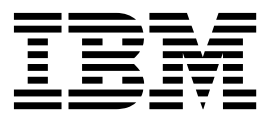


Version 11 Edition 1
15 mars 2019

*IBM Marketing Operations Module
d'intégration*



Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations de la rubrique «Remarques», à la page 33.

Table des matières

Chapitre 1. Présentation d'IBM

Marketing Operations Integration

Services 1

Conditions requises pour Marketing Operations

Integration Services 2

IBM Marketing Operations Généralités sur

Integration Services 3

Installation des services d'intégration 5

Contenu du kit du développeur 6

Fichiers JavaDoc hébergés 6

Documentation et aide relatives à Marketing

Operations 7

Chapitre 2. Marketing Operations

Service Web Integration 9

Marketing Operations Langage WSDL associé à

Integration Services 9

executeProcedure 9

Marketing Operations Type de données de service

Web Integration 10

Chapitre 3. IBM Marketing Operations

Procédures 15

Hypothèses 15

Paramètres de configuration 17

Conception 17

Cycle de vie de la procédure 18

Principales classes Java 19

Verrouillage des données 20

Transactions de procédure 20

Communication de procédure 21

Consignation des procédures 21

Fichier de définition du plug-in de procédure 21

Chapitre 4. IBM Marketing Operations

API SOAP d' 23

Contenu de l'API SOAP d'IBM Marketing

Operations 23

Interfaces de l'API SOAP 23

Exceptions courantes de l'API SOAP 24

Descripteurs de l'API SOAP 24

Classe AttributeSet de l'API SOAP 26

Types de données énumérées de l'API SOAP 27

Avant de contacter le support

technique d'IBM 31

Remarques 33

Marques 35

Règles de confidentialité et conditions d'utilisation 35

Chapitre 1. Présentation d'IBM Marketing Operations Integration Services

IBM® Marketing Operations Integration Services combine les services Web Marketing Operations Integration, les procédures d'API SOAP et les déclencheurs pour augmenter les fonctionnalités métier.

IBM Marketing Operations Integration Services se compose des éléments suivants :

- **Marketing Operations Service Web Integration**

La fonction Integration Services permet aux clients Marketing Operations, ainsi qu'aux applications IBM Professional Services d'intégrer Marketing Operations à d'autres applications qui s'exécutent dans leur environnement.

- **Marketing Operations Procédures et API SOAP de**

Vous pouvez définir des procédures personnalisées dans Marketing Operations pour étendre la logique métier Marketing Operations de façon arbitraire. Une fois définies, ces procédures peuvent être les cibles des appels de service Web Integration Services provenant d'autres applications. Il est également possible de définir des procédures pour envoyer des messages à d'autres applications.

- **Marketing Operations déclencheurs**

Les déclencheurs peuvent être associés à des événement et à des procédures dans Marketing Operations. Lorsque ce type d'événement se produit, le déclencheur associé est exécuté.

Les API REST n'utilisent pas les services d'intégration de Marketing Operations. Pour des informations sur l'API REST, voir le guide d'administration d'IBM Marketing Operations.

Gestion des versions et compatibilité amont

Les futures versions des services d'intégration seront compatibles avec les versions antérieures et toutes les éditions secondaires et de maintenance qui partagent le même numéro de version. Toutefois, IBM se réserve le droit d'abandonner la compatibilité avec la version antérieure pour les éditions principales "point zéro" (x.0) si le script commercial ou technique le justifie.

Le numéro de version principal de cette API est incrémenté si l'une des modifications suivantes est apportée :

- modification de l'interprétation des données ;
- modification de la logique métier (par exemple, modification des fonctions de méthode de service) ;
- modification des paramètres de méthode et/ou des types de retour.

Le numéro d'édition de cette API est incrémenté si l'une des modifications ci-après est apportée. Ces modifications sont compatibles avec une version antérieure par définition.

- ajout d'une nouvelle méthode ;
- ajout d'un nouveau type de données et restriction de son utilisation à une nouvelle méthode ;
- ajout d'un nouvel élément à un type énuméré ;

- définition d'une nouvelle version d'interface avec un suffixe de version.

Authentification

L'authentification n'est pas requise ; tous les clients sont associés à un utilisateur IBM Marketing Operations connu appelé PlanAPIUser. Les fonctions de sécurité de cet utilisateur spécial sont configurées par un administrateur système en fonction des besoins de tous les clients du service Web.

Environnement local

La seule langue prise en charge est la langue actuellement configurée pour l'instance de système IBM Marketing Operations. Les données qui dépendent de l'environnement local, tel que les messages et la devise, sont en principe intégrées à cet environnement.

Gestion d'état

Les API et le service Web sont *sans état* ; aucune information par client n'est sauvegardée par la mise en oeuvre du service au fil des appels API. Cette fonction permet de bénéficier d'une mise en œuvre de service plus efficace et simplifie la prise en charge du cluster.

Transactions de base de données

Marketing Operations Integration Services ne montre pas les transactions de base de données au client, mais utilise ces informations si elles sont incluses dans le contexte d'exécution. Si une transaction est démarrée, l'effet de tous les appels API au sein d'une procédure particulière peut être atomique. Cela signifie qu'un échec d'appel API laisse la base de données dans le même état que si aucun appel API n'avait été émis. Les autres utilisateurs de Marketing Operations ne voient pas les modifications tant que la procédure n'a pas validé la transaction.

Les appels API qui mettent à jour la base de données doivent tout d'abord acquérir un verrou d'édition afin d'empêcher les autres utilisateurs API de modifier les données sous-jacentes durant les appels API. Les autres utilisateurs ne peuvent pas mettre à jour les composants verrouillés tant que l'appel API n'est pas terminé. De même, l'utilisateur ou le client API Marketing Operations suivant doit acquérir le verrou des données avant la soumission d'un autre appel API.

Traitement d'événement

Les opérations effectuées sur les composants IBM Marketing Operations via cette API génèrent les mêmes événements que si l'opération avait été effectuée par un utilisateur Web de Marketing Operations. Les utilisateurs qui ont souscrit à certaines notifications (par exemple, un changement d'état d'un projet) seront avertis des changements d'état résultant des appels API et des actions des utilisateurs.

Conditions requises pour Marketing Operations Integration Services

Marketing Operations Les conditions suivantes sont requises pour Integration Services.

Marketing Operations La fonction Integration Services doit :

- Coupler de façon souple l'intégration du système

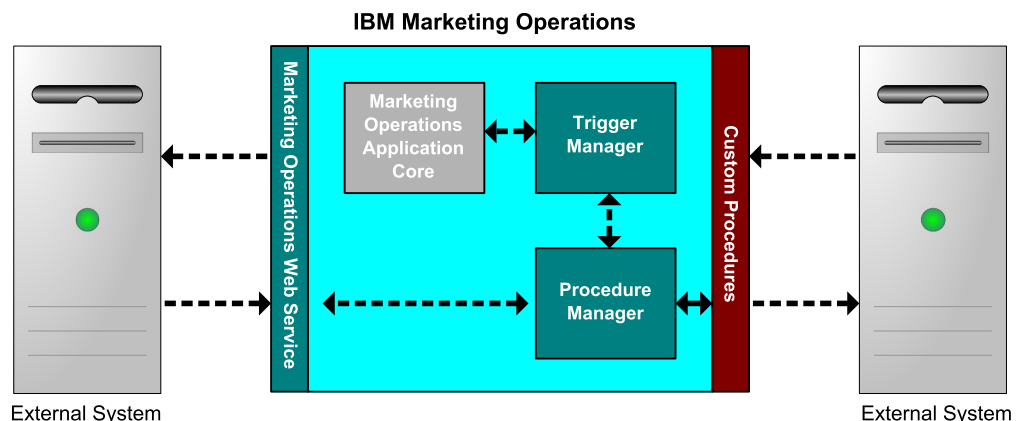
- Fournir un mécanisme permettant aux applications du client d'affecter Marketing Operations via des appels de service Web
- Fournir un mécanisme permettant aux applications du client d'être averties de certains événements dans Marketing Operations
- Fournir un modèle de programmation simple, facile à comprendre et à utiliser
- Etre robuste en cas de reprise sur incident
- Garantir l'intégrité des données
- S'intégrer aux clients Marketing Operations basés sur l'interface graphique existants et réduire les effets sur ces derniers
- Fournir un accès à granularité fine aux composants Marketing Operations tout en isolant les programmeurs des détails de mise en oeuvre sous-jacents

IBM Marketing Operations Généralités sur Integration Services

IBM Marketing Operations Integration Services permet de créer des procédures personnalisées. Vous pouvez utiliser ces procédures pour déclencher des événements externes lorsque certains événements se produisent dans Marketing Operations. Vous pouvez utiliser ces procédures pour exécuter les fonctions Marketing Operations à partir de systèmes ou de programmes externes.

