Oracle® Communications Marketing and Advertising

Integration Guide

Release 5.1

E20559-01

May 2011

ORACLE®

Oracle Communications Marketing and Advertising Integration Guide, Release 5.1

E20559-01

Contents

3 Using Web Services to Create and Manage Campaigns

5 Using Oracle Communications Marketing and Advertising as an Ad Server

6 Integrating with Siebel Enterprise Marketing Suite

Preface

Oracle Communications Marketing and Advertising is designed to be integrated on many levels in the operator's environment. This document describes these integration points.

Audience

This document is intended for system administrators and integrators charged with setting up Marketing and Advertising to work with other components within the operator's environment.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Marketing and Advertising Release 5.1 documentation set :

- *Oracle Communications Marketing and Advertising Concepts Guide*
- *Oracle Communications Marketing and Advertising Installation Guide*
- *Oracle Communications Marketing and Advertising System Administrator's Guide*

1

Overview

Oracle Communications Marketing and Advertising offers multiple points of integration for external applications and services that are a part of operator environments. This document provides a description of those integration points.

There are seven main areas that support integration in this product:

- Operations Web Services
- Ad Server Provider API
- Siebel Marketing Interface
- Oracle Fusion MapViewer Interface
- Oracle Business Activity Monitoring
- Oracle Communications Billing and Revenue Management/Diameter Rf
- Subscriber Information SPI

Preparing the Web Services

To use the Marketing and Advertising Web services, you must deploy the Web service Enterprise Archive (EAR) file, **ocma-***WebServiceName***-ear.ear**, in the Oracle WebLogic Server container. By default, the EAR file for each Web service is not deployed by the Marketing and Advertising installation program (except for the Basic Collocated domain configuration).

The EAR file can be found in ***Oracle_Home* /ocma_5.1/applications**. You should also have set up appropriate WS-Security for the Web services, using the Administrative console. For more information on deploying applications contained in EAR files, see the Administration Console Online Help at

`http://download.oracle.com/docs/cd/E12840_01/wls/docs103/ConsoleHelp/taskhelp/applications/ApplicationOverview.html`.

For information on setting up WS-Security in Oracle WebLogic Server, see:

- *Oracle WebLogic Server Securing WebLogic Web Services*
  `http://download.oracle.com/docs/cd/E12840_01/wls/docs103/webserv_sec/index.html`
- *Oracle WebLogic Server Understanding WebLogic Security*
  `http://download.oracle.com/docs/cd/E12840_01/wls/docs103/secintro/index.html`
- *Web Services Security specifications*

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

WSDL

You can get the WSDL for the Web service either from the WEB-INF directory in the **Ocma*WebServiceName*.war** file archived in the **ocma-***WebServiceName***-ear.ear** file or from

`http://host:port/OcmaWebServiceNameWebServiceWSDL`,

where *host* and *port* point to a running instance of the Web tier of Marketing and Advertising. Determine the *WebServiceName* using the WebLogic Administration console.

Operations Web Services

The following Web services provided with Marketing and Advertising handle common system operations. The Account Manager Web service is used to create and manage organizations and users for the application. Four campaign-related Web service operations are provided for campaign management. The Statistics Manager Web service is used to query statistical information about campaigns. For additional information about each Web service see:

- Account Manager Web Service
- Connection Configuration Web Service
- Campaign Management Web Services
- Statistics Manager Web Service

Note: The.ear file that provides these web services, `ocma-ws-ear.ear`, is deployed by default only in the Basic collocated domain configuration. In other domain types, it must be deployed manually. The ear file can be found in `<Oracle_Home>/ocma_5.1/applications`. For more information on deploying applications in Oracle WebLogic Server, see *Administration Console Online Help* at `http://download.oracle.com/docs/cd/E12840_01/wls/docs103/ConsoleHelp/taskhelp/applications/ApplicationOverview.html`

Account Manager Web Service

Marketing and Advertising organizations and users can be created using the GUI by system owners and ad service providers, but it may often be useful to integrate these activities into business workflows. The Account Manager Web service allows you to create and manage organizations and users. For more information on this Web service, see "Using the Account Manager Web Service".

For additional information about organizations and users in Marketing and Advertising see *Oracle Communications Marketing and Advertising Users and User Organizations* in the *Oracle Communications Marketing and Advertising Concepts Guide*.

Connection Configuration Web Service

Connection configurations act as the messaging pipelines through which campaign messages flow. The connection configuration Web service can create, list and delete

connection configurations in Marketing and Advertising. For more information on this Web service, see "Using Web Services to Create and Manage Campaigns".

For additional information about connection configurations in Marketing and Advertising see *Connection Configuration* in the *Oracle Communications Marketing and Advertising Concepts Guide*.

Campaign Management Web Services

A campaign consists of ad content, created by advertisers, and a container for that content called a campaign space. The campaign space, created by ad space owners, is the entity in which ad content is delivered to subscribers. Executing a campaign is a multi-step process with different organizations contributing to the success of the campaign.

Campaign activities can be executed using a set of campaign management Web services. For more information regarding the users, entities and procedures involved in creating and executing a campaign see *Oracle Communications Marketing and Advertising Concepts Guide*.

Campaign Manager Web Service

The campaign manager Web service creates new campaigns and ad content. This Web service also deletes campaigns and ad content, lists campaigns, campaign categories, subscriber lists and ad content and stops campaigns. For more information on the campaign manager Web service, see "The Campaign Manager Web Service"

Campaign Space Approval Manager Web Service

The Campaign Space Approval Manager Web service approves or rejects campaign spaces. For more information on the Campaign Space Approval Manager Web service, see "The Campaign Space Approval Manager Web Service".

Campaign Space Manager Web Service

The Campaign Space Manager Web service creates and manages campaign spaces and also provides additional information related to campaign spaces. For more information on the Campaign Space Manager Web service, see "The Campaign Space Manager Web Service".

Statistics Manager Web Service

Campaign statistical information available in the Marketing and Advertising Web client can also be obtained using the Statistics Manager Web service. Statistical campaign information is valuable for reporting purposes and for determining the success of campaigns.

Organizations that make use of external reporting or dashboard applications may query Marketing and Advertising using the Statistics Manager Web service for campaign information in various formats. For more information on this Web service, see "Using Web Services to Query Campaign Statistics"

For additional information about the statistical information available in Marketing and Advertising see the *Statistics* section in the *Oracle Communications Marketing and Advertising Entities* chapter of *Oracle Communications Marketing and Advertising Concepts Guide*.

Ad Server Provider API

Ad content stored in Marketing and Advertising can be retrieved by external applications via the Ad Server Provider Web service for targeted advertising based on Open Mobile Alliance (OMA) Mob-Ad specifications. Marketing and Advertising functions as an ad content server concurrently receiving and tracking campaign statistics generated by an external application. For information on using Marketing and Advertising as a Mob-Ad server see "Using Oracle Communications Marketing and Advertising as an Ad Server".

Siebel Marketing Interface

Organizations utilizing *Siebel Enterprise Marketing* can integrate with Marketing and Advertising. In such an integration, Siebel delivers marketing offers/campaigns over Web service to Marketing and Advertising for execution. For information on integrating these two applications see "Integrating with Siebel Enterprise Marketing Suite".

Oracle Fusion MapViewer Interface

Marketing and Advertising queries and displays visual map information available from the *Oracle Fusion MapViewer* application Web service when integrated. Mapping data can be used in configuring location-based campaigns in a visual manner instead of providing text coordinate data. For information on using the Oracle MapViewer with Marketing and Advertising see "Integrating with Oracle Map Viewer"

Oracle Business Activity Monitoring

Marketing and Advertising can deliver real-time chargeable event information to Oracle Business Activity Monitoring (BAM). BAM gives business executives the ability to monitor their enterprise in real-time by supporting the building of real-time operational dashboards, monitoring and alerting applications over the Web. To see more about integrating with Oracle BAM, see "Integrating with Oracle Business Activity Monitoring".

Oracle Communications Billing and Revenue Management/Diameter Rf

Marketing and Advertising can produce charging detail records (CDRs) in Diameter Rf format for use with Oracle Communications Billing and Revenue Management or other charging systems that accept Diameter Rf requests. For more information on integrating with billing systems, see "Integrating with Oracle Communications Billing and Revenue Management".

Subscriber Information SPI

Service operators personalize campaigns and protect their subscribers from unwanted intrusion using Marketing and Advertising integration to subscriber information data sources. Marketing and Advertising provides a highly configurable implementation based on LDAPv3 and Oracle Enterprise Database functionality. Marketing and Advertising also provides a Subscriber Information SPI, to allow operators to create implementations optimized for whatever data sources they wish to use. For more information on the Subscriber Information SPI, see .

2

Using the Account Manager Web Service

The following chapter describes the Account Manager Web service. The Account Manager Web service configures organizations and users for the application GUI.

Authorized Users

Only users who belong to the system owner organization can create all organizations and users. Users who belong to an ad service provider organization can create their own ad space owner organizations and users and their own advertiser organizations and users.

Errors

Errors are returned in a SOAP Message that includes a human readable `faultstring` element that describes the error.

Return Data Structures

The following data structures are returned in the Account Manager web service.

User

Table 2–1 describes the User data structure.

Table 2–1 The User Data Structure

Element Name	Type	Description
role	tns:userRoleEnum	Describes the type of user. One of the following: ▪ ServiceOwnerUser ▪ NetworkProviderUser ▪ AdServiceProviderUser ▪ AdSpaceOwnerUser ▪ AdvertiserUser
name	xs:string	Username assigned to this user.
id	xs:long	User ID assigned to this user in the system.
managedOrganizationId	xs:long	System generated organization ID for the organization associated with this user.

Table 2–1 (Cont.) The User Data Structure

Element Name	Type	Description
email	xs:string	Email address of the user, given when the user was created.

Organization

Table 2–2 describes the Organization data structure.

Table 2–2 The Organization Data Structure

Element Name	Type	Description
id	xs:long	Organization ID assigned to this organization in the system.
name	xs:string	Organization name assigned at creation.
autoApproved	xs:boolean	Allow automatic approval of campaign spaces (significant only in the case of Ad Space Owners).
description	xs:string	Description given when the organization was created.
extID	xs:string	An external identifier for correlation outside of the system.
type	tns:orgTypeEnum	The type of the organization. One of the following: ■ NetworkProvider ■ AdServiceProvider ■ AdSpaceOwner ■ Advertiser

Operations

The following describes the operations available in the Account Manager web service.

Operation: createOrganization

Creates a new organization

Table 2–3 Input Message Parameter: complexType - createOrganization

Element	Type	Description
OrganizationType	tns:orgTypeEnum	One of the following: ■ NetworkProvider ■ AdServiceProvider ■ AdSpaceOwner ■ Advertiser
Name	xs:sting	Name for the organization in the system.
Description	xs:string	Optional. Descriptive string for the organization.
ExternalID	xs:string	An external identifier for correlation outside of the system.

Table 2–3 (Cont.) Input Message Parameter: complexType - createOrganization

Element	Type	Description
ParentId	xs:long	Optional. The system ID for the parent organization. Default value is managedOrganizationId. See Table 2–1.
AutoApproved	xs:boolean	Allow automatic approval of campaign spaces (significant only in the case of Ad Space Owners).]

createOrganizationResponse:

Table 2–4 Output Message Parameter: complexType: createOrganizationResponse

Element	Type	Description
Organization	tns:organization	See Table 2–2.

Operation:createUser

Creates a new user account for the Organization with a given orgId

Table 2–5 Input Message Parameter: complexType - createUser

Element	Type	Description
OrganizationId	xs:long	Optional. The system ID returned upon organization creation.] Default value is managedOrganizationId. See Table 2–1.
UserName	xs:sting	Name for this user.
Password	xs:string	Password for this user.
Email	xs:string	Email address to notify this user.

createUserResponse:

Table 2–6 Output Message Parameter: complexType - createUserResponse

Element	Type	Description
User	tns:user	See Table 2–1.

Operation: deleteOrganization

Deletes an organization

Table 2–7 Input Message Parameter: complexType - deleteOrganization

Element	Type	Description
OrganizationId	xs:long	ID assigned to this organization in the system.

Operation: deleteUser

Deletes a user from an organization

Table 2–8 Input Message Parameter: complexType - deleteUser

Element	Type	Description
Id	xs:long	User ID assigned to this organization in the system.

Operation:getOrganizationByExternalId

Retrieves an organization by the external ID associated with it when it was created

Table 2–9 Input Message Parameter: complexType - getOrganizationByExternalId

Element	Type	Description
ExternalId	xs:string	An external identifier for correlation outside of the system, assigned at creation.

getOrganizationByExternalIdResponse

Table 2–10 Output Message Parameter: complexType - getOrganizatinByExternalIdResponse

Element	Type	Description
Organization	tns:organization	See Table 2–2.

Operation:getOrganizationById

Retrieves an organization by the id assigned to it by the system when it was created

Table 2–11 Input Message Parameter: complextType - getOrganizationByID

Element	Type	Description
OrganizationId	xs:long	ID assigned to the organization by the system.

getOrganizationByIdResponse:

Table 2–12 Output Message Parameter: complexType - getOrganizationByIdResponse

Element	Type	Description
Organization	tns:organization	See Table 2–2.

Operation: getOrganizationByName

Retrieves an organization by the name assigned to it when it was created

Table 2–13 Input Message Parameter: complexType - getOrganizationByName

Element	Type	Description
OrganizationType	tns:orgTypeEnum	One of the following: ■ NetworkProvider ■ AdServiceProvider ■ AdSpaceOwner ■ Advertiser
Name	xs:string	Name for the organization in the system

Table 2–13 (Cont.) Input Message Parameter: complexType - getOrganizationByName

Element	Type	Description
ParentId	xs:long	Optional. The system ID for the parent organization. Default value is managedOrganizationId. See Table 2–1.

getOrganizationByNameResponse:

Table 2–14 Output Message Parameter: complexType - getOrganizationByNameResponse

Element	Type	Description
Organization	tns:organization	See Table 2–2.

Operation: listOrganizations

Gets a list of organizations

Table 2–15 Input Message Parameter: complexType - listOrganizations

Element	Type	Description
OrganizationType	tns:orgTypeEnum	One of the following: ■ NetworkProvider ■ AdServiceProvider ■ AdSpaceOwner ■ Advertiser
FirstResult	xs:int	The index position of the first result.
MaxResults	xs:int	The maximum number of results to return. **Note**: Do not use zero.
ParentId	xs:long	Optional. The system ID for the parent organization. Default value is managedOrganizationId. See Table 2–1.

listOrganizationResponse

Table 2–16 Output Message Parameter: complexType - listOrganizationsResponse

Element	Type	Description
OrganizationList	tns:organization	List of data structures described in Table 2–2.

Operation: listUsers

Returns a list of users for an Organization

Table 2–17 Input Message Parameter: complexType - listUsers

Element	Type	Description
OrganizationId	xs:long	Optional. ID assigned to the organization by the system. Default value is managedOrganizationId. See Table 2–1.
FirstResult	xs:int	Optional. Index position of the first result. Default value is 0.
MaxResults	xs:int	Optional. Maximum number of results to return. Default value is 0.

listUsersResponse

Table 2–18 Output Message Parameter: complexType - listUsersResponse

Element	Type	Description
UserList	tns:user	List of data structures described in Table 2–1.

3

Using Web Services to Create and Manage Campaigns

This chapter describes the four Web services available for creating and managing campaigns:

- The Connection Configuration Manager Web Service
- The Campaign Space Manager Web Service
- The Campaign Space Approval Manager Web Service
- The Campaign Manager Web Service

Authorized Users

Only users who belong to the appropriate organizations can perform the tasks of those organizations. For example, a Web service request authenticated by the username of an ad space owner user can create campaign spaces but not create campaigns.

Errors

Errors are returned in a SOAP Message that includes a human readable `faultstring` element that describes the error.

Return Data Structures

The following data structures are returned in the campaign Web services.

- AdServer Connection Configuration
- MsgService Connection Configuration
- BulkMsg Connection Configuration
- Traffic Hour
- BulkMsg Marketing Space
- Dynamic Ad Space
- MsgService Ad Space
- BulkMsg Marketing Campaign
- MsgService Ad Campaign
- Dynamic Ad Campaign

- Subscriber Category
- SMS Ad Content
- MMS Ad Content
- Simple Html Snippet Ad Content
- Text Ad Content
- Url Ad Content
- WAP Push Ad Content
- Subscriber Address List

AdServer Connection Configuration

Table 3–1 describes the AdServer connection configuration data structure. This data structure describes configurations used in campaigns interacting with external ad content servers.

Table 3–1 The AdServerCC Data Structure

Element Name	Type	Description
id	xs:long	Connection configuration ID assigned by the system
name	xs:string	Name of the configuration
adSpaceOwnerId	xs:long	ID of the ad space owner to whom this configuration is assigned
networkProviderId	xs:long	ID of the Network Provider with which this configuration is associated
minimumPrice	xs:float	Minimum auction price for ads created using this configuration; the currency unit is set by the System Administrator; range is 0 to 9999.999
channelType	tns:channelType	Type of channel: one of the following: ■ TEXT ■ URL ■ SIMPLE HTML SNIPPET
maxAdSize	xs:int	Maximum number of characters allowed in an ad; range is 0 to 2147483647 characters - limited by channel type
adDelimitorPrefix	xs:string	String that sets off the top of the ad
adDelimitorPostfix	xs:string	String that sets off the bottom of the ad
shortCodes	xs:string	Array of short codes to be associated with ads created with this configuration
applicationId	xs:string	Identifies the application for the configuration's message delivery system in a charging data record (CDR); must be unique within the Marketing and Advertising system

Table 3–1 (Cont.) The AdServerCC Data Structure

Element Name	Type	Description
msgServiceType	tns:msgServiceType Enum	Type of message service: one of the following: ■ **A2P_POSTFIX**: Insert ad into traffic flowing from an application to an end user ■ **P2A_NEWMESSAGE**: Return ad to end user in response to traffic flowing from the user to an application

MsgService Connection Configuration

Table 3–2 describes the MsgService connection configuration data structure. This data structure describes configurations used in advertising campaigns.

Table 3–2 The MsgServiceCC Data Structure

Element Name	Type	Description
id	xs:long	Connection configuration ID assigned by the system
name	xs:string	Name of the configuration
adSpaceOwnerId	xs:long	ID of the ad space owner to whom this configuration is assigned
networkProviderId	xs:long	ID of the Network Provider with which this configuration is associated
minimumPrice	xs:float	Minimum auction price for ads created using this configuration; the currency unit is set by the System Administrator; range is 0 to 9999.999
channelType	tns:channelType	Type of channel: one of the following: ■ SMS ■ MMS
maxAdSize	xs:int	Maximum number of characters allowed in an ad; range is 0 to 2147483647 characters - limited by channel type
adDelimitorPrefix	xs:string	String that sets off the top of the ad
adDelimitorPostfix	xs:string	String that sets off the bottom of the ad
shortCodes	xs:string	Array of short codes to be associated with ads created with this configuration
applicationId	xs:string	Identifies the application for the configuration's message delivery system in a charging data record (CDR); must be unique within the Marketing and Advertising system

Table 3–2 (Cont.) The MsgServiceCC Data Structure

Element Name	Type	Description
msgServiceType	tns:msgServiceType Enum	Type of message service: one of the following: ■ **A2P_POSTFIX**: Insert ad into traffic flowing from an application to an end user ■ **P2A_NEWMESSAGE**: Return ad to end user in response to traffic flowing from the user to an application
ccFeatures	feature and price	List of premium features and pricing used in connection configuration including: ■ demography ■ geo (location) ■ personalization ■ response management ■ subscriber capping

BulkMsg Connection Configuration

Table 3–3 describes the Bulk Messaging connection configuration data structure. This data structure describes configurations used in marketing campaigns.

