

версия 10 выпуск 0
Июнь 2016 г.

*Руководство по
использованию набора для
разработки плагинов
валидации IBM Campaign*

IBM

Примечание

Перед тем как использовать данный документ и продукт, описанный в нем, прочтите сведения под заголовком “Замечания” на стр. 17.

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, 2016.

Содержание

Глава 1. Обзор комплекта инструментальных средств разработчика плагинов проверки (PDK) IBM 1

Содержимое проверки PDK	1
Два способа использования проверки API	2
Выполните сборку плагина класса Java, который загружен в приложение	3
Вызов приложения для обработки проверки	3
Предложение против проверки кампании	3
Демонстрационные блоки проверки допустимости, включенные в PDK проверки	4
Стеновая программа для Проверки PDK	4
Сценарии сборки для Проверки PDK	5

Глава 2. Разработка плагинов проверки для Campaign 7

Установка среды для использования Validation PDK.	7
Создание блока проверки допустимости	8
Конфигурирование Campaign для использования плагина проверки	8
validationClass	9

validationClasspath	9
validatorConfigString.	10
Тестирование конфигурации блока проверки допустимости.	10
Создание блока проверки допустимости	11
Пример сценария проверки: предотвращение редактирования кампании.	11

Глава 3. Вызов приложения для обработки проверки 13

Конфигурирование Campaign для использования демонстрационного выполняемого плагина	13
Ожидаемый исполнимый интерфейс использования	13

Прежде чем обращаться в службу технической поддержки IBM 15

Замечания 17

Товарные знаки	19
Замечания относительно политики конфиденциальности и положений об использовании	19

Глава 1. Обзор комплекта инструментальных средств разработчика плагинов проверки (PDK) IBM

Используйте Комплект инструментальных средств разработчика плагинов проверки (PDK) IBM® для разработки пользовательской логики проверки для применения в IBM Campaign.

Можно создать модули plugin для выполнения пользовательского алгоритма проверки для кампаний, предложений или и того, и другого.

Некоторые возможные применения логики проверки представлены ниже:

- Проверка расширенных (пользовательских) атрибутов
- Предоставление служб аутентификации, которые выходят за рамки IBM Marketing Platform (например, проверка того, которым пользователям разрешается редактировать те или иные расширенные атрибуты).

PDK проверки является подклассом более универсальной среды разработки плагинов, которая поставляется с IBM Campaign.

PDK проверки содержит справочную информацию Javadoc для API плагинов и примера кода. Для просмотра документации откройте в веб-браузере следующий файл:

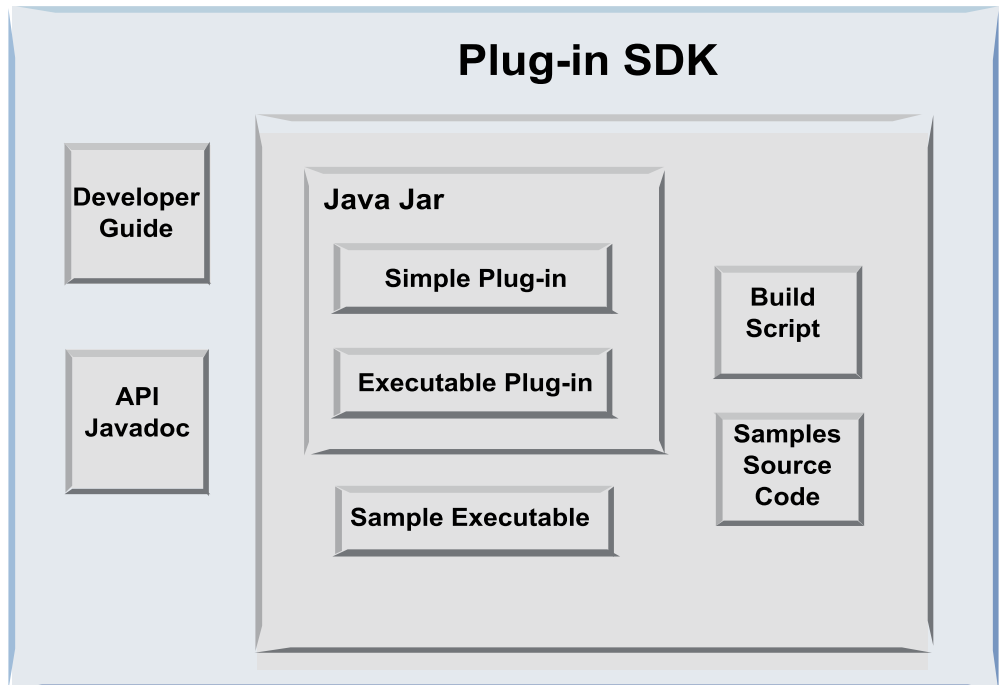
```
C:\IBM\IMS\Campaign_Home\devkits\validation\javadoc\index.html
```

