Oracle® Marketing

API Reference Guide

Release 11*i*

Part No. B10587-01

March 2003

ORACLE®

Oracle Marketing API Reference Guide, Release 11*i*

Part No. B10587-01

Primary Author: Sheralyn Fowler

Contents

2 User Hooks

3 Campaign

4 Campaign Schedule

5 Public Metrics

9 Event Registration

10 Events

11 Event Schedule

12 List

13 List Entries

14 List Generation

Send Us Your Comments

Oracle Marketing API Reference Guide, Release 11*i*

Part No. B10587_01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us via postal service:

Oracle Corporation
Oracle Marketing Documentation Manager
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Introduction

This API Reference guide describes the public APIs for Oracle Marketing and provides information to help you work effectively with these APIs.

Intended Audience

This guide is to be used by experienced system administrators and Oracle implementation consultants who understand Oracle CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning) applications. You must have an understanding of Oracle CRM and ERP applications and an understanding of the use of APIs in general, before proceeding.

How This Guide is Organized

The first two chapters are an introduction to Oracle Marketing and its technology requirements. The third chapter consists of an overview of the entire implementation process. The fourth and fifth chapters cover the basic implementation and the implementation of specific business functions. These are followed by chapters on administration settings, user setups and post implementation steps.

Following the chapters of this guide are a number of appendices summarizing the options and settings used in the guide as well as useful reference information.

This guide is optimized for online viewing as related topics and steps are hyperlinked for convenience. It is best viewed in PDF or HTML formats.

How to Use This Guide

This Guide is designed as a reference to the APIs which are available for Oracle Marketing.

Typographic Conventions

This document uses the following typographic conventions:

`Monospace text`	Monospace text represents code or SQL statements.
lowercase italics	Lowercase italics in text represent variables. Substitute an appropriate value for the variable.
UPPERCASE	Uppercase characters within the text represent command names, SQL reserved words and keywords, and terms associated with the Oracle database.
Indentation	Indentation helps to show structure within code examples, but is not required.
<text>	Text inside angle brackets can mean either of the following: ■ It denotes a variable that is replaced with an actual value at runtime. ■ It indicates XML elements in discussions about XML code.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Other Information Sources

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of Oracle CRM Application Foundation.

If this guide refers you to other Oracle Applications documentation, use only the Release 11i versions of those guides.

Online Documentation

Oracle*MetaLink*

Oracle*MetaLink* is your self-service support connection with web, telephone menu, and e-mail alternatives. Oracle supplies these technologies for your convenience, available 24 hours a day, 7 days a week. With Oracle*MetaLin*k, you can obtain information and advice from Top Tech Documents and forums, download patches, download the latest documentation, look at bug details, and create or update TARs. To use MetaLink, register at (http://metalink.oracle.com).

Alerts: You should check Oracle*MetaLink* alerts before you begin to install or upgrade any of your Oracle Applications. Navigate to the Alerts page as follows: Top Tech Documents/ ERP Applications/Applications Installation and Upgrade/Alerts.

Self-Service Toolkit: You may also find information by navigating to the Self-Service Toolkit page as follows: Top Tech Documents/ ERP Applications/Applications Installation and Upgrade.

All Oracle Applications documentation is available online (HTML or PDF). Online help patches are available on MetaLink.

Documents Related to All Products

Oracle Applications User's Guide

This guide explains how to enter data, query, run reports, and navigate using the graphical user interface (GUI).

You may access this user's guide online by selecting "Getting Started with Oracle Applications" from any Oracle Applications help file.

Documents Related to This Product

Oracle Marketing Implementation Guide

Use this manual to understand the necessary configuration and implementation steps required to install Oracle Marketing.

Oracle Marketing User's Guide

This manual provides basic conceptual and reference information needed to understand the Oracle Marketing application.

Installation and System Administration

Oracle Applications Concepts

This guide provides an introduction to the concepts, features, technology stack, architecture, and terminology for Oracle Applications Release 11i. It provides a useful first book to read prior to an installation of Oracle Applications. This guide also introduces the concepts behind application-wide features such as Business Intelligence (BIS), languages and character sets, and Self-Service Web Applications.

Installing Oracle Applications

This guide provides instructions for managing the installation of Oracle Applications products. In Release 11i, much of the installation process is handled using Oracle Rapid Install, which minimizes the time to install Oracle Applications, the Oracle8 technology stack, and the Oracle8i Server technology stack by automating many of the required steps. This guide contains instructions for using Oracle Rapid Install and lists the tasks you need to perform to finish your installation. You should use this guide in conjunction with individual product user's guides and implementation guides.

Oracle Applications Supplemental CRM Installation Steps

This guide contains specific steps needed to complete installation of a few of the CRM products. The steps should be performed immediately following the steps given in the Installing Oracle Applications guide.

Upgrading Oracle Applications

Refer to this guide if you are upgrading your Oracle Applications Release 10.7 or Release 11.0 products to Release 11i. This guide describes the upgrade process and lists database and product-specific upgrade tasks. Your system must be either at Release 10.7 (NCA, SmartClient, or character mode) or Release 11.0, to upgrade to

Release 11i. Your system cannot be upgraded to Release 11i directly from releases prior to 10.7.

Maintaining Oracle Applications

Use this guide to help you run the various application development (AD) utilities, such as AutoUpgrade, AutoPatch, AD Administration, AD Controller, AD Relink, License Manager, and others. It contains how-to steps, screenshots, and other information needed to run the AD utilities. This guide also provides information on maintaining the Oracle applications file system and database.

Oracle Applications System Administrator's Guide

This guide provides planning and reference information for the Oracle Applications System Administrator. It contains information on how to define security, customize menus and online help, and manage concurrent processing.

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Applications Developer's Guide

This guide contains the coding standards followed by the Oracle Applications development staff. It describes the Oracle Application Object Library components needed to implement the Oracle Applications user interface described in the Oracle Applications User Interface Standards for Forms-Based Products. It also provides information to help you build your custom Oracle Forms Developer 6i forms so that they integrate with Oracle Applications.

Other Implementation Documentation

Oracle Workflow Guide

This guide explains how to define new workflow business processes as well as customize existing Oracle Applications-embedded workflow processes. You also use this guide to complete the setup steps necessary for any Oracle Applications product that includes workflow-enabled processes.

Oracle eTechnical Reference Manuals

Each eTechnical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your

existing applications, integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on Metalink.

Training and Support

Training

Oracle offers training courses to help you and your staff master Oracle Marketing and reach full productivity quickly. You have a choice of educational environments. You can attend courses offered by Oracle University at any one of our many Education Centers, you can arrange for our trainers to teach at your facility, or you can use Oracle Learning Network (OLN), Oracle University's online education utility. In addition, Oracle training professionals can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization's structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals provides the help and information you need to keep Oracle Marketing working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, managing an Oracle8*i* or Oracle*9i* server, and your hardware and software environment.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data. Because Oracle Applications tables are interrelated, any change you make using Oracle Applications can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. Your

tables may get out of synchronization with each other and risk retrieving erroneous information. You also risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter or modify information into database tables using database tools, these validation checks are not performed and you may store invalid information. You also lose the ability to track who has changed the information because SQL*Plus and other database tools do not keep a record of changes.

About Oracle

Oracle Corporation develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as Oracle Applications, an integrated suite of more than 160 software modules for financial management, supply chain management, manufacturing, project systems, human resources and customer relationship management.

Oracle products are available for mainframes, minicomputers, personal computers, network computers and personal digital assistants, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle is the world's leading supplier of software for information management, and the world's second largest software company. Oracle offers its database, tools, and application products, along with related consulting, education, and support services, in over 145 countries around the world.

1

Introduction

The public APIs provided by Oracle Marketing and described in this document are grouped according to functionality. The following groups of APIs are covered in this guide:

- Campaign
- Campaign Schedule
- Metric
- Claim
- Accrual
- Lead
- Event
- Event Schedule
- Event Offers
- List
- List Entries
- List Generation

Note: The words *procedure* and *API* are used interchangeably throughout this guide.

1.1 Parameter Specifications

The specifications for the public APIs provided by the Oracle CRM Application Foundation define four categories of parameters:

- Standard IN
- Standard OUT
- Procedure specific IN
- Procedure specific OUT

Standard IN and OUT parameters are specified by the Oracle Applications business object API Coding Standards, and are discussed in the following sections.

Procedure specific IN and OUT parameter are related to the API being specified, and are discussed with that individual API.

1.1.1 Standard In Parameters

The following table describes standard IN parameters which are common to all APIs provided by Oracle Marketing.

Table 1–1 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	Yes	Default = FND_API.G_FALSE If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack. If set to false the calling program must initialize the message stack. This action is required to be performed only once, even in the case where more than one API is called.

Table 1–1 Standard IN Parameters

Parameter	Data Type	Required	Description
p_commit	VARCHAR2	No	Default = FND_API.G_FALSE If set to true, the API commits before returning to the calling program. If set to false, then it is the calling program's responsibility to commit the transaction.

1.1.2 Standard OUT parameters

The following table describes standard OUT parameters, which are common to all public APIs provided by Oracle CRM Application Foundation.

Note: All standard OUT parameters are required.

Table 1–2 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

1.1.3 Parameter Size

Verify the size of the column, from the base table for that column, when passing a parameter of a specific length. For example, if you pass a NUMBER value, first

query to find the exact value to pass. An incorrect value or data type can cause the API call to fail.

1.1.4 Missing Parameter Attributes

The following table describes optional IN parameters which are initialized to pre-defined values representing missing constants. These constants are defined for the common PL/SQL data types and should be used in the initialization of the API formal parameters.

Table 1–3 Missing Parameter Attributes

Parameter	Type	Initialized Value
G_MISS_NUM	CONSTANT	NUMBER:=99.99E125
G_MISS_CHAR	CONSTANT	VARCHAR2(1):=chr(0)
G_MISS_DATE	CONSTANT	DATE:=TO_DATE('1', 'j');

These constants are defined in the package FND_API in the file fndpapis.pls. All columns in a record definition are set to the G_MISS_X constant as defined for the data type.

1.1.5 Parameter Validations

The following types of parameters are always validated during the API call:

- Standard IN
- Standard OUT
- Required procedure specific IN
- Procedure specific OUT

1.1.6 Invalid Parameters

If an API encounters an invalid parameter during the API call, then one of the following actions will occur:

- An exception will be raised.
- An error message identifying the invalid parameter will be generated.
- All API actions will be cancelled.

1.2 Version Information

It is required that every API call pass a version number for that API as its first parameter (p_api_version).

This version number must match the internal version number of that API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.

Note: The currently supported version at this time is 1.0. Use only 1.0 for the API Version Number.

In addition, the object version number must be input for all update and delete APIs.

- If the object_version_number passed by the API matches that of the object in the database, the update is completed.
- If the object_version_number passed by the API does not match that of the object in the database, an error condition is generated.

Note: It is not required that all status notifications provide a number identifier along with the message, although, in many cases, it is provided.

1.3 Status Messages

Every API must return one of the following states as parameter x_return_status after the API is called:

- S (Success)
- E (Error)
- U (Unexpected error)

Each state can be associated with a status message. The following table describes each state.

Table 1–4 Status Messages

Status	Description
S	Indicates that the API performed all the operations requested by its caller. ■ A success return status may or may not be accompanied by messages in the API message list. ■ Currently, the Oracle Marketing APIs do not provide a message for a return status of success. ***VERIFY***
E	Indicates that the API failed to perform one or more of the operations requested by its caller. An error return status is accompanied by one or more messages describing the error.
U	Indicates that the API encountered an error condition it did not expect, or could not handle, and that it is unable to continue with its regular processing. For example, certain programming errors such as attempting to divide by zero cause this type of error. These types of errors usually cannot be corrected by the user and requires a system administrator or application developer to correct.

Warning and Information Messages

In addition to these three types of possible status messages, you may also code the following additional message types:

- Warnings
- Information

To create a warning message, perform the following steps:

1. Create a global variable to be used to signal a warning condition. For example, this could be similar to the following:

   ```
   G_RET_STS_WARNING := 'W'
   ```

   This global variable is not part of the FND_API package.

2. Return this value if the warning condition is encountered. For example, using the same example as in step one, set up the following code in the API to process the warning condition:

   ```
   x_return_status := G_RET_STS_WARNING
   ```

   This code replaces the more usual:

```
x_return_status := fnd_api.g_ret_sts_unexp_error for "U"
```

3. If desired, perform a similar procedure to create Information messages.

2

User Hooks

Many application implementations require some form of customization. Some of these customizations are not intrusive into the applications, such as adding reports or adding screens with new views of the data.

Other customizations are intrusive, requiring site-specific modification to product code. Often this customization is due to the need to incorporate business rules not already implemented in the application.

2.1 Introduction

User hooks provide the client with the ability to add logic to application processing and to disable optional product processing. These User Hooks take the form of procedures that may be called by the application, in sequence, when the application takes a specified action on a specified object type.

For example, the public API to create a campaign is comprised of the following procedures:

- AMS_Campaign_CUHK.Create_campaign_pre (User Hook Procedure)
- The private create campaign API (application internal execution code)
- AMS_Campaign_CUHK.Create_campaign_post (User Hook Procedure)

The parameter list of the User Hook Procedures is fixed. These parameters are noted for each API published in this guide and are the same for both the pre- and post- User Hook Procedures.

User Hooks will be available in the PL/SQL public APIs (entity or process) and will be implemented for create, update, delete, and validate procedures. User Hooks may not be required for lock procedures.

User Hook Procedures are named in the following manner (examples in parentheses). The application code (AMS) followed by the application object (Campaign) and the suffix of CUHK, which is an abbreviation of Customer User Hook. To the right of the period is the action (Create), the application object (Campaign) and an indicator of whether the procedure is called before (pre) or after (post) application action is taken.

For example: AMS_Campaign_CUHK.Create_Campaign_Pre is a procedure which is called before a campaign is created.

2.2 User Hook Registration

User Hooks are registered in the JTF_USER_HOOKS table with an execution flag set by default to "N".

If a User Hook Procedure is modified, the execute flag in the table must be set to "Y" in order for the User Hook Procedure to be executed.

Steps

1. Identify the row, in the JTF_USER_HOOKS table, for the User Hook to be customized.
2. Update the execute flag to "Y".

2.3 Available Campaign User Hooks

The following table lists Campaign and Campaign Schedule User Hook Procedures available in the Oracle Marketing API.

Table 2–1 User Hook Procedures

API Name	User Hook Procedure Name
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Create_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Create_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Delete_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Delete_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Lock_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Lock_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Update_campaign_Pre

Table 2–1 User Hook Procedures

API Name	User Hook Procedure Name
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Update_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Validate_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Validate_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Create_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Create_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Delete_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Delete_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Lock_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Lock_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Update_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Update_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Validate_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_VUHK.Validate_campaign_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_Schedule_CUHK.Create_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Create_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Delete_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Delete_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Lock_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Lock_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Update_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Update_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Validate_camp_schedule_Pre

Table 2–1 User Hook Procedures

API Name	User Hook Procedure Name
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Validate_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_Schedule_VUHK.Create_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Create_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Delete_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Delete_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Lock_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Lock_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Update_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Update_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Validate_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_VUHK.Validate_camp_schedule_Post

2.4 Available Event User Hooks

The following table lists Event and Event Schedule User Hook Procedures available in the Oracle Marketing API.

Table 2–2 User Hook Procedures

API Name	User Hook Procedure Name
AMS_EVENTHEADER_ PUB	AMS_EventHeader_CUHK.Create_EventHeader_Pre
AMS_EVENTHEADER_ PUB	AMS_EventHeader_CUHK.Create_EventHeader_Post

Table 2–2 User Hook Procedures

API Name	User Hook Procedure Name
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Delete_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Delete_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Lock_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Lock_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Update_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Update_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Validate_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Validate_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Create_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Create_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Delete_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Delete_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Lock_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Lock_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Update_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Update_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Validate_EventHeader_Pre

Table 2–2 User Hook Procedures

API Name	User Hook Procedure Name
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Validate_EventHeader_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Create_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Create_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Delete_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Delete_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Lock_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Lock_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Update_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Update_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Validate_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Validate_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Create_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Create_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Delete_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Delete_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Lock_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Lock_EventOffer_Post

Table 2–2 User Hook Procedures

API Name	User Hook Procedure Name
AMS_EVENTOFFER_ PUB	AMS_EventOffer_VUHK.Update_EventOffer_Pre
AMS_EVENTOFFER_ PUB	AMS_EventOffer_VUHK.Update_EventOffer_Post
AMS_EVENTOFFER_ PUB	AMS_EventOffer_VUHK.Validate_EventOffer_Pre
AMS_EVENTOFFER_ PUB	AMS_EventOffer_VUHK.Validate_EventOffer_Post
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.create_EventHeader_pre
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.create_EventHeader_post
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.delete_EventHeader_pre
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.delete_EventHeader_post
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.lock_EventHeader_pre
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.lock_EventHeader_post
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.update_EventHeader_pre
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.update_EventHeader_post
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.validate_EventHeader_pre
AMS_EventHeader_PUB	AMS_EventHeader_CUHK.validate_EventHeader_post
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.create_EventOffer_pre
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.create_EventOffer_post
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.delete_EventOffer_pre
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.delete_EventOffer_post
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.lock_EventOffer_pre
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.lock_EventOffer_post
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.update_EventOffer_pre
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.update_EventOffer_post
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.validate_EventOffer_pre
AMS_EventOffer_PUB	AMS_EventOffer_CUHK.validate_EventOffer_post
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.register_pre

Table 2–2 User Hook Procedures

API Name	User Hook Procedure Name
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.register_post
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Update_registration_pre
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Update_registration_post
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Delete_Registration_pre
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Delete_Registration_post
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Prioritize_reg_wailist_pre
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Prioritize_reg_wailist_post
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Substitute_enrollee_pre
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Substitute_enrollee_post
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Transfer_enrollee_pre
AMS_EvtRegs_PUB	AMS_EvtRegs_CUHK.Transfer_enrollee_post

2.5 Parameters for User Hook Procedures

There are two parameters for the User Hook Procedures:

- IN Parameter: The record type for the object, such as campaign or campaign schedule.
- OUT Parameter: A return status.

3

Campaign

The APIs for campaign provide a number of procedures for general campaign actions.

The procedures which make up the Campaign APIs are:

Table 3–1 Campaign APIs

Procedure	Description
Create Campaign	Creates a new campaign in which (a) the object version is set to one, (b) a unique campaign ID will be created if a unique campaign ID is not passed in, and (c) a flag column will be set to Y or N, depending on existence of optional parameters.
Delete Campaign	Sets a campaign to inactive rather than removing it from the database. Will raise an exception if the object version doesn't match the database record.
Lock Campaign	Locks the given campaign record. Will raise an exception if the object version doesn't match the database record.
Update Campaign	Updates the campaign record. The values which are not changed can be passed as g_miss record and will not be updated. Will raise an exception if the object version doesn't match the database record.
Validate Campaign	Validate different business rules like checking not null columns, valid flag values, and foreign key validation. In addition, it also does other business validation. The p_camp_rec parameter should be the complete campaign record.

3.1 User Hook Procedures

The User Hook Procedures available for Campaigns are:

Table 3–2 User Hook Procedures

API Name	User Hook Procedure Name
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Create_campaign_pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Create_campaign_post
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Delete_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Delete_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Lock_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Lock_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Update_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Update_campaign_Post
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Validate_campaign_Pre
AMS_CAMPAIGNS_PUB	AMS_Campaign_CUHK.Validate_campaign_Post

3.2 Type Declaration

This section defines the campaign record type declaration. Campaign record type is used as an IN parameter in some of the procedures for creation or updating. The actual definition of the record type resides in a private api. Hence the record type is referred to AMS_Campaign_PVT.camp_rec_type.

```
TYPE camp_rec_type IS RECORD(
    campaign_id                NUMBER,
    last_update_date           DATE,
    last_updated_by            NUMBER,
    creation_date              DATE,
    created_by                 NUMBER,
    last_update_login          NUMBER,
    object_version_number      NUMBER,
    custom_setup_id            NUMBER,
    owner_user_id              NUMBER,
    user_status_id             NUMBER,
    status_code                VARCHAR2(30),
    status_date                DATE,
    active_flag                VARCHAR2(1),
    private_flag               VARCHAR2(1),
    partner_flag               VARCHAR2(1),
    template_flag              VARCHAR2(1),
    cascade_source_code_flag   VARCHAR2(1),
```

```
inherit_attributes_flag    VARCHAR2(1),
source_code                VARCHAR2(30),
rollup_type                VARCHAR2(30),
campaign_type              VARCHAR2(30),
media_type_code            VARCHAR2(30),
priority                   VARCHAR2(30),
fund_source_type           VARCHAR2(30),
fund_source_id             NUMBER,
parent_campaign_id         NUMBER,
application_id             NUMBER,
qp_list_header_id          NUMBER,
media_id                   NUMBER,
channel_id                 NUMBER,
event_type                 VARCHAR2(30),
arc_channel_from           VARCHAR2(30),
dscript_name               VARCHAR2(256),
transaction_currency_code  VARCHAR2(15),
functional_currency_code   VARCHAR2(15),
budget_amount_tc           NUMBER,
budget_amount_fc           NUMBER,
forecasted_plan_start_date DATE,
forecasted_plan_end_date   DATE,
forecasted_exec_start_date DATE,
forecasted_exec_end_date   DATE,
actual_plan_start_date     DATE,
actual_plan_end_date       DATE,
actual_exec_start_date     DATE,
actual_exec_end_date       DATE,
inbound_url                VARCHAR2(120),
inbound_email_id           VARCHAR2(120),
inbound_phone_no           VARCHAR2(25),
duration                   NUMBER,
duration_uom_code          VARCHAR2(3),
ff_priority                VARCHAR2(30),
ff_override_cover_letter   NUMBER,
ff_shipping_method         VARCHAR2(30),
ff_carrier                 VARCHAR2(120),
content_source             VARCHAR2(120),
cc_call_strategy           VARCHAR2(30),
cc_manager_user_id         NUMBER,
forecasted_revenue         NUMBER,
actual_revenue             NUMBER,
forecasted_cost            NUMBER,
actual_cost                NUMBER,
forecasted_response        NUMBER,
```

```
actual_response            NUMBER,
target_response            NUMBER,
country_code               VARCHAR2(30),
language_code              VARCHAR2(30),
attribute_category         VARCHAR2(30),
attribute1                 VARCHAR2(150),
attribute2                 VARCHAR2(150),
attribute3                 VARCHAR2(150),
attribute4                 VARCHAR2(150),
attribute5                 VARCHAR2(150),
attribute6                 VARCHAR2(150),
attribute7                 VARCHAR2(150),
attribute8                 VARCHAR2(150),
attribute9                 VARCHAR2(150),
attribute10                VARCHAR2(150),
attribute11                VARCHAR2(150),
attribute12                VARCHAR2(150),
attribute13                VARCHAR2(150),
attribute14                VARCHAR2(150),
attribute15                VARCHAR2(150),
campaign_name              VARCHAR2(240),
campaign_theme             VARCHAR2(4000),
description                VARCHAR2(4000),
version_no                 NUMBER,
campaign_calendar          VARCHAR2(15),
start_period_name          VARCHAR2(15),
city_id                    NUMBER,
global_flag                VARCHAR2(1),
show_campaign_flag         VARCHAR2(1),
business_unit_id           NUMBER,
accounts_closed_flag       VARCHAR2(1),
task_id                    NUMBER,
related_event_from         VARCHAR2(30),
related_event_id           NUMBER,
program_attribute_category VARCHAR2(30),
program_attribute1         VARCHAR2(150),
program_attribute2         VARCHAR2(150),
program_attribute3         VARCHAR2(150),
program_attribute4         VARCHAR2(150),
program_attribute5         VARCHAR2(150),
program_attribute6         VARCHAR2(150),
program_attribute7         VARCHAR2(150),
program_attribute8         VARCHAR2(150),
program_attribute9         VARCHAR2(150),
program_attribute10        VARCHAR2(150),
```

```
    program_attribute11         VARCHAR2(150),
    program_attribute12         VARCHAR2(150),
    program_attribute13         VARCHAR2(150),
    program_attribute14         VARCHAR2(150),
    program_attribute15         VARCHAR2(150)
);
```

3.3 Standard Parameters for Campaign APIs

There are a number of standard parameters which are common for all of the following campaign APIs. Note that all the Standard OUT parameters are required. The parameters are listed in the tables below:

Table 3–3 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Default = FND_API.G_FALSE If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack. If set to false the calling program must initialize the message stack. This action is required to be performed only once, even in the case where more than one API is called.