Table 3–3 The BulkMsgCC Data Structure

Element Name	Type	Description
id	xs:long	Connection configuration ID assigned by the system
name	xs:string	Name of the configuration
adSpaceOwnerId	xs:long	ID of the ad space owner to whom this configuration is assigned.
networkProviderId	xs:long	ID of the Network Provider with which this configuration is associated
minimumPrice	xs:float	Minimum price for ads created using this configuration; the currency unit is set by the System Administrator; range is 0 to 9999.999
channelType	tns:channelType	Type of channel: one of the following: ■ SMS ■ MMS ■ WAPPUSH
thList	tns:trafficHour	List that sets the beginning and ending times for premium campaigns. Start time must be less than end time. See Table 3–4 for description of trafficHour data structure
maxAdSize	xs:int	Maximum number of characters allowed in an ad; range is 0 to 2147483647 characters

Table 3–3 (Cont.) The BulkMsgCC Data Structure

Element Name	Type	Description
adDelimitorPrefix	xs:string	String that sets off the top of the ad
adDelimitorPostfix	xs:string	String that sets off the bottom of the ad
largeAccount	xs:string	Short code to be associated with marketing messages created with this configuration
isPremium	xs:boolean	Premium marketing campaigns have more control over scheduling with a configurable start date and priority over non-premium in sending messages during peak traffic hour
ccFeatures	feature and price	List of premium features and pricing used in connection configuration including: ■ demography ■ geo (location) ■ personalization ■ response management ■ subscriber capping

Traffic Hour

Table 3–4 describes the Traffic Hour data structure. This data structure defines when a Bulk Service configuration is available for traffic.

Table 3–4 The TrafficHour Data Structure

Element Name	Type	Description
dayOfWeek	tns:dayOfWeek	Day of the week: one of the following: ■ SUNDAY ■ MONDAY ■ TUESDAY ■ WEDNESDAY ■ THURSDAY ■ FRIDAY ■ SATURDAY
startHour	xs:byte	Start hour
endHour	xs:byte	End hour
startMinute	xs:byte	Start minute
endMinute	xs:byte	End minute

BulkMsg Marketing Space

Table 3–5 describes the Bulk Messaging Marketing Space data structure. This data structure describes campaign space used in marketing campaigns.

Table 3–5 The BulkMsgMarketingSpace Data Structure

Element Name	Type	Description
id	xs:long	ID of the Bulk Messaging Marketing Space assigned by the system
name	xs:string	Name of the space
adSpaceOwnerId	xs:long	ID of the ad space owner to whom this space belongs
bulkCcEntityID	xs:long	ID of the connection configuration on which this marketing space is based
description	xs:string	Short descriptive text identifying space
endMarketTime	xs:dateTime	Date and time at which campaigns based on this space end. These values cannot overlap with those of any other of this ad space owner's marketing spaces.
endServiceTime	xs:dateTime	Identical to endMarketTime
minimumPrice	xs:float	Sets the price per ad to use this space. The currency unit is set by the System Administrator. Range is 0 to 9999.999
reason	xs:string	Explanation for STOPPED or REJECTED state. Null otherwise.
startMarketTime	xs:dateTime	Date and time at which campaigns based on this space can begin. These values cannot overlap with those of any other of this ad space owner's marketing spaces
startServiceTime	xs:dateTime	Identical to startMarketTime
state	tns:approvedEntityStateEnum	State of the space: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED
minimumListSize	xs:int	Minimum number of destination addresses per list for this space

Dynamic Ad Space

Table 3–6 describes the Dynamic Ad Space data structure. This data structure describes campaign space used in marketing campaigns.

Table 3–6 The Dynamic Ad Space Data Structure

Element Name	Type	Description
id	xs:long	ID of the Dynamic Space assigned by the system
name	xs:string	Name of the space

Table 3–6 (Cont.) The Dynamic Ad Space Data Structure

Element Name	Type	Description
adSpaceOwnerId	xs:long	ID of the ad space owner to whom this space belongs
bulkCcEntityID	xs:long	ID of the connection configuration on which this marketing space is based
description	xs:string	Short descriptive text identifying space
endMarketTime	xs:dateTime	Date and time at which campaigns based on this space end. These values cannot overlap with those of any other of this ad space owner's marketing space.
endServiceTime	xs:dateTime	Identical to endMarketTime
minimumPrice	xs:float	Sets the price per ad to use this space. The currency unit is set by the System Administrator. Range is 0 to 9999.999
reason	xs:string	Explanation for STOPPED or REJECTED state. Null otherwise.
startMarketTime	xs:dateTime	Date and time at which campaigns based on this space can begin. These values cannot overlap with those of any other of this ad space Owner's marketing spaces
startServiceTime	xs:dateTime	Identical to startMarketTime
state	tns:approvedEntityStateEnum	State of the space: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED
minimumListSize	xs:int	Minimum number of destination addresses per list for this space

MsgService Ad Space

Table 3–7 describes the MsgService ad space data structure. This data structure describes campaign space used in advertising campaigns.

Table 3–7 The MsgServiceAdSpace Data Structure

Element Name	Type	Description
id	xs:long	ID of the Message Service ad space assigned by the system
name	xs:string	Name of the space
adSpaceOwnerId	xs:long	ID of the ad space owner to whom this space belongs
bulkCcEntityID	xs:long	ID of the Connection Configuration on which this Ad space is based

Table 3–7 (Cont.) The MsgServiceAdSpace Data Structure

Element Name	Type	Description
description	xs:string	Short descriptive text identifying space
endMarketTime	xs:dateTime	Date and time at which bidding on this space ends
endServiceTime	xs:dateTime	Date and time at which campaigns based on this space must end
minimumPrice	xs:float	Sets the minimum bid per ad to use this space. The currency unit is set by the System Administrator. Range is 0 to 9999.999
reason	xs:string	Explanation for STOPPED or REJECTED state. Null otherwise.
startMarketTime	xs:dateTime	Date and time at which bidding on this space begins
startServiceTime	xs:dateTime	Date and time at which campaigns based on this space may start
state	tns:approvedEntityStateEnum	State of the space: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED

BulkMsg Marketing Campaign

Table 3–8 describes the Bulk Messaging Marketing campaign data structure. This data structure defines a marketing campaign.

Table 3–8 The BulkMsgMarketingCampaign Data Structure

Element Name	Type	Description
adContentId	xs:long	ID for the ad content assigned by the system
adSpaceId	xs:long	ID for the Ad space assigned by the system
advOwnerId	xs:long	ID for the advertiser who owns this campaign as assigned by the system
manuallyStopped	xs:boolean	If true, the campaign was stopped manually by the owner
messageLimit	xs:int	The maximum number of messages to be sent in the campaign
modifiable	xs:boolean	Whether this campaign can be modified
price	xs:float	The price per ad to use this space. The currency unit is set by the System Administrator. Range is 0 to 9999.999

Table 3–8 (Cont.) The BulkMsgMarketingCampaign Data Structure

Element Name	Type	Description
reason	xs:string	Explanation for STOPPED or REJECTED state. Null otherwise.
state	tns:approvedEntityStateEnum	State of the campaign: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED
startTime	xs:dateTime	Date and time the campaign starts
subscrAddressListId	xs:long	ID of the Subscriber Address List assigned by the system. For marketing campaigns that use uploaded lists
subscrCategories	tns:subscriberCategory	List of subscriber categories made available to Advertisers by the operator. For marketing campaigns that use categories. See Table 3–11 for more information.

MsgService Ad Campaign

Table 3–9 describes the MsgService Ad Campaign data structure. This data structure defines an advertising campaign.

Table 3–9 The MsgServiceAdCampaign Data Structure

Element Name	Type	Description
adContentId	xs:long	ID for the ad content assigned by the system
adSpaceId	xs:long	ID for the Ad space assigned by the system
advOwnerId	xs:long	ID for the advertiser who owns this campaign as assigned by the system
manuallyStopped	xs:boolean	If true, the campaign was stopped manually by the owner
messageLimit	xs:int	The maximum number of messages to be sent in the campaign
modifiable	xs:boolean	Whether this campaign can be modified
price	xs:float	The price per ad to use this space. The currency unit is set by the System Administrator. Range is 0 to 9999.999
reason	xs:string	Explanation for STOPPED or REJECTED state. Null otherwise.

Table 3–9 (Cont.) The MsgServiceAdCampaign Data Structure

Element Name	Type	Description
state	tns:approvedEntityStateEnum	State of the campaign: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED
maxMessagesPerSub	xs:int	Maximum number of messages to be sent to an individual subscriber

Dynamic Ad Campaign

Table 3–10 describes the Dynamic Ad Campaign data structure. This data structure defines a dynamic campaign.

Table 3–10 The DynamicAdCampaign Data Structure

Element Name	Type	Description
adContentId	xs:long	ID for the ad content assigned by the system
adSpaceId	xs:long	ID for the Ad space assigned by the system
advOwnerId	xs:long	ID for the advertiser who owns this campaign as assigned by the system
manuallyStopped	xs:boolean	If true, the campaign was stopped manually by the owner
messageLimit	xs:int	The maximum number of messages to be sent in the campaign
modifiable	xs:boolean	Whether this campaign can be modified
price	xs:float	The price per ad to use this space. The currency unit is set by the System Administrator. Range is 0 to 9999.999
reason	xs:string	Explanation for STOPPED or REJECTED state. Null otherwise.
state	tns:approvedEntityStateEnum	State of the campaign: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED
maxMessagesPerSub	xs:int	Maximum number of messages to be sent to an individual subscriber

Subscriber Category

Table 3–11 describes the Subscriber Category data structure. This data structure defines the categories that an operator has made available to Advertisers to use in personalizing and targeting ads for marketing campaigns.

Table 3–11 The SubscriberCategory Data Structure

Element Name	Type	Description
typeName	xs:string	Name of the category defined by the operator. For example: age
values	xs:string	Desired subsets of the category. For example: 11-18 and 18-32. There can be multiple values per typeName.

SMS Ad Content

Table 3–12 describes the SMS Ad Content data structure. This data structure describes content to be delivered using the SMS channel.

Table 3–12 The SmsAdContent Data Structure

Element Name	Type	Description
id	xs:long	ID for the content assigned by the system
name	xs:string	Name of the content
advOwnerId	xs:long	ID for the advertiser who created this content
textWithTokens	xs:string	Textual content of the ad, with tokens for personalization
textWithoutTokens	xs:string	Textual content of the ad

MMS Ad Content

Table 3–13 describes the MMS Ad Content data structure. This data structure describes content to be delivered using the MMS channel.

Table 3–13 The MmsAdContent Data Structure

Element Name	Type	Description
id	xs:long	ID for the content assigned by the system
name	xs:string	Name of the content
advOwnerId	xs:long	ID for the advertiser who created this content
textWithTokens	xs:string	Textual content of the ad, with tokens for personalization
textWithoutTokens	xs:string	Textual content of the ad
imageType	tns:mmsPictureEnum	Type of the image to be sent as content. One of the following: ■ PNG ■ GIF ■ JPG

Table 3–13 (Cont.) The MmsAdContent Data Structure

Element Name	Type	Description
subjectWithoutTokens	xs:string	Subject for the ad message

Simple Html Snippet Ad Content

Table 3–14 describes the Simple Html Snippet Ad Content data structure. This data structure describes the content to be delivered using the Simple Html Snippet channel.

Table 3–14 The SimpleHtmlSnippetAdContent Data Structure

Element Name	Type	Description
id	xs:long	ID for the content assigned by the system
name	xs:string	Name of the content
advOwnerId	xs:long	ID for the advertiser who created this content
textWithTokens	xs:string	Textual content of the ad, with tokens for personalization
textWithoutTokens	xs:string	Textual content of the ad
activeLink	xs:string	Specifies embedded clickable link URL in banner

Text Ad Content

Table 3–15 describes the Text Ad Content data structure. This data structure describes the content to be delivered using channels that support text ad content.

Table 3–15 The TextAdContent Data Structure

Element Name	Type	Description
id	xs:long	ID for the content assigned by the system
name	xs:string	Name of the content
advOwnerId	xs:long	ID for the advertiser who created this content
textWithTokens	xs:string	Textual content of the ad, with tokens for personalization
textWithoutTokens	xs:string	Textual content of the ad
activeLink	xs:string	Specifies embedded clickable link URL in banner

Url Ad Content

Table 3–16 describes the Url Ad Content data structure. This data structure describes the content to be delivered using channels that support url ad content.

Table 3–16 The UrlAdContent Data Structure

Element Name	Type	Description
id	xs:long	ID for the content assigned by the system
name	xs:string	Name of the content
advOwnerId	xs:long	ID for the advertiser who created this content
textWithTokens	xs:string	Textual content of the ad, with tokens for personalization
textWithoutTokens	xs:string	Textual content of the ad
activeLink	xs:string	Specifies embedded clickable link URL in banner

WAP Push Ad Content

Table 3–17 describes the WAP Push Ad Content data structure. This data structure describes the content to be delivered using the WAP Push channel.

Table 3–17 The WapPushAdContent Data Structure

Element Name	Type	Description
id	xs:long	ID for the content assigned by the system
name	xs:string	Name of the content
advOwnerId	xs:long	ID for the advertiser who created this content
textWithTokens	xs:string	Textual content of the ad, with tokens for personalization
textWithoutTokens	xs:string	Textual content of the ad
activeLink	xs:string	Specifies embedded clickable link URL in banner

Subscriber Address List

Table 3–18 describes the Subscriber Address List data structure. This data structure describes the subscriber address lists provided by an advertiser for marketing campaigns.

Table 3–18 The SubscriberAddressList Data Structure

Element Name	Type	Description
id	xs:long	ID for the list assigned by the system
name	xs:string	Name of the list
size	xs:int	Size of the list

The Connection Configuration Manager Web Service

The connection configuration Manager Web service creates, deletes, and lists existing connection configurations.

Operations

The following describes the operations available in the connection configuration Manager Web service. Additional details can be found in the WSDL.

Operation: deleteConnectionConfiguration

Deletes an existing connection configuration

Table 3–19 Input Message Parameter: complexType - deleteConnectionConfiguration

Element	Type	Description
ConnectionConfigurationId	xs:long	Connection configuration ID assigned by the system

Operation: createAdServerCc

Creates a connection configuration to be used by an external ad server

Table 3–20 Input Message Parameter: complexType - createAdServerCc

Element	Type	Description
NetworkProviderId	xs:long	Optional. ID of the Network Provider with which this configuration is associated. Default value is managedOrganizationId.
Name	xs:string	Name of the configuration
ChannelType	tns:channelType	Type of channel: one of the following: ■ SMS ■ MMS ■ WAPPUSH
AdSpaceOwnerId	xs:long	ID of the ad space owner to whom this configuration is assigned [optional - can be null]
MinPrice	xs:float	Optional. Minimum auction price for ads created using this configuration; the currency unit is set by the System Administrator; range is 0 to 9999.999. Default value is 0.
MaxAdSize	xs:int	Optional. Maximum number of characters allowed in an ad; range is 0 to 2147483647 characters - limited by channel type. Default value is 0.

Table 3–20 (Cont.) Input Message Parameter: complexType - createAdServerCc

Element	Type	Description
SpAppId	xs:string	Identifies application using the connection configuration
DeliveryNotifications	xs:boolean	Optional. Determines if delivery notifications should be sent for activity. Default value is False.
ccFeatures	feature and price	Optional. List of premium features and pricing used in connection configuration
MaxAdsReturnedPerRequest	xs:int	Maximum number of ads returned per request for connection configuration

createAdServerCcResponse

Table 3–21 Output Message Parameter: complexType - createAdServerCcResponse

Element	Type	Description
AdServerCc	tns:msgServiceCc	List of data structures described in Table 3–1

Operation: createMsgServiceCc

Creates a connection configuration to be used for advertising campaigns

Table 3–22 Input Message Parameter: complexType - createMsgServiceCc

Element	Type	Description
NetworkProviderId	xs:long	Optional. ID of the Network Provider with which this configuration is associated. Default value is managedOrganizationId.
Name	xs:string	Name of the configuration
TriggerType	tns:triggerType	Type of channel: one of the following: ■ SMS ■ MMS
ChannelType	tns:channelType	Type of channel: one of the following: ■ SMS ■ MMS
AdSpaceOwnerId	xs:long	ID of the ad space owner to whom this configuration is assigned
MinPrice	xs:float	Optional. Minimum auction price for ads created using this configuration; the currency unit is set by the System Administrator; range is 0 to 9999.999. Default value is 0.