L'interface API interagit avec IBM Marketing Operations au niveau du programme, de la même façon que vous utilisez l'interface graphique comme interface avec Marketing Operations au niveau utilisateur. L'API vous permet de construire des procédures. Ces dernières vous permettent d'établir une communication entre Marketing Operations et les systèmes externes. Le service Web Marketing Operations est l'objet conteneur de ces procédures, de l'API et des déclencheurs.

L'architecture de Marketing Operations Integration Services vous est présentée ici.

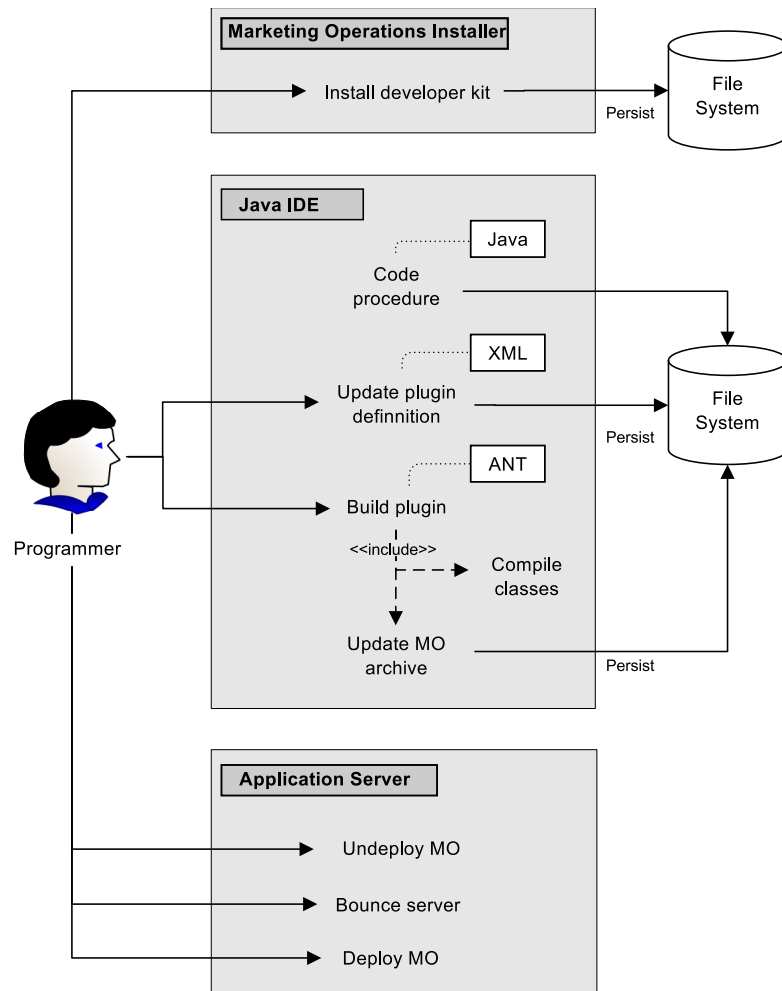


Voici les principaux composants des services Integration Services :

- Marketing Operations Procedure Manager : étend la logique métier en interagissant avec Marketing Operations via l'API.
- Marketing Operations Trigger Manager : associe une condition (par exemple, le changement d'état d'un projet marketing) à une action (une procédure à exécuter lorsque la condition associée au déclencheur est remplie).

Méthodes

Vous utilisez les composants d'IBM Marketing Operations Integration Services pour développer des procédures personnalisées, comme l'indique le diagramme suivant :



Après avoir installé le kit de développement, vous effectuez les étapes de base suivantes :

1. Codification de la procédure personnalisée.
2. Mise à jour de la définition du plug-in dans le fichier de définition XML.
3. Génération du plug-in :
 - a. Compilation des classes nécessaires.
 - b. Si vous utilisez une bibliothèque tiers qui ne fait pas partie de l'archive Marketing Operations, intégrez la bibliothèque dans le fichier `plan.war` et procédez au redéploiement.
4. Redémarrage de Marketing Operations. Les modifications apportées aux classes de procédure sont appliquées lorsque vous redémarrez le serveur d'application.

Remarque : Si vous modifiez le fichier **plan.war**, vous devez annuler le déploiement et redéployer Marketing Operations avec le nouveau fichier **plan.war**. Annulez le déploiement et redéployez Marketing Operations si vous utilisez une bibliothèque tiers qui ne fait pas partie de l'archive Marketing Operations et que vous modifiez le fichier **plan.war**.

Exemple de base de communication entre IBM Marketing Operations et l'API

L'exemple suivant décrit l'établissement d'une communication entre l'API et Marketing Operations. Il ne s'agit pas d'un travail utile mais d'un aller-retour entre Marketing Operations et Integration Services.

Cet exemple utilise des portions des exemples de procédures fournis avec le kit de développement de Marketing Operations Integration Services. Vous pouvez trouver le code référencé ici dans les fichiers suivants :

- PlanClientFacade.java
- PlanWSNOOPTestCase.java

La méthode noop est un appel de service Web en direction de Marketing Operations. Elle est définie dans la classe PlanClientFacade et transmet des valeurs nulles dans une matrice.

```
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);
}
```

La procédure testExecuteProcedure appelle la méthode noop à partir de PlanClientFacade pour établir un aller-retour avec l'application Marketing Operations.

```
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.");
}
```

Pour plus d'informations sur NameValueArrays, ProcedureResponse et les autres méthodes et types de données listés, voir *Marketing Operations Integration Module* et les fichiers JavaDoc.

Installation des services d'intégration

Le module des services d'intégration IBM Marketing Operations est un composant distinct et payant. Si vous achetez ce module, vous devez l'installer.

1. Téléchargez les programmes d'installation des services d'intégration IBM Marketing Operations.
2. Les programmes d'installation IBM Marketing Software détectent le module des services d'intégration.
3. Les propriétés de configuration suivantes sont définies : **Marketing Operations** | **umoConfiguration** | **integrationServices** | **enableIntegrationServices**. Vous

pouvez personnaliser l'installation en modifiant les paramètres de configuration. Pour plus d'informations, voir «Paramètres de configuration», à la page 17.

Contenu du kit du développeur

Le kit de développement de logiciels comporte une documentation qui contient toutes les classes et interfaces publicapi, ainsi qu'un exemple de code.

Pour l'API SOAP, tous les composants de Marketing Operations Integration Services sont installés dans un dossier dont le nom est `devkits`.

Le code exemple est installé dans les dossiers suivants :

- Le dossier **build**, qui contient les scripts permettant de générer et de déployer les procédures personnalisées.
- Le dossier **Classes**, qui contient les classes de procédures compilées.
Les utilisateurs doivent déployer les classes compilées de leurs procédures personnalisées dans le répertoire spécifié par le paramètre de configuration **integrationProcedureClasspathURL**. Ensuite, IBM Marketing Operations Procedure Manager les charge comme spécifié dans le fichier de configuration `procedure-plugins.xml`.
- Le dossier **lib**, qui contient les bibliothèques requises pour le développement et la compilation des procédures personnalisées.
- Le dossier **src**, qui contient les fichiers source pour les procédures personnalisées. Les utilisateurs peuvent stocker à cet emplacement les procédures personnalisées à utiliser en tant que déclencheurs ou services Web. L'API SOAP seulement prend en charge les procédures personnalisées.
 - Le dossier **src/procedure**, qui contient le fichier de configuration `procedure-plugins.xml`. Chaque procédure personnalisée exécutée en tant que déclencheur en fonction d'un événement ou via un service Web externe doit avoir une entrée dans ce fichier. Les entrées doivent contenir un chemin d'accès complet aux classes de procédures et aux paramètres d'initialisation requis.
 - Le dossier **src/procedure**, qui contient également des procédures exemple, incluses dans IBM Marketing Operations. Ces procédures peuvent être utilisées pour comprendre et développer vos propres procédures personnalisées.
Placez les procédures personnalisées dans une nouvelle structure de dossier sous le répertoire **src**, par exemple : `com/<ma_société>/<mon_package>`. Ne placez pas les procédures personnalisées dans le dossier des procédures exemple.
 - Le dossier **src/soap** contient les clients de service Web exemple développés en Java. Utilisez ces exemples comme point de départ pour développer les clients de service Web pour les services d'intégration. Ce dossier contient également des scripts binaires permettant de démarrer les clients exemple via la ligne de commande.

