

**IBM Marketing Operations**

**V9R1**

**2013 年 10 月 25 日**

**集成模块**

**IBM**

**注意**

在使用本信息及其支持的产品前，请先阅读第 31 页的『声明』中的信息。

---

# 目录

<b>第 1 章 什么是 IBM Marketing Operations Integration Services? . . . . .</b>	<b>1</b>
Marketing Operations Integration Services 的要求有哪些? . . . . .	2
IBM Marketing Operations Integration Services 基本. . . . .	3
安装 Integration Services . . . . .	5
软件开发人员工具包内容 . . . . .	5
托管的 Javadoc. . . . .	6
有关更多信息 . . . . .	6
<b>第 2 章 Marketing Operations 集成 Webservice . . . . .</b>	<b>9</b>
Marketing Operations Integration Services WSDL . . . . .	9
executeProcedure . . . . .	9
Marketing Operations 集成 Webservice 数据类型 . . . . .	10
<b>第 3 章 IBM Marketing Operations 过程 . . . . .</b>	<b>13</b>
假设 . . . . .	13
配置参数 . . . . .	14
设计 . . . . .	15

过程生命周期 . . . . .	15
关键 Java 类 . . . . .	17
数据锁定 . . . . .	17
过程事务 . . . . .	17
过程通信 . . . . .	18
过程日志记录 . . . . .	18
过程插件定义文件 . . . . .	18
<b>第 4 章 IBM Marketing Operations API 21</b>	<b>21</b>
IBM Marketing Operations API 的内容 . . . . .	21
API 接口 . . . . .	21
常见异常 . . . . .	22
句柄 . . . . .	22
属性映射 . . . . .	24
枚举数据类型 . . . . .	25
<b>与 IBM 技术支持联系之前 . . . . .</b>	<b>29</b>
<b>声明 . . . . .</b>	<b>31</b>
商标信息 . . . . .	32
隐私策略和使用条款注意事项 . . . . .	33



---

# 第 1 章 什么是 IBM Marketing Operations Integration Services?

IBM® Marketing Operations Integration Services 将 Marketing Operations Integration Webservice、API 过程和触发器组合在一起以扩展业务功能。

IBM Marketing Operations Integration Services 是以下对象的组合体。

- **Marketing Operations 集成 Webservice**

Integration Services 提供了一种方式，使 Marketing Operations 客户和 IBM Professional Services 可以将 Marketing Operations 与环境中运行的其他应用程序集成。

- **Marketing Operations 过程和 API**

可以在 Marketing Operations 中定义定制过程，以通过任意方式扩展 Marketing Operations 业务逻辑。定义过程后，这些过程就可以作为从其他应用程序进行 Integration Services Webservice 调用的目标。还可以定义过程来将消息发送至其他应用程序。

- **Marketing Operations 触发器**

可以将触发器与 Marketing Operations 中的事件和过程关联。发生某个此类事件时，将运行关联的触发器。

## 版本和向后兼容性

Integration Services 的将来版本将与共享同一主版本号的所有次要版本和维护版向后兼容。但是，对于点零 (x.0) 主要版本的较早版本，如果有业务理由或技术理由保证，那么 IBM 将保留中断与该版本兼容性的权限。

如果进行了任何以下更改，那么此 API 的主要版本号将递增。

- 数据解释更改
- 业务逻辑更改（例如，服务方法功能更改）
- 方法参数和/或返回类型更改

如果进行了任何以下更改，那么此 API 的次要版本号将递增。这些更改适用于较低版本（按定义）。

- 添加了新方法
- 添加了新数据类型并且其用途限于某个新方法
- 向枚举类型添加了新元素
- 使用版本后缀定义了接口的新版本

## 认证

无需认证；所有客户机都与名为 PlanAPIUse 的已知 IBM Marketing Operations 用户关联。系统管理员将配置此特殊用户的安全功能，以满足所有 Webservice 客户机的需求。

## 语言环境

支持的唯一语言环境是当前为 IBM Marketing Operations 系统实例配置的语言环境。假定所有语言环境相关数据（例如，消息和货币）都采用系统语言环境。

## 状态管理

API 和 webservice 无状态；即跨 API 调用的服务实现不会保存每个客户机的信息。次能提供更有效率的服务实现，并且简化了集群支持。

## 数据库事务

Marketing Operations Integration Services 不会向客户机显示数据库事务，但如果执行上下文中包含此类信息，那么将使用这些信息。如果已启动事务，那么特定过程中所有 API 调用的作用将采用原子方式。换句话说，失败的 API 调用将使数据库仍处于与从未调用过 API 一样的状态。在过程成功完成事务之前，Marketing Operations 的其他用户不会看到更改。

用来更新数据库的 API 调用必须首先获取编辑锁定，以防止其他用户在该 API 调用过程中修改底层数据。其他用户在 API 调用完成之前无法更新锁定的组件。同样，下一个 Marketing Operations 用户或 API 客户机必须获取数据锁定，然后才能提交另一个 API 调用。

## 事件处理

通过 API 在 IBM Marketing Operations 组件上执行的操作将如同 Marketing Operations 用户执行操作一样生成相同的事件。订阅特定通知（例如，项目状态更改）的用户将收到由 API 调用以及用户操作而引起的状态更改通知。

---

## Marketing Operations Integration Services 的要求有哪些？

Marketing Operations Integration Services 必须满足以下要求：

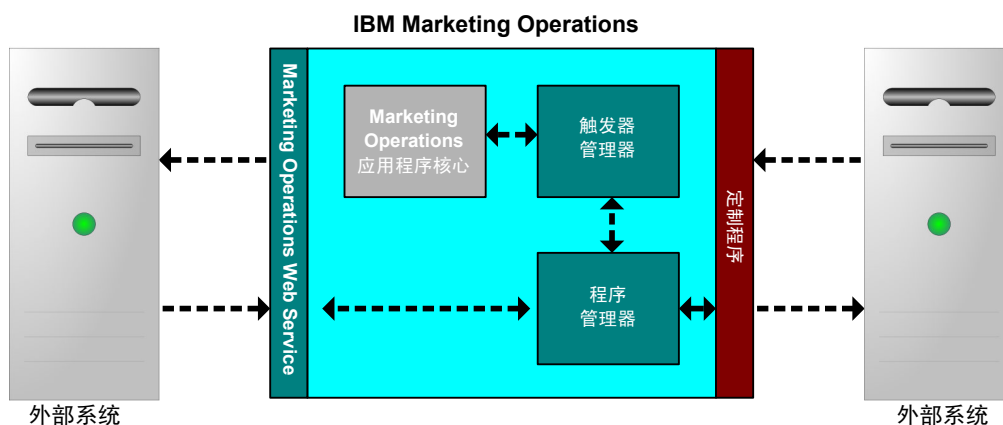
- 以松散方式进行系统集成的耦合。
- 提供一种机制，客户机应用程序可使用此机制通过 Webservice 调用来影响 Marketing Operations。
- 提供一种机制，可使用此机制在 Marketing Operations 中向客户应用程序通知某些事件。
- 提供易于理解和使用的简单编程模型。
- 具有强大的故障恢复功能
- 保证数据完整性。
- 与基于 GUI 的现有 Marketing Operations 客户集成，并将对这些客户的影响降至最低。
- 对 Marketing Operations 组件提供细颗粒度的访问，同时使编程人员免于接触底层实现详细信息。