Table 3–22 (Cont.) Input Message Parameter: complexType - createMsgServiceCc

Element	Type	Description
MaxAdSize	xs:int	Optional. Maximum number of characters allowed in an ad; range is 0 to 2147483647 characters - limited by channel type. Default value is 0.
AdDelimPrefix	xs:string	Optional. String that sets off the top of the ad.
AdDelimPostfix	xs:string	Optional. String that sets off the bottom of the ad.
MaxAdsReturnedPerRequest	xs:int	Maximum number of ads returned per request for connection configuration
MsgServiceType	tns:msgServiceTypeEnum	Type of message service: one of the following: ■ **A2P_POSTFIX**: Insert ad into traffic flowing from an application to an end user ■ **P2A_NEWMESSAGE**: Return ad to end user in response to traffic flowing from the user to an application
ShortCodes	xs:string	Array of short codes to be associated with ads created with this configuration
ApplicationId	xs:string	Identifies the application for the configuration's message delivery system in a charging data record (CDR); must be unique within the Marketing and Advertising system
DeliveryNotifications	xs:boolean	Optional. If true, users can subscribe for notification that a message was delivered. Default value is False.
ccFeatures	feature and price	Optional. List of premium features and pricing used in connection configuration

createMsgServiceCcResponse

Table 3–23 Output Message Parameter: complexType - createMsgServiceCcResponse

Element	Type	Description
MsgServiceCc	tns:msgServiceCc	See Table 3–2

Operation: createBulkMessagingCc

Creates a connection configuration to be used for marketing campaigns

Table 3–24 Input Message Parameter: complexType - createBulkMessagingCc

Element	Type	Description
NetworkProviderId	xs:long	Optional. ID of the Network Provider with which this configuration is associated. Default value is managedOrganizationId.
Name	xs:string	Name of the configuration
ChannelType	tns:channelType	Type of channel: one of the following: ■ SMS ■ MMS ■ WAPPUSH
TrafficHours	tns:trafficHour	See Table 3–4
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner to whom this configuration is assigned.
MinPrice	xs:float	Optional. Minimum auction price for ads created using this configuration; the currency unit is set by the System Administrator; range is 0 to 9999.999. Default value is 0.
MaxAdSize	xs:int	Optional. Maximum number of characters allowed in an ad; range is 0 to 2147483647 characters - limited by channel type. Default value is 0.
AdDelimPrefix	xs:string	String that sets off the top of the ad. Deprecated.
AdDelimPostfix	xs:string	String that sets off the bottom of the ad Deprectated.
ShortCode	xs:string	Short code to be associated with ads created with this configuration
Premium	xs:boolean	Premium marketing campaigns have more control over scheduling with a configurable start date and priority over non-premium in sending messages during peak traffic hour. Deprecated.
DeliveryNotifications	xs:boolean	If true, users can subscribe for notification that a message was delivered
ccFeatures	feature and price	Optional. List of premium features and pricing used in connection configuration.

createBulkMessagingCcResponse

Table 3–25 Output Message Parameter: complexType - createBulkServiceCcResponse

Element	Type	Description
MsgServiceCc	tns:bulkMsgCc	See Table 3–2

Operation: listAdServerCc

Retrieves a list of current connection configurations for external ad server(s)

Table 3–26 Input Message Parameter: complexType - listMsgAdServerCc

Element	Type	Description
NetworkProviderId	xs:long	Optional. ID of the Network Provider with which these configurations are associated. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of configurations where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.
ChannelType	tns:channelType	Optional. Type of channel: one of the following: ▪ SMS ▪ MMS

listMsgAdServerCcResponse

Table 3–27 Output Message Parameter: complexType - listMsgServiceCcResponse

Element	Type	Description
MsgServiceCcList	tns:msgServiceCc	List of data structures described in Table 3–2.

Operation: listMsgServiceCc

Retrieves a list of current connection configurations for advertising campaigns

Table 3–28 Input Message Parameter: complexType - listMsgServiceCc

Element	Type	Description
NetworkProviderId	xs:long	Optional. ID of the Network Provider with which these configurations are associated. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of configurations where fetching should start. Default value is 0.

Table 3–28 (Cont.) Input Message Parameter: complexType - listMsgServiceCc

Element	Type	Description
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.
ChannelType	tns:channelType	Optional. Type of channel: one of the following: ■ SMS ■ MMS

listMsgServiceCcResponse

Table 3–29 Output Message Parameter: complexType - listMsgServiceCcResponse

Element	Type	Description
MsgServiceCcList	tns:msgServiceCc	List of data structures described in Table 3–2

Operation: listBulkMsgCc

Retrieves a list of current connection configurations for advertising campaigns

Table 3–30 Input Message Parameter: complexType - listBulkMsgCc

Element	Type	Description
NetworkProviderId	xs:long	Optional. ID of the Network Provider with which these configurations are associated. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of configurations where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.
ChannelType	tns:channelType	Optional. Type of channel: one of the following: ■ TEXT ■ URL ■ SIMPLE_HTML_SNIPPET [optional - can be null]

listBulkMsgCcResponse

Table 3–31 Output Message Parameter: complexType - listBulkMsgCcResponse

Element	Type	Description
BulkMsgCcList	tns:msgServiceCc	List of data structures described in Table 3–3

The Campaign Space Manager Web Service

The Campaign Space Manager Web service creates and manages campaign spaces; searches for connection configurations by ad space owner; approves or rejects campaigns; and retrieves campaigns by type.

Operations

The following describes the operations available in the Campaign Space Manager Web service. Full details can be found in the WSDL.

- Operation: createBulkMsgMarketingSpace
- Operation: createDynamicAdSpace
- Operation: createMsgServiceAdSpace
- Operation: deleteCampaignSpace
- Operation: listBulkMsgMarketingSpaceByAdSpaceOwner
- Operation: listMsgServiceAdSpaceByAdSpaceOwner
- Operation: listBulkMsgCcByAdSpaceOwner
- Operation: listMsgServiceCcByAdSpaceOwner
- Operation: listDynamicSpaceByAdSpaceOwner
- Operation: approveCampaign
- Operation: rejectCampaign
- Operation: listBulkMsgMarketingCampaigns
- Operation: listDynamicAdCampaigns
- Operation: listMsgServiceAdCampaigns

Operation: createBulkMsgMarketingSpace

Creates a campaign space to be used for marketing campaigns

Table 3–32 Input Message Parameter: complexType - createBulkMsgMarketingSpace

Element	Type	Description
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner who is creating this space. Default value is managedOrganizationId.
Name	xs:string	Name of the space
Description	xs:string	Short textual description
ImageType	tns:imageTypeEnum	Type of the image that is being provided as an icon for display in the GUI. One of the following: ■ PNG ■ GIF ■ JPG
Image	xs:base64Binary	Image for the icon

Table 3–32 (Cont.) Input Message Parameter: complexType -

Element	Type	Description
ConnectionConfigurationId	xs:long	ID assigned by the system for the configuration on which this space is based
StartTime	xs:dateTime	Date and time at which this space is available for campaigns
EndTime	xs:dateTime	Date and time at which this space is no longer available for campaigns
MinPrice	xs:float	Optional. The minimum price per ad for campaigns based on this space. Default value is 0.
MinSubscriberListSize	xs:int	Optional. The minimum number of subscriber addresses for campaigns based on this space. Default value is 0.

createBulkMsgMarketingSpaceResponse

Table 3–33 Output Message Parameter: complexType - createBulkMsgMarkegingSpaceResponse

Element	Type	Description
BulkMsgMarketingSpace	tns:bulkMsgMarketingSpace	See Table 3–5.

Operation: createDynamicAdSpace

Creates a campaign space to be used for dynamic campaigns

Table 3–34 Input Message Parameter: complexType - createDynamicAdSpace

Element	Type	Description
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner who is creating this space [optional - can be null] Default value is managedOrganizationId.
ConnectionConfigurationId	xs:long	ID assigned by the system for the configuration on which this space is based
Name	xs:string	Name of the space
Description	xs:string	Short textual description
ImageType	tns:imageTypeEnum	Type of the image that is being provided as an icon for display in the GUI. One of the following: ■ PNG ■ GIF ■ JPG
Image	xs:base64Binary	Image for the icon

Table 3–34 (Cont.) Input Message Parameter: complexType - createDynamicAdSpace

Element	Type	Description
StartServiceTime	xs:dateTime	Date and time at which this space is available for campaigns
EndServiceTime	xs:dateTime	Date and time at which this space is no longer available for campaigns
MinPrice	xs:float	The minimum price per ad for campaigns based on this space
DynamicSpaceFilters	ns:string, string	Optional. Filters to be used in the dynamic space.
DynamicSpaceCategories	ns:string, string	Optional. Demographic categories fro the subscriber schema to be included in the ad space.

createDynamicAdSpaceResponse

Table 3–35 Output Message Parameter: complexType - createDynamicAdSpaceResponse

Element	Type	Description
BulkMsgMarketingSpace	tns:bulkMsgMarketingSpace	See Table 3–5.

Operation: createMsgServiceAdSpace

Creates a campaign space to be used for advertising campaigns

Table 3–36 Input Message Parameter: complexType - createMsgServiceAdSpace

Element	Type	Description
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner who is creating this space. Default value is managedOrganizationId.
Name	xs:string	Name of the space
Description	xs:string	Short textual description
ImageType	tns:imageTypeEnum	Type of the image that is being provided as an icon for display in the GUI. One of the following: ■ PNG ■ GIF ■ JPG
Image	xs:base64Binary	Image for the icon
ConnectionConfigurationId	xs:long	ID assigned by the system for the configuration on which this space is based
StartMarketTime	xs:dateTime	Date and time at which this space is available for auction

Table 3–36 (Cont.) Input Message Parameter: complexType - createMsgServiceAdSpace

Element	Type	Description
EndMarketTime	xs:dateTime	Date and time at which this space is no longer available for auction
StartServiceTime	xs:dateTime	Date and time at which this space is available for campaigns
EndServiceTime	xs:dateTime	Date and time at which this space is no longer available for campaigns
MinPrice	xs:float	Optional. The minimum price per ad for campaigns based on this space. Default value is 0.
spaceType	tns:adSpaceTypeEnum	
pricePerCampaign	xs:float	

createMsgServiceAdSpaceResponse

Table 3–37 Output Message Parameter: complexType - createMsgServiceAdSpaceResponse

Element	Type	Description
MsgServiceAdSapce	tns:msgServiceAdSpace	See Table 3–7.

Operation: deleteCampaignSpace

Deletes an existing campaign space

Table 3–38 Input Message Parameter: complexType - deleteCampaignSpace

Element	Type	Description
CampaignSpaceId	xs:long	ID of the space assigned by the system.

Operation: listBulkMsgMarketingSpaceByAdSpaceOwner

Fetches a list of an ad space owner's marketing spaces

Table 3–39 Input Message Parameter: complexType - listBulkMsgMarketingSpaceByAdSpaceOwner

Element	Type	Description
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner whose marketing space you wish to fetch. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list where fetching should start. Default value is 0.
MaxResults	xs:int	Optional. The maximum number of results to return. Default value is 0.

listBulkMsgMarketingSpaceByAdSpaceOwnerResponse

Table 3–40 Output Message Parameter: complexType - listBulkMsgMarketingSpaceByAdSpaceOwnerResponse

Element	Type	Description
BulkMsgMarketingSpaceList	tns:bulkMsgMarketingSpace	List of data structures described in Table 3–5.

Operation: listMsgServiceAdSpaceByAdSpaceOwner

Fetches a list of an ad space owner's Ad spaces

Table 3–41 Input Message Parameter: complexType - listMsgServiceAdSpaceByAdSpaceOwner

Element	Type	Description
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner whose advertising space you wish to fetch. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list where fetching should start. Default value is 0.
MaxResults	xs:int	Optional. The maximum number of results to return. Default value is 0.

listMsgServiceAdSpaceByAdSpaceOwnerResponse

Table 3–42 Output Message Parameter: complexType - listMsgServiceAdSpaceByAdSpaceOwnerResponse

Element	Type	Description
MsgServiceAdpaceList	tns:msgServiceAdSpace	List of data structures described in Table 3–7.

Operation: listBulkMsgCcByAdSpaceOwner

Fetches a list of the Bulk Msg connection configurations assigned to a particular ad space owner

Table 3–43 Input Message Parameter: complexType - listBulkMsgCcByAdSpaceOwner

Element	Type	Description
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner whose BulkMsg connection configurations you wish to fetch. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list where fetching should start. Default value is 0.

Table 3–43 (Cont.) Input Message Parameter: complexType -

Element	Type	Description
MaxResults	xs:int	Optional. The maximum number of results to return. Defualt value is 0.
ChannelType	tns:channelType	Optional. Type of channel: one of the following: ■ SMS ■ MMS ■ WAPPUSH

listBulkMsgCcByAdSpaceOwnerResponse

Table 3–44 Output Message Parameter: complexType - listBulkMsgCcByAdSpaceOwnerResponse

Element	Type	Description
BulkCcList	tnsbulkMsgCc	List of data structures described in Table 3–3.

Operation: listMsgServiceCcByAdSpaceOwner

Fetches a list of the MsgService connection configurations assigned to a particular ad space owner

Table 3–45 Input Message Parameter: complexType - listMsgServiceCcByAdSpaceOwner

Element	Type	Description
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner whose MsgService connection configurations you wish to fetch. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list where fetching should start. Default value is 0.
MaxResults	xs:int	Optional. The maximum number of results to return. Default value is 0.
ChannelType	tns:channelType	Optional. Type of channel: one of the following ■ SMS ■ MMS ■ TEXT ■ URL ■ SIMPLE_HTML_SNIPPET

listMsgServiceCcByAdSpaceOwnerResponse

Table 3–46 Output Message Parameter: complexType - listMsgServiceCcByAdSpaceOwnerResponse

Element	Type	Description
MsgServiceAdspaceList	tns:msgServiceAdSpace	List of data structures described in Table 3–7.

Operation: listDynamicSpaceByAdSpaceOwner

Fetches a list of the dynamic spaces assigned to a particular ad space owner

Table 3–47 Input Message Parameter: complexType - listDynamicSpaceByAdSpaceOwner

Element	Type	Description
AdSpaceOwnerId	xs:long	Optional. ID of the ad space owner whose MsgService connection configurations you wish to fetch. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list where fetching should start. Default value is 0.
MaxResults	xs:int	Optional. The maximum number of results to return. Default vaule is 0.

listDynamicSpaceByAdSpaceOwnerResponse

Table 3–48 Output Message Parameter: complexType - listDynamicSpaceByAdSpaceOwnerResponse

Element	Type	Description
DynamicAdspaceList	tns:DynamicAdSpace	List of data structures described in Table 3–6.

Operation: approveCampaign

Approves a campaign by ID

Table 3–49 Input Message Parameter: complexType - approveCampaign

Element	Type	Description
Id	xs:long	ID of the campaign assigned by the system.

approveCampaignResponse

Table 3–50 Output Message Parameter: complexType - approveCampaignResponse

Element	Type	Description
CampaignState	tns:approvedEntityStateEnum	Current state of campaign: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED

Operation: rejectCampaign

Rejects a campaign by ID

Table 3–51 Input Message Parameter: complexType - rejectCampaign

Element	Type	Description
Id	xs:long	ID of the campaign assigned by the system.
Reason	xs:string	A brief description of why the campaign was rejected.

rejectCampaignResponse

Table 3–52 Output Message Parameter: complexType - rejectCampaignResponse

Element	Type	Description
CampaignState	tns:approvedEntityStateEnum	Current state of campaign: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED

Operation: listBulkMsgMarketingCampaigns

Fetches a list of current BulkMsg marketing campaigns

Table 3–53 Input Message Parameter: complexType - listBulkMsgMarketingCampaigns

Element	Type	Description
CampaignSpaceId	xs:long	ID of the campaign space assigned by the system.
FirstResult	xs:int	Optional. The position in the list where fetching should start. Default value is 0.
MaxResults	xs:int	Optional. The maximum number of results to return. Default value is 0.

listBulkMsgMarketingCampaignsResponse

Table 3–54 Output Message Parameter: complexType - listBulkMsgMarketingCampaignsResponse

Element	Type	Description
BulkMsgMarketingCampaignList	tns:bulkMsgMarketingCampaign	List of data structures described in Table 3–8.

Operation: listDynamicAdCampaigns

Fetches a list of current Dynamic advertising campaigns

Table 3–55 Input Message Parameter: complexType - listDynamicAdCampaigns

Element	Type	Description
CampaignSpaceId	xs:long	ID of the campaign space assigned by the system.
FirstResult	xs:int	The position in the list where fetching should start.
MaxResults	xs:int	The maximum number of results to return. Do not set this to zero.

listMsgServiceAdCampaignsResponse

Table 3–56 Output Message Parameter: complexType - listDynamicAdCampaignsResponse

Element	Type	Description
MsgServiceAdCampaignList	tns:msgServiceAdCampaign	List of data structures described in Table 3–9.

Operation: listMsgServiceAdCampaigns

Fetches a list of current MsgService advertising campaigns

Table 3–57 Input Message Parameter: complexType - listMsgServiceAdCampaigns

Element	Type	Description
CampaignSpaceId	xs:long	ID of the campaign space assigned by the system.
FirstResult	xs:int	Optional. The position in the list where fetching should start. Default value is 0.
MaxResults	xs:int	Optional. The maximum number of results to return. Default value is 0.

listMsgServiceAdCampaignsResponse

Table 3–58 Output Message Parameter: complexType - listMsgServiceAdCampaignsResponse

Element	Type	Description
MsgServiceAdCampaignList	tns:msgServiceAd Campaign	List of data structures described in Table 3–9.

The Campaign Space Approval Manager Web Service

The Campaign Space Approval Manager Web service approves or rejects campaign space.

Operations

The following describes the operations available in the Campaign Space Approval Manager Web service. Full details can be found in the WSDL.

- Operation: approveCampaignSpace
- Operation: rejectCampaignSpace

Operation: approveCampaignSpace

Approves a campaign space

Table 3–59 Input Message Parameter: complexType - approveCampaignSpace

Element	Type	Description
CampaignSpaceId	xs:long	ID of the campaign space assigned by the system

approveCampaignSpaceResponse

Table 3–60 Output Message Parameter: complexType - approveCampaignSpaceResponse

Element	Type	Description
CampaignSpaceState	tns:approvedEntityStateEnum	Current state of space: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED

Operation: rejectCampaignSpace

Rejects a campaign space

Table 3–61 Input Message Parameter: complexType - rejectCampaignSpace

Element	Type	Description
CampaignSpaceId	xs:long	ID of the campaign space assigned by the system.
Reason	xs:string	A brief description of why the space was rejected.

rejectCampaignSpaceResponse

Table 3–62 Output Message Parameter: complexType - rejectCampaignSpaceResponse

Element	Type	Description
CampaignSpaceState	tns:approvedEntityStateEnum	Current state of space: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED

The Campaign Manager Web Service

The Campaign Manager Web service creates, deletes, and lists ad content; creates, deletes, and lists Subscriber Lists and Categories; creates, deletes, and stops campaigns; and retrieves current campaigns.

Operations

The following describes the operations available in the Campaign Manager Web service. Full details can be found in the WSDL.

- Operation: createSmsAdContent
- Operation: createMmsAdContent
- Operation: createSimpleHtmlSnippetAdContent
- Operation: createTextAdContent
- Operation: createUrlAdContent
- Operation: createWapPushAdContent
- Operation: deleteAdContent
- Operation: listSmsAdContents
- Operation: listMmsAdContents
- Operation: listSimpleHtmlSnippetAdContents
- Operation: listTextAdContents
- Operation: listUrlAdContents
- Operation: listWapPushAdContents
- Operation:createSubscriberAddressList
- Operation:listSubscriberAddressLists
- Operation:deleteSubscriberAddressList
- Operation:createCategory
- Operation:listCategories
- Operation:createMsgServiceAdCampaign
- Operation:createDynamicAdCampaign
- Operation:createBulkMsgMarketingCampaignWithAddressList

- Operation:createBulkMsgMarketingCampaignWithSelectionCriteria
- Operation:stopCampaign
- Operation:deleteCampaign
- Operation:listBulkMsgMarketingCampaignsByAdv
- Operation:listMsgServiceAdCampaignsByAdv
- Operation:listDynamicCampaignsByAdv
- Operation:getBulkMsgMarketingCampaign
- Operation:getDynamicCampaign
- Operation:getMsgServiceAdCampaign
- Operation:listBulkMsgMarketingSpacesByAdv
- Operation:listDynamicAdSpaceByAdv
- Operation:listDynamicAdSpaceByAdv

Operation: createSmsAdContent

Creates ad content for the SMS Channel

Table 3–63 Input Message Parameter: complexType - createSmsAdContent

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating this content. Default value is managedOrganizationId.
Name	xs:string	The name of this content.
TextWithTokens	xs:string	The textual content of the ad, with tokens for personalization.
TextWithoutTokens	xs:string	Optional. The textual content of the ad.

createSmsAdContentResponse

Table 3–64 Output Message Parameter: complexType - createSmsAdContentResponse

Element	Type	Description
SmsAdContent	tns:SmsAdContent	See Table 3–12.