Fichiers JavaDoc hébergés

Pour obtenir des informations spécifiques sur les méthodes API publiques, reportez-vous à la classe `iPlanAPI` dans les fichiers de documentation API JavaDoc.

Ces fichiers sont mis à disposition des différentes façons suivantes :

- Via les fichiers du répertoire <IBM_IMS>/<MarketingOperations_Home>/devkits/integration/javadocs pour l'API SOAP sur le serveur qui héberge Marketing Operations.
- Via la procédure suivante : connectez-vous à Marketing Operations et sélectionnez **Aide > Documentation sur le produit** à partir de n'importe quelle page, puis téléchargez le fichier IBM <version>PublicAPI.zip pour l'API SOAP.

Documentation et aide relatives à Marketing Operations

Différentes personnes de votre organisation utilisent IBM Marketing Operations pour exécuter différentes tâches. Les informations sur Marketing Operations sont disponibles dans un ensemble de guides, chacun d'eux étant destiné à être utilisé par des membres de l'équipe ayant des objectifs et des compétences spécifiques.

Le tableau ci-après présente les informations disponibles dans chaque guide.

Tableau 1. Guides de la documentation Marketing Operations.

Le tableau à trois colonnes suivant décrit les tâches dans une colonne, les noms des guides dans la deuxième et le public visé dans la troisième.

Si vous	Voir	Utilisateurs concernés
• Planifiez et gérez des projets • Etablissez des tâches, des jalons et du personnel de workflow • Assurez le suivi des dépenses d'un projet • Obtenez des révisions et des approbations de contenu • Générez des rapports	<i>IBM Marketing Operations Guide d'utilisation</i>	• Chefs de projet • Concepteurs • Directeurs marketing publipostage
• Concevez des modèles, des formulaires, des attributs et des indicateurs • Personnalisez l'interface utilisateur • Définissez des niveaux d'accès utilisateur et des mesures de sécurité • Implémentez des fonctions facultatives • Configurez et réglez Marketing Operations	<i>IBM Marketing Operations Guide d'administration :</i>	• Chefs de projet • Administrateurs informatiques • Conseillers en matière d'implémentation
• Créez des campagnes marketing • Planifiez des offres • Implémentez l'intégration entre Marketing Operations et Campaign • Implémentez l'intégration entre Marketing Operations et IBM Digital Recommendations	<i>IBM Marketing Operations et IBM - Guide d'intégration</i>	• Chefs de projet • Spécialistes d'exécution marketing • Directeurs marketing direct
• Découvrez les nouvelles fonctions système • Recherchez les problèmes et contournement connus	<i>IBM Marketing Operations Edition Notes</i>	Toute personne qui utilise Marketing Operations

Tableau 1. Guides de la documentation Marketing Operations (suite).

Le tableau à trois colonnes suivant décrit les tâches dans une colonne, les noms des guides dans la deuxième et le public visé dans la troisième.

Si vous	Voir	Utilisateurs concernés
• Installer Marketing Operations • Configurer Marketing Operations • Procédez à une mise à niveau vers Marketing Operations	<i>Guide d'installation IBM Marketing Operations</i>	• Conseillers en matière d'implémentation de logiciels • Administrateurs informatiques • Administrateurs de base de données
Créez des procédures personnalisées pour intégrer Marketing Operations à d'autres applications	<i>IBM Marketing Operations Integration Module</i> et l'API JavaDocs, disponibles lorsque vous cliquez sur Aide > Documentation sur le produit dans Marketing Operations, puis que vous téléchargez le fichier IBM<version>PublicAPI.zip pour l'API SOAP et IBM<version>PublicAPI-RestClient.zip pour l'API REST.	• Administrateurs informatiques • Administrateurs de base de données • Conseillers en matière d'implémentation
Découvrir la structure de la base de données Marketing Operations	<i>IBM Marketing Operations Schéma système</i>	Administrateurs de base de données
Avez besoin d'informations supplémentaires pendant que vous travaillez	• Accédez à l'aide et recherchez les manuels suivants : <i>Marketing Operations - Guide d'utilisation</i>, <i>Guide d'administration</i>, ou <i>Marketing Operations - Guide d'installation</i> : Cliquez sur Aide > Aide pour cette page • Accédez à tous les guides Marketing Operations : cliquez sur Aide > Documentation sur le produit • Accédez aux guides concernant tous les produits IBM Marketing Software : cliquez sur Aide > Ensemble de la documentation IBM Marketing Software Suite	Toute personne qui utilise Marketing Operations

Chapitre 2. Marketing Operations Service Web Integration

Le service Web fournit une vue client de Marketing Operations Integration Services, qui fait partie du déploiement du serveur IBM Marketing Operations. Le service est utilisé en même temps que les utilisateurs Web de Marketing Operations.

Le service Web prend en charge un appel API, `executeProcedure`.

C'est un client qui effectue directement cet appel de service Web.

Marketing Operations Langage WSDL associé à Integration Services

Le langage WSDL a été défini manuellement et constitue le point final de la définition du service Web.

Axis

Cette version du service Web utilise Axis2 1.5.2 pour générer les classes côté serveur qui constituent la mise en œuvre du service web à partir du fichier WSDL. Les utilisateurs peuvent employer n'importe quelle version de Axis ou une technique autre que Axis, pour créer une mise en œuvre côté client permettant une intégration avec l'API à partir du WSDL fourni.

Version du protocole

La version du protocole est explicitement liée au WSDL :

- Dans le nom WSDL, par exemple, `PlanIntegrationService1.0.wsdl`
- En tant que partie du `targetNamespace` WSDL, par exemple,
`xmlns:tns="http://webservices.unica.com /MktOps/services/PlanIntegrationServices1.0?wsdl"`

langage WSDL

Un fichier WSDL est fourni avec IBM Marketing Operations Integration Services : `PlanIntegrationServices1.0.wsdl`. Ce fichier WSDL est situé dans le répertoire `integration/examples/soap/plan`. L'exemple de script de génération utilise ce fichier pour générer les modules de remplacement côté client appropriés à connecter au service Web.

`executeProcedure`

`executeProcedure` est l'appel d'API pris en charge par le service Web.

Syntaxe

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

Renvoie

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

Description

Cette méthode appelle la procédure spécifiée avec une matrice de paramètres facultatifs. L'appel s'exécute de façon synchrone, c'est-à-dire qu'il bloque le client et renvoie le résultat à l'achèvement de l'exécution.

Paramètres

Tableau 2. Paramètres *executeProcedure*

Nom	Description
clé	Clé unique de la procédure à exécuter. Une erreur <i>RemoteException</i> est renvoyée si aucune procédure n'est liée à key .
ID de travail	Chaîne facultative qui identifie le travail associé à l'exécution de cette procédure. Cette chaîne est un élément passe-système mais elle peut être utilisée pour lier des tâches client à l'exécution d'une procédure particulière.
paramArray	Une matrice de paramètres à transmettre à la procédure. Un état et un message d'erreur sont renvoyés si un ou plusieurs des paramètres sont non valides (par exemple, type non valide ou valeur incorrecte). C'est au client qu'il revient de déterminer les paramètres, leur type et le nombre de valeurs requises par la procédure.

Paramètres de retour

Tableau 3. Paramètres de retour de *executeProcedure*

Nom	Description
statut	Code entier : • 0 indique que l'exécution de la procédure a abouti • un entier indique une erreur Les procédures peuvent utiliser l'état pour indiquer différents niveaux d'erreur.
messages	Une matrice de zéro ou plusieurs structures de données de message. Si status a pour valeur 0, cette matrice ne contient pas de messages d'ERREUR mais peut contenir des messages d'INFORMATION et d'AVERTISSEMENT. Si status est différent de zéro, les messages peuvent contenir un mélange de messages d'ERREUR, d'INFORMATION et d'AVERTISSEMENT.

