

версия 10 выпуск 0
23 сентября 2016 г.

*Модуль интеграции IBM
Marketing Operations*

IBM

Примечание

Перед тем как использовать данный документ и продукт, описанный в нем, прочтите сведения под заголовком “Замечания” на стр. 33.

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

© Copyright IBM Corporation 2002, 2016.

Содержание

Глава 1. Что такое IBM Marketing Operations Integration Services? 1

Требования Marketing Operations Integration Services	2
Основы IBM Marketing Operations Integration Services	3
Установка Integration Services	5
Содержание комплекта для разработчика программ	6
Расположенные на хосте JavaDocs	6
Документация и справка по Marketing Operations	7

Глава 2. Веб-служба интеграции Marketing Operations 9

WSDL Marketing Operations Integration Services	9
executeProcedure	9
Типы данных интеграции веб-службы Marketing Operations	10

Глава 3. Процедура IBM Marketing Operations 15

Предположения	15
Параметры конфигурации	17
Дизайн	17
Жизненный цикл процедуры	18
Ключевые классы Java	19

Блокировка данных	20
Транзакции процедур	20
Взаимодействие с процедурами	21
Запись процедуры в журнал	21
Файл определений плагинов процедур	21

Глава 4. IBM Marketing Operations SOAP API 23

Содержание IBM Marketing Operations SOAP API	23
Интерфейсы API SOAP	23
Общие исключительные ситуации API SOAP	24
Дескрипторы API SOAP	24
SOAP API AttributeMap	26
Перечислимые типы данных API SOAP	27

Прежде чем обращаться в службу технической поддержки IBM 31

Замечания 33

Товарные знаки	35
Замечания относительно политики конфиденциальности и положений об использовании	35

Глава 1. Что такое IBM Marketing Operations Integration Services?

IBM® Marketing Operations Integration Services объединяет Marketing Operations Integration Webservice, процедуры API SOAP и триггеры, расширяя бизнес-возможности.

IBM Marketing Operations Integration Services состоит из следующих компонентов:

- Веб-служба интеграции **Marketing Operations**

Служба Integration Services обеспечивает клиентам Marketing Operations и службам IBM Professional Services возможность интеграции Marketing Operations с другими приложениями, работающими в их среде.

- **Процедуры Marketing Operations и API SOAP**

Пользовательские процедуры можно задать в Marketing Operations, чтобы расширять бизнес-логику Marketing Operations произвольными способами. После создания определений процедур эти процедуры могут стать объектами назначения для вызовов веб-службы Integration Services из других приложений. Также можно задать процедуры для отправки сообщений в другие приложения.

- **Триггеры Marketing Operations**

Триггеры можно связать с событиями и процедурами в Marketing Operations. Когда происходит одно такое событие, запускается связанный триггер.

API REST не используют службы интеграции Marketing Operations. Информацию об API REST смотрите в публикации IBM Marketing Operations: Руководство администратора.

Версии и совместимость с предшествующими версиями

Последующие версии служб интеграции обратно совместимы со всеми более ранними дополнительными выпусками и выпусками обслуживания, которые совместно используют номер основной версии. Однако IBM сохраняет за собой право нарушить совместимость с более ранней версией для основного выпуска (x.0), если возникнет производственная или техническая причина.

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

- Изменения интерпретации данных
- Изменения бизнес-логики (например, изменения функций метода службы)
- Изменения параметров метода и/или типов возвращаемых значений

Если какое-либо из следующих изменений внесено, то номер младшей версии API будет увеличен. Эти изменения совместимы с более ранней версией по определению.

- Добавлен новый метод
- Добавлен новый тип данных, и его использование ограничено новым методом
- Добавлен новый элемент к перечислимому типу
- Определена новая версия интерфейса с суффиксом версии

Аутентификация

Аутентификация не требуется; все клиенты связаны с известным пользователем IBM Marketing Operations, PlanAPIUser. Системный администратор конфигурирует средства защиты для этого особого пользователя, чтобы выполнить требования всех клиентов веб-служб.

Локаль

Единственной поддерживаемой локалью является локаль, сконфигурированная в данный момент для системного экземпляра IBM Marketing Operations. Считается, что все зависимые от локали данные, например, сообщения и валюта, представлены в системной локали.

Управление состояниями

У API и веб-службы *нет состояния*; никакая информация о клиенте не сохраняется реализацией службы через вызовы API. Эта функция обеспечивает эффективную реализацию службы и упрощает поддержку кластера.

Транзакции базы данных

Marketing Operations Integration Services не показывает клиенту транзакции базы данных, но используют эту информацию, если она включена в контекст выполнения. Если транзакция запущена, эффект всех вызовов API в рамках особой процедуры будет атомарным. Другими словами, неудачный вызов API оставит базу данных в том же состоянии, как если бы вызов API вообще никогда не производился. Другие пользователи Marketing Operations не увидят никаких изменений, пока процедура успешно не выполнит транзакцию.

Вызовы API, обновляющие базу данных, должны сначала получить блокировку изменений, чтобы не допустить изменения основных данных во время вызовов API другими пользователями. Другие пользователи не смогут обновить заблокированные компоненты, пока не завершится вызов API. И аналогично, прежде чем можно будет передать другой вызов API, следующий пользователь Marketing Operations или клиент API должен получить блокировку для данных.

Обработка событий

Операции с компонентами IBM Marketing Operations через API генерируют такие же события, какие бы были сгенерированы, если бы операцию выполнил пользователь Marketing Operations. Пользователи, подписавшиеся на те или иные уведомления, например, об изменениях состояния проекта, уведомляются об изменениях состояния, произошедших в результате вызовов API и действий пользователей.

Требования Marketing Operations Integration Services

У Marketing Operations Integration Services есть следующие требования.

Компонент Marketing Operations Integration Services должен:

- Связывать интеграцию систем нежесткой связью.
- Обеспечивать механизм для приложений клиента, влияющий на Marketing Operations через вызовы веб-службы.
- Обеспечивать механизм для приложений клиента, обеспечивающий уведомления об определенных событиях в Marketing Operations.

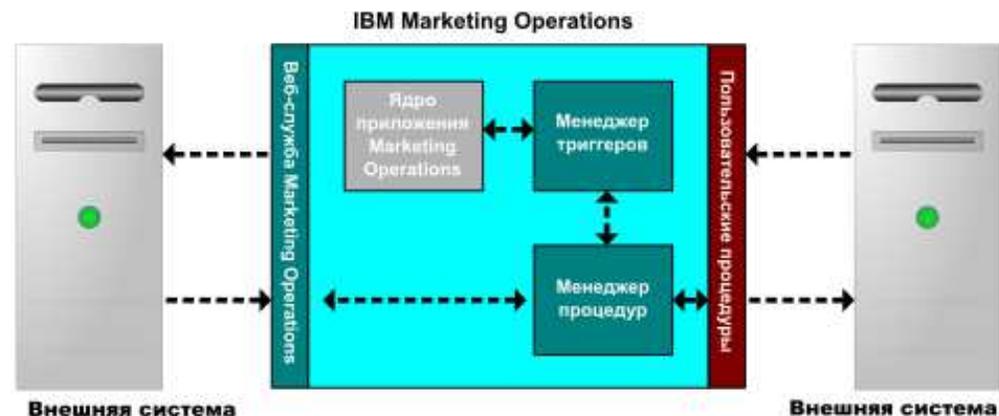
- Обеспечивать простую модель программирования, которую легко понять и использовать.
- Быть мощным средством восстановления после ошибки.
- Гарантировать целостность данных.
- Интегрироваться с существующими клиентами на основе графического пользовательского интерфейса Marketing Operations и сводить к минимуму влияние на них.
- Обеспечивать строго детализированный доступ к компонентам Marketing Operations, ограждая программистов от основных подробностей реализации.

Основы IBM Marketing Operations Integration Services

Вы используете IBM Marketing Operations Integration Services для создания пользовательских процедур. Эти процедуры позволяют инициировать внешние события, когда в Marketing Operations происходят определенные события. Эти процедуры можно использовать, чтобы запускать функции Marketing Operations из внешних систем или программ.