Operation: createMmsAdContent

Creates ad content for the MMS Channel

Table 3–65 Input Message Parameter: complexType - createMmsAdContent

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating this content. Default value is managedOrganizationId.
Name	xs:string	The name of this content

Table 3–65 (Cont.) Input Message Parameter: complexType - createMmsAdContent

Element	Type	Description
SubjectWithoutTokens	xs:string	Subject of the ad message
TextWithTokens	xs:string	Optional. The textual content of the ad, with tokens for personalization.
TextWithoutTokens	xs:string	Optional. The textual content of the ad.
MmsMediaList	ns: type and values	Optional. Specifies the type and value of media being used in the MMS message.

createMmsAdContentResponse

Table 3–66 Output Message Parameter: complexType - createMmsAdContentResponse

Element	Type	Description
MmsAdContent	tns:mmsAdContent	See Table 3–13.

Operation: createSimpleHtmlSnippetAdContent

Creates ad content for the Simple Html Snippet Channel

Table 3–67 Input Message Parameter: complexType - createSimpleHtmlSnippetAdContent

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating this content. Default value is managedOrganizationId.
Name	xs:string	The name of this content.
ActiveLink	xs:string	Specifies embedded clickable link URL in banner
BannerType	xs:string	The aspect ratio of the banner ad
BannerList	xs:string	List of banners uploaded by advertiser
BannerDefault	xs:int	Default bannerID to use from Banner list

createSimpleHtmlSnippetAdContentResponse

Table 3–68 Output Message Parameter: complexType - createSimpleHtmlSnippetAdContentResponse

Element	Type	Description
SimpleHtmlSnippetAdContent	tns:SimpleHtmlSnippetAdContent	See Table 3–14.

Operation: createTextAdContent

Creates ad content for the channels supporting text format

Table 3–69 Input Message Parameter: complexType - createTextAdContent

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating this content. Default value is managedOrganizationId.
Name	xs:string	The name of this content.
TextWithTokens	xs:string	The textual content of the ad, with tokens for personalization.
TextWithoutTokens	xs:string	Optional. The textual content of the ad.

createTextAdContentResponse

Table 3–70 Output Message Parameter: complexType - createTextAdContentResponse

Element	Type	Description
TextAdContent	tns:TextContent	See Table 3–15

Operation: createUrlAdContent

Creates ad content for the channels supporting Url format

Table 3–71 Input Message Parameter: complexType - createUrlAdContent

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating this content.] Default value is managedOrganizationId.
Name	xs:string	The name of this content.
URL	xs:string	Specifies embedded clickable link URL in banner

createUrlAdContentResponse

Table 3–72 Output Message Parameter: complexType - createUrlAdContentResponse

Element	Type	Description
UrlAdContent	tns:UrlAdContent	See Table 3–16.

Operation: createWapPushAdContent

Creates ad content for the WAP Push Channel

Table 3–73 Input Message Parameter: complexType - createWapPushAdContent

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating this content. Default value is managedOrganizationId.
Name	xs:string	The name of this content

Table 3–73 (Cont.) Input Message Parameter: complexType - createWapPushAdContent

Element	Type	Description
ActiveLink	xs:string	Specifies embedded clickable link URL in banner
TextWithTokens	xs:string	Optional. The textual content of the ad, with tokens for personalization.
TextWithoutTokens	xs:string	Optional. The textual content of the ad.

createWapPushAdContentResponse

Table 3–74 Output Message Parameter: complexType - createWapPushAdContentResponse

Element	Type	Description
WapPushAdContent	tns:wapPushAdContent	See Table 3–17

Operation: deleteAdContent

Deletes an existing ad content

Table 3–75 Input Message Parameter: complexType - deleteAdContent

Element	Type	Description
AdContentId	xs:long	ID of the ad content assigned by the system.

Operation: listSmsAdContents

Fetches a list of existing SMS Ad Contents

Table 3–76 Input Message Parameter: complexType - listSmsAdContents

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose contents are being fetched. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of Ad Contents where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listSmsAdContentsResponse

Table 3–77 Output Message Parameter: complexTy[e - listSmsAdContentsResponse

Element	Type	Description
Ad Content	tns:smsAdContent	List of data structures described in Table 3–12.

Operation: listMmsAdContents

Fetches a list of existing MMS Ad Contents

Table 3–78 Input Message Parameter: complexType - listMmsAdContents

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose contents are being fetched. Default value is managedOrganizationId.
FirstResult	xs:int	The position in the list of Ad Contents where fetching should start
MaxResult	xs:int	The maximum number of results to return. Do not set this to zero.

listMmsAdContentsResponse

Table 3–79 Output Message Parameter: complexType - listMmsAdContentsResponse

Element	Type	Description
AdContent	tns:mmsAdContent	List of data structures described in Table 3–13.

Operation: listSimpleHtmlSnippetAdContents

Fetches a list of existing Simple Html Snippet Ad Contents

Table 3–80 Input Message Parameter: complexType - listSimpleHtmlSnippetAdContents

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose contents are being fetched. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of Ad Contents where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listSimpleHtmlSnippetAdContentsResponse

Table 3–81 Output Message Parameter: complexType - listSimpleHtmlSnippetAdContentsResponse

Element	Type	Description
Ad Content	tns:simpleHtmlSnippetAdContent	List of data structures described in Table 3–14

Operation: listTextAdContents

Fetches a list of existing Text Ad Contents

Table 3–82 Input Message Parameter: complexType - listTextAdContents

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose contents are being fetched. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of Ad Contents where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listTextAdContentsResponse

Table 3–83 Output Message Parameter: complexType - listTextAdContentsResponse

Element	Type	Description
Ad Content	tns:textAdContent	List of data structures described in Table 3–15.

Operation: listUrlAdContents

Fetches a list of existing Url Ad Contents

Table 3–84 Input Message Parameter: complexType - listUrlAdContents

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose contents are being fetched. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of Ad Contents where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listUrlAdContentsResponse

Table 3–85 Output Message Parameter: complexType - listUrlAdContentsResponse

Element	Type	Description
Ad Content	tns:urlAdContent	List of data structures described in Table 3–16

Operation: listWapPushAdContents

Fetches a list of existing WAP Push Ad Contents

Table 3–86 Input Message Parameter: complexType - listWapPushAdContents

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose contents are being fetched. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of Ad Contents where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listWapPushAdContentsResponse

Table 3–87 Output Message Parameter: complexType - listWapPushAdContentsResponse

Element	Type	Description
Ad Content	tns:wapPushAdContent	List of data structures described in Table 3–17

Operation:createSubscriberAddressList

Uploads a Subscriber Address List for marketing campaigns

Table 3–88 Input Message Parameter: complexType - createSubscriberAddressList

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser who is providing this list for its use. Default value is managedOrganizationId.
Name	xs:string	Name for the list
SubscriberList	xs:base64Binary	The list, in.csv format or EOL delimited

createSubscriberAddressListResponse

Table 3–89 Output Message Parameter: complexType - createSubscriberAddressListResponse

Element	Type	Description
SubscriberAddressList	tns:subscriberAddressList	List of data structures described in Table 3–18.

Operation:listSubscriberAddressLists

Fetches a list of Subscriber Address Lists

Table 3–90 Input Message Parameter: complexType - listSubscriberAddressLists

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose lists are being fetched. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of Subscriber Lists where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listSubscriberAddressListsResponse

Table 3–91 Output Message Parameter: complexType - listSubscriberAddressListsResponse

Element	Type	Description
SubscriberAddressLists	tns:subscriberAddressList	List of data structures described in Table 3–18.

Operation:deleteSubscriberAddressList

Deletes a Subscriber Address List

Table 3–92 Input Message Parameter: complexType - deleteSubscriberAddressList

Element	Type	Description
SubscriberListId	xs:long	ID of the list to be deleted

Operation:createCategory

Creates a new subscriber category

Table 3–93 Input Message Parameter: complexType - createCategory

Element	Type	Description
CategoryTypeName	xs:string	Always true
isStatisticsCategory	xs:boolean	
CategoryValueType	ns	Range or enumeration
CategoryValue	ns	Operator attributes to describe category range

createCategoryResponse

Table 3–94 Output Message Parameter: complexType - createCategoryResponse

Element	Type	Description
Category	tns:subscriberCategory	Data structures described in Table 3–11

Operation:listCategories

Fetches a list of the Subscriber Categories currently being offered

Table 3–95 Input Message Parameter: complexType - listCategories

Element	Type	Description
isStatisticsCategory	xs:boolean	Always true

listCategoriesResponse

Table 3–96 Output Message Parameter: complexType - listCategoriesResponse

Element	Type	Description
CategoryList	tns:subscriberCategory	List of data structures described in Table 3–11

Operation:createMsgServiceAdCampaign

Creates an advertising campaign

Table 3–97 Input Message Parameter: complexType - createMsgServiceAdCampaign

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating the campaign. Default value is managedOrganizationId.
Name	xs:string	Name for the campaign
MsgServiceAdSpaceId	xs:long	ID of the Ad space on which this campaign is based
AdContentId	xs:long	ID of the content which is to be used
Price	xs:float	Optional. The bid price for this Ad space; the currency unit is set by the System Administrator; range is 0 to 9999.999. Defualt value is 0.0.
MessageLimit	xs:int	Optional. The maximum number of messages for this campaign. Default value is 0.
Modifiable	xs:boolean	Optional. Whether this campaign can be modified. Default value is False.
MaxMsgsPerSubscriber	xs:int	Optional. The maximum number of messages an individual subscriber can receive. Default value is 0.

Table 3–97 (Cont.) Input Message Parameter: complexType -

Element	Type	Description
campaignFeatures	ns	Optional. List of optional/premium features to include in the campaign including: ■ demography ■ geo(location) ■ personalization ■ response management ■ subscriber capping

createMsgServiceAdCampaignResponse

Table 3–98 Output Message Parameter: complexType - createMsgServiceAdCampaignResponse

Element	Type	Description
MsgServiceAdCampaign	tns:msgServiceAdCampaign	See Table 3–9

Operation:createDynamicAdCampaign

Creates a dynamic advertising campaign

Table 3–99 Input Message Parameter: complexType - createDynamicAdCampaign

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating the campaign. Default value is managedOrganizationId.
Name	xs:string	Name for the campaign
AdContentId	xs:long	ID of the content which is to be used
Price	xs:float	Optional. The bid price for this Ad space; the currency unit is set by the System Administrator; range is 0 to 9999.999. Default value is 0.0.
MessageLimit	xs:int	Optional. The maximum number of messages for this campaign. Default value is 0.
Modifiable	xs:boolean	Optional. Whether this campaign can be modified. Default value is false.

Table 3–99 (Cont.) Input Message Parameter: complexType -

Element	Type	Description
ManuallyStopped	xs:boolean	Optional. Whether the campaign can be manually stopped. Default value is false.
MaxMsgsPerSubscriber	xs:int	Optional. The maximum number of messages an individual subscriber can receive. Default value is 0.
StartTime	xs:dateTime	Date and time at which this campaign starts
EndTime	xs:dateTime	Date and time at which this campaign ends
DynamicCampaignCategories	ns: name and value	Optional. Categories to include in dynamic campaign.
campaignFeature	tns:campaignFeatures	Optional. List of optional/premium features to include in the campaign including: ■ demography ■ geo(location) ■ personalization ■ response management ■ subscriber capping

createDynamicAdCampaignResponse

Table 3–100 Output Message Parameter: complexType - createDynamicCampaignResponse

Element	Type	Description
MsgServiceAdCampaign	tns:msgServiceAdCampaign	See Table 3–9.

Operation:createBulkMsgMarketingCampaignWithAddressList

Creates a marketing campaign using an address list

Table 3–101 Input Message Parameter: complexType - createBulkMsgMarketingCampaignWithAddressList

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating the campaign. Default value is managedOrganizationId.
Name	xs:string	Name for the campaign
CampaignSpaceId	xs:long	ID of the marketing space on which this campaign is based
AdContentId	xs:long	ID of the content which is to be used

Table 3–101 (Cont.) Input Message Parameter: complexType - createBulkMsgMarketingCampaignWithAddressList

Element	Type	Description
Price	xs:float	Optional. The price per ad to use this space; the currency unit is set by the System Administrator; range is 0 to 9999.999. Default value is 0.0.
MessageLimit	xs:int	Optional. The maximum number of messages for this campaign. Default value is 0.
Modifiable	xs:boolean	Optional. Whether this campaign can be modified. Default value is 0.
SubscriberListId	xs:long	ID of the uploaded subscriber address list to be used
StartTime	xs:dateTime	Date and time at which this campaign starts
EndTime	xs:dateTime	Date and time at which this campaign ends
campaignFeature	tns:campaignFeatures	Optional. List of optional/premium features to include in the campaign including: ■ demography ■ geo(location) ■ personalization ■ response management ■ subscriber capping

createBulkMsgMarketingCampaignWithAddressListResponse

Table 3–102 Output Message Parameter: complexType - createBulkMsgMarketingCampaingWithAddressListResponse

Element	Type	Description
BulkMsgMarketingCampaign	tns:BulkMsgMarketingCampaign	See Table 3–8.

Operation:createBulkMsgMarketingCampaignWithSelectionCriteria

Creates a marketing campaign using subscriber categories

Table 3–103 Input Message Parameter: complexType - createBulkMsgMarketingCampaignWithSelectionCriteria

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser creating the campaign. Default value is managedOrganizationId.

Table 3–103 (Cont.) Input Message Parameter: complexType - createBulkMsgMarketingCampaignWithSelectionCriteria

Element	Type	Description
Name	xs:string	Name for the campaign
CampaignSpaceId	xs:long	ID of the marketing space on which this campaign is based
AdContentId	xs:long	ID of the content which is to be used
Price	xs:float	Optional. The price per ad to use this space; the currency unit is set by the System Administrator; range is 0 to 9999.999. Default value is 0.0.
MessageLimit	xs:int	Optional. The maximum number of messages for this campaign. Default value is 0.
Modifiable	xs:boolean	Optional. Whether this campaign can be modified. Default value is False.
SubscriberCategories	tns:subscriberCategories	Optional. See Table 3–11
StartTime	xs:dateTime	Date and time at which this campaign starts
EndTime	xs:dateTime	Date and time at which this campaign ends
campaignFeature	tns:campaignFeatures	Optional. List of optional/premium features to include in the campaign including: ■ demography ■ geo(location) ■ personalization ■ response management ■ subscriber capping

createBulkMsgMarketingCampaignWithSelectionCriteriaResponse

Table 3–104 Output Message Parameter: complexType - createBulkMsgMarketingCampaignWithSelectionCriteriaResponse

Element	Type	Description
BulkMsgMarketingCampaign	tns:BulkMsgMarketingCampaign	See Table 3–8

Operation:stopCampaign

Stops a campaign

Table 3–105 Input Message Parameter: complexType - stopCampaign

Element	Type	Description
Id	xs:long	ID of the campaign to be stopped
Reason	xs:string	A brief description of why the campaign is being stopped

Operation:deleteCampaign

Deletes a campaign

Table 3–106 Input Message Parameter: complexType - deleteCampaign

Element	Type	Description
Id	xs:long	ID of the campaign to be deleted

Operation:listBulkMsgMarketingCampaignsByAdv

Fetches BulkMsg marketing campaigns by the advertiser that created them

Table 3–107 Input Message Parameter: complexType - listBulkMsgMarketingCampaignsByAdv

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser that created the campaigns. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of campaigns where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.
ApprovedState	tns:approvedEntityStateEnum	State of the campaign: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED [optional - can be null]

listBulkMsgMarketingCampaignsByAdvResponse

Table 3–108 Output Message Parameter: complexType - listBulkMsgMarketingCampaignsByAdvResponse

Element	Type	Description
BulkMsgMarketingCampaignList	tns:bulkMsgMarketingAdCampaign	List of data structures described in Table 3–8

Operation:listMsgServiceAdCampaignsByAdv

Fetches MsgService advertising campaigns by the advertiser that created them

Table 3–109 Input Message Parameter: complexType - listMsgServiceAdCampaignsByAdv

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser that created the campaigns. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of campaigns where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.
ApprovedState	tns:approvedEntityStateEnum	Optional. State of the campaign: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED Default value is all non-deleted states.

listMsgServiceAdCampaignsByAdvResponse

Table 3–110 Output Message Parameter: complexType - listMsgServiceAdCampaignsByAdvResponse

Element	Type	Description
MsgServiceAdCampaignList	tns:msgServiceAdCampaign	List of data structures described in Table 3–9

Operation:listDynamicCampaignsByAdv

Fetches dynamic campaigns by the advertiser that created them

Table 3–111 Input Message Parameter: complexType - listDynamicAdCampaignsByAdv

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser that created the campaigns. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of campaigns where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.
ApprovedState	tns:approvedEntityStateEnum	Optional. State of the campaign: one of the following ■ CREATED ■ ACTIVE ■ STOPPED ■ REJECTED ■ DELETED Default value is all non-deleted states.

listDynamicAdCampaignsByAdvResponse

Table 3–112 Output Message Parameter: complexType - listDynamicAdCampaignsByAdvResponse

Element	Type	Description
DynamicAdCampaignList	tns:msgServiceAdCampaign	List of data structures described in Table 3–9

Operation:getBulkMsgMarketingCampaign

Fetches a specific BulkMsg marketing campaign

Table 3–113 Input Message Parameter: complexType - getBulkMsgMarketingCampaign

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser that created the campaign. Default value is managedOrganizationId.
Id	xs:long	ID of the campaign assigned by the system.

getBulkMsgMarketingCampaignResponse

Table 3–114 Output Message Parameter: complexType - getBulkMsgMarketingCampaignResponse

Element	Type	Description
return	tns:bulkMsgMarketingCampaign	See Table 3–8

Operation:getDynamicCampaign

Fetches a specific dynamic campaign

Table 3–115 Input Message Parameter: complexType - getDynamicCampaign

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser that created the campaign. Default value is managedOrganizationId.
Id	xs:long	ID of the campaign assigned by the system.

getDynamicCampaignResponse

Table 3–116 Output Message Parameter: complexType - getDynamicCampaignResponse

Element	Type	Description
return	tns:bulkDynamicCampaign	See Table 3–10.