Table 3–4 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

3.4 Campaign APIs

3.4.1 Create Campaign

This procedure creates a campaign with the supplied campaign ID, if it is unique, or if the ID is not supplied, a unique ID will be created.

Procedure Specification

```
PROCEDURE create_campaign(
    p_api_version        IN      NUMBER,
    p_init_msg_list      IN      VARCHAR2:= FND_API.g_false,
    p_commit             IN      VARCHAR2:= FND_API.g_false,
    p_validation_level   IN      NUMBER:= FND_API.g_valid_level_full,
    x_return_status      OUT     VARCHAR2,
    x_msg_count          OUT     NUMBER,
    x_msg_data           OUT     VARCHAR2,
    p_camp_rec           IN      AMS_Campaign_PVT.camp_rec_type,
    x_camp_id            OUT     NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Object_version_number will be set to 1.
2. If campaign_id is passed in, the uniqueness will be checked. An exception will be raised in case of duplicates.
3. If campaign_id is not passed in, a unique one will be generated from the sequence.
4. If a flag column is passed in, check if it is 'Y' or 'N'. An exception will be raised for invalid flag.
5. If a flag column is not passed in, each field will be defaulted to 'Y' or 'N' as appropriate.
6. Please don't pass in any FND_API.g_miss_char/num/date.

Table 3–5 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_validation_ level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_camp_rec	AMS_ Campaign_ PVT.camp_ rec_type	Yes	Record for the campaign. The record will be validated before creation of the campaign.

Table 3–6 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.
x_campaign_id	NUMBER	Id of the newly created campaign.

3.4.2 Delete Campaign

This procedure makes a campaign inactive rather than deleting it. It locates the campaign by the campaign ID and the object version number.

Procedure Specification

```
PROCEDURE delete_campaign(
    p_api_version        IN      NUMBER,
    p_init_msg_list      IN      VARCHAR2:= FND_API.g_false,
    p_commit             IN      VARCHAR2:= FND_API.g_false,

    x_return_status      OUT     VARCHAR2,
    x_msg_count          OUT     NUMBER,
    x_msg_data           OUT     VARCHAR2,

    p_camp_id            IN      NUMBER,
    p_object_version     IN      NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. Will set the campaign to be inactive, instead of removing it from the database.

Table 3–7 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_camp_id	NUMBER	Yes	Campaign ID.
p_object_ version_ number	NUMBER	Yes	Object version number of the campaign to be deleted. Based on the campaign ID and the object version number, the campaign record will be located and made inactive.

Table 3–8 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 3–8 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

3.4.3 Lock Campaign

This procedure locks the campaign record based on the campaign ID and the object version number passed. The API will raise an exception if the record matching the campaign ID and the object version number does not exist.

Procedure Specification

```
PROCEDURE lock_campaign(
    p_api_version        IN      NUMBER,
    p_init_msg_list      IN      VARCHAR2:= FND_API.g_false
    x_return_status      OUT     VARCHAR2,
    x_msg_count          OUT     NUMBER,
    x_msg_data           OUT     VARCHAR2,
    p_camp_id            IN      NUMBER,
    p_object_version     IN      NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.

Table 3–9 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.

Table 3–9 IN Parameters

Parameter	Data Type	Required	Description
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_camp_id	NUMBER	Yes	Campaign ID.
p_object_ version_ number	NUMBER	Yes	Object version number of the campaign to be locked. Based on the campaign ID and the object version number, the campaign record will be located and locked.

Table 3–10 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

3.4.4 Update Campaign

This procedure updates a campaign record based on the campaign ID and object version number. The record type for campaign can be initialized by g_miss rec and can be overridden by the values which are changed. For update the campaign ID and object version number are required fields in the record type. When the update is called, all the g_miss values are replaced with those of the database. When the record is updated, the object version number is incremented by 1.

Procedure Specification

```
PROCEDURE update_campaign(
```

```
    p_api_version        IN      NUMBER,
    p_init_msg_list      IN      VARCHAR2:=FND_API.g_false,
    p_commit             IN      VARCHAR2:=FND_API.g_false,
    p_validation_level   IN      NUMBER:=FND_API.g_valid_level_full,
    x_return_status      OUT     VARCHAR2,
    x_msg_count          OUT     NUMBER,
    x_msg_data           OUT     VARCHAR2,
    p_camp_rec           IN      AMS_Campaign_PVT.camp_rec_type
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. If an attribute is passed in as FND_API.g_miss_char/num/date, that column won't be updated.

Table 3–11 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_camp_rec	AMS_Campaign_PVT.camp_rec_type	Yes	Record for the campaign. The record will be validated before updating the campaign.

Table 3–12 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

3.4.5 Validate Campaign

This procedure validates a campaign record. This API will be called internally by the Create Campaign API to validate the data and the business rules.

Procedure Specification

```
PROCEDURE validate_campaign(
    p_api_version        IN      NUMBER,
    p_init_msg_list      IN      VARCHAR2:=FND_API.g_false,
    p_validation_level   IN      NUMBER:=FND_API.g_valid_level_full,
    x_return_status      OUT     VARCHAR2,
    x_msg_count          OUT     NUMBER,
    x_msg_data           OUT     VARCHAR2,
    p_camp_rec           IN      AMS_Campaign_PVT.camp_rec_type
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Oracle recommends that the p_camp_rec be the complete campaign record.

Table 3–13 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_camp_rec	AMS_ Campaign_ PVT.camp_ rec_type	Yes	Record for the campaign. The record will be validated before validation of the campaign.

4

Campaign Schedule

The APIs for campaign schedule provide a number of procedures for campaign schedule actions.

The procedures which make up the Campaign Schedule APIs are:

Table 4–1 Campaign Schedule APIs

Procedure	Description
Create Campaign Schedule	Creates a new campaign schedule in which (a) the object version is set to one, (b) a unique schedule ID will be created if a unique schedule ID is not passed in, and (c) a flag column will be set to Y or N, depending on existence of optional parameters.
Delete Campaign Schedule	If the schedule is of status New, the schedule is deleted. If the status is not New, the schedule is set to status Inactive, rather than removing it from the database. Will raise an exception if the object version doesn't match the database record.
Lock Campaign Schedule	Locks the given schedule record. Will raise an exception if the object version doesn't match the database record.
Update Campaign Schedule	Updates the schedule record. The values which are not changed can be passed as g_miss record and will not be updated. Will raise an exception if the object version doesn't match the database record.
Validate Campaign Schedule	Validate different business rules like checking not null columns, valid flag values, and foreign key validation. In addition it also do other business validation. The p_schedule_rec parameter should be the complete campaign schedule record.

Table 4–1 Campaign Schedule APIs

Procedure	Description
Copy Campaign Schedule	Copies the campaign schedule. When the schedule gets copied, the side navigation menu attributes selected by the user, get copied to the new schedule. The list of attributes available for the user to copy are determined by custom setup.

4.1 User Hook Procedures

The User Hook Procedures available for Campaign Schedules are:

Table 4–2 User Hook Procedures

API Name	User Hook Procedure Name
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_Schedule_CUHK.Create_camp_schedule_pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_Schedule_CUHK.Create_camp_schedule_post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Delete_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Delete_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Lock_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Lock_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Update_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Update_camp_schedule_Post
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Validate_camp_schedule_Pre
AMS_CAMP_ SCHEDULE_PUB	AMS_Camp_schedule_CUHK.Validate_camp_schedule_Post

4.2 Type Declaration

This section defines the campaign schedule record type declaration. Campaign Schedule record type is used as an IN parameter in some of the procedures for creation or updating.

```
TYPE schedule_rec_type IS RECORD(
    schedule_id                 NUMBER := FND_API.G_MISS_NUM,
    last_update_date            DATE := FND_API.G_MISS_DATE,
    last_updated_by             NUMBER := FND_API.G_MISS_NUM,
    creation_date               DATE := FND_API.G_MISS_DATE,
    created_by                  NUMBER := FND_API.G_MISS_NUM,
    last_update_login           NUMBER := FND_API.G_MISS_NUM,
    object_version_number       NUMBER := FND_API.G_MISS_NUM,
    campaign_id                 NUMBER := FND_API.G_MISS_NUM,
    user_status_id              NUMBER := FND_API.G_MISS_NUM,
    status_code                 VARCHAR2(30) := FND_API.G_MISS_CHAR,
    status_date                 DATE := FND_API.G_MISS_DATE,
    source_code                 VARCHAR2(30) := FND_API.G_MISS_CHAR,
    use_parent_code_flag        VARCHAR2(1) := FND_API.G_MISS_CHAR,
    start_date_time             DATE := FND_API.G_MISS_DATE,
    timezone_id                 NUMBER := FND_API.G_MISS_NUM,
    activity_type_code          VARCHAR2(30) := FND_API.G_MISS_CHAR,
    activity_id                 NUMBER := FND_API.G_MISS_NUM,
    arc_marketing_medium_from   VARCHAR2(30) := FND_API.G_MISS_CHAR,
    marketing_medium_id         NUMBER := FND_API.G_MISS_NUM,
    custom_setup_id             NUMBER := FND_API.G_MISS_NUM,
    triggerable_flag            VARCHAR2(1) := FND_API.G_MISS_CHAR,
    trigger_id                  NUMBER := FND_API.G_MISS_NUM,
    notify_user_id              NUMBER := FND_API.G_MISS_NUM,
    approver_user_id            NUMBER := FND_API.G_MISS_NUM,
    owner_user_id               NUMBER := FND_API.G_MISS_NUM,
    active_flag                 VARCHAR2(1) := FND_API.G_MISS_CHAR,
    cover_letter_id             NUMBER := FND_API.G_MISS_NUM,
    reply_to_mail               VARCHAR2(120) := FND_API.G_MISS_CHAR,
    mail_sender_name            VARCHAR2(120) := FND_API.G_MISS_CHAR,
    mail_subject                VARCHAR2(240) := FND_API.G_MISS_CHAR,
    from_fax_no                 VARCHAR2(25) := FND_API.G_MISS_CHAR,
    accounts_closed_flag        VARCHAR2(1) := FND_API.G_MISS_CHAR,
    org_id                      NUMBER := FND_API.G_MISS_NUM,
    objective_code              VARCHAR2(30) := FND_API.G_MISS_CHAR,
    country_id                  NUMBER := FND_API.G_MISS_NUM,
    campaign_calendar           VARCHAR2(20) := FND_API.G_MISS_CHAR,
    start_period_name           VARCHAR2(15) := FND_API.G_MISS_CHAR,
    priority                    VARCHAR2(30) := FND_API.G_MISS_CHAR,
```

```
workflow_item_key            VARCHAR2(240) := FND_API.G_MISS_CHAR,
transaction_currency_code    VARCHAR2(15) := FND_API.G_MISS_CHAR,
functional_currency_code     VARCHAR2(15) := FND_API.G_MISS_CHAR,
budget_amount_tc             NUMBER := FND_API.G_MISS_NUM,
budget_amount_fc             NUMBER := FND_API.G_MISS_NUM,
language_code                VARCHAR2(4) := FND_API.G_MISS_CHAR,
task_id                      NUMBER := FND_API.G_MISS_NUM,
related_event_from           VARCHAR2(30) := FND_API.G_MISS_CHAR,
related_event_id             NUMBER := FND_API.G_MISS_NUM,
attribute_category           VARCHAR2(30) := FND_API.G_MISS_CHAR,
attribute1                   VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute2                   VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute3                   VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute4                   VARCHAR2(240) := FND_API.G_MISS_CHAR,
attribute5                   VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute6                   VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute7                   VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute8                   VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute9                   VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute10                  VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute11                  VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute12                  VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute13                  VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute14                  VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute15                  VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute_category  VARCHAR2(240) := FND_API.G_MISS_CHAR,
activity_attribute1          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute2          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute3          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute4          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute5          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute6          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute7          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute8          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute9          VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute10         VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute11         VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute12         VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute13         VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute14         VARCHAR2(150) := FND_API.G_MISS_CHAR,
activity_attribute15         VARCHAR2(150) := FND_API.G_MISS_CHAR,
schedule_name                VARCHAR2(120) := FND_API.G_MISS_CHAR,
description                  VARCHAR2(4000):= FND_API.G_MISS_CHAR,
related_source_code          VARCHAR2(30) := FND_API.G_MISS_CHAR,
related_source_object        VARCHAR2(30) := FND_API.G_MISS_CHAR,
```

```
    related_source_id           NUMBER := FND_API.G_MISS_NUM,
    query_id                    NUMBER := FND_API.G_MISS_NUM,
    include_content_flag        VARCHAR2(1) := FND_API.G_MISS_CHAR,
    content_type                VARCHAR2(30) := FND_API.G_MISS_CHAR,
    test_email_address          VARCHAR2(250):= FND_API.G_MISS_CHAR,
    greeting_text               VARCHAR2(4000):= FND_API.G_MISS_CHAR,
    footer_text                 VARCHAR2(4000):= FND_API.G_MISS_CHAR
);
```

4.3 Standard Parameters for Campaign Schedule APIs

There are a number of standard parameters which are common for all of the following Campaign Schedule APIs. Note that all the Standard OUT parameters are required. The parameters are listed in the tables below:

Table 4–3 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Default = FND_API.G_FALSE If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack. If set to false the calling program must initialize the message stack. This action is required to be performed only once, even in the case where more than one API is called.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_validation_ level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.

Table 4–4 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

4.4 Campaign Schedule APIs

4.4.1 Create Campaign Schedule

This procedure creates a campaign schedule with the supplied campaign schedule ID, if it is unique, or if the ID is not supplied, a unique ID will be created.

Notes

The source code can be copied from the parent (Campaign) or user can enter the unique source code for the schedule. If neither of the above are provided, the system will create a unique source code.

Procedure Specification

```
PROCEDURE Create_Camp_Schedule(
    p_api_version_number IN      NUMBER,
    p_init_msg_list      IN      VARCHAR2 := FND_API.G_FALSE,
    p_commit             IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level   IN      NUMBER := FND_API.g_valid_level_full,
    x_return_status      OUT     VARCHAR2,
    x_msg_count          OUT     NUMBER,
    x_msg_data           OUT     VARCHAR2,
    p_schedule_rec       IN      schedule_rec_type := g_miss_schedule_rec,
```

```
    x_schedule_id        OUT     NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

The source code can be copied from the parent campaign or the user can enter a unique source code for the schedule. If neither of the above are provided, the system will create a unique source code.

Table 4–5 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_validation_ level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_schedule_rec	AMS_ Campaign_ PVT.camp_ rec_type	Yes	Record for the schedule. The record will be validated before creation of the schedule.

Table 4–6 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

4.4.2 Delete Campaign Schedule

Deletes the campaign schedule if the schedule status is New. If the status is not New, the schedule is marked inactive. The schedule ID and object version number will be used to deactivate the schedule.

Notes

If the schedule has not copied the source code from the parent campaign, the source code will be revoked.

Procedure Specification

```
PROCEDURE Delete_Camp_Schedule(
    p_api_version_number      IN      NUMBER,
    p_init_msg_list           IN      VARCHAR2 := FND_API.G_FALSE,
    p_commit                  IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level        IN      NUMBER := FND_API.g_valid_level_full,
    x_return_status           OUT     VARCHAR2,
    x_msg_count               OUT     NUMBER,
    x_msg_data                OUT     VARCHAR2,
    p_schedule_id             IN      NUMBER,
    p_object_version_number   IN      NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Table 4–7 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_validation_ level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_schedule_id	NUMBER	Yes	Schedule ID.
p_object_ version_ number	NUMBER	Yes	Object version number of the schedule.

Table 4–8 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 4–8 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

4.4.3 Lock Campaign Schedule

This procedure locks the campaign schedule record based on the schedule ID and the object version number. The API will raise an exception if the record matching the schedule ID and object version number does not exist.

Procedure Specification

```
PROCEDURE Lock_Camp_Schedule(
    p_api_version_number      IN      NUMBER,
    p_init_msg_list           IN      VARCHAR2 := FND_API.G_FALSE,
    x_return_status           OUT     VARCHAR2,
    x_msg_count               OUT     NUMBER,
    x_msg_data                OUT     VARCHAR2,
    p_schedule_id             IN      NUMBER,
    p_object_version          IN      NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Table 4–9 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.

Table 4–9 IN Parameters

Parameter	Data Type	Required	Description
p_schedule_rec	AMS_ Campaign_ PVT.camp_ rec_type	Yes	Record for the schedule. The record will be validated before locking the schedule.

Table 4–10 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

4.4.4 Update Campaign Schedule

This procedure updates a campaign schedule record based on the schedule ID and object version number. The record type for campaign schedule can be initialized by g_miss_rec and can be overridden by the values which are changed. For update, the campaign schedule ID and object version number are required fields in the record type. When the update is called, all the g_miss values are replaced with those of the database. When the record is updated, the object number version is incremented by 1.

Procedure Specification

```
PROCEDURE Update_Camp_Schedule(
    p_api_version_number      IN      NUMBER,
    p_init_msg_list           IN      VARCHAR2 := FND_API.G_FALSE,
    p_commit                  IN      VARCHAR2 := FND_API.G_FALSE,
```

```
    p_validation_level        IN     NUMBER := FND_API.g_valid_level_full,
    x_return_status           OUT    VARCHAR2,
    x_msg_count               OUT    NUMBER,
    x_msg_data                OUT    VARCHAR2,
    p_schedule_rec            IN     schedule_rec_type,
    x_object_version_number   OUT    NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Table 4–11 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_schedule_rec	schedule_rec_type	Yes	Record type for the schedule. Schedule ID and object version number are required in the record type.

Table 4–12 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

4.4.5 Validate Campaign Schedule

This procedure validates a campaign schedule record. This API is called internally by the Create Campaign Schedule API to validate the data and the business rules.

Procedure Specification

```
PROCEDURE Validate_Camp_Schedule(
    p_api_version             IN      NUMBER,
    p_init_msg_list           IN      VARCHAR2 := FND_API.g_false,
    p_validation_level        IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    x_return_status           OUT     VARCHAR2,
    x_msg_count               OUT     NUMBER,
    x_msg_data                OUT     VARCHAR2,
    p_validation_mode         IN      VARCHAR2,
    p_schedule_rec            IN      schedule_rec_type
);
```

Current Version

1.0

Parameter Descriptions

Table 4–13 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. NONE means no validation will be done in the API and FULL means all the validations (item level and record level) will be performed.
p_schedule_rec	schedule_ rec_type	Yes	Record type for the schedule.

Table 4–14 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

4.4.6 Copy Campaign Schedule

This API is used to copy the campaign schedules. This API copies the source schedule details and attributes to a new schedule.

Procedure Specification

```
PROCEDURE Copy_Camp_Schedule(
    p_api_version             IN      NUMBER,
    p_init_msg_list           IN      VARCHAR2 := FND_API.G_FALSE,
    p_commit                  IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level        IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    x_return_status           OUT     VARCHAR2,
    x_msg_count               OUT     NUMBER,
    x_msg_data                OUT     VARCHAR2,
    p_source_object_id        IN      NUMBER,
    p_attributes_table        IN      AMS_CpyUtility_PVT.copy_
                                      attributes_table_type,

    p_copy_columns_table      IN      AMS_CpyUtility_PVT.copy_
                                      columns_table_type,

    x_new_object_id           OUT     NUMBER,
    x_custom_setup_id         OUT     NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Table 4–15 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of Validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.

Table 4–15 IN Parameters

Parameter	Data Type	Required	Description
p_attributes_ table	(see description)	Yes	AMS_CpyUtility_PVT.copy_attributes_ table_type
p_copy_ columns_table	(see description)	Yes	AMS_CpyUtility_PVT.copy_columns_ table_type

Table 4–16 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Return status of the API. Can be "S" for Success, "E" for Expected Error, or "U" for Undefined Exception.
x_msg_count	NUMBER	Number of error messages returned by the API. If the error message returned is one then the message count will be zero.
x_msg_data	VARCHAR2	Error message returned by the API. If the messages returned are more than one, this parameter will be null and the messages have to be extracted from the message stack.
x_new_object_ id	NUMBER	New schedule ID.
x_custom_ setup_id	NUMBER	New custom setup ID.

5

Public Metrics

The public APIs for Metrics enables you to create, update and delete metric definitions.

5.1 Package Name: AMS Metric Pub

See the following table for procedures on this API.

Table 5–1 Procedure: Create_Metric

Parameter	Type	Description	Value
P_api_version	IN NUMBER	API Version Number	1
P_init_msg_list	IN VARCHAR2	Boolean to initialize the message buffer	FND_API.G_FALSE FND_API.G_TRUE
P_commit	IN VARCHAR2	Boolean to indicate commit, default is false	FND_API.G_FALSE FND_API.G_TRUE
P_validation_level	IN NUMBER	Indicates validation level	FND_API.G_VALIDATE_LEVEL_FULL
X_return_status	OUT VARCHAR2	Indicates success or failure of process	FND_API.G_RET_STS_SUCCESS FND_API.G_RET_STS_UNEXP_ERROR FND_API.G_RET_STS_ERROR
x_msg_count	OUT NOCOPY NUMBER	Number of messages returned	Integer

Table 5–1 Procedure: Create_Metric

Parameter	Type	Description	Value
X_msg_data	OUT NOCOPY VARCHAR2	Messages	Message text.
P_metric_rec	IN AMS_METRIC_ PVT. Metric_rec_type	Metric record.	See below
X_metric_id	OUT NOCOPY NUMBER	New Metric Id	Integer

5.1.1 Procedure: Update_Metric

Updating is restricted for metrics assigned to objects. Only metric_name, description, and enabled_flag may be modified.

Table 5–2 Procedure: Update_Metric

Parameters	Type	Description	Values
P_api_version	IN NUMBER	API Version number	1
P_init_msg_list	IN VARCHAR2	Boolean to initialize the message buffer	FND_API.G_FALSE FND_API.G_TRUE
P_commit	IN VARCHAR2	Boolean to indicate commit, default is false	FND_API.G_FALSE FND_API.G_TRUE
P_validation_level	IN NUMBER	Indicates validation level	FND_API.G_VALIDATE_LEVEL_ FULL
X_return_status	OUT VARCHAR2	Indicates success or failure of process	FND_API.G_RET_STS_SUCCESS FND_API.G_RET_STS_UNEXP_ ERROR FND_API.G_RET_STS_ERROR

Table 5–2 Procedure: Update_Metric

x_msg_count	OUT NOCOPY NUMBER	Number of messages returned	Integer
X_msg_data	OUT NOCOPY VARCHAR2	Messages	Message text.
P_metric_rec	IN AMS_METRIC_ PVT. Metric_rec_type	Metric record.	See below

5.1.2 Procedure: Delete_Metric

Delete is restricted if the metric is assigned to an object or it is the parent or summary of any other metric.