## IBM Marketing Operations Integration Services 基本

可使用 IBM Marketing Operations Integration Services 来创建定制过程。当 Marketing Operations 中发生特定事件时，可以使用这些过程来触发外部事件。可以使用这些过程来从外部系统或程序运行 Marketing Operations 功能。

API 接口与 IBM Marketing Operations 在编程级别上进行交互，正如 GUI 接口与 Marketing Operations 在用户级别上进行交互一样。使用 API 可以构建过程。可以使用这些过程在 Marketing Operations 与外部系统之间通信。Marketing Operations Webservice 是过程、API 和触发器的容器对象。

以下显示了 Marketing Operations Integration Services 的体系结构。

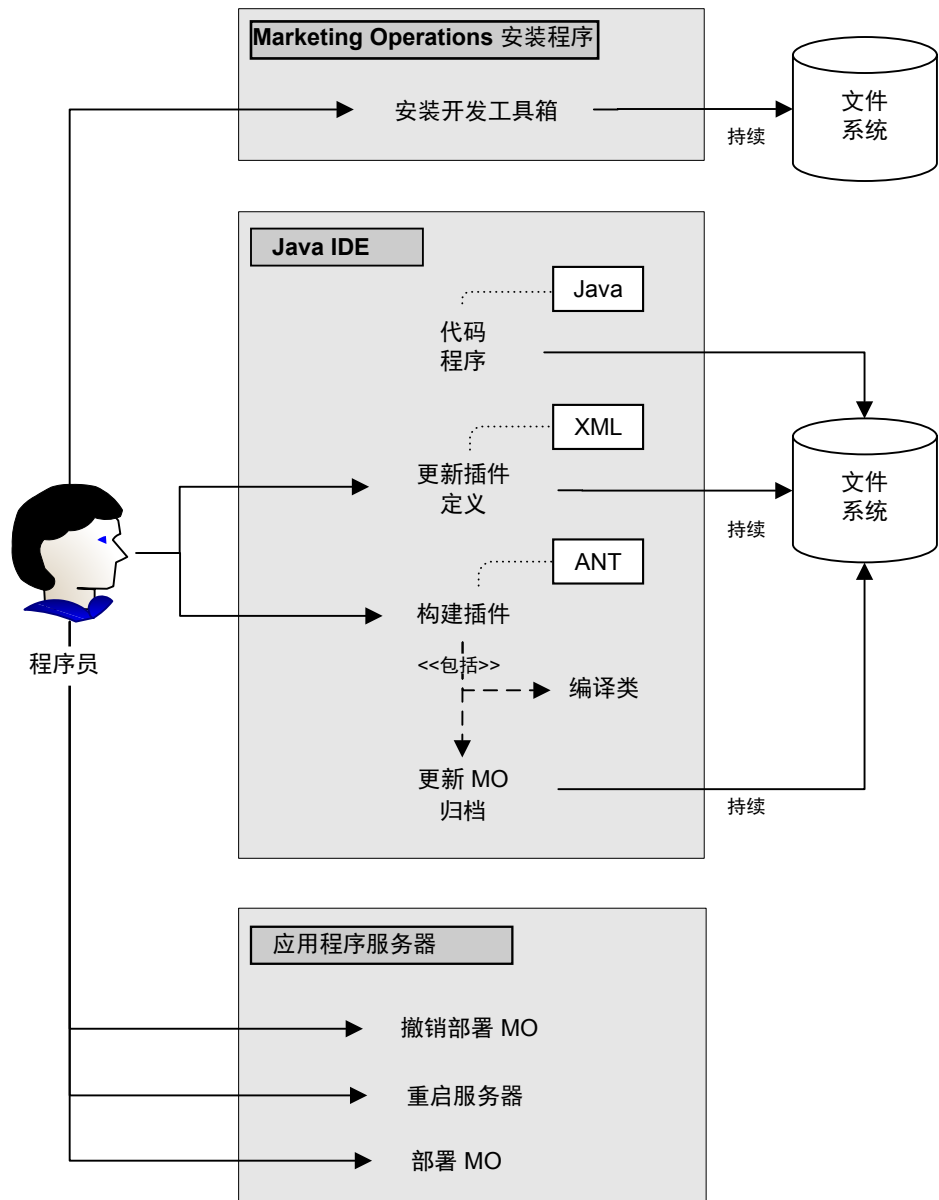


以下是 Integration Services 的主要组件。

- Marketing Operations 过程管理器：通过借助 API 与 Marketing Operations 交互，扩展业务逻辑。
- Marketing Operations 触发器管理器：将条件（例如市场营销对象的状态更改）与操作（当满足触发器的条件时要运行的过程）关联。

### 方法

使用 IBM Marketing Operations Integration Services 的组件来开发定制过程，如下图中所示：



安装开发人员工具包之后，请遵循以下基本步骤：

1. 对定制过程进行编码。当前必须使用 Java™。
2. 更新 XML 定义文件中的插件定义。
3. 构建插件：
  - a. 编译必要的类。
  - b. 如果您正在使用不在 Marketing Operations 归档中的第三方库，请将库捆绑在 plan.war 文件内并重新部署。
4. 重新启动 Marketing Operations。当您重新启动应用程序服务器时，会应用对过程类所做的更改。

**注：**如果您更改 **plan.war** 文件，那么必须取消部署然后使用新的 **plan.war** 文件重新部署 Marketing Operations。如果您使用不在 Marketing Operations 归档中的第三方库并编辑 **plan.war** 文件，请取消部署然后重新部署 Marketing Operations。



## 在 IBM Marketing Operations 与 API 之间通信的基本示例

以下基本示例描述在 API 与 Marketing Operations 之间建立通信。它并不执行任何有用的工作；它将执行 Marketing Operations 与 Integration Services 之间的往返通信。

此示例使用 Marketing Operations Integration Services 开发人员工具包随附的部分示例过程。具体来说，可以在以下文件中找到此处引用的代码。

- PlanClientFacade.java
- PlanWSNOOPTestCase.java

noop 方法是对 Marketing Operations 的 Webservice 调用。它在 PlanClientFacade 类中定义，并在数组中传递空值。

```
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);
}
```

过程 testExecuteProcedure 从 PlanClientFacade 调用 noop 方法，以建立与 Marketing Operations 应用程序的往返通信。

```
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.");
}
```

有关 NameValueArrays、ProcedureResponse 以及其他列出的方法和数据类型的详细信息，请参阅《Marketing Operations 集成模块》和 JavaDocs。

## 安装 Integration Services

IBM Marketing Operations Integration Services 模块是一种独立的付费组件。如果您购买 Integration Services 模块，那么必须对其进行安装。

1. 下载 IBM Marketing Operations Integration Services 安装程序。
2. IBM EMM 安装程序将检测 Integration Services 模块。
3. 该安装程序将设置位于 **Marketing Operations | umoConfiguration | integrationServices | enableIntegrationServices** 的配置属性。可通过更改配置参数来定制您的安装。有关更多信息，请参阅第 14 页的『配置参数』。