Интерфейс API взаимодействует с IBM Marketing Operations на программном уровне так же, как графические пользовательские интерфейсы взаимодействуют с Marketing Operations на уровне пользователей. Используя API, вы строите процедуры. С помощью этих процедур вы обеспечиваете взаимодействия между Marketing Operations и внешними системами. Marketing Operations Webservice - это объект контейнера для процедур, API и триггеров.

Ниже показана архитектура Marketing Operations Integration Services.

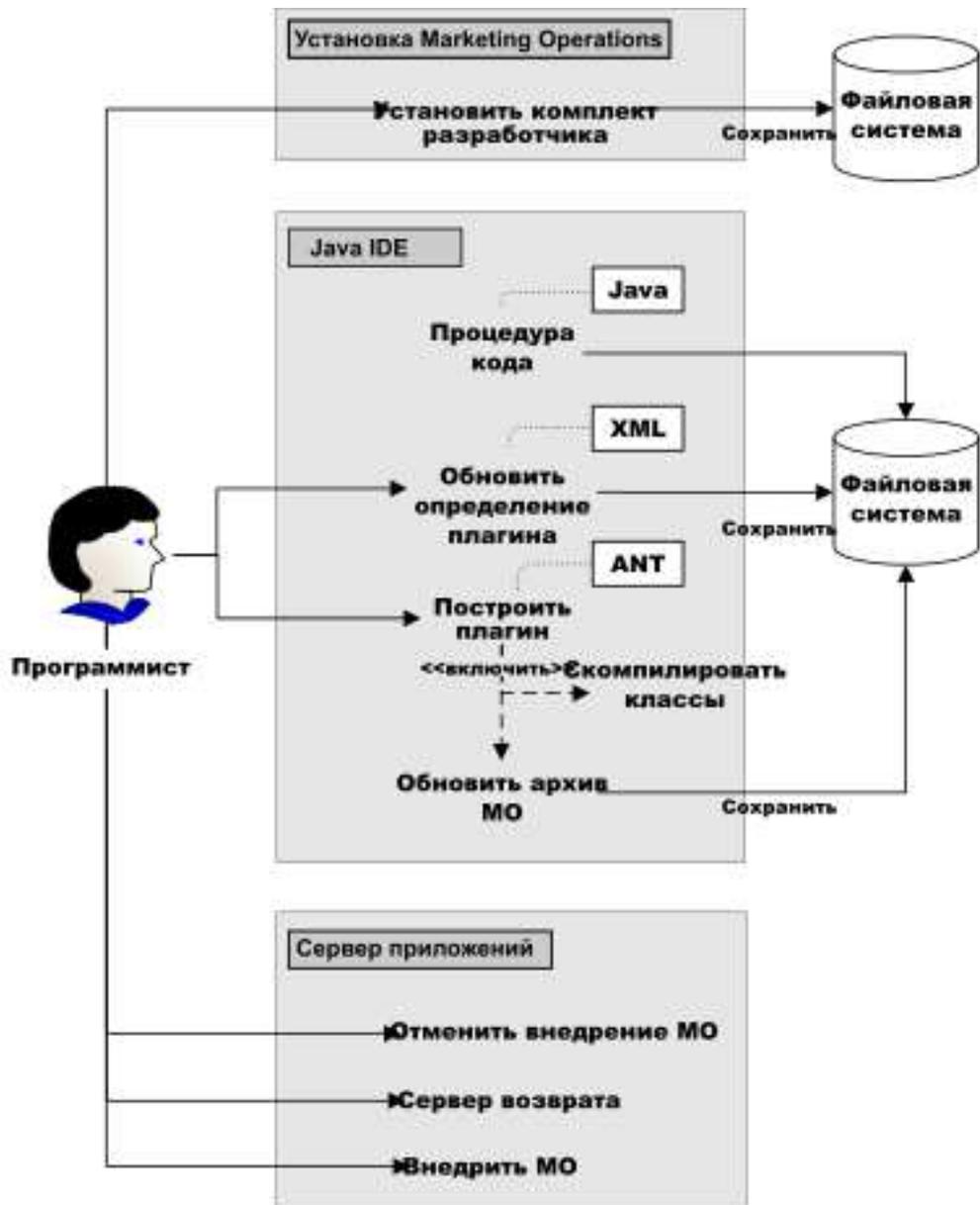


Ниже перечислены ключевые компоненты Integration Services:

- Менеджер процедур Marketing Operations: расширяет бизнес-логику путем взаимодействия с Marketing Operations через API.
- Менеджер триггеров Marketing Operations: связывает условие (например, изменение состояния маркетингового объекта) с действием (процедура, которую нужно выполнить, если будет выполнено условие для триггера).

Методы

Компоненты IBM Marketing Operations Integration Services используются для разработки пользовательских процедур, как показано на следующей схеме.



После установки комплекта разработчика вы выполняете следующие основные шаги:

1. Создайте код пользовательской процедуры.
2. Обновите определение модуля plug-in в файле определений XML.
3. Произведите сборку модуля plugin:
 - a. Скомпилируйте необходимые классы.
 - b. Если вы используете библиотеку стороннего поставщика, которой не было в архиве Marketing Operations, упакуйте библиотеку в файл `plan.war` и внедрите ее заново.
4. Перезапустите Marketing Operations. Изменения классов процедур применяются при перезапуске сервера приложений.

Примечание: Если вы измените файл `plan.war`, вы должны аннулировать внедрение и заново внедрить Marketing Operations с использованием нового файла `plan.war`. Аннулируйте внедрение и заново внедрите Marketing Operations, если вы

используете библиотеку стороннего поставщика, которой нет в архиве Marketing Operations, и если вы изменяете файл **plan.war**.

Основной пример по взаимодействию между IBM Marketing Operations и API

В приведенном ниже основном примере описано установление взаимодействий между API и Marketing Operations. Никакой полезной работы не выполняется; представлен законченный цикл между Marketing Operations и Integration Services.

В этом примере используются части примеров процедур, включенных в комплект разработчика Marketing Operations Integration Services. В частности, вы сможете найти код, ссылка на который есть в следующих файлах:

- PlanClientFacade.java
- PlanWSNOOPTestCase.java

Метод noop - это вызов веб-службы в Marketing Operations. Он задан в классе PlanClientFacade и передает нулевые значения в массив.

```
public ProcedureResponse noop(String jobId)
    throws RemoteException, ServiceException {
    NameValueArrays parameters =
        new NameValueArrays(null, null, null, null, null, null, null, null);
    return _serviceBinding.executeProcedure("uapNOOPProcedure", jobId, parameters);
}
```

Процедура testExecuteProcedure вызывает метод noop из PlanClientFacade, чтобы установить замкнутый цикл с приложением Marketing Operations.

```
public void testExecuteProcedure() throws Exception {
    // Тайм-аут через минуту
    int timeout = 60000;
    PlanClientFacade clientFacade = new PlanClientFacade(urlWebService, timeout);
    System.out.println("noop с параметрами/без параметров");
    long startTime = new Date().getTime();
    ProcedureResponse response = clientFacade.noop("junit-jobid");
    long duration = new Date().getTime() - startTime;

    // нулевое или положительное состояние => успех
    System.out.println("Состояние: " + response.getStatus());
    System.out.println("Длительность: " + duration + " мс");
    assertTrue(response.getStatus() >= 0);
    System.out.println("Готово.");
}
```

Сведения о NameValueArrays, ProcedureResponse и других перечисленных методах и типах данных смотрите в *Marketing Operations Integration Module* и в документах JavaDocs.

Установка Integration Services

Модуль Integration Services IBM Marketing Operations - отдельно оплачиваемый компонент. При покупке модуля Integration Services необходимо установить его.

1. Скачайте программы установки IBM Marketing Operations Integration Services.
2. Программы установки IBM Marketing Software обнаруживают модуль Integration Services.
3. Программа установки задает свойства конфигурации, выбирая **Marketing Operations | umoConfiguration | integrationServices | enableIntegrationServices**. Вы

можете настроить установку, изменяя параметры конфигурации.
Дополнительную информацию смотрите в разделе “Параметры конфигурации”
на стр. 17.

Содержание комплекта для разработчика программ

Комплект разработчика программ содержит документацию, содержащую все классы и интерфейсы `publicapi` и пример кода.