Например:

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

Содержимое проверки PDK

Проверка PDK содержит компоненты для разработки плагинов Java™ или выполняемых модулей командной строки для включения пользовательской проверки в IBM Campaign. PDK содержит зарегистрированные, сборные примеры того, как использовать PDK.



В следующей таблице описываются все компоненты.

Таблица 1. Компоненты проверки PDK

Компонент	Описание
Руководство разработчика	Документ в формате PDF под названием <i>IBM Campaign Руководство по проверке PDK</i> .
API Javadoc	Справочная информация для API модуля plug-in.
JAR-файл Java	Демонстрационный файл JAR, содержащий демонстрационные модули plugin. Файл JAR содержит: <ul style="list-style-type: none"> • Простой модуль plug-in: пример самодостаточного класса блока проверки. • Выполняемый плагин: блок проверки допустимости в качестве примера, выполняющий определяемую пользователем выполняемую команду командной строки для выполнения проверки.
Пример выполняемого файла	Выполняемый модуль командной строки, который может использоваться с выполняемым плагином на ОС UNIX.
Сценарий сборки	Сценарий Ant для встраивания включенного исходного кода в применимые плагины блока проверки допустимости.
Демонстрационный исходный код	Исходный код Java для простого блока проверки допустимости и выполняемого блока проверки допустимости.

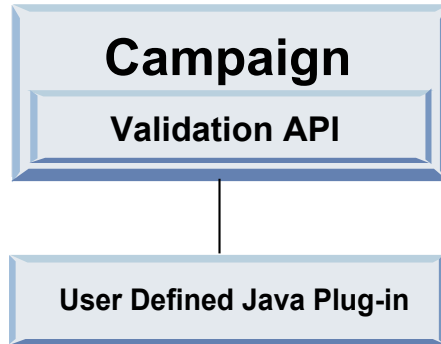
Два способа использования проверки API

Есть два способа использовать API Проверки.

- Использование плагина для создания загруженного в приложение плагина класса Java.
- Используйте один из включенных модулей plugin для обращений к исполняемому приложению для обработки проверки.

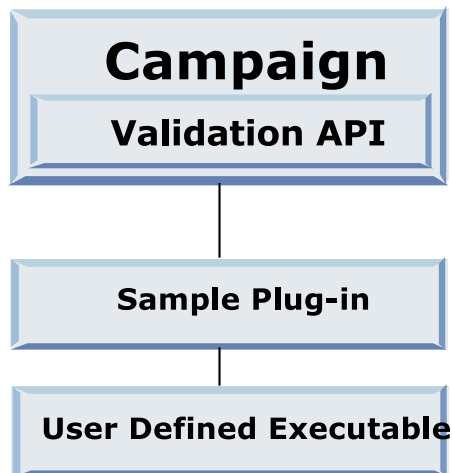
Выполните сборку плагина класса Java, который загружен в приложение

Проверка PDK обеспечивает интерфейсы, классы системы поддержки и инструментов разработчика для разработки таких классов.



Вызов приложения для обработки проверки

Используйте один из включенных плагинов проверки PDK для передачи ответного сигнала на выполняемое приложение для обработки проверки.



Выполняемый модуль может быть записан на любом языке, но должен находиться на IBM Campaign сервере и выполняться на таком сервере. Вызывающий выполняемый модуль плагин отправляет XML-файл, содержащий информацию для проверки; например, пользователь, редактирующий объект и значения до и после для всех стандартных и расширенных атрибутов такого объекта. IBM Campaign ожидает информацию о результатах в виде обратного XML-файла.

