

IBM Unica Campaign
バージョン 8 リリース 6
2012 年 4 月 30 日

オプナー API 仕様



注

本書および本書で紹介する製品をご使用になる前に、53ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM Unica Campaign バージョン 8 リリース 6 モディフィケーション 0、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM Unica Campaign
Version 8 Release 6
April 30, 2012
Offer API Specification

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.5

© Copyright IBM Corporation 1998, 2012.

目次

第 1 章 概要	1
変更の概要サマリー	1
Client API を使用する既存の実装	2
新しい Client API の jar ファイル	2
新しい従属 jar ファイル	2
Client API コンストラクターでの変更	2
サポートするクラスのパラメーター化されたコンストラクターでの変更	2
WSDL を直接使用する既存の実装	3
WSDL の場所およびサービス URL	3
スタブおよびクラスの生成	3
生成されたスタブおよびサポートするクラスの使用	3
参照	4
要件	5
設計の概要	5
注意	6
第 2 章 データ型	9
WSReference	9
WSVersion	9
WSServiceInfo	9
WSAttributeTypeEnum	10
WSAttributeStatusEnum	10
WSAccessTypeEnum	10
WSSelectTypeEnum	10
WSRunStatusEnum	10
WSRunTypeEnum	11
WSAttribute	11
WSAttributeMetadata	12
WSCampaignInfo	14
WSComponentOrFolderInfo	14
WSTargetCellInfo	15
WSMetricsInfo	15
WSRunResults	15
WSOfferInfo	16
WSOfferCodeOrName	16
WSOfferValidationInfo	16
WSOfferTemplateInfo	17
WSBulkOfferInfo	17
WSOfferInfoStatus	17
第 3 章 一般的な例外	19
RemoteException	19
AuthenticationException	19
AuthorizationException	19
DataException	19
LockException	20
InvalidComponentException	20
InvalidAttributeException	20
AttributeNotFoundException	20

AttributeExistsException	20
CompositeException	20

第 4 章 CampaignServices API メソッド	21
サービス・メソッド	21
getServiceInfo	21
属性	21
getAttributesByName	22
updateAttributes	22
getAttributeMetadataByName	24
createAttributeMetadata	24
updateAttributeMetadata	25
deleteAttributeMetadata	27
キャンペーン	28
generateCampaignCode	29
createCampaign	29
listCampaignsByPage	30
ターゲット・セル・メソッド	31
createTargetCell	32
bulkCreateTargetCells	33
listTargetCells	34
bulkUpdateTargetCells	35
getRunResultsByCell	36
bulkDeleteTargetCells	37
分析	37
getCampaignMetrics	37
オファー、オファー・リスト、およびオファー・テンプレートのメソッド	38
listOffersAndFolders	39
searchOffersBasic	39
listOffersByPage	40
validateOffers	41
createOffer	42
retireOffers	43
deleteOffers	44
listOfferTemplates	44
bulkCreateOffers	45

第 5 章 API の使用	47
Client API の jar の使用	47
コード OfferAPI.java	47
WSDL の使用	49
パフォーマンスについての考慮事項	49
パッケージ化	50
IBM Unica 技術サポートへの連絡	51
特記事項	53
商標	55

第 1 章 概要

注: このドキュメントでは、IBM® Unica Campaign Services アプリケーション・プログラミング・インターフェース (CampaignServices と呼ばれる) のバージョン 3.0 のオファー部分を定義します。本書の中で説明されているオファー・サービスのみがサポートされます。

注: IBM Unica Campaign バージョン 8.2 以降にアップグレードする場合で、現在 IBM Unica Campaign Services API が実装されている場合は、AXIS バージョン 1.3 から AXIS2 1.4.1 へのアップグレードに伴い API が変更されるため、アプリケーション・コードの変更が必要となります。詳しくは、2 ページの『新しい従属 .jar ファイル』を参照してください。

このセクションには、以下の主要なトピックが含まれています。

- 『変更の概要サマリー』
- 2 ページの『Client API を使用する既存の実装』
- 3 ページの『WSDL を直接使用する既存の実装』
- 4 ページの『参照』
- 5 ページの『要件』
- 5 ページの『設計の概要』

変更の概要サマリー

- AXIS バージョン 1.3 から AXIS2 1.4.1 への SOAP エンジンの移行
- WSDL は、必須/オプションのパラメーターを処理する際の問題に対応するように再構成されました。
- Client API の .jar ファイルは、WSDL の変更に伴い変更されたので、生成されるスタブおよびクラスも変更されました。Client API メソッドのパラメーターは変更されていませんが、AXIS2 WSDL2Java コンバーターの使用のために、サポートする値オブジェクトのコンストラクターが変更されました。
- Web サービス URL は、次の場所を指すように変更されました。

`http://<host>:<port>/Campaign/services/CampaignServices30Service`

また、対応する WSDL は次の場所から取得できます。

`http://<host>:<port>/Campaign/services/CampaignServices30Service?wsdl`

これらの変更に伴い、現在 API を使用している場合は、アプリケーション・コードを変更する必要があります。詳しくは、Client API または WSDL のどちらを使用するかに応じて、以下のセクションを参照してください。

- 2 ページの『Client API を使用する既存の実装』
- 3 ページの『WSDL を直接使用する既存の実装』

Client API を使用する既存の実装

Client API の .jar ファイルを使用して Campaign Web アプリケーションと対話する場合、次のセクションには注意の必要な変更点が詳しく説明されています。

新しい Client API の .jar ファイル

Client API クラスの内部実装が変更されると、Java アプリケーションは次の場所にある新しい .jar ファイルを使用する必要があります。

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/  
CampaignServicesClient30.jar
```

新しいオファーの作成を示している Java の例は、47 ページの『コード OfferAPI.java』を参照してください。同じ例は、Campaign インストール環境の次の場所にもあります。

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/samples/OfferAPI.java
```

新しい従属 .jar ファイル

AXIS2 バージョン 1.4.1 へのアップグレードに伴い、Java アプリケーションも AXIS2 1.4.1 配布の .jar ファイルを使用するようにアップグレードする必要があります。それは、CampaignServicesClient30.jar がそれらの .jar ファイルに従属しているためです。従属するすべての .jar ファイルは、アプリケーションの Java クラスパスに含まれている必要があります、次の場所にある Campaign.war ファイル内に見つけることができます。

```
<CAMPAIGN_HOME>/Campaign.war
```

.jar ファイルを Campaign.war から抽出して、Java クラスパスに含めます。

Client API コンストラクターでの変更

Client API オブジェクトを構成するとき、次の例に示されているように Web サービス URL および例外シグニチャーを変更します。

```
try {  
    URL serviceURL = new URL(PROTOCOL, HOST, PORT,  
        "/Campaign/services/CampaignServices30Service");  
    CampaignServices30SoapClient client = new  
        CampaignServices30SoapClient(serviceURL, TIMEOUT);  
    } catch (RemoteException exception) {  
        exception.printStackTrace();  
    }  
}
```

サポートするクラスのパラメーター化されたコンストラクターでの変更

AXIS2 エンジンでは、生成されるクラスおよびスタブに、パラメーター化されたコンストラクターがなくなりました。代わりに、これらのクラスには、メンバー用の setter と getter を持つデフォルトの引数のないコンストラクターだけがあります。

つまり、

```
WSReference wsRef = new WSReference(WSComponentTypeEnum typeEnum, Long id);
```

は次のように変更されます。

```
WSReference wsRef = new WSReference();
wsRef.setComponentTypeEnum(typeEnum);
wsRef.setId(id);
```

WSDL を直接使用する既存の実装

Campaign Web サービスの WSDL は、サード・パーティー製のコンバーター・ツールを使用してクライアント・サイド・スタブおよびサポートするクラスを生成するために使用されます。WSDL を使用して Campaign Web アプリケーションと対話する場合、次のセクションには注意の必要な変更点が詳しく説明されています。ここに示す例は、Apache AXIS2 1.4.1 の WSDL2Java ツールを使用しています。

WSDL の場所およびサービス URL

IBM Unica Campaign 8.5.0 の Campaign Web サービスは、次の場所に配置されます。

```
http://host:port/Campaign/services/CampaignServices30Service
```

対応する WSDL は、次の場所から取得できます。

```
http://host:port/Campaign/services/CampaignServices30Service?wsdl
```

スタブおよびクラスの生成

Apache AXIS2 1.4.1 の WSDL2Java ツールは、スタブおよびサポートする Java クラスを WSDL から作成するために使用できます。サンプルの Ant タスクがここに示されています。

このツールは、類似の引数のセットを使用して、コマンド行からも使用できます。引数値は環境に合うように変更できます。

注: 次の WSDL2Java コンバーター例には、デフォルトの ADB バインディングが使用されます。

```
<java classname="org.apache.axis2.wsd1.WSDL2Java" fork="true">
  <classpath refid="axis2.class.path"/> <!--Class path having
AXIS2 libraries -->
  <arg value="-uri"/>
  <arg file="CampaignServices30.wsdl"/> <!--Actual location of
WSDL -->
  <arg value="-s"/> <!-- Generate sync style code -->
  <arg value="-Euwc"/> <!-- Takes care of generating Wrapper
java types for nillable = true elements. -->
  <arg value="-uw"/> <!-- Unwrap params -->
  <arg value="-u"/> <!-- Unpack classes -->
  <arg value="-ns2p"/> <!-- Namespace to package mapping. Customer
can have their own package names. -->
  <arg value="http://webservices.unica.com/campaign/CampaignServices/
3.0=com.unica.publicapi.campaign.campaignservices.soap.v30"/>
  <arg value="-o"/> <!-- Output directory -->
  <arg file="${autogen.java.dir}"/>
</java>
```

生成されたスタブおよびサポートするクラスの使用

スタブは以下の方法で使用できます。

```
CampaignServices30ServiceStub serviceStub = new
CampaignServices30ServiceStub(serviceURL);

serviceStub._getServiceClient().getOptions().setTimeoutInMillisecs
(webServiceTimeout); //Timeout in milliseconds.
```

オファーは以下の方法で作成できます。

```
try {
    WSAttribute[] wsAttributes = {
        WSAttributeUtils.getWSTextAttribute(
            IAttributeMetadata.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME,
            null,
            new String[]{"description " + System.currentTimeMillis()}),
    };

    // convert to WSAttributeArrays
    WSAttributeArrays wsAttributeArrays =
    WSAttributeUtils.makeWSAttributeArrays(wsAttributes);

    String offerName = "1st Offer"; //Name of the offer to be created.
    String templateName = "Offer Template"; //Existing offer template name.

    // make campaign Webservice call
    WSCreateOfferResponse wsResponse = getWebService().createOffer(
        WSHelper.getCreateOffer(authorizationLoginName, partitionName,
            Locale.US.toString(), securityPolicyName, name, folderID,
            templateName, wsAttributeArrays));
    } catch (Exception exception) {
        //Handle the Exception here.
    }

    // process status
    WSRequestStatus status = wsResponse.getStatus();
    // done
    WSOfferInfo offerInfo = wsResponse.getOfferInfo();
```

この例で、createOffer() はタイプ CreateOffer のパラメーターを 1 つだけ受け入れるようになりました。

AXIS2 エンジンでは、生成されるクラスおよびスタブにパラメーター化されたコンストラクターがなくなりました。詳しくは、2 ページの『サポートするクラスのパラメーター化されたコンストラクターでの変更』を参照してください。

参照

この仕様を準備するために、以下の参照が使用されました。

- 「Basic Profile Version 1.1」、Web Service Interoperability Organization (WS-I)、2006 年 4 月 10 日。 (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-0310.html>)
- 「SOAP 1.2 (draft)」、W3C Soap working group、2003 年 6 月 24 日 (<http://www.w3.org/TR/soap/>)
- 「JAX-RPC 1.1」、Sun Microsystems、2003 年 10 月 14 日 (<http://java.sun.com/webservices/jaxrpc/index.jsp>)
- Apache Web services working group (<http://ws.apache.org/axis2>)