Для API SOAP все компоненты Marketing Operations Integration Services устанавливаются в папке `devkits`.

Пример программного кода установлен в следующих папках.

- В папке **build** содержатся сценарии, позволяющие строить и внедрять пользовательские процедуры.
- В папке **Classes** содержатся скомпилированные классы процедур.
Пользователи должны внедрить скомпилированные классы пользовательских процедур в пути, заданном параметром конфигурации **integrationProcedureClasspathURL**. После этого менеджер процедур IBM Marketing Operations загрузит их, как указано в файле конфигурации `procedure-plugins.xml`.
- В папке **lib** содержатся необходимые библиотеки для разработки и компилирования пользовательских процедур.
- Папка **src** содержит исходные файлы для пользовательских процедур.
Пользователи могут разместить здесь пользовательские процедуры, которые должны запускаться как триггеры или веб-службы. Пользовательские процедуры поддерживаются только API SOAP.
 - В папке **src/procedure** содержится файл конфигурации `procedure-plugins.xml`.
Для каждой пользовательской процедуры, которая выполняется как триггер на основе события или через внешнюю веб-службу, в этом файле должна быть запись. Записи должны содержать полный путь класса процедуры и необходимых параметров инициализации.
 - В папке **src/procedure** также содержатся некоторые примеры процедур, включенные в IBM Marketing Operations. Эти процедуры можно использовать, чтобы получить представление о пользовательских процедурах и их разработать.
Поместите пользовательские процедуры в каталог **src** в новой структуре папок, например, `com/<моя_компания>/<мой_пакет>`. Не помещайте пользовательские процедуры в папку примеров процедур.
 - В папке **src/soap** содержатся примеры клиентов веб-служб, разработанные с использованием Java. Используйте эти примеры как отправную точку для разработки клиентов на основе веб-служб для Integration Services. Эта папка также содержит двоичные сценарии для запуска примеров клиентов из командной строки.

Расположенные на хосте JavaDocs

Конкретную информацию об общедоступных методах API смотрите в описании класса `iPlanAPI` в файлах документации по API JavaDocs.

Доступ к этим файлам можно получить следующими способами:

- Используя файлы в каталоге `<IBM_IMS>/<домашний_каталог_MarketingOperations>/devkits/integration/javadocs` для API SOAP на сервере, который является хостом для Marketing Operations.

- Войдя в Marketing Operations и выбрав **Справка > Документация по продукту** на любой странице, а затем скачав файл IBM <версия>PublicAPI.zip для API SOAP.

Документация и справка по Marketing Operations

Разные сотрудники в вашей организации используют IBM Marketing Operations для выполнения различных задач. Информация о Marketing Operations предоставляется в виде набора руководств, каждое из которых предназначено для использования членами бригад с определенными целями и наборами квалификаций.

В следующей таблице описано, какую информацию содержит каждое Руководство.

Таблица 1. Руководства в наборе документации по Marketing Operations.

В следующей таблице, состоящей из трех столбцов, приводится описание задач в одном столбце, имена руководств - во втором столбце и аудитория - в третьем столбце.

Если вы	Смотрите	Для кого предназначен этот документ
<ul style="list-style-type: none"> • Планируете проекты и управляете ими • Задаете задачи рабочих потоков, этапы и персонал • Отслеживаете расходы на проекты • Получаете результаты проверки и утверждений содержимого • Создаете отчеты 	<i>IBM Marketing Operations: Руководство пользователя</i>	<ul style="list-style-type: none"> • Менеджеры проектов • Креативные дизайнеры • Маркетинговые менеджеры, занимающиеся прямой почтовой рассылкой
<ul style="list-style-type: none"> • Разрабатываете шаблоны, формы, атрибуты и метрики • Настраиваете пользовательский интерфейс • Задаете уровни доступа пользователей и безопасность • Реализуете дополнительные функции • Конфигурируете и настраиваете Marketing Operations 	<i>IBM Marketing Operations: Руководство администратора</i>	<ul style="list-style-type: none"> • Менеджеры проектов • Администраторы информационных технологий • Консультанты по реализации
<ul style="list-style-type: none"> • Создаете маркетинговые кампании • Планируете предложения • Реализуете интеграцию между Marketing Operations и Campaign • Реализуете интеграцию между Marketing Operations и IBM Digital Recommendations 	<i>IBM Marketing Operations и IBM : Руководство по реализации</i>	<ul style="list-style-type: none"> • Менеджеры проектов • Маркетинговые исполнительные специалисты • Прямые маркетинговые менеджеры
<ul style="list-style-type: none"> • Узнаете о новых системных функциях • Изучаете известные проблемы и обходные пути 	<i>IBM Marketing Operations: Замечания по выпуску</i>	Все, кто используют Marketing Operations
<ul style="list-style-type: none"> • Устанавливаете Marketing Operations • Конфигурируете Marketing Operations • Производите обновление до новой версии Marketing Operations 	<i>IBM Marketing Operations Руководство по установке</i>	<ul style="list-style-type: none"> • Консультанты по реализации программ • Администраторы информационных технологий • Администраторы базы данных

Таблица 1. Руководства в наборе документации по Marketing Operations (продолжение).

В следующей таблице, состоящей из трех столбцов, приводится описание задач в одном столбце, имена руководств - во втором столбце и аудитория - в третьем столбце.

Если вы	Смотрите	Для кого предназначен этот документ
Создаете пользовательские процедуры для интеграции Marketing Operations с другими приложениями	<p><i>IBM Marketing Operations: Модуль интеграции</i> и документы API JavaDoc будут доступны, если вы выберете Справка > Документация по продукту в Marketing Operations, а затем загрузите файл IBM<версия>PublicAPI.zip для SOAP API и файл IBM<версия>PublicAPI-RestClient.zip для REST API.</p>	<ul style="list-style-type: none"> • Администраторы информационных технологий • Администраторы баз данных • Консультанты по реализации
Узнаете о структуре базы данных Marketing Operations	<p><i>IBM Marketing Operations Системная схема</i></p>	Администраторы баз данных
Нуждаетесь в дополнительной информации во время работы	<ul style="list-style-type: none"> • Получите справку и произведите поиск или обзор публикаций <i>Marketing Operations: Руководство пользователя, Руководство администратора</i> или <i>Marketing Operations: Руководство по установке</i>: Выберите Справка > Справка по этой странице. • Получите доступ ко всем руководствам Marketing Operations: выберите Справка > Документация по продукту • Получите доступ к руководствам по всем продуктам IBM Marketing Software: Выберите Справка > Вся документация по комплекту IBM Marketing Software 	Все, кто используют Marketing Operations

Глава 2. Веб-служба интеграции Marketing Operations

Веб-служба обеспечивает представление клиента для Marketing Operations Integration Services, что является частью внедрения сервера IBM Marketing Operations. Служба используется параллельно с веб-пользователями Marketing Operations.

Веб-служба поддерживает один вызов API, executeProcedure.

Клиент выполняет непосредственный вызов этой веб-службы.

WSDL Marketing Operations Integration Services

Язык определений веб-служб (Web Services Definition Language, WSDL) разработан вручную, и это - последнее слово в определении веб-служб.

Axis

В этой версии веб-службы используется Axis2 1.5.2 для генерирования на стороне сервера классов, которые обеспечивают реализацию веб-службы из файла WSDL. Пользователи могут использовать любую версию Axis или технику, не являющуюся техникой Axis, чтобы создать реализацию на стороне клиента для интеграции с API из заданного WSDL.

Версия протокола

Версия протокола явным образом связана с WSDL следующим образом:

- Как часть имени WSDL, например, PlanIntegrationService1.0.wsdl
- Как часть целевого пространства имен WSDL, например, xmlns:tns="http://webservices.unica.com /MktOps/services/PlanIntegrationServices1.0?wsdl"

WSDL

К IBM Marketing Operations Integration Services прилагается один файл WSDL: PlanIntegrationServices1.0.wsdl. WSDL поставляется в каталоге integration/examples/soap/plan. В примере сценария сборки этот файл используется в качестве примера, чтобы сгенерировать соответствующие стаб-файлы на стороне клиента для соединения с веб-службой.

executeProcedure

executeProcedure - это вызов в API, поддерживаемый веб-службой.

