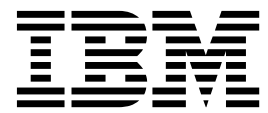


Version 9 Release 1.2
23. September 2015

*IBM Marketing Operations
Integration Module*



Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 33 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9, Release 1, Modifikation 2 von IBM Marketing Operations und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM Marketing Operations Integration Module, Version 9 Release 1.2,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2002, 2015

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
September 2015

© Copyright IBM Corporation 2002, 2015.

Inhaltsverzeichnis

Kapitel 1. Was sind die IBM Marketing Operations Integration Services? 1

Welche Voraussetzungen gelten für Marketing Operations Integration Services?	2
Grundlegende Informationen zu IBM Marketing Operations Integration Services	3
Installieren von Integration Services	6
Inhalt des Software-Development-Kits	6
Per Hosting bereitgestellte JavaDocs	7
Dokumentation und Hilfe zu Marketing Operations	7

Kapitel 2. Marketing Operations Integration Webservice 9

WSDL von Marketing Operations Integration Services	9
executeProcedure	9
Marketing Operations Integration Webservice-Datentypen	10

Kapitel 3. IBM Marketing Operations-Prozeduren 15

Voraussetzungen	15
Konfigurationsparameter	17
Design	17
Lebenszyklus von Prozeduren	18

Wichtige Java-Klassen	19
Datensperre	20
Prozedurtransaktionen.	20
Prozedurkommunikation	21
Prozedurprotokollierung	21
Prozedur-Plug-in-Definitiondatei	21

Kapitel 4. IBM Marketing Operations-API 23

Inhalt des IBM Marketing Operations API	23
API-Schnittstellen	23
Häufig auftretende Ausnahmen.	24
Handles	24
Attributzuordnung (AttributeMap)	26
Aufgelistete Datentypen	27

Kontaktaufnahme zum technischen Support von IBM 31

Bemerkungen. 33

Marken.	35
Hinweise zu Datenschutzrichtlinien und Nutzungsbedingungen	35

Kapitel 1. Was sind die IBM Marketing Operations Integration Services?

IBM®Marketing Operations Integration Services kombiniert Marketing Operations Integration Webservice, API-Prozeduren und Trigger, um Geschäftskompetenzen zu erweitern.

Die IBM Marketing Operations Integration Services setzen sich aus folgenden Komponenten zusammen.

- **Marketing Operations Integration Webservice**

Integration Services bietet Marketing Operations-Kunden und IBM Professional Services eine Möglichkeit, Marketing Operations in andere Anwendungen in ihrer Umgebung zu integrieren.

- **Marketing Operations-Prozeduren und -API**

Benutzerdefinierte Prozeduren können in Marketing Operations definiert werden, um Marketing Operations-Geschäftslogiken beliebig zu erweitern. Nachdem Sie Prozeduren definiert haben, können diese Prozeduren die Ziele für Aufrufe des Web-Service Integration Services aus anderen Anwendungen sein. Prozeduren können auch zum Senden von Nachrichten an andere Anwendungen definiert werden.

- **Marketing Operations-Trigger**

Trigger können Ereignissen und Prozeduren in Marketing Operations zugeordnet sein. Wenn eines dieser Ereignisse auftritt, wird der zugeordnete Trigger ausgeführt.

Versionen und Abwärtskompatibilität

Zukünftige Versionen von Integration Services werden abwärtskompatibel mit allen untergeordneten Releases und Wartungsreleases mit gemeinsamer Hauptversionsnummer sein. IBM behält sich jedoch das Recht vor, keine Kompatibilität mit früheren Versionen für Hauptreleases (x.0) zu gewährleisten, wenn die vorliegende Geschäftssituation bzw. die technischen Umstände dies rechtfertigen.

Die Hauptversionsnummer dieser API wird inkrementell erhöht, wenn eine der folgenden Änderungen vorgenommen wird.

- Änderungen der Dateninterpretation
- Änderung der Geschäftslogik (beispielsweise Änderungen in den Funktionen der Servicemethode)
- Änderungen der Methodenparameter und/oder der Rückgabetypen

Die Nebenversionsnummer dieser API wird inkrementell erhöht, wenn eine der folgenden Änderungen vorgenommen wird. Diese Änderungen sind per Definition mit einer früheren Version kompatibel.

- Neue Methode hinzugefügt
- Neuer Datentyp hinzugefügt und dessen Verwendung auf eine neue Methode beschränkt
- Neues Element zu einem Aufzählungstyp hinzugefügt
- Eine neue Version einer Benutzeroberfläche wird mit einem Versionsuffix definiert

Authentifizierung

Eine Authentifizierung ist nicht erforderlich. Alle Clients sind einem bekannten IBM Marketing Operations-Benutzer mit dem Namen "PlanAPIUser" zugeordnet. Ein Systemadministrator konfiguriert die Sicherheitsfunktionen dieses Benutzers mit Sonderberechtigung entsprechend den Anforderungen aller Web-Service-Clients.

Ländereinstellung

Als Ländereinstellung wird ausschließlich die derzeit für die IBM Marketing Operations-Systeminstanz konfigurierte Ländereinstellung unterstützt. Es wird für alle von der Ländereinstellung abhängigen Daten, wie Nachrichten und Währungen, die Ländereinstellung des Systems vorausgesetzt.

Statusverwaltung

API und Web-Service sind *statusunabhängig*. Die Serviceimplementierung speichert keine clientspezifischen Informationen für nachfolgende API-Aufrufe. Mit dieser Funktion wird eine effiziente Serviceimplementierung ermöglicht und die Clusterunterstützung vereinfacht.

Datenbanktransaktionen

Marketing Operations Integration Services legt dem Client keine Datenbanktransaktionen offen, verwendet aber entsprechende Informationen, sofern sie im Ausführungskontext miteinbezogen sind. Nach dem Start einer Transaktion sind die Auswirkungen aller API-Aufrufe innerhalb einer bestimmten Prozedur atomar. Das bedeutet, dass die Datenbank bei einem fehlgeschlagenen API-Aufruf in dem Status verbleibt, den sie vor dem Aufruf der API hatte. Andere Benutzer von Marketing Operations können die Änderungen erst einsehen, nachdem die Prozedur die Transaktion erfolgreich ausgeführt hat.

API-Aufrufe, die die Datenbank aktualisieren, müssen zunächst eine Bearbeitungssperre anfordern, um zu verhindern, dass andere Benutzer die zugrundeliegenden Daten während der API-Aufrufe verändern. Andere Benutzer können gesperrte Komponenten nicht aktualisieren, bis der API-Aufruf abgeschlossen ist. Ebenso müssen die nächsten Marketing Operations-Benutzer oder API-Clients die Daten-sperre anfordern, bevor ein weiterer API-Aufruf übergeben wird.

Ereignisverarbeitung

Wenn eine Operation auf einer IBM Marketing Operations-Komponente über die API ausgeführt wird, werden dabei die gleichen Ereignisse generiert wie bei einer Ausführung der Operation durch einen Marketing Operations-Benutzer. Benutzer, die bestimmte Benachrichtigungen abonniert haben, beispielsweise bei der Statusänderung eines Projekts, werden über Statusänderungen benachrichtigt, die aus API-Aufrufen sowie aus Aktionen von Benutzern folgen.

Welche Voraussetzungen gelten für Marketing Operations Integration Services?

Für Marketing Operations Integration Services bestehen die folgenden Voraussetzungen.

Marketing Operations Integration Services müssen folgende Voraussetzungen erfüllen:

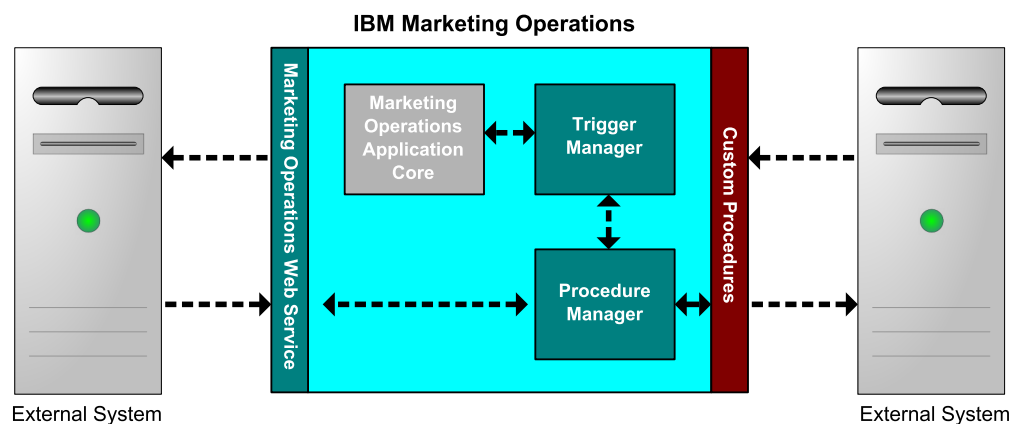
- Systemintegration flexibel verbinden.
- Einen Mechanismus für Kundenanwendungen bereitstellen, Marketing Operations über Web-Service-Aufrufe zu steuern.
- Einen Mechanismus für Kundenanwendungen bereitstellen, über bestimmte Ereignisse in Marketing Operations benachrichtigt zu werden.
- Ein einfaches Programmiermodell bereitstellen, das leicht verständlich und anwendbar ist.
- Bei der Wiederherstellung nach einer Störung stabil funktionieren.
- Datenintegrität garantieren.
- Integration mit vorhandenen grafisch orientierten Kunden von Marketing Operations bei minimalen Auswirkungen.
- Differenzierten Zugriff auf Marketing Operations-Komponenten bieten und dabei die zugrunde liegenden Implementierungsdetails vor Änderungen durch Programmierer schützen.