Marketing Operations Type de données de service Web Integration

Les types de données utilisés par le service Web, indépendamment d'une liaison de service ou d'une mise en œuvre de programme particulière.

La notation suivante est utilisée :

- **<type>** : *<type definition>* définit un type de données simple. Par exemple :
Handle: string
- **<type>** : [*<type definition>*] définit un type de données complexe ou une structure de données.
- **<type>**: { *<type definition>* } définit un type de données complexe ou une structure de données.

Les éléments de type complexe et les paramètres API peuvent utiliser ces types pour déclarer des matrices. Par exemple :

```
Handle [] handles
```

Le type, handles, est une matrice de types Handle.

Types primitifs

Les types primitifs sont limités aux types définis dans la table qui suit pour simplifier la prise en charge des liaisons SOAP 1.1. Tous les types peuvent être déclarés sous forme de matrices, par exemple, **String** []. Fondamentalement, les types de données binaires tels que **long** peuvent être représentés sous forme de chaînes par une liaison de protocole (par exemple, SOAP). Cependant, cette représentation n'a aucun effet sur la sémantique du type, les valeurs admises, etc., tels qu'ils sont vus par le client.

Tableau 4. Types primitifs

Type API	Description	Type SOAP	Java™ Type
Booléen	Valeur booléenne : true ou false	xsd:Boolean	Booléen
dateTime	Valeur de date/heure	xsd:datetime	Date
décimal	Valeur décimale à précision arbitraire	xsd:decimal	java.math.BigDecimal
double	Valeur décimale signée à double précision	xsd:double	double
int	Valeur de type entier signée 32 bits	xsd:int	int
entier	Valeur de type entier signée à précision arbitraire	xsd:integer	java.math.BigInteger
long	Valeur de type entier signée 64 bits	xsd:long	long
chaîne	Chaîne de caractères Unicode	xsd:string	java.lang.String

MessageTypeEnum

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

MessageTypeEnum est un type énuméré qui définit tous les types de message possibles.

- INFORMATION : message d'information
- WARNING : message d'avertissement
- ERROR : message d'erreur

Message

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

Message est une structure de données qui définit le résultat d'un appel API de service Web. Il fournit des zones facultatives pour le code non localisé, le texte localisé et le détail du journal. Actuellement, tous les textes localisés utilisent la langue définie pour l'instance de serveur IBM Marketing Operations.

Tableau 5. Paramètres du message

Paramètre	Description
type	MessageTypeEnum définissant le type du message.
code	Code facultatif au format chaîne (string) pour le message.
localizedText	Chaîne de texte facultative à associer au message.
logDetail	Message de trace de pile facultatif.

NameValue

NameValue: [string name, int sequence]

NameValue est un type complexe de base qui définit une paire nom-valeur. Il définit également une séquence facultative utilisée par le service pour construire les matrices de valeur nécessaires (les séquences sont de base zéro).

Tous les NameValues portant le même nom mais possédant des numéros de séquence différents sont convertis dans une matrice de valeurs et associés au nom commun.

La taille de la matrice dépend du numéro de séquence maximal. Les éléments de matrice non spécifiés possèdent la valeur NULL. Les numéros de séquence de matrice doivent être uniques. La valeur et son type sont fournis par le type étendu.

Tableau 6. Paramètres de NameValue

Paramètre	Description
nom	Chaîne qui définit le nom d'un type NameValue.
séquence	Entier de base zéro qui définit le numéro de séquence de la valeur NameValue concernée.

Les types NameValue étendus sont définis pour chaque type primitif, comme suit :

Tableau 7. Types NameValue étendus

Type étendu	Description
BigDecimalNameValue: NameValue [decimal value]	Type NameValue dont la valeur est un nombre décimal à précision arbitraire.
BigIntegerNameValue: NameValue [integer value]	Type NameValue dont la valeur est un entier signée de façon arbitraire.
BooleanNameValue: NameValue [Boolean value]	Type NameValue dont la valeur est un booléen.
CurrencyNameValue: NameValue [string locale, decimal value]	Type NameValue convenant pour représenter les devises dans une langue spécifique. La langue est représentée par un code de langue ISO, c'est-à-dire un code à deux lettres en minuscules, tel que défini par la norme ISO-639. Actuellement, la langue doit correspondre à la langue définie dans l'instance de serveur IBM Marketing Operations.
DateNameValue: NameValue [datetime value]	Type NameValue dont la valeur est une date.

Tableau 7. Types NameValue étendus (suite)

Type étendu	Description
DecimalNameValue: NameValue [double value]	Type NameValue dont la valeur est un nombre décimal à double précision.
IntegerNameValue: NameValue [long value]	Type NameValue dont la valeur est un entier de 64 bits.
String NameValue: NameValue [string value]	Type NameValue dont la valeur est une chaîne.

Une matrice des types NameValue étendus est définie afin d'être utilisée lorsque vous avez besoin de définir un jeu de NameValues de différents types.

```

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

```

Chapitre 3. IBM Marketing Operations Procédures

Une "procédure" est une classe Java personnalisée ou standard, hébergée par IBM Marketing Operations, qui exécute une unité de travail. Les procédures permettent aux clients et aux services IBM Professional Services d'étendre la logique métier Marketing Operations de façon arbitraire.

Les procédures suivent un modèle de programmation simple avec une API bien définie pour affecter des composants gérés par Marketing Operations. La reconnaissance des procédures s'effectue via un mécanisme de recherche simple et un fichier de définition XML. Marketing Operations exécute les procédures en fonction des besoins de ses "clients". (par exemple, en réponse à une demande d'intégration (entrante) ou à l'action d'un déclencheur (interne ou sortant)).

Les procédures s'exécutent de façon synchronisée avec leur client. Les résultats sont directement mis à la disposition du client via un mécanisme d'audit persistant. L'exécution d'une procédure peut également provoquer d'autres événements et déclencheurs dans Marketing Operations.

Les procédures doivent être écrites en Java.

Hypothèses

Les classes d'implémentation de procédure sont regroupées dans une arborescence de classes différente ou dans un autre fichier JAR et sont mises à la disposition d'IBM Marketing Operations via un chemin URL.

Implémentation de procédure

Le gestionnaire d'exécution de procédure utilise un chargeur de classe indépendant pour charger ces classes en fonction des besoins. Par défaut, Marketing Operations recherche dans le répertoire suivant :

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

Pour changer ce paramètre par défaut, définissez le paramètre **integrationProcedureClasspathURL** sous **Paramètres > Configuration > Marketing Operations > umoConfiguration > integrationServices**.

Le nom de la classe de mise en œuvre de procédure obéit aux conventions de dénomination Java acceptées, afin d'éviter des collisions de package avec "unica" et avec les classes des autres fournisseurs. Les clients ne doivent pas placer des procédures dans l'arborescence de packages "com.unica" ou "com.unicacorp".

La mise en œuvre de procédure est codée dans la version Java Runtime utilisée par IBM Marketing Operations sur le serveur d'applications (au minimum JRE 1.5.10).

La classe d'implémentation de procédure est chargée par la règle de chargement de classe qui est normalement utilisée par IBM Marketing Operations (généralement **parent-last**). Le serveur d'application peut fournir des outils de développement et des options permettant de recharger les classes que vous pouvez appliquer aux procédures Marketing Operations mais qui ne sont pas requises.

Bibliothèques

IBM Marketing Operations fournit des bibliothèques Open Source et tierces ; les serveurs d'applications utilisent également différentes versions de ces bibliothèques.

En général, cette liste varie d'une version à une autre. Les bibliothèques tierces suivantes sont prises en charge :

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

Si une procédure ou les classes secondaires importées par la procédure utilisent ces packages, leur utilisation doit être totalement conforme à celle des packages fournis par Marketing Operations ou par le serveur d'applications. Dans ce cas, il est nécessaire de retravailler votre code de procédure si une version ultérieure de Marketing Operations met à niveau ou abandonne une bibliothèque.

Procédures et unités d'exécution

La procédure doit autoriser les unités d'exécution multiples concernant son propre état. Cela signifie que sa méthode d'exécution ne peut pas dépendre des changements d'état internes d'un appel à un autre. Une procédure ne peut pas créer des unités d'exécution par elle-même.