Синтаксис

```
executeProcedure(string key, string jobid, NameValueArrays paramArray)
```

Возвращает

```
int: status  
Message[]: messages
```

Описание

Этот метод вызывает указанную процедуру с необязательным массивом параметров. Вызов выполняется синхронно; то есть, он блокирует клиент и возвращает результат после завершения.

Параметры

Таблица 2. Параметры `executeProcedure`

Имя	Описание
key	Уникальный ключ процедуры, которую нужно выполнить. Если с ключом не связано никакой процедуры, будет возвращена ошибка <code>RemoteException</code> .
jobid	Необязательная строка, указывающая задание, связанное с выполнением этой процедуры. Эта строка - промежуточный элемент, но ее можно использовать, чтобы связать задания клиентов с выполнением определенной процедуры.
paramArray	Массив параметров, которые нужно передать процедуре. Состояние ошибки и сообщение будут возвращены, если один или более параметров окажутся недопустимыми (например, у них будет не тот тип или неправильное значение). Клиент сам определяет параметры, их тип и число значений, которые требуются процедуре.

Возвращаемые параметры

Таблица 3. Возвращаемые параметры `executeProcedure`

Имя	Описание
status	Целочисленный код <ul style="list-style-type: none">0 указывает, что процедура выполнена успешноЦелое число указывает на ошибку Процедуры могут использовать состояние, чтобы указать разные уровни ошибок.
messages	Массив из нуля или более структур данных сообщений. Если значение status равно 0, этот массив не содержит сообщений об ошибках (ERROR), но может содержать информационные сообщения (INFORMATION) и сообщения с предупреждениями (WARNING). Если значение status не равно нулю, сообщения могут содержать любую комбинацию сообщений об ошибках, информационных сообщений и сообщений с предупреждениями (ERROR, INFORMATION и WARNING).

Типы данных интеграции веб-службы Marketing Operations

Типы данных, используемые веб-службой, не зависят ни от какой отдельной привязки службы или реализации программирования.

Используется следующая форма записи:

- <тип>: <определение типа> задает простой тип данных. Например:
Handle: string
- <тип>: [<определение типа>] задает сложный тип данных или структуру данных.
- <тип>: { <определение типа> } задает сложный тип данных или структуру данных.

Элементы сложного типа и параметры API могут использовать эти типы для объявления массивов. Например:

```
Handle [] handles
```

Тип handles - это массив типов Handle.

Типы примитивов

Типы примитивов ограничены типами, заданными в таблице, которая приводится ниже, чтобы упростить поддержку привязок SOAP 1.1. Все типы можно объявить как массивы, например, **String []**. По своей сути, такие двоичные типы данных, как **long**, можно представить в виде строки по привязке к протоколу (например, SOAP). Однако такое представление никак не влияет на семантику типа, разрешенных значений и т.п., которые видит клиент.

Таблица 4. Типы примитивов

Тип API	Описание	Тип SOAP	Тип Java™
Boolean	Логическое значение: true или false	xsd:Boolean	логическое значение
dateTime	Значение даты и времени	xsd:datetime	Дата
decimal	Условная точность, десятичное значение	xsd:decimal	java.math.BigDecimal
double	Двойная точность, со знаком, десятичное значение	xsd:double	double
int	32-разрядное, целочисленное значение со знаком	xsd:int	int
integer	Условная точность, со знаком, целочисленное значение	xsd:integer	java.math.BigInteger
long	64-разрядное, целочисленное значение со знаком	xsd:long	long
string	Строка символов Unicode	xsd:string	java.lang.String

MessageTypeEnum

```
MessageTypeEnum: { INFORMATION, WARNING, ERROR }
```

MessageTypeEnum - это нумерованный тип, который задает все возможные типы сообщений.

- INFORMATION: информационное сообщение
- WARNING: сообщение с предупреждением
- ERROR: сообщение об ошибке

Сообщение

```
Message: [MessageTypeEnum type, string code, string localizedText, string logDetail]
```

Message - это структура данных, которая определяет результат вызова API веб-службы. В ней есть необязательные поля для нелокализованного кода,

локализованного текста и сведения журнала. В настоящее время весь локализованный текст использует локаль, заданную для экземпляра сервера IBM Marketing Operations.

Таблица 5. Параметры Message

Параметр	Описание
type	MessageTypeEnum, который задает тип сообщения.
code	Необязательный код в строчном формате для сообщения.
localizedText	Необязательная текстовая строка для связывания с сообщением.
logDetail	Необязательное сообщение трассировки стека.

NameValue

NameValue: [string name, int sequence]

NameValue - это базовый сложный тип, который задает пару имя-значение. Он также задает необязательную последовательность, используемую службой, чтобы построить массивы значения по мере необходимости (последовательности основаны на нуле).

Все NameValue с одним и тем же именем, но разными порядковыми номерами, преобразуются в массив значений и связываются с общим именем.

Размер массива определяется максимальным порядковым номером; у не заданных элементов массива будут нулевые значения. Порядковые номера массива должны быть уникальными. Значение и его тип определяются расширенным типом.

Таблица 6. Параметры NameValue

Параметр	Описание
name	Строка, которая задает имя типа NameValue.
sequence	Целое число на основе нуля, которое задает порядковый номер для подразумеваемого значения NameValue.

Расширенные типы NameValue заданы для каждого типа примитива следующим образом:

Таблица 7. Расширенные типы NameValue

Расширенный тип	Описание
BigDecimalNameValue: NameValue [decimal value]	Тип NameValue, значением которого является десятичное число с условной точностью.
BigIntegerNameValue: NameValue [integer value]	Тип NameValue, значением которого является целое число условного размера.
BooleanNameValue: NameValue [Boolean value]	Тип NameValue, значение которого является логическим.
CurrencyNameValue: NameValue [string locale, decimal value]	Тип NameValue, подходящий для представления валюты в той или иной локали. Локаль - это код языка ISO, то есть, двухбуквенный код в нижнем регистре, заданный в ISO-639. В настоящее время локаль должна соответствовать локали, заданной в экземпляре сервера IBM Marketing Operations.

Таблица 7. Расширенные типы NameValue (продолжение)

Расширенный тип	Описание
DateNameValue: NameValue [datetime value]	Тип NameValue, значение которого является дата.
DecimalNameValue: NameValue [double value]	Тип NameValue, значением которого является десятичное число с двойной точностью.
IntegerNameValue: NameValue [long value]	Тип NameValue, значением которого является 64-разрядное целое число.
String NameValue: NameValue [string value]	Тип NameValue, значение которого является строкой.

И, наконец, массив расширенных типов NameValue, заданный для использования, когда вам нужно задать набор NameValue с разными типами.

```

NameValueArrays: [
  BooleanNameValue[]    booleanValues,
  StringNameValue[]     stringValues,
  IntegerNameValue[]    integerValues,
  BigIntegerNameValue[] bigIntegerValues,
  DecimalNameValue[]    decimalValues,
  BigDecimalNameValue[] bigDecimalValues,
  DateNameValue[]       dateNameValues,
  CurrencyNameValue[]   currencyValues
]

```

Глава 3. Процедура IBM Marketing Operations

"Процедура" - это пользовательский или стандартный класс Java, размещенный в компоненте IBM Marketing Operations, который выполняет некую часть работы. Процедуры обеспечивают клиентам и службам IBM Professional Services возможность расширять бизнес-логику Marketing Operations произвольными способами.

Процедуры соответствуют простой модели программирования с четко заданным API, чтобы влиять на компоненты, управляемые программой Marketing Operations. Процедуры "обнаруживаются" с использованием простого механизма поиска и основанного на XML файле определения. Marketing Operations запускает процедуры в соответствии с требованиями "клиентов". Например, в ответ на требование интеграции (входное) или на срабатывание триггера (входное или исходящее).

Процедуры выполняются синхронно с их клиентом; результаты становятся доступны непосредственно клиенту и через сохраняемый механизм аудита. Выполнение процедуры также может заставить другие события и триггеры сработать в Marketing Operations.

Процедуры следует писать на языке Java.