Table 5–3 Procedure: Delete_Metric

Parameters	Type	Description	Values
P_api_version	IN NUMBER	API Version number	1
P_init_msg_list	IN VARCHAR2	Boolean to initialize the message buffer	FND_API.G_FALSE FND_API.G_TRUE
P_commit	IN VARCHAR2	Boolean to indicate commit, default is false	FND_API.G_FALSE FND_API.G_TRUE
X_return_status	OUT VARCHAR2	Indicates success or failure of process	FND_API.G_RET_STS_SUCCESS FND_API.G_RET_STS_UNEXP_ ERROR FND_API.G_RET_STS_ERROR
x_msg_count	OUT NOCOPY NUMBER	Number of messages returned	Integer
X_msg_data	OUT NOCOPY VARCHAR2	Messages	Message text.
P_metric_id	IN NUMBER	Metric_id	Valid metric_id
P_object_version_ number	IN NUMBER	Object version number	The current object version number of the metric.

5.1.3 Record: AMS_METRIC_PVT.metric_rec_type

See the following table for AMS_Metric_PVT.metric_rec_type parameters.

Table 5–4 Record: AMS_Metric_PVT.metric_rec_type

Element	Type	Description	Values	Requirements
metric_id	NUMBER	Unique identifier for the metric	Integer	Required for update
last_update_date	DATE	Date last updated	Date	Not required
Last_updated_by	NUMBER	User last updated	User id	Not required
Creation_date	DATE	Date first created	Date	Not required
Created_by	NUMBER	User who created	User id	Not required
Last_update_login	NUMBER	User last updated	User id	Not required
Object_version_ number	NUMBER	Incremental version number	Integer	Required for update. Must correspond to current value.
Application_id	NUMBER	Application the created this object	Integer, OMO=530	Set to 530 for creation. Not required for update and delete.
Arc_metric_used_ for_object	VARCHAR2(30)	Object type: From AMS_LOOKUP, lookup_type= 'AMS_ SYS_ARC_ QUALIFIER' and ANY for ROLLUP and SUMMARY metrics	RCAM, CAMP, CSCH, EVEH, EVEO, EONE, DELV, etc.	Required for create and update.
Metric_calculation_ type	VARCHAR2(30)	Calculation type	MANUAL, FUNCITON, ROLLUP, SUMMARY	Required for create and update
Metric_category	NUMBER	Main category from AMS_CATEGORIES_ VL, arc_category_ user_for='METR' and parent_category_id is null	901=Cost, 902=Revenue, etc.	Required for create and update
Accrual_type	VARCHAR2(30)	Fixed or Variable.	'FIXED', 'VARIABLE'	Required for create and update

Table 5–4 Record: AMS_Metric_PVT.metric_rec_type

Value_type	VARCHAR2(30)	Number or Ratio	'N', 'R'	Required for create and update
Sensitive_data_flag	VARCHAR2(1)	Unused	'Y', 'N'	
Enabled_flag	VARCHAR2(1)	Indicates enabled, disabled	'Y', 'N'	Required for create and update
Metric_sub_category	NUMBER	Sub category from AMS_CATEGORIES_VL, arc_category_user_for='METR' and parent_category_id is not null	750= Deliverable Cost, etc.	Optional for create and update
Function_name	VARCHAR2(4000)	Executable PL/SQL function	Valid PL/SQL Procedure or function.	Required for FUNCTION type metrics.
Metric_parent_id	NUMBER	Rollup to metric parent	Valid ROLLUP type metric id.	Optoinal
Summary_metric_id	NUMBER	Summarize to metric	Valid SUMMARY type metric id	Optional
compute_using_function	VARCHAR2(4000)	For VARIABLE accrual type, contains metric ID of multiplier	Valid metric id	Required for accrual type=VARIABLE
default_uom_code	VARCHAR2(3)	Unused		
uom_type	VARCHAR2(10)	Unused		
formula	VARCHAR2(4000)	Unused		
metrics_name	VARCHAR2(120)	Display name	Any valid string	Required
Description	VARCHAR2(4000)	Additional description	Any valid string	Optional
Hierarchy_id	NUMBER	Unused		
Set_function_name	VARCHAR2(4000)	Unused		

5.2 Package Name: AMS_ACTMETRIC_PUB

Use this API to Create, Update and Delete activity metrics.

5.2.1 Procedure: Create_ActMetric

Activity metrics are created from the leaf level only. That is, only calculation types of Manual, and Function can be added using this procedure.

Table 5–5 Procedure: Create_ActMetric

Parameters	Type	Description	Values
P_api_version	IN NUMBER	API Version number	1
P_init_msg_list	IN VARCHAR2	Boolean to initialize the message buffer	FND_API.G_FALSE FND_API.G_TRUE
P_commit	IN VARCHAR2	Boolean to indicate commit, default is false	FND_API.G_FALSE FND_API.G_TRUE
P_validation_level	IN NUMBER	Indicates validation level	FND_API.G_VALIDATE_LEVEL_FULL
X_return_status	OUT VARCHAR2	Indicates success or failure of process	FND_API.G_RET_STS_SUCCESS FND_API.G_RET_STS_UNEXP_ERROR FND_API.G_RET_STS_ERROR
x_msg_count	OUT NOCOPY NUMBER	Number of messages returned	Integer
X_msg_data	OUT NOCOPY VARCHAR2	Messages	Message text.
P_act_metric_rec	IN `AMS_ActMetric_PVT.act_metric_rec_type`	Activity Metric record.	See below
X_activity_metric_id	OUT NOCOPY NUMBER	New Activity Metric Id	Integer

5.2.2 Procedure: Update_ActMetric

Only leaf level activity metrics can be updated, including Manual and Function type. (The refresh engine automatically updates Summary and Rollups.)

Table 5–6 Procedure: Update_ActMetric

Parameters	Type	Description	Values
P_api_version	IN NUMBER	API Version number	1
P_init_msg_list	IN VARCHAR2	Boolean to initialize the message buffer	FND_API.G_FALSE FND_API.G_TRUE
P_commit	IN VARCHAR2	Boolean to indicate commit, default is false	FND_API.G_FALSE FND_API.G_TRUE
P_validation_level	IN NUMBER	Indicates validation level	FND_API.G_VALIDATE_LEVEL_FULL
X_return_status	OUT VARCHAR2	Indicates success or failure of process	FND_API.G_RET_STS_SUCCESS FND_API.G_RET_STS_UNEXP_ERROR FND_API.G_RET_STS_ERROR
x_msg_count	OUT NOCOPY NUMBER	Number of messages returned	Integer
X_msg_data	OUT NOCOPY VARCHAR2	Messages	Message text.
P_act_metric_rec	IN `AMS_ActMetric_PVT.act_metric_rec_type`	Activity Metric record.	See below

5.2.3 Procedure: Delete_ActMetric

Deleting activity metrics is restricted to only those with no child activity metrics, and Active, Completed, and Archived marketing objects cannot delete activity metrics.

Table 5–7 Procedure: Delete_ActMetric

Parameters	Type	Description	Values
P_api_version	IN NUMBER	API Version number	1
P_init_msg_list	IN VARCHAR2	Boolean to initialize the message buffer	FND_API.G_FALSE FND_API.G_TRUE

Table 5–7 Procedure: Delete_ActMetric

P_commit	IN VARCHAR2	Boolean to indicate commit, default is false	FND_API.G_FALSE FND_API.G_TRUE
X_return_status	OUT VARCHAR2	Indicates success or failure of process	FND_API.G_RET_STS_SUCCESS FND_API.G_RET_STS_UNEXP_ERROR FND_API.G_RET_STS_ERROR
x_msg_count	OUT NOCOPY NUMBER	Number of messages returned	Integer
X_msg_data	OUT NOCOPY VARCHAR2	Messages	Message text.
P_activity_metric_id	IN NUMBER	Activity Metric_id	Valid activity_metric_id
P_object_version_number	IN NUMBER	Object version number	The current object version number of the metric.

5.2.4 Record: AMS_ActMetric_PVT.act_metric_rec_type

See the following table for Record: AMS_ActMetric_PVT.act_metric_rec_type parameters.

Table 5–8 Record: AMS_ActMetric_PVT.act_metric_rec_type

Element	Type	Description	Values	Requirements
Activity_metric_id	NUMBER	Unique identifier for the activity metric	Integer	Required for update
last_update_date	DATE	Date last updated	Date	Not required
Last_updated_by	NUMBER	User last updated	User id	Not required
Creation_date	DATE	Date first created	Date	Not required
Created_by	NUMBER	User who created	User id	Not required
Last_update_login	NUMBER	User last updated	User id	Not required
Object_version_number	NUMBER	Incremental version number	Integer	Required for update. Must correspond to current value.

Table 5–8 Record: AMS_ActMetric_PVT.act_metric_rec_type

Application_id	NUMBER	Application the created this object	Integer, OMO=530	Set to 530 for creation. Not required for update and delete.
act_metric_used_by_id	NUMBER	Object ID of the association	Id from a Campaign or other valid object	Required
Arc_act_metric_used_by	VARCHAR2(30)	Type of object associated	For Campaigns= 'CAMP', etc.	Required
Purcahse_req_raised_flag	VARCHAR2(1)	Unused		
Sensitive_data_flag	VARCHAR2(1)	Unused		
Budget_id	NUMBER	Unused		
Metric_id	NUMBER	Metric Id to associate with the object.	Includes metric_ids of type Manual, and Function	Required on Creation. For Update this value is known from the database.
Transaction_currency_code	VARCHAR2(15)	Currency Code of displayed value.	From FND_CURRENCIES	Required for categories of 901 and 902 (Cost and Revenues).
Trans_forecasted_value	NUMBER	Displayed forecasted value	User entered value.	Optional
Trans_commited_value	NUMBER	Unused		
Trans_Actual_value	NUMBER	Displayed Actual Value	User entered value.	Optional.
Functional_currency_code	NUMBER	Currency Code for internally stored value	System set according to profile value `'AMS_DEFAULT_CURR_CODE'`	Not required
Func_forecasted_value	NUMBER	Internal value to total like kind values	Converted from trans_forecasted_value according to currency or UOM.	Not required

Table 5–8 Record: AMS_ActMetric_PVT.act_metric_rec_type

Dirty_flag	VARCHAR2(1)	Used by refresh engine when value has changed	Set to 'Y' in normal cases	Required.
Func_commited_ value	NUMBER	Unused		
Func_actual_value	NUBMER	Internal value to total like kind values	Converted from trans_actual_ value according to currency or UOM.	Not required. For budget allocation, it's allocation amount
Last_calculated_date	DATE	When value was last calculated by the refresh engine		Not required for update or create.
Variable_value	NUMBER	Unit value to multiply for variable type metrics		Optional: only needed for variable accrual type
Computed_using_ function_value	NUMBER	Unused		
Metric_uom_code	VARCHAR2(3)	Unit Of Measure conversion code. Used for allocations.		Not required by API.
Org_id	NUMBER	Unused		
Difference_since_ last_calc	NUMBER	Unused		
Activity_metric_ origin_id	NUMBER	Used for allocations, ID of object.		Not required by API.
Arc_activity_metric_ origin	VARCHAR2(30)	Used for allocations, type of object		Not required by API.
Days_since_last_ refresh	NUMBER	Delta since object was last recalculated by the refresh engine. Used for rollups and summary metrics only.		Not required.

Table 5–8 Record: AMS_ActMetric_PVT.act_metric_rec_type

Scenario_id	NUMBER	Unused		
Summarize_to_metric	NUMBER	Activity_metric_id of the summary to total with in the same object.		Not required by API.
Rollup_to_metric	NUMBER	Activity_metric_id of the rollup to total to parent object level.		Not required by API.
Hierarchy_id	NUMBER	Used for budget allocation hierarchy id		Only required for budget allocation,
Start_node	NUMBER	Used for budget allocation start node id.		Only required for budget allocation,
From_level	NUMBER	Used for budget allocation start level number.		Only required for budget allocation,
To_level	NUMBER	Used for budget allocation end level number.		Only required for budget allocation,
From_date	DATE	Used for budget allocation start date.		optional
To_date	DATE	Used for budget allocation end date.		optional
Amount1	NUMBER	Unused		
Amount2	NUMBER	Unused		
Amount3	NUMBER	Unused		
Percent1	NUMBER	Unused		
Percent2	NUMBER	Unused		
Percent3	NUMBER	Unused		

Table 5–8 Record: AMS_ActMetric_PVT.act_metric_rec_type

Published_flag	VARCHAR2(1)	Whether value has been posted to the budget. Only used for costs. Once set to 'Y' it cannot be changed.	'Y', 'N'	Required for Costs.
Pre_function_name	VARCHAR2(4000)	Unused		
Post_function_name	VARCHAR2(4000)	Unused		
Attribute_category	VARCHAR2(30)	Used for flex fields		Not required.
Attribute[1-15]	VARCHAR2(150)	Used for flex fields		Not required.
Act_metric_date	DATE	Transaction Date, for user reference only.		Optional. For budget allocation, it is allocation approval/cancellation date
Depend_act_metric	NUMBER	Activity_metric_id for variable metrics to reference multiplier		Not required by API.
Description	VARCHAR2(4000)	Description field for user reference only.		Optional.
Function_used_by_id	NUMBER	Object id referenced for function metrics.		Required for function metrics attached multiple times to same business object.
Arc_function_used_by	VARCHAR2(30)	Object type referenced for function metrics		Required for function metrics attached multiple times to same business object.
Hierarchy_type	VARCHAR2(30)	Required for budget allocation hierarchy type		Only required for budget allocation, lookup code in AMS_FUND_ALLOC_HIERTYPE.

Table 5–8 Record: AMS_ActMetric_PVT.act_metric_rec_type

Status_code	VARCHAR2(30)	Required for budget allocation status		Only required for budget allocation, lookup code in AMS_FUND_ALLOC_STATUS
Method_code	VARCHAR2(30)	Required for budget allocation method		Only required for budget allocation, lookup code in AMS_FUND_ALLOC_METHOD
Action_code	VARCHAR2(30)	Required for budget allocation action		Only required for budget allocation, lookup code in AMS_FUND_ALLOC_ACTION
Basis_year	NUMBER	Used by budget allocations		Only required for budget allocation when method is PRIOR_YEEAR_SALES.
Ex_start_node	VARCHAR2(1)	Used by budget allocations flag for excluding start node		Optional.
Ex_start_node	VARCHAR2(1)	Required for budget allocations		Optional.

6

Claim

The APIs for Trade Management Claims provides a number of procedures for Trade Management actions.

The procedures which make up the Claims APIs are:

Table 6–1 Trade Management APIs

Procedure	Description
Create Claim	Creates a claim and its lines based on the input information. A unique id will be assigned to the claim and a claim number will be generated if not provided.
Update Claim	Update a claim and its lines based on the input information.
Delete Claim	Delete a claim and its lines. Only claims that are in NEW status can be deleted.
Create Claim Line Table	Create claim lines for an existing claim based on the input. The claim id should be provided.
Update Claim Line Table	Update claim lines for an existing claim based on the input. The claim id should be provided.
Delete Claim Line Table	Delete claim lines for an existing claim based on the input. The claim id should be provided.

6.1 Type Declaration

There are two record type used in this API. They are claim_rec_type and claim_line_rec_type. Since a claim can have multiple lines associated with it, only the colletion verions of the Claim_line_rec_type, TYPE claim_line_tbl_type is used.

```
TYPE claim_rec_type IS RECORD
(
    claim_id                    NUMBER := FND_API.G_MISS_NUM,
    object_version_number       NUMBER := FND_API.G_MISS_NUM,
    last_update_date            DATE := FND_API.G_MISS_DATE,
    last_updated_by             NUMBER := FND_API.G_MISS_NUM,
    creation_date               DATE := FND_API.G_MISS_DATE,
    created_by                  NUMBER := FND_API.G_MISS_NUM,
    last_update_login           NUMBER := FND_API.G_MISS_NUM,
    request_id                  NUMBER := FND_API.G_MISS_NUM,
    program_application_id      NUMBER := FND_API.G_MISS_NUM,
    program_update_date         DATE := FND_API.G_MISS_DATE,
    program_id                  NUMBER := FND_API.G_MISS_NUM,
    created_from                VARCHAR2(30) := FND_API.G_MISS_CHAR,
    batch_id                    NUMBER := FND_API.G_MISS_NUM,
    claim_number                VARCHAR2(30) := FND_API.G_MISS_CHAR,
    claim_type_id               NUMBER := FND_API.G_MISS_NUM,
    claim_line_id               NUMBER := FND_API.G_MISS_NUM,
    claim_class                 VARCHAR2(30) := FND_API.G_MISS_CHAR,
    claim_date                  DATE := FND_API.G_MISS_DATE,
    due_date                    DATE := FND_API.G_MISS_DATE,
    owner_id                    NUMBER := FND_API.G_MISS_NUM,
    history_event               VARCHAR2(30) := FND_API.G_MISS_CHAR,
    history_event_date          DATE := FND_API.G_MISS_DATE,
    history_event_description   VARCHAR2(2000) := FND_API.G_MISS_CHAR,
    split_from_claim_id         NUMBER := FND_API.G_MISS_NUM,
    duplicate_claim_id          NUMBER := FND_API.G_MISS_NUM,
    split_date                  DATE := FND_API.G_MISS_DATE,
```

```
root_claim_id               NUMBER := FND_API.G_MISS_NUM,
amount                      NUMBER := FND_API.G_MISS_NUM,
amount_adjusted             NUMBER := FND_API.G_MISS_NUM,
amount_remaining            NUMBER := FND_API.G_MISS_NUM,
amount_settled              NUMBER := FND_API.G_MISS_NUM,
acctd_amount_settled        NUMBER := FND_API.G_MISS_NUM,
acctd_amount_adjusted       NUMBER := FND_API.G_MISS_NUM,
acctd_amount                NUMBER := FND_API.G_MISS_NUM,
acctd_amount_remaining      NUMBER := FND_API.G_MISS_NUM,
tax_amount                  NUMBER := FND_API.G_MISS_NUM,
tax_code                    VARCHAR2(50) := FND_API.G_MISS_CHAR,
tax_calculation_flag        VARCHAR2(1) := FND_API.G_MISS_CHAR,
currency_code               VARCHAR2(15) := FND_API.G_MISS_CHAR,
exchange_rate_type          VARCHAR2(30) := FND_API.G_MISS_CHAR,
exchange_rate_date          DATE := FND_API.G_MISS_DATE,
exchange_rate               NUMBER := FND_API.G_MISS_NUM,
set_of_books_id             NUMBER := FND_API.G_MISS_NUM,
original_claim_date         DATE := FND_API.G_MISS_DATE,
source_object_id            NUMBER := FND_API.G_MISS_NUM,
source_object_class         VARCHAR2(15) := FND_API.G_MISS_CHAR,
source_object_type_id       NUMBER := FND_API.G_MISS_NUM,
source_object_number        VARCHAR2(30) := FND_API.G_MISS_CHAR,
cust_account_id             NUMBER := FND_API.G_MISS_NUM,
cust_billto_acct_site_id    NUMBER := FND_API.G_MISS_NUM,
cust_shipto_acct_site_id    NUMBER := FND_API.G_MISS_NUM,
location_id                 NUMBER := FND_API.G_MISS_NUM,
pay_related_account_flag    VARCHAR2(1) := FND_API.G_MISS_CHAR,
related_cust_account_id     NUMBER := FND_API.G_MISS_NUM,
```

```
related_site_use_id          NUMBER := FND_API.G_MISS_NUM,
relationship_type            VARCHAR2(30) := FND_API.G_MISS_CHAR,
vendor_id                    NUMBER := FND_API.G_MISS_NUM,
vendor_site_id               NUMBER := FND_API.G_MISS_NUM,
reason_type                  VARCHAR2(30) := FND_API.G_MISS_CHAR,
reason_code_id               NUMBER := FND_API.G_MISS_NUM,
task_template_group_id       NUMBER := FND_API.G_MISS_NUM,
status_code                  VARCHAR2(30) := FND_API.G_MISS_CHAR,
user_status_id               NUMBER := FND_API.G_MISS_NUM,
sales_rep_id                 NUMBER := FND_API.G_MISS_NUM,
collector_id                 NUMBER := FND_API.G_MISS_NUM,
contact_id                   NUMBER := FND_API.G_MISS_NUM,
broker_id                    NUMBER := FND_API.G_MISS_NUM,
territory_id                 NUMBER := FND_API.G_MISS_NUM,
customer_ref_date            DATE := FND_API.G_MISS_DATE,
customer_ref_number          VARCHAR2(30) := FND_API.G_MISS_CHAR,
assigned_to                  NUMBER := FND_API.G_MISS_NUM,
receipt_id                   NUMBER := FND_API.G_MISS_NUM,
receipt_number               VARCHAR2(30) := FND_API.G_MISS_CHAR,
doc_sequence_id              NUMBER := FND_API.G_MISS_NUM,
doc_sequence_value           NUMBER := FND_API.G_MISS_NUM,
gl_date                      DATE := FND_API.G_MISS_DATE,
payment_method               VARCHAR2(15) := FND_API.G_MISS_CHAR,
voucher_id                   NUMBER := FND_API.G_MISS_NUM,
voucher_number               VARCHAR2(30) := FND_API.G_MISS_CHAR,
payment_reference_id         NUMBER := FND_API.G_MISS_NUM,
payment_reference_number     VARCHAR2(15) := FND_API.G_MISS_CHAR,
payment_reference_date       DATE := FND_API.G_MISS_DATE,
```

```
payment_status              VARCHAR2(10) := FND_API.G_MISS_CHAR,
approved_flag               VARCHAR2(1) := FND_API.G_MISS_CHAR,
approved_date               DATE := FND_API.G_MISS_DATE,
approved_by                 NUMBER := FND_API.G_MISS_NUM,
settled_date                DATE := FND_API.G_MISS_DATE,
settled_by                  NUMBER := FND_API.G_MISS_NUM,
effective_date              DATE := FND_API.G_MISS_DATE,
custom_setup_id             NUMBER := FND_API.G_MISS_NUM,
task_id                     NUMBER := FND_API.G_MISS_NUM,
country_id                  NUMBER := FND_API.G_MISS_NUM,
order_type_id               NUMBER := FND_API.G_MISS_NUM,
comments                    VARCHAR2(2000) := FND_API.G_MISS_CHAR,
activity_type               VARCHAR2(30):=FND_API.G_MISS_CHAR,
activity_id                 NUMBER := FND_API.G_MISS_NUM,
earnings_associated_flag    VARCHAR2(1):=FND_API.G_MISS_CHAR,
quantity                    NUMBER:=FND_API.G_MISS_NUM,
quantity_uom                VARCHAR2(30):=FND_API.G_MISS_CHAR,
rate                        NUMBER:=FND_API.G_MISS_NUM,
item_id                     NUMBER:=FND_API.G_MISS_NUM,
item_description            VARCHAR2(240):=FND_API.G_MISS_CHAR,
performance_complete_flag   VARCHAR2(1):= FND_API.G_MISS_CHAR,
performance_attached_flag   VARCHAR2(1):=FND_API.G_MISS_CHAR,
utilization_id              NUMBER:=FND_API.G_MISS_NUM,
plan_id                     NUMBER:=FND_API.G_MISS_NUM,
offer_id                    NUMBER:=FND_API.G_MISS_NUM,
valid_flag                  VARCHAR2(1):=FND_API.G_MISS_CHAR,
claim_currency_amount       NUMBER:=FND_API.G_MISS_NUM,
split_from_claim_line_id    NUMBER:=FND_API.G_MISS_NUM,
```

```
line_number                NUMBER:=FND_API.G_MISS_NUM,
attribute_category          VARCHAR2(30) := FND_API.G_MISS_CHAR,
attribute1                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute2                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute3                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute4                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute5                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute6                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute7                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute8                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute9                 VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute10                VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute11                VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute12                VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute13                VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute14                VARCHAR2(150) := FND_API.G_MISS_CHAR,
attribute15                VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute_category   VARCHAR2(30) := FND_API.G_MISS_CHAR,
deduction_attribute1       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute2       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute3       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute4       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute5       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute6       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute7       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute8       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute9       VARCHAR2(150) := FND_API.G_MISS_CHAR,
deduction_attribute10      VARCHAR2(150) := FND_API.G_MISS_CHAR,
```

```
    deduction_attribute11        VARCHAR2(150) := FND_API.G_MISS_CHAR,
    deduction_attribute12        VARCHAR2(150) := FND_API.G_MISS_CHAR,
    deduction_attribute13        VARCHAR2(150) := FND_API.G_MISS_CHAR,
    deduction_attribute14        VARCHAR2(150) := FND_API.G_MISS_CHAR,
    deduction_attribute15        VARCHAR2(150) := FND_API.G_MISS_CHAR,
    org_id                  NUMBER := FND_API.G_MISS_NUM
 );


TYPE claim_line_rec_type IS RECORD
(
 claim_line_id           NUMBER  ,
 object_version_number    NUMBER ,
 last_update_date         DATE ,
 last_updated_by          NUMBER,
 creation_date           DATE,
 created_by              NUMBER,
 last_update_login        NUMBER,
 request_id              NUMBER,
 program_application_id    NUMBER,
 program_update_date      DATE,
 program_id              NUMBER,
 created_from            VARCHAR2(30),
 claim_id                NUMBER,
 line_number             NUMBER,
 split_from_claim_line_id  NUMBER,
 amount                  NUMBER,
 claim_currency_amount    NUMBER,
```

```
acctd_amount              NUMBER,
currency_code             VARCHAR2(15),
exchange_rate_type        VARCHAR2(30),
exchange_rate_date        DATE ,
exchange_rate             NUMBER,
set_of_books_id           NUMBER,
valid_flag                VARCHAR2(1),
source_object_id          NUMBER,
source_object_class       VARCHAR2(15),
source_object_type_id     NUMBER,
source_object_line_id     NUMBER,
plan_id                   NUMBER,
offer_id                  NUMBER,
utilization_id            NUMBER,
payment_method            VARCHAR2(15),
payment_reference_id      NUMBER,
payment_reference_number  VARCHAR2(15),
payment_reference_date    DATE ,
voucher_id                NUMBER,
voucher_number            VARCHAR2(30),
payment_status            VARCHAR2(10),
approved_flag             VARCHAR2(1),
approved_date             DATE ,
approved_by               NUMBER,
settled_date              DATE ,
settled_by                NUMBER,
performance_complete_flag VARCHAR2(1),
performance_attached_flag VARCHAR2(1),
```

```
item_id                 NUMBER,
item_description        VARCHAR2(240),
quantity                NUMBER,
quantity_uom            VARCHAR2(30),
rate                    NUMBER,
activity_type           VARCHAR2(30),
activity_id             NUMBER,
related_cust_account_id NUMBER,
relationship_type       VARCHAR2(30),
earnings_associated_flag VARCHAR2(1),
comments                VARCHAR2(2000),
tax_code                VARCHAR2(50),
attribute_category      VARCHAR2(30),
attribute1              VARCHAR2(150),
attribute2              VARCHAR2(150),
attribute3              VARCHAR2(150),
attribute4              VARCHAR2(150),
attribute5              VARCHAR2(150),
attribute6              VARCHAR2(150),
attribute7              VARCHAR2(150),
attribute8              VARCHAR2(150),
attribute9              VARCHAR2(150),
attribute10             VARCHAR2(150),
attribute11             VARCHAR2(150),
attribute12             VARCHAR2(150),
attribute13             VARCHAR2(150),
attribute14             VARCHAR2(150),
attribute15             VARCHAR2(150),
```

```
 org_id                 NUMBER,
 update_from_tbl_flag   VARCHAR2(1)   := FND_API.g_false
);

TYPE claim_line_tbl_type is TABLE OF claim_line_rec_type
INDEX BY BINARY_INTEGER;
```