Grundlegende Informationen zu IBM Marketing Operations Integration Services

Mithilfe von IBM Marketing Operations Integration Services erstellen Sie benutzerdefinierte Prozeduren. Durch diese Prozeduren können Sie externe Ereignisse auslösen, wenn bestimmte Ereignisse innerhalb von Marketing Operations auftreten. Mit diesen Prozeduren können Sie Marketing Operations-Funktionen aus externen Systemen oder Programmen ausführen.

Die API-Schnittstelle interagiert mit IBM Marketing Operations auf Programmebene wie die Benutzeroberfläche von Marketing Operations auf Benutzerebene interagiert. Das Erstellen von Prozeduren erfolgt unter Verwendung der API. Mithilfe dieser Prozeduren ermöglichen Sie die Kommunikation zwischen Marketing Operations und externen Systemen. Der Web-Service Marketing Operations ist ein Containerobjekt für Prozeduren, API und Trigger.

Die Architektur der Marketing Operations Integration Services wird hier dargestellt.



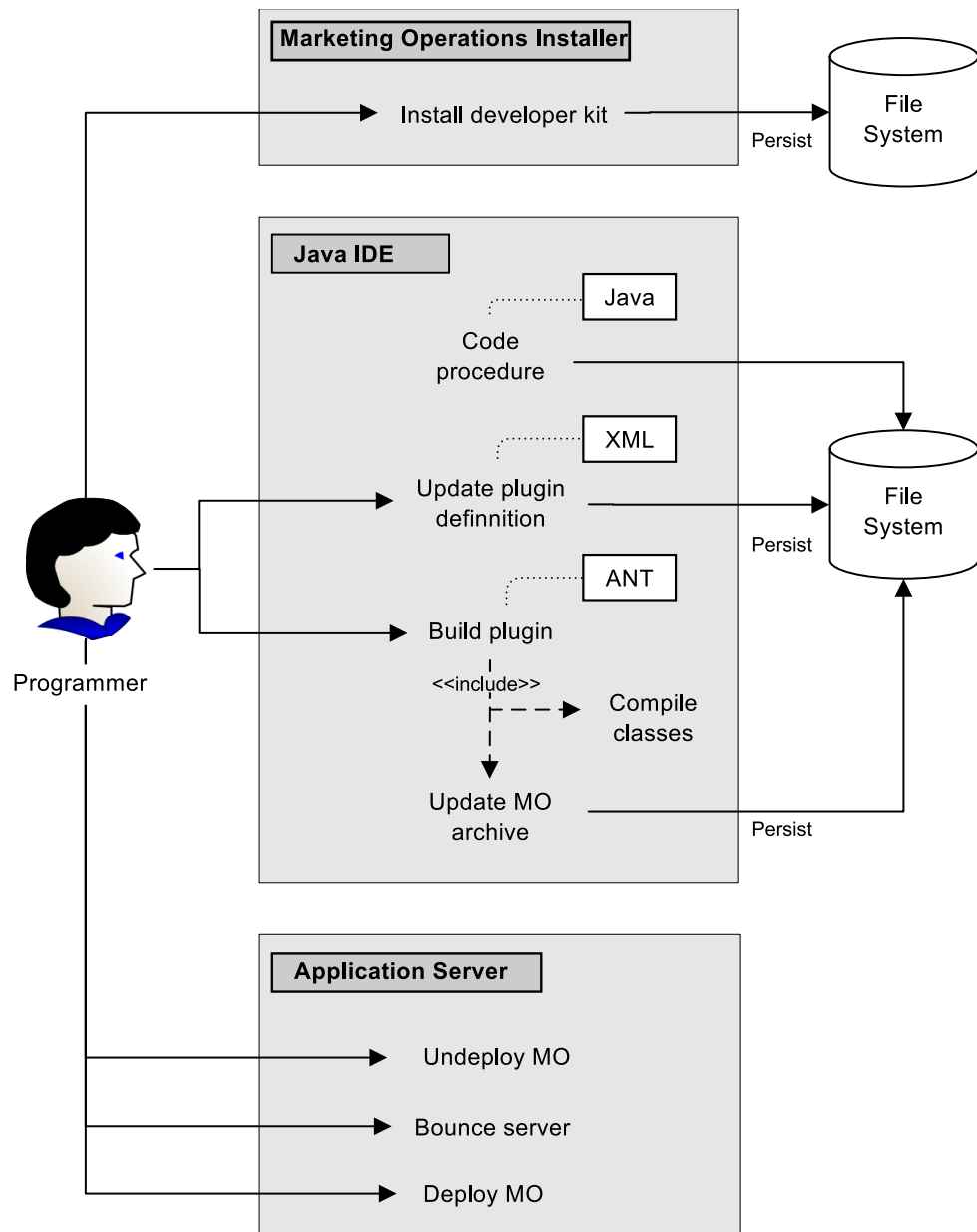
Die folgenden sind Schlüsselkomponenten der Integration Services.

- Marketing Operations-Prozedurmanager: erweitert die Geschäftslogik durch Interaktion mit Marketing Operations über die API.

- Marketing Operations-Trigger-Manager: ordnet eine Bedingung (zum Beispiel die Statusänderung eines Marketingobjekts) einer Aktion zu (eine Prozedur, die ausgeführt wird, wenn die Bedingung für den Trigger erfüllt ist).

Methoden

Die Komponenten von IBM Marketing Operations Integration Services werden, wie im folgenden Diagramm dargestellt, zur Entwicklung von benutzerdefinierten Prozeduren verwendet.



Führen Sie nach der Installation des Development-Kits die folgenden grundlegenden Schritte aus:

1. Geben Sie den Code für die benutzerdefinierte Prozedur ein. Derzeit müssen Sie hierzu Java™ verwenden.
2. Aktualisieren Sie die Plug-in-Definition in der XML-Definitionsdatei.

3. Erstellen Sie das Plug-in:
 - a. Kompilieren Sie die erforderlichen Klassen.
 - b. Wenn Sie eine Bibliothek von Fremdanbietern verwenden, die nicht im Marketing Operations-Archiv vorhanden ist, müssen Sie die Bibliothek in der Datei `plan.war` bündeln und die Bereitstellung erneut ausführen.
4. Starten Sie Marketing Operations erneut. Änderungen an den Prozedurklassen werden angewendet, wenn Sie den Anwendungsserver erneut starten.

Anmerkung: Wenn Sie die Datei `plan.war` ändern, müssen Sie die Bereitstellung von Marketing Operations zurücknehmen und die Bereitstellung mit der neuen Datei `plan.war` erneut ausführen. Wenn Sie eine Bibliothek von Fremdanbietern verwenden, die sich nicht im Marketing Operations-Archiv befindet, und Sie die Datei `plan.war` ändern, müssen Sie die Bereitstellung von Marketing Operations zurücknehmen und die Bereitstellung erneut ausführen.

Einfaches Beispiel für die Kommunikation zwischen IBM Marketing Operations und der API

Im folgenden einfachen Beispiel wird die Einrichtung der Kommunikation zwischen API und Marketing Operations beschrieben. Die Kommunikation in diesem Beispiel erfüllt keinen sinnvollen Zweck, es findet lediglich ein Umlauf zwischen Marketing Operations und den Integration Services statt.

In diesem Beispiel werden Teile der Beispielprozeduren verwendet, die im Development-Kit für Marketing Operations Integration Services enthalten sind. Sie finden die hier genannten Codebeispiele in den folgenden Dateien.

- `PlanClientFacade.java`
- `PlanWSNOOPTestCase.java`

Die `noop`-Methode ist ein Web-Service-Aufruf an Marketing Operations. Sie ist in der `PlanClientFacade`-Klasse definiert, und übergibt Nullwerte in einem Array.

```
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);
}
```

Die Prozedur `"testExecuteProcedure"` ruft die `noop`-Methode aus `"PlanClientFacade"` auf, um einen Umlauf mit der Anwendung Marketing Operations einzurichten.

```
public void testExecuteProcedure() throws Exception {
    // Time out after a minute
    int timeout = 60000;
    PlanClientFacade clientFacade = new PlanClientFacade(urlWebService, timeout);
    System.out.println("noop w/no parameters");
    long startTime = new Date().getTime();
    ProcedureResponse response = clientFacade.noop("junit-jobid");
    long duration = new Date().getTime() - startTime;

    // zero or positive status => success
    System.out.println("Status: " + response.getStatus());
    System.out.println("Duration: " + duration + " ms");
    assertTrue(response.getStatus() >= 0);
    System.out.println("Done.");
}
```

Ausführliche Informationen zu `NameValueArrays`, `ProcedureResponse` und anderen aufgelisteten Methoden und Datentypen finden Sie im Handbuch *Marketing*

Installieren von Integration Services

Das Modul IBM Marketing Operations Integration Services ist eine separate, kostenpflichtige Komponente. Wenn Sie das Integration Services-Modul gekauft haben, müssen Sie es installieren.

1. Laden Sie das Installationsprogramm für IBM Marketing Operations Integration Services herunter.
2. Die IBM EMM-Installationsprogramme erkennen das Integration Services-Modul.
3. Die Installationsprogramme legen Konfigurationseigenschaften auf **Marketing Operations | umoConfiguration | integrationServices | enableIntegrationServices** fest. Sie können die Installation durch Änderung der Konfigurationsparameter anpassen. Weitere Informationen hierzu finden Sie im Abschnitt „Konfigurationsparameter“ auf Seite 17.

Inhalt des Software-Development-Kits

Normalerweise werden alle Marketing Operations Integration Services-Komponenten in einem Ordner mit der Bezeichnung "devkits" installiert.

Das Software-Development-Kit enthält Dokumentationen (Javadoc-Dokumentationen mit allen publicapi-Klassen und Schnittstellen) und Mustercode.

Der Mustercode wird in den folgenden Ordnern installiert.