## 软件开发人员工具包内容

通常，所有 Marketing Operations Integration Services 组件都安装在标签为“devkits”文件夹中。

软件开发人员工具包包含文档、包含所有 `publicapi` 类和接口的 Javadoc 以及示例代码。

示例代码安装在以下文件夹中。

- **build** 文件夹包含用于构建和部署定制过程的脚本。
- **Classes** 文件夹包含已进行编译的过程类。

用户必须部署位于配置参数 `integrationProcedureClasspathURL` 所指定路径的定制过程的编译类。然后，IBM Marketing Operations Procedure Manager 将按 `procedure-plugins.xml` 配置文件中指定的方式装入这些类。

- **lib** 文件夹包含用于开发和编译定制过程的必需库。
- **src** 文件夹包含定制过程的源文件。用户可将要作为触发器或 `web-service` 启动的定制过程放置在此处。
  - **src/procedure** 文件夹包含 `procedure-plugins.xml` 配置文件。作为基于事件的触发器或者通过外部 `web service` 运行的每个定制过程都必须在此文件中具有一个条目。这些条目必须包含过程的标准类路径以及必需的初始化参数。
  - **src/procedure** 文件夹还包含 IBM Marketing Operations 随附的某些样本过程。这些过程可用来了解和开发定制过程。

将定制过程放置在新文件夹结构中的 **src** 目录下，例如，`com/<mycompany>/<mypackage>`。不要将定制过程放置在样本过程文件夹中。

- **src/soap** 文件夹包含在 Java 中开发的样本 `web service` 客户机。使用这些样本作为针对 `Integration Services` 开发基于 `web service` 的客户机的起始点。此文件夹还包含用于通过命令行启动样本客户机的二进制脚本。

---

## 托管的 Javadoc

有关公共 API 方法的特定信息，请参阅 JavaDocs API 文档文件中的 `iPlanAPI` 类。

通过以下方式可获取这些文件：

- 通过主管 `Marketing Operations` 的服务器上的 `<IBM_EMM>/<MarketingOperations_Home>/devkits/integration/javadocs` 目录中的文件。
- 登录到 `Marketing Operations` 并从任何页面选择 [帮助 > 产品文档](#)，然后下载 `IBM <V>PublicAPI.zip` 文件。

---

## 有关更多信息

组织中的不同人员使用 `IBM Marketing Operations` 来完成不同的任务。一系列指南中提供了有关 `Marketing Operations` 的信息，其中每种指南都旨在供具有特定目标和技能集合的团队成员使用。

下表描述了每个指南中提供的信息。

表 1. Marketing Operations 文档集中的指南

如果您执行以下任务	请参阅	受众
<ul style="list-style-type: none"> <li>计划和管理项目</li> <li>确定工作流程任务、里程碑和人员</li> <li>跟踪项目费用</li> <li>获得有关内容的审阅和审批</li> <li>生成报告</li> </ul>	《IBM Marketing Operations 用户指南》	<ul style="list-style-type: none"> <li>项目经理</li> <li>创意设计师</li> <li>直接邮寄市场营销经理</li> </ul>
<ul style="list-style-type: none"> <li>设计模板、表单、特性和度量</li> <li>定制用户界面</li> <li>定义用户访问级别和安全</li> <li>实现可选功能</li> <li>配置和调优 Marketing Operations</li> </ul>	《IBM Marketing Operations 管理员指南》	<ul style="list-style-type: none"> <li>项目经理</li> <li>IT 管理员</li> <li>实现顾问</li> </ul>
<ul style="list-style-type: none"> <li>创建营销营销活动</li> <li>计划供给</li> <li>实现 Marketing Operations 与 Campaign 的集成</li> <li>实现 Marketing Operations 与 IBM Digital Recommendations 的集成</li> </ul>	《IBM Marketing Operations 和 IBM Campaign 集成指南》	<ul style="list-style-type: none"> <li>项目经理</li> <li>市场营销执行专家</li> <li>直销经理</li> </ul>
<ul style="list-style-type: none"> <li>了解有关新系统功能的信息</li> <li>研究已知问题和变通方法</li> </ul>	《IBM Marketing Operations 发行说明》	使用 Marketing Operations 的任何人
<ul style="list-style-type: none"> <li>安装 Marketing Operations</li> <li>配置 Marketing Operations</li> <li>升级到 Marketing Operations 的新版本</li> </ul>	《IBM Marketing Operations 安装指南》	<ul style="list-style-type: none"> <li>软件实现顾问</li> <li>IT 管理员</li> <li>数据库管理员</li> </ul>
创建定制过程以将 Marketing Operations 与其他应用程序集成	《IBM Marketing Operations 集成模块》，以及当您在 Marketing Operations 中单击帮助 > 产品信息并下载 IBM <version>PublicAPI.zip 文件时提供的 API JavaDoc。	<ul style="list-style-type: none"> <li>IT 管理员</li> <li>数据库管理员</li> <li>实现顾问</li> </ul>
了解有关 Marketing Operations 数据库结构的信息	《IBM Marketing Operations 系统模式》	数据库管理员
在工作时需要更多信息	<ul style="list-style-type: none"> <li>获取帮助并搜索或浏览用户指南、管理员指南或安装指南: 单击帮助 &gt; 此页的帮助</li> <li>访问所有 Marketing Operations 指南: 单击帮助 &gt; 产品文档</li> <li>访问所有 IBM Enterprise Marketing Management (EMM) 产品的指南: 单击帮助 &gt; 全部 IBM EMM Suite 文档</li> </ul>	使用 Marketing Operations 的任何人



---

## 第 2 章 Marketing Operations 集成 Webservice

Webservice 提供 Marketing Operations Integration Services 的客户机视图，而 Integration Services 是 IBM Marketing Operations 服务器部署的一部分。该服务供 Marketing Operations Web 用户并发使用。

Webservice 提供一个 API 调用，即 `executeProcedure`。

客户机直接进行此 Webservice 调用。

---

### Marketing Operations Integration Services WSDL

Web 服务定义语言 (WSDL) 通过手动定义，并且是关于 Webservice 定义的最终表述。

#### Axis

此版本的 Webservice 使用 Axis2 1.5.2 来通过 WSDL 文件生成构成 Web service 实现的服务器端类。用户可以通过提供的 WSDL，随意使用任何版本的 Axis 或任何其他非 Axis 技术来创建客户机端实现，以与 API 集成。

#### 协议版本

协议的版本与 WSDL 显式绑定，如下所示：

- 作为 WSDL 名称的一部分，例如，`PlanIntegrationService1.0.wsdl`
- 作为 WSDL `targetNamespace` 的一部分，例如，`xmlns:tns="http://webservices.unica.com/MktOps/services/PlanIntegrationServices1.0?wsdl"`

#### WSDL

IBM Marketing Operations Integration Services 随附一个 WSDL 文件：`PlanIntegrationServices1.0.wsdl`。该 WSDL 在 `integration/examples/soap/plan` 目录中提供。示例构建脚本使用此文件来生成相应的客户机端存根以连接至 Webservice。