Предположения

Классы реализации процедур упакованы в отдельное дерево классов или файл JAR и сделаны доступными для продукта IBM Marketing Operations через путь URL.

Реализация процедуры

Менеджер выполнения процедур использует независимый загрузчик классов для загрузки этих классов по мере необходимости. По умолчанию, Marketing Operations производит поиск в следующем каталоге:

```
<домашний_каталог_MarketingOperations>/devkits/integration/examples/classes
```

Чтобы изменить это значение по умолчанию, задайте параметр **integrationProcedureClasspathURL** в окне **Параметры > Конфигурация > Marketing Operations > umoConfiguration > integrationServices**.

Имя класса реализации процедур соответствует принятому порядку присвоения имен Java, чтобы не допустить конфликтов пакета с "unica" и классами от других поставщиков. В частности, клиенты не должны помещать процедуры в дерево пакетов "com.unica" или "com.unicasorp".

Реализация процедуры закодирована в версии среды выполнения Java, используемой компонентом IBM Marketing Operations на сервере приложений (как минимум, JRE 1.5.10).

Класс реализации процедуры загружается политикой загрузки классов, которая обычно используется компонентом IBM Marketing Operations (как правило, **родительский-последний**). Сервер приложений может обеспечивать средства разработки и опции для перезагрузки классов, которые бы применялись к процедурам Marketing Operations, но это не требуется.

Библиотеки

В IBM Marketing Operations есть ряд библиотек с открытым исходным кодом и библиотек сторонних поставщиков; серверы приложений также используют разные версии этих библиотек.

В общем случае этот список изменяется от одного выпуска до другого. Поддерживаются следующие сторонние библиотеки:

- Ant 1.6.5 (ant.jar)
- Axis2 1.5.2 и зависимые компоненты
 - axiom-api-1.2.9.jar
 - axiom-impl-1.2.9.jar
 - axis2-adb-codegen-1.5.2.jar
 - axis2-codegen-1.5.2.jar
 - axis2-adb-1.5.2.jar
 - axis2-kernel-1.5.2.jar
 - axis2-transport-http-1.5.2.jar
 - axis2-transport-local-1.5.2.jar
 - commons-codec.jar
 - commons-httpclient-3.1.jar
 - commons-logging.jar
 - httpcore-4.0.jar
 - neethi-2.0.4.jar
 - geronimo-stax-api_1.0_spec-1.0.1.jar
 - jaxrpc.jar
 - xlxpScanner.jar
 - xlxpScannerUtils.jar
 - xlxpWASParsers.jar
 - wsdl4j-1.6.2.jar
 - XmlSchema-1.4.3.jar
- JavaMail 1.4.3 (activation.jar, mail.jar)
- JUnit 4.4 (junit-4.4.jar)
- IBM Marketing Operations API (affinium_plan.jar)
- IBM Marketing Platform API (unica-common.jar)

Если процедура или вторичные классы, которые импортирует процедура, действительно использует такие пакеты, их использование должно точно согласовываться с пакетами, предоставляемыми продуктом Marketing Operations или сервером приложений. В этом случае потребуется переработать код процедуры, если последняя версия Marketing Operations обновится или отбросит библиотеку.

Процедуры и потоки

Процедура должна быть безопасна с точки зрения потоков относительно своего собственного состояния; то есть, метод ее выполнения не может зависеть от изменений внутреннего состояния от вызова к вызову. Процедура не может сама создавать потоки.

Параметры конфигурации

При установке модуля интеграции Marketing Operations программа установки задает три свойства конфигурации. Свойства конфигурации можно изменить, чтобы настроить поведение модуля интеграции.

Чтобы найти свойства конфигурации для модуля интеграции, выберите **Marketing Operations | umoConfiguration | integrationServices**.

- Свойство конфигурации **enableIntegrationServices** позволяет включать и выключить модуль служб интеграции.
- Параметр **integrationProcedureDefinitionPath** содержит полный путь XML-файла пользовательского определения процедуры.
Значение по умолчанию:
<домашний_каталог_IBM_IMS><домашний_каталог_MarketingOperations>/devkits/integration/ examples/src/procedure/procedure-plugins.xml/.
- Параметр **integrationProcedureClasspathURL** содержит пути классов для пользовательских процедур.
Значение по умолчанию: file:///<IBM_IMS_Home><MarketingOperations_Home>/devkits/ integration/examples/classes/.

Примечание: Символ '/' в конце пути `integrationProcedureClasspathURL` обязателен, чтобы правильно загрузились классы процедуры.

Дизайн

Класс реализации процедуры использует API IBM Marketing Operations, чтобы читать и обновлять компоненты Marketing Operations, запускать службы и т.д. Другие пакеты Java можно использовать для выполнения других задач.

В вашей разработке уделите внимание тому, чтобы создать одну единицу работы, выполняющуюся отдельно. В идеальном случае процедура выполняет некую серию задач, которые можно запланировать в асинхронном режиме для выполнения в более позднее время. Такая модель интеграции типа "выстрелил и забыл" обеспечивает наименьшую нагрузку на обе системы.

Примечание: В последующих выпусках Marketing Operations будут поддерживаться только документированные классы и методы. Считайте все остальные классы и методы в Marketing Operations запрещенными.

После того как вы закодируете и скомпилируете классы реализации процедур, вы делаете их доступными для Marketing Operations. Сценарии сборки, поставляемые вместе со службами интеграции Marketing Operations, помещают скомпилированные процедуры в распоряжение по умолчанию. Заключительный шаг разработки заключается в обновлении файла пользовательского определения модуля `plug-in` процедуры, используемого компонентом Marketing Operations для обнаружения пользовательских процедур.

Процедура должна реализовывать интерфейс **`com.unica.publicapi.plan.plugin.procedure.IProcedure`**, и у нее должен быть конструктор без параметров (обычно модель JavaBeans). Кодирование и компиляция каждой процедуры выполняется в Java IDE по выбору клиента, например, в Eclipse, Borland JBuilder либо Idea. Пример кода прилагается к IBM Marketing Operations как инструментарий разработчика и находится в следующем расположении:

Жизненный цикл процедуры

Каждая процедура проходит через полный жизненный цикл.

Жизненный цикл среды выполнения процедуры включает в себя следующие шаги:

1. Обнаружение и инициализация
2. Выбор (необязательно)
3. Выполнение
4. Уничтожение

Обнаружение и инициализация

Продукт IBM Marketing Operations должен знать о всех стандартных и пользовательских процедурах, доступных для отдельного экземпляра установки. Этот процесс называется обнаружением.

Примечание: Стандартные процедуры (процедуры, заданные командой инженеров Marketing Operations) известны, и не нужно выполнять никаких действий, чтобы их обнаружить.

Пользовательские процедуры заданы в файле определения плагина (plug-in) процедуры. Менеджер плагинов Marketing Operations читает этот файл во время инициализации. Для каждой найденной процедуры менеджер плагинов выполняет следующие шаги:

1. Создает экземпляр процедуры; переводит ее в состояние INSTANTIATED.
2. Создает запись аудита процедуры.
3. Если создать экземпляр процедуры не удастся, будет вызван ее метод **initialize()** со всеми параметрами инициализации, находящимися в ее файле описания модуля плагин (plug-in). Если этот метод сгенерирует исключительную ситуацию, состояние будет записано в журнал, а процедура будет отброшена. В противном случае процедура перейдет в состояние INITIALIZED. Теперь она готова к выполнению.
4. Создает запись аудита процедуры.
5. Если процедуру не удастся инициализировать, будет вызван ее метод **getKey()**, чтобы определить, какой ключ используется клиентами для ссылки на процедуру. Этот ключ связывается с экземпляром и сохраняется для последующего поиска.

Выбор

Время от времени IBM Marketing Operations может предоставлять пользователям список доступных процедур, например, чтобы позволить администраторам задать триггер. Marketing Operations представляет такой список только после инициализации процедуры, используя методы **getDisplayname()** и **getDescription()** процедуры.

Выполнение

В тот же момент после инициализации процедуры IBM Marketing Operations получает требование выполнения процедуры. Это требование может создаваться параллельно с другими процедурами (или с той же процедурой), выполняющимися в других потоках.