6.2 Standard Parameters for Claims APIs

The Claims API parameters are the same Standard parameters as in events.

6.2.1 Claim Public API

This procedure creates a claim.

Procedure Specification

```
PROCEDURE Create_Claim(
   p_api_version_number       IN   NUMBER,
   p_init_msg_list            IN   VARCHAR2    := FND_API.G_FALSE,
   p_commit                   IN   VARCHAR2    := FND_API.G_FALSE,
   p_validation_level         IN    NUMBER   := FND_API.g_valid_level_full,
   x_return_status            OUT NOCOPY  VARCHAR2,
   x_msg_count                OUT NOCOPY  NUMBER,
   x_msg_data                 OUT NOCOPY  VARCHAR2,
   p_claim_rec                IN   claim_rec_type := AMS_Claim_PUB.g_miss_claim_rec,
   p_claim_line_tbl           IN   claim_line_tbl_type,
   x_claim_id                 OUT NOCOPY  NUMBER
    );
```

Current Version

1.0

Parameters

Table 6–2 IN Parameters

Parameter	Data Type	Required	Description
p_api_version_number	Number	Yes	Caller Versio number. Will be compared against the API version number to detect incompatibility.
p_init_msg_list	VarChar	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
P_validation_level	Number	No	Level of validation requested. Default: FND_API.g_valid_level_full
p_claim_rec	AMS_Claim_PUB.claim_rec_type	Yes	Record of claim to be created.
p_claim_line_tbl	AMS_Claim_PUB.claim_line_tbl_type	No	Collection of claim line of table.

Table 6–3 OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar	Return status of the API. The values could be: FND_API.G_RET_STS_SUCCESS: procedure runs successfully. FND_API.G_RET_STS_ERROR: Procedure failed. A validation or data error has happened. FND_API.G_RET_STS_UNEXP_ERROR: Procedure failed. An unexpected error happened during execution.
x_msg_count	Number	Number of the error message returns by the API.
x_msg_data	VarChar2	Error message returned by API.
x_claim_id	Number	New Claim identifier.

6.2.2 Update Claim

This procedure updates the claim. .

Procedure Specification

```
PROCEDURE Update_Claim(
  p_api_version_number       IN   NUMBER,
  p_init_msg_list            IN   VARCHAR2    := FND_API.G_FALSE,
  p_commit                   IN   VARCHAR2    := FND_API.G_FALSE,
  p_validation_level         IN    NUMBER   := FND_API.g_valid_level_full,
  x_return_status            OUT NOCOPY  VARCHAR2,
  x_msg_count                OUT NOCOPY  NUMBER,
  x_msg_data                 OUT NOCOPY  VARCHAR2,
  p_claim_rec                IN   claim_rec_type,
  p_claim_line_tbl           IN   claim_line_tbl_type,
  x_object_version_number    OUT NOCOPY  NUMBER
  );
```

Current Version

1.0

Parameters

Table 6–4 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller Version number. Will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.

Table 6–4 IN Parameters

Parameter	Data Type	Required	Description
p_claim_record	AMS_Claim_PUB.claim_rec_type	Yes	Record of claim to be updated.
p_claim_line_tbl	AMS_Claim_PUB.claim_line_tbl_type	No	

Table 6–5 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.
x_object_version_number	Number	New object vesion number of the claim

6.2.3 Delete Claim

This procedure deletes the claim.

Procedure Specification

```
PROCEDURE Delete_Claim(
  p_api_version_number      IN   NUMBER,
  p_init_msg_list           IN   VARCHAR2    := FND_API.G_FALSE,
```

```
p_commit                IN   VARCHAR2    := FND_API.G_FALSE,
p_validation_level      IN    NUMBER   := FND_API.g_valid_level_full,
x_return_status          OUT NOCOPY  VARCHAR2,
x_msg_count              OUT NOCOPY  NUMBER,
x_msg_data              OUT NOCOPY  VARCHAR2,
p_claim_id              IN  NUMBER,
p_object_version_number     IN   NUMBER
);
```

Current Version

1.0

Parameter Descriptions

Table 6–6 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the change should be commited on success.
p_validation_ level	Number	No	Level of validation requested. Default: FND_API.g_valid_level_full
p_claim_id	Number	Yes	Identification of the claim to be deleted
p_object_ version_ number	Number	Yes	Object version number of the claim to be deleted

Table 6–7 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

6.2.4 Create Claim Line Table

Procedure Specification

PROCEDURE Create_Claim_Line_Tbl(

p_api_version IN NUMBER

,p_init_msg_list IN VARCHAR2 := FND_API.g_false

,p_commit IN VARCHAR2 := FND_API.g_false

,p_validation_level IN NUMBER := FND_API.g_valid_level_full

,x_return_status OUT NOCOPY VARCHAR2

,x_msg_data OUT NOCOPY VARCHAR2

,x_msg_count OUT NOCOPY NUMBER

,p_claim_line_tbl IN claim_line_tbl_type

,x_error_index OUT NOCOPY NUMBER);

Current Version

1.0

Parameter Descriptions

Table 6–8 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	Number	No	Level of validation requested. Default: FND_API.g_valid_level_full
p_claim_line_tbl	AMS_Claim_PUB.claim_line_tbl_type	Yes	Collection of claim line of the claim to be created. The id of claim which the claim line belongs to should be popluated in this record.

Table 6–9 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

Table 6–9 OUT Parameters

Parameter	Data Type	Description
x_error_index	Number	The index of the claim line which caused error during claim line creation.

6.2.5 Update Claim Line Table

Procedure Specification

PROCEDURE Update_Claim_Line_Tbl(

p_api_version IN NUMBER

,p_init_msg_list IN VARCHAR2 := FND_API.g_false

,p_commit IN VARCHAR2 := FND_API.g_false

,p_validation_level IN NUMBER := FND_API.g_valid_level_full

,x_return_status OUT NOCOPY VARCHAR2

,x_msg_data OUT NOCOPY VARCHAR2

,x_msg_count OUT NOCOPY NUMBER

,p_claim_line_tbl IN claim_line_tbl_type

,p_change_object_version IN VARCHAR2 := FND_API.g_false

,x_error_index OUT NOCOPY NUMBER);

Current Version

1.0

Parameter Descriptions

Table 6–10 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.

Table 6–10 IN Parameters

Parameter	Data Type	Required	Description
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VarChar2	No	Flag to indicate if the change should be commited on success. Default: FND_API.g_false.
p_validation_level	Number	No	Level of validation requested. Default: FND_API.g_valid_level_full
p_claim_line_tbl	AMS_Claim_PUB.claim_line_tbl_type	Yes	Collection of claim line of the claim to be updated. The id of claim which the claim line belongs be should be popluated in this record.
p_change_object_version	VarChar2	No	Indicator whether to change the claim line version. Default is FND_API.g_false

Table 6–11 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.
x_error_index	Number	The index of the claim line which caused error during claim line update.

6.2.6 Delete Claim Line Table

Procedure Specification

PROCEDURE Delete_Claim_Line_Tbl(

p_api_version IN NUMBER

,p_init_msg_list IN VARCHAR2 := FND_API.g_false

,p_commit IN VARCHAR2 := FND_API.g_false

,p_validation_level IN NUMBER := FND_API.g_valid_level_full

,x_return_status OUT NOCOPY VARCHAR2

,x_msg_data OUT NOCOPY VARCHAR2

,x_msg_count OUT NOCOPY NUMBER

,p_claim_line_tbl IN claim_line_tbl_type

,p_change_object_version IN VARCHAR2 := FND_API.g_false

,x_error_index OUT NOCOPY NUMBER);

Current Version

1.0

Parameter Descriptions

Table 6–12 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.

Table 6–12 IN Parameters

Parameter	Data Type	Required	Description
p_claim_line_ tbl	AMS_Claim_ PUB.claim_ line_tbl_type	Yes	Collection of claim line of the claim to be deleted.
p_change_ object_version	VarChar2	No	Indicator whether to change the claim line version. Default is FND_API.g_false

Table 6–13 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.
x_error_index	Number	The index of the claim line which caused error during claim line delete.

7

Third-Party Accruals

The Third Party Accrual API enables customers to generate accruals on the orders that they made through third party whole salers. The API simulates the pricing of the orders and then creates order information in chargeback order tables. It posts the difference between customer paid price and simulated price to a budget that is setup by the customer. Any discount and accrual applied to the order is accrualed.

Order information is stored in ozf_chargeabck_int_all table. The API process orders from direct customers as well as indirect customers. For indirect customer orders, the API does not run the pricing simulation. It copied the order information to the chargeback order tables. For direct customers, the API validates the data, runs the simulation, creates the order and posts the accrual amount. Direct_customer_flag in ozf_chargeback_int_all indicates whether the order record is from a direct customer or not. Discount and accrual related information is stored in ozf_chargebabck_ price_adj_all table.

For any exception generated during the process, a log record is created in an interface log table. User can use this table to modify the data.

This API consists of the following tables:

- Ozf_chargeback_int_all
- Ozf_chargeback_headers_all
- Ozf_chargeback_lines_all
- Ozf_chargeback_price_adj_all
- Ozf_chargeback_int_log

7.1 Validation Rules

The followings causes an error log generated for any record:

- No order number for the order.
- No bill_to_account_name for indirect customer orders.
- No bill_to_cust_account_id for direct customer orders.
- No UOM for direct customer orders.
- Bill_to_cust_account_id is invalid for direct customer orders.
- UOM is invalid for direct customer orders.
- Any error returned by the pricing engine.
- Any error returned by the funds utilization call.

7.2 Concurrent Programs

This API consists of two concurrent programs:

7.2.1 AMS-TM: Chargeback Accrual

This program reads records in ozf_chargeback_int_all table. For each unique combination of order number and bill_to_cust_account_id, it creates an order header in ozf_chargeback_headers_all and copies order line information to ozf_chargeback_lines_all. For each order, it runs the pricing simulation API to get the price for each order line. If the new price is smaller then the paid price, the program will update the utilized amount in funds utilization, otherwise the nothing will be posted to that account. Any discount and accrual that should be applied to the order is accrualed for the customer.

7.2.2 AMS-TM: Chargeback Purge

This program deletes the processed order from the interface table.

7.2.3 Preparing the Concurrent Programs

In order for the Concurrent Program to run properly the following conditions must exist:

- A user needs to setup the profile: AMS: Price Differenct Budget for the concurrent program to run.

- GL account information in ozf_sys_parameters needs to be setup properly.
- Account generator workflow needs to be compiled properly.

7.2.4 Tables Updated by the Concurrent Program

- Ozf_chargeback_int_all
- Ozf_chargeback_headers_all
- Ozf_chargeback_lines_all
- Ozf_chargeback_price_adj_all
- Ozf_chargeback_int_log
- Ozf_funds_utilized_all_b
- Ozf_funds_all_b
- Ams_act_budgets

8

Leads

The APIs for Leads provides 13 procedures for managing Lead information.

The procedures which make up the Leads APIs are:

Table 8–1 Leads APIs

Procedure	Description
Create Sales Lead	Creates a new sales lead with the specified parameters. A sales lead record with the parameters set needs to be passed in. A unique sales lead ID will be created.
Update Sales Lead	Update the sales lead record. The sales lead record being updated must refer to a valid sales lead ID in the database and must have a valid last_update_date (who column) passed in. This is used to check if the record has not been updated by someone else since it was last loaded. If a g_miss value is passed for other fields in the sales lead record type, then these values will not be updated.
Create Sales Lead Lines	Creates one or more new sales lead lines with the specified parameters. A table of sales lead line records with the parameters set needs to be passed in. A unique sales lead line ID will be created for each sales lead line. All the sales lead lines created will be associated with the sales lead ID which is passed in the sales lead line record.
Update Sales Lead Lines	Updates one or more sales lead lines. A table of sales lead line records with the parameters needs to be passed in. Each sales lead line record must refer to a valid sales lead line ID in the database and must have a valid last_update_date (who column passed in). This is used to check if the sales lead line record has not been updated by someone else since it was last loaded. If a g_miss value is passed for other fields in the sales lead line record type, then these will not be updated.

Table 8–1 Leads APIs

Procedure	Description
Delete Sales Lead Lines	Deletes one or more sales lead lines. A table of sales lead line records needs to be passed in. Each sales lead line record must refer to a valid sales lead line ID in the database.
Create Sales Lead Contacts	Creates one or more new sales lead contacts with the specified parameters. A table of sales lead contact records with the parameters set needs to be passed in. A unique lead contact ID will be created for each sales lead contact. All the sales lead contacts created will be associated with the sales lead ID which is passed in the sales lead contact record.
Update Sales Lead Contacts	Updates one or more sales lead contacts. A table of sales lead contacts records with the parameters needs to be passed in. Each sales lead contact record must refer to a valid sales lead contact ID in the database and must have a valid last_update_ date (who column passed in). This is used to check if the sales lead contact record has not been updated by someone else since it was last loaded. If a g_miss value is passed for other fields in the sales lead contact record type, then these will not be updated.
Delete Sales Lead Contacts	Deletes one or more sales lead contacts. A table of sales lead contact records needs to be passed in. Each sales lead contact record must refer to a valid lead contact ID in the database.
Get Sales Lead Score	Gets the score and rank of the Sales Lead based on the Score card
Run Lead Engines	Run qualification engine, rating engine, and channel selection engine
Build Lead Sales Team	Build lead sales team based on territory definition and adds lead creator as one of lead sales team member.
Rebuild Lead Sales Team	Rebuild lead sales team to reflect the latest lead information.
Start Partner Matching	Start a workflow to do partner matching.

8.1 Type Declarations

This section defines the various record types used in the Leads APIs. They are used as an IN parameter in many of the APIs for creation or updating of Leads objects.

8.1.1 Sales Lead Type

```
TYPE SALES_LEAD_Rec_Type IS RECORD(
    SALES_LEAD_ID                   NUMBER := FND_API.G_MISS_NUM,
    LAST_UPDATE_DATE                DATE := FND_API.G_MISS_DATE,
    LAST_UPDATED_BY                 NUMBER := FND_API.G_MISS_NUM,
    CREATION_DATE                   DATE := FND_API.G_MISS_DATE,
    CREATED_BY                      NUMBER := FND_API.G_MISS_NUM,
    LAST_UPDATE_LOGIN               NUMBER := FND_API.G_MISS_NUM,
    REQUEST_ID                      NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_APPLICATION_ID          NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_ID                      NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_UPDATE_DATE             DATE := FND_API.G_MISS_DATE,
    LEAD_NUMBER                     VARCHAR2(30) := FND_API.G_MISS_CHAR,
    STATUS_CODE                     VARCHAR2(30) := FND_API.G_MISS_CHAR,
    CUSTOMER_ID                     NUMBER := FND_API.G_MISS_NUM,
    ADDRESS_ID                      NUMBER := FND_API.G_MISS_NUM,
    SOURCE_PROMOTION_ID             NUMBER := FND_API.G_MISS_NUM,
    INITIATING_CONTACT_ID           NUMBER := FND_API.G_MISS_NUM,
    ORIG_SYSTEM_REFERENCE           VARCHAR2(240) := FND_API.G_MISS_CHAR,
    CONTACT_ROLE_CODE               VARCHAR2(30) := FND_API.G_MISS_CHAR,
    CHANNEL_CODE                    VARCHAR2(30) := FND_API.G_MISS_CHAR,
    BUDGET_AMOUNT                   NUMBER := FND_API.G_MISS_NUM,
    CURRENCY_CODE                   VARCHAR2(15) := FND_API.G_MISS_CHAR,
    DECISION_TIMEFRAME_CODE         VARCHAR2(30) := FND_API.G_MISS_CHAR,
    CLOSE_REASON                    VARCHAR2(30) := FND_API.G_MISS_CHAR,
    LEAD_RANK_ID                    NUMBER := FND_API.G_MISS_NUM,
    LEAD_RANK_CODE                  VARCHAR2(30) := FND_API.G_MISS_CHAR,
    PARENT_PROJECT                  VARCHAR2(80) := FND_API.G_MISS_CHAR,
    DESCRIPTION                     VARCHAR2(2000) := FND_API.G_MISS_CHAR,
    ATTRIBUTE_CATEGORY              VARCHAR2(30) := FND_API.G_MISS_CHAR,
    ATTRIBUTE1                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE2                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE3                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE4                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE5                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE6                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE7                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE8                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE9                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE10                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    TTRIBUTE11                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE12                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE13                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE14                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
```

```
    ATTRIBUTE15                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ASSIGN_TO_PERSON_ID             NUMBER := FND_API.G_MISS_NUM,
    ASSIGN_TO_SALESFORCE_ID         NUMBER := FND_API.G_MISS_NUM,
    ASSIGN_SALES_GROUP_ID           NUMBER := FND_API.G_MISS_NUM,
    ASSIGN_DATE                     DATE := FND_API.G_MISS_DATE,
    BUDGET_STATUS_CODE              VARCHAR2(30) := FND_API.G_MISS_CHAR,
    ACCEPT_FLAG                     VARCHAR2(1) := FND_API.G_MISS_CHAR,
    VEHICLE_RESPONSE_CODE           VARCHAR2(30) := FND_API.G_MISS_CHAR,
    TOTAL_SCORE                     NUMBER := FND_API.G_MISS_NUM,
    SCORECARD_ID                    NUMBER := FND_API.G_MISS_NUM,
    KEEP_FLAG                       VARCHAR2(1) := FND_API.G_MISS_CHAR,
    URGENT_FLAG                     VARCHAR2(1) := FND_API.G_MISS_CHAR,
    IMPORT_FLAG                     VARCHAR2(1) := FND_API.G_MISS_CHAR,
    REJECT_REASON_CODE              VARCHAR2(30) := FND_API.G_MISS_CHAR,
    DELETED_FLAG                    VARCHAR2(1) := FND_API.G_MISS_CHAR,
    OFFER_ID                        NUMBER := FND_API.G_MISS_NUM,
    INCUMBENT_PARTNER_PARTY_ID      NUMBER := FND_API.G_MISS_NUM,
    INCUMBENT_PARTNER_RESOURCE_ID   NUMBER := FND_API.G_MISS_NUM,
    PRM_EXEC_SPONSOR_FLAG           VARCHAR2(1) := FND_API.G_MISS_CHAR,
    PRM_PRJ_LEAD_IN_PLACE_FLAG      VARCHAR2(1) := FND_API.G_MISS_CHAR,
    PRM_SALES_LEAD_TYPE             VARCHAR2(30) := FND_API.G_MISS_CHAR,
    PRM_IND_CLASSIFICATION_CODE     VARCHAR2(30) := FND_API.G_MISS_CHAR,
    QUALIFIED_FLAG                  VARCHAR2(1) := FND_API.G_MISS_CHAR,
    ORIG_SYSTEM_CODE                VARCHAR2(30) := FND_API.G_MISS_CHAR,
    PRM_ASSIGNMENT_TYPE             VARCHAR2(30) := FND_API.G_MISS_CHAR,
    AUTO_ASSIGNMENT_TYPE            VARCHAR2(30) := FND_API.G_MISS_CHAR,
    PRIMARY_CONTACT_PARTY_ID        NUMBER := FND_API.G_MISS_NUM,
    PRIMARY_CNT_PERSON_PARTY_ID     NUMBER := FND_API.G_MISS_NUM,
    PRIMARY_CONTACT_PHONE_ID        NUMBER := FND_API.G_MISS_NUM,
    REFERRED_BY                     NUMBER := FND_API.G_MISS_NUM,
    REFERRAL_TYPE                   VARCHAR2(30) := FND_API.G_MISS_CHAR,
    REFERRAL_STATUS                 VARCHAR2(30) := FND_API.G_MISS_CHAR,
    REF_DECLINE_REASON              VARCHAR2(30) := FND_API.G_MISS_CHAR,
    REF_COMM_LTR_STATUS             VARCHAR2(30) := FND_API.G_MISS_CHAR,
    REF_ORDER_NUMBER                NUMBER := FND_API.G_MISS_NUM,
    REF_ORDER_AMT                   NUMBER := FND_API.G_MISS_NUM,
    REF_COMM_AMT                    NUMBER := FND_API.G_MISS_NUM
);
```