要件

このセクションでは、設計において満たす必要のある要件について要約しています。 CampaignServices API は、以下の要件を満たす必要があります。

- クライアントが基礎となる実装の詳細を意識することなく、IBM Unica Campaign コンポーネントに対して、きめの細かい作成、ディスカバリー、読み取り、更新を行えるようにする
- 既存の IBM Unica Campaign GUI ベースのユーザーと共存し、それに与える影響を最小にする
- データ保全性を保証する
- IBM Unica Campaign のセキュリティー・アーキテクチャーをサポートする
- 保護された認証を含む、業界標準の SOAP をサポートする

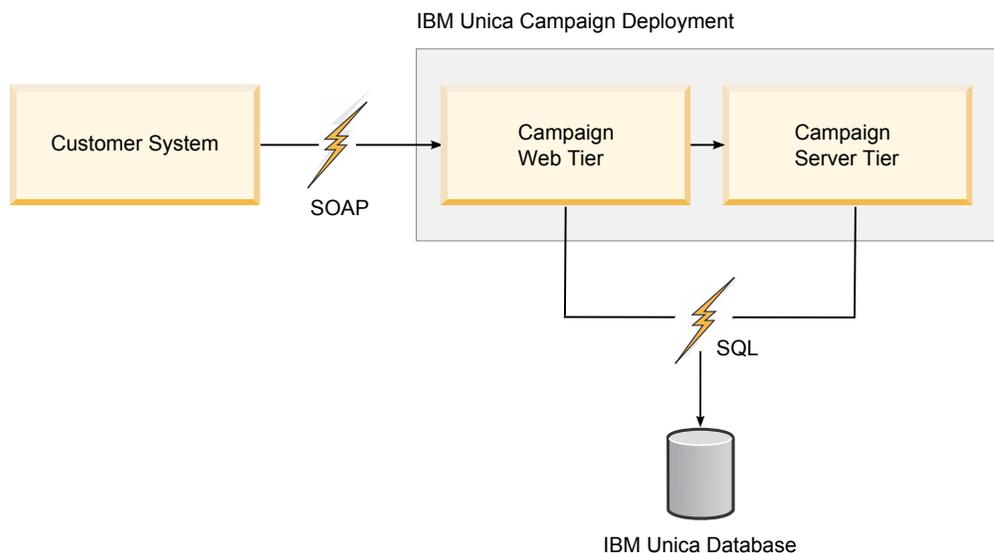
設計の概要

CampaignServices API は、実行中の IBM Unica Campaign アプリケーション・インスタンスのクライアント・ビューを提供するファサードです。 IBM Unica Campaign の機能のサブセットだけが公開されていますが、 Campaign 機能の主な側面を活用するためには十分です。この API は、IBM Unica Campaign Web ユーザーおよび他の API スレッドと共に使用するよう設計されています。

通常、この API は、キャンペーン、オファー、およびターゲット・セルのコンポーネントで、以下のタイプの操作をサポートします。

- コンポーネントの作成
- コンポーネントのディスカバリー
- コンポーネントの削除
- コンポーネント属性および属性メタデータの作成、検査、および変更
- フローチャート実行結果の取得

次の図は、CampaignServices 3.0 の配置のサンプルを示しています。



注意

このセクションでは、設計に関する特定の点について取り上げます。

バージョンおよび後方互換性

通常、CampaignServices API の将来のバージョンは、以前のバージョンで同じメジャー・バージョン番号を共有するすべてのマイナー・リリースやメンテナンス・リリースとの互換性を持ちます。ただし、IBM Unica は、業務上または技術上の問題によりそうすることが正当化される場合に、「ドット・ゼロ」(x.0) のメジャー・リリースで以前のバージョンとの互換性を中断する権利を保有します。

次のいずれかの変更が行われた場合に、この API のメジャー・バージョン番号が増分します。

- データの変換処理の変更
- ビジネス・ロジックつまりサービス・メソッドの機能の変更
- メソッドのパラメーターまたは戻りの型 (またはその両方) の変更

次のいずれかの変更が行われた場合に、API のマイナー・バージョン番号が増分します。

- 新しいメソッドの追加
- 新しいデータ型の追加 (新しいメソッドでの使用に制限される)
- 列挙型への新しい型の追加
- 新しいバージョンのインターフェースの定義

特に、IBM Unica は公開された WSDL、SOAP クライアント、および SOAP オフラーの実装に使用される Apache Axis のバージョンのサポートを、少なくとも IBM Unica の次のメジャー・リリースまで続けます。実際には、これは複数のバージョン特定の Web サービスを同時にサポートすることにより実現されます。(IBM Unica は、このサービスの複数のバージョンを既に内部的にサポートしています。)

ユーザー認証

認証は、ユーザーの身元を確立するための処理です。

注: このリリースでは、ユーザー認証はクライアント・アプリケーションが担当します。

ユーザー許可

許可は、API によって公開されているコンポーネントおよび操作と比較して、認証されたユーザーが持つ権限に関する処理です。

ユーザーの認証が正常に行われても、キャンペーンのサマリー情報の編集など、いくつかの操作を行うためには十分な権限がない場合もあります。この場合、API メソッドは `AuthorizationException` をスローします。

ロケール

API 要求には、特定の要求の実行に使用するためのロケールを定義する、オプションの `requestedLocale` パラメーターがあります。このパラメーターを定義しない場合は、API はデフォルトの IBM Unica ユーザーの優先ロケールを使用します。要

求されたロケールでメッセージや他のローカライズされたテキストを返すときには、通常の Java ベストエフォート・マッチング・アルゴリズムが使用されます。

このパラメーターのタイプは `java.util.Locale` クラスです。

注: キャンペーンについての説明など、いくつかのユーザー指定テキストには、そのテキストを指定したユーザーのロケールが適用されます。IBM Unica Campaign は、そのようなデータをローカライズしようとはしません。通知、警告、およびエラー・メッセージのみが API によってローカライズされます。

状態管理

CampaignServices API はステートレスです。つまり、複数の呼び出しの間でクライアントごとの情報が API によって保存されることはありません。

特定の API 呼び出しによって Campaign が管理する基礎となるコンポーネント・インスタンスの状態が変更され、その状態変更がデータベースに維持されている可能性は確かにあります。

第 2 章 データ型

このセクションでは、CampaignServices API によって使用されるパブリック・データ型を定義します。

WSReference

データベース ID を囲む単純なラッパー。

- **componentTypeEnum**: ID によって指定されるコンポーネント・タイプを示す列挙型。以下のいずれかです。
 - FOLDER
 - CAMPAIGN
 - FLOWCHART
 - TCS_CELL
 - OFFER
 - OFFER_LIST
 - OFFER_TEMPLATE
- *id*: 参照を示すデータベース固有の一意の数値 ID を定義する *Long* 値。

WSVersion

バージョンのさまざまなコンポーネントを取得するラッパー・タイプ。以下が含まれます。

- *major*: メジャー・バージョン番号を定義する整数。例えば完全なバージョンが 8.1.2.3 であれば、「8」になります。
- *minor*: マイナー・バージョン番号を定義する整数。例えば完全なバージョンが 8.1.2.3 であれば、「1」になります。
- *maintenance*: バージョンのメンテナンス番号 (該当する場合) を定義する整数 (オプション)。例えば完全なバージョンが 8.1.2.3 であれば、「2」になります。API バージョンでは提供されません。
- *patch*: バージョンのパッチ・リリース番号 (該当する場合) を定義する整数 (オプション)。例えば完全なバージョンが 8.1.2.3 であれば、「3」になります。API バージョンでは提供されません。

WSServiceInfo

サービスに関する情報を囲む単純なラッパー・タイプ。これには次のフィールドがあります。

- *apiVersion*: サービスによってサポートされる API の最新バージョンを定義する、*WSVersion* インスタンス。(*apiVersion* にはメジャーおよびマイナー・バージョン情報のみが含まれることに注意してください。)

- *campaignVersion*: 基礎となる IBM Unica Campaign インスタンスの完全なバージョンを定義する *WSVersion* インスタンス。
- *name*: サービスの内部名。「CampaignServices30Service」など。

WSAttributeTypeEnum

可能なすべての属性タイプを定義する列挙型。次のいずれかになります。

- STANDARD: Campaign によって定義される、標準または基本属性。
- CUSTOM: 別の IBM Unica アプリケーション、顧客、または他のサード・パーティーによって定義される属性。
- INPUT_PARAMETER: 入力パラメーター。IBM Unica Campaign フローチャートの実行に使用される属性など。
- OUTPUT_PARAMETER: 出力パラメーター。IBM Unica Campaign でのフローチャート実行により完成する値を持つ属性。

WSAttributeStatusEnum

可能なすべての属性ステータス・コードの列挙。次のいずれかになります。

- ACTIVE: 属性はアクティブで、任意に使用できます。
- RETIRED: 属性はサービスから削除されているので、使用できません。

WSAccessTypeEnum

次のいずれかになる、可能なすべての属性値アクセス・タイプを定義する列挙型。

- READ_ONLY: 属性値の読み取りおよび表示ができますが、変更はできません。
- READ_WRITE: 属性値の読み取り、表示、および変更ができます。

属性アクセスはセキュリティ権限に対して付加的なものなので、例えばクライアント・ユーザーのセキュリティ・ポリシーが特定の属性に対する読み取りアクセスを拒否している場合、属性アクセスがそのセキュリティ設定をオーバーライドすることはできません。実際には、API は属性をクライアントに返しませんが、

WSSelectTypeEnum

特定の属性値に対するすべての可能な選択タイプを定義します。以下のいずれかになります。

- NONE: 選択がありません (*hasOptions* は false です)。
- SINGLE_SELECT: 可能なオプションのリストから一度に 1 つの属性オプションのみを選択できます (属性 *hasOptions* の場合にのみ有効です)。
- MULTIPLE_SELECT: SINGLE_SELECT と似ていますが、一度に 1 つ以上のオプションを選択できます。

WSRunStatusEnum

すべての可能なフローチャート、ブランチ、またはセルの実行ステータスの列挙型。以下のいずれかになります。

- NOT_STARTED: 実行はスケジュール設定されましたが、まだ開始されていません。
- RUNNING: 実行が進行中です。
- CANCELLED: 実行が Campaign ユーザーまたはこの API によってキャンセルされました。
- SUCCEEDED: 実行が正常に終了しました。
- FAILED: 実行が失敗しました。エラー詳細は別途報告されます。(15 ページの『WSRunResults』を参照してください。)

WSRunTypeEnum

すべての可能な実行タイプの列挙型。以下のいずれかになります。

- NOT_RUN
- TEST_RUN
- PRODUCTION_RUN
- RUN_SKIPPED
- TEST_FLOWCHART
- PRODUCTION_FLOWCHART
- TEST_BRANCH
- PRODUCTION_BRANCH
- TEST_PROCESS
- PRODUCTION_PROCESS

WSAttribute

属性は、キャンペーンの名前などの標準データ、*gender* などのフローチャート実行の入力パラメーター、または別の IBM Unica アプリケーションや IBM Unica の顧客によって指定される任意のカスタム・データの、いずれかの任意のデータを API によってアクセス可能なコンポーネント・インスタンスに添付するための、単純で拡張可能なメカニズムを提供します。

注: この API で、属性は Campaign カスタム属性だけでなく、ほとんどのコンポーネント・データをモデル化するために使用されます。

コンポーネントには、*AttributeMap* と呼ばれる特別な型付きマップとして CampaignServices API により公開されている、多くの属性が通常は関連付けられます。10 進 (倍精度数値) データを含む属性では、属性データが *WSDecimalAttribute* などの厳密に型指定された具象クラスとして API 全体で表されます。