---

### executeProcedure

#### 语法

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

#### 返回值

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

#### 描述

此方法使用可选的参数数组调用指定过程。调用将以同步方式执行；即，它将阻止客户机并在完成时返回结果。

## 参数

表 2. *executeProcedure* 参数

名称	描述
key	要运行的过程的唯一键。如果没有为 <b>key</b> 绑定任何过程，那么将返回 <i>RemoteException</i> 错误。
jobid	标识与此过程执行关联的作业的可选字符串。此字符串是传递项目，但它可用于将客户机任务与特定过程的执行关联在一起。
paramArray	要传递至过程的参数数组。如果一个或多个参数无效（例如，错误类型或不正确值），那么将返回错误状态和消息。将由客户机来确定参数、参数类型以及过程所需值的数目。

## 返回参数

表 3. *executeProcedure* 返回参数

名称	描述
status	整数代码： <ul style="list-style-type: none"><li>• 0 指示过程已成功运行</li><li>• 整数指示错误</li></ul> 过程可以使用状态来指示不同级别的错误。
messages	零或多个消息数据结构的数组。如果 <b>status</b> 为 0，那么此数组不包含 ERROR 消息，但可以包含 INFORMATION 和 WARNING 消息。  如果 <b>status</b> 非零，那么 messages 可以包含 ERROR、INFORMATION 和 WARNING 消息的任何组合。

---

## Marketing Operations 集成 Webservice 数据类型

Webservice 使用的数据类型与任何特定服务绑定或编程实现无关。

将使用以下表示法。

- `<type>`: `<type definition>` 定义简单的数据类型。例如：

Handle: string

- `<type>`: `[ <type definition> ]` 定义复杂的数据类型或数据结构。
- `<type>`: `{ <type definition> }` 定义复杂的数据类型或数据结构。

复杂的类型元素和 API 参数可以使用这些类型来声明数组。例如：

Handle [] handles

类型 handles 是 Handle 类型的数组。

### 基本类型

基本类型仅限于下表中定义的类型，用于简化 SOAP 1.1 绑定的支持。可以将所有类型声明为数组，例如 **String [ ]**。实质上，二进制数据类型（如 **long**）可以由协议绑定（例如 SOAP）表示为字符串。然而，此表示对于客户机看到的类型语义、许可的值等没有作用。

表 4. 基本类型

API 类型	描述	SOAP 类型	Java 类型
布尔值	布尔值: <b>true</b> 或 <b>false</b>	xsd:Boolean	布尔值
dateTime	日期时间值	xsd:datetime	Date
decimal	任意精度十进制值	xsd:decimal	java.math.BigDecimal
double	带有符号的双精度十进制值	xsd:double	double
int	带有符号的 32 位整数值	xsd:int	int
integer	带有符号的任意精度整数值	xsd:integer	java.math.BigInteger
long	带有符号的 64 位整数值	xsd:long	long
字符串	Unicode 字符的字符串	xsd:string	java.lang.String

## MessageTypeEnum

MessageTypeEnum: { INFORMATION, WARNING, ERROR }

MessageTypeEnum 是定义所有可能消息类型的枚举类型。

- INFORMATION: 参考消息
- WARNING: 警告消息
- ERROR: 错误消息

## 消息

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

Message 是定义 Webservice API 调用结果的数据结构。它提供非本地化代码、本地化文本和日志详细信息的可选字段。当前，所有本地化文本都使用为 IBM Marketing Operations 服务器实例设置的语言环境。

表 5. 消息参数

参数	描述
type	MessageTypeEnum, 设置消息的类型。
code	消息的可选代码, 采用字符串格式。
localizedText	用于与消息关联的可选文本字符串。
logDetail	可选堆栈跟踪消息。

## NameValue

NameValue: [string name, int sequence]

NameValue 是定义名称/值对的基本复杂类型。它还定义服务用于根据需要构造值数组的可选序列（序列基于零）。

名称相同但序号不同的所有 NameValue 都将转换为值的数组，并与公共名称关联。

数组大小由最大序号确定；未指定的数组元素具有空值。数组序号必须唯一。值和值的类型由扩展类型提供。

表 6. NameValue 参数

参数	描述
name	定义 NameValue 类型的名称的字符串。
顺序	基于零的整数，它设置 NameValue 暗含值的序号。

为每个基本类型定义了扩展 NameValue 类型，如下所示：

表 7. 扩展 NameValue 类型

扩展类型	描述
BigDecimalNameValue: NameValue [ decimal value]	值为任意精度十进制数字的 NameValue 类型。
BigIntegerNameValue: NameValue [ integer value]	值为任意大小整数的 NameValue 类型。
BooleanNameValue: NameValue [ Boolean value]	值为布尔值的 NameValue 类型。
CurrencyNameValue: NameValue [ string locale, decimal value]	适用于表示某些语言环境中的货币的 NameValue 类型。语言环境是 ISO 语言代码，即 ISO-639 定义的两个小写字母的代码。  当前，语言环境必须与 IBM Marketing Operations 服务器实例中设置的语言环境一致。
DateNameValue: NameValue [ datetime value]	值为日期的 NameValue 类型。
DecimalNameValue: NameValue [ double value]	值为双精度十进制数字的 NameValue 类型。
IntegerNameValue: NameValue [ long value]	值为 64 位整数的 NameValue 类型。
String NameValue: NameValue [ string value]	值为字符串的 NameValue 类型。