Operation:getMsgServiceAdCampaign

Fetches a specific Msg Service ad campaign

Table 3–117 Input Message Parameter: complexType - getMsgServiceAdCampaign

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser that created the campaign. Default value is managedOrganizationId.
Id	xs:long	ID of the campaign assigned by the system

getMsgServiceAdCampaignResponse

Table 3–118 Output Message Parameter: complexType - getMsgServiceAdCampaignResponse

Element	Type	Description
return	tns:MsgServiceAdCampaign	See Table 3–9

Operation:listBulkMsgMarketingSpacesByAdv

Fetches Bulk Messaging marketing space by the advertiser that used them to create marketing campaigns

Table 3–119 Input Message Parameter: complexType - listBulkMsgMarketingSpacesByAdv

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose use of BulkMsg marketing space is being queried. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of this advertiser's spaces where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listBulkMsgMarketingSpacesByAdvResponse

Table 3–120 Output Message Parameter: complexType - listBulkMsgMarketingSpacesByAdvResponse

Element	Type	Description
BulkMsgMarketingSpaceLis	tns:bulkMsgMarketingSpace	List of data structures described in Table 3–5

Operation:listMsgServiceAdSpaceByAdv

Fetches MsgService Ad space by the advertiser that used them to create advertising campaigns

Table 3–121 Input Message Parameter: complexType - listMsgServiceAdSpaceByAdv

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose use of MsgService Ad space is being queried. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of this advertiser's spaces where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listMsgServiceAdSpaceByAdvResponse

Table 3–122 Output Message Parameter: complexType - listMsgServiceAdSpaceByAdvResponse

Element	Type	Description
MsgServiceAdSpaceLis	tns:msgServiceAdSpace	List of data structures described in Table 3–7

Operation:listDynamicAdSpaceByAdv

Fetches DynamicAd space by the advertiser that used them to create advertising campaigns

Table 3–123 Input Message Parameter: complexType - listDynamicAdSpaceByAdv

Element	Type	Description
AdvertiserId	xs:long	Optional. ID of the advertiser whose use ofDynamic Ad space is being queried. Default value is managedOrganizationId.
FirstResult	xs:int	Optional. The position in the list of this advertiser's spaces where fetching should start. Default value is 0.
MaxResult	xs:int	Optional. The maximum number of results to return. Default value is 0.

listDynamicAdSpaceByAdvResponse

Table 3–124 Output Message Parameter: complexType - listDynamicAdSpaceByAdvResponse

Element	Type	Description
MsgServiceAdSpaceLis	tns:msgServiceAdSpace	List of data structures described in Table 3–7

4

Using Web Services to Query Campaign Statistics

This chapter describes the operations available for querying campaign statistics using the Statistics Manager web service. Oracle Communications Marketing and Advertising collects campaign data for the following ad metrics:

- Ads Failed
- Ad Impressions
- Ads Served
- Ad Click Through Responses

Authorized Users

Only statistics available to authorized users can be queried via the web service. For example, an advertiser will only be able to retrieve statistics for campaigns that are created and executed by him/herself.

Errors

Errors are returned in a SOAP Message that includes a human readable `faultstring` element that describes the error.

Return Data Structures

The following data structures are returned in the Statistics Manager web service.

ocmaBean

Table 4–1 contains the ocmaBean data structure. This data structure describes the entity for which statistics are returned by the Statistics Manager Web service.

Table 4–1 The ocmaBean Data Structure

Element Name	Type	Description
id	xs:long	System assigned ID of entity for which statistics are being queried (e.g. campaign, connection configuration, etc.)
name	xs:string	Name of the entity for which statistics are being queried

statistics

Table 4–2 contains the ocmaBean data structure. This data structure describes the entity for which statistics are returned by the Statistics Manager Web service.

Table 4–2 The statistics Data Structure

Element Name	Type	Description
abstractStatistics	xs:extension base	See Table 4–3

abstractStatistics

Table 4–3 contains the abstractStatistics data structure. This data structure describes the statistical ad elements returned by the Statistics Manager Web service.

Table 4–3 The abstractStatistics Data Structure

Element Name	Type	Description
adFailed	xs:lint	Number of failed ads for the entity for which statistics are being queried
adImpression	xs:int	Number of ads actually delivered for the entity for which statistics are being queried
adServed	xs:int	Number of ads served for the entity for which statistics are being queried
clickThrough	xs:int	Number of click through responses generated from the ads served

categoryStatistics

Table 4–4 contains the categoryStatistics data structure. This data structure describes the statistical category elements returned by the Statistics Manager Web service getCategoryStatistics operation.

Table 4–4 The categoryStatistics Data Structure

Element Name	Type	Description
abstractStatistics	xs:extension base	See Table 4–3
typeName	xs:string	Category name
value	xs:string	Category value

timeBasedStatistics

Table 4–5 contains the timeBasedStatistics data structure. This data structure describes the statistical elements returned by the Statistics Manager Web service getTimeBasedStatistics operation.

Table 4–5 The timeBasedStatistics Data Structure

Element Name	Type	Description
abstractStatistics	xs:extension base	See Table 4–3
recordTime	xs:dateTime	Start time of the campaign being queried

Table 4–5 (Cont.) The timeBasedStatistics Data Structure

Element Name	Type	Description
resolution	tns:resolution	Time measurement: ■ Hourly ■ Daily ■ Monthly

The Statistics Manager Web Service

The Statistics Manager web service can be used to retrieve statistical information about campaigns based on various entity filters (e.g. Connection Configuration, Ad Space, etc.). A number of operations are provided for querying Marketing and Advertising campaign statistics.

Operations

The following describes the operations available in the Statistics Manager web service. Full details can be found in the WSDL and xsd files.

- Operation: getCategoryStatistics
- Operation: getStatisticsByConnectionConfig
- Operation: getStatisticsByID
- Operation: getStatisticsPerAdSpace
- Operation: getTimeBasedStatistics

Operation: getCategoryStatistics

Retrieves campaign statistical information for the specified category. Table 4–6 contains the input message parameters for the getCategoryStatistics operation

Table 4–6 Input Message Parameter: complexType - getCategoryStatistics

Element	Type	Description
Id	xs:long	Category ID assigned by the system
Category	xs:string	Name of Category

getCategoryStatisticsResponse

Table 4–7 contains the output parameter in the getCategoryStatisticsResponse message.

Table 4–7 Output Message Parameter: complexType - getCategoryStatisticsResponse

Element	Type	Description
StatisticsList	tns:categoryStatistics	List of abstract statistics complex elements for the queried Category

Operation: getStatisticsByConnectionConfig

Retrieves statistical information about campaigns based on Connection Configuration. Table 4–8 contains the input message parameter for the getStatisticsByConnectionConfig operation

Table 4–8 Input Message Parameter: complexType - getStatisticsByConnectionConfig

Element	Type	Description
Id	xs:long	Connection Configuration ID assigned by the system

getStatisticsByConnectionConfigResponse

Table 4–9 contains the output parameter in the getStatisticsByConnectionConfigResponse message.

Table 4–9 Output Message Parameter: complexType - getStatisticsByConnectionConfigResponse

Element	Type	Description
Statistics	tns:statistics	Abstract statistics complex element for the queried Connection Configuration

Operation: getStatisticsByID

Retrieves statistical information about campaigns based on campaign ID. Table 4–10 contains the input message parameter for the getStatisticsById operation

Table 4–10 Input Message Parameter: complexType - getStatisticsById

Element	Type	Description
Id	xs:long	Campaign ID assigned by the system

getStatisticsByIdResponse

Table 4–11contains the output parameter in the getStatisticsByIdResponse message.

Table 4–11 Output Message Parameter: complexType - getStatisticsByIdResponse

Element	Type	Description
Statistics	tns:statistics	Abstract statistics complex element for the queried Campaign

Operation: getStatisticsPerAdSpace

Retrieves statistical information about campaigns per Ad Space. Table 4–12 contains the input message parameter for the getStatisticsPerAdSpace operation

Table 4–12 Input Message Parameter: complexType - getStatisticsPerAdSpace

Element	Type	Description
Id	xs:long	Ad Space ID assigned by the system

getStatisticsPerAdSpaceResponse

Table 4–13 contains the output parameter in the getStatisticsPerAdSpaceResponse message.

Table 4–13 Output Message Parameter: complexType - getStatisticsPerAdSpaceResponse

Element	Type	Description
StatisticsList	tns:categoryStatistics	List of abstract statistics complex elements for the queried Ad Space

Operation: getTimeBasedStatistics

Retrieves statistical information about campaigns based on time parameters. Table 4–14 contains the input message parameters for the getTimeBasedStatistics operation

Table 4–14 Input Message Parameter: complexType - getTimeBasedStatistics

Element	Type	Description
Id	xs:long	Campaign ID assigned by the system
StartTime	xs:dateTime	Start time boundary for range of query
EndTime	xs:dateTime	End time boundary for range of query
Resolution	tns:resolution	Time measurement: ■ Hourly ■ Daily ■ Monthly

getTimeBasedStatisticsResponse

Table 4–15 contains the output parameter in the in the getTimeBasedStatisticsResponse message.

Table 4–15 Output Message Parameter: complexType - getStatisticsPerAdSpaceResponse

Element	Type	Description
TimeBasedStatisticsList	tns:timeBasedStatistics	List of abstract statistics complex elements for the queried time duration

5

Using Oracle Communications Marketing and Advertising as an Ad Server

Oracle Communications Marketing and Advertising can serve ad content to external applications when acting as an Open Mobile Alliance (OMA) Ad Server over web services. MobAd-compliant content stored in Marketing and Advertising can be retrieved and displayed through various channels in OMA Mobile Advertising v1.0 compliant format. For more information on the OMA and Mob Ad see:

http://www.openmobilealliance.org/Technical/release_program/mobad_v1_0.aspx

Overview

There are many types of applications that integrate with ad servers for targeted advertising. Examples of such applications can include ad-aware content applications or downloadable applications or games.

Marketing and Advertising can be configured to accept MobAd ad requests, return ad content and manage campaigns associated with external entities. Statistical information from external applications/entities about served OMA ads can also be collected by Marketing and Advertising for reporting purposes.

The Ad Server functionality provided by Marketing and Advertising is web service based. To setup an integration of an external application with the Ad Server Web Service you should review the following Web Service information and sample campaign configuration steps:

1. Understanding the Ad Server Web Service
2. Marketing and Advertising Ad Server Processes

Understanding the Ad Server Web Service

The Marketing and Advertising Ad Server API is web service based utilizing SOAP transmission over HTTP 1.1 protocol. The WSDL provided by the Ad Server web service is based on the OMA MobAd specification XSD/draft specfication. As the specification is subject to change be sure to verify compatibility between OMA MobAd version and the Marketing and Advertising release version.

Errors

Errors are returned in a SOAP Message that includes a human readable `faultstring` element that describes the error. Table 5–1 contains a list of fault codes associated with the Ad Server web service.

Table 5–1 Ad Server Error Codes

Error Code	Description
OCMA-1022	A validation exception occurred for field *field*
OCMA-1049	Failed to load DAO object id=*objectID*, class=*classID*
OCMA-1053	The user *user* is not allowed to execute *operation* operation on *entity*
OCMA-1058	Failed to load image
OCMA-1060	Invalid parameter *parameter* for *entity*
OCMA-1070	*Attribute* not supported
OCMA-1077	Banner image type error

Data Structures Used in the Web Service

The following data structures appear in the web service.

SPAppAdRequestInfo

This elements in Table 5–2 represent the actual request for an ad. Although the xsd allows an unbounded list of these elements to retrieve multiple ads per a single request, the current implementation allows only one of these elements per request.

Table 5–2 SPAppAdRequestInfo Complex Type

Parameter	XML Type	Data Type	Cardinality	Comments
AdUnit	Attribute	xs:string	0...1	Not supported
ProvideAdContent	Attribute	xs:boolean	0..1	If FALSE, the type of ad returned is URL Content. TRUE for other ad types It is always supported to omit this parameter (provide NULL)
AdValidity	Attribute	xs:string	0..1	Not supported
ContextuaData	Sub-element	mobad:ContextualDataType	0..N	Not supported

SPAppProvidedAd

This elements in Table 5–3 represent the entire returned ad.

Table 5–3 SPAppProvidedAd Complex Type

Parameter	XML Type	Data Type	Cardinality	Comments
AdContentData	Sub-element	See Table 5–4	1	The content of the returned ad
AdUsage	Sub-element	See Table 5–5	0..1	Contains only the FrequencyCap attribute set to "1"
AdID	Attribute	xs:string	0..1	"ocma" + campaignID (within the system)
Outdated-AdID	Attribute	xs:string	0..N	Not supported. Omitted in response

Table 5–3 (Cont.) SPAppProvidedAd Complex Type

Parameter	XML Type	Data Type	Cardinality	Comments
ContextualDataID	Attribute	mobad:ContextualDataID	0..1	Not supported. Omitted in response
AdType	Attribute	xs:integer	0..1	Not supported. Omitted in response
AdUnit	Attribute	mobad:AdUnit	0..1	Not supported. Omitted in response

AdContentData

This elements in Table 5–4 represent the content in the returned ad.

Table 5–4 AdContentData Complex Type

Parameter	XML Type	Data Type	Cardinality	Comments
MIME-Type	Attribute	xs:string	1	Indicates the mime-media type of the content. For URL Content, an empty string is returned
MIME-Version	Attribute	xs:string	0..1	Not supported. Omitted in response
AdWidth	Attribute	Ixs:integer	0..1	Not supported. Omitted in response
AdHeight	Attribute	xs:integer	0..1	Not supported. Omitted in response
AdDuration	Attribute	xs:integer	0..1	Not supported. Omitted in response
AdContentSize	Attribute	xs:integer	1	Indicates the size in bytes of the Ad payload. For URL Content, 0 is returned
TextEncoding	Attribute	xs:string	1, if used	Indicates the encoding scheme for text, such as Unicode or UTF-8. Used in cases where the MIME-Type is of type text
AdContentPayload	Sub-element	xs:hexBinary	1, if used	Contains the actual Ad content. Used in cases where content is not URL Content
AdLocationURI	Sub-element	xs:anyURI	1, if used	Contains the URI. Used in cases where content is URL Content
InteractivityAction	Sub-element	mobad:InteractivityActionType	0..N	Not supported. Omitted in response

AdUsage

This elements in Table 5–5 represent the number of times the ad is placed.

Table 5–5 AdUsage Complex Type

Parameter	XML Type	Data Type	Cardinality	Comments
DisplayTime	Attribute	xs:string	0..1	Not supported. Omitted in response

Table 5–5 (Cont.) AdUsage Complex Type

Parameter	XML Type	Data Type	Cardinality	Comments
FrequencyCap	Attribute	xs:integer	0..1	Always **1**.
PreserveAdFormat	Attribute	xs:boolean	0..1	Not supported. Omitted in response
AdValidity	Attribute	xs:string	0..1	Not supported. Omitted in response

AdMetricsData

This elements in Table 5–6 represent the campaign metrics for an ad.

Table 5–6 AdMetricsData Complex Type

Parameter	XML Type	Data Type	Cardinality	Comments
AdID	Attribute	mobad:AdID	1	Must be identical to AdID in Table 5–2. **ocma** + campaignID
MetricType	Attribute	xs:string	0..1	Must be either **a** or **impression**
MetricValue	Attribute	xs:string	0..1	The number of messages covered by this report. If omitted, assumed to be 1.
ViewDuration	Attribute	xs:integer	0..1	Not supported. Omitted in response
DisplayDateTime	Attribute	xs:dateTime	0..1	A timestamp on the statistic. For example, 2008-01-09+01:00

SpApp Ad Web Service

The SpApp Ad web service is made up of two operations. The first operation, SpAppAdRequest, requests an ad. The second operation, SPAppMetricsReport, allows the external entity to report that it has placed the ad, which increments the Ad Impression counter for statistical purposes.

Operations Provided by the Web Service

The two operations provided by the web service follow.

Operation:SPAppAdRequest

This operation represents an external application request for Marketing and Advertising hosted ad content data.

Table 5–7 describes the SPAppAdRequest operation.

Table 5–7 SPAppAdRequest Input Message

Parameter Name	XML Type	Data Type	Cardinality	Comments
SPAppAdRequestInfo	Element	See Table 5–2	1	Only a single ad can be requested at a time
Version	Attribute	mobad:Version	0..1	Based on the Mob-Ad specification
SPAppID	Attribute	mobad:SPAppID	1	Identical to the SPAppID used to create the OMA Server Connection Configuration

Table 5–7 (Cont.) SPAppAdRequest Input Message

Parameter Name	XML Type	Data Type	Cardinality	Comments
RequestUrgency	Attribute	mobad:RequestUrgency	0..1	Not supported
PrincipalID	Attribute	xs:string	0..1	Principal ID of the application making request

Table 5–8 table describes the SPAppAdResponse operation.

Table 5–8 SPAppAdResponse Output Message

Parameter Name	XML Type	Data Type	Cardinality	Comments
SPAppProvidedAd	Element	See Table 5–3	1	Only a single ad is provided at a time

Operation: SPAppMetricsReport

This operation allows an external application/entity to update Marketing and Advertising with campaign statistics, such as number of impressions, for the application/entity.

Table 5–9 describes the SPAppMetricsReport operation.

Table 5–9 SPAppMetricsReport Input Message

Parameter Name	XML Type	Data Type	Cardinality	Comments
SPAppID	Attribute	xs:string	1	Identical to the SPAppID used to create the OMA Server Connection Configuration
AdMetricsData	Element	See Table 5–6	1..N	The impression to be reported

Marketing and Advertising Ad Server Processes

OMA Ad Server campaigns in Marketing and Advertising follow the same general process flow as native ad campaigns. An external application requesting Marketing and Advertising ad content must have an associated Connection Configuration. A Network Provider creates this Connection Configuration which contains an application ID (SpAppID). External application operators act as Marketing and Advertising Ad Space Owners creating marketable Ad Spaces to sell to Advertisers.

In a production environment an external application submits a web service Mob-Ad request containing the identifying SpAppID to Marketing and Advertising. Marketing and Advertising uses this SpAppID to determine the proper ad content for the Ad Space purchased by advertisers. The retrieved ad content is returned in the web service response to the external application.

The following steps outline configuring Marketing and Advertising to process an external application ad request. For additional information on the Marketing and Advertising entities mentioned see "About Marketing and Advertising Entities". Additional information can also be found in the Online Help for the specified users.

1. Understanding OMA Ad Server Concepts

2. Creating an OMA Ad Server Connection Configuration
3. Bidding and Purchasing OMA Ad Space
4. Responding to External Application Requests Ad
5. Reporting Ad Server Statistics

Understanding OMA Ad Server Concepts

Marketing and Advertising relies on the external OMA application requesting ad content to identify itself using an application identifier in order to determine the correct connection configuration assigned to the application. The connection configuration inherently contains channel information indicating what type of ad content can be used to fulfill the request. Marketing and Advertising supports text, URL and simple HTML snippet ad content for Mob-Ad requests.

OMA SP Application Identifier

In the context of the OMA specification, the SP Application is the external entity that is requesting ads from the Ad Server. For tracking purposes, an ID is assigned to this entity using this parameter. In effect, this identifier indicates a particular external space in which an ad might appear. It could be part of an existing HTML page; it could be a game engine on a mobile device, etc.