各属性には、以下が含まれます。

- *Name*: 属性の一意的名前。この名前は、属性およびそのメタデータに、それが出現するコンポーネント・インスタンス内でアクセスするためのキーとなります。名前のフォーマットは定義されていません。場合によっては、サービス、クライアント、または IBM Unica Campaign ユーザーによって、フォーマットが割り当てられることもあります。

通常この名前は、Campaign やクライアント・ユーザーに表示される表示名ではありません。それは *uacDescription* などのように API によって標準化されることもあり、フローチャートを公開するときに IBM Unica Campaign によって割り当てられることもあり、カスタム属性を定義するときに IBM Unica アプリケーションまたは顧客によって割り当てられることもあります。ただし、すべてのケースで、名前は一意であることが保証されます。

- **Metadata:** (オプション) 属性のデータに関する情報。例えば、値のデータ型、表示名、説明、プロンプト、デフォルト値、選択タイプ、長さ (テキスト)、精度 (小数)、オプション (単一または複数選択の場合) などがあります。
『WSAttributeMetadata』を参照してください。
- **Values:** 厳密に型指定されたゼロ以上の値オブジェクトの配列。値項目は具象属性クラスによって指定されます。各値の型は等しく、属性のメタデータ項目の型定義に適合していなければなりません。ただし、すべての属性が複数値をサポートするわけではありません。

以下の具象属性タイプがサポートされます。

- **WSBooleanAttribute:** 値がブール、つまり *true* または *false* の属性。
- **WSIntegerAttribute:** 整数値 (*java.lang.Long*)。
- **WSDecimalAttribute:** 倍精度 10 進数値 (*java.lang.Double*)。
- **WSCurrencyAttribute:** オプションの通貨値の ISO 4217 通貨コード (US ドルを表す「USD」など) と、*Double* として取得される通貨値とを含む、複合の通貨値。通貨コードが指定されない場合、IBM Unica Campaign で使用されるデフォルトが想定されます。

国、通貨記号、およびコードのリストは、<http://www.xe.com/symbols.php> を参照してください。通貨値に使用されるロケールが、ユーザーの優先ロケールとは異なる可能性があることに注意してください。
- **WSCalendarAttribute:** いくつかのタイム・ゾーンおよびロケールで、値がカレンダー日付つまり *datetimes* となる。
- **WSTextAttribute:** Unicode テキストの文字列 (おそらく NULL または空)。

注: 可能な属性のリストは、通常はコンポーネントのタイプごとに異なりますが、リストがオーバーラップすることもあります。

WSAttributeMetadata

WSAttributeMetadata は、特定の型付き属性のデータに関する情報、例えば値データ型、ローカライズされたテキスト (表示名、説明、プロンプトなど)、そのデフォルト値、暗黙的値の範囲、選択タイプ、オプション (単一または複数選択の場合)、その他を定義します。属性の場合と同様に、属性メタデータは厳密に型設定されています。そのため、例えば WSDecimalAttribute *myNumber* には WS *DecimalAttributeMetadata* がバインドされている必要があり、属性値、メタデータのデフォルト値、可能なオプション値を含むすべての値には *Double* 型が設定されます。

説明、ラベル、および他の属性メタデータのテキストは、通常はローカライズされます。ただし、ユーザー指定のテキストはユーザーが入力した状態でのみ表示されることがあります。各 API 呼び出しには、クライアント・コードが、特定のユーザ

ーがローカライズされたメッセージの表示を希望する場合のロケールを定義するために使用できる、要求されたロケールが含まれます。要求に応じるために、通常の Java ロケール・フォールバック・ポリシーが使用されます。

WSAttributeMetadata には、以下の項目が含まれます。

- *name*: 標準またはカスタムの属性名。また、このメタデータにバインドされる属性によって使用される名前。標準属性はシステムによって定義され、予約名スペースに標準名が入っています (つまり、「uac」プレフィックスを使用します)。カスタム名は他の任意の命名規則を使用できます。

注: 属性名は一意である必要があり、ローカライズされず、長さ制限 (文字コンテンツとデータベースに依存する) があります。名前はケース・インセンシティブであり、Unicode 文字、数字、および下線文字「_」の任意の組み合わせで構成できますが、先頭を数字にすることはできません。

- *description*: 属性の説明 (オプション)。ツールチップや他のユーザー・インターフェースを表示するために適しています。
- *Predicates*: 属性について説明する、以下に示すようなさまざまな述部。
 - *isRequired*: 属性が必須の場合には true。
 - *isInternal*: 属性がシステムによって定義され、内部使用に限定される (ユーザーには表示されない) 場合は true。
 - *isGenerated*: ターゲット・セル・コードなど、属性の値がコンポーネントの作成時に IBM Unica Campaign によって自動生成される場合は true。通常、生成される値の *accessTypeEnum* は READ_ONLY となります。
 - *hasOptions*: 属性にオプションがある場合は true。このメタデータに対してオプションが定義されていることと、*selectTypeEnum* が SINGLE_SELECT または MULTIPLE_SELECT であることが暗黙に示されます。
- *typeEnum*: 属性のタイプを定義する *WSAttributeTypeEnum*。STANDARD や CUSTOM など。
- *statusEnum*: 属性のステータスを定義する *WSAttributeStatusEnum*。ACTIVE など。
- *accessTypeEnum*: 属性値へのアクセス権限のタイプを定義する *WSAccessTypeEnum*。READ_ONLY など。
- *selectTypeEnum*: 属性に使用される選択のタイプを定義する *WSAccessTypeEnum*。SINGLE など。キャンペーンおよびセル属性の場合、またはオプションが指定されない場合には、NONE でなければなりません。
- *componentTypeEnum*: API によって公開されるすべての可能なキャンペーン・コンポーネントの *WSComponentTypeEnum*。CAMPAIGN、FOLDER など。
- *defaultValue* (フローチャートのみ): 属性のタイプのデフォルト値 (オプション)。この値は、具象属性メタデータ・クラスによって提供されます。例えば、*WSTextAttributeMetadata* のデフォルト値のタイプは文字列となります。(属性値に関する説明を参照してください)。フローチャート以外のコンポーネントについては、デフォルト値は定義されません。
- *options*: この属性のオプションのリスト (オプション)。属性の複数のオプションを一緒に指定することにより、その属性の暗黙的値のセットを正確に定義できます。各オプションは厳密に型指定されているので、例えば *WSTextAttributeMetadata* には *WSTextAttributeOption* だけをバインドできます。

注: オプションについては制約があり、テキスト属性だけがサポートされます。

各オプションは、以下を定義します。

- *prompt*: プルダウン・メニューに適したオプションのプロンプト。性別属性のオプションとして「男性」など。メタデータのプロンプトとは異なり、オプションの表示名には句読点が通常含まれないことに注意してください。
- *description*: オプションのローカライズされた説明。「性別が男性の人」など。ツールチップのテキストに適しています。
- *isDefault*: この特定のオプションがデフォルトの場合は **true**。
MULTIPLE_SELECT 選択タイプでは、複数のオプションをデフォルトとしてマーク付けできます。
- *value*: 型付きのオプションの値。属性メタデータ *defaultValue* と同様に、この値は具象オプション・サブクラスによって提供されます。例えば、*WSDecimalAttributeOption*'s 値の方は **Decimal** です。(属性値に関する説明を参照してください)。上記の性別 の例を続けると、値は文字列 "m" で宣言すること (*WSTextAttributeOption*)、または数字コード 123 で宣言すること (*WSDecimalAttributeOption*) ができます。

WSCampaignInfo

キャンペーン属性データを囲む単純なラッパー・タイプ。

これには次のフィールドがあります。

- *reference*: キャンペーンの参照。
- *name*: キャンペーン名 (*uacName*)。一意的であることは保証されていません。
- *description*: キャンペーンの説明 (オプション)(*uacDescription*)。
- *campaignCode*: 一意的なキャンペーン・コード (*uacCampaignCode*)。クライアントまたは Campaign によって割り当てられます。

WSComponentOrFolderInfo

表示名やその参照など、ラップされたキャンペーンまたはフォルダーの属性データの組み合わせを含みます。

これには次のフィールドがあります。

- *reference*: コンポーネントまたはフォルダーの参照。
- *name*: コンポーネントまたはフォルダーの名前 (*uacName*)。一意的であることは保証されていません。
- *description*: コンポーネントまたはフォルダーの説明 (オプション)(*uacDescription*)。
- *componentCode*: コンポーネントの一意のコード。または、フォルダーの場合には NULL。

WSTargetCellInfo

ターゲット・セルの行属性のデータを囲む単純なラッパー。

これには次のフィールドがあります。

- *reference*: セルの参照。
- *name*: セル名 (*uacName*)。一意的であることは保証されていません。
- *description*: セルの説明 (オプション)(*uacDescription*)。
- *cellCode*: セル・コード (*uacCellCode*)。クライアントまたはキャンペーンによって割り当てられます。IBM Unica Campaign *DuplicateCellCodesAllowed* 構成パラメーターを *false* に設定することにより、セル・コードを強制的に一意にすることができることに注目してください。
- *flowchartName*: セルがバインドされるフローチャートの名前 (オプション)。

WSMetricsInfo

コンタクトの数を含む、キャンペーン分析データを囲む単純なラッパー・タイプ。

これには次のフィールドがあります。

- *totalContacts*: コンタクトの合計数を示す *long* 値。
- *responses*: *WSMetricsResponse* インスタンスの型付きリスト。以下のように、各インスタンスは 1 つのレスポンスに関するコンタクト情報を定義します。
 - *typeCode*: レスポンス・タイプ・コードを定義する文字列。電話によるコンタクトを示す *PHC* など。
 - *count*: このコンタクトが発生した回数を示す *long* 値。

WSRunResults

フローチャート、プロセス・ボックス、またはセルの実行 (まだ進行中の可能性もある) の結果を囲むラッパー・タイプ。この結果は、実行のステータス、フローチャート実行の開始と終了の日時、およびカウントを含みます。

これには次の項目が含まれます。

- *sourceReference*: 実行結果のソースの参照 (オプション)。これにより、実行結果が取得されたコンテキストに応じて、フローチャート、フローチャート・プロセス・ボックス、またはターゲット・セルを参照できます。いずれの場合でも、残りの実行結果データはこのソースを参照します。
- *flowchartName*: 実行したフローチャートの名前。
- *flowchartId*: フローチャートのデータベース ID。
- *runId*: 実行のデータベース ID。
- *typeEnum*: 結果を生成した実行の種類を定義する列挙型。
PRODUCTION_PROCESS など (*WSRunTypeEnum* を参照してください)。
- *statusEnum*: 実行ステータスを定義する列挙型。*RUNNING* など (*WSRunStatusEnum* を参照してください)。
- *statusCode*: 整数の状況コード (オプション)。
- *statusMessage*: 状況メッセージ (オプション)。

- *startDate*: 実行が開始された時点を示すカレンダーの日時 (オプション)。実行が開始されていない場合は、NULL になります。
- *endDate*: *startDate* と似ていますが、実行が (失敗でも成功でも) 終了した日時である点が異なります。実行が開始されていない場合やまだ終了していない場合は、NULL になります。
- *count*: 実行によって選択されたコンタクトの合計数 (オプション)。実行が完了していない場合は、ゼロまたは NULL になります。

WSOfferInfo

オファーまたはオファー・リストの属性データを囲む単純なラッパー・タイプ。

これには次のフィールドがあります。

- *reference*: オファーまたはオファー・リストの参照。
- *name*: オファーまたはオファー・リストの名前 (*uacName*)、一意的であることは保証されていません。
- *description*: 説明 (オプション)(*uacDescription*)。
- *offerCode*: オファーの場合はオファー・コード (*uacOfferCode*)、またはオファー・リストの場合は NULL。(一意的であることは保証されていません。)