最终，将定义扩展 NameValue 类型的数组，以供在必须定义不同类型 NameValue 集合时使用。

```

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



---

## 第 3 章 IBM Marketing Operations 过程

“过程”是 IBM Marketing Operations 托管的定制或标准 Java 类，它执行某些工作单元。过程提供了一种方式，使客户和 IBM Professional Services 可以通过任意方式扩展 Marketing Operations 业务逻辑。

过程遵循简单的编程模型，它使用正确定义的 API 来影响 Marketing Operations 管理的组件。通过简单的查找机制和基于 XML 的定义文件来“发现”过程。Marketing Operations 根据过程的“客户机”的需求来运行这些过程。例如，对集成请求（入局）或触发器触发（内部或出局）进行响应。

过程与其客户机以同步方式运行；结果通过持久的审计机制直接提供给客户机。过程的执行可能还会导致在 Marketing Operations 中触发其他事件和触发器。

必须使用 Java 编写过程。

---

### 假设

#### 过程实现

过程实现类封装在单独的类树或 JAR 文件中，并通过 URL 路径提供给 IBM Marketing Operations。过程执行管理器使用独立的类装入器来根据需要装入这些类。缺省情况下，Marketing Operations 在以下目录中查找：

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

要更改此缺省值，请对**设置 > 配置 > Marketing Operations > umoConfiguration > integrationServices** 下的 **integrationProcedureClasspathURL** 参数进行设置。

过程实现类名遵循公认的 Java 命名约定，以避免与“unica”和其他供应商提供的类发生程序包冲突。特别是客户不得将过程置于“com.unica”或“com.unicacorp”程序包树下。

已使用应用程序服务器上的 IBM Marketing Operations 所使用的 Java 运行时版本（至少为 JRE 1.5.10）来对过程实现进行编码。

过程实现类由 IBM Marketing Operations 通常使用的类装入策略（通常是父类后装入）来装入。应用程序服务器可能会提供开发工具和选项以重新装入适用于 Marketing Operations 过程的类，但此行为并非必需。

#### 库

IBM Marketing Operations 提供一些开放式源代码和第三方库；应用程序服务器也使用这些库的不同版本。

通常，此列表随发行版的不同而更改。对于 V9.1.0，以下第三方库受支持。

- Ant 1.6.5 (ant.jar)
- Axis2 1.5.2 和依赖关系

- 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
- xlpScanner.jar
- xlpScannerUtils.jar
- xlpWASparsing.jar
- wsdl4j-1.6.2.jar
- XmlSchema-1.4.3.jar
- JavaMail 1.4.3 (activation.jar 和 mail.jar)
- JUnit 4.4 (junit-4.4.jar)
- IBM Marketing Operations API (affinium\_plan.jar)
- IBM Marketing Platform API (unica-common.jar)

如果过程或该过程导入的次级类使用此类程序包，那么这些包的使用必须与 Marketing Operations 或应用程序服务器提供的包完全一致。在这种情况下，如果 Marketing Operations 的将来版本对库进行升级或放弃库，那么您需要重新编写过程代码。

## 过程和线程

对于过程自己的状态，过程必须是线程安全的；即它的运行方法不能依赖于不同调用间的内部状态更改。过程不能自己创建线程。

---

## 配置参数

当您安装 Marketing Operations 集成模块时，安装程序将设置三个配置属性。可修改配置属性以定制集成模块的行为。

集成模块的配置属性位于 **Marketing Operations | umoConfiguration | integrationServices** 下。

- **enableIntegrationServices** 配置属性，用于关闭和开启 Integration Services 模块。
- **integrationProcedureDefinitionPath** 参数，包含定制过程定义 XML 文件的完整文件路径。

缺省值为 `<IBM_EMM_Home><MarketingOperations_Home>/devkits/integration/examples/src/procedure/procedure-plugins.xml/`。

- **integrationProcedureClasspathURL** 参数，包含定制过程类路径的 URL。

缺省值为 `file:///<IBM_EMM_Home><MarketingOperations_Home>/devkits/integration/examples/classes/`。

注：integrationProcedureClasspathURL 路径末尾的“/”是正确装入过程类所必需的。

---

## 设计

过程实现类使用 IBM Marketing Operations API 来读取和更新 Marketing Operations 组件、启动服务等等。其他 Java 包可用于执行其他任务。

设计应着重于生成以原子方式执行的单个工作单元。在理想情况下，过程执行一系列任务，可通过异步方式将这些任务调度为在将来某时运行。此“一劳永逸”的集成模型导致两个系统上都存在最小的负载。

注：在 Marketing Operations 的将来发行版中，将仅支持文档化的类和方法。应将 Marketing Operations 中的所有其他类和方法都视为禁止使用。

在编写过程实现类的代码并进行编译之后，应使它们对于 Marketing Operations 可用。随 Marketing Operations Integration Services 提供的构建脚本将编译后的过程置于缺省位置。最后的开发步骤是更新 Marketing Operations 使用的定制过程插件定义文件，以发现定制过程。

过程必须实现 `com.unica.publicapi.plan.plugin.procedure.IProcedure` 接口，并必须具有无参数的构造方法（常见的 JavaBeans 模型）。每个过程的编码和编译将在客户选择的 Java IDE（如 Eclipse、Borland JBuilder 或 Idea）中完成。IBM Marketing Operations 随附了样本代码作为开发者工具包，它们位于以下位置：

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

---

## 过程生命周期

每个过程都在完整生命周期中运行。

过程的运行时生命周期包括下列步骤。

1. 发现和初始化
2. 选择（可选）
3. 执行
4. 销毁

### 发现和初始化

必须使 IBM Marketing Operations 了解可用于特定安装实例的所有标准和定制过程。此过程称为发现。

注：标准过程（由 Marketing Operations 工程团队定义的过程）是隐式已知的，因此无需任何操作就可以发现这些过程。

定制过程在过程插件定义文件中定义。Marketing Operations 插件管理器在初始化期间读取此文件。对于找到的每个过程，插件管理器将完成下列步骤。

1. 对过程进行实例化；将其状态转换为 INSTANTIATED。
2. 创建过程审计记录。
3. 如果已对过程进行实例化，那么将使用其插件描述文件中找到的任何初始化参数来调用其 **initialize()** 方法。如果此方法抛出异常，那么将记录状态，并且会放弃该过程。否则，该过程的状态将转换为 INITIALIZED。现在该过程随时可运行。
4. 创建过程审计记录。
5. 如果已对过程进行初始化，那么将调用其 **getKey** 方法来确定客户机用于引用该过程的键。会将此键与实例关联，并保存以供将来查找。

## 选择

有时，IBM Marketing Operations 可能会向用户提供可用过程的列表，例如使管理员可以设置触发器。Marketing Operations 在初始化过程之后，将使用过程的 **getDisplayName()** 和 **getDescription()** 方法仅提供此列表。

## 执行

在已对过程进行初始化后的某个时刻，IBM Marketing Operations 将收到运行该过程的请求。这可能会与其他线程上运行的其他过程（或同一过程）同时发生。

在运行时，过程执行管理器将完成下列步骤。

1. 启动数据库事务。
2. 将过程状态设置为 EXECUTING。
3. 创建过程审计记录。
4. 带执行上下文和由客户机提供的任何运行参数来调用过程的 **execute()** 方法。方法实现将根据需要使用 Marketing Operations API，同时获取编辑锁定并沿执行上下文方向传递。如果 **run** 方法抛出异常，那么执行管理器将标记事务以进行回滚。
5. 根据执行结果落实或回滚事务；将过程状态设置为 EXECUTED。
6. 释放任何未决的编辑锁定。
7. 创建过程审计记录。

注: **execute()** 方法不应改变过程实例数据。

## 销毁

当 IBM Marketing Operations 关闭时，过程插件管理器将遍历所有装入的过程。对于找到的每个过程，它将完成下列步骤。

1. 调用过程的 **destroy()** 方法，以使过程可在销毁实例之前进行清理。
2. 将过程的状态更改为 FINALIZED（无法运行该过程）。