8.1.2 Sales Lead Line Type

```
TYPE SALES_LEAD_LINE_Rec_Type IS RECORD(
    SALES_LEAD_LINE_ID              NUMBER := FND_API.G_MISS_NUM,
    LAST_UPDATE_DATE                DATE := FND_API.G_MISS_DATE,
    LAST_UPDATED_BY                 NUMBER := FND_API.G_MISS_NUM,
```

```
    CREATION_DATE                   DATE := FND_API.G_MISS_DATE,
    CREATED_BY                      NUMBER := FND_API.G_MISS_NUM,
    LAST_UPDATE_LOGIN               NUMBER := FND_API.G_MISS_NUM,
    REQUEST_ID                      NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_APPLICATION_ID          NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_ID                      NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_UPDATE_DATE             DATE := FND_API.G_MISS_DATE,
    SALES_LEAD_ID                   NUMBER := FND_API.G_MISS_NUM,
    STATUS_CODE                     VARCHAR2(30) := FND_API.G_MISS_CHAR,
    INTEREST_TYPE_ID                NUMBER := FND_API.G_MISS_NUM,
    PRIMARY_INTEREST_CODE_ID        NUMBER := FND_API.G_MISS_NUM,
    SECONDARY_INTEREST_CODE_ID      NUMBER := FND_API.G_MISS_NUM,
    INVENTORY_ITEM_ID               NUMBER := FND_API.G_MISS_NUM,
    ORGANIZATION_ID                 NUMBER := FND_API.G_MISS_NUM,
    UOM_CODE                        VARCHAR2(3) := FND_API.G_MISS_CHAR,
    QUANTITY                        NUMBER := FND_API.G_MISS_NUM,
    BUDGET_AMOUNT                   NUMBER := FND_API.G_MISS_NUM,
    SOURCE_PROMOTION_ID             NUMBER := FND_API.G_MISS_NUM,
    ATTRIBUTE_CATEGORY              VARCHAR2(30) := FND_API.G_MISS_CHAR,
    ATTRIBUTE1                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE2                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE3                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE4                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE5                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE6                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE7                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE8                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE9                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE10                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE11                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE12                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE13                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE14                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE15                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    OFFER_ID                        NUMBER:= FND_API.G_MISS_NUM
);
```

8.1.3 Sales Lead Line Out Type

```
TYPE SALES_LEAD_LINE_OUT_Rec_Type   IS RECORD(
    SALES_LEAD_LINE_ID              NUMBER,
    RETURN_STATUS                   VARCHAR2(1)
);
```

8.1.4 Sales Lead Contact Type

```
TYPE SALES_LEAD_CONTACT_Rec_Type IS RECORD(
    LEAD_CONTACT_ID                 NUMBER := FND_API.G_MISS_NUM,
    SALES_LEAD_ID                   NUMBER := FND_API.G_MISS_NUM,
    CONTACT_ID                      NUMBER := FND_API.G_MISS_NUM,
    LAST_UPDATE_DATE                DATE := FND_API.G_MISS_DATE,
    LAST_UPDATED_BY                 NUMBER := FND_API.G_MISS_NUM,
    CREATION_DATE                   DATE := FND_API.G_MISS_DATE,
    CREATED_BY                      NUMBER := FND_API.G_MISS_NUM,
    LAST_UPDATE_LOGIN               NUMBER := FND_API.G_MISS_NUM,
    REQUEST_ID                      NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_APPLICATION_ID          NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_ID                      NUMBER := FND_API.G_MISS_NUM,
    PROGRAM_UPDATE_DATE             DATE := FND_API.G_MISS_DATE,
    ENABLED_FLAG                    VARCHAR2(1) := FND_API.G_MISS_CHAR,
    RANK                            VARCHAR2(30) := FND_API.G_MISS_CHAR,
    CUSTOMER_ID                     NUMBER := FND_API.G_MISS_NUM,
    ADDRESS_ID                      NUMBER := FND_API.G_MISS_NUM,
    PHONE_ID                        NUMBER := FND_API.G_MISS_NUM,
    CONTACT_ROLE_CODE               VARCHAR2(30) := FND_API.G_MISS_CHAR,
    PRIMARY_CONTACT_FLAG            VARCHAR2(1) := FND_API.G_MISS_CHAR,
    ATTRIBUTE_CATEGORY              VARCHAR2(30) := FND_API.G_MISS_CHAR,
    ATTRIBUTE1                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE2                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE3                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE4                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE5                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE6                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE7                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE8                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE9                      VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE10                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE11                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE12                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE13                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE14                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    ATTRIBUTE15                     VARCHAR2(150) := FND_API.G_MISS_CHAR,
    CONTACT_PARTY_ID                NUMBER := FND_API.G_MISS_NUM
);
```

8.1.5 Lease Lead Count Out Type

```
TYPE SALES_LEAD_CNT_OUT_Rec_Type   IS RECORD(
    LEAD_CONTACT_ID                NUMBER,
    RETURN_STATUS                  VARCHAR2(1)
);
```

8.1.6 Lead Engines Out Type

```
TYPE LEAD_ENGINES_OUT_Rec_Type   IS RECORD(
    qualified_flag                 VARCHAR2(1),
    lead_rank_id                   NUMBER,
    channel_code                   VARCHAR2(30),
    indirect_channel_flag          VARCHAR2(1),
    sales_team_flag                VARCHAR2(1)
);
```

8.2 Standard Parameters for List Generation APIs

There are a number of standard parameters which are common for Leads APIs. Note that all the Standard OUT parameters are required. These parameters are listed in the tables below:

Table 8–2 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version_ number	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_int_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false
p_validation_ level	NUMBER	No	Level of validation required. NONE means no validation will be cone in the API and FULL means all validations (item level, record level) will be performed.
p_check_ access_flag	VARCHAR2	No	Flag to indicate whether to perform access security check or not.
p_admin_flag	VARCHAR2	No	Flag to indicate if the current user is an administrator or not.

Table 8–2 Standard IN Parameters

Parameter	Data Type	Required	Description
p_admin_ group_ID	NUMBER	No	Salesgroup ID of the current user in case they are an administrator.
p_identity_ salesforce_ID	NUMBER	No	Resource ID of the current user.

Table 8–3 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list. If only one error message is returned, the message count will be zero.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

8.3 Sales Lead APIs

8.3.1 Create Sales Lead

This procedure creates a new sales lead with the parameters specified. A unique sales lead ID will be created.

Procedure Specification

```
PROCEDURE Create_sales_lead(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    P_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
```

```
    P_Validation_Level          IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag         IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Flag                IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id            IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id    IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl    I       AS_UTILITY_PUB.Profile_Tbl_Type :=
                                        AS_UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_Rec            IN      SALES_LEAD_Rec_Type :=
                                        G_MISS_SALES_LEAD_REC,
    P_SALES_LEAD_LINE_tbl       IN      SALES_LEAD_LINE_tbl_type
                                        DEFAULT G_MISS_SALES_LEAD_LINE_tbl,
    P_SALES_LEAD_CONTACT_tbl    IN      SALES_LEAD_CONTACT_tbl_type
                                        DEFAULT G_MISS_SALES_LEAD_CONTACT_tbl,
    X_SALES_LEAD_ID             OUT     NUMBER,
    X_SALES_LEAD_LINE_OUT_Tbl   OUT     SALES_LEAD_LINE_OUT_Tbl_Type,
    X_SALES_LEAD_CNT_OUT_Tbl    OUT     SALES_LEAD_CNT_OUT_Tbl_Type,
    X_Return_Status             OUT     VARCHAR2,
    X_Msg_Count                 OUT     NUMBER,
    X_Msg_Data                  OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Notes

1. A unique sales lead ID is generated from the sequence.
2. In P_Sales_Lead_Rec, the required parameters are status_code, customer_id and source_promotion_ID (based on the profile).
3. If P_sales_lead_line_tbl and/or P_sales_lead_contact_tbl is passed in, then the appropriate create APIs for sales lead lines and sales lead contacts is also called after creating the sales lead header.

Table 8–4 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_ profile_tbl	as_utility_ pub.Profile_ Tbl_Type	No	Datatype to store the access security related profile values (they may be cached in the middle tier and passed to the API).

Table 8–4 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_rec	as_sales_leads_pub.Sales_Lead_Rec_Type	No	Sales Lead Record.
p_sales_lead_line_tbl	as_sales_leads_pub.Sales_Lead_Line_Tbl_Type	No	Table of sales lead line records.
p_sales_lead_contact_tbl	as_sales_leads_pub.Sales_Lead_Contact_Tbl_Type	No	Table of sales lead contact records.

Table 8–5 OUT Parameters

Parameter	Data Type	Description
x_sales_lead_ID	NUMBER	Sales lead ID of the sales lead just created.
x_sales_lead_line_out_tbl	as_sales_leads_pub.Sales_Lead_Line_Out_Tbl_Type	Table of sales lead line out record types. Each record type consists of the sales lead line ID created and the return status.
x_sales_lead_contact_out_tbl	as_sales_leads_pub.Sales_Lead_Contact_Out_Tbl_Type	Table of sales lead contact out record types. Each record type consists of the lead contact ID and the return status.

8.3.2 Update Sales Lead

This procedure updates a sales lead with the parameters specified. A valid sales lead ID must be passed in.

Procedure Specification

```
PROCEDURE Update_sales_lead(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
```

```
    P_Commit                    IN      VARCHAR2 := FND_API.G_FALSE,
    P_Validation_Level          IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag         IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Flag                IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id            IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id    IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl    IN      AS_UTILITY_PUB.Profile_Tbl_Type :=
                                        AS_UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_Rec            IN      SALES_LEAD_Rec_Type
                                        DEFAULT G_MISS_SALES_LEAD_REC,
    X_Return_Status             OUT     VARCHAR2,
    X_Msg_Count                 OUT     NUMBER,
    X_Msg_Data                  OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Table 8–6 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_ profile_tbl	as_ utility.pub. Profile_Tbl_ Type	No	Datatype to store the access security related profile values (they may be cached in the middle tier and passed to the API).
p_sales_lead_ rec	as_sales_ leads_pub. Sales_Lead_ Rec_Type	No	Sales lead record type for update.

Table 8–7 OUT Parameters

Parameter	Data Type	Description
Standard OUT parameters		

8.4 Sales Lead Lines APIs

8.4.1 Create Sales Lead Lines

This procedure creates one or more sales lead lines with the parameters specified and associates them with the sales lead ID passed in.

Procedure Specification

```
PROCEDURE Create_sales_lead_lines(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2:= FND_API.G_FALSE,
    P_Commit                     IN      VARCHAR2:= FND_API.G_FALSE,
    p_validation_level           IN      NUMBER:= FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag          IN      VARCHAR2:= FND_API.G_MISS_CHAR,
    P_Admin_Flag                 IN      VARCHAR2:= FND_API.G_MISS_CHAR,
    P_Admin_Group_Id             IN      NUMBER:= FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN      NUMBER:= FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl     IN      AS_UTILITY_PUB.Profile_Tbl_Type :=
                                         AS_UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_LINE_Tbl        IN      SALES_LEAD_LINE_Tbl_Type :=
                                         G_MISS_SALES_LEAD_LINE_Tbl,
    P_SALES_LEAD_ID              IN      NUMBER,
    X_SALES_LEAD_LINE_OUT_Tbl    OUT     SALES_LEAD_LINE_OUT_Tbl_Type,
    X_Return_Status              OUT     VARCHAR2,
    X_Msg_Count                  OUT     NUMBER,
    X_Msg_Data                   OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Notes

1. A unique sales lead line ID is generated from the sequence for each sales lead line.
2. In P_Sales_Lead_Line_Rec, the required parameters are sales_lead_id, source_promotion_id and either product category (interest_type_id, primary_interest_code_id, or secondary_interest_id) or inventory item (inventory_item_id and organization_id).

Table 8–8 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_ profile_tbl	as_utility_ pub.Profile_ Tbl_Type	No	Datatype to store the access security related profile values (they may be cached in the middle tier and passed to the API).
p_sales_lead_id	NUMBER	Yes	Sales lead id for which the sales lead lines are being created.
p_sales_lead_ line_tbl	as_sales_ leads_pub. Sales_Lead_ Line_Tbl_ Type	No	Table of sales lead line records

Table 8–9 OUT Parameters

Parameter	Data Type	Description
x_sales_lead_ line_out_tbl	as_sales_leads_ pub.Sales_Lead_ Line_Out_Tbl_ Type	Table of sales line out record types. Each record type consists of the sales lead line ID and the return status.

8.4.2 Update Sales Lead Lines

This procedure updates one or more sales lead lines with the parameters specified.

Procedure Specification

```
PROCEDURE Update_sales_lead_lines(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    P_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level           IN      NUMBER :=
                                         FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag          IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Flag                 IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id             IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl     IN      AS_UTILITY_PUB.Profile_Tbl_Type :=
                                         AS_UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_LINE_Tbl        IN      SALES_LEAD_LINE_Tbl_Type,
    X_SALES_LEAD_LINE_OUT_Tbl    OUT     SALES_LEAD_LINE_OUT_Tbl_Type,
    X_Return_Status              OUT     VARCHAR2,
```

```
    X_Msg_Count                 OUT     NUMBER,
    X_Msg_Data                  OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Notes

1. In p_sales_lead_line_rec, the required parameters are sales_lead_id, source_promotion_id and either product category (interest_type_id, primary_interest_code_id, or secondary_interest_code_id) or inventory item (inventory_item_id and organization_id).
2. Last_update_date must be passed in for each sales lead line being updated.

Table 8–10 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_profile_tbl	as_utility.pub.Profile_Tbl_Type	No	Datatype to store the access security related profile values (they may be cached in the middle tier and passed to the API).
p_sales_lead_line_tbl	as_sales_leads_pub.Sales_lead_Line_Tbl_Type	No	Table of sales lead line records.

Table 8–11 OUT Parameters

Parameter	Data Type	Description
x_sales_lead_line_out_tbl	as_sales_leads_pub.Sales_Lead_Line_Out_Tbl_Type	Table of sales lead line out record types. Each record type consists of the sales lead line ID processed and the return status.

8.4.3 Delete Sales Lead Lines

This procedure deletes one or more sales lead lines.

Procedure Specification

```
PROCEDURE Delete_sales_lead_lines(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    P_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level           IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag          IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Flag                 IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id             IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl     IN      AS_UTILITY_PUB.Profile_Tbl_Type :=
                                         AS_UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_LINE_Tbl        IN      SALES_LEAD_LINE_Tbl_type,
    X_SALES_LEAD_LINE_OUT_Tbl    OUT     SALES_LEAD_LINE_OUT_Tbl_Type,
    X_Return_Status              OUT     VARCHAR2,
    X_Msg_Count                  OUT     NUMBER,
    X_Msg_Data                   OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Notes

1. In p_sales_lead_line_rec, the required parameters are sales_lead_id.

Table 8–12 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_ profile_tbl	as_ utility.pub. Profile_Tbl_ Type	No	Datatype to store the access security related profile values (they may be cached in the middle tier and passed to the API).
p_sales_lead_ line_tbl	as_sales_ leads_pub. Sales_lead_ Line_Tbl_ Type	No	Table of sales lead line records.

Table 8–13 OUT Parameters

Parameter	Data Type	Description
x_sales_lead_ line_out_tbl	as_sales_leads_ pub.Sales_Lead_ Line_Out_Tbl_ Type	Table of sales lead line out record types. Each record type consists of the sales lead line ID processed and the return status.

8.5 Sales Lead Contact APIs

8.5.1 Create Sales Lead Contacts

This procedure creates one or more Sales Lead contacts for an existing sales lead.

Procedure Specification

```
PROCEDURE Create_sales_lead_contacts(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    P_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level           IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag          IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Flag                 IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id             IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl     IN      AS_UTILITY_PUB.Profile_Tbl_Type :=
                                         AS_UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_CONTACT_Tbl     IN      SALES_LEAD_CONTACT_Tbl_Type :=
                                         G_MISS_SALES_LEAD_CONTACT_Tbl,
    p_SALES_LEAD_ID              IN      NUMBER,
    X_SALES_LEAD_CNT_OUT_Tbl     OUT     SALES_LEAD_CNT_OUT_Tbl_Type,
    X_Return_Status              OUT     VARCHAR2,
    X_Msg_Count                  OUT     NUMBER,
    X_Msg_Data                   OUT     VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Table 8–14 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_ profile_tbl	as_utility_ pub.Profile_ Tbl_Type	No	Datatype to store the access security related profile values (they may be cached in the middle tier and passed to the API).
p_sales_lead_ line_tbl	as_sales_ leads_pub. Sales_Lead_ Line_Tbl_ Type	No	Table of sales lead line records.

Table 8–15 OUT Parameters

Parameter	Data Type	Description
x_sales_lead_ cnt_out_tbl	Sales_Lead_Cnt_ Out_Tbl_Type	Contains the record which in turn contains the Lead Contact ID for the sales lead.

8.5.2 Update Sales Lead Contacts

This procedure is used to update sales contact information in the table. the API will raise an exception if the record matching the Sales Lead Contact ID, and the object version number passed into it, do not exist.

Procedure Specification

```
PROCEDURE Update_sales_lead_contacts(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    P_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level           IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag          IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Flag                 IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id             IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl     IN      AS_UTILITY_PUB.Profile_Tbl_Type :=
                                         AS_UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_CONTACT_Tbl     IN      SALES_LEAD_CONTACT_Tbl_Type,
    X_SALES_LEAD_CNT_OUT_Tbl     OUT     SALES_LEAD_CNT_OUT_Tbl_Type,
    X_Return_Status              OUT     VARCHAR2,
    X_Msg_Count                  OUT     NUMBER,
```

```
    X_Msg_Data                  OUT     VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. The API will raise an exception if the object version number does not match.

Table 8–16 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_ profile_tbl	as_utility_ pub.Profile_ Tbl_Type	No	Datatype to store the access security related profile values (they may be cached in the middle tier and passed to the API).
p_sales_lead_id	NUMBER	Y	The unique identifier of the sales lead.

Table 8–17 OUT Parameters

Parameter	Data Type	Description
x_sales_lead_ cnt_out_tbl	Sales_Lead_cnt_ Out_Tbl_Type	Contains the record which in turn contains the Lead Contact ID for the sales lead.

8.5.3 Delete Sales Lead Contacts

This procedure calls table handler Delete_Sales_Lead_Contacts and then calls a procedure to update As_Sales_Lead_Contact table if the primary contact is deleted and another contact is marked as primary.

Procedure Specification

```
PROCEDURE Delete_sales_lead_contacts(
    P_Api_Version_Number        IN      NUMBER,
    P_Init_Msg_List             IN      VARCHAR2 := FND_API.G_FALSE,
    P_Commit                    IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level          IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Check_Access_Flag         IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Flag                IN      VARCHAR2 := FND_API.G_MISS_CHAR,
    P_Admin_Group_Id            IN      NUMBER := FND_API.G_MISS_NUM,
```

```
    P_identity_salesforce_id     IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Profile_Tbl     IN      AS_UTILITY_PUB.Profile_Tbl_Type :=
                                         AS_UTILITY_PUB.G_MISS_PROFILE_TBL,
    P_SALES_LEAD_CONTACT_Tbl     IN      SALES_LEAD_CONTACT_Tbl_type,
    X_SALES_LEAD_CNT_OUT_Tbl     OUT     SALES_LEAD_CNT_OUT_Tbl_Type,
    X_Return_Status              OUT     VARCHAR2,
    X_Msg_Count                  OUT     NUMBER,
    X_Msg_Data                   OUT     VARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. The API will raise an exception if the object version number does not match.

Table 8–18 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_ profile_tbl	as_utility_ pub.Profile_ Tbl_Type	No	Datatype to store the access security related profile values (they may be cached in the middle tier and passed to the API).
p_sales_lead_id	NUMBER	Y	The unique identifier of the sales lead.

Table 8–19 OUT Parameters

Parameter	Data Type	Description
x_sales_lead_ cnt_out_tbl	Sales_Lead_cnt_ Out_Tbl_Type	Contains the record which in turn contains the Lead Contact ID for the sales lead.

8.6 Sales Lead Score APIs

8.6.1 Get Sales Lead Score

This procedure calculates and returns the score and rank of the Sales Lead based on the score card.

Procedure Specification

```
Procedure Get_Score (
    p_api_version            IN    NUMBER := 2.0,
    p_init_msg_list          IN    VARCHAR2 := FND_API.G_FALSE,
    p_commit                 IN    VARCHAR2 := FND_API.G_FALSE,
    p_validation_level       IN    NUMBER := AS_UTILITY_PUB.G_VALID_LEVEL_ITEM,
    P_Check_Access_Flag      IN    VARCHAR2 := FND_API.G_MISS_CHAR,
    p_sales_lead_id          IN    NUMBER,
    p_scorecard_id           IN    NUMBER,
    p_marketing_score        IN    NUMBER := 0,
    p_identity_salesforce_id IN    NUMBER,
    p_admin_flag             IN    VARCHAR2,
    p_admin_group_id         IN    NUMBER,
    x_rank_id                      OUTNUMBER,
    X_SCORE                        OUTNUMBER,
    x_return_status          OUT   VARCHAR2,
    x_msg_count                    OUTNUMBER,
    x_msg_data                     OUTVARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Table 8–20 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_id	NUMBER	Yes	The unique identifier of the sales lead.
p_scorecard_id	NUMBER	Yes	The unique identifier of the score card.
p_marketing_ score	NUMBER	No	The additional score which could be used to alter the total score of the sales lead in turn to change the rank of the sales lead.

Table 8–21 OUT Parameters

Parameter	Data Type	Description
x_rank_id	NUMBER	The rank ID of the Sales Lead.
x_score	NUMBER	The total score of the Sales Lead.

8.6.2 Run Lead Engines

This API should be called after lead is created, or "Run Engine" button is clicked. If user doesn't specify a qualified flag, rank, or sales channel, and the profile setting is to do them automatically, this API will run the qualification, rating and channel selection engines. In addition, if the user sets the profile to run 11.5.7 engines, it will run 11.5.7 qualification and ranking engines as well.

Procedure Specification

```
PROCEDURE Run_Lead_Engines (
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    p_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
    p_Validation_Level           IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Admin_Group_Id             IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN      NUMBER := FND_API.G_MISS_NUM,
    P_Salesgroup_id              IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Id              IN      NUMBER,
    X_Lead_Engines_Out_Rec       OUT     LEAD_ENGINES_OUT_Rec_Type,
    X_Return_Status              OUT     VARCHAR2,
    X_Msg_Count                  OUT     NUMBER,
    X_Msg_Data                   OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Notes

1. The API_version_number will be set to 2.0
2. If p_salesgroup_id is not passed in, this API will find a group_id for the current user.
3. If a flag column is passed in, it will be validated for "Y" or "N" and an exception will be raised for an invalid flag.
4. If a flag column is not passed in, it will be defaulted to "Y" or "N".

Table 8–22 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_id	NUMBER	Yes	The Sales Lead ID for which the user wants to build a sales team.

Table 8–23 OUT Parameters

Parameter	Data Type	Description
x_lead_ engines_out_rec	Lead_Engines_ Out_Rec_Type	Result of qualification, rating, and channel selection engines.

8.7 Lead Sales Team APIs

8.7.1 Build Lead Sales Team

This API should be called after Run_Lead_Engines API is called. It builds lead sales team based on territory definitions and adds lead creator as one of lead sales team members.