WSOfferCodeOrName

オファー・コードまたはオファー・リスト名のデータを囲む単純なラッパー・タイプ。

これには次のフィールドがあります。

- *isCode*: *codeOrName* 項目が、オファー・コード (true) またはオファー・リストの名前 (false) のどちらに推定されるかを示すブール値。
- *codeOrName*: オファーの場合は一意のオファー・コード (*uacOfferCode*)、またはオファー・リストの名前。

WSOfferValidationInfo

オファー検証情報を囲む単純なラッパー・タイプ。

これには次のフィールドがあります。

- *errorCode*: NULL ではない場合、英数字の検証エラー・コードを定義します。エラー・コードについては、IStandardDefinitions クラスを参照してください。
- *errorMessage*: エラー (発生した場合) を説明するローカライズされたメッセージ (オプション)。
- *codeOrName*: 検証されたオファー・コードまたはオファー・リスト名。
- *reference*: 有効な場合、オファーまたはオファー・リストの参照。

WSOfferTemplateInfo

オファー・テンプレートのデータを囲む単純なラッパー・タイプ。

これには次のフィールドがあります。

- *reference*: オファー・テンプレートの参照。
- *name*: オファー・テンプレートの名前。一意的であることが保証されています。
- *description*: 説明 (オプション)(*uacDescription*)。
- *offerTemplateID*: 一意のオファー・テンプレート・データベース ID。

WSBulkOfferInfo

大量にオファーを作成するために使用します。

これには次のフィールドがあります。

- *offerName*: 作成されるオファーの名前。
- *attributes*: オファー属性を示す *WSAttribute* タイプの配列。

WSOfferInfoStatus

大量のオファーの作成のステータスを示す、*bulkCreateOffers()* API メソッドの戻りの型。

これには次のフィールドがあります。

- *name*: オファーの名前。
- *code*: オファー・コード。オファーの作成が失敗した場合には NULL になります。
- *description*: オファーの説明。
- *reference*: 作成されたオファーの *WSReference*。オファーの作成が失敗した場合には NULL になります。
- *status*: オファー作成のステータスを示す *WSRequestStatus* のインスタンス。

第 3 章 一般的な例外

このセクションでは、CampaignServices API によってスローされるいくつかの一般的な例外について説明します。

すべてのローカライズされた例外のメッセージは、IBM Unica Campaign で使用可能な場合、要求されたロケールに置かれます。通常の Java ロケール・フォールバック・ポリシーが適用されます。

RemoteException

注: このアイテムは、SOAP インターフェースにのみ適用されます。

API に対するすべての SOAP 呼び出しは、SOAP エンベロープ処理レイヤー (Axis) での問題や何かの理由による Web サービス WSDL で定義された制約の違反など、システム・レベルのエラーが検出された場合に、*RemoteException* をスローすることがあります。

DataException など、チェック済みおよび未チェックの普通の API 例外は、*RemoteException* としてではなくエラー状態として返されます。

詳しくは、SOAP インターフェースのセクションを参照してください。

AuthenticationException

ユーザーを指定の Campaign パーティションに対して認証できません。Marketing Platform で、ユーザー役割の設定を確認してください。

AuthorizationException

ユーザーは、要求された操作の実行を許可されていません。この例外はどの API メソッドでもスローできるので、宣言されません (検査されません)。IBM Unica Marketing Platform でユーザーの役割に割り当てられている権限を確認してください。

DataException

Campaign で、基礎となるデータベース・レイヤーに重大な例外が生じた (チェックされていません)。

詳しくは、Campaign フローチャートおよびリスナー・ログを確認してください。

LockException

別のユーザーによって編集されているコンポーネント (フローチャートなど) をクライアントが更新しようとするときにスローされる、一時的な例外。通常は、幾らかの時間待機して操作を再試行することにより、この例外から回復できます。ただし、再試行のロジックに関してはクライアントが担当します。

InvalidComponentException

無効または不明なコンポーネント (キャンペーン、フローチャート、ターゲット・セルなど) を参照しようとした。例外の `getComponentReference()` メソッドは、問題のコンポーネントの参照を返します。

InvalidAttributeException

クライアントが無効な属性 (正しくないデータ型が使用されている、許可されていない配列の値が使用されているなど) を指定または参照すると、例外がスローされます。例外の `getAttributeName()` メソッドは問題のある属性の名前を返し、`getAttributeValue()` は値を返し、`getComponentReference()` はコンポーネント (またはバルク・インデックス) を示します。

AttributeNotFoundException

クライアントが不明な属性 (キャンペーン、フローチャート、ターゲット・セル、その他) を参照しようとするとき常にスローされます。例外の `getAttributeName()` メソッドは、一致しない属性の名前を返します。`getComponentReference()` はコンポーネント (またはバルク・インデックス) を示します。

AttributeExistsException

クライアントがコンポーネントに重複する属性のメタデータを定義しようとするときスローされます。例外の `getAttributeName()` メソッドは、重複する属性の名前を返します。`getComponentReference()` はコンポーネント (またはバルク・インデックス) を示します。

CompositeException

CompositeException は、複数のエラーを呼び出し元に報告して戻すために、いくつかの API によって使用されます。これには通常、複数の原因がバインドされます。すべての原因は、発生した順序によるリストとしてキャプチャーされます。例外の `getCauseList()` メソッドはこのリストを返します。そのリストをさらに検査して、各エラーの詳細を知ることができます。

注: 通常、API は正常に完了するか、または *CompositeException* をスローする前まで作業をロールバックします。例えば、31 ページの『ターゲット・セル・メソッド』で説明されているバルク・ターゲット・セル・スプレッドシート API を参照してください。

第 4 章 CampaignServices API メソッド

このセクションでは、CampaignServices 3.0 API によってエクスポートされる基本メソッドを定義します。

サービス・メソッド

API は、サービス自体についての識別情報を示す手段を提供します。

getServiceInfo

```
WSServiceInfo getServiceInfo()  
    throws CampaignServicesException;
```

サポートされる最新の API バージョンや、基礎となる IBM Unica Campaign インスタンスの完全なバージョンといった、サービスに関する情報を返します。

注: この呼び出しではクライアント情報は不要で、セキュリティー許可は適用されていません。

パラメーター

なし。

戻り値

WSServiceInfo インスタンスを返します。

エラー

なし。

属性

ほとんどのコンポーネント・インスタンス・データは、API によって属性 または属性メタデータ として公開されます。属性メタデータ定義が IBM Unica Campaign (キャンペーン・カスタム属性など) に対してグローバルなケースもあれば、それらが特定のコンポーネント (フローチャート・ユーザー変数など) に制限されるケースもあります。

特に明記されていない限り、クライアントに十分なセキュリティー権限がある場合には、すべての属性が読み取り可能です。

注: アクティブでクライアントからアクセス可能なコンポーネントのみが、この API によって公開されます。公開がサポートされるのは、オファー属性、オファー・テンプレート、およびメタデータに限定されています。

getAttributesByName

```
Map<String, WSAttribute>  
getAttributesByName(String userCredential, String partitionName,  
    Locale requestedLocale,  
    WSReference reference,  
    String[] names)  
throws CampaignServicesException;
```

指定されたコンポーネント・インスタンス (空の場合もある) に関連付けられた、名前の指定された属性を取得します。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。指定されない場合は、IBM Unica ユーザーのロケール設定が使用されます。必要な場合は、通常のロケールのデフォルト化アルゴリズムが適用されます。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。定義されない場合、デフォルトのパーティションが使用されます。

reference: 必要な属性を含む、コンポーネント・インスタンスの参照。参照が無効であるか、コンポーネントが存在しない場合、*InvalidComponentException* がスローされます。

names: 取得する属性の名前 (表示名ではありません) の配列 (オプション)。指定されない場合、すべての属性が返されます。名前の指定された属性が存在しない場合、*AttributeNotFoundException* をスローします。

戻り値

ゼロ以上の属性の型付きマップ。属性名はマップのエントリー・キーで、属性インスタンスはエントリー値です。

エラー

InvalidComponentException、*AttributeNotFoundException*

AuthorizationException、*DataException*

注: これらの例外すべては、*CampaignServicesException* 内でラップされます。

updateAttributes

```
void updateAttributes(String userCredential, String partitionName,  
    Locale requestedLocale, WSReference reference,  
    boolean allowCreate,  
    WSAttribute[] attributes)  
throws CampaignServicesException;
```

コンポーネント・インスタンスの 1 つ以上の属性を、指定の属性値で更新します。

更新のロジック

更新のロジックは、以下のとおりです。

提供された属性マップに含まれる属性ごとに、以下を行います。

1. 属性名が既存の属性と一致する場合、その値項目を提供された値項目で上書きすることを試行します。
2. 属性がまだ存在しない場合、`allowCreate` が `true` でそのメタデータが既知であれば、属性を作成します。これは、グローバル属性メタデータおよびインスタンス属性（フローチャートを除く）に適用されます。
3. 値タイプ、または属性のメタデータ定義の他の側面が一致しない場合、または提供された 1 つ以上の値が無効であったり範囲外であったりする場合は、`InvalidAttributeException` をスローします。
4. または、名前属性が存在しない場合、`AttributeNotFoundException` をスローします。

注: 例外が生じた場合は、いずれの更新もコミットされません。

この特定のメソッドは、新しいカスタム属性を定義することをサポートしません。そのためには `createAttributeMetadata()` メソッドを使用します。

すべてのケースで、属性の更新操作は通常のセキュリティ制約および検証に従います。特定のコンポーネント・インスタンスに必要な属性や正しいタイプなどは、クライアントが決定する必要があります。

パラメーター

`userCredential`: クライアント・ユーザー資格情報。

`requestedLocale`: この要求で使用するロケール（オプション）。

`partitionName`: 使用するキャンペーン・パーティションの名前（オプション）。

`reference`: 更新する属性を含むコンポーネント・インスタンスの参照。

`allowCreate`: コンポーネントの属性がまだ存在しない場合、それを新しく作成するかどうかを指定します。（『更新のロジック』を参照してください。）

`attributes`: 更新する属性の配列。属性名は更新する属性を見つけるために使用され、新しい値は、既存の属性の値を適切なタイプの単一オブジェクトまたは配列として更新するために使用されます（該当する場合）。（19 ページの『第 3 章 一般的な例外』を参照してください。）

戻り値

なし。

エラー

`InvalidComponentException`、`AttributeNotFoundException`、`InvalidAttributeException`

`AuthorizationException`、`DataException`

getAttributeMetadataByName

```
Map<String, WSAttributeMetadata>  
getAttributeMetadataByName(String userCredential,  
    String partitionName, Locale requestedLocale,  
    WSReference reference, String[] names)  
throws CampaignServicesException;
```

特定のコンポーネントやテンプレートにバインドされた、またはグローバルに定義されている、名前の指定された属性メタデータ定義を取得します。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

reference: 必要な属性メタデータを含むコンポーネントまたはテンプレートの参照 (オプション)。 `ComponentTypeEnum` のみが指定された場合、取得はそのタイプのコンポーネントに制限されます。参照がまったく指定されない場合、取得はすべてのコンポーネント・タイプについて、すべてのグローバル・メタデータ定義を返します。指定された参照が無効な場合は、`InvalidComponentException` をスローします。

names: 取得する属性メタデータの名前の配列 (オプション)。指定されない場合、コンポーネントのすべてのメタデータ、または参照が指定されない場合にはグローバルに定義されたメタデータが返されます。指定された 1 つ以上の属性メタデータ定義が存在しない場合は `AttributeNotFoundException` をスローします。

戻り値

ゼロ以上の属性メタデータ定義の型付きマップ。属性名はマップのエントリー・キーで、属性のメタデータはエントリー値です。

エラー

`InvalidComponentException`、`AttributeNotFoundException`

`AuthorizationException`、`DataException`

createAttributeMetadata

```
void createAttributeMetadata(String userCredential,  
    String partitionName,  
    Locale requestedLocale, WSReference reference,  
    WSAttributeMetadata[] attributeMetadata)  
throws CampaignServicesException;
```