3. 创建过程审计记录。
4. 销毁过程的实例。

---

## 关键 Java 类

提供的集成开发人员工具包包含公用 IBM Marketing Operations API 和支持类的一组 Javadoc。

此处列出了最重要的一些 Java 类。

- **IProcedure** (com.unica.publicapi.plan.plugin.procedure.IProcedure): 所有过程必须实现的接口。过程经历正确定义的生命周期，并访问 Marketing Operations API 以执行工作。
- **ITriggerProcedure** (com.unica.publicapi.plan.plugin.procedure.ITriggerProcedure): 所有触发器过程必须实现的接口（标记接口）。
- **IExecutionContext** (com.unica.publicapi.plan.plugin.procedure.IExecutionContext): 由执行管理器传递至过程的不透明上下文对象的接口。此对象具有用于记录和编辑锁定管理的公用方法。过程还会将此对象传递至所有 PlanAPI 调用。
- **IPlanAPI** (com.unica.publicapi.plan.api.IPlanAPI): Marketing Operations API 的接口。执行上下文提供 **getPlanAPI()** 方法以检索正确的实现。

---

## 数据锁定

IBM Marketing Operations 使用保守式编辑锁定方案；即，每次只向一个用户授予组件实例的更新访问权。对于 GUI 用户，此锁定在可视选项卡级别上完成。在某些情况下，将锁定部分实例的数据，例如，“项目摘要”选项卡。在其他情况下，将锁定许多实例内的数据，例如，“工作流程”选项卡。用户获取锁定之后，所有其他用户对于相关数据都只能进行读访问。

为确保其他用户不会在无意中覆盖过程对组件实例或实例组进行的更改，过程在更新组件数据之前必须获取相应的锁定。将使用传递至过程 **execute()** 方法的执行上下文对象来实现锁定数据。

在过程更新任何数据之前，它必须为所需的每个锁定调用上下文的 **acquireLock()** 方法。例如，如果过程将更新项目和关联的工作流程，那么该过程必须获取这两者的锁定。

如果其他用户已拥有锁定，那么 **acquireLock()** 方法将立即抛出 **LockInUseException**。为尽量减少冲突，过程必须在更新对象后立即释放锁定。

当 **execute** 方法返回时，执行管理器将自动释放所有未决的锁定。在任何情况下，仅在数据库事务的生存期中保持锁定。这就是说，如果超出了特定于数据库的事务超时值，那么锁定将到期。

**注：**编辑锁定与数据库事务不同。

---

## 过程事务

过程执行管理器自动使用数据库事务对过程的执行进行打包，根据过程执行的结果进行落实或回滚。

包装过程执行和数据库交易将确保在落实之前其他用户不会看到对 IBM Marketing Operations 数据库进行的更新。同时以原子方式进行更新。

过程写程序仍然必须获取必要的编辑锁定，以确保在过程执行完成之前其他用户无法向数据库写入更改。

---

## 过程通信

过程的 **execute()** 方法将整数状态码返回到 IBM Marketing Operations 过程审计表中。过程的 **execute()** 方法返回整数状态码，会将这些信息记录并持久保存在过程审计表中。

客户机也可以使用某些其他的方式来传递状态信息。

---

## 过程日志记录

IBM Marketing Operations 具有用于过程的单独日志文件: `<MarketingOperations_Home>\logs\procedure.log`

过程执行管理器将记录每个过程的生命周期并创建审计记录。

- **logInfo()**: 将参考消息写入到过程日志。
- **logWarning()**: 将警告消息写入到过程日志。
- **logError**: 将错误消息写入到过程日志。
- **logException()**: 将异常的堆栈跟踪转储到过程日志。

---

## 过程插件定义文件

此过程插件定义文件定义有关要在 IBM Marketing Operations 中托管的定制过程的实现类、元数据和其他信息。

缺省情况下，假定过程插件定义位于以下路径中：

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

此文件是包含以下信息的 XML 文档。

过程：零或多个 **Procedure** 元素的列表。

过程：定义过程的元素。每个过程包含以下元素。

- **key** (可选)：定义过程的查找键的字符串。对于特定 Marketing Operations 实例托管的所有标准 (IBM 提供的) 过程和定制过程，此键必须唯一。如果未定义，那么将缺省为 **className** 元素的标准版本。以字符串“uap”开头的名称保留供 IBM Marketing Operations 使用。
- **className** (必需)：过程类的标准程序包名。此类必须实现 **IProcedure** 类 (`com.unica.public.plan.plugin.procedure.IProcedure`)。
- **initParameters** (可选)：零或多个 **initParameter** 元素的列表。

**initParameter** (可选)：要传递至过程的 `initialize()` 方法的参数。此元素包括嵌套的参数 `name`、`type` 和 `value` 元素。

– `name`: 定义参数名称的字符串

– `type`: 定义参数值类型的 Java 包装器类的可选类名。必须是以下某个类型：

- java.lang.String (缺省值)
  - java.lang.Integer
  - java.lang.Double
  - java.lang.Calendar
  - java.lang.Boolean
- value: 根据类型而定的属性值的字符串格式





---

## 第 4 章 IBM Marketing Operations API

IBM Marketing Operations API 是一种外观，它提供正在运行的 Marketing Operations 实例的客户机视图。

仅向用户显示部分 Marketing Operations 功能。该 API 供 Marketing Operations Web 用户与 Marketing Operations Integration Services Webservice SOAP 请求和触发器并发使用。该 API 支持以下类型的操作：

- 组件创建和删除
- 发现（按组件类型、属性值和其他值）
- 组件检查（通过其属性、专用链接和其他值）
- 组件修改

---

### IBM Marketing Operations API 的内容

com.unica.publicapi.plan.api 软件包提供了 IBM Marketing Operations API。

此软件包提供了接口和异常，并且包含以下类型的类：

- 枚举数据类型。
- 用于标识对象和组件实例的句柄。
- 一种 Java 映射，AttributeMap。

通过在 Marketing Operations 实例中单击[帮助 > 产品文档](#)然后下载 IBM `<version>PublicAPI.zip` 文件可获得 API 的完整文档，包括所有方法和可能的值。

### API 接口

IBM Marketing Operations 应用程序编程接口 (API) 包含以下接口：**IPlanAPI** 和 **IExecutionContext**。

Marketing Operations API 包含以下接口。

#### **IPlanAPI**

定义 Marketing Operations 的公用 API。提供用于创建、发现和修改对象（包括文件夹、项目、工作流程任务和团队成员）的方法。

对于启用了可选的与 IBM Campaign 集成选项的系统，还提供用于创建、发现和修改商品的方法。

#### **IExecutionContext**

定义在 API 中执行方法的触发器和锁定。

### API 方法

有关公共 API 方法的特定信息，请参阅 JavaDocs API 文档文件中的 `iPlanAPI` 类。

通过登录到 Marketing Operations 并可从任何页面选择[帮助 > 产品文档](#)，然后下载 `<V>PublicAPI.zip` 文件来获取这些文件。

## 常见异常

API 抛出的常见异常包括以下几种：  
NotFoundException、AuthorizationException、DataException、InvalidExecutionContextException 和 NotLockedException。

以下列表说明可能发生这些常见异常的原因。

- `<object type>NotFoundException`: 系统无法返回指定的项或对象。
- `AuthorizationException`: 与执行上下文关联的用户无权执行请求的操作。任何 API 方法都可以抛出此异常，因此未声明此异常。
- `DataException`: IBM Marketing Operations 中的底层数据库层中发生了异常。请查看 SQL 日志以获取详细信息。
- `InvalidExecutionContextException`: 传递至 API 方法的执行上下文发生问题，例如，未对该方法正确初始化。任何 API 都可以抛出此异常，因此未声明此异常。
- `NotLockedException`: 未首先获取必需的锁定就尝试更新组件数据。请参阅 `IExecutionContext` 接口的 `acquireLock()` 方法。

## 句柄

句柄是引用 IBM Marketing Operations 实例中某个特定对象实例的特殊 URL 对象。句柄包括组件类型、内部数据标识和实例基本 URL。