Procedure Specification

```
PROCEDURE Build_Lead_Sales_Team(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    p_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
    p_Validation_Level           IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Admin_Group_Id             IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN      NUMBER := FND_API.G_MISS_NUM,
    P_Salesgroup_id              IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Id              IN      NUMBER,
    X_Return_Status              OUT     VARCHAR2,
    X_Msg_Count                  OUT     NUMBER,
    X_Msg_Data                   OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Notes

1. The API_version_number will be set to 2.0
2. If p_salesgroup_id is not passed in, this API will find a group_id for the current user.
3. If a flag column is passed in, it will be validated for "Y" or "N" and an exception will be raised for an invalid flag.
4. If a flag column is not passed in, it will be defaulted to "Y" or "N".

Table 8–24 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_id	NUMBER	Yes	The Sales Lead ID for which the user wants to build a sales team.

Table 8–25 OUT Parameters

Parameter	Data Type	Description
Standard OUT Parameters		

8.7.2 Rebuild Lead Sales Team

This API should be called after lead header is updated and lines are created/updated/deleted. When user does change to the lead, the lead may not match the territory it originally met, and match other territories. This API will rebuild lead sales team to reflect the latest lead information.

Procedure Specification

```
PROCEDURE Rebuild_Lead_Sales_Team(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    p_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
    p_Validation_Level           IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Admin_Group_Id             IN      NUMBER := FND_API.G_MISS_NUM,
    P_identity_salesforce_id     IN      NUMBER := FND_API.G_MISS_NUM,
    P_Salesgroup_id              IN      NUMBER := FND_API.G_MISS_NUM,
    P_Sales_Lead_Id              IN      NUMBER,
    X_Return_Status              OUT     VARCHAR2,
```

```
    X_Msg_Count                 OUT     NUMBER,
    X_Msg_Data                  OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Notes

1. The API_version_number will be set to 2.0
2. If p_salesgroup_id is not passed in, this API will find a group_id for the current user.
3. If a flag column is passed in, it will be validated for "Y" or "N" and an exception will be raised for an invalid flag.
4. If a flag column is not passed in, it will be defaulted to "Y" or "N".

Table 8–26 IN Parameters

Parameter	Data Type	Required	Description
p_sales_lead_id	NUMBER	Yes	The Sales Lead ID for which the user wants to build a sales team.

Table 8–27 OUT Parameters

Parameter	Data Type	Description
Standard OUT Parameters		

8.8 Partner Matching APIs

8.8.1 Start Partner Matching

This API is a workflow to do partner matching. Since partner matching may take a long time, it defers the process and exits immediately so a user won't have to wait until the partner matching process completes.

Procedure Specification

```
PROCEDURE Start_Partner_Matching(
    P_Api_Version_Number         IN      NUMBER,
    P_Init_Msg_List              IN      VARCHAR2 := FND_API.G_FALSE,
    P_Commit                     IN      VARCHAR2 := FND_API.G_FALSE,
    P_Validation_Level           IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    P_Admin_Group_Id             IN      NUMBER:= FND_API.G_MISS_NUM,
    P_Identity_Salesforce_Id     IN      NUMBER,
    P_Salesgroup_Id              IN      NUMBER,
    P_Lead_id                    IN      NUMBER,
    X_Return_Status              OUT     VARCHAR2,
    X_Msg_Count                  OUT     NUMBER,
    X_Msg_Data                   OUT     VARCHAR2
);
```

Current Version

2.0

Parameter Descriptions

Notes

1. The API_version_number will be set to 2.0
2. If p_salesgroup_id is not passed in, this API will find a group_id for the current user.
3. If a flag column is passed in, it will be validated for "Y" or "N" and an exception will be raised for an invalid flag.
4. If a flag column is not passed in, it will be defaulted to "Y" or "N".

Table 8–28 IN Parameters

Parameter	Data Type	Required	Description
p_lead_id	NUMBER	Yes	Lead identifier for which the user wants to perform partner matching.

9

Event Registration

The public APIs for Event Registration provides a number of procedures for registration actions.

The procedures which make up the Event Registration APIs are:

Table 9–1 Leads APIs

Procedure	Description
Register	Registers a person for Event, given a party id for Registrant and Attendent.
Initialize Registration Record	Initialize the Registrant details record. Used internally.
Update Registration	Update the registration information.
Cancel Registration	Cancel the registration.
Delete Registration	Deprecated. Can be used to delete registrations.
Prioritize Registration Waitlist	Given Event id , prioritize the waitlist. (Get the lowest waitlist, find if any seat available, if yes, register and move waitlist up)
Substitute Enrollee	Given a confirmation number, update the Registrant and attendent information.
Transfer Enrollee	Transfer the registrant and attendent to another event.
Get Registration Record	Get the registration record to do above registration processes.

9.1 Event Registration User Hooks

See the following table for Event Registration User Hook information.

Table 9–2 Event Registration User Hooks

Procedure	Parameter
AMS_EvtRegs_ PUB.Register	AMS_EvtRegs_CUHK.register_pre
AMS_EvtRegs_ PUB.Register	AMS_EvtRegs_CUHK.register_post
AMS_EvtRegs_ PUB.Update_registration	AMS_EvtRegs_CUHK.Update_registration_pre
AMS_EvtRegs_ PUB.Update_registration	AMS_EvtRegs_CUHK.Update_registration_post
AMS_EvtRegs_ PUB.Delete_Registration	AMS_EvtRegs_CUHK.Delete_Registration_pre
AMS_EvtRegs_ PUB.Delete_Registration	AMS_EvtRegs_CUHK.Delete_Registration_post
AMS_EvtRegs_ PUB.Prioritize_Reg_ Wailist	AMS_EvtRegs_CUHK.Prioritize_reg_wailist_pre
AMS_EvtRegs_ PUB.Prioritize_Reg_ Wailist	AMS_EvtRegs_CUHK.Prioritize_reg_wailist_post
AMS_EvtRegs_ PUB.Substitute_Enrollee	AMS_EvtRegs_CUHK.Substitute_enrollee_pre
AMS_EvtRegs_ PUB.Substitute_Enrollee	AMS_EvtRegs_CUHK.Substitute_enrollee_post
AMS_EvtRegs_ PUB.Transfer_Enrollee	AMS_EvtRegs_CUHK.Transfer_enrollee_pre
AMS_EvtRegs_ PUB.Transfer_Enrollee	AMS_EvtRegs_CUHK.Transfer_enrollee_post

9.2 Type Declarations

This section defines the registration record type declaration. Registration record type is used as an IN parameter in Register and Update_Registration API. The actual declaration of the record type resides in a private api. Hence the record type is refered to as AMS_EvtRegs_PVT.evt_regs_Rec_Type.

.

```
TYPE evt_regs_Rec_Type IS RECORD
(
       EVENT_REGISTRATION_ID           NUMBER,
       LAST_UPDATE_DATE                DATE,
       LAST_UPDATED_BY                 NUMBER ,
       CREATION_DATE                   DATE ,
       CREATED_BY                      NUMBER ,
       LAST_UPDATE_LOGIN               NUMBER ,
       OBJECT_VERSION_NUMBER           NUMBER ,
       EVENT_OFFER_ID                  NUMBER ,
       APPLICATION_ID                  NUMBER,
       ACTIVE_FLAG                     VARCHAR2(1),
       OWNER_USER_ID                   NUMBER ,
       SYSTEM_STATUS_CODE              VARCHAR2(30),
       DATE_REGISTRATION_PLACED        DATE ,
       USER_STATUS_ID                  NUMBER ,
       LAST_REG_STATUS_DATE            DATE ,
       REG_SOURCE_TYPE_CODE            VARCHAR2(30) ,
       REGISTRATION_SOURCE_ID          NUMBER ,
       CONFIRMATION_CODE               VARCHAR2(30),
       SOURCE_CODE                     VARCHAR2(30),
       REGISTRATION_GROUP_ID           NUMBER ,
       REGISTRANT_PARTY_ID             NUMBER ,
       REGISTRANT_CONTACT_ID           NUMBER ,
       REGISTRANT_ACCOUNT_ID           NUMBER,
       ATTENDANT_PARTY_ID              NUMBER ,
       ATTENDANT_CONTACT_ID            NUMBER ,
       ATTENDANT_ACCOUNT_ID            NUMBER,
       ORIGINAL_REGISTRANT_CONTACT_ID  NUMBER,
       PROSPECT_FLAG                   VARCHAR2(1) ,
       ATTENDED_FLAG                   VARCHAR2(1) ,
       CONFIRMED_FLAG                  VARCHAR2(1) ,
       EVALUATED_FLAG                  VARCHAR2(1) ,
       ATTENDANCE_RESULT_CODE          VARCHAR2(4000) ,
       WAITLISTED_PRIORITY             NUMBER ,
       TARGET_LIST_ID                  NUMBER ,
       INBOUND_MEDIA_ID                NUMBER ,
       INBOUND_CHANNEL_ID              NUMBER ,
       CANCELLATION_CODE               VARCHAR2(30) ,
       CANCELLATION_REASON_CODE        VARCHAR2(30) ,
       ATTENDANCE_FAILURE_REASON       VARCHAR2(30) ,
       ATTENDANT_LANGUAGE              VARCHAR2(4) ,
       SALESREP_ID                     NUMBER ,
       ORDER_HEADER_ID                 NUMBER ,
       ORDER_LINE_ID                   NUMBER ,
```

```
    DESCRIPTION                     VARCHAR2(4000) ,
    MAX_ATTENDEE_OVERRIDE_FLAG      VARCHAR2(1),
    INVITE_ONLY_OVERRIDE_FLAG       VARCHAR2(1),
    PAYMENT_STATUS_CODE             VARCHAR2(30),
    AUTO_REGISTER_FLAG              VARCHAR2(1),
    ATTRIBUTE_CATEGORY              VARCHAR2(30) ,
    ATTRIBUTE1                      VARCHAR2(150) ,
    ATTRIBUTE2                      VARCHAR2(150) ,
    ATTRIBUTE3                      VARCHAR2(150) ,
    ATTRIBUTE4                      VARCHAR2(150) ,
    ATTRIBUTE5                      VARCHAR2(150) ,
    ATTRIBUTE6                      VARCHAR2(150) ,
    ATTRIBUTE7                      VARCHAR2(150) ,
    ATTRIBUTE8                      VARCHAR2(150) ,
    ATTRIBUTE9                      VARCHAR2(150) ,
    ATTRIBUTE10                     VARCHAR2(150) ,
    ATTRIBUTE11                     VARCHAR2(150) ,
    ATTRIBUTE12                     VARCHAR2(150) ,
    ATTRIBUTE13                     VARCHAR2(150) ,
    ATTRIBUTE14                     VARCHAR2(150) ,
    ATTRIBUTE15                     VARCHAR2(150) ,
    ATTENDEE_ROLE_TYPE              VARCHAR2(30) ,
    NOTIFICATION_TYPE               VARCHAR2(30) ,
    LAST_NOTIFIED_TIME              DATE,
    EVENT_JOIN_TIME                 DATE,
    EVENT_EXIT_TIME                 DATE,
    MEETING_ENCRYPTION_KEY_CODE     VARCHAR2(150)

);
```

9.3 Standard Parameters for List Generation APIs

There are a number of standard parameters which are common for all of the following registration APIs. Note that all the Standard OUT parameters are required.

The parameters are listed in the tables below::

Table 9–3 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility .

Table 9–3 Standard IN Parameters

Parameter	Data Type	Required	Description
p_init_msg_list	VarChar2	No	Flag to indicate if the message stack should be initialized. Default : FND_API.g_false
p_commit	VarChar2	No	Flag to indicate if the message stack should be initialized. Default : FND_API.g_false
p_commit	VarChar2	No	Flag to indicate if the changes should be commited on success. Default : FND_API.g_false

Table 9–4 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	Return status of the API. Can be 'S' Success, 'E' Expected error or 'U' Undefined exception.
x_msg_count	Number	Number of error messages returned by the api. If the error message returned is one then the message count will be zero.
x_msg_data	VarChar2	Error message returned by the API. If the messages returned are more than one, this parameter will be null and messages have to be extracted from the message stack.

9.4 Registration APIs

This procedure will register the person for the event offer. If the capacity of the event offer allows , the person will be waitlisted. If the capacity is full and the waitlist is allowed for the event offer, the person will be placed on the waitlist. If there is no place to book for the event offer, the api will return error. Upon successful registration , API will return the registration confirmation code.

9.4.1 Procedure Specification

```
PROCEDURE Register(
P_Api_Version_NumberINNUMBER,
P_Init_Msg_ListINVARCHAR2 := FND_API.G_FALSE,
P_CommitINVARCHAR2 := FND_API.G_FALSE,
P_evt_regs_RecINAMS_EvtRegs_PVT.evt_regs_Rec_Type,
x_event_registration_idOUTNUMBER,
x_confirmation_codeOUTVARCHAR2,
```

```
x_system_status_codeOUTVARCHAR2,
X_Return_StatusOUTVARCHAR2,
X_Msg_CountOUTNUMBER,
X_Msg_DataOUTVARCHAR2
);
```

9.4.2 Parameter Description

1. Object_version_number will set to 1.
2. If the registration id is passed in, the uniqueness will be checked. If unique, the registration will be created by this id, else exception will be thrown due to duplicates.
3. If the registration id is not passed, system will generate unique id for registration.
4. Flag columns will be checked for 'Y' or 'N'. If nothing is passed, it will be defaulted to 'Y' or 'N' as appropriate.
5. Please don't pass in any FND_API.g_miss_char/num/date for creation.
6. There is out of box fulfillment capability for sending confirmation letter. To enable this, set the profile "AMS : Enable Fulfillment" to Yes. Specific event registration can be blocked by passing p_block fulfillment as 'T' (Default is 'F')
7. On Registration, Interaction history will be logged for the registrant using attendant party id.
8. If the event offer is invite only, the validation will be done to check if the registrant is in the invite list. If not error will be thrown.
9. You can only register for the event if the event is not frozen for registration and if the event requires registration. API will return error if either of these conditions are true.

Table 9–5 IN Parameter

Parameter	Data Type	Required	Description
p_api_version_ number	Number	Yes	See "Standard In Parameters
p_init_msg_list	VarChar	No	See Standard In Parameters
p_commit	VarChar	No	See Stardard In Parameters

Parameter	Data Type	Required	Description
p_validation_ level	Number	No	Default = FND_API.G_VALID_LEVEL_ FULLLevel of validation required.None means no validation will be done in the API and Full means all the validations (item level and record level)will be performed.
p_evt_regs_rec	AMS_ EvtRegs_ PVT.evt_ regs_Rec_ Type	Yes	Record type for registration. The record will be validated for 1. Required parameters 2. Unique key validations 3. foreign key validations 4. lookup validations 5. flags.
p_block_ fulfillment	VarChar2	No	Default : F (False). If set to T (True), then the api will not call fulfillment to send confirmation letter.

Table 9–6 Standard OUT parameters

Parameter	Data Type	Description
x_return_status	VarChar2	Indicates the return status of the API.The values returned are one of the following:FND_API.G_RET_ STS_SUCCESS which indicates the API call was successful.FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error.FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	Number	Holds the number of messages in the messa e list.
x_msg_data	VarChar2	Error message returned by the API.If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.
x_event_ registration_id	Number	Returns the id of the registration created.
x_ confirmation_ code	VarChar2	Unique system generated confirmation code for registration on successful registration confirmation/ waitlist confirmation.
x_system_ status_code	VarChar2	Status code to indicate if the person is registered or put on waitlist.

9.4.3 Initialization Registration Record

Call this method to initialize the reg rec before calling Update_registration.

Procedure Specification

```
PROCEDURE init_reg_rec(
x_evt_regs_recOUTAMS_EvtRegs_PVT.evt_regs_Rec_Type
);
```

Current Version

1.0

Parameter Descriptions

Table 9–7 IN Parameters

Parameter	Data Type	Required	Description
None			

Table 9–8 OUT Parameters

Parameter	Data Type	Description
x_evt_regs_rec		AMS_EvtRegs_PVT.evt_regs_Rec_Type

9.4.4 Update Registration

This procedure updates the Event Registration with the parameters specified.

Procedure Specification

```
PROCEDURE Update_registration(
P_Api_Version_NumberINNUMBER,
P_Init_Msg_ListINVARCHAR2 := FND_API.G_FALSE,
P_CommitINVARCHAR2 := FND_API.G_FALSE,
P_evt_regs_RecINAMS_EvtRegs_PVT.evt_regs_Rec_Type,
X_Return_StatusOUTVARCHAR2,
X_Msg_CountOUTNUMBER,
X_Msg_DataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Table 9–9 IN Parameters

Parameter	Data Type	Required	Description
p_api_version_ number	Number	Yes	See standard IN parameters.
p_init_msg_list	VarChar2	No	See standard IN parameters.
p_commit	VarChar2	Optional	See standard IN parameters.
p_evt_regs_Rec		Yes	AMS_EvtRegs_PVT.evt_regs_Rec_Type

Table 9–10 OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	See standard OUT parameters.
x_msg_count	Number	See standard OUT parameters.
x_msg_data	VarChar2	See standard OUT parameters.

9.4.5 Cancel Registration

Use this API to cancel the registration for the event. API will cancel the registration and will return the cancellation code. When the slot is cancelled for the event, api will also reprioritize the waitlist and check if any waitlisted slot get confirmed.

Procedure Specification

```
PROCEDURE Cancel_Registration(
P_Api_Version_NumberINNUMBER,
P_Init_Msg_ListINVARCHAR2 := FND_API.G_FALSE,
P_CommitINVARCHAR2 := FND_API.G_FALSE,
p_object_versionINNUMBER,
p_event_offer_idINNUMBER,
p_registrant_party_idINNUMBER,
p_confirmation_codeINVARCHAR2,
p_registration_group_idINNUMBER,
p_cancellation_reason_codeINVARCHAR2,
x_cancellation_codeOUTVARCHAR2,
```

```
X_Return_StatusOUTVARCHAR2,
X_Msg_CountOUTNUMBER,
X_Msg_DataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

1. API will raise exception if the object_version_number doesn't match.
2. Confirmation code should be given if the cancellation is not for the group. If the group registration has to be cancelled, the confirmation code will be derived internally.

Notes

Table 9–11 IN Parameters

Parameter	Data Type	Required	Description
p_api_version number	Number	Yes	See standard IN parameters.
p_init_msg_list	VarChar2	No	See standard IN parameters.
p_commit	VarChar2	No	See standard IN parameters.
p_object_ version	Number	Yes	Object version number , used for locking.
p_event_offer_ id	Number	Yes	Identifier for the event for which you have to cancel registration.
p_registrant_ party_id	Number	Yes	Identifier of the registrant party.
p_ confirmation_ code	VarChar2	No	Confirmation code for the registration.

Table 9–11 IN Parameters

Parameter	Data Type	Required	Description
p_registration_group_id	Number	Yes	Group id if the group registration have to be cancelled.
p_cancellation_reason_code	VarChar2	No	Reason for Cancellation

Table 9–12 OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	See standard OUT parameters.
x_msg_count	Number	See standard OUT parameters.
x_msg_data	VarChar2	See standard OUT parameters.
x_cancellation_code	VarChar2	Cancellation code to confirm cancellation.

9.4.6 Delete Registration

This procedure deletes registration.

Procedure Specification

```
PROCEDURE delete_Registration(
P_Api_Version_NumberINNUMBER,
P_Init_Msg_ListINVARCHAR2 := FND_API.G_FALSE,
P_CommitINVARCHAR2 := FND_API.G_FALSE,
p_object_versionINNUMBER,
p_event_registration_idINNUMBER,
X_Return_StatusOUTVARCHAR2,
X_Msg_CountOUTNUMBER,
X_Msg_DataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. When deleted the active flag will be marked in the registrations table.
2. The api is deprecated.

Table 9–13 IN Parameters

Parameter	Data Type	Required	Description
p_api_version_number	Number	Yes	See standard IN parameters.
p_init_msg_list	VarChar2	No	See standard IN parameters.
p_commit	VarChar2	No	See standard OUT parameters.
p_object_version	Number	Yes	Object version number, use for locking
p_event_registration_id	Numer	Yes	Event Registration id

Table 9–14 OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	See standard OUT parameters.
x_msg_count	Number	See standard OUT parameters.
x_msg_data	VarChar2	See standard OUT parameters.

9.4.7 Prioritize Registration Waitlist

This procedure prioritizes the registration waitlist.

Procedure Specification

```
PROCEDURE prioritize_reg_wailist(
p_api_version_numberINNUMBER,
p_Init_Msg_ListINVARCHAR2 := FND_API.G_FALSE,
P_CommitINVARCHAR2 := FND_API.G_FALSE,
p_event_offer_idINNUMBER,
x_return_statusOUTVARCHAR2,
x_msg_countOUTNUMBER,
```

```
x_msg_dataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

This api will be called internally from Cancel Registration api to check if the new slots are available after cancellation.

Table 9–15 IN Parameters

Parameter	Data Type	Required	Description
p_api_version_number	Number	Yes	See standard IN parameters.
p_init_msg_list	VarChar2	No	See standard IN parameters.
p_event_offer_id	Number	Yes	Event offer id.

Table 9–16 OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	See standard OUT parameters.
x_msg_count	Number	See standard OUT parameters.
x_msg_data	VarChar2	See standard OUT parameters.

9.4.8 Substitute Enrollee

This api will substitute an enrollee (attendee) for an existing event registration. If registrant information is also provided, the existing registrant information is replaced. Attendant information is mandatory, but for account information if registrant information is changed, reg_contact id is stored in original_re_contact_id column.

Procedure Specification

```
P_Api_Version_NumberINNUMBER,
P_Init_Msg_ListINVARCHAR2 := FND_API.G_FALSE,
P_CommitINVARCHAR2 := FND_API.G_FALSE,
p_confirmation_codeINVARCHAR2,
```

```
p_attendant_party_idINNUMBER,
p_attendant_contact_idINNUMBER,
p_attendant_account_idINNUMBER,
p_registrant_party_idINNUMBER,
p_registrant_contact_idINNUMBER,
p_registrant_account_idINNUMBER,
X_Return_StatusOUTVARCHAR2,
X_Msg_CountOUTNUMBER,
X_Msg_DataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Table 9–17 IN Parameters

Parameter	Data Type	Required	Description
p_api_version_ number	Number	Yes	See standard IN parameters.
p_init_msg_list	VarChar	No	See standard IN parameters.
p_commit	VarChar2	No	See standard IN parameters.
p_ confirmation_ code	VarChar	Yes	Confirmation registration code.
p_attendant_ party_ id	Number	Yes	Attendant Party ID
p_attendant_ contact_id	Number	Yes	Contact id for the attendent. Can be same as party id of the attendent.
p_attendant_ account_id	Number	Yes	Account ID for the attendant.
p_registrant_ party_ id	Number	Yes	Registrant Party ID.
p_registrant_ contact_id	Number	Yes	Registrant Contact id, Can be same as the party id of the registrant.
p_registrant_ account_id	Number	Yes	Registrant Account ID.

Table 9–18 OUT Parameters

Parameter	Data Type	Description
x_return_status	VarChar2	See standard OUT parameters.
x_msg_count	Number	See standard OUT parameters.
x_msg_data	VarChar2	See standard OUT parameters.

9.4.9 Transfer Enrollee

Use this api to Transfer an enrollee for an existing event registration to another event. The Waitlist flag input is mandatory which means if the other offering is full and the attendant is willing to get waitlisted they will be transferred. If they are not willing to be waitlisted and the event is full, or if the event and the event waitlist are full, the transfer will fail.

Procedure Specification

```
PROCEDURE transfer_enrollee(
P_Api_Version_NumberINNUMBER,
P_Init_Msg_ListINVARCHAR2 := FND_API.G_FALSE,
P_CommitINVARCHAR2 := FND_API.G_FALSE,
p_object_versionINNUMBER,
p_old_confirmation_codeINVARCHAR2,
p_old_offer_idINNUMBER,
p_new_offer_idINNUMBER,
p_waitlist_flagINVARCHAR2,
p_registrant_account_idINNUMBER,
p_registrant_party_idINNUMBER,
p_registrant_contact_idINNUMBER,
p_attendant_party_idINNUMBER,
p_attendant_contact_idINNUMBER,
x_new_confirmation_codeOUTVARCHAR2,
x_old_cancellation_codeOUTVARCHAR2,
x_new_registration_idOUTNUMBER,
x_old_system_status_codeOUTVARCHAR2,
x_new_system_status_codeOUTVARCHAR2,
X_Return_StatusOUTVARCHAR2,
X_Msg_CountOUTNUMBER,
X_Msg_DataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

When transferred the registration for the existing event will be cancelled and the new registration will be created for the new event.