- Der Ordner **build** enthält Scripts zum Erstellen und Bereitstellen von benutzerdefinierten Prozeduren.
- Der Ordner **Classes** enthält die kompilierten Prozedurklassen.
Benutzer müssen die kompilierten Klassen ihrer benutzerdefinierten Prozeduren unter dem Pfad bereitstellen, der vom Konfigurationsparameter **integrationProcedureClasspathURL** angegeben wird. Anschließend werden sie vom IBM Marketing Operations-Prozedurmanager geladen, wie in der Konfigurationsdatei `procedure-plugins.xml` angegeben.
- Der Ordner **lib** enthält die erforderlichen Bibliotheken zur Entwicklung und Kompilierung von benutzerdefinierten Prozeduren.
- Der Ordner **src** enthält die Quellendateien der benutzerdefinierten Prozeduren. Benutzer können benutzerdefinierte Prozeduren, die als Trigger oder Web-Service gestartet werden sollen, hier speichern.
 - Der Ordner **src/procedure** enthält die Konfigurationsdatei `procedure-plugins.xml`. Jede benutzerdefinierte Prozedur, die als ein ereignisbasierter Trigger oder ein externer Web-Service ausgeführt wird, muss einen Eintrag in dieser Datei haben. Die Einträge müssen einen vollständig qualifizierten Klassenpfad der Prozeduren und der erforderlichen Initialisierungsparameter enthalten.
 - Der Ordner **src/procedure** enthält außerdem einige Beispielprozeduren, die im Lieferumfang von IBM Marketing Operations enthalten sind. Diese Prozeduren können verwendet werden, um benutzerdefinierte Prozeduren zu verstehen und zu entwickeln.
Speichern Sie benutzerdefinierte Prozeduren im Verzeichnis **src** in einer neuen Ordnerstruktur, wie beispielsweise `com/<mycompany>/<mypackage>`. Speichern Sie benutzerdefinierte Prozeduren nicht im Ordner für Beispielprozeduren.
 - Der Ordner **src/soap** enthält Beispiel-Web-Service-Clients, die in Java entwickelt werden. Verwenden Sie diese Beispiele als Ausgangspunkt, um Web-

Service-basierte Clients für Integration Services zu entwickeln. Dieser Ordner enthält außerdem binäre Scripts, um einfache Clients über die Befehlszeile zu starten.

Per Hosting bereitgestellte JavaDocs

Ausführliche Informationen über die öffentlichen API-Methoden finden Sie in der iPlanAPI-Klasse in den API-Dokumentationsdateien der JavaDocs.

Diese Dateien sind auf folgende Weise verfügbar:

- Greifen Sie auf die Dateien in dem Verzeichnis *<IBM_EMM>/<MarketingOperations_Home>/devkits/integration/javadocs* auf dem Server zu, der Marketing Operations hostet.
- Melden Sie sich bei Marketing Operations an, wählen Sie auf einer beliebigen Seite **Hilfe > Produktdokumentation** aus und laden Sie dann die Datei *IBM <version>PublicAPI.zip* herunter.

Dokumentation und Hilfe zu Marketing Operations

IBM Marketing Operations wird in Ihrem Unternehmen von verschiedenen Personen für verschiedene Aufgaben verwendet. Informationen zu Marketing Operations finden Sie in einer Reihe von Handbüchern, die jeweils für Teammitglieder mit bestimmten Zielen und speziellen Qualifikationsprofilen erstellt wurden.

In der folgenden Tabelle sind die Informationen, die in den einzelnen Handbüchern verfügbar sind, aufgelistet.

Tabelle 1. Handbücher in der Gesamtdokumentation zu Marketing Operations.

Die folgende dreispaltige Tabelle enthält in einer Spalte die Aufgaben, in der zweiten Spalte die Handbuchnamen und in der dritten Spalte die Zielgruppe.

In folgenden Fällen:	Siehe	Zielgruppe
• Projekte planen und verwalten • Workflowaufgaben, Eckdaten und Personal erstellen • Projektausgaben überwachen • Prüfungen und Freigaben zum Inhalt abrufen • Berichte erstellen	<i>IBM Marketing Operations-Benutzerhandbuch</i>	• Projektleiter • Designer • Marketing-Manager für Direktmailing
• Vorlagen, Formulare, Attribute und Metriken entwerfen • Benutzerschnittstelle anpassen • Ebenen des Benutzerzugriffs und der Sicherheit definieren • Zusatzfunktionen implementieren • Marketing Operations konfigurieren und optimieren	<i>IBM Marketing Operations-Administratorhandbuch</i>	• Projektleiter • IT-Administratoren • Implementierungsberater

Tabelle 1. Handbücher in der Gesamtdokumentation zu Marketing Operations (Forts.).

Die folgende dreispaltige Tabelle enthält in einer Spalte die Aufgaben, in der zweiten Spalte die Handbuchnamen und in der dritten Spalte die Zielgruppe.

In folgenden Fällen:	Siehe	Zielgruppe
- Marketingkampagnen erstellen - Angebote planen - Integration zwischen Marketing Operations und Campaign implementieren - Integration zwischen Marketing Operations und IBM Digital Recommendations implementieren	<i>IBM Marketing Operations- und IBM -Integrationshandbuch</i>	- Projektleiter - Fachleute für Marketingumsetzung - Direktmarketing-Manager
- Informationen zu neuen Systemfeatures - Bekannte Probleme und deren Umgehung recherchieren	<i>IBM Marketing Operations Releaseinformationen</i>	Alle Benutzer von Marketing Operations
- Marketing Operations installieren - Konfigurieren von Marketing Operations - Upgrade auf eine neue Version von Marketing Operations	<i>IBM Marketing Operations- Installationshandbuch</i>	- Berater für Softwareimplementierungen - IT-Administratoren - Datenbankadministratoren
Benutzerdefinierte Verfahren zur Integration von Marketing Operations in andere Anwendungen erstellen	<i>IBM Marketing Operations Integration Module</i> und die API-JavaDocs, die zur Verfügung stehen, wenn Sie in Marketing Operations auf Hilfe > Produktdokumentation klicken und dann die Datei IBM<Version>PublicAPI.zip herunterladen	- IT-Administratoren - Datenbankadministratoren - Implementierungsberater
Informationen zur Struktur der Marketing Operations-Datenbank	<i>IBM Marketing Operations- Systemschema</i>	Datenbankadministratoren
Weiterführende Informationen in der Praxis	- Hilfe aufrufen und <i>Marketing Operations-Benutzerhandbuch</i>, <i>-Administratorhandbuch</i> oder <i>Marketing Operations-Installationshandbuch</i> suchen oder durchsuchen: Klicken Sie auf Hilfe > Hilfe zu dieser Seite - Alle Handbücher zu Marketing Operations aufrufen: Klicken Sie auf Hilfe > Produktdokumentation. - Handbücher zu allen IBM Enterprise Marketing Management (EMM)-Produkten aufrufen: Klicken Sie auf Hilfe > Komplette Dokumentation IBM EMM Suite.	Alle Benutzer von Marketing Operations

Kapitel 2. Marketing Operations Integration Webservice

Der Web-Service stellt eine Clientansicht der Marketing Operations Integration Services im Rahmen der Bereitstellung des IBM Marketing Operations-Servers zur Verfügung. Der Service wird gleichzeitig von Marketing Operations-Webbenutzern verwendet.

Der Web-Service unterstützt einen API-Aufruf: `executeProcedure`.

Dieser Web-Service-Anruf wird von einem Client direkt ausgeführt.

WSDL von Marketing Operations Integration Services

Die Web Service Definition Language (WSDL) wurde manuell definiert und ist die ultimative Web-Service-Definition.

Axis

Diese Web-Service-Version verwendet Axis2 1.5.2 zur Generierung der serverseitigen Klassen, aus denen die Web-Service-Implementierung in der WSDL-Datei besteht. Benutzer können jede beliebige Version von Axis oder auch ein anderes Verfahren als Axis verwenden, um eine clientseitige Implementierung für die Integration in die API aus der bereitgestellten WSDL zu erstellen.

Protokollversion

Die Version des Protokolls ist wie folgt explizit an die WSDL gebunden:

- Als Teil des WSDL-Namens, beispielsweise `PlanIntegrationService1.0.wsdl`
- Als Teil des WSDL-Zielnamensbereichs (`targetNamespace`), zum Beispiel
`xmlns:tns="http://webservices.unica.com /MktOps/services/
PlanIntegrationServices1.0?wsdl"`

WSDL

Eine WSDL-Datei wird mit IBM Marketing Operations Integration Services bereitgestellt: `PlanIntegrationServices1.0.wsdl`. Die WSDL befindet sich bei Auslieferung im Verzeichnis `integration/examples/soap/plan`. Das Beispiel-Erstellungsskript verwendet diese Datei bei der Generierung der erforderlichen clientseitigen Stubs für die Verbindung mit dem Web-Service.

`executeProcedure`

`executeProcedure` ist der vom Web-Service unterstützte API-Aufruf.

Syntax

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

Rückgabe

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

Beschreibung

Diese Methode ruft die angegebene Prozedur mit einem optionalen Array von Parametern auf. Der Aufruf wird synchron ausgeführt, das heißt, der Client wird blockiert und bei Abschluss des Aufrufs wird das Ergebnis ausgegeben.

Parameter

Tabelle 2. *executeProcedure-Parameter*

Name	Beschreibung
key	Der eindeutige Schlüssel der auszuführenden Prozedur. Es wird ein <i>RemoteException</i> -Fehler zurückgegeben, wenn an key keine Prozedur gebunden ist.
jobid	Optionale Zeichenfolge, die den Job ermittelt, der der Ausführung dieser Prozedur zugeordnet ist. Diese Zeichenfolge ist ein Durchgriffselement, sie kann jedoch verwendet werden, um Clientjobs an die Ausführung einer bestimmten Prozedur zu binden.
paramArray	Ein Array aus Parametern, das an die Prozedur übergeben werden soll. Es werden ein Fehlerstatus und eine Fehlermeldung zurückgegeben, wenn mindestens ein Parameter ungültig ist (beispielsweise ein falscher Typ oder ein unzulässiger Wert). Es ist dem Client überlassen, die Parameter, ihre Typen und die Anzahl der Werte zu bestimmen, die für die Prozedur erforderlich sind.