1 つ以上の新しい属性メタデータ定義を作成し、オプションでそれらを特定のコンポーネントまたはテンプレートにバインドします。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

reference: メタデータをバインドする必要があるコンポーネントまたはテンプレートの参照 (オプション)。指定しない場合、作成されるメタデータ定義はグローバルになります。無効な参照が指定された場合は、*InvalidComponentException* をスローします。

attributeMetadata: バインドする属性メタデータ定義の配列。指定された 1 つ以上のメタデータが既にコンポーネントにバインドされている場合、つまり名前が一意ではない場合は、*AttributeExistsException* をスローします。指定された 1 つ以上のメタデータに問題がある場合、つまり内部的に不整合である場合は、*InvalidAttributeException* をスローします。

戻り値

なし。

エラー

InvalidComponentException、*AttributeExistsException*、*InvalidAttributeException*

AuthorizationException、*DataException*

updateAttributeMetadata

```
void updateAttributeMetadata(String userCredential,  
    String partitionName,  
    Locale requestedLocale, WSReference reference,  
    boolean allowCreate,  
    WSAttributeMetadata[] attributeMetadata)  
    throws CampaignServicesException;
```

指定されたコンポーネントまたはテンプレートの 1 つ以上の属性メタデータ定義を更新します。オプションで、必要であれば新しいメタデータ定義を作成します。

更新のロジック

更新のロジックは、以下のとおりです。

提供された配列に含まれる属性メタデータ定義ごとに、以下を行います。

1. 属性名がコンポーネントにバインドされた既存のメタデータと一致しない場合、*allowCreate* パラメーター値に基づいて以下を行ってください。
 - a. *True*: 新しいメタデータ定義を作成します。*createAttributeMetadata()* 要求を使用した場合と機能的には同じです。
 - b. *False*: *AttributeNotFoundException* をスローします。
2. 属性メタデータのデータ型が異なる場合、*InvalidAttributeException* をスローします。
3. 指定されたメタデータの項目値を使用して、既存の属性メタデータ定義の上書きを試行します。該当しない場合は、*InvalidAttributeException* をスローします。以下の更新だけがサポートされます (該当しない場合は、*InvalidAttributeException* をスローします)。
 - a. *name*: 変更できません (*name* はキーです)。
 - b. *displayName*: 新しい値を受け入れます。

- c. *description*: 新しい値を受け入れます。
 - d. *isRequired*: *true* から *false* への変更のみを許可します。
 - e. *isInternal*: 新しい値を受け入れます。
 - f. *isGenerated*: 変更は許可されません。
 - g. *attributeTypeEnum*: 変更は許可されません。
 - h. *accessTypeEnum*: 新しい値を受け入れます。
 - i. *selectTypeEnum*: オプションが指定された場合に以下の遷移を受け入れます。
 - 1) *NONE* から *SINGLE_SELECT* または *MULTIPLE_SELECT* へ
 - 2) *SINGLE_SELECT* から *MULTIPLE_SELECT* へ
 - j. *options*: オプションは追加できますが、削除できません。次のオプションの変更だけがサポートされます (値の一致により)。
 - 1) *displayName*: 新しい値を受け入れます (変動はありません)。
 - 2) *description*: 新しい値を受け入れます (変動はありません)。
 - 3) *isDefault*: 新しい値を受け入れます。ただし、*SelectTypeEnum* と一致しなければなりません。
 - 4) *value*: 変更は許可されません (値はキーです)。
 - k. *defaultValue* (フローチャートのみ): 新しいデフォルト値を受け入れます。
 - l. *maxLength* (テキストのみ): 大きすぎる場合は、新しい長さを受け入れません。
4. 属性メタデータ定義が内部的に整合していない場合、*InvalidAttributeException* をスローします。
5. 必要な場合、更新された属性メタデータを参照するすべてのコンポーネント・インスタンスを検索し、必要に応じて更新します。

注: 例外が生じた場合は、いずれの更新もコミットされません。

すべてのケースで、属性の更新操作は通常のセキュリティ制約および検証に従います。

createAttributeMetadata()、*deleteAttributeMetadata()* を参照してください。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

reference: 必要な属性を含むコンポーネント・インスタンスの参照 (オプション)。指定されない場合、更新はグローバルなメタデータ定義に限定されます。指定された参照が無効な場合は、*InvalidComponentException* をスローします。

allowCreate: *true* の場合、現在存在しないメタデータ定義が作成されます (*createAttributeMetadata()* メソッドを使用した場合と機能的に同等です)。

attributeMetadata: 更新する (そして *allowCreate* フラグが *true* の場合には追加する) 属性メタデータ定義の配列。属性名は更新するメタデータ定義を見つけるために使

用され、その他のデータは既存の定義を更新するために使用されます。(25 ページの『更新のロジック』を参照してください。)

戻り値

なし。

エラー

`InvalidComponentException`、`InvalidAttributeException`

`AuthorizationException`、`DataException`

deleteAttributeMetadata

```
void deleteAttributeMetadata(String userCredential,  
    String partitionName,  
    Locale requestedLocale, WSReference reference,  
    String[] names)  
    throws CampaignServicesException;
```

指定されたコンポーネント、テンプレート (カスタム属性のメタデータのみ)、またはグローバル属性のメタデータ定義から、名前の指定された 1 つ以上の属性メタデータ定義を削除します。

このタスクの一部として、メソッドは削除されたメタデータを参照するすべてのコンポーネントを検索し、それらを必要に応じて更新します。

注: ただし、例外が生じた場合は、いずれの削除もコミットされません。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

reference: 削除する属性を含むコンポーネントまたはテンプレートの参照 (オプション)。指定されない場合、削除はグローバルなメタデータ定義に限定されます。指定された参照が無効な場合は、`InvalidComponentException` をスローします。

注: オプションである属性メタデータの配列名が指定されない場合、このメソッドはコンポーネントに関連付けられたすべてのカスタム属性メタデータの削除を試行するか、または参照が指定されない場合にはすべてのグローバル定義の削除を試行します。

names: 削除する属性メタデータの名前の配列 (オプション)。名前の指定された 1 つ以上の属性メタデータが存在しない場合は `AttributeNotFoundException` をスローします。属性を削除できなかった場合は `InvalidAttributeException` をスローします。

戻り値

なし。

エラー

InvalidComponentException、AttributeNotFoundException、InvalidAttributeException
AuthorizationException、DataException

キャンペーン

API は、キャンペーンで以下の操作をサポートします (セキュリティ許可に従います)。

- 新しいキャンペーンの作成
- ディスカバリー (さまざまな基準によるキャンペーンのリスト作成)
- 属性の作成、読み取り、および更新 (属性 API による)

キャンペーンには、API によって公開される、いくつかの標準属性が関連付けられています。このリストは、カスタム属性を追加することにより、クライアントが任意に拡張できます (属性 API を参照してください)。

標準のキャンペーン属性には、以下のものがあります。

- *uacName*: キャンペーン名 (一意であるとは限りません)。
- *uacDescription*: キャンペーンについて説明する文字列 (オプション)。
- *uacCampaignCode*: キャンペーンを一意的に識別する文字列コード。通常はキャンペーンによって自動生成されますが、クライアントが指定することもできます。
- *uacCreateDate*: キャンペーンがサーバーによって作成された日時を定義するカレンダー。
- *uacUpdateDate*: キャンペーンがサーバーによって最後に更新された日時を示すカレンダー。
- *uacInitiative*: キャンペーン・イニシアチブを定義する文字列 (オプション)。
- *uacObjectives*: キャンペーンの目的を示す文字列 (オプション)。
- *uacStartDate*: キャンペーンがサーバーによって開始された日時、または開始するようにスケジュール設定されている日時を示すカレンダー (オプション)。
- *uacEndDate*: *uacStartDate* と似ていますが、キャンペーンが完了した日時、または完了するようにスケジュール設定されている日時を定義します。 *uacStartDate* よりも後の日時でなければなりません。
- *uacLastRunDate*: キャンペーンにバインドされたフローチャートが最後に実行された日時を示すカレンダー (オプション)。実行されていない場合は NULL となります。
- *uacExternalLinkOwner*: 外部リンクの所有者の名前を定義する文字列 (オプション)。 *uacExternalLinkReference* 属性を参照してください。 IBM Unica での使用に限定されます。以下のいずれかでなければなりません。
 - 「Plan」 (現在は IBM Unica Marketing Operations と呼ばれます)
 - 「Collaborate」 (現在は IBM Unica Distributed Marketing と呼ばれます)
- *uacExternalLinkId*: このキャンペーンにリンクされたオブジェクトに対して別の IBM Unica アプリケーションによって割り当てられている、数値によるデータベース ID (オプション)。 IBM Unica での使用に限定されます。
uacExternalLinkOwner 属性も参照してください。

generateCampaignCode

```
String generateCampaignCode(String userCredential,  
    String partitionName,  
    Locale requestedLocale);
```

新しいキャンペーン・コードを生成します。

このコードは必ず一意的なものとなり、このメソッドや `createCampaign()` メソッドに対する以前または将来の呼び出しから返される値や、IBM Unica Campaign GUI を介して作成されたキャンペーンのために生成される値とは異なります。

注: クライアントのキャンペーン・コードが指定されない場合には `createCampaign()` API が生成するので、このメソッドの使用はオプションです。

`createCampaign()` を参照してください。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。キャンペーンのインストール済み環境にパーティションが 1 つのみある場合、この引数は NULL にすることもできます。

戻り値

生成されたキャンペーン・コード。

エラー

AuthorizationException、DataException

createCampaign

```
CampaignInfo createCampaign(String userCredential,  
    String partitionName,  
    Locale requestedLocale,  
    String securityPolicyName,  
    String name, Attribute[] attributes)  
throws InvalidFolderException, AttributeNotFoundException,  
    InvalidAttributeException;
```

指定の属性を適用して、クライアント、パーティション、および `securityPolicyName` のための新しいキャンペーンを作成します。この API によって作成されるすべてのキャンペーンは、ルート・フォルダーの下に置かれます。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

securityPolicyName: キャンペーンを作成するために使用するキャンペーン・セキュリティ・ポリシーの名前 (オプション)。このキャンペーンでの後続の操作はすべて、このポリシーを使用します。定義されていない場合、グローバル・ポリシーが使用されます。

name: 新しいキャンペーン・インスタンスを割り当てる名前 (その「*uacName*」属性)。

attributes: 初期化属性の配列 (オプション)。指定した属性により、キャンペーンのデフォルト値が上書きされます。その他の値はそのまま存続します。例えば、*uacCampaignCode* 属性を指定した場合、自動生成された属性の代わりにそれが使用されます。キャンペーンに必要な属性やそのタイプなどは、クライアントが決定する必要があります。

名前の指定された 1 つ以上の属性が存在しない場合は *AttributeNotFoundException* をスローし、属性値が無効な場合 (データ型が正しくないなど) は *InvalidAttributeException* をスローします。

戻り値

作成されたキャンペーンのための *CampaignInfo* の単一インスタンス。

エラー

InvalidAttributeException、*AttributeNotFoundException*

AuthorizationException、*DataException*

listCampaignsByPage

```
List<CampaignInfo>  
listCampaignsByPage(String userCredential, String partitionName,  
    Locale requestedLocale, Attribute[] attributes,  
    long pageOffset, int pageSize)  
throws AttributeNotFoundException, InvalidAttributeException,  
    RangeException;
```

オプションの属性値と一致するキャンペーンの「ページ」を、指定のページ・オフセットから開始して列挙します。フォルダーは無視されます。

一度取得されると、返される各 *CampaignInfo* を現状のまま使用して、サマリー・リストを表示することなどができます。または属性メソッドを使用して、キャンペーンの属性を取得することや更新することができます。

