общедоступных системах эти показатели будут теми же. Более того, некоторые результаты могут быть получены путем экстраполяции. Реальные результаты могут отличаться от них. Пользователи должны проверить данные для своих конкретных сред.

Информация о продуктах других фирм получена от поставщиков этих продуктов, из их опубликованных объявлений или из других общедоступных источников. IBM не проверяла эти продукты и не может подтвердить точность измерений, совместимость или прочие утверждения о продуктах других фирм. Вопросы о возможностях продуктов других фирм следует направлять поставщикам этих продуктов.

Все утверждения о будущих направлениях разработок или намерениях фирмы IBM могут быть пересмотрены или отменены без дополнительного объявления, и отражают исключительно предполагаемые цели фирмы.

Все указанные здесь цены IBM являются текущими рекомендуемыми ценами на продукты IBM, и они могут измениться безо всякого уведомления. Дилерские цены могут отличаться от них.

Эта информация содержит примеры данных и отчетов, иллюстрирующие типичные деловые операции. Чтобы эти примеры были правдоподобны, в них включены имена лиц, названия компаний и товаров. Все эти имена и названия вымышлены и любое их сходство с реальными именами и адресами полностью случайно.

**ЛИЦЕНЗИЯ НА ПЕРЕПЕЧАТКУ:**

Эта информация содержит примеры прикладных программ на языках программирования, иллюстрирующих приемы программирования для различных операционных платформ. Разрешается копировать, изменять и распространять эти примеры программ в любой форме без оплаты IBM для целей разработки, использования, сбыта или распространения прикладных программ, соответствующих интерфейсу прикладного программирования операционных платформ, для которых эти примеры программ написаны. Эти примеры не были всесторонне проверены во всех возможных условиях. Поэтому IBM не может гарантировать их надежность, пригодность и функционирование. Пробные программы предоставляются по принципу 'как есть', без какой-либо гарантии. IBM не несет ответственности ни за какой ущерб, вызванный использованием пробных программ.

Если вы просматриваете эту информацию на экране, фотографии и цветные иллюстрации могут быть не видны.

---

## Товарные знаки

IBM, логотип IBM и [ibm.com](http://ibm.com) - товарные знаки или зарегистрированные товарные знаки корпорации International Business Machines во многих юрисдикциях мира. Прочие названия продуктов и услуг могут быть товарными знаками IBM или других компаний. Текущий список товарных знаков IBM находится в веб на странице "Copyright and trademark information" (Информация об авторских правах и товарных знаках) по адресу: [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Замечания относительно политики конфиденциальности и положений об использовании

В программных продуктах IBM, включая программу как служебное решение ("Предложения относительно программ"), могут использоваться элементы cookie или другие технологии для сбора информации об использовании продукта, чтобы помочь улучшить опыт работы конечного пользователя, настроить взаимодействия с конечным пользователем или для других целей. Элемент cookie - это фрагмент данных, которые веб-сайт может отправить в ваш браузер и которые затем могут храниться на вашем компьютере в виде тега, идентифицирующего ваш компьютер. Во многих случаях никакой личной информации эти компоненты cookie не собирают. Если используемое вами Предложение относительно программ позволяет вам собирать личную информацию через компоненты cookie и аналогичные технологии, мы информируем вас ниже о соответствующих особенностях.

В зависимости от внедренных конфигураций данное Предложение относительно программ может использовать сеансы и хранимые компоненты cookie, которые собирают имя каждого пользователя и другую личную информацию для управления сеансами, усовершенствованной работы пользователей или других целей, касающихся отслеживания использования, или функциональных целей. Эти компоненты cookie можно отключить, но при их отключении также будут отключены функции, для поддержки которых они предназначены.

Различные законодательства регулируют сбор личной информации через компоненты cookies и аналогичные технологии. Если конфигурации, внедренные для этого Предложения относительно программ, обеспечивают вам, как заказчику, возможность собирать личную информацию от конечных пользователей через cookies и другие технологии, вы должны обратиться за местной юридической рекомендацией о том, существуют ли какие-либо законы, применимые к такому сбору данных, включая все требования относительно предоставления замечаний и согласований в тех случаях, где это применимо.

IBM требует, чтобы Клиенты (1) обеспечивали четкую и явную связь с терминами веб-сайта Заказчика относительно использования (например, политики конфиденциальности), включая связь со сбором и практикой использования данных IBM и Клиентом, (2) сообщали о том, что элементы cookie и явные элементы gif/веб-маяки помещались на компьютер посетителя компанией IBM от имени Клиента вместе с пояснением цели такой технологии, и (3) в той степени, в которой это требуется законом, получали согласие от посетителей веб-сайта перед помещением элементов cookie и явных элементов gif/веб-маяков Клиентом или компанией IBM от имени Клиента на устройства посетителя веб-сайта.

Более подробную информацию об использовании для этих целей различных технологий, включая компоненты cookie, смотрите в документе IBM Online Privacy Statement (Заявление об электронной конфиденциальности) по адресу: <http://www.ibm.com/privacy/details/us/en>, в разделе "Cookies, Web Beacons and Other Technologies" (Элементы cookie, веб-маяки и другие технологии).





Напечатано в Дании