The API will raise an exception if the object version number does not match.

Table 9–19 IN Parameters

Parameter	Data Type	Required	Description
p_api_version_ number	Number	Yes	See standard IN parameters.
p_init_msg_list	VarChar2	No	See standard IN parameters.
p_commit	VarChar2	No	See standard IN parameters.
p_object_ version	Number	No	See standard IN parameters.
p_old_ confirmation_ code	VarChar2	Yes	Confirmation code for the old event offer.
p_old_offer_id	Number	Yes	Old offer id.
p_new_offer_id	Number	Yes	New offer id.
p_waitlist_flag	VarChar2	Number	Flag to indicate if the attendee wish to go waitlisted for another event after the transfer.
p_registrant_ account_id	Number	No	Account ID for registrant.
p_registrant_ party_ id	Number	No	Party id for registrant.
p_registrant_ contact_id	Number	No	Registrant contact id.
p_attendant_ party_id	Number	No	Attendant party id.
p_attendant_ contact_id	Number	No	Attendant contact id.

Table 9–20 OUT Parameters

Parameter	Data Type	Description
x_new_ confirmation_ code	VarChar2	New confirmation code for the event.
x_old_ cancellation_ code	VarChar2	Cancellation code for old event.
x_new_ registration_id	VarChar2	New Registration id.
x_old_system_ status_code	VarChar2	Old system status code.
x_new_system_ status_code	VarChar2	New system status code.
x_return_status	VarChar2	See standard OUT parameters.
x_msg_count	Number	See standard OUT parameters.
x_msg_data	VarChar2	See standard OUT parameters.

9.4.10 Get Registration Record

Call this api to get the registration record. This record can be used to create or update the registrations.

Procedure Specification

PROCEDURE GET_Reg_Rec RETURN AMS_EvtRegs_PVT.evt_regs_Rec_Type;

Current Version

1.0

Parameter Descriptions

No parameters required.

10

Events

The APIs for Event provide a number of procedures for event actions.

The procedures which make up the Event APIs are:

Table 10–1 Event APIs

Procedure	Description
Create Event	Creates a new event in which (a) the object version is set to one, (b) a unique event ID will be created if a unique event ID is not passed in, and (c) a flag column will be set to Y or N, depending on existence of optional parameters.
Delete Event	Sets an event to inactive rather than removing it from the database. Will raise an exception if the object version doesn't match the database record.
Lock Event	Locks the given event record. Will raise an exception if the object version doesn't match the database record.
Update Event	Updates the event record. The values which are not changed can be passed as g_miss record and will not be updated. Will raise an exception if the object version doesn't match the database record.
Validate Event	Validate different business rules like checking not null columns, valid flag values, and foreign key validation. In addition it also do other business validation. The p_evh_rec parameter should be the complete event record.

10.1 Type Declaration

The Event record type will be used as input parameters in some of the APIs. The record type will be initialized to g_miss values before used for updating. The actual

definition of the record type resides in the private API, hence the record type is referred to as AMS_EventHeader_PVT.evh_rec_type.

```
TYPE evh_rec_type IS RECORD(
    event_header_idNUMBER,
    last_update_dateDATE,
    last_updated_byNUMBER,
    creation_dateDATE,
    created_byNUMBER,
    last_update_loginNUMBER,
    object_version_numberNUMBER,
    event_levelVARCHAR2(30),
    application_idNUMBER,
    event_type_codeVARCHAR2(30),
    active_flagVARCHAR2(1),
    private_flagVARCHAR2(1),
    user_status_idNUMBER,
    system_status_codeVARCHAR2(30),
    last_status_dateDATE,
    stream_type_codeVARCHAR2(30),
    source_codeVARCHAR2(30),
    event_standalone_flagVARCHAR2(1),
    day_of_eventVARCHAR2(30),
    agenda_start_timeDATE,
    agenda_end_timeDATE,
    reg_required_flagVARCHAR2(1),
    reg_charge_flagVARCHAR2(1),
    reg_invited_only_flagVARCHAR2(1),
    partner_flagVARCHAR2(1),
    overflow_flagVARCHAR2(1),
    parent_event_header_idNUMBER,
    durationNUMBER,
    duration_uom_codeVARCHAR2(3),
    active_from_dateDATE,
    active_to_dateDATE,
    reg_maximum_capacityNUMBER,
    reg_minimum_capacityNUMBER,
    main_language_codeVARCHAR2(4),
    cert_credit_type_codeVARCHAR2(30),
    certification_creditsNUMBER,
    inventory_item_idNUMBER,
    organization_idNUMBER,
    org_idNUMBER,
    forecasted_revenueNUMBER,
    actual_revenueNUMBER,
```

```
forecasted_costNUMBER,
actual_costNUMBER,
coordinator_idNUMBER,
fund_source_type_codeVARCHAR2(30),
fund_source_idNUMBER,
fund_amount_tcNUMBER,
fund_amount_fcNUMBER,
currency_code_tcVARCHAR2(30),
currency_code_fcVARCHAR2(30),
owner_user_idNUMBER,
urlVARCHAR2(4000),
emailVARCHAR2(120),
phoneVARCHAR2(25),
priority_type_codeVARCHAR2(30),
cancellation_reason_codeVARCHAR2(30),
inbound_script_nameVARCHAR2(240),
attribute_categoryVARCHAR2(30),
attribute1VARCHAR2(150),
attribute2VARCHAR2(150),
attribute3VARCHAR2(150),
attribute4VARCHAR2(150),
attribute5VARCHAR2(150),
attribute6VARCHAR2(150),
attribute7VARCHAR2(150),
attribute8VARCHAR2(150),
attribute9VARCHAR2(150),
attribute10VARCHAR2(150),
attribute11VARCHAR2(150),
attribute12VARCHAR2(150),
attribute13VARCHAR2(150),
attribute14VARCHAR2(150),
attribute15VARCHAR2(150),
event_header_nameVARCHAR2(240),
event_mktg_messageVARCHAR2(4000),
descriptionVARCHAR2(4000),
custom_setup_idNUMBER,
country_codeVARCHAR2(30),
business_unit_idNUMBER,
event_calendarVARCHAR2(15),
start_period_nameVARCHAR2(15),
end_period_nameVARCHAR2(15),
global_flagVARCHAR2(1),
task_idNUMBER,
program_idNUMBER,
create_attendant_lead_flagVARCHAR2(1),
```

```
    create_registrant_lead_flagVARCHAR2(1),
    event_purpose_codeVARCHAR2(30)
);
```

10.2 Standard Parameters for Event APIs

There are a number of standard parameters which are common for all of the following APIs. Note that all the Standard OUT parameters are required. The parameters are listed in the tables below:

Table 10–2 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Default = FND_API.G_FALSE If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack. If set to false the calling program must initialize the message stack. This action is required to be performed only once, even in the case where more than one API is called.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.

Table 10–3 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

10.3 Event APIs

See the following sections for more information about Event APIs.

10.3.1 Create Event

This procedure creates an event with the supplied event ID, if its unique, or, if the ID is not supplied, a unique ID will be created.

Procedure Specification

```
PROCEDURE create_EventHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    p_commitINVARCHAR2 := FND_API.g_false,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evh_recINAMS_EventHeader_PVT.evh_rec_type,
    x_evh_idOUTNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Object_version_number will be set to 1.
2. If an EventHeader_id is passed in, the uniqueness will be checked. An exception will be raised in case of duplicates.
3. If an EventHeader_id is not passed in, a unique one will be generated from the sequence.
4. If a flag column is passed in, check if it is 'Y' or 'N'. Raise exception for invalid flag.
5. If a flag column is not passed in, default it to 'Y' or 'N'.
6. Please don't pass in any FND_API.g_mess_char/num/date.

Table 10–4 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_evh_rec	AMS_Event_Header_PVT.evh_rec_type	Yes	Record for the event. The record will be validated before creation of the event.

Table 10–5 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned are more than one, this parameter will be null and messages will have to be extracted from the message stack.
x_evh_id	NUMBER	New Event ID

10.3.2 Delete Event

When this API is called, the active flag of the event is changed from Yes to No. The event id and the object version number will be used to locate the event.

Procedure Specification

```
PROCEDURE delete_EventHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    p_commitINVARCHAR2 := FND_API.g_false,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evh_idINNUMBER,
    p_object_versionINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. Will set the event to be inactive, instead of removing it from the database.

Table 10–6 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_evh_id	NUMBER	Yes	Event ID.
p_object_ version_	NUMBER	Yes	Object version number of the event to be deleted. Based on the event ID and the object version number, the event record will be located and made inactive.

Table 10–7 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 10–7 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

10.3.3 Lock Event

This procedure locks the event record based on the event ID and the object version number passed. The API will raise an exception if the record matching the event id and the object version number does not exist.

Procedure Specification

```
PROCEDURE lock_EventHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evh_idINNUMBER,
    p_object_versionINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Only the Standard OUT parameters are required for this API.

Notes

1. If the object_version_number doesn't match, an exception will be raised.

Table 10–8 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.

Table 10–8 IN Parameters

Parameter	Data Type	Required	Description
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_evh_id	NUMBER	Yes	Event ID.
p_object_ version_	NUMBER	Yes	Object version number of the event to be locked. Based on the event ID and the object version number, the event record will be located and locked.

Table 10–9 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

10.3.4 Update Event

This procedure updates a event record based on the event ID and object version number. The record type for event can be initialized by g_miss rec and can be overridden by the values which are changed. For update the event ID and object version number are required fields in the record type. When the update is called, all the g_miss values are replaced with those of the database. When the record is updated, the object version number is incremented by 1.

Procedure Specification

```
PROCEDURE update_EventHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    p_commitINVARCHAR2 := FND_API.g_false,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evh_recINAMS_EventHeader_PVT.evh_rec_type
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. If an attribute is passed in as FND_API.g_miss_char/num/date, that column won't be updated.

Table 10–10 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. NONE means no validation will be done in the API and FULL means all the validations (item level and record level) will be performed.

Table 10–10 IN Parameters

Parameter	Data Type	Required	Description
p_evh_rec	AMS_ EventHeader _PVT.evh_ rec_type	Yes	Record for the Event. The record will be validated before updating the event.

Table 10–11 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

10.3.5 Validate Event

This procedure validates a event record. This API will be called internally by the Create Event API to validate the data and the business rules.

Procedure Specification

```
PROCEDURE validate_EventHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evh_recINAMS_EventHeader_PVT.evh_rec_type
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Oracle recommends that the p_evh_rec be the complete event record.

Table 10–12 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_evh_rec	AMS_ EventHeader _PVT.evh_ rec_type	Yes	Record for the event.

Table 10–13 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 10–13 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

10.4 Event API User Hooks

The following table lists Event HeaderUser Hook Procedures available in Oracle Marketing APIs.

Table 10–14 User Hook Procedures

Procedure	Parameter
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Create_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Create_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Delete_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Delete_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Lock_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Lock_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Update_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Update_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Validate_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Validate_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Create_EventHeader_Pre

AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Create_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Delete_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Delete_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Lock_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Lock_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Update_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Update_EventHeader_Post
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Validate_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_VUHK.Validate_EventHeader_Post

11

Event Schedule

The APIs for event schedule provide a number of procedures for event schedule actions.

The procedures which make up the Event Schedule APIs are:

Table 11–1 Event Schedule APIs

Procedure	Description
Create Event Schedule	Creates a new event schedule in which (a) the object version is set to one, (b) a unique event schedule ID will be created if a unique event schedule ID is not passed in, and (c) a flag column will be set to Y or N, depending on existence of optional parameters.
Delete Event Schedule	Rather that delete an event schedule, the event schedule record will have its active flag set to No when this API is called.
Lock Event Schedule	Locks the given event schedule record. Will raise an exception if the object version doesn't match the database record.
Update Event Schedule	Updates the event schedule record. The values which are not changed can be passed as g_miss record and will not be updated. Will raise an exception if the object version doesn't match the database record.
Validate Event Schedule	Validate different business rules like checking not null columns, valid flag values, and foreign key validation. In addition it also do other business validation. The p_evh_rec parameter should be the complete event schedule record.

11.1 Type Declaration

The Event record type will be used as input parameters in some of the APIs. The record type will be initialized to g_miss values before used for updating. The actual definition of the record type resides in the private API, hence the record type is referred to as AMS_EventOffer_PVT.evo_rec_type.

```
TYPE evo_rec_type IS RECORD(
    EVENT_OFFER_IDNUMBER,
    LAST_UPDATE_DATEDATE,
    LAST_UPDATED_BYNUMBER,
    CREATION_DATEDATE,
    CREATED_BYNUMBER,
    LAST_UPDATE_LOGINNUMBER,
    OBJECT_VERSION_NUMBERNUMBER,
    APPLICATION_IDNUMBER,
    EVENT_HEADER_IDNUMBER,
    PRIVATE_FLAGVARCHAR2(1),
    ACTIVE_FLAGVARCHAR2(1),
    SOURCE_CODEVARCHAR2(30),
    EVENT_LEVELVARCHAR2(30),
    USER_STATUS_IDNUMBER,
    LAST_STATUS_DATEDATE,
    SYSTEM_STATUS_CODEVARCHAR2(30),
    EVENT_TYPE_CODEVARCHAR2(30),
    EVENT_DELIVERY_METHOD_IDNUMBER,
    EVENT_DELIVERY_METHOD_CODEVARCHAR2(30),
    EVENT_REQUIRED_FLAGVARCHAR2(1),
    EVENT_LANGUAGE_CODEVARCHAR2(30),
    EVENT_LOCATION_IDNUMBER,
    CITYVARCHAR2(60),
    STATEVARCHAR2(60),
    PROVINCEVARCHAR2(60),
    COUNTRYVARCHAR2(60),
    OVERFLOW_FLAGVARCHAR2(1),
    PARTNER_FLAGVARCHAR2(1),
    EVENT_STANDALONE_FLAGVARCHAR2(1),
    REG_FROZEN_FLAGVARCHAR2(1),
    REG_REQUIRED_FLAGVARCHAR2(1),
    REG_CHARGE_FLAGVARCHAR2(1),
    REG_INVITED_ONLY_FLAGVARCHAR2(1),
    REG_WAITLIST_ALLOWED_FLAGVARCHAR2(1),
    REG_OVERBOOK_ALLOWED_FLAGVARCHAR2(1),
    PARENT_EVENT_OFFER_IDNUMBER,
    EVENT_DURATIONNUMBER,
```

```
EVENT_DURATION_UOM_CODEVARCHAR2(3),
EVENT_START_DATEDATE,
EVENT_START_DATE_TIMEDATE,
EVENT_END_DATEDATE,
EVENT_END_DATE_TIMEDATE,
REG_START_DATEDATE,
REG_START_TIMEDATE,
REG_END_DATEDATE,
REG_END_TIMEDATE,
REG_MAXIMUM_CAPACITYNUMBER,
REG_OVERBOOK_PCTNUMBER,
REG_EFFECTIVE_CAPACITYNUMBER,
REG_WAITLIST_PCTNUMBER,
REG_MINIMUM_CAPACITYNUMBER,
REG_MINIMUM_REQ_BY_DATEDATE,
INVENTORY_ITEM_IDNUMBER,
INVENTORY_ITEMVARCHAR2(1000),
ORGANIZATION_IDNUMBER,
PRICELIST_HEADER_IDNUMBER,
PRICELIST_LINE_IDNUMBER,
ORG_IDNUMBER,
WAITLIST_ACTION_TYPE_CODEVARCHAR2(30),
STREAM_TYPE_CODEVARCHAR2(30),
OWNER_USER_IDNUMBER,
EVENT_FULL_FLAGVARCHAR2(1),
FORECASTED_REVENUENUMBER,
ACTUAL_REVENUENUMBER,
FORECASTED_COSTNUMBER,
ACTUAL_COSTNUMBER,
FUND_SOURCE_TYPE_CODEVARCHAR2(30),
FUND_SOURCE_IDNUMBER,
CERT_CREDIT_TYPE_CODEVARCHAR2(30),
CERTIFICATION_CREDITSNUMBER,
COORDINATOR_IDNUMBER,
PRIORITY_TYPE_CODEVARCHAR2(30),
CANCELLATION_REASON_CODEVARCHAR2(30),
AUTO_REGISTER_FLAGVARCHAR2(1),
EMAILVARCHAR2(120),
PHONEVARCHAR2(25),
FUND_AMOUNT_TCNUMBER,
FUND_AMOUNT_FCNUMBER,
CURRENCY_CODE_TCVARCHAR2(15),
CURRENCY_CODE_FCVARCHAR2(15),
URLVARCHAR2(4000),
TIMEZONE_IDNUMBER,
```

```
EVENT_VENUE_IDNUMBER,
PRICELIST_HEADER_CURRENCY_CODE VARCHAR2(30),
PRICELIST_LIST_PRICENUMBER,
INBOUND_SCRIPT_NAMEVARCHAR2(240),
ATTRIBUTE_CATEGORYVARCHAR2(30),
ATTRIBUTE1VARCHAR2(150),
ATTRIBUTE2VARCHAR2(150),
ATTRIBUTE3VARCHAR2(150),
ATTRIBUTE4VARCHAR2(150),
ATTRIBUTE5VARCHAR2(150),
ATTRIBUTE6VARCHAR2(150),
ATTRIBUTE7VARCHAR2(150),
ATTRIBUTE8VARCHAR2(150),
ATTRIBUTE9VARCHAR2(150),
ATTRIBUTE10VARCHAR2(150),
ATTRIBUTE11VARCHAR2(150),
ATTRIBUTE12VARCHAR2(150),
ATTRIBUTE13VARCHAR2(150),
ATTRIBUTE14VARCHAR2(150),
ATTRIBUTE15VARCHAR2(150),
EVENT_OFFER_NAMEVARCHAR2(240),
EVENT_MKTG_MESSAGEVARCHAR2(4000),
DESCRIPTIONVARCHAR2(4000),
CUSTOM_SETUP_IDNUMBER,
COUNTRY_CODEVARCHAR2(30),
BUSINESS_UNIT_IDNUMBER,
EVENT_CALENDARVARCHAR2(15),
START_PERIOD_NAMEVARCHAR2(15),
END_PERIOD_NAMEVARCHAR2(15),
GLOBAL_FLAGVARCHAR2(1),
TASK_IDNUMBER,
--PROGRAM_IDNUMBER,
PARENT_TYPEVARCHAR2(30),
PARENT_IDNUMBER,
CREATE_ATTENDANT_LEAD_FLAGVARCHAR2(1),
CREATE_REGISTRANT_LEAD_FLAGVARCHAR2(1),
EVENT_OBJECT_TYPEVARCHAR2(30),
REG_TIMEZONE_IDNUMBER,
EVENT_PASSWORDVARCHAR2(30),
RECORD_EVENT_FLAGVARCHAR2(1),
ALLOW_REGISTER_IN_MIDDLE_FLAG  VARCHAR2(1),
PUBLISH_ATTENDEES_FLAGVARCHAR2(1),
DIRECT_JOIN_FLAGVARCHAR2(1),
EVENT_NOTIFICATION_METHODVARCHAR2(30),
ACTUAL_START_TIMEDATE,
```

```
    ACTUAL_END_TIMEDATE,
    SERVER_IDNUMBER,
    OWNER_FND_USER_IDNUMBER,
    MEETING_DIAL_IN_INFOVARCHAR2(4000),
    MEETING_EMAIL_SUBJECTVARCHAR2(4000),
    MEETING_SCHEDULE_TYPEVARCHAR2(30),
    MEETING_STATUSVARCHAR2(30),
    MEETING_MISC_INFOVARCHAR2(4000),
    PUBLISH_FLAGVARCHAR2(1),
    MEETING_ENCRYPTION_KEY_CODEVARCHAR2(150),
    NUMBER_OF_ATTENDEESNUMBER,
    EVENT_PURPOSE_CODEVARCHAR2(30)
);
```

11.2 Standard Parameters for Event Schedule APIs

There are a number of standard parameters which are common for all of the following APIs. Note that all the Standard OUT parameters are required. The parameters are listed in the tables below:

Table 11–2 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Default = FND_API.G_FALSE If set to true, then the API makes a call to fnd_msg_pub.initialize to initialize the message stack. If set to false the calling program must initialize the message stack. This action is required to be performed only once, even in the case where more than one API is called.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.

Table 11–2 Standard IN Parameters

Parameter	Data Type	Required	Description
p_validation_ level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.

Table 11–3 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

11.3 Event Schedule APIs

11.3.1 Create Event Schedule

This procedure creates an event schedule with the supplied event schedule ID, if it is unique, or, if the ID is not supplied, a unique ID will be created.

Procedure Specification

```
PROCEDURE create_EventOffer(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    p_commitINVARCHAR2 := FND_API.g_false,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
```

```
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evo_recINAMS_EventOffer_PVT.evo_rec_type,
    x_evo_idOUTNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Object_version_number will be set to 1.
2. If an event schedule id is passed in, the uniqueness will be checked. Raise exception in case of duplicates.
3. If an event schedule id is not passed in, generate a unique one from the sequence.
4. If a flag column is passed in, check if it is 'Y' or 'N'. Raise exception for invalid flag.
5. If a flag column is not passed in, default it to 'Y' or 'N'.
6. Please don't pass in any FND_API.g_mess_char/num/date.

Table 11–4 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_validation_ level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.

Table 11–4 IN Parameters

Parameter	Data Type	Required	Description
p_evo_rec	AMS_Event_ Header_ PVT.evo_rec_ type	Yes	Record for the event schedule. The record will be validated before creation of the event schedule.

Table 11–5 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages returned by the API. If the number of error messages returned is one, the message count will be zero.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned are more than one, this parameter will be null and messages will have to be extracted from the message stack.
x_evo_id	NUMBER	New Event Schedule ID

11.3.2 Delete Event Schedule

When this API is called, the active flag of the event is changed from Yes to No. The event schedule ID and the object version number will be used to locate the event schedule.

Procedure Specification

```
PROCEDURE delete_EventOffer(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
```

```
    p_commitINVARCHAR2 := FND_API.g_false,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evo_idINNUMBER,
    p_object_versionINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. Will set the Event Schedule to be inactive, instead of removing it from the database.

Table 11–6 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_evo_id	NUMBER	Yes	Event ID.
p_object_ version_	NUMBER	Yes	Object version number of the event schedule to be deleted. Based on the event schedule ID and the object version number, the event schedule record will be located and disabled.

Table 11–7 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

11.3.3 Lock Event Schedule

This procedure locks the campaign record based on the event schedule ID and the object version number passed. The API will raise an exception if the record matching the event schedule ID and the object version number does not exist.

Procedure Specification

```
PROCEDURE lock_EventOffer(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evo_idINNUMBER,
    p_object_versionINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Raise exception if the object_version_number doesn't match.

Table 11–8 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_evo_id	NUMBER	Yes	Event ID.
p_object_ version_	NUMBER	Yes	Object version number of the event schedule to be locked. Based on the event schedule ID and the object version number, the event schedule record will be located and locked.

Table 11–9 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

11.3.4 Update Event Schedule

This procedure updates a event schedule record based on the event schedule ID and object version number. The record type for event schedule can be initialized by g_ miss rec and can be overridden by the values which are changed. For update, the event schedule ID and object version number are required fields in the record type. When the update is called, all the g_miss values are replaced with those of the database. When the record is updated, the object version number is incremented by 1.