Channel Types

In Marketing and Advertising connection configurations, Channel Type refers to the type of traffic that will carry the ad message - SMS, MMS, or WAP Push. In an OMA connection configuration, only the ad itself is being provided, so the Channel Type simply defines the type of ad made available. There are three possible types:

- Text: A text-only ad
- URL: A URL to content hosted elsewhere on the Web: for example, an IP TV ad streaming from another site
- Simple HTML Snippet: An ad that can contain text, a URL, and an image that is fetched from a servlet running within the system. The snippet is provided as hex code. See Example 5–1 and Example 5–2. The URL to the image is retrieved from the **OCMA Web URL** field in the **General** tab of the OCMA Administration console.

Example 5–1 Example HTML Snippet

```
<a href="http://SomeWebSite.com/">
 Buy This!
 <img
src="http://localhost:8001/ocma/SnippetContent?segId=2&cId=126&rId=769337682332825
0879"/>
</a>
```

Example 5–2 Example HTML Snippet as Hex Code

```
3c 61 20 68 72 65 66 3d 22 68 74 74 70 3a 2f 2f 53 6f 6d 65 57 65 62 53 69 74 65
2e 63 6f 6d 2f 22 3e 0d 0a 20 42 75 79 20 54 68 69 73 21 20 20 0d 0a 3c 69 6d 67
20 73 72 63 3d 22 68 74 74 70 3a 2f 2f 6c 6f 63 61 6c 68 6f 73 74 3a 38 30 30 31
2f 6f 63 6d 61 2f 53 6e 69 70 70 65 74 43 6f 6e 74 65 6e 74 3f 73 65 67 49 64 3d
32 26 63 49 64 3d 31 32 36 26 72 49 64 3d 37 36 39 33 33 37 36 38 32 33 33 32 38
32 35 30 38 37 39 22 2f 3e 0d 0a 20 3c 2f 61 3e 20 20 20 20 20
```

Just as with other connection configuration types, the OMA Server connection configuration must be associated with an ad space owner provisioned in the Marketing and Advertising system.

Creating an OMA Ad Server Connection Configuration

The Marketing and Advertising network provider user must create an OMA Server connection configuration. To create an OMA Server connection configuration:

1. As a network provider user click the **Connection Configurations** menu item
2. Click the **Create Connection Configuration** button
3. Click the **OMA Server Connection Configuration** button
4. Enter a name in the **Name** field
5. Enter an application ID in the **OMA SP Application Identifier** field
6. Leave a value of **1** in the **Maximum return ads number in one request** field

> **Note:** Only a value of **1** is supported in this release of Marketing and Advertising. This value corresponds to a single ad or URI return for each external application ad request.

7. Select the **Channel Type** to be used for the external application
8. Assign an ad space owner using the search functionality
9. Enter a value in the **Max Ad Message Size** field if needed
10. Select if **Delivery Notification** should be enabled
11. Click the **search** button to enable enhanced features if needed
12. Click **Save**

Create an Ad Space Using the OMA Server Connection Configuration

A Marketing and Advertising Ad Space must be associated with the newly created OMA Ad Server connection configuration. To create an Ad Space for use with an external application:

1. As an Ad Space Owner user click the **Campaign Spaces & Campaigns** menu item
2. Click the **Create New Campaign Space** button
3. Click the **Create Ad Space** button
4. Enter a name in the **Name** field
5. Enter a description in the **description** field
6. Click the **Search** button for the **Connection** field
7. Select the **OMA Server Connection Configuration** menu item to search for available OMA Server Connection Configurations
8. Select the desired Connection Configuration
9. Click **OK**
10. Select if **Dynamic Matching Category** options if needed

11. Click the **Calendar** button for each of the following fields and enter the required Bid and Campaign start and end times:
    - **Bid Start Time**
    - **Bid End Time**
    - **Campaign Start Time**
    - **Campaign End Time**
12. Select the checkbox labeled **Yes** if the Ad Space is Fixed Price
13. Enter a currency value for the **Minimum Bid Price**
14. Click the **Browse** button to select an icon for the Ad Space if needed
15. Click **Save**

The Ad Space created will become active and available for Advertiser bidding or fixed price purchase at the specified time. At the conclusion of the auction the campaign will start at the specified Campaign Start Time.

Bidding and Purchasing OMA Ad Space

Once the bidding period of an OMA Ad Server campaign is in effect advertiser users in Marketing and Advertising can bid or purchase Ad Space in the Marketplace. Ad Spaces eligible for bidding are presented in the advertiser web interface.

To use an OMA Server Ad Space, an advertiser must create Ad Content for the Channel Types that OMA Server Connection Configurations support. For information on creating Ad Content see "Creating New Ad Content" in the Advertiser User's Online Help.

To enter a new bid on an OMA Ad Server Ad Space:

1. As an Advertiser user click the **Marketplace** menu item to display Ad Spaces open for bidding
2. Select the Ad Space to bid on
3. Click the **Enter New Bid** button
4. Enter a name in the **Name** field
5. Enter a currency value in the **Campaign bid price** field
6. Enter a value for the **Maximum Number of Messages** field to calculate the **Total Price** of the bid
7. Select the desired ad content for use with the campaign

> **Note:** The Marketing and Advertising OMA Ad Server implementation supports text, URL and simple html snippet ad content as specified in the connection configuration.
>
> Ad Content is created and managed in the Advertiser user's **Ad Portfolio** found in the menu. Marketing and Advertising will only display ad content valid for the connection configuration being used.

8. Click **Save**

A winning bid by an advertiser results in the bid's ad content being made available to an external application during the duration of the campaign.

Responding to External Application Requests Ad

When an external application/entity calls the SPAppAdRequest operation providing an SpAppID, Marketing and Advertising identifies and authenticates that the request is coming from the ad space owner. Marketing and Advertising will retrieve the winning advertiser's ad content and serve it to the external application/entity in a SPAppAdResponse message.

Reporting Ad Server Statistics

The external application/entity can provide statistical information on ads served (impressions) to Marketing and Advertising by utilizing the SPAppMetricsReport operation. See "SpApp Ad Web Service" for additional information.

6

Integrating with Siebel Enterprise Marketing Suite

Oracle Communications Marketing and Advertising can be integrated with the Siebel Enterprise Marketing Suite in order to leverage the Siebel interface and datastore in creating marketing campaigns.

Overview

Coordinated campaigns can be initiated in Siebel as a new offer and then sent to Marketing and Advertising for execution via various delivery channels. Integration between the Siebel Marketing and the Marketing and Advertising applications takes place over productized web services. Marketing and Advertising campaigns are invoked from Siebel Marketing as part of the *Campaign Launch* workflow.

Integrated offer to campaign management allows organizations using both Siebel Enterprise Marketing Suite and Marketing and Advertising to create, execute, manage and monitor marketing initiatives from within Siebel.

To use this feature, you must complete the following steps:

1. Preparing Siebel and Marketing and Advertising
2. Configuring Siebel
3. Configuring Oracle Communications Marketing and Advertising
4. Creating a Test Campaign in Siebel Marketing
5. Verifying the Integration

Note: Marketing and Advertising integrates with Siebel Enterprise Marketing Suite by using the embedded Siebel Email Marketing Server. Use of a single Siebel Marketing instance for both native Siebel email and Marketing and Advertising campaigns is not supported. It is recommended that a dedicated Siebel Marketing instance is used with Marketing and Advertising.

Preparing Siebel and Marketing and Advertising

The first step in integrating Siebel Enterprise Marketing Suite with Marketing and Advertising is ensuring that the applications and modules have been properly installed. This section will cover the high level environment prerequisites needed to successfully launch offers from Siebel Marketing into Marketing and Advertising.

For detailed information on specific application installation procedures please consult the product documentation available at the referenced links.

Siebel Prerequisites

Marketing and Advertising works with Siebel Enterprise Marketing Suite version 8.1. The Siebel Enterprise Marketing Suite installation consists of a Siebel Server deployment and its required subcomponents (Siebel Gateway Name Server, Database and Web server Extensions).

To successfully integrate Siebel Marketing and Marketing and Advertising ensure that all the necessary Siebel components are installed and operational by accessing the Siebel Marketing web interface at the following URL:

`http://Siebel Server Address:Port/marketing_enu`

If Siebel Marketing is unavailable please consult *Siebel Marketing Installation and Administration Guide* for information on installation and troubleshooting at:

http://download.oracle.com/docs/cd/E14004_01/books/MKTG_InstallAdmin/MKTG_InstallAdminTOC.html

Consult your Siebel application administrator if assistance is required.

Oracle Communications Marketing and Advertising Prerequisites

Marketing and Advertising version 5.1 or later must be installed and functional.

You must have both Oracle WebLogic Administration Console Administrator rights for the domain on which Marketing and Advertising is running as well as valid credentials for a system owner user of the application.

Check the following URL to ensure that you can access and log into the Oracle WebLogic Administration Console as an Administrator:

`http://Marketing Server Address:WebLogic Access Port/console`

Ensure that the Marketing and Advertising Siebel Campaign Manager Enterprise and Web Applications are deployed and running in your domain. The *Summary of Deployments* should contain an entry for the **ocma-siebel** module.

Check the following URL to ensure that you can access and log into the Marketing and Advertising Server as a system owner:

`http://Marketing Server Address:WebLogic Access Port/ocma`

If the Oracle WebLogic and Marketing and Advertising servers are unavailable please consult *Oracle Communications Marketing and Advertising Installation Guide* for information on installing and starting the application.

Shared Filesystem

The Siebel file system is used to share ad content and subscriber address information from Siebel Marketing to Marketing and Advertising for use in campaigns. The Siebel **fs** directory must be shared from the Siebel server and mounted on the Marketing and Advertising server.

Depending on the operating system(s) in use there will be different ways to configure the shared filesystem. The following example depicts configuring the file share on a Unix operating system using the Network File System (NFS) protocol:

1. Login to the Siebel server as root or a user with administrative rights.

2. Edit the /etc/exports file by adding an entry for the Siebel **fs** directory, Marketing and Advertising server IP address and necessary permissions on a single line:

   /u01/app/siebel/fs <Oracle Communications Marketing and Advertising server IP> (rw,sync,all_squash)

3. Start the NFS services on the Siebel server:

   ```
   /etc/init.d nfs start
   ```

4. Login to the Marketing and Advertising server as root or a user with administrative rights.

5. Create a new directory in the **$OCMA_HOME** directory to mount the Siebel share:

   ```
   mkdir $OCMA_HOME/siebel
   ```

6. Mount the remote Siebel fs directory locally on the Marketing and Advertising server:

   ```
   mount <Siebel Server IP Address>:/u01/app/siebel/fs $OCMA_HOME/siebel
   ```

Please consult the necessary operating system documentation and systems administrators for your environment if assistance is needed. Be sure to consider security requirements when making the Siebel **fs** directory available via file sharing.

Configuring Siebel

The Siebel Marketing application has a number of necessary setup activities needed for integrating with Marketing and Advertising. All of the following settings are configured from within the Siebel Marketing GUI.

1. Configuring the Siebel Outbound Web Service
2. Configuring the Siebel WebMarketing Inbound Web Service
3. Configuring the Siebel Email Marketing Server
4. Enabling the Siebel Component Groups
5. Starting the Siebel Marketing Workflow Processes

To complete the necessary procedures login to the Siebel Marketing GUI at the following URL as a user with administrative rights:

`http://`*`Siebel Server Address:Port`*`/marketing_enu`

The following procedures assume familiarity with Siebel Marketing. If additional detail or assistance is required please contact your Siebel application administrator.

Configuring the Siebel Outbound Web Service

The Siebel Marketing SendMailingService Web Service must be altered with the Marketing and Advertising WSDL URL representing the target web service to invoke during campaign execution.

To make the necessary configuration changes in the Siebel Marketing web client:

1. From the **Navigate** menu, select **Site Map**.
2. Select **Administration - WebServices**.
3. Select **Outbound Web Services**.
4. Scroll down the list of Outbound Web Services and select the SendMailingService to bring up the Service Port configuration.

5. Replace the **Address** value for SendMailing with the following URL containing values specific to your environment:

```
http://Marketing Server Address:WebLogic Access
Port/SiebelCampaignManager/SendMailing?WSDL
```

Configuring the Siebel WebMarketing Inbound Web Service

Do not believe this step is necessary anymore with the Siebel callback issue fixed post Load 8. Will confirm with Dev that this is true.

Marketing and Advertising requires configuration of the Siebel WebMarketing Inbound Web Service.

To configure the Siebel WebMarketing Inbound Web Service:

1. From the **Navigate** menu, select **Site Map**.
2. Select **Administration - WebServices**.
3. Select **Inbound Web Services**.
4. Scroll down the list of Outbound Web Services and select the **WebMarketing** Inbound Web Service.
5. In the **Outbound Web Service Port** field edit the **Address** column with the actual value:

```
http://Siebel Server Address:Port/eai_
enu/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&WSSOAP=1
```

This URL address is found in the WebMarketing WSDL file in the **soap address** element whose value is the inbound web service URL.

Configuring the Siebel Email Marketing Server

When an offer is created in Siebel one or more treatments or delivery channels need to be assigned for use in the marketing campaign. In order to use Marketing and Advertising as a delivery channel an Email Marketing Server must be configured to use the SendMailing Outbound Web Service Port previously configured.

To create the necessary Email Marketing Server in the Siebel Marketing Web Client:

1. From the **Navigate** menu, select **Site Map**.
2. Select **Administration - Marketing**.
3. Select **Servers**.
4. Click the **New** button.
5. Under the **Type** value select **Email Marketing Server.**

After creating the new Email Marketing Server ensure that it is selected and proceed to the lower configuration area for the new server.

1. Click the **New** button.
2. Select the **Parameter Type** of **Email sending daemon (ESD)**.
3. In the **Outbound Web Service Port** field select the **SendMailing** port from the **Pick Web Service Port** popup window.

Enabling the Siebel Component Groups

The Siebel Marketing to Marketing and Advertising integration requires the following Siebel Component Groups to be enabled:

- Workflow Management
- Enterprise Application Integration
- Marketing Server
- Communication Management
- Marketing Object Manager

Check that the listed Siebel Component Groups are enabled in the Siebel Marketing Web Client:

1. From the **Navigate** menu, select **Site Map**.
2. Select **Administration - Server Configuration**.
3. Select **Enterprises**.
4. Select **Component Groups**.
5. Scroll through the list of Components and, if necessary, highlight the necessary Components and click the **Enable** button.
6. Restart the Siebel Marketing instance to activate the configuration changes.

   After the Siebel instance has been restarted, navigate to the **Component Groups** screen and make sure the listed component groups remain enabled.

Starting the Siebel Marketing Workflow Processes

Prepare Siebel Marketing to launch marketing campaigns by activating the Marketing Workflows in the Siebel Marketing Web Client:

1. From the **Navigate** menu, select **Site Map**.
2. Select **Administration - Business Process**.
3. Select **Repository Workflow Processes** under **Workflow Deployment**.
4. In the Repository Workflow Processes pane click the **Query** button.
5. In the **Group** field select **Marketing**.
6. Click the **Go** button.
7. Activate the required Marketing Workflows by highlighting the Workflow and clicking the **Activate** button.

Configuring Oracle Communications Marketing and Advertising

Marketing and Advertising must be configured for integration with Siebel Marketing. Two main activities are required before Siebel requests can be handled. These steps are described in the following sections:

1. Configuring the Web Service in WebLogic
2. Preparing Oracle Communications Marketing and Advertising for Siebel Marketing

Configuring the Web Service in WebLogic

Marketing and Advertising must be provided with Siebel integration parameters to allow communication between the applications. These parameters include Siebel Marketing application user credentials, the previously described Siebel **fs** share location on the Marketing and Advertising server and the Siebel Web Service Callback endpoint URL for communicating with Siebel.

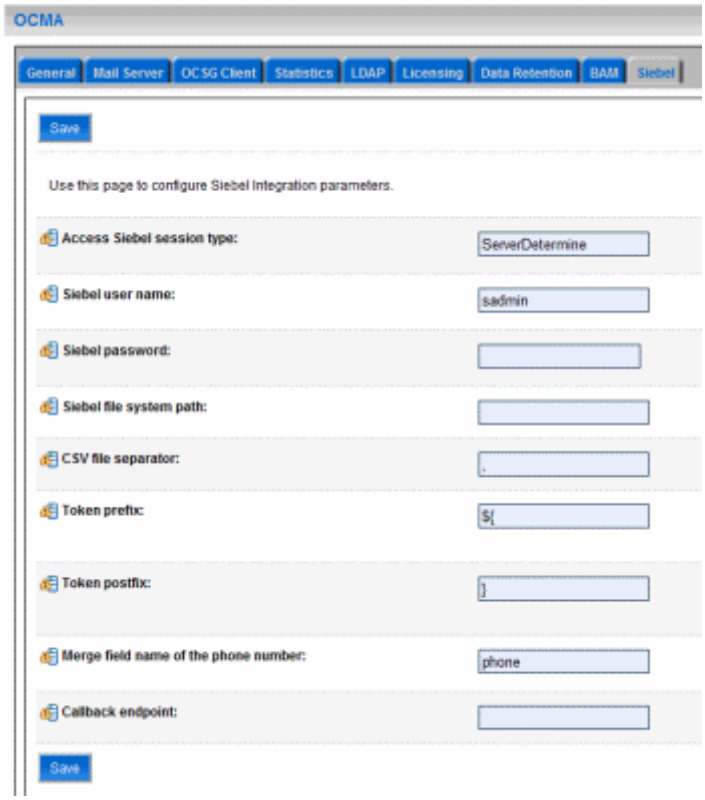
All of the required parameters are configured in a single location within the Oracle WebLogic Administration Console for the domain on which Marketing and Advertising is running. The following steps indicate how to accomplish the necessary configuration:

1. Access the Oracle WebLogic Administration Console for the domain on which Marketing and Advertising is running at:

   `http://Marketing Server Address:WebLogic Access Port/console`

2. Login to the Administration Console with a user with administrative permissions.
3. Under **Domain Structure** on the left side of the screen select **OCMA**.
4. Click the **Siebel** tab.

   The Siebel parameter page appears.



This screenshot depicts the Siebel configuration page in the Oracle WebLogic Administration Console for the domain where Oracle Communications Marketing and Advertising is running. There are a number of fields described in detail below that require values inputted for the Siebel Marketing integration.

****

5. Enter the Siebel information into the fields as follows:
   - Leave the **Access Siebel session type** value of **ServerDetermine**.
   - In the **Siebel user name** field, enter the Siebel administrative user `sadmin`.
   - In the **Siebel password** field, enter the sadmin user password.
   - In the **Siebel file system path** field, enter the local path to where the Siebel **fs** share is mounted.
   - Leave the default entries for the **CSV file separator**, **Token prefix**, **Token postfix**, and the **Merge field name of the phone number** fields for now.
   - In the **Callback endpoint** field, enter the following URL customized for your environment:

     `http://`*`Siebel Server Address:Port`*`/eai_enu/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&WSSOAP=1`
6. Click **Save**.

Preparing Oracle Communications Marketing and Advertising for Siebel Marketing

A number of Siebel-related entities must be created in the Marketing and Advertising application. Marketing and Advertising utilizes these entities in executing campaigns originating from Siebel Marketing. The entities provide the roles and functionality needed to translate a Siebel Marketing campaign launch into a running campaign in Marketing and Advertising.