Во время выполнения менеджер выполнения процедур делает следующее:

1. Запускает транзакцию базы данных.
2. Задает для процедуры состояние EXECUTING.
3. Создает запись аудита процедуры.
4. Вызывает метод **execute()** процедуры с контекстом выполнения и всеми параметрами запуска, заданными клиентом. Реализация метода использует API Marketing Operations (если это потребуется), получая блокировки изменений и передавая данные по контексту выполнения. Если метод выполнения сгенерирует исключительную ситуацию, менеджер выполнения пометит транзакцию, как подлежащую откату.
5. Принимает или откатывает транзакцию в соответствии с результатами выполнения; задает для процедуры состояние EXECUTED.
6. Снимает все устаревшие блокировки изменений.
7. Создает запись аудита процедуры.

Примечание: Метод **execute()** не предназначен для изменения данных экземпляра процедуры.

Уничтожение

Когда IBM Marketing Operations завершает работу, менеджер плагинов (plug-in) процедуры проходит через все загруженные процедуры. Для каждой найденной процедуры он выполняет следующие шаги:

1. Вызывает метод `destroy()` процедуры, чтобы разрешить очистку процедуры до уничтожения экземпляра.
2. Изменяет состояние процедуры на FINALIZED (ее нельзя запустить).
3. Создает запись аудита процедуры.
4. Уничтожает экземпляр процедуры.

Ключевые классы Java

Поставляемый комплект разработчика интеграции содержит набор документов Javadoc для общедоступных API IBM Marketing Operations и классов поддержки.

Ниже перечислены наиболее важные классы Java:

- `IProcedure` (`com.unica.publicapi.plan.plugin.procedure.IProcedure`): интерфейс, который должны реализовывать все процедуры. Процедуры проходят четко заданный жизненный цикл и получают доступ к API Marketing Operations для выполнения работы.
- `ITriggerProcedure` (`com.unica.publicapi.plan.plugin.procedure.ITriggerProcedure`): интерфейс, который должны реализовывать все триггерные процедуры (интерфейс маркера).
- `IExecutionContext` (`com.unica.publicapi.plan.plugin.procedure.IExecutionContext`): интерфейс непрозрачного объекта контекста, передаваемый процедуре менеджером выполнения. У этого объекта есть общедоступные методы для записи в журнал и изменения управления блокировками. Процедура также передает этот объект всем вызовам `PlanAPI`.
- `IPlanAPI` (`com.unica.publicapi.plan.api.IPlanAPI`): интерфейс для API Marketing Operations. Контекст выполнения обеспечивает метод **`getPlanAPI()`** для получения надлежащей реализации.

Блокировка данных

IBM Marketing Operations использует схему блокировки изменений, при которой право доступа для обновления экземпляров компонентов предоставляется только одному пользователю одновременно. В случае пользователя графического пользовательского интерфейса эта блокировка выполняется на уровне визуальной вкладки. В некоторых случаях данные блокируются для подмножества экземпляра, например, для вкладки сводки проекта. В других случаях данные блокируются для многих экземпляров, например, для вкладки рабочего потока. После того как пользователь получит блокировку, доступ для всех остальных пользователей ограничивается только чтением связанных данных.

Чтобы убедиться, что изменения, внесенные процедурой в экземпляр компонента или в группу экземпляров, не будут случайно перезаписаны другим пользователем, процедура должна получить соответствующие блокировки, прежде чем сможет обновить данные компонента. Для блокировки данных используется объект контекста выполнения, передаваемый методу **execute()** процедуры.

Прежде чем процедура обновит какие-либо данные, она должна вызвать метод **acquireLock()** контекста для каждой нужной ей блокировки. Например, если процедура собирается обновить проект и связанный рабочий поток, процедура должна получить блокировки для них обоих.

Если у другого пользователя уже есть блокировка, метод **acquireLock()** сразу же сгенерирует исключительную ситуацию **LockInUseException**. Чтобы свести к минимуму число конфликтов, процедура должна снимать блокировку, как только обновит объект.

Когда метод выполнения возвращает управление, менеджер по выполнению автоматически снимает все просроченные блокировки. В любом случае блокировки удерживаются только в течение срока жизни транзакции базы данных. То есть, блокировка истекает, если наступает тайм-аут транзакции, связанной с базой данных.

Примечание: Блокировки изменений - это не то же самое, что транзакции базы данных.

Транзакции процедур

Менеджер выполнения процедур автоматически заключает выполнение процедуры в транзакцию базы данных, принимая или откатывая ее назад на основе результата выполнения процедуры.

Заключение выполнения процедуры и транзакции базы данных в оболочку гарантирует, что обновления базы данных IBM Marketing Operations не будут видны другим пользователям, пока не будут приняты. При этом обновления становятся атомарными.

Тому, кто написал процедуру, все еще могут потребоваться необходимые блокировки изменения, чтобы другие пользователи не смогли записать изменения в базу данных до завершения выполнения процедуры.

Взаимодействие с процедурами

Метод **execute()** процедуры возвращает целочисленный код состояния в таблицу аудита процедур IBM Marketing Operations. Метод **execute()** процедуры может также возвращать ноль или более сообщений в таблицу аудита процедуры, где они регистрируются и сохраняются.

Клиент также может передавать информацию о состоянии каким-либо другим способом.

Запись процедуры в журнал

У IBM Marketing Operations есть отдельный файл журнала для процедур:
<домашний_каталог_MarketingOperations>\logs\system.log.

Менеджер выполнения процедуры записывает жизненный цикл каждой процедуры и создает записи аудита.

- **logInfo ()**: записывает информационное сообщение в журнал процедуры.
- **logWarning ()**: записывает предупреждающее сообщение в журнал процедуры.
- **logError ()**: записывает сообщение об ошибке в журнал процедуры.
- **logException()**: Записывает в журнал процедуры дампы трассировки стека для исключительной ситуации.

Файл определений плагинов процедур

В файле определений плагинов процедур задан класс реализации, метаданные и другая информация о пользовательских процедурах, которые будут размещены в IBM Marketing Operations.

По умолчанию определение модуля plug-in процедуры, как предполагается, находится в следующем пути:

```
<домашний_каталог_MarketingOperations>/devkits/integration/examples/src/  
procedures/procedure-plugins.xml
```

Этот файл представляет собой XML-документ, содержащий следующую информацию:

Процедуры: Список из нуля или большего числа элементов **Procedure**.

Процедура: Элемент, который задает процедуру. Каждая процедура содержит следующие элементы:

- **key** (необязательно): Строка, которая задает ключ поиска для процедуры. Этот ключ должен быть уникальным для всех стандартных (предоставленных IBM) и пользовательских процедур, находящихся на определенном экземпляре Marketing Operations. Если он не задан, по умолчанию используется полная версия элемента **className**. Имена, начинающиеся со строки "uap", зарезервированы для использования компонентом IBM Marketing Operations.
- **className** (обязательно): Полное имя пакета класса процедуры. Этот класс должен реализовывать класс `IProcedure (com.unica.public.plan.plugin.procedure.IProcedure)`.
- **initParameters** (необязательно): Список из нуля или более элементов `initParameter`.
initParameter (необязательно): Параметр, который нужно передать методу `initialize()` процедуры. Этот элемент содержит вложенные элементы имени, типа и значения параметра.

- name: Строка, которая задает имя параметра
- type: Необязательное имя класса класса оболочки Java, которое задает тип значения параметра. Это должен быть один из следующих типов:
 - java.lang.String (по умолчанию)
 - java.lang.Integer
 - java.lang.Double
 - java.lang.Calendar
 - java.lang.Boolean
- value: Строчная форма значения атрибута в соответствии с его типом

Глава 4. IBM Marketing Operations SOAP API

API SOAP IBM Marketing Operations - это внешняя сторона, обеспечивающая клиенту представление работающего экземпляра Marketing Operations.

Пользователям показано только подмножество возможностей Marketing Operations. API параллельно используются веб-пользователями Marketing Operations и требованиями и триггерами Marketing Operations Integration Services WebService SOAP. API поддерживает следующие типы операций:

- Создание и удаление компонентов
- Обнаружение (по типу компонентов, значениям атрибутов и другим значениям)
- Инспекция компонентов (через их атрибуты, специализированные ссылки и другие значения)
- Изменение компонентов

Примечание: API Marketing Operations предназначены только для использования администратором.