API によって維持される状態はないので、任意の順序でそれを呼び出すことが可能です。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

attributes: 照合するための属性の配列 (オプション)。属性の名前、データ型、および値が一致を判別するために使用されます。属性が配列をサポートする場合、指定されたすべての値が一致する必要があります。暗黙指定されている一致の演算子は AND なので、指定されたすべての属性値と一致するキャンペーンだけが返されます。

属性名が存在しない場合は *AttributeNotFoundException* をスローし、指定された 1 つ以上の属性が無効な場合は *InvalidAttributeException* をスローします。

pageOffset: 列挙を開始する位置を示す、可能なすべてのキャンペーンの開始オフセット (ゼロ値)。例えば、列挙が 1000 のキャンペーンと一致して、この値が 10 に設定されている場合、ページは 11 番目のコンポーネントから開始します。指定されたオフセットが範囲外の場合、*RangeException* がスローされます。

pageSize: ページに対して返される、一致したキャンペーンの最大数 (500 を超えることはできません)。

戻り値

ページ内で一致するキャンペーンごとに 1 つの、ゼロ以上の *CampaignInfo* データ・ラッパー・インスタンスの型付きリスト。

エラー

AttributeNotFoundException、*InvalidAttributeException*、*RangeException*

InvalidExecutionContextException、*AuthorizationException*

ターゲット・セル・メソッド

ターゲット・セルは、IBM Unica Campaign によって管理されるキャンペーン結果のいくつかの既知のサブセットを、ターゲット・セル・スプレッドシート (TCS) として抽象化したものです。ターゲット・セルは、キャンペーンに対してグローバルであったり、特定のキャンペーン・フローチャートに関連付けられていたりします。

API は、ターゲット・セルで以下の操作をサポートします。

- 1 つ以上の新しいグローバル・ターゲット・セルの作成
- 1 つ以上の既存のターゲット・セルのバルク更新
- ディスカバリー (ターゲット・セルのリスト作成)
- 属性の作成、読み取り、および更新 (属性 API による)
- 既存のターゲット・セルの削除
- 1 つ以上のセルに関連付けられた実行結果の取得

ターゲット・セルには、API によって公開される、いくつかの標準属性が関連付けられています。このリストは、カスタム属性のメタデータ定義を追加することにより、クライアントが任意に拡張できます (属性 API を参照してください)。各属性メタデータは、TCS 内の 1 つの列と見なすことができます。スプレッドシートのレイアウトは、クライアントが決定する必要があります。

標準のターゲット・セル属性には、以下のものがあります。

- *uacName*: セル名。
- *uacDescription*: フローチャートについて説明する文字列 (オプション)。
- *uacCellCode*: セルを一意的に識別するコード文字列。通常は Campaign が自動生成しますが、クライアントが指定することもできます。
- *uacCreateDate*: セルがサーバーによって作成された日時を指定するカレンダー・インスタンス。
- *uacUpdateDate*: セルがサーバーによって最後に更新された日時を定義するカレンダー・インスタンス。
- *uacIsControl*: コントロール・セル (true) であるか、そうでない (false) かを示すブール値。他のセルはこのセルをコントロール・セルとして参照することがあります (*uacControlCell* を参照してください)。
- *uacControlCell*: コントロール・セルの参照 (オプション。ただし、コントロール・セルの場合には許可されません)。 *uacIsControl* 属性を参照してください。
- *uacIsApproved*: セルが承認されている (true) か、そうでない (false) かを示すブール値。
- *uacIsReadOnly*: セルが読み取り専用 (true) か、そうでない (false) かを示すブール値。
- *uacDisplayOrder*: このセル (行) の順序を、ターゲット・セル・スプレッドシート内の他のセルとの相対関係で示す整数。
- *uacIsTopDown*: セルがトップダウンかどうかを示すブール値。
- *uacAssignedOffers*: このセルに割り当てられたオファーまたはオファー・リストの、1 つ以上の参照の配列 (オプション)(コントロール・セルの場合には許可されません)。
- *uacFlowchartName*: このセルがリンクされているフローチャートの名前 (オプション)(IBM Unica Campaign GUI によって読み取り専用設定される必要があります。コントロール・セルの場合には許可されません)。
- *uacFlowchartId*: このセルがリンクされているフローチャートのデータベース ID (オプション)(上記と同様に読み取り専用)。

createTargetCell

```
TargetCellInfo
createTargetCell(String userCredential, String partitionName,
                 Locale requestedLocale,
                 Reference campaignReference,
                 Attribute[] attributes)
throws InvalidComponentException, CompositeException;
```

指定されたセルごとの属性とユーザー情報を適用して、新しいキャンペーン固有のターゲット・セル行を作成します。

指定する属性は標準でもカスタムでも可能ですが、カスタムにする場合は、対応するグローバル属性メタデータ定義が既に存在している必要があります。

ターゲット・セルが作成された後に、属性 API を使用して属性値を変更できます。

`listTargetCells()`、`bulkCreateTargetCells()` を参照してください。

`createAttributeMetadata()`、`listAttributeMetata()`、`getAttributesByName()` を参照してください。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

campaignReference: 更新するターゲット・セル・スプレッドシートを含むキャンペーンの参照。キャンペーンが存在しない場合、または参照が無効な場合には、*InvalidComponentException* を累積します。

attributes: 新しいセルのための TCS 属性の配列 (オプション)。指定される各属性要素は、対応するセル属性のデフォルト値を上書きします。その他の値は変更されずにそのままとなります。セルに必要な属性やそのタイプなどは、クライアントが決定します。指定された属性に問題がある場合は、*InvalidAttributeException* を累積します。

例外が累積された場合、このメソッドは *CompositeException* をスローして、すべての作成が取り消されます。例外の原因リストには、エラーを生じた各属性の例外が含まれ、参照の代わりに数値インデックスが含まれ、属性の名前、および通常は問題となっている値などが含まれます。原因リストは、入力 *attributeList* と同様に順序付けされます。

戻り値

作成される TCS セルの *TargetCellInfo* データ・ラッパー。

エラー

InvalidComponentException、*CompositeException*

AuthorizationException、*DataException*

bulkCreateTargetCells

```
List<TargetCellInfo>  
    bulkCreateTargetCells(String userCredential,  
                          String partitionName,  
                          Locale requestedLocale,  
                          Reference campaignReference,  
                          List<Attribute[]> attributesList)  
    throws InvalidComponentException, CompositeException;
```

指定されたセルごとの属性とユーザー情報を適用して、一度に多数の新しいキャンペーン固有のターゲット・セル行を作成します。

指定する属性は標準でもカスタムでも可能ですが、カスタムにする場合は、対応するグローバル属性メタデータ定義が既に存在している必要があります。

ターゲット・セルが作成された後に、属性 API を使用して属性値を変更できます。

`listTargetCells()` を参照してください。

`createAttributeMetadata()`、`listAttributeMetata()`、`getAttributesByName()` を参照してください。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

campaignReference: 更新するターゲット・セル・スプレッドシートを含むキャンペーンの参照。キャンペーンが存在しない場合、または参照が無効な場合には、*InvalidComponentException* を累積します。

attributeList: 作成するターゲット・セル行ごとに 1 つの、セルごとの属性配列のリスト (オプション)。特定のリスト要素に対して属性が指定される場合、それは対応するセル属性のデフォルト値を上書きします。その他の値は変更されずにそのままとなります。セルに必要な属性やそのタイプなどは、クライアントが決定します。指定された属性に問題がある場合は、*InvalidAttributeException* を累積します。

例外が累積された場合、このメソッドは *CompositeException* をスローして、すべての作成が取り消されます。例外の原因リストには、エラーを生じた各属性の例外が含まれ、参照の代わりに数値インデックスが含まれ、属性の名前などが含まれません。原因リストは、入力 *attributeList* と同様に順序付けされます。

戻り値

入力 *attributesList* パラメーターの要素の順序に従って順序付けされた、作成されたインスタンスごとに 1 つの、*TargetCellInfo* データ・ラッパーのリスト。

エラー

InvalidComponentException、*CompositeException*

AuthorizationException、*DataException*

listTargetCells

```
List<TargetCellInfo>  
listTargetCells(String userCredential,  
Reference campaignReference, Locale requestedLocale,  
Attribute[] attributes)  
throws InvalidComponentException, InvalidAttributeException;
```

指定されたキャンペーンについて、またはキャンペーンが指定されない場合はグローバルに、指定された属性に一致して現在存在するすべてのターゲット・セルに関する情報をリストします。

`getAttributeMetadata()`、`getAttributesByName()` を参照してください。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

campaignReference: 親キャンペーンの参照。キャンペーンが存在しない場合、または参照が無効な場合には、*InvalidComponentException* をスローします。

attributes: 一致する属性の配列 (オプション)。暗黙指定されている一致の演算子は AND なので、指定されたすべての属性値と一致するセルだけが返されます。

指定された 1 つ以上の属性が無効な場合、*InvalidAttributeException* をスローします。

戻り値

一致したセルに関するゼロ以上の *TargetCellInfo* インスタンスのリストを返します。

エラー

InvalidComponentException、*InvalidAttributeException*

AuthorizationException、*DataException*

bulkUpdateTargetCells

```
void bulkUpdateTargetCells(String userCredential,  
    String partitionName,  
    Locale requestedLocale,  
    Map<Reference, Attribute[]> attributesMap)  
    throws CompositeException;
```

1 つ以上のターゲット・セルの属性を更新します。

更新のロジックは、以下のとおりです。

提供された *attributesMap* の各要素で、エントリー・キーは更新するターゲット・セルの参照、エントリー値はそのセルの更新属性の配列です。ターゲット・セルが存在しない場合、*InvalidComponentException* を累積します。

ターゲット・セルが見つかった場合、指定された属性ごとに、以下を行います。

1. 属性名が既存の属性と一致する場合、その値項目を提供された値項目で上書きすることを試行します。
2. 値タイプ、または属性のメタデータ定義の他の側面が一致しない場合、または提供された 1 つ以上の値が無効や範囲外などの場合は、*InvalidAttributeException* を累積します。
3. また、名前属性が存在しない場合、*AttributeNotFoundException* を累積します。

例外が累積された場合、このメソッドは *CompositeException* をスローして、すべての更新が取り消されます。例外の原因リストには、上にリストされている例外が含まれます。エラーを生じた属性ごとに、参照および属性名の両方が記録されます。

すべてのケースで、属性の更新操作は通常のセキュリティー制約および検証に従います。特定のコンポーネント・インスタンスに必要な属性や正しいタイプなどは、クライアントが決定する必要があります。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

attributesMap: 更新するターゲット・セルのマップ。エントリー・キーは更新するセルの参照、エントリー値は更新属性の配列です。属性名は更新する属性を見つけるために使用され、新しい属性値は、既存の属性の値を適切なタイプの単一オブジェクトまたは配列として更新するために使用されます (該当する場合)。上の例外を参照してください。

戻り値

なし。

エラー

ComponentException

AuthorizationException、DataException

getRunResultsByCell

```
List<RunResults>  
getRunResultsByCell(String userCredential, String partitionName,  
    Locale requestedLocale,  
    Reference[] cellReferences)  
throws InvalidComponentException;
```

開始しなかったフローチャートや進行中のフローチャートなど、1 つ以上のターゲット・セルの実行結果を取得します。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

cellReferences: 実行結果が必要とされるターゲット・セルの参照の配列。1 つ以上のセル参照が無効であるか、存在しないセルを参照している場合は、*InvalidComponentException* をスローします。

戻り値

名前の指定されたセルの実行結果を、入力参照配列に従って順序付けされた型付きリストの形式で返します。

各実行ステータスは、基礎となるフローチャートのプロセス・ボックスがまだ実行中の場合は **RUNNING**、何らかの理由で実行が失敗した場合は **FAILED**、プロセス・ボックスの実行が開始されていない場合は **NOT_STARTED** となります。ステータスの詳細も示されます。

エラー

InvalidComponentException

AuthorizationException、DataException

bulkDeleteTargetCells

```
void bulkDeleteTargetCells(String userCredential,  
    String partitionName,  
    Locale requestedLocale,  
    Reference[] cellReferences)  
    throws CompositeException;
```