The following Marketing and Advertising entities need to be created:

- Organizations
- Users
- Connection Configuration
- Campaign Space
- Marketing Space

For information on how to create the required entities in Marketing and Advertising see *Oracle Communications Marketing and Advertising System Owner Online Help*.

Creating Oracle Communications Marketing and Advertising Organizations and Users

Login to the Marketing and Advertising Web Client as a system owner user and create the following entities:

- A Siebel *Network Provider Organization*, e.g. `siebelNPO`
- A Siebel *Network Provider User*, e.g. `siebelNPuser`
- A Siebel *Ad Service Provider Organization*, e.g. `siebelASPO`
- A Siebel *Ad Service Provider User*, e.g. `siebelASPuser`
- A Siebel *Ad Space Owner Organization*, e.g. `siebelASO`
- A Siebel *Ad Space Owner* User, e.g. `siebelASOuser`
- A Siebel *Advertiser User*, e.g. `siebelADuser`

Creating a Usable Campaign Space

To execute a marketing campaign launched from Siebel Marketing there must be a valid Marketing Space for Siebel. Configuring a Marketing Space in Marketing and Advertising requires a Marketing Connection configuration to be setup first.

Configure the necessary entities as follows:

1. Create a Siebel *Marketing Connection Configuration*, e.g. `SiebelCC`, using the previously created Siebel Ad Space Owner.
2. Create a Siebel *Marketing Space*, e.g. `SiebelMS`, using the Connection Configuration created in the previous step also populating the remaining required fields.

Setting the Oracle Communications Services Gatekeeper Network Provider Name

To identify the Marketing and Advertising instance servicing Siebel Marketing set the **Ocsg Network Provider** in the Oracle WebLogic Administration Console for the domain in which Marketing and Advertising is running.

1. Access the Oracle WebLogic Administration Console for the domain on which Marketing and Advertising is running at:

   `http://`*`Marketing Server Address:WebLogic Access Port`*`/console`

2. Login to the Administration Console with a user with administrative permissions.
3. Under **Domain Structure** on the left side of the screen select **OCMA**.
4. In the **General** tab replace the entry in the **Ocsg Network Provider** with the name of the Siebel Network Provider created above (e.g. `SiebelNPO`).

Creating a Test Campaign in Siebel Marketing

Once all of the previous steps have been completed the integration between Siebel Marketing and Marketing and Advertising is functional.

The following section provides a high level demonstration on how to test and operate the integration between the two applications. Detailed procedures for creating Siebel Marketing campaigns/offers, including more information on treatments and treatment templates, can be found in *Siebel Marketing User Guide*.

http://download.oracle.com/docs/cd/E14004_01/books/MKTG_User/MKTG_UserTOC.html

Consult your Siebel application administrator if assistance is required.

Create Siebel Marketing Offer

A Siebel Marketing sourced campaign running in Marketing and Advertising is based on a Siebel Marketing offer. To initiate a new integrated campaign begin by creating a new Siebel Marketing offer.

1. Access the Siebel Marketing interface and login with a user with the necessary permissions to create a new offer at:

   `http://`*`Siebel Server Address:Port`*`/marketing_enu`

2. Select the **Offers** tab.
3. Create a new offer by clicking on the **New** button.
4. In the **Name** field enter a value for the new offer (e.g. `TestOffer`).

5. In the **Type** field select one of the values from the pulldown.
6. Save the record by clicking on the **Menu** button and selecting **Save Record**.

Create Siebel Marketing List

A Siebel Marketing *list* has various functions. Marketing and Advertising uses a marketing list of contacts contained in Siebel Marketing as the selected recipients of an offer/campaign. To test the integration between Siebel Marketing and Marketing and Advertising there must either be an existing list of contacts which contain the needed information for a campaign (e.g. mobile phone numbers for a SMS campaign) or a new list must be generated from within the Siebel Marketing interface.

To generate a new list:

1. Access the Siebel Marketing interface and login with a user with the necessary permissions to manage lists at:

   `http://Siebel Server Address:Port/marketing_enu`
2. Click the **Navigate** pull down menu and select **Site Map**.
3. Click the **List Management** link
4. Under **List Management** click the **All Lists** link.

   If a list suitable for testing the integration already exists proceed to Creating Siebel Treatments Using Sample Template Files.
5. Click the **New** button to create a new list.
6. Save the newly created list by clicking another field or clicking the **Menu** button.

   The **List Name** becomes a clickable hyperlink.
7. Click the list hyperlink to edit the contacts.
8. In the **List Contacts and Prospects** tab click the **New** button to add contacts to the list.

   Ensure that the contacts you add to the list possess a value in the Mobile Phone # field so that Marketing and Advertising can deliver a targeted campaign message to the contact. You may add any number of contacts to the list for testing purposes. For production campaigns and lists you will likely have a large number of contacts in the lists used for offers/campaigns.

Consult *Siebel Bookshelf v8.1* and your Siebel application administrator if more assistance is required configuring lists.

Creating Siebel Treatments Using Sample Template Files

A channel-specific (e.g. SMS) instance of a Siebel Marketing offer requires a specific treatment.

Sample treatment xsd files for use with Marketing and Advertising are provided in the `ocma-siebel-jar-5.0-SNAPSHOT.jar` jar file contained in the `ocma-siebel-ear.ear` application.

Uncompress the sample template xsd files so that they can be used to create valid treatment templates for use with Siebel Marketing.

1. Login to the server where the Oracle WebLogic domain hosting Marketing and Advertising is running.

2. Uncompress the `ocma-siebel-ear.ear` file located in the `$ORACLE_MIDDLEWARE_HOME/ocma_5.1/applications` directory (e.g. `C:\Oracle\Middleware\ocma5.1\applications` on Windows operating systems).
3. In the `lib` folder uncompress the `ocma-siebel-jar-5.0-SNAPSHOT.jar` file.
4. Extract and save the following xsd template files to a location where they can be edited with a text editor and then uploaded from within the Siebel Marketing Web client:
   - mms_template.xsd
   - sms_template.xsd
   - wap_template.xsd

   Each sample template xsd file should be used as a basis for creation of channel-specific treatment templates in Siebel Marketing.

Creating Treatment Templates

The sample template xsd files must be edited for your specific environment and offer/campaign requirements. A number of elements within the templates specify Marketing and Advertising entities to be used by offers/campaigns. Replace the entity fields with the values specific to your environment.

The *text* element is used for personalization of offers using contact information stored in Siebel Marketing lists and shared with Marketing and Advertising. Any number and combination of the following Siebel contact fields can be included for use in the text field of a treatment template:

- First Name
- Middle Name
- Last Name
- Work Phone #
- Home Phone #
- Fax Phone #
- Cellular Phone #
- Email Address
- Job Title
- Account
- Account Location
- Current Date
- Source Code
- Contact Id
- Prospect Id
- Camp Con Id
- Segment/List Name
- Forwarder Message
- Subscription List

Note: The *Cellular Phone #* value corresponds to the *Mobile Phone #* field for a Siebel contact.

A simple example of a valid SMS treatment template file is provided here based on Marketing and Advertising entity names used in this chapter.

The treatment template indicates that the recipient of an offer based on this template is identified by the *Cellular Phone #* field. The offer treatment is customized in this example to include the recipient's first and last names as well as a short text string.

```
<smsContent>
subscriberNumberField>[Field:Cellular Phone#]</subscriberNumberField>
<advertiser>siebelADuser
<text> Hello [Field:First Name] [Field:Last Name]</text>
<textWithoutToken> Please visit www.oracle.com</textWithoutToken>
<marketingSpace>siebelMS</marketingSpace>
<channelType>SMS</channelType>
<serviceProviderName>siebelASPuser</serviceProviderName>
</smsContent>
```

MMS and WAP treatment templates can be created in the same manner using the sample xsd files. Template contents will vary depending on marketing requirements. Offer/campaign strategy will dictate the reusability and specificity of template contents.

Uploading Treatment Templates

Once a valid treatment template has been created it needs to be uploaded into Siebel Marketing so that it can be used to launch an integrated campaign in Marketing and Advertising.

To upload the treatment template into Siebel Marketing:

1. Access the Siebel Marketing interface and login with a user with the necessary permissions to manage treatment templates at:

   `http://Siebel Server Address:Port/marketing_enu`

2. Select the **Offers** tab.
3. Click the **Treatment Templates** link.
4. Click the **New File** button and navigate to the location where the treatment template is saved.
5. Select the template file and click the **Open** button to upload the template into Siebel Marketing.
6. Edit the description field if desired.

Create Treatment

A Siebel Marketing treatment based on a Marketing and Advertising compliant template can now be generated. The treatment will include the offer. This treatment will be used by a Siebel campaign.

To create a new Siebel treatment:

1. Access the Siebel Marketing interface and login with a user with the necessary permissions to manage treatment templates at:

`http://Siebel Server Address:Port/marketing_enu`

2. Select the **Offers** tab.
3. Click the **Email Treatments** link.
4. Click the **New** button and enter a name for the new treatment.
5. In the **Treatment Details** tab select the name of the offer previously created in Create Siebel Marketing Offer (e.g. `TestOffer`).
6. Select the **Edit Email** tab.
7. Click the **New** button.
8. In the **Template Name** field search for and select the Marketing and Advertising template previously uploaded.

   The **Template Contents** field should display the properly formatted template.

   Note: MMS treatments require an image file attachment. If a MMS template is being used to create a MMS channel treatment you must select the **attachments** tab and add an image file.

9. Save the treatment.

Creating a Siebel Campaign

When the campaign is launched as a new offer in Siebel, Marketing and Advertising will be notified to execute the campaign based on the contents of the treatment and the marketing list specified in the campaign and stored on the shared file system.

Note: The following Siebel Campaign procedure may not be indicative of your organization's Siebel Marketing implementation. The steps described here are high-level in nature and do not account for possible workflows in your campaign processes (e.g. approvals). Consult your Siebel administrator for implementation specific details of your campaign creation process.

To create a new Siebel Campaign:

1. Access the Siebel Marketing interface and login with a user with the necessary permissions to manage campaigns at:

   `http://Siebel Server Address:Port/marketing_enu`

2. Select the **Campaigns** tab.
3. Click the **New** button.
4. In the **Name** field enter a value for the new campaign.
5. Save the newly created campaign by clicking another field or the Menu button

   The campaign **Name** becomes a clickable hyperlink.

6. Click the name hyperlink to edit the **Campaign Details**.
7. Select the **Design** tab.
8. In the **Offers** sub-tab Click the **New** button.

9. Select the previously created Offer (e.g. `TestOffer`).
10. In the lower **Treatments** section click the **New** button.
11. Select the previously created treatment based on the uploaded template.
12. Select the **Segments/List** sub-tab.
13. Click the **Add List** button.
14. Select the previously created list.
15. Select the **Allocation** sub-tab.
16. Click the **Update Count** button and verify that the **Net Count** corresponds to the number of members in the selected contact list.

Launching the Siebel Campaign

Once the campaign has been created in Siebel Marketing it must be launched.

To launch the newly created Siebel Marketing campaign:

1. Access the Siebel Marketing interface and login with a user with the necessary permissions to manage campaigns at:

   `http://`*`Siebel Server Address:Port`*`/marketing_enu`
2. Select the **Campaigns** tab.
3. Select the campaign name hyperlink containing the Marketing and Advertising treatment.
4. Select the **Execute** tab.
5. Click the **Launch Wave** button.
6. Confirm the **Scheduled Start** date and time and click **OK**.

Verifying the Integration

Once the Siebel Campaign wave has been launched verify that the treatment has been successfully initiated in Siebel Marketing and that the campaign has been created in Marketing and Advertising.

Confirming Siebel Campaign Status

In the Siebel Marketing verify the successfully launch of the campaign wave by:

1. Confirming that the **Status** field in the **Execution Status** sub-tab for the wave has changed to *launched*.
2. Confirming that the **Status** field in the **System Tasks** sub-tab for **Campaign Launch** process has changed to *Completed*.
3. Confirming that the mime content and contacts list files have been generated in the respective shared directories on the Siebel server host:

   `$SIEBEL_HOME/fs/Marketing/EmailOfferContent/<name>.mime`

   `$SIEBEL_HOME/fs/Marketing/Lists/<name>.txt`

Confirming Oracle Communications Marketing and Advertising Campaign Creation

In the Marketing and Advertising Web Client, verify the successful launch of the marketing campaign by confirming that a campaign has been created for the Siebel Advertiser user.

7

Integrating with Oracle Business Activity Monitoring

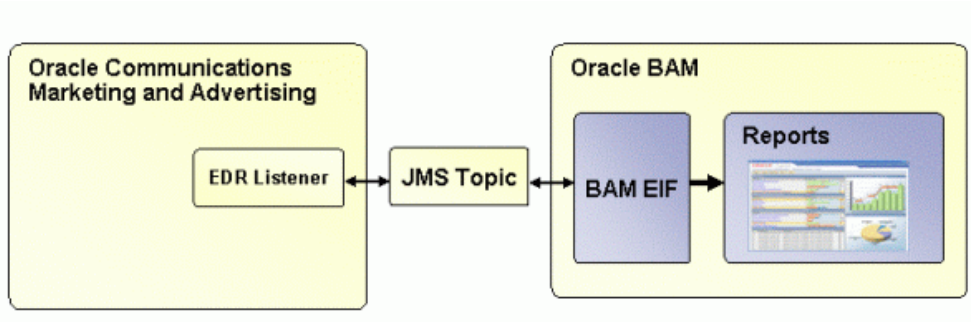
Oracle Communications Marketing and Advertising can be set up to deliver real-time event data to Oracle Business and Activity Monitoring (BAM).

Overview

Marketing and Advertising servers continuously record basic information about their status through an internal event mechanism. An event is fired at designated times as the server functions, for example, whenever a request crosses module boundaries, specific methods are called, or exceptions are thrown. This event information is sent to the EDR Service, where it is processed using an XML-based filter, which separates them into types. The filters can also be used to transform the data in the original event, including adding other useful information.

Out of the box, there are three filters that produce three distinct types of data: Event Data Records, Charging Data Records, and Alarms. When the information has been processed by the filters, it is delivered to type-specific listeners. To communicate with Oracle BAM, Marketing and Advertising delivers a special version of the Charging Data Record modified for this purpose to an internal EDR listener which then publishes it to a JMS topic on a JMS server accessible to both it and Oracle BAM. These records describe both "Ad Served" and "Ad Impression" events. See Figure 7–1.

Figure 7–1 Publishing to JMS Topic



The illustraion shows the Marketing and Advertising Edr listener communicating with BAM using a JMS topic. BAM is shown with the BAM EIF and a small screen capture of aa sample reports dashboard.

****

Both Marketing and Advertising and Oracle BAM must be configured to communicate with this JMS server. These instructions cover only the Marketing and Advertising part of the process.

Configuring the Connection

Setting up Marketing and Advertising for delivering event information is done using the Administration console. For complete information see the BAM section of "Configuring the Web Application" in *System Administrator's Guide*, another document in this set.

Note: The **JMS topic JNDI name** and **JMS Connection Factory JNDI name** values entered for the Marketing and Advertising and the Oracle BAM configurations must be identical.

JMS Message Content

Messages delivered to the JMS topic are of type `javax.jmsTextMessage` and are enclosed in a <MultiMessage> element with individual messages enclosed in a <PRE> element. Each individual message corresponds to a single Ad Impression or Ad Served event and is organized as key/value pairs:

```
<key1>value1</key1><key2>value2</key2>...<keyn>valuen</keyn>
```

The following information is sent:

- **Timestamp**: The time of the event, as recorded on the Marketing and Advertising server, presented as a localized version of a Java Date
- **Campaign Id**: The internal ID for the campaign in the system
- **Campaign Name**: The name given the campaign when it was initiated
- **Advertiser Id**: The internal ID for the Advertiser organization that initiated the campaign
- **Service Provider Id**: The internal ID for the Ad Service Provider organization that is responsible for the Advertiser organization that initiated the campaign
- **Network Provider Id**: The internal ID for the Network Provider organization that is responsible for the Connection Configuration used by the campaign
- **Price**: The price paid for the message, in currency defined by the Marketing and Advertising server
- **Event Type**: An enumeration. Either "Ad Served" or "Ad Impression". An "Ad Served" indicates that the message has been successfully delivered to the underlying network. An "Ad Impression" indicates that a Delivery Notification concerning the message has been received back from the underlying network
- **Category List**: The types used to define demographic categories used to send bulk messages. The types depend on configuration, but could include such things as Age, Gender, and so forth
- **Subscriber List**: The address of the subscriber who received the message
- **Delivery Notification Enabled**: A boolean, indicates if Delivery Notification is enabled for this campaign

8

Integrating with Oracle Map Viewer

Oracle Communications Marketing and Advertising can be set up to query map information from an Oracle Map Viewer mapping service to use enhanced visual mapping features used in Location Based Marketing.

Overview

The Location Based Marketing features in Marketing and Advertising can acquire map presentation data for use in campaign configuration.

Visual map data is sourced from an optional integration with the *Oracle Fusion Middleware Map Viewer*. The *Oracle Fusion Middleware Map Viewer* product is separately licensable and may have other dependent software products from Oracle (e.g. *Oracle Spatial*) as well as third-party providers (e.g. map data vendor).

For more information about *Oracle Fusion Middleware Map Viewer* consult the product information page at:

`http://www.oracle.com/technetwork/middleware/mapviewer/overview/index.html`

For more information about Oracle Spatial consult the product information page at:

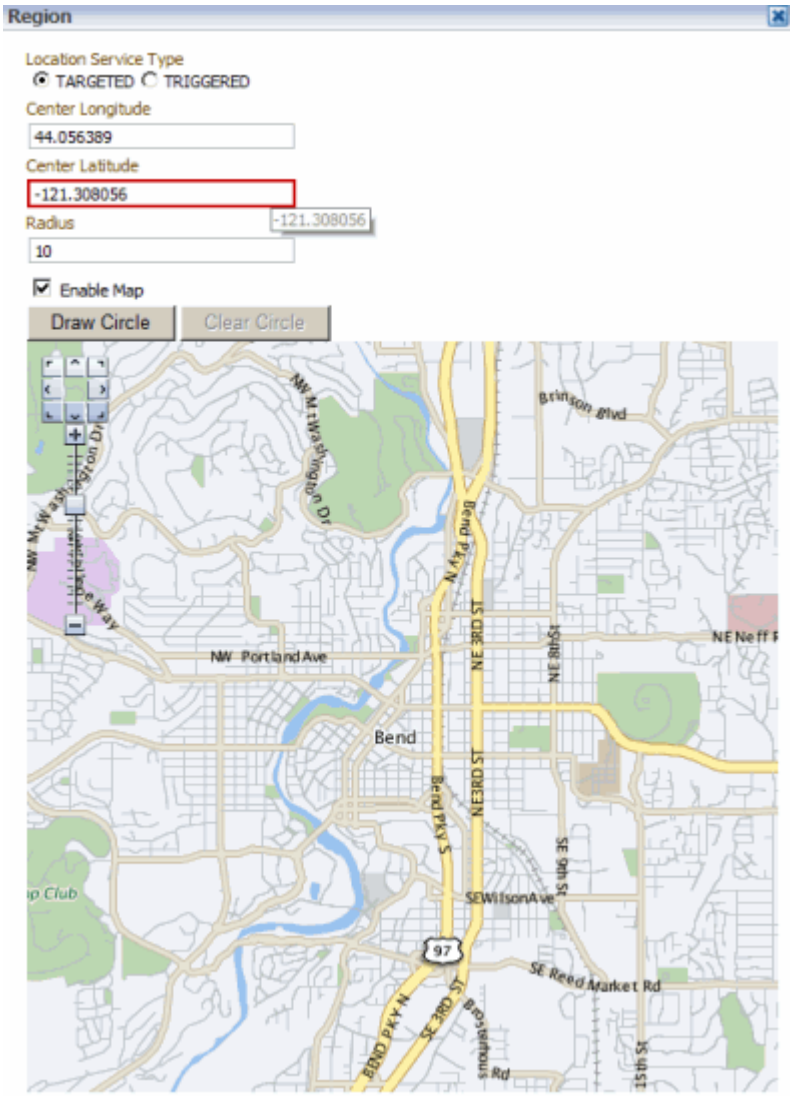
`http://www.oracle.com/technetwork/database/options/spatial/index.html`

Additional details about Oracle products and licensing can also be obtained from your Oracle license representative.