Paramètres de configuration

Lorsque vous installez le module d'intégration Marketing Operations, trois propriétés de configuration sont définies. Vous pouvez modifier les propriétés de configuration afin de personnaliser le comportement du module d'intégration.

Les propriétés de configuration du module d'intégration sont sous **Marketing Operations | umoConfiguration | integrationServices**.

- La propriété de configuration **enableIntegrationServices** permet d'activer ou de désactiver le module du service d'intégration.
- Le paramètre **integrationProcedureDefinitionPath** contient le chemin d'accès complet au fichier XML de définition de procédure personnalisée.

La valeur par défaut est `<IBM_IMS_Home><MarketingOperations_Home>/devkits/integration/examples/src/procedure/procedure-plugins.xml/`.

- Le paramètre **integrationProcedureClasspathURL** contient l'adresse URL du chemin d'accès aux classes pour les procédures personnalisées.

La valeur par défaut est `file:///<IBM_IMS_Home><MarketingOperations_Home>/devkits/integration/examples/classes/`.

Remarque : Le caractère '/' à la fin du chemin du paramètre `integrationProcedureClasspathURL` est requis pour que le chargement des classes de procédures s'effectue correctement.

Conception

La classe d'implémentation de procédure utilise l'API IBM Marketing Operations pour lire et mettre à jour les composants Marketing Operations, les services de lancement, etc. D'autres packages Java peuvent être utilisés pour l'exécution d'autres tâches.

Lors de la phase de conception, vous devez vous concentrer sur la production d'une unité de travail unique qui fonctionne de façon atomique. Idéalement, une procédure exécute des séries de tâches qui peuvent être planifiées de façon asynchrone pour être exécutées ultérieurement. Ce modèle d'intégration de type "lancer et oublier" permet d'obtenir une charge minimale sur chacun des deux systèmes.

Remarque : Seuls les classes et les méthodes documentées seront prises en charge dans les futures versions de Marketing Operations. Vous devez considérer toutes les autres classes et méthodes de Marketing Operations comme non autorisées.

Une fois que vous avez codé et compilé les classes d'implémentation de procédure, vous les mettez à la disposition de Marketing Operations. Les scripts de génération qui sont fournis avec la fonction Marketing Operations Integration Services placent les procédures compilées à l'emplacement par défaut. L'étape de développement final consiste à mettre à jour le fichier de définition du plug-in de procédure personnalisée qui est utilisé par Marketing Operations pour reconnaître les procédures personnalisées.

La procédure doit implémenter l'interface **com.unica.publicapi.plan.plugin.procedure.IProcedure** et comporter un constructeur dans paramètre (modèle JavaBeans habituel). La procédure de codification et de compilation de chaque procédure est effectuée dans un environnement IDE Java choisi par le client (par exemple, Eclipse, Borland JBuilder

ou Idea). Un exemple de code est fourni avec IBM Marketing Operations sous forme de kits d'outils de développement à l'emplacement suivant :

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

Cycle de vie de la procédure

Chaque procédure s'exécute via un cycle de vie complet.

Le cycle de vie d'exécution d'une procédure comprend les étapes suivantes :

1. Reconnaissance et initialisation
2. Sélection (facultatif)
3. Exécution
4. Destruction

Reconnaissance et initialisation

IBM Marketing Operations doit connaître toutes les procédures standard ou personnalisées disponibles pour une instance d'installation particulière. Ce processus est appelé détection.

Remarque : Les procédures standard (procédures définies par l'équipe d'ingénierie Marketing Operations) sont connues implicitement et ne nécessitent donc pas de reconnaissance.

Les procédures personnalisées sont définies dans le fichier de définition du plug-in de procédure. Le gestionnaire de plug-in Marketing Operations lit ce fichier au cours de l'initialisation. Pour chaque procédure détectée, le gestionnaire de plug-in effectue les tâches suivantes :

1. Instanciation de la procédure ; passage de son état à INSTANCIÉE.
2. Création d'un enregistrement d'audit de procédure.
3. Si la procédure a été instanciée, sa méthode **initialize()** est appelée avec tout paramètre d'initialisation trouvé dans son fichier de description de plug-in. Si cette méthode émet une exception, le statut est consigné et la procédure est abandonnée. Dans le cas contraire, la procédure passe à l'état INITIALISÉE. Elle est alors prête à être exécutée.
4. Création d'un enregistrement d'audit de procédure.
5. Si la procédure a été initialisée, sa méthode **getKey()** est appelée pour identifier la clé utilisée par les clients pour référencer la procédure. Cette clé est associée à l'instance et sauvegardée pour une recherche ultérieure.

Sélectionner

De temps en temps, il peut arriver que IBM Marketing Operations présente une liste des procédures disponibles aux utilisateurs, par exemple, pour permettre aux administrateurs de définir un déclencheur. Marketing Operations présente cette liste uniquement une fois que la procédure a été initialisée, via ses méthodes **getDisplayName()** et **getDescription()**.

Exécution

Une fois la procédure initialisée, IBM Marketing Operations reçoit une demande d'exécution de la procédure. Cela peut se produire en même temps que pour d'autres procédures (ou pour la même) s'exécutant sur d'autres unités d'exécution.

Au moment de l'exécution, le gestionnaire d'exécution de procédure effectue les tâches suivantes :

1. Démarrage d'une transaction de base de données.
2. Définition de l'état de la procédure sur EN COURS D'EXECUTION.
3. Création d'un enregistrement d'audit de procédure.
4. Appel de la méthode **execute()** de la procédure avec un contexte d'exécution et tout paramètre d'exécution fourni par le client. La mise en œuvre de la méthode utilise l'API Marketing Operations si nécessaire, en acquérant les verrous d'édition et en transmettant le contexte d'exécution. Si la méthode d'exécution émet une exception, le gestionnaire d'exécution marque la transaction comme devant être annulée.
5. Validation ou annulation de la transaction en fonction des résultats de l'exécution ; définition de l'état de la procédure sur EXECUTEE.
6. Libération de tout verrou d'édition en suspens.
7. Création d'un enregistrement d'audit de procédure.

Remarque : La méthode **execute()** ne doit pas modifier les données d'instance de la procédure.

Destruction

A l'arrêt d'IBM Marketing Operations, le gestionnaire du plug-in de procédure passe en revue toutes les procédures chargées. Pour chaque procédure détectée, il effectue les tâches suivantes :

1. Appel de la méthode `destroy()` de la procédure afin de permettre à cette dernière d'effectuer un nettoyage avant la destruction de l'instance.
2. Passage de l'état de la procédure à FINALISEE (elle ne peut pas être exécutée).
3. Création d'un enregistrement d'audit de procédure.
4. Destruction de l'instance de la procédure.

Principales classes Java

Le kit de développement d'intégration fourni contient une série de fichiers Javadoc pour l'API IBM Marketing Operations publique et pour les classes de support.

Les classes Java les plus importantes sont répertoriées ci-après :

- `IProcedure` (`com.unica.publicapi.plan.plugin.procedure.IProcedure`) : interface que toutes les procédures doivent implémenter. Les procédures ont un cycle de vie bien défini et accèdent à l'API Marketing Operations pour effectuer un travail.
- `ITriggerProcedure` (`com.unica.publicapi.plan.plugin.procedure.ITriggerProcedure`) : interface que toutes les procédures de déclencheur doivent implémenter (interface de marqueur).
- `IExecutionContext` (`com.unica.publicapi.plan.plugin.procedure.IExecutionContext`) : interface d'objet contextuel opaque transmis à la procédure par le gestionnaire d'exécution. Cet objet comporte des méthodes publiques pour la consignation et la gestion des verrous d'édition. La procédure transmet également cet objet à tous les appels PlanAPI.

- IPlanAPI (com.unica.publicapi.plan.api.IPlanAPI) : interface vers l'API Marketing Operations. Le contexte d'exécution fournit une méthode **getPlanAPI()** qui extrait la mise en oeuvre appropriée.

Verrouillage des données

IBM Marketing Operations utilise un schéma de verrouillage d'édition pessimiste, c'est-à-dire qu'à un moment donné, un seul utilisateur possède des droits de mise à jour sur les instances du composant. Pour l'utilisateur de l'interface graphique, ce verrouillage est effectué au niveau visuel de l'onglet. Dans certains cas, les données sont réservées pour un sous-ensemble d'une instance, par exemple via un onglet de récapitulatif de projet. Dans d'autres cas, les données sont partagées entre pour plusieurs instances, par exemple via l'onglet workflow. Lorsqu'un utilisateur a acquis un verrou, tous les autres utilisateurs ne possèdent plus qu'un accès en lecture seule aux données concernées.