Rückgabeparameter

Tabelle 3. *Rückgabeparameter für "executeProcedure"*

Name	Beschreibung
Status	Ein ganzzahliger Code: • 0 gibt an, dass die Prozedur erfolgreich ausgeführt wurde • eine ganze Zahl gibt einen Fehler an Prozeduren können über den Status verschiedene Fehlerstufen angeben.
messages	Ein Array aus null oder mehr Nachrichtendatenstrukturen. Wenn status den Wert 0 hat, enthält dieses Array keine ERROR-Nachrichten, kann jedoch INFORMATION- und WARNING-Nachrichten enthalten. Wenn status einen anderen Wert als 0 hat, können die Nachrichten aus einer beliebigen Kombination aus ERROR-, INFORMATION- und WARNING-Nachrichten bestehen.

Marketing Operations Integration Webservice-Datentypen

Die vom Web-Service verwendeten Datentypen sind unabhängig von einer bestimmten Servicebindung oder Programmierimplementierung.

Dies erfolgt in folgender Schreibweise:

- **<type>**: *<type definition>* definiert einen einfachen Datentyp. Beispiel:
Handle: Zeichenfolge
- **<type>**: [*<type definition>*] definiert einen komplexen Datentyp oder eine Datenstruktur.

- `<type>: { <type definition> }` definiert einen komplexen Datentyp oder eine Datenstruktur.

Elemente, die einen komplexen Typ aufweisen, und API-Parameter können mithilfe dieser Typen Arrays deklarieren. Beispiel:

```
Handle [] handles
```

Der Typ "handles" ist ein Array aus Handle-Typen.

Primitive Typen

Primitive Typen sind auf Typen beschränkt, die in der nachfolgenden Tabelle definiert sind, um die Unterstützung von SOAP 1.1-Bindungen zu erleichtern. Alle Typen können als Arrays deklariert sein, zum Beispiel **String []**. An sich können binäre Datentypen, wie **long**, von Protokollbindungen (z. B. SOAP) in Form von Zeichenfolgen dargestellt werden. Diese Darstellung hat jedoch keinerlei Auswirkungen darauf, wie die Semantik des Typs (z. B. gültige Werte) dem Client angezeigt wird.

Tabelle 4. Primitive Typen

API-Typ	Beschreibung	SOAP-Typ	Java-Typ
boolean	boolescher Wert: wahr oder falsch	xsd:boolean	boolean
dateTime	Ein Datum-Zeit-Wert	xsd:dateTime	Datum
decimal	Ein Dezimalwert mit beliebiger Genauigkeit	xsd:decimal	java.math.BigDecimal
double	Ein signierter Dezimalwert mit doppelter Genauigkeit	xsd:double	double
int	Ein signierter 32-Bit-Ganzzahlwert	xsd:int	int
integer	Ein signierter Ganzzahlwert mit beliebiger Genauigkeit	xsd:integer	java.math.BigInteger
long	Ein signierter 64-Bit-Ganzzahlwert	xsd:long	long
string	Eine Zeichenfolge aus Unicode-Zeichen	xsd:string	java.lang.String

MessageTypeEnum

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

"MessageTypeEnum" ist ein Aufzählungstyp, der alle möglichen Nachrichtentypen definiert.

- INFORMATION: eine Informationsnachricht
- WARNING: eine Warnnachricht
- ERROR: eine Fehlernachricht

Message

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

"Message" ist eine Datenstruktur, die das Ergebnis eines Web-Service-API-Aufrufs definiert. Sie stellt optionale Felder für nicht lokalisierten Code, lokalisierten Text und Protokolldetails bereit. Derzeit wird in allen lokalisierten Texten die für die IBM Marketing Operations-Serverinstanz festgelegte Ländereinstellung verwendet.

Tabelle 5. Nachrichtenparameter

Parameter	Beschreibung
type	"MessageTypeEnum", legt den Typ der Nachricht fest.
code	Ein optionaler Code für die Nachricht im Zeichenfolgeformat.
localizedText	Eine optionale Textzeichenfolge, die der Nachricht zugeordnet werden soll.
logDetail	Eine optionale Stack-Trace-Nachricht.

NameValue

NameValue: [string name, int sequence]

"NameValue" ist ein komplexer Basistyp, der ein Name/Wert-Paar definiert. Dieser Typ definiert auch eine optionale Sequenz, durch die der Service nach Bedarf Werte-Arrays erstellt (die Sequenzen sind nullbasiert).

Alle "NameValues" mit dem gleichen Namen und unterschiedlichen Folgenummern werden in ein Werte-Array konvertiert und dem gemeinsamen Namen zugeordnet.

Die Array-Größe wird durch die höchste Folgenummer bestimmt, nicht angegebene Array-Elemente haben Nullwerte. Die Array-Folgenummern müssen eindeutig sein. Der Wert und sein Typ werden durch den erweiterten Typ angegeben.

Tabelle 6. Parameter von "NameValue"

Parameter	Beschreibung
name	Eine Zeichenfolge, die den Namen eines NameValue-Typs definiert.
sequence	Eine nullbasierte ganze Zahl, die die Folgenummer für den durch "NameValue" angegebenen Wert festlegt.

Erweiterte NameValue-Typen werden wie folgt für die einzelnen primitiven Typen definiert:

Tabelle 7. Erweiterte NameValue-Typen

Erweiterter Typ	Beschreibung
BigDecimalNameValue: NameValue [decimal value]	Ein NameValue-Typ, dessen Wert eine Dezimalzahl mit beliebiger Genauigkeit ist.
BigIntegerNameValue: NameValue [integer value]	Ein NameValue-Typ, dessen Wert eine Ganzzahl beliebiger Größe ist.
BooleanNameValue: NameValue [boolescher Wert]	Ein NameValue-Typ, der einen booleschen Wert hat.

Tabelle 7. Erweiterte NameValue-Typen (Forts.)

Erweiterter Typ	Beschreibung
CurrencyNameValue: NameValue [string locale, decimal value]	Ein NameValue-Typ, der zur Darstellung der Währung für eine Ländereinstellung geeignet ist. Die Ländereinstellung wird im ISO-Sprachencode angegeben, eine für die einzelnen Regionen in ISO-639 festgelegte Kombination aus zwei Kleinbuchstaben. Derzeit muss die Ländereinstellung mit der für IBM Marketing Operations-Serverinstanz festgelegte Ländereinstellung übereinstimmen.
DateNameValue: NameValue [datetime value]	Ein NameValue-Typ, dessen Wert ein Datum ist.
DecimalNameValue: NameValue [double value]	Ein NameValue-Typ, dessen Wert eine Dezimalzahl mit doppelter Genauigkeit ist.
IntegerNameValue: NameValue [long value]	Ein NameValue-Typ, dessen Wert eine 64-Bit-Ganzzahl ist.
String NameValue: NameValue [string value]	Ein NameValue-Typ, dessen Wert eine Zeichenfolge ist.

Ein Array aus erweiterten NameValue-Typen ist für die Verwendung definiert, wenn Sie eine Gruppe aus NameValue-Werten mit unterschiedlichen Typen definieren müssen.

```

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

```

Kapitel 3. IBM Marketing Operations-Prozeduren

Eine "Prozedur" ist eine benutzerdefinierte oder eine Java-Standardklasse, die von IBM Marketing Operations per Hosting bereitgestellt wird und eine Arbeitseinheit ausführt. Prozeduren bieten Kunden und IBM Professional Services die Möglichkeit, Marketing Operations-Geschäftslogiken auf beliebige Weise zu erweitern.

Um Auswirkung auf die von Marketing Operations verwalteten Komponenten zu haben, folgen Prozeduren einem einfachen Programmiermodell mit einer klar strukturierten API. Prozeduren werden mithilfe eines einfachen Suchmechanismus und einer XML-basierten Definitionsdatei "erkannt". Marketing Operations führt die Prozeduren gemäß den Bedürfnissen der "Clients" aus. Beispielsweise als Reaktion auf eine Integrationsanforderung (eingehend) oder eine Initialisierung eines Triggers (intern oder ausgehend).

Prozeduren werden synchron mit dem Client ausgeführt. Ergebnisse werden dem Client direkt zur Verfügung gestellt und über einen persistenten Überwachungsmechanismus. Die Ausführung einer Prozedur kann auch ein Auslösen weiterer Ereignisse und Trigger in Marketing Operations mit sich bringen.

Prozeduren müssen in Java geschrieben sein.

Voraussetzungen

Die Prozedurimplementierungsklassen sind als Paket in einer separaten Klassenstruktur bzw. einer JAR-Datei zusammengefasst und werden IBM Marketing Operations über einen URL-Pfad zur Verfügung gestellt.

Prozedurimplementierung

Der Prozedurenausführungsmanager verwendet ein unabhängiges Klassenladeprogramm, um diese Klassen bei Bedarf zu laden. Marketing Operations sucht standardmäßig im folgenden Verzeichnis.

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

Um diese Standardeinstellung zu ändern, setzen Sie den Parameter **integrationProcedureClasspathURL** unter **Einstellungen > Konfiguration > Marketing Operations > umoConfiguration > integrationServices**.

Der Name der Prozedurimplementierungsklasse entspricht der gültigen Java-Namenskonvention, um Konflikte der Paketbenennungen mit "unica" und Klassen anderer Anbieter zu vermeiden. Es ist wichtig, dass der Kunde Prozeduren nicht in der Paketstruktur "com.unica" oder "com.unicacorp" ablegt.

Die Prozedurimplementierung ist entsprechend der Java-Laufzeitversion codiert, die IBM Marketing Operations auf dem Anwendungsserver verwendet (JRE 1.5.10 oder höher).

Die Prozedurimplementierungsklasse wird von der Klassenladerichtlinie geladen, die IBM Marketing Operations normalerweise verwendet (in der Regel **parent-last**). Der Anwendungsserver kann Entwicklungstools und Optionen zum erneuten La-

den von Klassen bereitstellen, die für Marketing Operations-Prozeduren gelten, dies ist jedoch nicht erforderlich.