由 API 使用或生成的句柄可以外部化为完整 URL。可通过不同方法使用生成的 URL。可以使用 URL 以在 Marketing Operations GUI 中打开组件视图，在电子邮件中发送或在其他过程中用作参数。

句柄仅对于特定 Marketing Operations 服务实例或集群实例有效，但对于已部署服务的整个生存期都有效。因此，可以将句柄保存在文件中以供将来引用，但无法将其用于访问其他 Marketing Operations 实例上的组件。此限制也适用于在同一物理主机服务器上的实例。但是，Marketing Operations 确实提供了一种将不同的基本 URL 映射至当前实例的机制，以适应将实例重新放置至其他服务器（例如，当设备发生故障时）。

句柄独立于客户机。例如，触发器可以将句柄传递至过程，该过程将其作为参数用在对第三方系统的 SOAP 调用中。第三方系统向 Marketing Operations 发回 SOAP 请求，以启动更新属性的过程。

`Handle` 类的成员具有从各种类型的 URL 创建句柄的工厂方法。以下是示例。

### 审批

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

### 资产

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

### 资产文件夹

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

### 资产库

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

## 附件

<http://mychane:7001/plan/affiniumplan.jsp?cat=attachmentview&attachid=101&parentObjectId=101&parentObjectType=project>

## 财务帐户

<http://localhost:7001/plan/affiniumplan.jsp?cat=accountdetails&accountid=101>

## 文件夹

[http://mymachine:7001/plan/affiniumplan.jsp?cat=grouping\\_folder&folderid=1234](http://mymachine:7001/plan/affiniumplan.jsp?cat=grouping_folder&folderid=1234)

## 发票

<http://localhost:7001/plan/affiniumplan.jsp?cat=invoicedetails&invoiceid=134>

## 发票行项

[http://localhost:7001/plan/affiniumplan.jsp?cat=invoicedetails&invoiceid=134&line\\_item\\_id=101](http://localhost:7001/plan/affiniumplan.jsp?cat=invoicedetails&invoiceid=134&line_item_id=101)

## 营销对象

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

## 市场营销对象网格

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

## 市场营销对象网格行

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

## 计划团队

<http://mychane:7001/plan/affiniumplan.jsp?cat=teamdetails&func=edit&teamid=100001>

## 计划用户

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

## 项目

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

## 项目网格

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

## 项目网格行

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

## 项目行项

<http://localhost:7001/plan/affiniumplan.jsp?cat=projecttabs&projectid=1234&projectlineitemid=123&projectlineitemisversionfinal=false>

## 工作流程阶段

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

## 工作流程任务

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

## 属性映射

AttributeMap 类是仅包含属性的 Java 映射。属性 `<Name>` 是映射条目键，而属性 `<values>` 数组（请注意复数）是映射条目值。

AttributeMap 类包含下列字段。

- `<Name>`: 属性的编程名称。此名称用作在发生映射的组件实例中访问该属性的唯一键。

**注:** `<Name>` 不一定是 GUI 中向用户显示的显示名。对于根据模板（如项目或工作流程任务）创建的组件，属性名由模板元素定义指定。属性名称必须是唯一的。对于其他组件，属性名通常以编程方式从服务器端组件实例派生而来（例如，通过 Java 自省）。

**注:** 按照惯例，定制属性包括用于定义可编辑版本的格式的名称：`<form_name>.<attribute_name>`。

- `Values`: Java 对象数组，它包含零或多个属性值。每个值的类型必须相同并且与 Marketing Operations 中定义的属性类型一致。仅支持以下 Java 包装器和 Marketing Operations 类型：
  - AssetLibraryStateEnum: AssetLibraryStateEnum 枚举类型值。
  - AssetStateEnum: AssetStateEnum 枚举类型值。
  - AttachmentTypeEnum: AttachmentTypeEnum 枚举类型值。
  - AttributeMap: 保存属性的映射。
  - BudgetPeriodEnum: BudgetPeriodEnum 枚举类型值。
  - BudgetTypeEnum: BudgetTypeEnum 枚举类型值。
  - Handle: 对组件实例、网格行、属性等的引用。
  - InvoiceStateEnum: InvoiceStateEnum 枚举类型值。
  - java.io.File: 文件的说明。
  - java.lang.Boolean: 布尔值，True 或 False
  - java.lang.Double: 双精度十进制数值。
  - java.lang.Float: 单精度十进制数值
  - java.lang.Integer: 32 位整数值
  - java.lang.Long: 64 位整数值
  - java.lang.Object: 类属 Java 对象
  - java.lang.String: 零或多个 Unicode 字符组成的字符串
  - java.math.BigDecimal: 带有符号的任意精度十进制数值。适用于货币；该值的解释取决于客户机的货币语言环境。
  - java.math.BigInteger: 任意精度整数值。
  - java.net.URL: 统一资源定位器 (URL) 对象。
  - java.util.ArrayList: 对象列表。
  - java.util.Calendar: 特定语言环境的日期时间值。
  - java.util.Date: 日期时间值。建议不要使用此类型。请改为使用 java.util.Calendar 或 java.util.GregorianCalendar。

**注:** 要实现日期，用户可以选择 java.util.Calendar 或 java.util.GregorianCalendar。

- java.util.GregorianCalendar: GregorianCalendar 是 java.util.Calendar 的具体子类, 并提供世界上大部分地区使用的标准日历系统。
- MonthEnum: MonthEnum 枚举类型值。
- ProjectStateEnum: ProjectStateEnum 枚举类型值。
- QuarterEnum: QuarterEnum 枚举类型值。
- TaskStateEnum: TaskStateEnum 枚举类型值。
- WeekEnum: WeekEnum 枚举类型值。

属性的元数据（如已翻译的显示名和描述）由与该属性及其父对象实例关联的模板定义。属性提供了一种简单而又可扩展的机制，可用于显示必需和可选的对象实例属性，如项目名称、代码和开始日期。

## 枚举数据类型

### **ApprovalMethodEnum**

ApprovalMethodEnum 定义有效的审批方法。可能的值有:

- SEQUENTIAL
- SIMULTANEOUS

### **ApprovalStateEnum**

ApprovalStateEnum 定义有效的审批状态。可能的值有:

- CANCELLED
- COMPLETED
- IN\_PROGRESS
- NOT\_STATED
- ON\_HOLD

### **AssetLibraryStateEnum**

AssetLibraryStateEnum 定义有效的资产库状态。可能的值有:

- DISABLED
- ENABLED

### **AssetStateEnum**

AssetStateEnum 定义有效的资产状态。可能的值有:

- ARCHIVE
- DRAFT
- FINALIZE
- LOCK