Afin d'éviter que les modifications apportées par une procédure à une instance de composant ou à un groupe d'instances ne soient écrasées par inadvertance par un autre utilisateur, une procédure doit acquérir les verrous appropriés avant la mise à jour des données du composant. C'est l'objet contexte d'exécution transmis à la méthode **execute()** de la procédure qui est utilisé pour verrouiller les données.

Avant de mettre à jour des données, la procédure doit appeler la méthode **acquireLock()** du contexte pour chaque verrou dont elle a besoin. Par exemple, si une procédure doit mettre à jour un projet et le workflow associé, elle doit acquérir des verrous pour ces deux éléments.

Si un autre utilisateur possède déjà un verrou, la méthode **acquireLock()** émet immédiatement une exception **LockInUseException**. Afin de réduire le nombre de collisions, la procédure doit libérer le verrou dès qu'elle met à jour l'objet.

Le gestionnaire d'exécution libère automatiquement tout verrou en suspens lorsque la méthode d'exécution revient. Dans tous les cas, les verrous ne sont détenus qu'à concurrence de la durée de vie de la transaction de base de données. Les verrous expirent si le délai d'attente de la transaction propre à la base de données a été dépassé.

Remarque : Les verrous d'édition sont différents des transactions de base de données.

Transactions de procédure

Le gestionnaire d'exécution de procédure effectue automatiquement en boucle l'exécution de la procédure avec une transaction de base de données, en la validant ou en l'annulant en fonction du résultat de l'exécution de la procédure.

L'exécution en boucle de la procédure et de la transaction de base de données permet d'être sûr que les mises à jour de la base de données IBM Marketing Operations ne sont pas visibles des autres utilisateurs tant qu'elles ne sont pas validées, et rend également les mises à jour automatiques.

L'auteur de la procédure doit cependant acquérir les verrous d'édition nécessaires afin d'être sûr que les autres utilisateurs ne puissent pas copier des modifications dans la base de données avant la fin de l'exécution de la procédure.

Communication de procédure

La méthode **execute()** d'une procédure renvoie un code d'état sous forme d'entier à la table d'audit de procédure IBM Marketing Operations. La méthode **execute()** d'une procédure peut également renvoyer zéro ou plusieurs messages qui sont consignés et conservés dans la table d'audit de procédure.

Le client peut également communiquer les informations d'état d'une autre manière.

Consignation des procédures

IBM Marketing Operations possède un fichier journal distinct pour les procédures : `<MarketingOperations_Home>\logs\system.log`

Le gestionnaire d'exécution de procédure consigne le cycle de vie de chaque procédure et crée des enregistrements d'audit.

- **logInfo()** : un message d'information est écrit dans le journal des procédures.
- **logWarning()** : un message d'avertissement est écrit dans le journal des procédures.
- **logError()** : un message d'erreur est écrit dans le journal des procédures.
- **logException()** : la trace de pile pour l'exception est vidée dans le journal des procédures.

Fichier de définition du plug-in de procédure

Le fichier de définition du plug-in de procédure fichier définit la classe de mise en oeuvre, les métadonnées et d'autres informations relatives aux procédures personnalisées à héberger dans IBM Marketing Operations.

Par défaut, la définition du plug-in de procédure est dans le répertoire suivant :

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

Ce fichier est un document XML qui contient les informations présentées ci-après.

Procedures : liste de zéro ou plusieurs éléments **Procedure**.

Procedure : élément qui définit une procédure. Chaque procédure contient les éléments suivants :

- **key** (facultatif) : chaîne qui définit la clé de recherche de la procédure. Cette clé doit être unique entre toutes les procédures standard (fournie par IBM) et personnalisées qui sont hébergées par une instance Marketing Operations donnée. Si elle n'est pas définie, elle prend par défaut la valeur de la version qualifiée complète de l'élément **className**. Les noms commençant par la chaîne "uap" sont réservés à IBM Marketing Operations.
- **className** (obligatoire) : nom de package qualifié complet de la classe de procédure. Cette classe doit implémenter la classe `IProcedure` (`com.unica.public.plan.plugin.procedure.IProcedure`).
- **initParameters** (facultatif) : liste de zéro ou plusieurs éléments **initParameter**.
initParameter (facultatif) : paramètre à transmettre à la méthode `initialize()` de la procédure. Cet élément inclut le nom du paramètre imbriqué, son type et les éléments de valeur.

- name : chaîne qui définit le nom du paramètre
- type : nom de classe facultatif de la classe d'encapsuleur Java qui définit le type de la valeur du paramètre. Il doit s'agir de l'un des types suivants :
 - java.lang.String (valeur par défaut)
 - java.lang.Integer
 - java.lang.Double
 - java.lang.Calendar
 - java.lang.Boolean
- value : forme de la chaîne associée à la valeur d'attribut en fonction de son type

Chapitre 4. IBM Marketing Operations API SOAP d'

L'API SOAP d'IBM Marketing Operations est une façade qui offre une vue client d'une instance de Marketing Operations en cours d'exécution.

Nous n'exposerons aux utilisateurs qu'une petite partie des possibilités de Marketing Operations. L'API est utilisée simultanément par des utilisateurs Web de Marketing Operations et par les demandes et les déclencheurs Marketing Operations Integration Services WebService SOAP. Cette API prend en charge les types suivants d'opérations :

- Création et suppression de composant
- Reconnaissance (par type de composant, valeur d'attribut, etc.)
- Inspection de composant (via ses attributs, de liens spécialisés, etc.)
- Modification de composant

Remarque : Marketing Operations seuls les administrateurs peuvent utiliser les API de .

Contenu de l'API SOAP d'IBM Marketing Operations

Le package `com.unica.publicapi.plan.api` fournit l'API SOAP d'IBM Marketing Operations.

Ce package offre des interfaces et des exceptions et contient les types de classe suivants :

- Des types de données énumérées.
- Des descripteurs qui identifient les instances d'objet et de composant.
- Une mappe Java, `AttributeMap`.

La documentation complète relative à l'API, y compris toutes les méthodes et toutes les valeurs possibles, est disponible en cliquant sur **Aide > Documentation sur le produit** dans une instance de Marketing Operations, puis en téléchargeant le fichier `IBM <version>PublicAPI.zip`.

Interfaces de l'API SOAP

L'interface de programme d'application SOAP d'IBM Marketing Operations inclut **IPlanAPI** et **IExecutionContext**.

L'API SOAP de Marketing Operations inclut les interfaces suivantes :

IPlanAPI

Définit l'API publique pour Marketing Operations. Fournit des méthodes permettant de créer, découvrir et modifier des objets, notamment des dossiers, des projets, des programmes, des tâches de workflow et des membres d'une équipe.

Pour les systèmes sur lesquels l'intégration facultative à IBM Campaign est activée, fournit également des méthodes permettant de créer, découvrir et modifier des offres.

IExecutionContext

Définit les déclencheurs et verrouille ces méthodes d'exécution dans l'API.

Méthodes API

Pour obtenir des informations spécifiques sur les méthodes API publiques, reportez-vous à la classe iPlanAPI dans les fichiers de documentation API JavaDoc.

Pour accéder à ces fichiers, connectez-vous à Marketing Operations et sélectionnez **Aide > Documentation produit** dans n'importe quelle page, puis téléchargez le fichier `<version>PublicAPI.zip`.

Exceptions courantes de l'API SOAP

Les exceptions courantes qui sont émises par l'API SOAP sont notamment NotFoundException, AuthorizationException, DataException, InvalidExecutionContextException et NotLockedException.

La liste ci-après explique pourquoi ces exceptions peuvent se produire.

- *<object type>*NotFoundException : le système n'est pas en mesure de renvoyer l'élément ou l'objet spécifié.
- AuthorizationException : l'utilisateur qui est associé au contexte d'exécution n'est pas autorisé à effectuer l'opération demandée. Cette exception peut être émise par n'importe quelle méthode API, par conséquent, elle n'est pas déclarée.
- DataException une exception s'est produite dans la couche de base de données sous-jacente dans IBM Marketing Operations. Pour plus d'informations, voir le journal SQL.
- InvalidExecutionContextException: Un problème lié au contexte d'exécution transmis à une méthode API s'est produit (par exemple, la méthode n'a pas été correctement initialisée). Cette exception peut être émise par n'importe quelle API, par conséquent, elle n'est pas déclarée.
- NotLockedException: tentative de mise à jour des données de composant sans acquisition préalable du verrou requis. Voir la méthode acquireLock() de l'interface IExecutionContext.