1 つ以上の既存のターゲット・セル、およびそのすべての従属コンポーネント（つまりフローチャート結合、属性、その他）を削除します。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール（オプション）。

partitionName: 使用するキャンペーン・パーティションの名前（オプション）。

cellReferences: 削除するセルの配列または 1 つ以上の参照。指定された参照の 1 つに問題がある場合、またはセルが存在しない場合には、*InvalidComponentException* が累積されます。

例外が累積された場合、このメソッドは *CompositeException* をスローして、すべての削除が取り消されます。例外の原因リストには、上にリストされている例外が含まれます。エラーを生じたセルごとに、参照が記録されます。

戻り値

なし。

エラー

CompositeException

AuthorizationException、DataException

分析

API は、IBM Unica Campaign からの簡単なメトリックの取得をサポートします。

getCampaignMetrics

```
MetricsInfo getCampaignMetrics(String userCredential,  
    String partitionName,  
    Locale requestedLocale,  
    Reference campaignReference)  
    throws InvalidComponentException;
```

指定のキャンペーンのメトリックを取得します。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

campaignReference: 親キャンペーンの参照。キャンペーン参照に問題がある場合、またはキャンペーンが存在しない場合には、*InvalidComponentException* をスローします。

戻り値

キャンペーンの *MetricsInfo* インスタンスを返します。

エラー

InvalidComponentException

AuthorizationException、*DataException*

オファー、オファー・リスト、およびオファー・テンプレートのメソッド

API は、オファーで以下の操作をサポートします。

- ディスカバリー (フォルダーごとのリスト (オファー、オファー・リスト、およびサブフォルダー)、属性 (オファーおよびオファー・テンプレート)、または検索値 (オファー))
- 検証
- 情報検索 (特定のオファーまたはオファー・テンプレートの属性の取得)
- オファーの作成、編集、回収、および削除

オファーには、いくつかの標準属性が関連付けられています。このリストは、カスタム属性のメタデータ定義を追加することにより、クライアントが任意に拡張できます (属性 API を参照してください)。

標準のオファー属性には、以下のものがあります。

- *uacName*: オファー名。
- *uacDescription*: オファーについて説明する文字列 (オプション)。
- *uacOfferCode*: オファーを一意的に識別するコード文字列。通常は IBM Unica Campaign が自動生成しますが、クライアントが指定することもできます。
- *uacCreateDate*: オファーがサーバーによって作成された日時を示すカレンダー・インスタンス。
- *uacUpdateDate*: オファーがサーバーによって最後に更新された日時を定義するカレンダー・インスタンス。

オファー・テンプレートには、標準属性およびカスタム属性もあります。標準のオファー・テンプレート属性には、以下のものがあります。

- *uacName*: オファー・テンプレート名。
- *uacDescription*: オファー・テンプレートについて説明する文字列 (オプション)。

- *uacCreateDate*: オファー・テンプレートがサーバーによって作成された日時を示すカレンダー・インスタンス。
- *uacUpdateDate*: オファー・テンプレートがサーバーによって最後に更新された日時を定義するカレンダー・インスタンス。

listOffersAndFolders

```
List<WSComponentOrFolderInfo>
  listOffersAndFolders(String userCredential, String partitionName,
    Locale requestedLocale,
    WSReference parentReference)
  throws CampaignServicesException;
```

オプションの親フォルダーの下に、すべてのオファー、オファー・リスト、およびフォルダーをリストします。

一度取得されると、返される各 *WSComponentOrFolderInfo* インスタンスを現状のまま使用して、フォルダー階層の次のレベルを表示することなどができます。また属性 API を使用して、含まれているオファーを取得または更新することができます。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

parentReference: リストする親フォルダーの参照 (オプション)。この親フォルダーの直接の子オファー、オファー・リスト、およびフォルダーのみが列挙されるので、フォルダー階層全体でのナビゲートには、この API に対する後続の呼び出しが必要となります (ただし通常はとても浅い階層です)。親が指定されない場合、ルートの下にあるすべてのコンポーネントおよびフォルダーが返されます。

指定された親フォルダー参照に問題がある場合、*InvalidFolderException* をスローします。

一致するコンポーネントまたはフォルダーごとに 1 つの、ゼロ以上の *WSComponentOrFolderInfo* データ・ラッパー・インスタンスの型付きリスト。

エラー

InvalidFolderException

InvalidExecutionContextException, *AuthorizationException*

searchOffersBasic

```
List<WSOfferInfo>
  searchOffersBasic(String userCredential, Locale requestedLocale,
    String partitionName, long folderID,
    String searchCriteria, boolean includeRetired,
    int pageOffset, int pageSize)
  throws CampaignServicesException;
```

「名前」、「記述」、「作成者」、または「オファー・コード」項目で指定された検索基準を含むオファーの「ページ」を、指定のページ・オフセットから開始して列挙します。検索はオプションのフォルダー入力に基づいて行われます。(folderID に 0 が指定された場合、ルート・オファー・フォルダーがデフォルトで使用されます)。一致が返される基準は、検索文字列が「contains」一致であることです。

一度取得されると、返される各 *WSOfferInfo* を現状のまま使用して、サマリー・リストを表示することなどができます。または属性メソッドを使用して、オファーの属性を取得することや更新することができます。

API によって維持される状態はないので、任意の順序でそれを呼び出すことが可能です。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

folderID: 検索対象のオファー・フォルダーの ID。folderID に 0 が指定された場合、ルート・フォルダーが検索されます。

searchCriteria: 検索句。

includeRetired: 検索結果に回収されたオファーを含むかどうかを指定するブール値。有効な値は TRUE および FALSE で、TRUE は回収されたオファーが含まれることを示し、FALSE は回収されたオファーが含まれないことを示します。

pageOffset: 列挙を開始する位置を示す、可能なすべてのコンポーネントの開始オフセット (ゼロ値)。例えば、列挙が 1000 のオファーと一致して、この値が 10 に設定されている場合、ページは 11 番目のコンポーネントから開始します。指定されたオフセットが範囲外の場合、*RangeException* がスローされます。

pageSize: ページに対して返される、一致したコンポーネントの最大数 (500 を超えることはできません)。

戻り値

ページ内で返されるオファーごとに 1 つの、ゼロ以上の *Offer* データ・ラッパー・インスタンスの型付きリストを返します。

エラー

RangeException

listOffersByPage

```
List<OfferInfo>
listOffersByPage(String userCredential, String partitionName,
                 Locale requestedLocale, Attribute[] attributes,
                 long pageOffset, int pageSize)
throws AttributeNotFoundException, InvalidAttributeException,
RangeException;
```

オプションの属性値と一致するオファーの「ページ」を、指定のページ・オフセットから開始して列挙します。フォルダーは無視されます。一致が返されるための基準は、文字列については「類似」一致 (文字列にクエリー値が含まれる場合は条件が満たされているとみなされる)、日付および数値については完全一致となります。

一度取得されると、返される各 *OfferInfo* を現状のまま使用して、サマリー・リストを表示することなどができます。または属性メソッドを使用して、オファーの属性を取得することや更新することができます。

API によって維持される状態はないので、任意の順序でそれを呼び出すことが可能です。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

attributes: 照合するための属性の配列 (オプション)。属性の名前、データ型、および値が一致を判別するために使用されます。属性が配列をサポートする場合、指定されたすべての値が一致する必要があります。暗黙指定されている一致の演算子は OR なので、指定されたいずれかの属性値と一致するコンポーネントが返されます。

属性名が存在しない場合は *AttributeNotFoundException* をスローし、指定された 1 つ以上の属性が無効な場合は *InvalidAttributeException* をスローします。

pageOffset: 列挙を開始する位置を示す、可能なすべてのコンポーネントの開始オフセット (ゼロ値)。例えば、列挙が 1000 のオファーと一致して、この値が 10 に設定されている場合、ページは 11 番目のコンポーネントから開始します。指定されたオフセットが範囲外の場合、*RangeException* がスローされます。

pageSize: ページに対して返される、一致したコンポーネントの最大数 (500 を超えることはできません)。

戻り値

ページ内で一致するコンポーネントごとに 1 つの、ゼロ以上の *OfferInfo* データ・ラッパー・インスタンスの型付きリスト。

エラー

AttributeNotFoundException、*InvalidAttributeException*、*RangeException*

InvalidExecutionContextException、*AuthorizationException*

validateOffers

```
List<OfferValidationInfo>  
validateOffers(String userCredential, String partitionName,  
Locale requestedLocale,  
OfferCodeOrName[] codeOrNames);
```

提供されたオファー・コードまたはオファー・リスト名を検証して、それぞれの検証情報を返します。「検証」は、一致するオファーまたはオファー・リストがデータベース内に厳密に 1 つだけ存在するかどうかの検査からなります。

`OfferValidationInfo` オブジェクトには、指定のコードまたは名前と一致するオファーまたはオファー・リストがない場合に、オファー情報の代わりにエラー・メッセージが含まれます。指定のコードまたは名前が複数のオファーまたはオファー・リストと一致した場合にも、一致の代わりにエラーが返されます。リストは、指定されたときと同じ順序で返されます。オファー・コードおよびオファー・リストの名前は、オファーとの完全一致を基準として検証されます。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

codeOrNames: 検証するすべてのオファー・コードまたはオファー・リストの名前の配列。

注: このメソッドによってスローされる例外はありません。代わりに、指定されたすべてのコードまたは名前に対して検証情報が返されます。

戻り値

ゼロ以上の `OfferValidationInfo` データ・ラッパー・インスタンスの型付きリスト。

エラー

なし。

createOffer

```
OfferInfo createOffer(String userCredential, String partitionName,
    Locale requestedLocale,
    String securityPolicyName,
    String name, String templateName,
    Attribute[] attributes)
    throws InvalidFolderException, AttributeNotFoundException,
    InvalidAttributeException;

public WSOfferInfo createOffer(String authorizationLoginName, String
    partitionName, Locale requestedLocale, String
    securityPolicyName, String name, long folderID,
    String templateName, WSAttribute[] wsAttributes)
    throws CampaignServicesException;
```

指定の属性を適用して、クライアントのための新しいオファーを作成します。

パラメーター

authorizationLoginName: オファーを作成しているユーザーのユーザー名。このメソッドを使用するためには、ユーザーに「オファーの追加」権限が付与されている必要があります。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

requestedLocale: この要求で使用するロケール (オプション)。

securityPolicyName: オファーを作成するために使用するキャンペーン・セキュリティ・ポリシーの名前 (オプション)。このオファーでの後続の操作はすべて、このポリシーを使用します。定義されていない場合、グローバル・ポリシーが使用されます。

name: 新しいオファー・インスタンスを割り当てる名前 (その *uacName* 属性)。

folderID: オファーが作成されるオファー・フォルダーの ID。この ID は正確かどうかを検証され、ID が無効な場合には例外がスローされます。

templateName: 新しいオファーに使用する必要のある、既存のオファー・テンプレートの必須 (固有) 名。

wsAttributes: 初期化属性の配列。指定した属性により、オファーのデフォルト値が上書きされます。その他の値はそのまま存続します。例えば、*uacOfferCode* 属性を指定した場合、自動生成された属性の代わりにそれが使用されます。オファーに必要な属性やそのタイプなどは、クライアントが決定します。

以下のいずれかの条件が発生した場合、*CampaignServicesException* をスローします。

- *folderID* パラメーターが無効 (存在しないかオファーのタイプではない)。
- ユーザーにこの操作の実行が許可されていない。
- *wsAttributes* に無効な属性が指定されている。
- その他のランタイム例外が生じた。

戻り値

作成されたオファーのための *OfferInfo* の単一インスタンス。

エラー

CampaignServicesException

retireOffers

```
void retireOffers(String userCredential, String partitionName,  
                 Locale requestedLocale, WSReference[] references)  
    throws CampaignServicesException;
```