Предложение против проверки кампании

Выполненный с помощью Проверки PDK плагин Campaign может выполнить пользовательскую логику проверки для кампаний, предложений, или кампаний и предложений.

Проверка PDK может проверять предложения и кампании. Если плагин проверки определен, он автоматически получает название IBM Campaign при каждом

сохранении объекта предложения или кампании. IBM Campaign устанавливает флаг при вызове метода проверки плагина. IBM Campaign передает следующие флаги:

- `ValidationInputData.CAMPAIGN_VALIDATION` при добавлении или изменении кампании
или
- `ValidationInputData.OFFER_VALIDATION` при добавлении или редактировании предложения.

Можно потом использовать эти флаги для построения правил проверки, применяющихся к предложениям и кампаниям.

Демонстрационные блоки проверки допустимости, включенные в PDK проверки

В PDK проверки Campaign включены два демонстрационных блока проверки допустимости: `SimpleCampaignValidator` и `ExecutableCampaignValidator`.

- `SimpleCampaignValidator` - независимый плагин, показывающий, как выполнять такие действия как пользовательская авторизация и проверка допустимых имен кампании. Его можно найти по следующему адресу:

```
devkits\validation\src\com\unica\campaign\core\validation\
samples\SimpleCampaignValidator.java
```

Рекомендуем вам создать копию класса перед его редактированием, чтобы в случае необходимости у вас оставалась оригинальная версия.

- `ExecutableCampaignValidator` представляет собой плагин Java, который обращается к исполняемому приложению для выполнения проверки. Исходный код для `ExecutableCampaignValidator` включен в тот же каталог, что и `SimpleCampaignValidator`:

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

Однако реальная цель этого примера - использоваться в качестве исполняемой программы командной строки для проверки. Этот файл расположен по следующему адресу:

```
devkits/validation/src/com/unica/campaign/core/validation/
samples/validate.sh
```

Этот файл - демонстрационная циклическая исполняемая программа, показывающая общие типы работы проверки.

Стендовая программа для Проверки PDK

Способность тестирования кода доступа без перетаскивания в IBM Campaign ускоряет процесс Разработчика плагинов.

Клиенты, использующие экстремальное программирование и другие быстрые и гибкие методы, прибегают к активному использованию модульного тестирования. Проверка PDK поддерживает эти методы путем выполнения плагина стендовой программы за пределами Campaign.

Используйте стендовую программу следующим образом:

1. Измените случай модульного тестирования для отображения логики проверки в плагине.
2. Выполните сценарий сборки:
 - Чтобы создать плагин без каких-либо испытаний устройств, выполните команду "ant jar".

- Чтобы создать плагин и выполнить модульное тестирование, выполните сценарии сборки с помощью "ant run-test".

Сценарии сборки для Проверки PDK

Сценарии сборки в Проверке PDK компилируют все классы в каталоге и помещают их в файл JAR, подходящий для использования в IBM Campaign.

Предоставленный сценарий сборки использует следующий каталог:

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

Глава 2. Разработка плагинов проверки для Campaign

Плагин представляет собой класс Java, который загружается во время запуска и вызывается каждый раз при проверке кампании или предложения.

Проверка происходит каждый раз, когда пользователь сохраняет кампанию. Собственные плагины Java можно создать с помощью инструментов в Проверка PDK. PDK содержит исходный код для демонстрационных плагинов и файла Ant (Apache Ant представляет собой инструмент сборки на основе Java), которые используются для компиляции плагинов.

В следующих шагах приводится описание по установке вашей среды для разработки плагинов, а также пошаговое описание создания вашего собственного плагина.

1. “Установка среды для использования Validation PDK”
2. “Создание блока проверки допустимости” на стр. 8
3. “Конфигурирование Campaign для использования плагина проверки” на стр. 8
4. “Тестирование конфигурации блока проверки допустимости” на стр. 10
5. “Создание блока проверки допустимости” на стр. 11

Установка среды для использования Validation PDK

Чтобы использовать Validation PDK с Campaign, вы должны изменить путь и настроить переменную среды JAVA_HOME.