Содержание IBM Marketing Operations SOAP API

Пакет `com.unica.publicapi.plan.api` доставляет API SOAP IBM Marketing Operations.

Этот пакет предлагает интерфейсы и исключительные ситуации, а также содержит следующие типы классов:

- Нумерованные типы данных.
- Дескрипторы для идентификации экземпляров объектов и компонентов.
- Карта Java, `AttributeMap`.

Полную документацию по API, включая все методы и возможные значения, можно получить, выбрав **Справка > Документация по продукту** в экземпляре Marketing Operations, а затем скачав файл `IBM <версия>PublicAPI.zip`.

Интерфейсы API SOAP

Интерфейс прикладного программирования (application programming interface, API) SOAP IBM Marketing Operations включает в себя **IPlanAPI** и **IExecutionContext**.

API SOAP Marketing Operations содержит следующие интерфейсы:

IPlanAPI

Задаёт общедоступный API для Marketing Operations. Обеспечивает методы создания, обнаружения и изменения объектов, включая папки, проекты, программы, задачи рабочего потока и членов команд.

В системах, для которых включена необязательная интеграция с IBM Campaign, также обеспечивает методы создания, обнаружения и изменения предложений.

IExecutionContext

Задаёт триггеры и блокировки, выполняющие методы в API.

Методы API

Конкретную информацию об общедоступных методах API смотрите в описании класса `iPlanAPI` в файлах документации по API JavaDocs.

Доступ к этим файлам можно получить, войдя в Marketing Operations и выбрав **Справка > Документация по продукту** на любой странице, а затем загрузив файл `<версия>PublicAPI.zip`.

Общие исключительные ситуации API SOAP

Общие исключительные ситуации, генерируемые API SOAP, включают в себя `NotFoundException`, `AuthorizationException`, `DataException`, `InvalidExecutionContextException` и `NotLockedException`.

В следующем списке показано, почему могут происходить эти общие исключительные ситуации:

- `<тип объектов>NotFoundException`: Система не может вернуть указанную позицию или объект.
- `AuthorizationException`: Пользователь, связанный с контекстом выполнения, не авторизован для выполнения затребованной операции. Эту исключительную ситуацию может сгенерировать любой метод API, поэтому она не объявляется.
- `DataException`: Исключительная ситуация произошла в базовом слое базы данных в IBM Marketing Operations. Подробную информацию смотрите в журнале SQL.
- `InvalidExecutionContextException`: Возникла проблема с контекстом выполнения, переданным методу API (например, метод не инициализирован должным образом). Эту исключительную ситуацию может сгенерировать любой API, поэтому она не объявляется.
- `NotLockedException`: Предпринята попытка обновить данные компонента, не получив сначала необходимую блокировку. Смотрите метод `acquireLock()` интерфейса `IExecutionContext`.

Дескрипторы API SOAP

Дескриптор - это специальный объект URL, ссылающийся на отдельный экземпляр объекта в экземпляре IBM Marketing Operations. Дескрипторы включают в себя тип компонента, внутренний идентификатор данных и базовый URL экземпляра.

Дескрипторы, используемые или сгенерированные API, можно преобразовать в полный URL. Полученный URL можно использовать разными способами. Используя URL, можно открыть представление компонента в графическом пользовательском интерфейсе Marketing Operations, можно отправить его в сообщениях электронной почты или использовать в другой процедуре как параметр.

Дескрипторы действительны только для отдельного экземпляра службы или кластеризованного экземпляра Marketing Operations, и действуют они в течение срока жизни внедренной службы. В результате этого дескрипторы можно сохранить в файле для дальнейшего использования, но их нельзя использовать для получения доступа к компонентам в другом экземпляре Marketing Operations. Это ограничение также касается экземпляров на том же физическом хост-сервере. Однако в Marketing Operations нет механизма отображения разных базовых URL в текущий экземпляр, чтобы учесть перемещение экземпляра на другой сервер (например, если функционирование оборудования нарушено).

Дескрипторы не зависят от клиентов. Например, триггер может передать дескриптор процедуре, которая будет использовать его как параметр в вызове SOAP для

сторонней системы. После этого сторонняя система может сгенерировать требование SOAP обратно в Marketing Operations, чтобы запустить процедуру, обновляющую атрибут.

У членов класса Handle есть фабричные методы создания дескрипторов из разных типов URL. Примеры приведены ниже.

Утверждение

`http://mymachine:7001/plan/affiniumplan.jsp?cat=approvaldetail&approvalid=101`

Актив

`http://mymachine:7001/plan/affiniumplan.jsp?cat=asset&assetMode=VIEW_ASSET&assetid=101`

Папка активов

`http://mymachine:7001/plan/affiniumplan.jsp?cat=folder&id=101`

Библиотека активов

`http://mymachine:7001/plan/affiniumplan.jsp?cat=library&id=101`

Вложение

`http://mymachine:7001/plan/affiniumplan.jsp?cat=attachmentview&attachid=101&parentObjectId=101&parentObjectType=project`

Финансовый счет

`http://mymachine:7001/plan/affiniumplan.jsp?cat=accountdetails&accountid=101`

Папка

`http://mymachine:7001/plan/affiniumplan.jsp?cat=grouping_folder&folderid=1234`

Инвойс

`http://mymachine:7001/plan/affiniumplan.jsp?cat=invoicedetails&invoiceid=134`

Позиция строки инвойса

`http://mymachine:7001/plan/affiniumplan.jsp?cat=invoicedetails&invoiceid=134&line_item_id=101`

Объект маркетинга

`http://mymachine:7001/plan/affiniumplan.jsp?cat=componenttabs&componentid=creatives&componentinstid=1234`

Сетка объектов маркетинга

`http://mymachine:7001/plan/affiniumplan.jsp?cat=componenttabs&componentid=creatives&componentinstid=1234&gridid=grid`

Строка сетки объектов маркетинга

`http://mymachine:7001/plan/affiniumplan.jsp?cat=componenttabs&componentid=creatives&componentinstid=1234&gridid=grid&gridrowid=101`

Команда плана

`http://mymachine:7001/plan/affiniumplan.jsp?cat=teamdetails&func=edit&teamid=100001`

Пользователь плана

`http://mymachine:7001/plan/affiniumplan.jsp?cat=adminuserpermissions&func=edit&userId=101`

Программа

`http://mymachine:7001/plan/affiniumplan.jsp?cat=programtabs&programid=125`

Сетка программы

`http://mymachine:7001/plan/affiniumplan.jsp?cat=programtabs&programid=1234&gridid=grid`

Строка сетки программы

`http://mymachine:7001/plan/affiniuplan.jsp?cat=programtabs&programid=1234&gridid=grid&gridrowid=101`

Проект

`http://mymachine:7001/plan/affiniuplan.jsp?cat=projecttabs&projectid=1234`

Сетка проекта

`http://mymachine:7001/plan/affiniuplan.jsp?cat=projecttabs&projectid=1234&gridid=grid`

Строка сетки проекта

`http://mymachine:7001/plan/affiniuplan.jsp?cat=projecttabs&projectid=1234&gridid=grid&gridrowid=101`

Позиция строки проекта

`http://mymachine:7001/plan/affiniuplan.jsp?cat=projecttabs&projectid=1234&projectlineitemid=123&projectlineitemisversionfinal=false`

Этап рабочего потока

`http://mymachine:7001/plan/affiniuplan.jsp?cat=projectworkflow&projectid=1234&taskid=5678`

Задача рабочего потока

`http://mymachine:7001/plan/affiniuplan.jsp?cat=projectworkflow&projectid=1234&taskid=5678`

SOAP API AttributeMap

Класс AttributeMap - это карта Java, содержащая только атрибуты. Атрибут *<Имя>* - это ключ записи карты, а массив *<значений>* атрибута (множественное число) - это значение записи карты.

Класс AttributeMap содержит следующие поля:

- *<Имя>*: программное имя атрибута. Это имя служит ключом для доступа к атрибуту в экземпляре компонента, в котором оно встречается.