1 つ以上の既存のオファーを回収します。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

references: 回収するオファーの参照の配列。特定の参照に問題があるかまたはオファーが存在しない場合には、*InvalidComponentException* がスローされます。

戻り値

なし。

エラー

`InvalidComponentException`

`AuthorizationException`、`DataException`

deleteOffers

```
void deleteOffers(String userCredential, String partitionName,  
                 Locale requestedLocale, WSReference[] references)  
    throws CampaignServicesException;
```

1 つ以上の既存のオファーを削除します。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

reference: 削除するオファーの参照の配列。指定された参照に問題があるかまたはオファーが存在しない場合には、`InvalidComponentException` がスローされます。

戻り値

なし。

エラー

`InvalidComponentException`

`AuthorizationException`、`DataException`

listOfferTemplates

```
List<WSOfferTemplateInfo>  
listOfferTemplates(String userCredential, String partitionName,  
                  Locale requestedLocale)  
    throws CampaignServicesException;
```

ユーザーが表示する権限を持つすべてのオファー・テンプレートをリストします。

一度取得されると、返される各 `WSOfferTemplateInfo` インスタンスを現状のまま使用することができます。または、1 つ以上の属性 API を使用して、リストされたいずれかのテンプレートを取得または更新することができます。

パラメーター

userCredential: クライアント・ユーザー資格情報。

requestedLocale: この要求で使用するロケール (オプション)。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

戻り値

返されるテンプレートごとに 1 つの、ゼロ以上の *WSOfferTemplateInfo* データ・ラッパー・インスタンスの型付きリスト。

エラー

InvalidExecutionContextException、*AuthorizationException*

DataException

bulkCreateOffers

```
WSOfferInfoStatus[] bulkCreateOffers(String authorizationLoginName,  
    String partitionName, Locale requestedLocale,  
    String securityPolicyName, String templateName, long folderID,  
    WSBulkOfferInfo[] offers)  
    throws CampaignServicesException;
```

offers パラメーターに指定された属性を各オファーに使用して、オファーをバルクで作成します。すべてのオファーは、指定の *folderID* に指定の *templateName* を使用して作成されます。

パラメーター

authorizationLoginName: クライアント・ユーザー資格情報。

partitionName: 使用するキャンペーン・パーティションの名前 (オプション)。

requestedLocale: この要求で使用するロケール (オプション)。

securityPolicyName: オファーを作成するために使用するキャンペーン・セキュリティ・ポリシーの名前 (オプション)。定義されていない場合、グローバル・ポリシーが使用されます。

templateName: システム内の既存のオファー・テンプレートの名前。すべてのオファーは、このテンプレートを使用して作成されます。

folderID: オファーが作成されるオファー・フォルダーの ID。この ID は検証されて、ID が無効な場合には例外がスローされます。

offers: オファー名および属性を定義する *WSBulkOfferInfo* オブジェクトの配列。詳しくは、*WSBulkOfferInfo* データ型を参照してください。

戻り値

各オファーの *WSOfferInfoStatus* インスタンスの配列。ステータスおよびオファー情報が含まれます。ステータスは、オファーの作成が成功したかどうかを示します。

エラー

CampaignServicesException

第 5 章 API の使用

このセクションでは、Campaign Web サービス API の使用方法について説明しています。また、Campaign API サービスを使用して Campaign 内にオファーを作成する例も示しています。

Campaign サービス API を使用するには、次の 2 つの方法があります。

- 『Client API の .jar の使用』
- 49 ページの 『WSDL の使用』

Client API の .jar の使用

IBM Unica Campaign には、SOAP Web サービスを使用して Campaign Web アプリケーションと対話する Client API が備わっています。このラッパーは、クライアント・アプリケーションが Campaign API を呼び出すために使用できる .jar ファイル内にバンドルされています。

この .jar ファイルは、次の場所にあります。

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/  
CampaignServicesClient30.jar
```

次の例は、Campaign のルート・オファー・フォルダー・レベルで新しいオファーを作成する方法を示しています。同じ例は、次の場所にもあります。

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/samples/OfferAPI.java
```

注: この例では、パラメーター用にいくつかのダミーの値を使用しています。実際の値は異なることがあります。

Campaign Web サービスの URL が `http://host:port/Campaign/services/CampaignServices30Service` であることにも注意してください。ここで、host および port は、Campaign Web アプリケーションが配置されるマシンのホスト名およびポート番号を表します。

提供されているサンプルを使用する場合には、それをクライアント環境に合うように変更してください。

コード OfferAPI.java

```
import java.net.URL;  
import java.util.Locale;  
import com.unica.publicapi.campaign.campaignservices.CampaignServicesException;  
import com.unica.publicapi.campaign.campaignservices.attribute.metadata.  
    IAttribute Metadata;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.  
    CampaignServices30SoapClient;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSAttribute;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSOfferInfo;  
import com.unica.publicapi.campaign.campaignservices.utils.WSAttributeUtils;
```

```

/**
 * This is the sample java client class that shows the usage of Campaign's SOAP
 services API.
 * This sample uses CampaignServices30SoapClient facade to interact with Campaign
 web service.
 * Here the creation of Offer is shown. Please refer to the API guide for
 more details.
 *
 * @author AGijare
 *
 */
public class OfferAPI {

    /**
     * @param args
     */
    protected static CampaignServices30SoapClient CLIENT = null;

    private static void setup(){
        try {
            String protocol = "http"; //http or https
            String host = "localhost"; //Host name of deployed Campaign.
            Use proper host name here.
            int port = 7001; //port number of deployed Campaign
            long timeOut = 2*60*1000; // 2 minutes
            String servicesURI = "/Campaign/services/CampaignServices30Service";
            CLIENT = new CampaignServices30SoapClient(
                new URL(protocol, host, port, servicesURI),
                timeOut);
        } catch (Exception exception) {
            exception.printStackTrace();
            System.exit(-1);
        }
    }

    public static void main(String[] args) {
        //Please change the values of following variables to match your
 environment.
        String userName = "user_name"; //login user name
        String partitionName = "partition1"; //Use proper partition name of
 Campaign
        Locale loc = Locale.US;
        String securityPolicy = "Global"; //Use your security policy of
 Campaign

        String offerName = "Offer1";
        String offerTemplate = "Offer Template"; // Template from which
 Offer will be created.
        long folderID = 1002; //Actual ID of the folder where this offer
 will be created. For folderID <=0, offer will be created at root level.
        //Attributes of Offer
        WSAttribute[] wsAttributes = {
            WSAttributeUtils.getWSTextAttribute(IAttributeMeta
 data.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME, null, new String[]{"description "
 + System.currentTimeMillis()})
        };

        setup();

        try {
            WSOfferInfo wsOfferInfo = CLIENT.createOffer(userName,
 partitionName, loc, securityPolicy,
 offerName, folderID, offerTemplate, wsAttributes);
            System.out.println("Created offer: " + wsOfferInfo.getName());
        } catch (CampaignServicesException e) {

```

```
        e.printStackTrace();
    }
}
```

注: 上に示されている Java サンプルをコンパイルして実行するには、従属するすべての .jar ファイルを Java クラスパスに含める必要があります。

CampaignServicesClient30.jar ファイルは、<CAMPAIGN_HOME>/Campaign.war にある Campaign.war ファイル内の Apache AXIS2 SOAP エンジン .jar ファイルおよび他の共通 Apache .jar ファイルに従属します。 .jar ファイルを Campaign.war から抽出して、Java クラスパスに含めます。

WSDL の使用

Campaign サービスは、次の Campaign Web サービスの WSDL ファイルを使用して呼び出すこともできます。

CampaignServices30.wsdl

これは次の場所にあります。

<http://host:port/Campaign/services/CampaignServices30Service?wsdl>

または、次の場所にある Campaign 配布に含まれています。

<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/

クライアント Java アプリケーションは、サード・パーティー製の WSDL から Java へのコンバーター・ツールを使用して WSDL から生成される、クラスおよびスタブを使用する必要があります。IBM Unica は、Apache AXIS の使用をお勧めします。

Apache AXIS2 を使用して WSDL から生成されたクラスおよびスタブから作成される javadocs は、次の場所にあります。

<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/javadocs/index.html

注: 従属するすべての .jar ファイルは、Java クラスパスに含まれている必要があります。CampaignServicesClient30.jar ファイルは、<CAMPAIGN_HOME>/Campaign.war にある Campaign.war ファイル内の Apache AXIS2 SOAP エンジン .jar ファイルおよび他の共通 Apache .jar ファイルに従属します。 .jar ファイルを Campaign.war から抽出して、Java クラスパスに含めます。

パフォーマンスについての考慮事項

現在の CampaignServices API 実装のパフォーマンス・プロファイルは、GUI を利用したときのアプリケーションのパフォーマンス・プロファイルと似ています。いくつかの API は、明示的にパフォーマンスのために設計されています。特に listCampaignsByPage() API は、相対的に効率の良いページ編集を可能にします。

SOAP インターフェースでは、場合によっては大変に冗長なものとなる XML 形式にすべてのデータが変換されるので、その性質上、待ち時間およびオーバーヘッド

が生じます。例えば、単純なループバック SOAP 呼び出しは、標準的なネットワークで 100 ミリ秒かかります (Java 1.4.x ではさらに長くかかりました)。API は、SOAP のパフォーマンスが適切になるよう、標準的なポータルや他のクライアント・アプリケーションのビジネス・ユース・ケース (see `listOffersByPage()`) などを参照

ただしクライアントは、通常の Web ユーザー要求の CampaignServices サービスにかかる負荷が過大にならないように注意する必要があります。一般的に、API ユーザーが必要とする処理量は、典型的な IBM Unica Campaign Web ユーザーが必要とする処理量を超えないと预期されています。

パッケージ化

この仕様は、IBM Unica Campaign と共にインストールされる CampaignServices Software Developer's Toolkit (devkits) の一部として配布されます。

インストーラーによって配置される devkits ディレクトリーには、例、ビルドとテキストのスクリプト、パブリック・クラスとインターフェースの javadoc、リリース・ノートなどが含まれます。

IBM Unica 技術サポートへの連絡

ドキュメンテーションを参照しても解決できない問題があるなら、指定されているサポート窓口を通じて IBM Unica 技術サポートに電話することができます。このセッションの情報を使用するなら、首尾よく効率的に問題を解決することができます。

サポート窓口が指定されていない場合は、IBM Unica 管理者にお問い合わせください。

収集する情報

IBM Unica 技術サポートに連絡する前に、以下の情報を収集しておいてください。

- 問題の性質の要旨。
- 問題発生時に表示されるエラー・メッセージの詳細な記録。
- 問題を再現するための詳しい手順。
- 関連するログ・ファイル、セッション・ファイル、構成ファイル、およびデータ・ファイル。
- 「システム情報」の説明に従って入手した製品およびシステム環境についての情報。

システム情報

IBM Unica 技術サポートに電話すると、実際の環境に関する情報について尋ねられることがあります。

問題が発生してもログインは可能である場合、情報の大部分は「バージョン情報」ページで入手できます。そのページには、インストールされている IBM Unica のアプリケーションに関する情報が表示されます。

「バージョン情報」ページは、「ヘルプ」>「バージョン情報」を選択することにより表示できます。「バージョン情報」ページを表示できない場合、どの IBM Unica アプリケーションについても、そのインストール・ディレクトリの下にある `version.txt` ファイルを表示することにより、各アプリケーションのバージョン番号を入手できます。

IBM Unica 技術サポートの連絡先情報

IBM Unica 技術サポートとの連絡を取る方法については、IBM Unica 製品技術サポートの Web サイト (<http://www.unica.com/about/product-technical-support.htm>) を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

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

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのもと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

できます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴ、および ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、『www.ibm.com/legal/copytrade.shtml』をご覧ください。



Printed in Japan