Validation PDK может быть установлен на любом компьютере, но плагины, которые вы создаете с его помощью, должны находиться на компьютере, на котором выполняется IBM Campaign. Мы рекомендуем установить PDK на компьютере, где вы тестируете свои модули plugin.

PDK требует от вас наличия на вашем компьютере комплекта разработчика Apache Ant и Sun Java для создания плагинов Java. Для обеспечения совместимости используйте пакеты Ant и JDK, которые поставляются с вашим сервером приложений.

Чтобы настроить среду для использования Validation PDK:

1. Добавьте в ваш путь папку, в которой содержится исполняемая программа Ant. Приводятся два примера.
 - Для WebLogic 11gR1, установленной в каталоге по умолчанию на Windows, добавьте в ваш путь следующую строку: C:\oracle\Middleware\wlserver_10.3\common\bin
 - Для WebSphere 7.0, установленной в каталоге по умолчанию на Windows, добавьте в ваш путь следующую строку: C:\IBM\WebSphere\AppServer1\bin
2. Присвойте переменной среды JAVA_HOME каталог, содержащий каталоги JDK bin и lib. Приводятся два примера.
 - Для WebLogic 11gR1 на Windows присвойте параметру JAVA_HOME значение C:\oracle\Middleware\jdk160_18
 - Для WebSphere 7.0 на Windows присвойте параметру JAVA_HOME значение C:\IBM\WebSphere\AppServer1\java\jre

Создание блока проверки допустимости

PDK проверки IBM Campaign предоставляет сценарий Ant, который может построить весь код в примерах файлов.

Поведение по умолчанию для сценария должно создать файл JAR, содержащий классы проверки. Дополнительно можно также создать Javadoc и выполнить испытания блоков проверки допустимости, чтобы убедиться, чтобы они работают в Campaign перед использованием плагин в производственной среде.

Создать блок проверки допустимости:

1. Перейдите в каталог PDK, <домашний_каталог_IBM_IMS\
домашний_каталог_Campaign>\devkits\validation\build
Например: C:\IBM\IMS\Campaign\devkits\validation\build
В этом каталоге содержится сценарий Ant, build.xml.

2. Выполните контейнер Ant в командной строке.

- Чтобы создать плагин, не выполняя никаких проверок модулей, используйте команду ant jar.
- Чтобы создать плагин, а также произвести проверку модулей, используйте команду ant run-test.

Ant выполнит сценарий и сгенерирует файл JAR validator.jar в подкаталоге lib.
Например:

```
C:\IBM\IMS\Campaign\devkits\validation\build\lib
```

Теперь у вас теперь есть нестандартный элемент проверки, который может использоваться в IBM Campaign. Ваш следующий шаг - конфигурация Campaign для использования этого блока проверки допустимости.

Конфигурирование Campaign для использования плагина проверки

Чтобы настроить Campaign для использования плагина проверки, используйте настройки конфигурирования в разделах Campaign > > разделе [n] > проверка.

Свойства конфигурации содержат инструкции по Campaign поиску класса плагинов и способпередачи конфигурационной информации на плагины.

Примечание: Проверка работает с множественными разделами; раздел [n] может быть изменен на любое имя раздела для обеспечения методовпроверки разделов.

Можно настроить следующие параметры конфигурации проверки:

- “validationClass” на стр. 9
- “validationClasspath” на стр. 9
- “validatorConfigString” на стр. 10

Чтобы использовать SimpleCampaignValidator, настройте свойства следующим образом:

- validationClasspath: *Unica*\campaign\devkits\validation\lib\validator.jar
- validationClass:
com.unica.campaign.core.validation.samples.SimpleCampaignValidator
- Свойство validatorConfigString не должно быть установлено для использования SimpleCampaignValidator, така как не использует строку конфигурации.

Чтобы использовать ExecutableCampaignValidator, установите свойства следующим образом:

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

validationClass

validationClass указывает Campaign название класса, которые должен использоваться для валидации с помощью плагина Validation PDK.