Bibliotheken

IBM Marketing Operations stellt eine Reihe von Open-Source-Bibliotheken und Bibliotheken von Fremdanbietern bereit. Anwendungsserver verwenden ebenfalls verschiedene Versionen dieser Bibliotheken.

Im Allgemeinen wird diese Liste von Release zu Release geändert. Die folgenden Bibliotheken von Fremdanbietern werden in der Version 9.1.0 unterstützt.

- Ant 1.6.5 (ant.jar)
- Axis2 1.5.2 und abhängige Versionen
 - 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
 - xlsxScanner.jar
 - xlsxScannerUtils.jar
 - xlsxWASparsing.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-APIs (affinium_plan.jar)
- IBM Marketing Platform-APIs (unica-common.jar)

Wenn eine Prozedur oder die Sekundärklasse, die von der Prozedur importiert wird, solche Pakete verwendet, muss deren Verwendung exakt mit den Paketen übereinstimmen, die von Marketing Operations oder dem Anwendungsserver bereitgestellt werden. In diesem Fall ist eine Nachbearbeitung Ihres Prozedurcodes erforderlich, wenn eine spätere Version von Marketing Operations eine Bibliothek aktualisiert oder nicht weiter verwendet.

Prozeduren und Threads

Die Prozedur muss bezüglich ihres eigenen Status threadsicher sein, das heißt, ihre Ausführungsmethode darf nicht von internen Statusänderungen zwischen einzel-

nen Aufrufen abhängen. Eine Prozedur kann keine eigenen Threads erstellen.

Konfigurationsparameter

Bei der Installation von Marketing Operations Integration Module legt das Installationsprogramm drei Konfigurationseigenschaften fest. Sie können die Konfigurationseigenschaften ändern, um das Verhalten von Integration Module anzupassen.

Sie finden die Konfigurationseigenschaften für Integration Module unter **Marketing Operations | umoConfiguration | integrationServices**.

- Die Konfigurationseigenschaft **enableIntegrationServices** schaltet das Integration Services-Modul an und aus.
- Der Parameter **integrationProcedureDefinitionPath** enthält den vollständigen Dateipfad zur XML-Datei der benutzerdefinierten Prozedurdefinition.
Der Standardwert ist `<IBM_EMM_Home><MarketingOperations_Home>/devkits/integration/examples/src/procedure/procedure-plugins.xml/`.
- Der Parameter **integrationProcedureClasspathURL** enthält die URL zum Klassenpfad für benutzerdefinierte Prozeduren.
Der Standardwert ist `file:///<IBM_EMM_Home><MarketingOperations_Home>/devkits/integration/examples/classes/`.

Anmerkung: Das Zeichen "/" am Ende des Pfads `integrationProcedureClasspathURL` wird benötigt, um Prozedurklassen ordnungsgemäß zu laden.

Design

Die Prozedurimplementierungsklasse verwendet die IBM Marketing Operations-API zum Lesen und Aktualisieren von Marketing Operations-Komponenten, Aufrufen von Services und mehr. Andere Java-Pakete können zur Ausführung weiterer Aufgaben eingesetzt werden.

Konzentrieren Sie sich in Ihrem Design auf das Erstellen einer einzelnen Arbeitseinheit, die atomar bedient werden muss. Im Idealfall werden von einer Prozedur Serien von Aufgaben ausgeführt, die asynchron zur Ausführung zu einem späteren Zeitpunkt eingeplant werden können. Dieses "fire and forget"-Integrationsmodell sorgt für die geringste Last in beiden Systemen.

Anmerkung: Nur die dokumentierten Klassen und Methoden werden in zukünftigen Releases von Marketing Operations unterstützt. Betrachten Sie alle anderen Klassen und Methoden in Marketing Operations als nicht zulässig.

Nachdem Sie die Prozedurimplementierungsklassen codiert und kompiliert haben, stellen Sie diese Marketing Operations zur Verfügung. Die Erstellungsscripts, die zusammen mit Marketing Operations Integration Services bereitgestellt werden, legen die kompilierten Prozeduren unter der Standardposition ab. Der abschließende Entwicklungsschritt besteht in der Aktualisierung der Plug-in-Definitionsdatei für die benutzerdefinierte Prozedur, die von Marketing Operations zur Erkennung der benutzerdefinierten Prozeduren verwendet wird.

Die Prozedur muss die **com.unica.publicapi.plan.plugin.procedure.IProcedure**-Benutzeroberfläche implementieren und einen parameterlosen Konstruktor (normales JavaBeans-Modell) aufweisen. Die Codierung und Kompilierung der einzelnen Prozeduren erfolgt in einem beliebigen Java-Tool nach Wahl des Kunden, z. B. Ec-

lipse, Borland JBuilder oder Idea. Beispielcode wird mit IBM Marketing Operations in Form von Developer Toolkits an folgender Speicherposition zur Verfügung gestellt:

`<MarketingOperations_Home>/devkits/integration/examples/src/procedure`

Lebenszyklus von Prozeduren

Jede Prozedur durchläuft einen vollständigen Lebenszyklus.

Der Laufzeitlebenszyklus einer Prozedur umfasst die folgenden Schritte.

1. Erkennung und Initialisierung
2. Auswahl (optional)
3. Ausführung
4. Vernichtung

Erkennung und Initialisierung

Für IBM Marketing Operations müssen alle benutzerdefinierten und Standardprozeduren definiert werden, die für eine bestimmte Installationsinstanz zur Verfügung stehen. Dieser Prozess wird als Erkennung bezeichnet.

Anmerkung: Standardprozeduren (vom Marketing Operations-Entwicklungsteam definierte Prozeduren) sind implizit bekannt, darum sind keinerlei Aktionen zur Erkennung dieser Prozeduren erforderlich.

Benutzerdefinierte Prozeduren werden in der Plug-in-Definitionsdatei der Prozedur definiert. Der Plug-in-Manager von Marketing Operations liest diese Datei während der Initialisierung. Für jede erkannte Prozedur führt der Plug-in-Manager die folgenden Schritte aus.

1. Instanzieren der Prozedur, Überführung ihres Status in INSTANTIATED.
2. Erstellen eines Auditdatensatzes für die Prozedur.
3. Wenn die Prozedur instanziiert wurde, wird ihre **initialize()**-Methode mit allen Initialisierungsparametern aufgerufen, die in ihrer Plug-in-Beschreibungsdatei vorhanden sind. Wenn von dieser Methode eine Ausnahmebedingung ausgegeben wird, wird der Status protokolliert und die Prozedur abgebrochen. Andernfalls geht die Prozedur in den Status INITIALIZED über. Die Prozedur ist nun zur Ausführung bereit.
4. Erstellen eines Auditdatensatzes für die Prozedur.
5. Wenn die Prozedur initialisiert wurde, wird ihre **getKey()**-Methode aufgerufen, um den von Clients zum Verweis auf die Prozedur verwendeten Schlüssel zu bestimmen. Dieser Schlüssel wird der Instanz zugeordnet und für spätere Suchvorgänge gespeichert.

Auswahl

Gelegentlich zeigt IBM Marketing Operations dem Benutzer eine Liste der verfügbaren Prozeduren an, um Administratoren z. B. das Einrichten eines Triggers zu ermöglichen. Marketing Operations zeigt diese Liste erst an, nachdem die Prozedur unter Verwendung der Methoden **getDisplayName()** und **getDescription()** initialisiert wurde.

Ausführung

Nach der Initialisierung der Prozedur erhält IBM Marketing Operations eine Anforderung zur Ausführung der Prozedur. Dies kann zeitgleich mit weiteren Prozeduren (oder der gleichen Prozedur) in anderen Threads erfolgen.

Während der Laufzeit führt der Ausführungsmanager der Prozedur die folgenden Schritte aus.

1. Starten der Datenbanktransaktion.
2. Festlegen des Prozedurstatus auf EXECUTING.
3. Erstellen eines Auditdatensatzes für die Prozedur.
4. Aufrufen der **execute()**-Methode der Prozedur mit einem Ausführungskontext und allen Ausführungsparametern, die der Client zur Verfügung stellt. Bei der Methodenimplementierung wird bei Bedarf die Marketing Operations-API verwendet, um Bearbeitungssperren anzufordern und den Ausführungskontext weiterzugeben. Wenn die Ausführungsmethode eine Ausnahme auslöst, markiert der Ausführungsmanager die Transaktion für den Rollback.
5. Entsprechend den Ausführungsergebnissen Commit oder Rollback für die Transaktion durchführen und den Prozedurstatus auf EXECUTED setzen.
6. Jegliche ausstehende Bearbeitungssperren auflösen.
7. Erstellen eines Auditdatensatzes für die Prozedur.

Anmerkung: Die **execute()**-Methode ist nicht dazu gedacht, Prozedurinstanzdaten zu ändern.

Vernichtung

Wenn IBM Marketing Operations beendet wird, geht der Plug-in-Manager der Prozedur alle geladenen Prozeduren durch. Für jede erkannte Prozedur führt der Plug-in-Manager die folgenden Schritte aus.

1. Aufrufen der Destroy()-Methode der Prozedur, um der Prozedur eine Bereinigung zu ermöglichen, bevor die Instanz vernichtet wird.
2. Änderung des Status der Prozedur in FINALIZED (sie kann nicht ausgeführt werden).
3. Erstellen eines Auditdatensatzes für die Prozedur.
4. Vernichten der Instanz der Prozedur.

Wichtige Java-Klassen

Das bereitgestellte Integration-Development-Kit enthält einen Satz Javadoc-Informationen für die öffentlichen IBM Marketing Operations-API-Klassen und unterstützende Klassen.

Die wichtigsten Java-Klassen sind hier aufgelistet.