The following steps are required in order to use the visual map features in Oracle Communications Marketing and Advertising

1. Determine the Oracle Fusion Middleware Map Viewer Service URL
2. Add the Web Service URL in the System Owner GUI Interface

Visual map data is used in Marketing and Advertising to allow advertisers to graphically target subscribers in a geographical area when configuring campaigns. The following image depicts the visual map data displayed in Marketing and Advertising that is sourced from *Oracle Fusion Middleware Map Viewer*.



The graphic depicts the Region popup window shown when configuring the location inclusion area for lcampaigns utilizing the location premium feature. Users use this window to either enter coordinates and radius information or use a map to draw a circle indicating the location inclusion area.

****

Determine the Oracle Fusion Middleware Map Viewer Service URL

Consult the system administrator for your licensed *Oracle Fusion Middleware Map Viewer* installation and obtain the mapping service URL for your instance. The URL should be in the following format:

`http://Map Viewer Server Address:Port/mapviewer`

The URL will be used in the Marketing and Advertising web client by a system owner user to designate the mapviewer Web service URL.

For testing purposes, Oracle provides the following demonstration mapping service URL for use with *Oracle Fusion Middleware Mapviewer*:

`http://elocation.oracle.com/mapviewer`

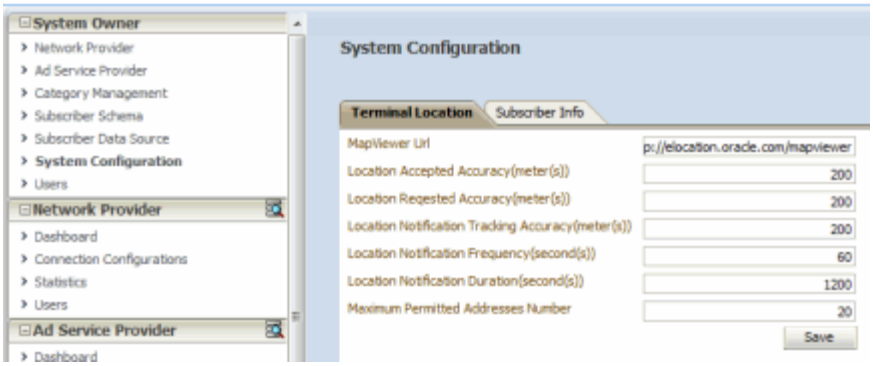
Add the Web Service URL in the System Owner GUI Interface

Marketing and Advertising is configured to use the mapviewer service URL by a system owner user. To add the mapviewer service URL to your instance of Marketing and Advertising:

1. Access the Marketing and Advertising web interface at:

   http://*Marketing Server Address*:*WebLogic Access Port*/ocma

2. Login to the Marketing and Advertising web interface as a system owner user
3. Under the **System Owner** menu in the left panel select the **System Configuration** hyperlink
4. In the **Terminal Location** tab enter the mapviewer service URL in the **MapViewer Url** field as shown in the following screenshot.



The screenshot depicts the Terminal Location tab in the System Configuration area of the Oracle Communications Marketing and Advertising webclient for a System Owner user. The graphic is intended to show both the MapViewer Url field and the Save button used in entering the webservice URL information for the mapviewer integration.

****

5. Click **Save**.

9

Integrating with Oracle Communications Billing and Revenue Management

Oracle Communications Marketing and Advertising can be set up to deliver charging data to Oracle Communications Billing and Revenue Management, or any billing system that supports Diameter Rf.

Overview

The CDRtoDiameter module in Marketing and Advertising converts Charging Data Records (CDRs) to Diameter Rf format. To use this feature, you must do the following steps:

1. Deploy the CDRtoDiameter Service
2. Configure the CDRtoDiameter Service

Deploy the CDRtoDiameter Service

The CDRtoDiameter module is not deployed by default. If you wish to use this functionality, you must first deploy the appropriate.ear file. For information on how to deploy the service, see *Oracle WebLogic Server Deploying Applications to WebLogic Server* at:

`http://download.oracle.com/docs/cd/E12840_01/wls/docs103/deployment/`

The service is packaged in two ears, depending on the type of installation you have. The `cdr_to_diameter-single.ear` file is for use in single server domains and `cdr_to_diameter.ear` is for clustered installations. Both files are found in `<OCSG_Home>/applications`.

The service is a cluster singleton, so it will execute only on one server at any given time, and is transferred to another server in case of server failure. The management part is distributed to all servers in the cluster, so it can be managed from any server in the cluster.

> **Note:** Some Diameter requests may be dropped during patching, redeployment, or upgrade of the CDRtoDiameter module. Check the database for the time period during which the transition took place.

Configure the CDRtoDiameter Service

When the module has been deployed, it will appear in the Container Services section of the Administration console. From the **Domain Structure** panel, open the **OCSG**

node ->**<ServerName>**->**Container Services**. To configure the service, follow the instructions in "CDRs and Diameter" in the *Oracle Communications Services Gatekeeper System Administrator's Guide* at:

http://download.oracle.com/docs/cd/E14148_01/wlcp/ocsg41_otn/admin/cdrtodiameter.html

AVPs in Diameter Rf

The Diameter Rf message that is sent from Marketing and Advertising is made up of a somewhat different set of AVPs than the Diameter Rf messages that are sent from Oracle Communications Services Gatekeeper. Table 9–1 describes what AVPs are sent.

Table 9–1 AVP s for Oracle Communications Marketing and Advertising

AVP	AVP Code	Type	Specification	Description
Session_ID	263	UTF8String	RFC 3588	Used to identify a specific session
Origin-Host	264	DiameterIdentity	RFC 3588	Identifies the endpoint that originated this message. Is equal to the delivery-mechanism host
Origin_Realm	296	DiameterIdentity	RFC 3588	Identifies the Realm of the originator of the delivery-mechanism
Destination-Host	293	DiameterIdentity	RFC 3588	Identifies the destination host
Destination-Realm	283	DiameterIdentity	RFC 3588	Identifies the Realm of the destination host
Accounting-Record-Type	480	Enumerated	RFC 3588	The type of accounting record being sent
Accounting-Record_Number	485	Unsigned32	RFC 3588	Identifies this record within this session
Acct-Application-ID	259	Unsigned32	RFC 3588	Advertises support of the Accounting portion of an application. The field corresponds to the application ID of the Diameter Accounting Application and is defined with the value 3
User-Name	1	UTF8String	RFC 3588	The Advertiser External ID, assigned when the organization was created
Event-Timestamp	55	Time	RFC 3588	Time the event happened, in seconds since January 1, 1900 00:00 UTC
Service-Indication	704	OctetString	3GPP 29.329	For this product, the value is always "ADVERT_SERVICE"
Event-Type	823	Grouped	3GPP 32.299	Contains information about the type of event for which the accounting-request is generated
Event-Type.Event	825	UTF8String	3GPP 32.299	Specifies the Ad-event type. Always AD_SERVED
Participants-Involved	887	UTF8String	3GPP 32.299	Lists the external ID of the related ASO, ASP and NP. The IDs are delimited by# mark

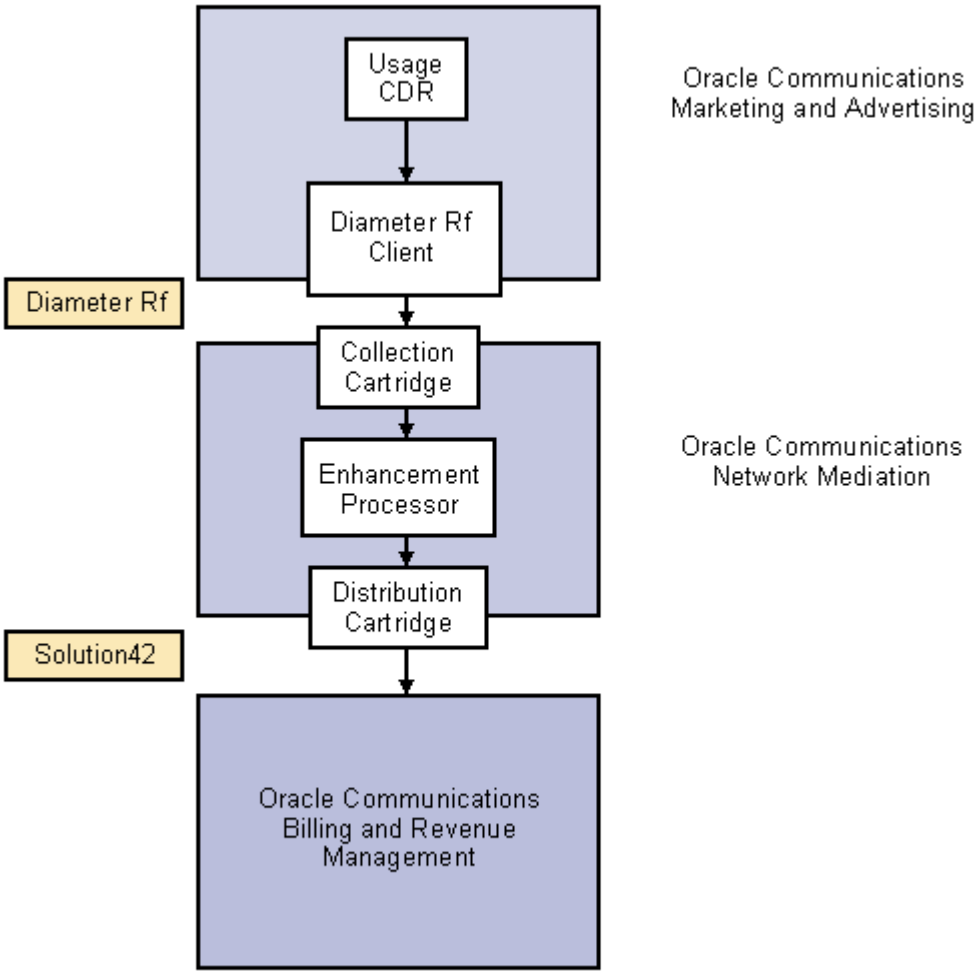
Table 9–1 (Cont.) AVP s for Oracle Communications Marketing and Advertising

AVP	AVP Code	Type	Specification	Description
CC-Money	413	Grouped	RFC4006	Specifies the monetary amount in the given currency, used for storing the price of the ad message
CC-Money.Unit-Value	445	Grouped	RFC4006	Specifies the units as decimal value
Unit-Value.Value-Digits	447	Integer64	RFC4006	Contains the significant digits of the number.
Unit-Value.Exponent	429	Integer32	RFC4006	Contains the exponent value to be applied for the Value-Digit AVP within the Unit-Value AVP
CC-Money.Currency-Code	425	Unsigned32	RFC4006	Specifies the currency, given as the numeric values defined in the ISO 4217 standard.

Note: In the case of overlay installations, a single CDR for an Advertising campaign can create two Diameter Rf messages, one using the Services Gatekeeper message structure and one using the Marketing and Advertising structure. This allows BRM to track both the application (the Ad Space Owner) that initially sent the message and the Advertiser who created the campaign that uses the message as a bearer of its own Ad Content, facilitating charging and revenue sharing arrangements.

Oracle Communications Network Mediation

BRM does not understand Diameter Rf natively. In order to deliver the Marketing and Advertising Diameter Rf message to BRM, an intermediary, Oracle Communications Network Mediation (NM) must be introduced to translate the message into BRM's proprietary Solution42 format. The CDRtoDiameter service delivers the Diameter Rf message to the Network Mediation Collection Cartridge, which functions as the input site. The message is processed internally using an Enhancement Cartridge, and finally is delivered to a Distribution Cartridge that has been created to transform the message into Solution42. The transformed message is delivered to BRM. The flow is illustrated in the graphic below.



The graphic shows a process flow, from top to bottom, representing the translation of Marketing and Advertising Usage CDRs into rateable events in BRM. Usage CDRs go from Marketing and Advertising via a Diameter Rf Client through Network Mediations's Collection Cartridge, Enhancement Processor and Distribution Cartridge into BRM.

****

10

Using the Subscriber Information SPI

Oracle Communications Marketing and Advertising provides a way of querying external datastore subscriber information repositories for statistical and personalization information and for creating subscriber address lists based on demographic categories. Operator data sources can be of many types including Oracle database, LDAPv3 or any other data store when using a custom connector. The Subscriber Information SPI allows operators to customize this functionality to suit the structure and requirements of their own environment.

Overview

The SPI supports two main functions:

- Querying for Information About a Single Address
- Creating an Address List Based on Demographic Categories

Querying for Information About a Single Address

The first functionality that the SPI supports is querying a data source for information about a single address. This information is used in two ways:

- To collect general statistical information (Age, Gender, Opt In/Out preferences, etc., based on the categories defined by the system owner in the Category Management configuration screen in the Web GUI) about the subscriber associated with this address so that this information can be used in category optimized campaigns.
- To retrieve information needed to personalize a promotional message based on tokens in the ad content. For example, an ad might begin "Hi, `%FirstName%`" where FirstName is a displayName in the Marketing and Advertising subscriber schema mapped to a value in the underlying data source. An advertiser can configure the use of tokens in the Marketing and Advertising Web client when creating ad content.

Creating an Address List Based on Demographic Categories

The second functionality that the SPI supports is creating a random list of addresses from the subscriber data source based on demographic categories such as Age or Interests. These categories are defined using the system owner user Web client interface **Category Management** interface. These addresses can then be used by advertisers to create marketing campaigns directed at particular demographic groups.

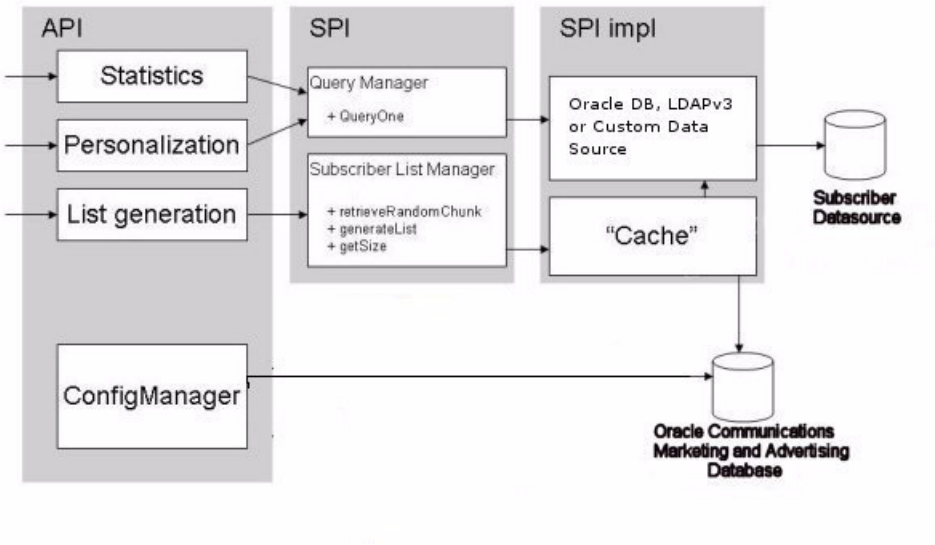
Ad space owners can also tag ad spaces for sale as relevant to certain categories (e.g. an ad space contained in a SMS message from an application providing World Cup

tournament scores can be tagged as relevant to subscribers identified as having an interest in a football category) so that advertisers can better target their campaigns.

The LDAP Example

The standard LDAP implementation illustrates the use of the Subscriber Information SPI. See Figure 10–1.

Figure 10–1 The LDAP Implementation



A graphic with the internal API on the left, the SPI in the center, and the SPI implementation on the right is shown in this illustration. The various components of each tier are described in detail in the chapter text.

****

The left side of the graphic details the internal modules that use the SPI: Statistics, Personalization, and List Generation. It also show the ConfigManager, which manages the mapping between the Marketing and Advertising subscriber categories and subscriber information stored in external data sources.

The center of the graphic shows the SPI itself, with its two modules, the SubscriberInfoQueryManager and the SubscriberListManager. The Query Manager handles the queries that underlie the Statistics and Personalization functionality. The Subscriber List Manager handles the queries that support the generation of lists.

The right side of the graphic illustrates the standard implementation of the SPI, based on LDAPv3. For this particular implementation, the Query Manager queries the LDAP data source directly for single address information. But to manage the overhead of many, many LDAP calls, the list generation function retrieves all the LDAP subscriber information at configurable intervals and stores it, pre-segmented in categories, in the Marketing and Advertising database. To accommodate very large LDAPs, the SPI implementation uses RFC 2696 and retrieves the entire set of nodes under the defined subscriberBaseDN. For information on configuring the subscriber information data source implementation, as well as information on creating the Mappings and Categories XML files, see "Setting Up LDAP and Subscriber Info" in *Oracle Communications Marketing and Advertising System Administrator's Guide*, a separate document in this set.

The Subscriber Information SPI

Subscriber information data sources are created by a system owner user in the Marketing and Advertising Web client. For more information on creating database, LDAP and db customized data sources for Marketing and Advertising to extract demographic information for use in category management see *System Owner Online Help*.

All of the files for the Subscriber Information SPI are contained in the **ocma-new-subscriberinfo-spi-5.0-SNAPSHOT.jar** located in either **ocma-network-node-single-ear.ear** or **ocma-network-node-cluster-ear.ear** (depending on your environment configuration). The ear files can be found in the *Oracle_Middleware_Home*/**ocma_5.1/applications** of your Marketing and Advertising installation. The **ocma-new-subscriberinfo-spi-5.0-SNAPSHOT-javadoc.jar**, which contains the complete Javadoc for the SPI, is located in the same directory.