Свойство	Описание
Описание	Имя класса для использования при проверке. Значение свойства validationClasspath указывает на месторасположение данного класса.
Сведения	Класс должен быть задан полностью вместе с его именем пакета. Если данное свойство не установлено, Campaign не выполняет никакой выборочной проверки с валидацией.
Пример	com.unica.campaign.core.validation.samples.SimpleCampaignValidator Данный пример задает validationClass для класса SimpleCampaignValidator из примера кода.
По умолчанию	По умолчанию путь не устанавливается: <название свойства="validationClass" />

validationClasspath

validationClasspath назначает для Campaign месторасположение класса, который должен использоваться для валидации с помощью плагина Validation PDK.

Свойство	Описание
Описание	Путь к классу, используемому для пользовательской проверки.
Сведения	Используйте или полный путь, или относительный путь. Если путь относительный, поведение зависит от сервера приложений, на котором выполняется Campaign. WebLogic использует путь рабочего каталога домена, которым, по умолчанию, является c:\bea\user_projects\domains\mydomain. Если путь заканчивается на наклонную черту (/ для UNIX или \ для Windows), Campaign считает, что он ведет к месторасположению класса плагина Java, который должен использоваться. Если путь не заканчивается на наклонную черту, Campaign допускает, что это - название файла .jar, в котором содержится класс Java в соответствии с примером ниже. Если путь не определен, Campaign не принимает попыток загрузить плагин.
Пример	/<CAMPAIGN_HOME>/devkits/validation/lib/validator.jar Это - путь на платформе UNIX, который ведет к файлу JAR, упакованному с комплектом плагинов разработчика.

Свойство	Описание
По умолчанию	По умолчанию путь не устанавливается: <code><имя свойства="validationClasspath" /></code>
Смотрите также	Смотрите “validationClass” на стр. 9 для получения более детальной информации о назначении класса, который необходимо использовать.

validatorConfigString

validatorConfigString передается в плагин проверки доступности, когда она загружается с помощью Campaign.

Свойство	Описание
Описание	Строка, передающаяся плагину проверки доступности, когда он загружается с помощью Campaign.
Сведения	Как модуль plug-in использует эту строку, решать разработчику. Вы можете использовать его для отправки строки конфигурации вашему плагину, когда он загружается системой. Например, ExecutableCampaignValidator (из примера выполняемого плагина, который включен с PDK) использует данное свойство для указания на то, какая исполняемая программа должна выполняться.
Пример	Для запуска пробного сценария оболочки Борна в качестве проверяющего сценария, установите validatorConfigString по /opt/unica/campaign/devkits/validation/src/com/unica/campaign/core/validation/samples/validate.sh
По умолчанию	По умолчанию путь не устанавливается: <code><имя свойства="validatorConfigString" /></code>

Тестирование конфигурации блока проверки допустимости

Плагин можно протестировать и использовать после создания файла validator.jar, содержащего класс SimpleCampaignValidator, а также после внесения необходимых изменений конфигурации.

Следующий пример плагина не разрешает Campaign пользователям сохранить кампанию под названием "badCampaign".

Чтобы протестировать вашу конфигурацию:

1. Повторно внедрите свой сервер приложений, тогда изменения вступят в силу. Инструкции смотрите в документации по вашему серверу.
2. Войдите в систему IBM Campaign и перейдите на страницу создания кампании.
3. Создайте кампанию с именем **badCampaign** и попытайтесь сохранить ее.

Если всё правильно сконфигурировано, то вы не сможете сохранить новую кампанию. При получении сообщения об ошибке из блока проверки допустимости, вам станет известно, что она работает правильно.

Создание блока проверки допустимости

Выполните эти инструкции, чтобы создать модуль плагин проверки, очень похожий на SimpleCampaignValidator, но запрещающий создавать кампании с именем "badCampaign2."

1. Создайте копию примера блока проверки SimpleCampaignValidator.java в каталоге <домашний_каталог_IBM_IMS\домашний_каталог_Campaign>\devkits\validation\src\com\unica\campaign\core\validation\samples. Задайте имя копии как MyCampaignValidator.java и оставьте в том же самом каталоге в виде источника. Например:
C:\IBM\IMS\Campaign\devkits\validation\src\com\unica\campaign\core\validation\samples\MyCampaignValidator.java
2. Откройте MyCampaignValidator.java в редакторе. Найдите в документе слово "badCampaign" и замените его словом "badCampaign2".
3. Сохраните файл и закройте редактор.
4. Повторно создайте блоки проверки допустимости. Дополнительная информация приведена в разделе "Создание блока проверки допустимости" на стр. 8. Если сервер приложений блокирует файл validate.jar, когда он используется, остановите сервер перед созданием блоков проверки.
5. Переконфигурируйте файл campaign_config.xml, чтобы использовать новый класс: <property name="validationClass" value="com.unica.campaign.core.validation.samples.MyCampaignValidator">
6. Протестируйте блок проверки допустимости. Дополнительная информация приведена в разделе "Тестирование конфигурации блока проверки допустимости" на стр. 10.