- **IProcedure** (com.unica.publicapi.plan.plugin.procedure.IProcedure): Schnittstelle, die von allen Prozeduren implementiert werden muss. Prozeduren durchlaufen einen klar strukturierten Lebenszyklus und greifen auf die Marketing Operations-API zu, um Arbeiten durchzuführen.
- **ITriggerProcedure** (com.unica.publicapi.plan.plugin.procedure.ITriggerProcedure): Schnittstelle, die von allen auslösenden Prozeduren implementiert werden muss (Markierungsschnittstelle).

- `IExecutionContext` (com.unica.publicapi.plan.plugin.procedure.IExecutionContext): Schnittstelle des nicht transparenten Kontextobjekts, das der Ausführungsmanager an die Prozedur übergeben hat. Dieses Objekt verfügt über öffentliche Methoden zur Protokollierung und zur Verwaltung von Bearbeitungssperren. Die Prozedur übergibt dieses Objekt auch an alle PlanAPI-Aufrufe.
- `IPlanAPI` (com.unica.publicapi.plan.api.IPlanAPI): Schnittstelle zur Marketing Operations-API. Der Ausführungskontext stellt eine **getPlanAPI()**-Methode zum Abrufen der ordnungsgemäßen Implementierung bereit.

Datensperre

IBM Marketing Operations verwendet ein pessimistisches Bearbeitungssperrensche-ma, das bedeutet, es erhält immer nur ein Benutzer den Aktualisierungszugriff auf eine Komponenteninstanz. Für den Benutzer der grafischen Benutzeroberfläche er-folgt diese Sperre auf der sichtbaren Benutzeroberfläche auf den Registerkarten. In manchen Fällen sind Daten eines Subsets einer Instanz gesperrt, beispielsweise eine Registerkarte mit Projektübersicht. In anderen Fällen sind Daten instanzen-übergreifend gesperrt, beispielsweise die Registerkarte "Workflow". Nachdem ein Benutzer eine Sperre angefordert hat, besteht für alle anderen Benutzer lediglich Lesezugriff auf die betreffenden Daten.

Um sicherzustellen, dass die von einer Prozedur an einer Komponenteninstanz oder einer Gruppe von Instanzen vorgenommenen Änderungen nicht versehentlich von einem anderen Benutzer überschrieben werden, muss eine Prozedur vor dem Aktualisieren von Komponentendaten die geeigneten Sperren anfordern. Zur Sperrung der Daten wird das an die **execute()**-Methode der Prozedur übergebene Ausführungskontextobjekt verwendet.

Bevor die Prozedur Daten aktualisiert, muss sie die **acquireLock()**-Methode des Kontexts für alle erforderlichen Sperren aufrufen. Wenn beispielsweise eine Proze-dur ein Projekt und den zugehörigen Workflow aktualisiert, muss sie für beide eine Sperre anfordern.

Wenn ein anderer Benutzer bereits eine Sperre angefordert hat, löst die **acquire-Lock()**-Methode unverzüglich die Ausnahme **LockInUseException** aus. Um die An-zahl an Kollisionen zu minimieren, muss die Prozedur die Sperre aufheben, sobald alle Aktualisierungen vorgenommen wurden.

Der Ausführungsmanager hebt automatisch alle ausstehenden Sperren auf, nach-dem die execute-Methode zurückgegeben wurde. Sperren bleiben grundsätzlich nur für die Lebensdauer der Datenbanktransaktion bestehen. Folglich verfallen Sperren, wenn das datenbankspezifische Transaktionszeitlimit überschritten ist.

Anmerkung: Bearbeitungssperren sind nicht das gleiche wie Datenbanktransaktio-nen.

Prozedurtransaktionen

Der Prozedurausführungsmanager schließt die Ausführung der Prozedur automa-tisch in eine Datenbanktransaktion ein, indem er entsprechend dem Ergebnis der Prozedurausführung ein Commit oder ein Rollback durchführt.

Durch das Einschließen der Prozedurausführung in eine Datenbanktransaktion wird sichergestellt, dass Aktualisierungen der IBM Marketing Operations-Daten-bank für andere Benutzer nicht sichtbar sind, bis das Commit erfolgt ist. Das Ein-schließen macht Aktualisierungen außerdem atomar.

Die Prozedurenschreibfunktion muss weiterhin die erforderlichen Bearbeitungssperren anfordern, um sicherzustellen, dass andere Benutzer keine Änderungen in die Datenbank schreiben können, bevor die Prozedurenausführung abgeschlossen ist.

Prozedurkommunikation

Die Methode **execute()** einer Prozedur gibt einen Ganzzahlstatuscode an die IBM Marketing Operations-Audittabelle der Prozedur zurück. Die Methode **execute()** einer Prozedur kann außerdem null oder mehrere Nachrichten an die Audittabelle der Prozedur zurückgeben, die protokolliert und gespeichert werden.

Der Client kann die Statusinformationen auch auf andere Weise kommunizieren.

Prozedurprotokollierung

IBM Marketing Operations verfügt über eine separate Protokolldatei für Prozeduren: `<MarketingOperations_Home>\logs\procedure.log`

Der Prozedurenausführungsmanager protokolliert den Lebenszyklus jeder Prozedur und erstellt Prüfdatensätze.

- **logInfo()**: eine Informationsnachricht in das Prozedurprotokoll schreiben.
- **logWarning()**: einen Warnhinweis in das Prozedurprotokoll schreiben.
- **logError()**: eine Fehlernachricht in das Prozedurprotokoll schreiben.
- **logException()**: Speicherauszug des Stack-Trace für die Ausnahme im Prozedurprotokoll erstellen.

Prozedur-Plug-in-Definitiondatei

Die Plug-in-Definitiondatei für die Prozedur definiert die Implementierungsklasse, die Metadaten und weitere Informationen zu den benutzerdefinierten Prozeduren, die in IBM Marketing Operations gehostet werden.

Die Plug-in-Definitiondatei für eine Prozedur wird standardmäßig in folgendem Verzeichnis abgelegt:

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

Diese Datei ist ein XML-Dokument, das die nachfolgend beschriebenen Informationen enthält.

Prozeduren: eine Liste von null oder mehr **Procedure**-Elementen.

Prozedur: Ein Element, das eine Prozedur definiert. Jede Prozedur enthält die folgenden Elemente.

- **key** (optional): Zeichenfolge, die den Suchschlüssel für die Prozedur definiert. Dieser Schlüssel muss für alle Standardprozeduren (durch IBM bereitgestellt) sowie benutzerdefinierte Prozeduren, die von einer bestimmten Marketing Operations-Instanz per Hosting bereitgestellt werden, eindeutig sein. Wird kein Schlüssel definiert, dient automatisch die vollständig qualifizierte Version des **className**-Elements als Schlüssel. Namen, die mit der Zeichenfolge "uap" beginnen, sind zur Verwendung durch IBM Marketing Operations reserviert.

- **className** (erforderlich): vollständig qualifizierter Paketname der Prozedurklasse. Diese Klasse muss die IProcedure-Klasse implementieren (com.unica.public.plan.plugin.procedure.IProcedure).
- **initParameters** (optional): eine Liste aus null oder mehr initParameter-Elementen.
 - initParameter** (optional): Parameter, der an die Methode "initialize()" der Prozedur übergeben werden soll. Dieses Element umfasst den Namen des verschachtelten Parameters sowie seinen Typ und die Wertelemente.
 - name: Zeichenfolge, die den Parameternamen definiert
 - type: optionaler Klassenname der Java-Wrapperklasse, die den Typ des Parameterwerts definiert. Die folgenden Typen sind zulässig:
 - java.lang.String (Standardwert)
 - java.lang.Integer
 - java.lang.Double
 - java.lang.Calendar
 - java.lang.Boolean
 - value: Zeichenfolgenformat des Attributtyps entsprechend dem Typ

Kapitel 4. IBM Marketing Operations-API

Die IBM Marketing Operations-API ist eine Art Kulisse, auf der eine Client-Ansicht einer laufenden Marketing Operations-Instanz bereitgestellt wird.

Benutzern wird nur eine Untergruppe des Marketing Operations-Leistungsspektrums angezeigt. Die API wird gleichzeitig von Marketing Operations-Webbenutzern sowie von SOAP-Anforderungen und Triggern des Web-Service Marketing Operations Integration Services verwendet. Die API unterstützt die folgenden Operationstypen.

- Komponentenerstellung und Löschen von Komponenten
- Erkennung (nach Komponententyp, Attributwert und zusätzlichen Werten)
- Komponentenprüfung (anhand der Attribute, spezialisierter Links und anderer Werte)
- Komponentenbearbeitung

Inhalt des IBM Marketing Operations API

Das `com.unica.publicapi.plan.api`-Paket stellt das IBM Marketing Operations-API bereit.

Dieses Paket bietet Schnittstellen und Ausnahmen und enthält folgende Typen für Klassen:

- aufgelistete Datentypen
- Handles zum Ermitteln von Objekt- und Komponenteninstanzen
- eine Java-Map, `AttributeMap`

Sie können die vollständige Dokumentation der API einschließlich aller Methoden und möglicher Werte aufrufen, indem Sie auf **Hilfe > Produktdokumentation** in einer Instanz von Marketing Operations klicken und anschließend die Datei `IBM <version>PublicAPI.zip` herunterladen.

API-Schnittstellen

Die API von IBM Marketing Operations umfasst **IPlanAPI** und **IExecutionContext**.

Die API von Marketing Operations umfasst die folgenden Schnittstellen.

IPlanAPI

Definiert das öffentliche API für Marketing Operations. Stellt Methoden zum Erstellen, Erkennen und Ändern von Objekten bereit, einschließlich Ordern, Projekten, Programmen, Workflowaufgaben und Teammitgliedern.

Für Systeme, bei denen die optionale Integration in IBM Campaign aktiviert ist, werden auch Methoden zum Erstellen, Erkennen und Ändern von Angeboten zur Verfügung gestellt.

IExecutionContext

Definiert die Auslöser und sperrt die Ausführungsmethoden im API.

API-Methoden

Ausführliche Informationen über die öffentlichen API-Methoden finden Sie in der iPlanAPI-Klasse in den API-Dokumentationsdateien der JavaDocs.