Примечание: *<Имя>* не обязательно является выводимым на экран именем, которое увидит пользователь в графическом пользовательском интерфейсе. Для компонентов, созданных из шаблонов (например, проектов или задач рабочего потока), имя атрибута задано определением элемента шаблона. Имя атрибута должно быть уникальным. Для других компонентов имя атрибута обычно получают программным способом из экземпляра компонента на стороне сервера (например, путем самодиагностики Java).

Примечание: В соответствии с принятыми соглашениями, пользовательские атрибуты содержат имя формы, в которой задана доступная для изменения версия: *<имя_формы>.<имя_атрибута>*.

- *Значения*: Массив объектов Java, содержащий ноль или более значений атрибутов. Тип всех значений должен быть одним и тем же и должен согласовываться с типом атрибута, заданным в Marketing Operations. Поддерживается только следующая оболочка Java и типы Marketing Operations:
 - AssetLibraryStateEnum: Нумерованное значение типа AssetLibraryStateEnum.
 - AssetStateEnum: Нумерованное значение типа AssetStateEnum.
 - AttachmentTypeEnum: Нумерованное значение типа AttachmentTypeEnum.
 - AttributeMap: карта, содержащая атрибуты.
 - BudgetPeriodEnum: Нумерованное значение типа BudgetPeriodEnum.
 - BudgetTypeEnum: Нумерованное значение типа BudgetTypeEnum.

- `Handle`: Ссылка на экземпляр компонента, строку сетки, атрибут, и т.д.
- `InvoiceStateEnum`: Нумерованное значение типа `InvoiceStateEnum`.
- `java.io.File`: представление файла.
- `java.lang.Boolean`: Логическое значение, `True` или `False`
- `java.lang.Double`: Десятичное числовое значение с двойной точностью.
- `java.lang.Float`: Десятичное числовое значение с одинарной точностью.
- `java.lang.Integer`: 32-разрядное целочисленное значение.
- `java.lang.Long`: 64-разрядное целочисленное значение.
- `java.lang.Object`: Универсальный объект Java
- `java.lang.String`: Строка, содержащая ноль или большее число символов Unicode.
- `java.math.BigDecimal`: Десятичное числовое значение со знаком с условной точностью. Подходит для валюты; интерпретация значения зависит от локали валюты у клиента.
- `java.math.BigInteger`: Целочисленное значение с условной точностью.
- `java.net.URL`: Объект Universal Resource Locator (URL).
- `java.util.ArrayList`: Список объектов.
- `java.util.Calendar`: Значение даты и времени для определенной локали.
- `java.util.Date`: Значение даты и времени. Этот тип устарел. Вместо него используйте `java.util.Calendar` или `java.util.GregorianCalendar`.

Примечание: Чтобы реализовать дату, пользователи могут выбрать либо `java.util.Calendar`, либо `java.util.GregorianCalendar`.

- `java.util.GregorianCalendar`: `GregorianCalendar` - это конкретный подкласс `java.util.Calendar`, который обеспечивает стандартную календарную систему, используемую в большей части мира.
- `MonthEnum`: Нумерованное значение типа `MonthEnum`.
- `ProjectStateEnum`: Нумерованное значение типа `ProjectStateEnum`.
- `QuarterEnum`: Нумерованное значение типа `QuarterEnum`.
- `TaskStateEnum`: Нумерованное значение типа `TaskStateEnum`.
- `WeekEnum`: Нумерованное значение типа `WeekEnum`.

Метаданные атрибута (например, переведенное выводимое на экран имя и описание) заданы шаблоном, который связан с атрибутом и его родительским экземпляром объекта. Атрибуты обеспечивают простой, но растяжимый механизм, позволяющий показать как обязательные, так и необязательные атрибуты экземпляров объектов, например, имя проекта, код и начальную дату.

Перечислимые типы данных API SOAP

API SOAP IBM Marketing Operations поддерживает следующие перечисленные типы данных и значения.

ApprovalMethodEnum

`ApprovalMethodEnum` задает допустимые методы утверждения. Возможные значения:

- SEQUENTIAL
- SIMULTANEOUS

ApprovalStateEnum

`ApprovalStateEnum` задает допустимые состояния утверждения. Возможные значения:

- CANCELLED
- COMPLETED
- IN_PROGRESS
- NOT_STATED
- ON_HOLD

AssetLibraryStateEnum

AssetLibraryStateEnum задает допустимые состояния библиотеки активов. Возможные значения:

- DISABLED
- ENABLED

AssetStateEnum

AssetStateEnum задает допустимые состояния активов. Возможные значения:

- ARCHIVE
- DRAFT
- FINALIZE
- LOCK

AttachmentTypeEnum

AttachmentTypeEnum задает допустимые типы вложений. Возможные значения:

- ASSET
- FILE
- URL

BudgetPeriodEnum

BudgetPeriodEnum задает возможные периоды бюджета. Возможные значения:

- ALL
- MONTHLY
- QUARTERLY
- WEEKLY
- YEARLY

BudgetTypeEnum

BudgetTypeEnum задает допустимые типы бюджета. Возможные значения:

- ACTUAL
- ALLOCATED
- COMMITTED
- FORECAST
- TOTAL

ComponentTypeEnum

ComponentTypeEnum указывает допустимые типы компонентов Marketing Operations. Возможные значения:

- APPROVAL
- ASSET
- ASSET_FOLDER
- ASSET_LIBRARY
- ATTACHMENT
- FINANCIAL_ACCOUNT

- GROUPING_FOLDER
- INVOICE
- MARKETING_OBJECT
- PLAN_TEAM
- PLAN_USER
- PROGRAM
- PROJECT
- PROJECT_REQUEST
- TASK
-

InvoiceStateEnum

InvoiceStateEnum задает допустимые состояния инвойсов. Возможные значения:

- CANCELLED
- DRAFT
- PAID
- PAYABLE

MonthEnum

MonthEnum задает допустимые значения месяцев.

OfferStateEnum

OfferStateEnum задает допустимые состояния предложений. Возможные значения:

- STATE_OFFER_DRAFT
- STATE_OFFER_PUBLISHED
- STATE_OFFER_RETIRED

ProjectCopyTypeEnum

ProjectCopyTypeEnum задает допустимые методы копирования проекта. Возможные значения:

- COPY_USING_PROJECT_METRICS
- COPY_USING_TEMPLATES_METRICS

ProjectParticipantLevelEnum

ProjectParticipantLevelEnum указывает роли, которые могут быть у пользователей в проекте. Возможные значения:

- OWNER
- PARTICIPANT
- REQUESTER

ProjectStateEnum

ProjectStateEnum задает допустимые состояния проектов и требований. Возможные значения:

- ACCEPTED
- CANCELLED
- COMPLETED
- DRAFT
- IN_PROGRESS
- IN_RECONCILIATION
- LATE: проект не запущен к его запланированной начальной дате.

- NOT_STARTED
- ON_HOLD
- OVERDUE: проект не выполнен до его запланированной даты окончания.
- RETURNED
- SUBMITTED

Дополнительную информацию о состоянии проектов и задач смотрите в публикации *IBM Marketing Operations: Руководство пользователя*.

QuarterEnum

QuarterEnum задает допустимые значения кварталов: Q1, Q2, Q3 и Q4.

TaskStateEnum

TaskStateEnum задает допустимые состояния задач рабочего потока.

Возможные значения:

- ACTIVE
- DISABLED
- FINISHED
- PENDING
- SKIPPED

WeekEnum

WeekEnum задает допустимые значения недель в году, от WEEK_1 до WEEK_53.

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 надо собрать

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

- Краткое описание характера вашей проблемы.
- Подробно: сообщения об ошибках, появляющиеся при возникновении проблемы.
- Подробно: шаги, позволяющие воспроизвести проблему.
- 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 сообщить информацию о вашей среде.

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

О том, 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).

Примечание: Чтобы ввести требование о поддержке, нужно войти в систему с использованием учетной записи 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 раздел **Support Resources > Entitled Software Support** (Ресурсы поддержки > Предоставляемая поддержка программ) в портале поддержки.

Замечания

Представленные сведения относятся 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:

Законодательство по защите и
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 - 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.

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