Descripteurs de l'API SOAP

Un descripteur est un objet URL spécial qui référence une instance d'objet particulière dans une instance d'IBM Marketing Operations. Le type de composant, l'identificateur des données interne et une URL de base d'instance sont des descripteurs.

Les descripteurs utilisés ou générés par les API peuvent être externalisés vers une adresse URL complète. Vous pouvez utiliser cette URL de différentes manières, par exemple pour ouvrir une vue du composant dans l'interface graphique Marketing Operations, l'envoyer dans des courriers électroniques ou l'utiliser dans une autre procédure en tant que paramètre.

Les descripteurs sont uniquement valides pour une instance de service ou une instance en cluster Marketing Operations donnée, mais leur validité est garantie pour toute la durée de vie du service déployé. Par conséquent, les descripteurs peuvent être sauvegardés dans un fichier pour une référence ultérieure, mais ils ne peuvent pas être utilisés pour accéder à des composants sur une autre instance Marketing Operations. Cette restriction s'applique également aux instances présentes sur le même serveur hôte physique. Toutefois, Marketing Operations ne fournit pas de mécanisme de mappage de différentes URL de base à l'instance en cours en vue de permettre le déplacement d'une instance sur une autre serveur (par exemple, si le matériel présente un dysfonctionnement).

Les descripteurs sont indépendants du client. Par exemple, un déclencheur peut transmettre un descripteur à une procédure, qui l'utilise ensuite en tant que paramètre dans un appel SOAP vers un système tiers. Le système tiers peut ensuite renvoyer une demande SOAP à Marketing Operations pour démarrer une procédure de mise à jour d'un attribut.

Les membres de la classe Handle comportent des méthodes de fabrique destinées à créer des descripteurs pour divers types d'URL. Exemples :

Approbation

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

Document

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

Dossier de documents

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

Bibliothèque de documents

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

Pièce jointe

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

Compte financier

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

Dossier

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

Facture

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

Ligne de facture

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

Objet marketing

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

Grille d'objet marketing

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

Ligne de grille d'objet marketing

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

Equipe de plan

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

Utilisateur de plan

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

Programme

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

Grille de programme

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

Ligne de grille de programme

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

Projet

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

Grille de projet

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

Ligne de grille de projet

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

Ligne de projet

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

Etat de workflow

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

Tâche de workflow

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

Classe AttributeMap de l'API SOAP

La classe AttributeMap est une mappe Java qui contient uniquement des attributs. L'attribut *<Name>* est la clé d'entrée de mappe et la matrice des *<values>* d'attribut (notez l'emploi du pluriel) est la valeur d'entrée de mappe.

La classe AttributeMap contient les zone suivantes :

- *<Name>* : nom défini par programme de l'attribut. Ce nom sert de clé unique pour accéder à l'attribut dans l'instance de composant où il apparaît.

Remarque : *<Name>* n'est pas obligatoirement le nom d'affichage présenté à l'utilisateur dans l'interface graphique. Pour les composants créés à partir de modèles (tels que les projets ou les tâches de workflow), le nom d'attribut est spécifié par la définition d'élément du modèle. Ce nom doit être unique. Pour les autres composants, le nom d'attribut est généralement dérivé par voie de programme de l'instance de composant côté serveur (par exemple, via l'inspection Java).

Remarque : Par convention, les attributs personnalisés incluent le nom du formulaire dans lequel la version modifiable est définie : *<form_name>.<attribute_name>*.

- *Values* : matrice d'objet Java contenant zéro, une ou plusieurs valeurs d'attribut. Le type de chaque valeur doit être identique et en accord avec le type de l'attribut défini dans Marketing Operations. Seul l'encapsuleur Java et les types Marketing Operations suivants sont pris en charge :
 - AssetLibraryStateEnum : valeur AssetLibraryStateEnum de type énuméré.
 - AssetStateEnum : valeur AssetStateEnum de type énuméré.
 - AttachmentTypeEnum : valeur AttachmentTypeEnum de type énuméré.

- `AttributeMap` : mappe qui contient des attributs.
- `BudgetPeriodEnum` : valeur `BudgetPeriodEnum` de type énuméré.
- `BudgetTypeEnum` : valeur `BudgetTypeEnum` de type énuméré.
- `Handle` : référence à une instance de composant, une ligne de grille, un attribut, etc.
- `InvoiceStateEnum` : valeur `InvoiceStateEnum` de type énuméré.
- `java.io.File` : représentation d'un fichier.
- `java.lang.Boolean` : valeur booléenne (`True` ou `False`)
- `java.lang.Double` : valeur de nombre décimal à double précision.
- `java.lang.Float` : valeur de nombre décimal à simple précision.
- `java.lang.Integer` : valeur de type entier 32 bits
- `java.lang.Long` : valeur de type entier 64 bits
- `java.lang.Object` : Object Java générique
- `java.lang.String` : chaîne comprenant zéro ou plusieurs caractères Unicode
- `java.math.BigDecimal` : valeur de nombre décimal signée à précision arbitraire. Convient pour les devises ; l'interprétation de la valeur dépend de la langue utilisée pour les devises pour le client.
- `java.math.BigInteger` : valeur de type entier à précision arbitraire.
- `java.net.URL` : objet URL.
- `import java.util.ArrayList` : liste des objets.
- `java.util.Calendar` : valeur date-heure pour une langue particulière.
- `java.util.Date` : valeur date-heure. Ce type est obsolète. Utilisez à la place `java.util.Calendar` ou `java.util.GregorianCalendar`.

Remarque : Pour mettre en œuvre la date, les utilisateurs peuvent utiliser `java.util.Calendar` ou `java.util.GregorianCalendar`.

- `java.util.GregorianCalendar` : `GregorianCalendar` est une sous-classe concrète de `java.util.Calendar` et fournit un système de calendrier standard utilisé dans la plupart des pays du monde entier.
- `MonthEnum` : valeur `MonthEnum` de type énuméré.
- `ProjectStateEnum` : valeur `ProjectStateEnum` de type énuméré.
- `QuarterEnum` : valeur `QuarterEnum` de type énuméré.
- `TaskStateEnum` : valeur `TaskStateEnum` de type énuméré.
- `WeekEnum` : valeur `WeekEnum` de type énuméré.

Les métadonnées d'un attribut (telles que le nom d'affichage traduit et la description associée) sont définies par le modèle qui est associé à l'attribut et à son instance d'objet parent. Les attributs fournissent un mécanisme simple et extensible pour afficher les attributs d'instance d'objet facultatifs et obligatoires, tels que le nom du projet, le code et la date de début.

Types de données énumérées de l'API SOAP

L'API SOAP d'IBM Marketing Operations prend en charge les types de données énumérées et les valeurs qui suivent.

ApprovalMethodEnum

`ApprovalMethodEnum` définit des méthodes d'approbation valides. Les valeurs possibles sont:

- SEQUENTIEL

- SIMULTANE

ApprovalStateEnum

ApprovalStateEnum définit des états d'approbation valides. Les valeurs possibles sont:

- CANCELLED
- TERMINE
- EN COURS
- PAS D'ETAT
- EN ATTENTE

AssetLibraryStateEnum

AssetLibraryStateEnum définit des états de bibliothèque de documents valides. Les valeurs possibles sont:

- DESACTIVE
- ENABLED

AssetStateEnum

AssetStateEnum définit des états de document valides. Les valeurs possibles sont:

- ARCHIVE
- DRAFT
- FINALISE
- VERROUILLE

AttachmentTypeEnum

AttachmentTypeEnum définit des types de pièce jointe valides. Les valeurs possibles sont:

- RESSOURCE
- FILE
- URL

BudgetPeriodEnum

BudgetPeriodEnum définit les périodes de budget possibles. Les valeurs possibles sont:

- ALL
- MENSUEL
- TRIMESTRIEL
- HEBDOMADAIRE
- ANNUEL

BudgetTypeEnum

BudgetTypeEnum définit les types de budget possibles. Les valeurs possibles sont:

- REEL
- ALLOUE
- ENGAGE
- PREVU
- TOTAL

ComponentTypeEnum

ComponentTypeEnum identifie les types de composant Marketing Operations accessibles. Les valeurs possibles sont:

- APPROBATION
- RESSOURCE
- DOSSIER_DOCUMENTS
- BIBLIOTHEQUE_DOCUMENTS
- ATTACHMENT
- COMPTE_FINANCIER
- DOSSIER_GROUPEMENT
- FACTURE
- OBJET_MARKETING
- EQUIPE_PLAN
- UTILISATEUR_PLAN
- PROGRAM
- PROJECT
- DEMANDE_PROJET
- TASK
-

InvoiceStateEnum

InvoiceStateEnum définit des états de facture valides. Les valeurs possibles sont:

- CANCELLED
- DRAFT
- PAYE
- PAYABLE

MonthEnum

MonthEnum définit des valeurs de mois valides.