Sie können auf diese Dateien zugreifen, indem Sie sich bei Marketing Operations anmelden, auf einer beliebigen Seite **Hilfe > Produktdokumentation** auswählen und dann die Datei `<version>PublicAPI.zip` herunterladen.

Häufig auftretende Ausnahmen

Zu den häufigen Ausnahmen, die von der API ausgelöst werden, gehören NotFoundException, AuthorizationException, DataException, InvalidExecutionContextException und NotLockedException.

In der folgenden Liste wird erklärt, warum diese häufigen Ausnahmen auftreten können.

- `<object type>NotFoundException`: Das System kann das angegebene Element oder Objekt nicht liefern.
- `AuthorizationException`: Der Benutzer, der dem Ausführungskontext zugeordnet ist, ist nicht berechtigt, die angeforderte Operation auszuführen. Diese Ausnahme kann von einer beliebigen API-Methode ausgelöst werden, sie ist also nicht deklariert.
- `DataException`: In der zugrundeliegenden Datenbankebene in IBM Marketing Operations ist eine Ausnahme aufgetreten. Details können Sie dem SQL-Protokoll entnehmen.
- `InvalidExecutionContextException`: Es ist ein Problem mit einem Ausführungskontext aufgetreten, der an eine API-Methode übergeben wurde (beispielsweise wurde die Methode nicht richtig initialisiert). Diese Ausnahme kann von einem beliebigen API ausgelöst werden, sie ist also nicht deklariert.
- `NotLockedException`: Versuch der Aktualisierung von Komponentendaten, ohne vorher die erforderliche Sperre angefordert zu haben. Siehe `acquireLock()`-Methode der Schnittstelle "IExecutionContext".

Handles

Ein Handle ist ein spezielles URL-Objekt, das innerhalb einer IBM Marketing Operations-Instanz auf eine bestimmte Objektinstanz verweist. Handles umfassen den Komponententyp, IDs von internen Daten und die Basis-URL einer Instanz.

Handles, die von dem API verwendet oder generiert werden, können zu einer vollständigen URL externalisiert werden. Sie können die resultierende URL auf verschiedene Arten verwenden. Sie können die URL verwenden, um eine Ansicht der Komponente auf der grafischen Benutzeroberfläche von Marketing Operations zu öffnen, um diese Ansicht als E-Mail-Nachrichten zu senden oder um sie in einer anderen Prozedur als Parameter einzusetzen.

Handles sind lediglich für eine bestimmte Marketing Operations-Serviceinstanz oder -Clusterinstanz gültig, sie sind aber immer für die Laufzeit des bereitgestellten Service gültig. Das bedeutet, Handles können zur späteren Verwendung in einer Datei gespeichert werden, sie können jedoch nicht für den Zugriff auf Komponenten einer anderen Instanz von Marketing Operations verwendet werden. Diese Einschränkung gilt auch für Instanzen auf demselben physischen Host-Server. Marketing Operations stellt einen Mechanismus zur Verfügung, mit dem verschiedene Basis-URLs der aktuellen Instanz zugeordnet werden können, um das Verlagern einer Instanz zu einem anderen Server zu ermöglichen (etwa bei Störungen im System).

Handles sind clientunabhängig. Beispielsweise kann ein Trigger ein Handle an eine Prozedur übergeben, die diesen als Parameter für einen SOAP-Aufruf an ein Drittanbietersystem verwendet. Dieses sendet daraufhin eine SOAP-Anforderung an Marketing Operations zurück, um eine Prozedur aufzurufen, die ein Attribut aktualisiert.

Die Mitglieder der Handle-Klasse verfügen über Factory-Methoden zum Erstellen von Handles aus verschiedenen Typen von URLs. Nachfolgend sind Beispiele aufgeführt.

Genehmigung

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

Asset

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

Assetordner

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

Assetbibliothek

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

Anhang

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

Finanzkonto

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

Ordner

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

Rechnung

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

Rechnungsposition

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

Marketingobjekt

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

Marketingobjektraster

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

Marketingobjekt-Rasterzeile

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

Planungsteam

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

Planbenutzer

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

Programm

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

Programmraster

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

Programmrasterzeile

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

Projekt

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

Projektraster

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

Projektrasterzeile

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

Projektposition

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

Workflowstufe

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

Workflowaufgabe

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

Attributzuordnung (AttributeMap)

Die Klasse `AttributeMap` ist eine Java-Map, die ausschließlich Attribute enthält. Das Attribut `<Name>` ist der Schlüssel des Zuordnungseintrags, und das Array des Attributs `<values>` (Plural!) ist der Wert des Zuordnungseintrags.

Die Klasse `AttributeMap` enthält die folgenden Felder.

- `<Name>`: der programmorientierte Name des Attributs. Dieser Name dient als eindeutiger Schlüssel für den Zugriff auf das Attribut innerhalb der Komponenteninstanz, in der es vorkommt.

Anmerkung: Dieser `<Name>` entspricht nicht unbedingt dem Anzeigenamen, der auf der grafischen Benutzeroberfläche erscheint. Der Attributname von Komponenten, die mithilfe einer Vorlage erstellt wurden (z. B. Projekte oder Workflowaufgaben), wird in der Vorlagenelementdefinition festgelegt und muss eindeutig sein. Bei anderen Komponenten wird der Attributname in der Regel programmgesteuert von der serverseitigen Komponenteninstanz abgeleitet (z. B. durch Java-Introspektion).

Anmerkung: Gemäß der Konvention umfassen kundenspezifische Attribute den Namen des Formats, in dem die bearbeitbare Version definiert ist: `<form_name>.<attribute_name>`.

- `Values`: ein Java-Objekt-Array mit null oder mehr Attributwerten. Die einzelnen Werte müssen den gleichen Typ aufweisen und dem Typ des Attributs entsprechend, der in Marketing Operations definiert ist. Es werden ausschließlich die folgenden Java-Wrapper und Marketing Operations-Typen unterstützt:
 - `AssetLibraryStateEnum`: ein `AssetLibraryStateEnum`-Aufzählungstypwert.
 - `AssetStateEnum`: ein `AssetStateEnum`-Aufzählungstypwert.
 - `AttachmentTypeEnum`: ein `AttachmentTypeEnum`-Aufzählungstypwert.

- `AttributeMap`: eine Zuweisung, die Attribute enthält.
- `BudgetPeriodEnum`: ein `BudgetPeriodEnum`-Aufzählungstypwert.
- `BudgetTypeEnum`: ein `BudgetTypeEnum`-Aufzählungstypwert.
- `Handle`: ein Verweis auf eine Komponenteninstanz, Rasterzeile, ein Attribut usw.
- `InvoiceStateEnum`: ein `InvoiceStateEnum`-Aufzählungstypwert.
- `java.io.File`: Darstellung einer Datei.
- `java.lang.Boolean`: ein boolescher Wert, entweder "wahr" oder "falsch".
- `java.lang.Double`: ein Dezimalzahlenwert mit doppelter Genauigkeit.
- `java.lang.Float`: ein Dezimalzahlenwert mit einfacher Genauigkeit.
- `java.lang.Integer`: ein 32-Bit-Ganzzahlwert.
- `java.lang.Long`: ein 64-Bit Ganzzahlwert.
- `java.lang.Object`: generisches Java-Objekt.
- `java.lang.String`: eine Zeichenfolge aus null oder mehr Unicode-Zeichen.
- `java.math.BigDecimal`: Dezimalzahlenwert mit beliebiger Genauigkeit. Geeignet für Währung; die Interpretation des Wertes hängt von der Ländereinstellung für die Währung des Clients ab.
- `java.math.BigInteger`: Ganzzahlwert mit beliebiger Genauigkeit.
- `java.net.URL`: ein URL (Universal Resource Locator)-Objekt.
- `java.util.ArrayList`: Liste von Objekten.
- `java.util.Calendar`: ein Datum/Uhrzeit-Wert für eine bestimmte Ländereinstellung.
- `java.util.Date`: ein Datum/Uhrzeit-Wert. Dieser Typ wird nicht weiter unterstützt. Verwenden Sie stattdessen "java.util.Calendar" oder "java.util.GregorianCalendar".

Anmerkung: Zur Implementierung des Datums stehen dem Benutzer "java.util.Calendar" oder "java.util.GregorianCalendar" zur Verfügung.

- `/java.util.GregorianCalendar`: "GregorianCalendar" ist eine konkrete Unterklasse von "java.util.Calendar" und stellt das Standardkalendersystem bereit, dass in den meisten Ländern verwendet wird.
- `MonthEnum`: ein `MonthEnum`-Aufzählungstypwert.
- `ProjectStateEnum`: ein `ProjectStateEnum`-Aufzählungstypwert.
- `QuarterEnum`: ein `QuarterEnum`-Aufzählungstypwert.
- `TaskStateEnum`: ein `TaskStateEnum`-Aufzählungstypwert.
- `WeekEnum`: ein `WeekEnum`-Aufzählungstypwert.

Die Metadaten eines Attributs (wie übersetzter Anzeigenname und Beschreibung) werden von der Vorlage, die dem Attribut zugeordnet ist, und dessen übergeordneter Objektinstanz definiert. Attribute stellen einen einfachen, dabei aber erweiterbaren Mechanismus zum Anzeigen von erforderlichen und optionalen Objektinstanzattributen bereit, beispielsweise Projektname, Code und Startdatum.

Aufgelistete Datentypen

Die IBM Marketing Operations-API unterstützt die folgenden aufgelisteten Datentypen und Werte.