Убедитесь, что блок проверки работает: вы не должны иметь возможность сохранять кампании с именем "badCampaign2."

Пример сценария проверки: предотвращение редактирования кампании

В этом примере объясняется, как использовать проверку для предотвращения конкретных изменений в кампании.

При попытке предотвратить редактирование кампании для изменения кода кампании, можно использовать пользовательскую подпрограмму проверки кампании. Подпрограмма убеждается, что следующая проверка была выполнена в момент сохранения кампании:

```
new_campaign_code == old_campaign_code
```

Чтобы выполнить эту задачу при первом создании кампании, установите флаг для подпрограммы с указанием проверки новой (создание) или существующей (редактирование) кампании. Если этот флаг указывает **правка**, тогда сравните коды кампании.

Приложение кампании устанавливает этот флаг на значение InputValidationData при передаче на плагин. Модуль plug-in читает флаг, определяющий, проводится ли проверка для новой или измененной кампании.

Глава 3. Вызов приложения для обработки проверки

Проверка PDK включает демонстрационный агент проверки, ExecutableCampaignValidator, который выполняет выполняемый модуль validate.sh в командной строке для выполнения проверки.

В следующих разделах приводится разъяснение по:

- Конфигурации Campaign для выполнения демонстрационного плагина выполняемого модуля, и
- Создания своего собственного плагина выполняемого модуля, который соответствует интерфейсу использования выполняемого модуля.

Конфигурирование Campaign для использования демонстрационного выполняемого плагина

Чтобы настроить Campaign для использования плагина проверки, используйте настройки конфигурирования в разделах Campaign > > разделе [n] > проверка.

Задайте свойства следующим образом:

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

Демонстрационный сценарий, который соединяется с инструментом проверки PDK, является сценарием оболочки Bourne для ОС UNIX. Он отказывает в создании кампании любому, у кого имя пользователя "badUser". Можно просмотреть код для этой исполняемой программы в следующем каталоге:

```
devkits\validation\src\com\unica\campaign\core\validation\
samples\validate.sh
```

Надо разработать собственный сценарий, выполняющий соответствующую проверку для реализации. Языки сценариев, такие как PERL и Python, являются хорошими вариантами для сценариев обработки текста; при этом приемлем любой язык, который может быть выполнен из командной строки.

Ожидаемый исполнимый интерфейс использования

Плагин ExecutableCampaignValidator вызывает выполняемый файл с помощью командной строкой, содержащей следующие параметры.

- executable_name: строка, установленная на validatorConfigString в IBM Marketing Platform.
- data_filename: имя файла, который выполняемый модуль считывает как входные данные. Входные данные должны быть отформатированы в XML.
- expected_result_filename: имя файла, который выполняемый модуль должен отправить в виде выходных данных. Ожидаемые результаты представлены в виде файла данныхxxx.xml, где XXX - число.

- Ниже приводится пример того, как отправляются успешные данные:

```
<ValidationResult result="0" generalFailureMessage="" />
```
- Ниже приводится пример того, как отправляются сбойные данные:

```
<ValidationResult result="1" generalFailureMessage="">  
  <AttributeError attributeName="someAttribute" errorMessage="something" />  
  <AttributeError attributeName="someAttribute2" errorMessage="something2" />  
</ValidationResult>
```
- Текст в XML-файле должен быть закодирован обычными символами ASCII или UTF-8.

Примечание: Настоятельно рекомендовано сделать для пользователей легкие для понимания сообщения об ошибках, чтобы пользователи могли исправить проблему перед тем, как попытаться сохранить операцию повторно.

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).

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 - 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