OfferStateEnum

OfferStateEnum définit des états d'offre valides. Les valeurs possibles sont:

- OFFRE_ETAT_BROUILLON
- OFFRE_ETAT_PUBLIEE
- OFFRE_ETAT_RETIREE

ProjectCopyTypeEnum

ProjectCopyTypeEnum définit des méthodes valides de copie d'un projet. Les valeurs possibles sont:

- COPIE_AVEC_INDICATEURS_PROJET
- COPIE_AVEC_INDICATEURS_MODELE

ProjectParticipantLevelEnum

ProjectParticipantLevelEnum identifie les rôles qui peuvent être attribués aux utilisateurs dans un projet. Les valeurs possibles sont:

- OWNER
- PARTICIPANT
- DEMANDEUR

ProjectStateEnum

ProjectStateEnum définit des état de projet et de demande valides. Les valeurs possibles sont:

- ACCEPTE
- CANCELLED

- TERMINE
- DRAFT
- EN COURS
- EN RAPPROCHEMENT
- EN RETARD : le projet n'a pas démarré à la date de début prévue.
- NON DEMARRE
- EN ATTENTE
- DEPASSE : le projet ne s'est pas terminé avant la date de fin prévue.
- RENVOYE
- SUBMITTED

Pour plus d'informations sur des statuts des projets et des tâches, voir *IBM Marketing Operations - Guide d'utilisation*.

QuarterEnum

QuarterEnum définit des valeurs de trimestre valides : Q1, Q2, Q3 et Q4.

TaskStateEnum

TaskStateEnum définit des états de tâche de workflow valides. Les valeurs possibles sont:

- ACTIVE
- DESACTIVE
- FINISHED
- PENDING
- SKIPPED

WeekEnum

WeekEnum définit des valeurs de semaine valides sur une année, comprises entre SEMAINE_1 et SEMAINE_53.

Avant de contacter le support technique d'IBM

Si vous rencontrez un problème que vous ne parvenez pas à résoudre en consultant la documentation, le correspondant désigné pour le support technique de votre entreprise peut contacter le support technique d'IBM. Utilisez les instructions de cette section pour garantir la résolution efficace de votre problème.

Si vous n'êtes pas le correspondant désigné pour le support technique dans votre société, contactez l'administrateur IBM pour plus d'informations.

Remarque : Le support technique n'écrit ni ne crée aucun script d'API. Pour obtenir une aide relative à l'implémentation de nos offres d'API, prenez contact avec IBM Professional Services.

Informations à réunir

Avant de contacter le support technique d'IBM, rassemblez les informations suivantes :

- Une brève description de la nature du problème rencontré
- Les messages d'erreur détaillés s'affichant lorsque le problème se produit
- La liste des étapes complètes permettant de reproduire l'erreur.
- les fichiers journaux, fichiers de session, fichiers de configuration et fichiers de données connexes
- Les informations sur l'environnement de votre système et de votre produit , que vous pouvez obtenir en procédant comme indiqué dans la section "Informations système".

Informations système

Lorsque vous appellerez le support technique d'IBM, vous devrez sans doute fournir des informations relatives à votre environnement.

Si le problème rencontré ne vous empêche pas de vous connecter, vous trouverez la plupart de ces informations sur la page A propos de, qui fournit des informations sur les applications IBM installées.

Vous pouvez accéder à la page A propos de en sélectionnant **Aide > A propos de**. Si vous ne pouvez pas accéder à la page A propos de, recherchez un fichier `version.txt` situé dans le répertoire d'installation de votre application.

Informations de contact du support technique d'IBM

Pour savoir comment contacter le support technique IBM, consultez le site Web de support technique IBM : (http://www.ibm.com/support/entry/portal/open_service_request).

Remarque : Pour entrer une demande de support, vous devez vous connecter avec un compte IBM. Ce compte doit être lié à votre numéro de client IBM. Pour en savoir plus sur l'association de votre compte à votre numéro de client IBM, accédez à **Ressources de support > ESS (Entitled Software Support)** dans le portail du support.

Remarques

Le présent document a été développé pour des produits et des services proposés aux Etats-Unis.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

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

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan, Ltd. 19-21, N

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales : LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

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

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA (IBM Customer Agreement), des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou

implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les exemples de programmes sont fournis "EN L'ETAT" sans garantie d'aucune sorte. IBM ne sera en aucun cas responsable des dommages directs, indirects, particuliers ou autres dommages liés à l'utilisation de ces programmes.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Marques

IBM, le logo IBM et ibm.com sont des marques ou des marques déposées d'International Business Machines Corp. dans diverses juridictions à travers le monde. Les autres noms de produit et service peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web Copyright and trademark information à www.ibm.com/legal/copytrade.shtml.

Règles de confidentialité et conditions d'utilisation

Les Logiciels IBM, y compris les Logiciels sous forme de services ("Offres Logiciels") peuvent utiliser des cookies ou d'autres technologies pour collecter des informations sur l'utilisation des produits, améliorer l'acquis utilisateur final, personnaliser les interactions avec celui-ci, ou dans d'autres buts. Un cookie est une donnée qu'un site Web peut envoyer à votre navigateur et qui peut ensuite être stockée sur votre ordinateur sous la forme d'une balise identifiant ce dernier. Bien souvent, aucune information personnelle identifiable n'est collectée par les Offres Logiciels. Si la présente Offre Logiciels utilise des cookies pour collecter des informations personnelles identifiables, des informations spécifiques sur cette utilisation sont fournies ci-dessous.

Selon la configuration déployée, la présente Offre Logiciels peut utiliser des cookies de session et des cookies persistants destinés à collecter le nom et le mot de passe des utilisateurs pour les fonctions de gestion des sessions et d'authentification, pour faciliter l'utilisation des produits, ou pour d'autres objectifs de suivi de l'utilisation ou fonctionnels. Ces cookies peuvent être désactivés mais leur désactivation élimine également la fonctionnalité qu'ils activent.

Diverses juridictions régulent la collecte d'informations personnelles via les cookies et autres technologies similaires. Si les configurations déployées de cette Offre Logiciels vous permettent, en tant que client, de collecter des informations permettant d'identifier les utilisateurs par l'intermédiaire de cookies ou par d'autres techniques, vous devez solliciter un avis juridique sur la réglementation applicable à ce type de collecte, notamment en termes d'information et de consentement.

IBM demande à ses clients (1) de fournir un lien clair et visible vers les conditions d'utilisation et la politique de protection des renseignements personnels du site Web du Client, ainsi qu'un lien vers la collecte de données et les pratiques d'utilisation d'IBM et du Client, (2) de signaler que les cookies et les images de pistage (clear gifs/web beacons) sont copiés sur l'ordinateur du visiteur par IBM au nom du Client, et de fournir une explication sur l'objectif et l'utilisation de ces technologies, et (1) selon les conditions requises par la loi, d'obtenir le consentement des visiteurs du site Web avant de placer les cookies et les images de pistage déposés par le Client ou par IBM au nom du Client sur leurs machines.

Pour plus d'informations sur l'utilisation à ces fins des différentes technologies, y compris celle des cookies, consultez les Points principaux de la Déclaration IBM de

confidentialité sur Internet à l'adresse <http://www.ibm.com/privacy/details/us/>
en dans la section intitulée "Cookies, Web Beacons and Other Technologies."



Imprimé en France