### **AttachmentTypeEnum**

AttachmentTypeEnum 定义有效的附件类型。可能的值有:

- ASSET
- FILE
- URL

### **BudgetPeriodEnum**

BudgetPeriodEnum 定义可能的预算周期。可能的值包括:

- ALL
- MONTHLY
- QUARTERLY
- WEEKLY
- YEARLY

### **BudgetTypeEnum**

BudgetTypeEnum 定义有效的预算类型。可能的值有:

- ACTUAL
- ALLOCATED
- COMMITTED
- FORECAST
- TOTAL

### **ComponentTypeEnum**

ComponentTypeEnum 标识可访问的 Marketing Operations 组件类型。可能的值有:

- APPROVAL
- ASSET
- ASSET\_FOLDER
- ASSET\_LIBRARY
- ATTACHMENT
- FINANCIAL\_ACCOUNT
- GROUPING\_FOLDER
- INVOICE
- MARKETING\_OBJECT
- PLAN\_TEAM
- PLAN\_USER
- PROJECT
- PROJECT\_REQUEST
- TASK
- 

### **InvoiceStateEnum**

InvoiceStateEnum 定义有效的发票状态。可能的值有:

- CANCELLED
- 草稿
- PAID
- PAYABLE

### **MonthEnum**

MonthEnum 定义月份的有效值。

### **OfferStateEnum**

OfferStateEnum 定义有效的供给状态。可能的值有:

- STATE\_OFFER\_DRAFT
- STATE\_OFFER\_PUBLISHED
- STATE\_OFFER\_RETIRED

### **ProjectCopyTypeEnum**

ProjectCopyTypeEnum 定义用于复制项目的有效方法。可能的值有:

- COPY\_USING\_PROJECT\_METRICS
- COPY\_USING\_TEMPLATES\_METRICS

### **ProjectParticipantLevelEnum**

ProjectParticipantLevelEnum 标识用户的项目中可充当的角色。可能的值有:

- OWNER
- PARTICIPANT
- REQUESTER

### **ProjectStateEnum**

ProjectStateEnum 定义有效的项目和请求状态。可能的值包括:

- ACCEPTED
- CANCELLED
- COMPLETED
- DRAFT
- IN\_PROGRESS
- IN\_RECONCILIATION
- LATE: 项目未在其调度开始日期之前启动。
- NOT\_STARTED
- ON\_HOLD
- OVERDUE: 项目未在其调度结束日期之前完成。
- RETURNED
- SUBMITTED

有关项目和任务状态的更多信息, 请参阅《IBM Marketing Operations 用户指南》。

### **QuarterEnum**

QuarterEnum 定义季度的有效值: Q1、Q2、Q3 和 Q4。

### **TaskStateEnum**

TaskStateEnum 定义有效的工作流程任务状态。可能的值有:

- ACTIVE
- DISABLED
- FINISHED
- PENDING
- SKIPPED

### **WeekEnum**

WeekEnum 定义一年中的星期, 从 WEEK\_1 到 WEEK\_53。





---

## 与 IBM 技术支持联系之前

如果遇到无法通过查阅文档解决的问题，那么贵公司的指定支持联系可致电 IBM 技术支持。使用这些准则来确保高效而成功地解决问题。

如果您不是贵公司的指定支持联系，请与 IBM 管理员联系以了解相关信息。

### 要收集的信息

与 IBM 技术支持联系之前，请收集以下信息：

- 有关问题性质的简短描述。
- 发生问题时您看到的详细错误消息。
- 重现该问题的详细步骤。
- 相关的日志文件、会话文件、配置文件和数据文件。
- 有关产品和系统环境的信息，可按“系统信息”中所述获得此信息。

### 系统信息

致电 IBM 技术支持时，可能会要求您提供有关系统环境的信息。

如果产生的问题并不妨碍您登录，那么在“关于”页面上可获得大部分此类信息，该页面提供有关已安装的 IBM 应用程序的信息。

可以通过选择帮助 > 关于来访问“关于”页面。如果无法访问“关于”页面，请查看位于应用程序安装目录下的 `version.txt` 文件。

### IBM 技术支持的联系信息

有关与 IBM 技术支持进行联系的方式，请查看 IBM 产品技术支持 Web 站点：[http://www.ibm.com/support/entry/portal/open\\_service\\_request](http://www.ibm.com/support/entry/portal/open_service_request)。

**注：**要输入支持请求，您必须使用 IBM 帐户进行登录。此帐户必须链接至您的 IBM 客户编号。要了解有关将您的帐户与 IBM 客户编号相关联的更多信息，请参阅“支持门户网站”上的[支持资源 > 授权的软件支持](#)。



---

## 声明

本信息是为在美国提供的产品和服务而编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

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

有关双字节 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：International Business Machines Corporation“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本资料中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及 (ii) 允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation  
170 Tracer Lane  
Waltham, MA 02451  
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本文档中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际程序许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估算的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时变更或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

所显示的所有 IBM 的价格均是 IBM 当前的建议零售价，可随时更改，而不另行通知。经销商的价格可能会有所不同。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口 (API) 进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。样本程序都是“按现状”提供的，不附有任何种类的保证。对于因使用样本程序所引起的任何损害，IBM 概不负责。

如果您正以软拷贝格式查看本信息，那么图片和彩色图例可能无法显示。

---

## 商标信息

IBM、IBM 徽标和 [ibm.com](http://ibm.com)<sup>®</sup> 是 International Business Machines Corp. 在全球许多管辖区域内注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。当前的 IBM 商标列表，可从 Web 站点 [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) 上『版权和商标信息』部分获取。

---

## 隐私策略和使用条款注意事项

IBM 软件产品（包括软件即服务解决方案，“软件产品”）可能使用 cookie 或其他技术来收集产品使用信息，以帮助改善最终用户体验，定制与最终用户的交互或用于其他目的。cookie 是 Web 站点可发送至浏览器的数据，随后可将其存储在您的计算机上作为标识您的计算机的标记。在许多情况下，这些 cookie 不会收集个人信息。如果您要使用的软件产品允许您通过 cookie 或类似技术收集个人信息，我们将在下面告知您具体情况。

根据已部署的配置，此软件产品可能使用会话和持久性 cookie，它们收集各个用户的用户名和其他个人信息以用于会话管理、增强用户可用性或其他使用跟踪或功能性目的。可以禁用这些 cookie，但禁用它们也将使它们所支持的功能无法使用。

不同管辖区域对通过 cookie 和类型技术收集个人信息进行不同程度的规管。如果此软件产品的已部署配置为您（作为客户）提供通过 cookie 和其他技术收集个人信息的能力，那么您应当主动征求有关任何适用于此类数据收集（包括提供通知和同意的任何要求，如适用）的法律的法律建议。

IBM 要求客户端 (1) 提供明确、显著的指向客户的 Web 站点使用条款（其中包括指向 IBM 和客户端的数据收集和使用实践的链接）的链接（如隐私政策），(2) 告知访问者 IBM 代表客户将 cookie 和透明 GIF/网络信标存放在访问者的计算机上并说明此类技术的目的，并且 (3) 在法律允许的范围内，在客户或 IBM 代表客户将 cookie 和透明 GIF/网络信标存放在 Web 站点访问者的设备上之前征得 Web 站点访问者的同意。

有关用于这些用途的各种技术（包括 cookie）的使用的更多信息，请参阅 IBM 的联机隐私声明（<http://www.ibm.com/privacy/details/us/en>）中标题为“Cookie、网络信标和其他技术”的部分。



Printed in China