ApprovalMethodEnum

`ApprovalMethodEnum` definiert gültige Genehmigungsmethoden. Folgende Werte sind möglich:

- SEQUENTIAL
- SIMULTANEOUS

ApprovalStateEnum

ApprovalStateEnum definiert gültige Genehmigungsstatus. Folgende Werte sind möglich:

- CANCELLED
- COMPLETED
- IN_PROGRESS
- NOT_STATED
- ON_HOLD

AssetLibraryStateEnum

AssetLibraryStateEnum definiert gültige Assetbibliotheksstatus. Folgende Werte sind möglich:

- DISABLED
- ENABLED

AssetStateEnum

AssetStateEnum definiert gültige Assetstatus. Folgende Werte sind möglich:

- ARCHIVE
- ENTWURF
- FINALIZE
- LOCK

AttachmentTypeEnum

AttachmentTypeEnum definiert gültige Anlagentypen. Folgende Werte sind möglich:

- ASSET
- FILE
- URL

BudgetPeriodEnum

BudgetPeriodEnum definiert die möglichen Planungsperioden. Folgende Werte sind möglich:

- ALL
- MONTHLY
- QUARTERLY
- WEEKLY
- YEARLY

BudgetTypeEnum

BudgetTypeEnum definiert gültige Budgettypen. Folgende Werte sind möglich:

- ACTUAL
- ALLOCATED
- COMMITTED
- FORECAST
- TOTAL

ComponentTypeEnum

ComponentTypeEnum ermittelt die zugänglichen Marketing Operations-Komponententypen. Folgende Werte sind möglich:

- APPROVAL
- ASSET
- ASSET_FOLDER
- ASSET_LIBRARY
- ANHANG
- FINANCIAL_ACCOUNT
- GROUPING_FOLDER
- INVOICE
- MARKETING_OBJECT
- PLAN_TEAM
- PLAN_USER
- PROGRAM
- PROJECT
- PROJECT_REQUEST
- TASK
-

InvoiceStateEnum

InvoiceStateEnum definiert gültige Rechnungsstatus. Folgende Werte sind möglich:

- CANCELLED
- ENTWURF
- PAID
- PAYABLE

MonthEnum

MonthEnum definiert gültige Werte für den Monat.

OfferStateEnum

OfferStateEnum definiert gültige Angebotsstatus. Folgende Werte sind möglich:

- STATE_OFFER_DRAFT
- STATE_OFFER_PUBLISHED
- STATE_OFFER_RETIRED

ProjectCopyTypeEnum

ProjectCopyTypeEnum definiert gültige Methoden zum Kopieren eines Projekts. Folgende Werte sind möglich:

- COPY_USING_PROJECT_METRICS
- COPY_USING_TEMPLATES_METRICS

ProjectParticipantLevelEnum

ProjectParticipantLevelEnum gibt die Rollen an, die die Benutzer innerhalb eines Projekts haben. Folgende Werte sind möglich:

- OWNER
- PARTICIPANT
- REQUESTER

ProjectStateEnum

ProjectStateEnum definiert gültige Projekt- und Anforderungsstatus. Folgende Werte sind möglich:

- ACCEPTED

- CANCELLED
- COMPLETED
- ENTWURF
- IN_PROGRESS
- IN_RECONCILIATION
- LATE: das Projekt wurde nicht zum geplanten Anfangsdatum gestartet.
- NOT_STARTED
- ON_HOLD
- OVERDUE: das Projekt wurde nicht vor dem geplanten Enddatum abgeschlossen.
- RETURNED
- SUBMITTED

Weitere Informationen über die Projekt- und Aufgabenstatus finden Sie im *IBM Marketing Operations-Benutzerhandbuch*.

QuarterEnum

QuarterEnum legt die gültigen Werte für die Quartale fest: Q1, Q2, Q3 und Q4.

TaskStateEnum

TaskStateEnum definiert gültige Workflowaufgaben-Status. Folgende Werte sind möglich:

- ACTIVE
- DISABLED
- FINISHED
- PENDING
- SKIPPED

WeekEnum

WeekEnum definiert gültige Werte für die Wochen eines Jahres, von WEEK_1 bis WEEK_53.

Kontaktaufnahme zum technischen Support von IBM

Sollte sich ein Problem nicht mithilfe der Dokumentation beheben lassen, können sich die für den Support zuständigen Kontaktpersonen Ihres Unternehmens telefonisch an den technischen Support von IBM wenden. Damit Ihr Problem möglichst schnell und erfolgreich gelöst wird, sollten Sie vor Tätigung des Anrufs entsprechende Informationen zusammenstellen.

Wenn Sie wissen möchten, wer die für den Support zuständige Kontaktperson Ihres Unternehmens ist, wenden Sie sich an Ihren IBM Administrator.

Zusammenzustellende Informationen

Halten Sie folgende Informationen bereit, wenn Sie sich an den technischen Support von IBM wenden:

- Kurze Beschreibung der Art Ihres Problems
- Detaillierte Fehlermeldungen, die beim Auftreten des Problems angezeigt werden.
- Schritte zum Reproduzieren des Problems
- Entsprechende Protokolldateien, Sitzungsdateien, Konfigurationsdateien und Daten
- Informationen zu Ihrer Produkt- und Systemumgebung, die Sie entsprechend der Beschreibung unter "Systeminformationen" abrufen können.

Systeminformationen

Bei Ihrem Anruf beim technischen Support von IBM werden Sie um verschiedene Informationen gebeten.

Sofern das Problem Sie nicht an der Anmeldung hindert, finden Sie einen Großteil der benötigten Daten auf der Info-Seite. Dort erhalten Sie Informationen zur IBM Anwendung.

Sie können über **Hilfe > Info** auf die Info-Seite zugreifen. Wenn Sie nicht auf die Info-Seite zugreifen können, können Sie die Versionsnummer einer beliebigen IBM Anwendung abrufen, indem Sie die Datei `version.txt` anzeigen, die sich im Installationsverzeichnis jeder Anwendung befindet.

Kontaktinformationen für den technischen Support von IBM

Wenn Sie sich an den technischen Support von IBM wenden möchten, finden Sie weitere Informationen auf der Website des technischen Supports für IBM Produkte (http://www.ibm.com/support/entry/portal/open_service_request).

Anmerkung: Um eine Supportanforderung einzugeben, müssen Sie sich mit einem IBM Account anmelden. Dieses Konto muss, wenn möglich, mit Ihrer IBM Kundennummer verknüpft sein. Weitere Informationen zum Zuordnen Ihres Kontos zu Ihrer IBM Kundennummer finden Sie unter **Unterstützungsressourcen > Gültige Softwareunterstützung** im Support-Portal.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für die in diesem Handbuch beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation
B1WA LKG1

550 King Street
Littleton, MA 01460-1250
USA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können unter Umständen von den hier genannten Preisen abweichen.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielpprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielpprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter www.ibm.com/legal/copytrade.shtml.

Hinweise zu Datenschutzrichtlinien und Nutzungsbedingungen

IBM Softwareprodukte, einschließlich Software as a Service-Lösungen ("Softwareangebote"), können Cookies oder andere Technologien verwenden, um Informationen zur Produktnutzung zu erfassen, die Endbenutzererfahrung zu verbessern und Interaktionen mit dem Endbenutzer anzupassen oder zu anderen Zwecken. Ein Cookie ist ein Datenelement, das von einer Website an Ihren Browser gesendet wird und dann als Tag auf Ihrem Computer gespeichert werden kann, mit dem Ihr Computer identifiziert wird. In vielen Fällen werden von diesen Cookies keine personenbezogenen Daten erfasst. Wenn ein Softwareangebot, das von Ihnen verwendet wird, die Erfassung personenbezogener Daten anhand von Cookies und ähnlichen Technologien ermöglicht, werden Sie im Folgenden über die hierbei geltenden Besonderheiten informiert.

Abhängig von den bereitgestellten Konfigurationen kann dieses Softwareangebot Sitzungscookies und permanente Cookies verwenden, mit denen der Benutzername des Benutzers und andere personenbezogene Daten zum Zwecke des Sitzungsmanagements, zur Verbesserung der Benutzerfreundlichkeit und zu anderen funktionsbezogenen Zwecken sowie zur Nutzungsüberwachung erfasst werden. Diese Cookies können deaktiviert werden. Durch die Deaktivierung kann jedoch auch die von ihnen bereitgestellte Funktionalität nicht mehr genutzt werden.

Die Erfassung personenbezogener Daten mithilfe von Cookies und ähnlichen Technologien wird durch verschiedene rechtliche Bestimmungen geregelt. Wenn die für dieses Softwareangebot implementierten Konfigurationen Ihnen als Kunde die Möglichkeit bieten, personenbezogene Daten von Endbenutzern über Cookies und andere Technologien zu erfassen, dann sollten Sie ggf. juristische Beratung zu den geltenden Gesetzen für eine solche Datenerfassung in Anspruch nehmen. Dies gilt auch in Bezug auf die Anforderungen, die vom Gesetzgeber in Bezug auf Hinweise und die Einholung von Einwilligungen vorgeschrieben werden.

IBM setzt voraus, dass Kunden folgende Bedingungen erfüllen: (1) Sie stellen einen klar erkennbaren und auffälligen Link zu den Nutzungsbedingungen der Kundenwebsite (z. B. Datenschutzerklärung) bereit. Dieser Link muss wiederum einen Link zu der Vorgehensweise von IBM und des Kunden bei der Datenerhebung und Datennutzung umfassen. (2) Sie weisen darauf hin, dass Cookies und Clear GIFs/Web-Beacons von IBM im Auftrag des Kunden auf dem Computer des Besuchers platziert werden. Dieser Hinweis muss eine Erläuterung hinsichtlich des Zwecks dieser Technologie umfassen. (3) Sie müssen in dem gesetzlich vorgeschriebenen Umfang die Einwilligung von Websitebesuchern einholen, bevor Cookies und Clear GIFs/Web-Beacons vom Kunden oder von IBM im Auftrag des Kunden auf den Geräten der Websitebesucher platziert werden.

Weitere Informationen zur Verwendung verschiedener Technologien einschließlich der Verwendung von Cookies zu diesen Zwecken finden Sie im IBM Online Privacy Statement unter der Webadresse <http://www.ibm.com/privacy/details/us/en> im Abschnitt mit dem Titel "Cookies, Web Beacons and Other Technologies".