Procedure Specification

```
PROCEDURE update_EventOffer(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    p_commitINVARCHAR2 := FND_API.g_false,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evo_recINAMS_EventHeader_PVT.evo_rec_type
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception is raised.
2. If an attribute is passed in as FND_API.g_miss_char/num/date, that column won't be updated.

Table 11–10 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.

Table 11–10 IN Parameters

Parameter	Data Type	Required	Description
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_evo_rec	AMS_eventoffer_PVT.evo_rec_type	Yes	Record for the event schedule. The record will be validated before updating of the event schedule.

Table 11–11 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

11.3.5 Validate Event Schedule

This procedure validates a event schedule record. This API will be called internally by the Create Event Schedule API to validate the data and the business rules.

Procedure Specification

```
PROCEDURE validate_EventOffer(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.g_false,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_evo_recINAMS_EventHeader_PVT.evo_rec_type
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Oracle recommends that p_camp_rec be the complete event schedule record.

Table 11–12 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_evo_rec	AMS_ eventoffer_ PVT.evo_rec_ type	Yes	Record for the event schedule.

Table 11–13 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

11.4 Event Offer API User Hooks

The following table lists Event Offer User Hook Procedures available in Oracle Marketing APIs.

Table 11–14 Event Offer API Hooks

AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Create_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Create_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Delete_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Delete_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Lock_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Lock_EventOffer_Post

AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Update_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Update_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Validate_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_CUHK.Validate_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Create_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Create_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Delete_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Delete_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Lock_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Lock_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Update_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Update_EventOffer_Post
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Validate_EventOffer_Pre
AMS_EVENTOFFER_PUB	AMS_EventOffer_VUHK.Validate_EventOffer_Post

12

List

The APIs for List provides a number of procedures for list management.

The procedures which make up the List APIs are:

Table 12–1 List APIs

Procedure	Description
Create List	Creates a new list in which (a) the object version is set to one, (b) a unique list ID will be created if a unique list header ID is not passed in. In marketing applications the list Name is unique, and in sales and partner applications the combination of the list name and the owner is unique. When the list is created, the initial status is Draft.
Delete List	Deletes the List from the database. This procedure also deletes child records from ams_list_select_actions, ams_list_entries, ams_list_src_type_usages and ams_list_rule_usages tables.
Lock List	Locks the given list record. Will raise an exception if the object version doesn't match the database record.
Update List	Updates the list record. The values which are not changed can be passed as g_miss record and will not be updated. Will raise an exception if the object version doesn't match the database record.
Validate List	Validate different business rules like checking not null columns, valid flag values, and foreign key validation. In addition it also do other business validation. The p_list_ header_rec parameter should be the complete list record.

Table 12–1 List APIs

Procedure	Description
Copy List Header	Takes the list header ID of the list to copy from, list name, public flag, purge flag, owner user ID and description for the new list and generates a new list ID. Copies the records pertaining to a particular list in ams_list_select_actions, ams_list_queries_all, and ams_list_entries into a new set and associate them with a new list.

12.1 Type Declaration

This section defines the list record type declaration. The list record type is used as an IN parameter in some of the procedures for creation or updating. The actual definition of the record type resides in the private API, hence the record type is referred to as AMS_LISTHEADER_PVT.list_header_rec_type.

```
TYPE list_header_rec_type IS RECORD(
    list_header_idnumber,
    last_update_datedate,
    last_updated_bynumber,
    creation_datedate,
    created_bynumber,
    last_update_loginnumber,
    object_version_numbernumber,
    request_idnumber,
    program_idnumber,
    program_application_idnumber,
    program_update_datedate,
    view_application_idnumber,
    list_namevarchar2(240),
    list_used_by_idnumber,
    arc_list_used_byvarchar2(30),
    list_typevarchar2(30),
    status_codevarchar2(30),
    status_datedate,
    generation_typevarchar2(30),
    repeat_exclude_typevarchar2(30),
    row_selection_typevarchar2(30),
    owner_user_idnumber,
    access_levelvarchar2(30),
    enable_log_flagvarchar2(1),
    enable_word_replacement_flagvarchar2(1),
    enable_parallel_dml_flagvarchar2(1),
    dedupe_during_generation_flagvarchar2(1),
```

```
generate_control_group_flagvarchar2(1),
last_generation_success_flagvarchar2(1),
forecasted_start_datedate,
forecasted_end_datedate,
actual_end_datedate,
sent_out_datedate,
dedupe_start_datedate,
last_dedupe_datedate,
last_deduped_by_user_idnumber,
workflow_item_keynumber,
no_of_rows_duplicatesnumber,
no_of_rows_min_requestednumber,
no_of_rows_max_requestednumber,
no_of_rows_in_listnumber,
no_of_rows_in_ctrl_groupnumber,
no_of_rows_activenumber,
no_of_rows_inactivenumber,
no_of_rows_manually_enterednumber,
no_of_rows_do_not_callnumber,
no_of_rows_do_not_mailnumber,
no_of_rows_randomnumber,
org_idnumber,
main_gen_start_timedate,
main_gen_end_timedate,
main_random_nth_row_selectionnumber,
main_random_pct_row_selectionnumber,
ctrl_random_nth_row_selectionnumber,
ctrl_random_pct_row_selectionnumber,
repeat_source_list_header_idvarchar2(4000),
result_textvarchar2(4000),
keywordsvarchar2(4000),
descriptionvarchar2(4000),
list_prioritynumber,
assign_person_idnumber,
list_sourcevarchar2(240),
list_source_typevarchar2(30),
list_online_flagvarchar2(1),
random_list_idnumber,
enabled_flagvarchar2(1),
assigned_tonumber,
query_idnumber,
owner_person_idnumber,
archived_bynumber,
archived_datedate,
attribute_categoryvarchar2(30),
```

```
    attribute1varchar2(150),
    attribute2varchar2(150),
    attribute3varchar2(150),
    attribute4varchar2(150),
    attribute5varchar2(150),
    attribute6varchar2(150),
    attribute7varchar2(150),
    attribute8varchar2(150),
    attribute9varchar2(150),
    attribute10varchar2(150),
    attribute11varchar2(150),
    attribute12varchar2(150),
    attribute13varchar2(150),
    attribute14varchar2(150),
    attribute15varchar2(150),
    timezone_idnumber,
    user_entered_start_timedate,
    user_status_idnumber,
    quantumnumber,
    release_control_alg_idnumber,
    dialing_methodvarchar2(10),
    calling_calendar_idnumber,
    release_strategyvarchar2(10),
    custom_setup_idnumber,
    countrynumber,
    callback_priority_flagvarchar2(1),
    call_center_ready_flagvarchar2(1),
    languagevarchar2(4) ,
    purge_flagvarchar2(1),
    public_flagvarchar2(1),
    list_categoryvarchar2(120),
    quotanumber,
    quota_resetnumber,
    recycling_alg_idnumber,
    source_langvarchar2(4)
);
```

12.2 Standard Parameters for List APIs

There are a number of standard parameters which are common for all of the following APIs. Note that all the Standard OUT parameters are required. The parameters are listed in the tables below:

Table 12–2 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.

Table 12–3 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

12.3 List APIs

The following sections contain information on List APIs.

12.3.1 Create List

This procedure creates a list with the supplied list ID, if it is unique, or if the ID is not supplied, a unique ID will be created.

Procedure Specification

```
PROCEDURE Create_Listheader (
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_listheader_recINAMS_LISTHEADER_PVT.list_header_rec_type,
    x_listheader_idOUTNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Object_version_number will be set to 1.
2. If list_header_id is passed in, the uniqueness will be checked. An exception will be made in case of duplicates.
3. If list_entry_id is not passed in, generate a unique one from the sequence.
4. If a flag column is passed in, check if it is "Y" or "N". Raise exception for an invalid flag.
5. If a flag column is not passed in, fields will be defaulted to "Y" or "N" as appropriate.
6. Please don't pass in any FND_API.g_mess_char/num/date.
7. The list header status will be set to Draft by default.

Table 12–4 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.

Table 12–4 IN Parameters

Parameter	Data Type	Required	Description
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_listheader_rec	AMS_Listheader_PVT.list_header_rec_type	Yes	Record for the list. The record will be validated before creation of the list.

Table 12–5 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

12.3.2 Delete List Header

This procedure deletes a list by the list records and the corresponding details.

Procedure Specification

```
PROCEDURE Delete_ListHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_listheader_idINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. Will delete the record from the list header table and also delete the child records from the ams_list_select_actions, ams_list_entries, ams_list_src_type_usages, and ams_list_rule_usages tables.

Table 12–6 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_listheader_id	NUMBER	Yes	Identifier for the list.

Table 12–6 IN Parameters

Parameter	Data Type	Required	Description
p_object_ version_ number	NUMBER	Yes	Object version number of the list to be deleted. Based on the list ID and object version number, the list record will be located and deleted.

Table 12–7 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

12.3.3 Lock List

This procedure locks the list record based on the list ID and the object version number. The API will raise an exception if the record matching the list ID and object version number do not exist.

Procedure Specification

```
PROCEDURE Lock_ListHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
```

```
    p_listheader_idINNUMBER,
    p_object_versionINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Table 12–8 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_listheader_id		Yes	List ID.
p_object_ version_ number	NUMBER	Yes	Object version number of the list header to be locked.

Table 12–9 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 12–9 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

12.3.4 Update List

This procedure updates a list record based on the list ID and object version number. The record type for list can be initialized by g_miss_rec and can be overridden for those fields whose values are changed. For update, the list ID and object version number are required fields in the record type. When the update is called, all the g_ miss values are replaced with those of the database. When the record is updated, the object number version is incremented by 1.

Procedure Specification

```
PROCEDURE Lock_ListHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_listheader_idINNUMBER,
    p_object_versionINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. If an attribute is passed in as FND_API.g_miss_char/num/date, that column won't be updated.

Table 12–10 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_listheader_rec	AMS_Listheader_PVT.list_header_rec_type	Yes	Record for the list. The record will be validated before updating of the list.

Table 12–11 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

12.3.5 Validate List

This procedure validates a list record. This API is called internally by the Create List API to validate the data and the business rules.

Procedure Specification

```
PROCEDURE Validate_ListHeader(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2:= FND_API.G_FALSE,
    p_validation_levelINNUMBER:= FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_listheader_recINAMS_LISTHEADER_PVT.list_header_rec_type
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Oracle recommends that the p-listheader_rec be the complete list record.

Table 12–12 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. NONE means no validation will be done in the API and FULL means all the validations (item level and record level) will be performed.
p_listheader_ rec	AMS_ Listheader_ PVT.list_ header_rec_ type	Yes	Record for the list header. The record will be validated before creation of the list.

Table 12–13 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

12.3.6 Copy List

This API copies a list record and the corresponding detail records.

Procedure Specification

```
PROCEDURE Copy_List(
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.g_valid_level_full,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_source_listheader_idINNUMBER,
    p_listheader_recINAMS_LISTHEADER_PVT.list_header_rec_type,
    p_copy_select_actionsINVARCHAR2 := 'Y',
    p_copy_list_queriesINVARCHAR2 := 'Y',
    p_copy_list_entriesINVARCHAR2 := 'Y',
    x_listheader_idOUTNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Oracle recommends that the p-listheader_rec should be the complete list record.

Table 12–14 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_source_list_ header_id	VARCHAR2	Yes	ID for the list. The record will be validated before creation of the list.
p_listheader_ rec	AMS_ Listheader_ PVT.list_ header_rec_ type	Yes	Record for the list header. The record will be validated before creation of the list header.
p_copy_select_ actions	VARCHAR2	No	Should the select actions be copied.
p_copy_list_ queries	VARCHAR2	No	Should the list queries be copied.
p_copy_list_ entries	VARCHAR2	No	Should the entries be copied.

Table 12–15 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.
x_listheader_id	NUMBER	List ID of the new list that was created.

13

List Entries

The APIs for List Entries provide a number of procedures for List Entry actions.

The procedures which make up the List Entries APIs are:

Table 13–1 List Entries APIs

Procedure	Description
Create List Entries	Creates new list entries in which (a) the object version is set to one, (b) a unique list entries ID will be created if a unique entries ID is not passed in, and (c) a valid list ID needs to be passed to the procedure.
Delete List Entries	Deletes the list entry from the database. This procedure takes a list_entry_id as the input and deletes the entry.
Lock List Entries	Locks the given list record. Will raise an exception if the object version doesn't match the database record.
Update List Entries	Updates the list entry record. The values which are not changed can be passed as g_miss record and will not be updated. Will raise an exception if the object version doesn't match the database record.
Validate List Entries	Validate different business rules like checking not null columns, valid flag values, and foreign key validation. In addition, it also does other business validation. The p_list_ entries_rec parameter should be the complete list record.
Copy List Entries	Takes the list ID of the list to copy from and copies the list entries to the destination list ID.

13.1 Type Declaration

This section defines the List Entries record type declaration. The List Entries record type is used as an IN parameter in some of the procedures for creation or updating. The actual definition of the record type resides in the private API, hence the record type is referred to as AMS_LISTENTRIES_PVT.list_entries_rec_type.

```
TYPE list_Entries_rec_type IS RECORD(
    list_entry_idNUMBER := FND_API.G_MISS_NUM,
    list_header_idNUMBER := FND_API.G_MISS_NUM,
    last_update_dateDATE := FND_API.G_MISS_DATE,
    last_updated_byNUMBER := FND_API.G_MISS_NUM,
    creation_dateDATE := FND_API.G_MISS_DATE,
    created_byNUMBER := FND_API.G_MISS_NUM,
    last_update_loginNUMBER := FND_API.G_MISS_NUM,
    object_version_numberNUMBER := FND_API.G_MISS_NUM,
    list_select_action_idNUMBER := FND_API.G_MISS_NUM,
    arc_list_select_action_fromVARCHAR2(30) := FND_API.G_MISS_CHAR,
    list_select_action_from_nameVARCHAR2(254) := FND_API.G_MISS_CHAR,
    source_codeVARCHAR2(30) := FND_API.G_MISS_CHAR,
    arc_list_used_by_sourceVARCHAR2(30) := FND_API.G_MISS_CHAR,
    source_code_for_idNUMBER := FND_API.G_MISS_NUM,
    pin_codeVARCHAR2(30) := FND_API.G_MISS_CHAR,
    list_entry_source_system_idNUMBER := FND_API.G_MISS_NUM,
    list_entry_source_system_typeVARCHAR2(30) := FND_API.G_MISS_CHAR,
    view_application_idNUMBER := FND_API.G_MISS_NUM,
    manually_entered_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    marked_as_duplicate_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    marked_as_random_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    part_of_control_group_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    exclude_in_triggered_list_flag  VARCHAR2(1) := FND_API.G_MISS_CHAR,
    enabled_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    cell_codeVARCHAR2(30) := FND_API.G_MISS_CHAR,
    dedupe_keyVARCHAR2(500) := FND_API.G_MISS_CHAR,
    randomly_generated_numberNUMBER := FND_API.G_MISS_NUM,
    campaign_idNUMBER := FND_API.G_MISS_NUM,
    media_idNUMBER := FND_API.G_MISS_NUM,
    channel_idNUMBER := FND_API.G_MISS_NUM,
    channel_schedule_idNUMBER := FND_API.G_MISS_NUM,
    event_offer_idNUMBER := FND_API.G_MISS_NUM,
    customer_idNUMBER := FND_API.G_MISS_NUM,
    market_segment_idNUMBER := FND_API.G_MISS_NUM,
    vendor_idNUMBER := FND_API.G_MISS_NUM,
    transfer_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    transfer_statusVARCHAR2(1) := FND_API.G_MISS_CHAR,
```

```
list_sourceVARCHAR2(240) := FND_API.G_MISS_CHAR,
duplicate_master_entry_idNUMBER := FND_API.G_MISS_NUM,
marked_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
lead_idNUMBER := FND_API.G_MISS_NUM,
letter_idNUMBER := FND_API.G_MISS_NUM,
picking_header_idNUMBER := FND_API.G_MISS_NUM,
batch_idNUMBER := FND_API.G_MISS_NUM,
suffixVARCHAR2(30) := FND_API.G_MISS_CHAR,
first_nameVARCHAR2(150) := FND_API.G_MISS_CHAR,
last_nameVARCHAR2(150) := FND_API.G_MISS_CHAR,
customer_nameVARCHAR2(500) := FND_API.G_MISS_CHAR,
titleVARCHAR2(150) := FND_API.G_MISS_CHAR,
address_line1VARCHAR2(500) := FND_API.G_MISS_CHAR,
address_line2VARCHAR2(500) := FND_API.G_MISS_CHAR,
cityVARCHAR2(100) := FND_API.G_MISS_CHAR,
stateVARCHAR2(100) := FND_API.G_MISS_CHAR,
zipcodeVARCHAR2(100) := FND_API.G_MISS_CHAR,
countryVARCHAR2(100) := FND_API.G_MISS_CHAR,
faxVARCHAR2(150) := FND_API.G_MISS_CHAR,
phoneVARCHAR2(150) := FND_API.G_MISS_CHAR,
email_addressVARCHAR2(500) := FND_API.G_MISS_CHAR,
COL1VARCHAR2(500) := FND_API.G_MISS_CHAR,
COL240VARCHAR2(500) := FND_API.G_MISS_CHAR,
COL241VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL242VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL243VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL244VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL245VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL246VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL247VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL248VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL249VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL250VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL251VARCHAR2(500) := FND_API.G_MISS_CHAR,
COL300VARCHAR2(500) := FND_API.G_MISS_CHAR,
CURR_CP_COUNTRY_CODEVARCHAR2(30) := FND_API.G_MISS_CHAR,
CURR_CP_PHONE_NUMBERVARCHAR2(10) := FND_API.G_MISS_CHAR,
CURR_CP_RAW_PHONE_NUMBERVARCHAR2(60) := FND_API.G_MISS_CHAR,
CURR_CP_AREA_CODENUMBER := FND_API.G_MISS_NUM,
CURR_CP_IDNUMBER := FND_API.G_MISS_NUM,
CURR_CP_INDEXNUMBER := FND_API.G_MISS_NUM,
CURR_CP_TIME_ZONENUMBER := FND_API.G_MISS_NUM,
CURR_CP_TIME_ZONE_AUXNUMBER := FND_API.G_MISS_NUM,
party_idNUMBER := FND_API.G_MISS_NUM,
parent_party_idNUMBER := FND_API.G_MISS_NUM,
```

```
    imp_source_line_idNUMBER := FND_API.G_MISS_NUM,
    usage_restrictionVARCHAR2(1) := FND_API.G_MISS_CHAR,
    next_call_timeDATE := FND_API.G_MISS_DATE,
    callback_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    do_not_use_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    do_not_use_reasonVARCHAR2(30) := FND_API.G_MISS_CHAR,
    record_out_flagVARCHAR2(1) := FND_API.G_MISS_CHAR,
    record_release_timeDATE := FND_API.G_MISS_DATE
);
```

13.2 Standard Parameters for List Entries APIs

There are a number of standard parameters which are common for all of the following APIs. Note that all the Standard OUT parameters are required. The parameters are listed in the tables below:

Table 13–2 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.

Table 13–3 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 13–3 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

13.3 List Entries APIs

13.3.1 Create List Entries

This procedure creates list entries with the supplied list entries ID, if it is unique, or if the ID is not supplied, a unique ID will be created.

Procedure Specification

```
PROCEDURE Create_List_Entries(
    p_api_version_numberINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.G_VALID_LEVEL_FULL,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_list_entries_recINlist_entries_rec_type := g_miss_list_entries_rec,
    x_list_entry_idOUTNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. Object_version_number will be set to 1.
2. If list_entry_id is passed in, the uniqueness will be checked. An exception will be raised in case of duplicates.
3. If list_entry_id is not passed in, a unique one will be generated from the sequence.

4. Please don't pass in any FND_API.g_mess_char/num/date.

Table 13–4 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_list_entries_rec	AMS_List_Entries_PVT.List_entries_rec_type	Yes	Record for the List Entries. The record will be validated before creation of the List Entries.

Table 13–5 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 13–5 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

13.3.2 Delete List Entries

This procedure deletes list entries from ams_list_entries based on the list_entry_id.

Procedure Specification

```
PROCEDURE Delete_List_Entries(
    p_api_version_numberINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.G_VALID_LEVEL_FULL,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_list_entry_idINNUMBER,
    p_object_version_numberINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. Will delete the record from the list entries table.

Table 13–6 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_list_entry_id	NUMBER	Yes	Identifier for the list entry.
p_object_version_number	NUMBER	Yes	Object version number of the list entries to be deleted. Based on the list entries ID and object version number the list entries record will be located and deleted.

Table 13–7 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 13–7 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

13.3.3 Lock List Entries

This procedure locks the list entries record based on the list entry ID and the object version number. The API will raise an exception if the record matching the list entry ID and object version number do not exist.

Procedure Specification

```
PROCEDURE Lock_List_Entries(
    p_api_version_numberINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_list_entry_idINNUMBER,
    p_object_versionINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Table 13–8 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	Caller version number, will be compared against the API version number to detect incompatibility.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_list_entry_rec		Yes	Record ID for the list entry.

Table 13–8 IN Parameters

Parameter	Data Type	Required	Description
p_object_ version_ number	NUMBER	Yes	Object version number of the list entries to be deleted. Based on the list entries id and object version number the list entries record will be located and deleted.

Table 13–9 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

13.3.4 Update List Entries

This procedure updates a list entries record based on the list entry ID and object version number. The record type for list entries can be initialized by g_miss_rec and can be overridden for those fields whose values are changed. For update, the list entries ID and object version number are required fields in the record type. When the update is called, all the g_miss values are replaced with those of the database. When the record is updated, the object number version is incremented by 1.

Procedure Specification

```
PROCEDURE Update_List_Entries(
    p_api_version_numberINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
```

```
    p_validation_levelINNUMBER := FND_API.G_VALID_LEVEL_FULL,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_list_entries_recINlist_entries_rec_type,
    x_object_version_numberOUTNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. If the object_version_number doesn't match, an exception will be raised.
2. If an attribute is passed in as FND_API.g_miss_char/num/date, that column won't be updated.

Table 13–10 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_list_entry_rec	AMS_List_ Entries_ PVT.list_ entries_rec_ type	Yes	Record ID for the list entry.

Table 13–11 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

13.3.5 Validate List Entries

This procedure validates a list entries record. This API is called internally by the Create List Entries API to validate the data and the business rules.

Procedure Specification

```
PROCEDURE Validate_list_entries(
    p_api_version_numberINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.G_VALID_LEVEL_FULL,
    p_list_entries_recINlist_entries_rec_type,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Table 13–12 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_list_entries_ rec	AMS_List_ Entries_PVT. list_entries_ rec_type	Yes	Record for the list entries. The record will be validated before creation of the list entries.

Table 13–13 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

13.3.6 Copy List Entries

This API copies a list entries record and the corresponding detail records.

Procedure Specification

```
PROCEDURE Copy_List_Entries(
    p_api_version_numberINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_FALSE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.G_VALID_LEVEL_FULL,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2,
    p_list_header_idINNUMBER,
    p_new_list_header_idINNUMBER
);
```

Current Version

1.0

Parameter Descriptions

Table 13–14 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_validation_ level	NUMBER	No	Level of Validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_list_header_ id	NUMBER	Yes	List header ID of the list from which the list entries will be copied.
p_new_list_ header_id	NUMBER	Yes	List header ID of the list to which the entries will be copies.

Table 13–15 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

14

List Generation

The APIs for List Generation provides two procedures for generating lists. The procedures which make up the List Generation APIs are:

Table 14–1 List Generation APIs

Procedure	Description
Generate List	Generates a list based on the list header ID passed to the procedure.
Create List based on Query	Creates and generates a list based on the list name and query string provided to the procedure.

14.1 Standard Parameters for List Generation APIs

There are a number of standard parameters which are common for both of the following APIs. Note that all the Standard OUT parameters are required. The parameters are listed in the tables below:

Table 14–2 Standard IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.

Table 14–3 Standard OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

14.2 List Generation APIs

See the following sections for more information on List Generation APIs.

14.2.1 Generate List

This procedure generates list entries give the list header ID.

Procedure Specification

```
PROCEDURE Generate_List (
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_TRUE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER:= FND_API.G_VALID_LEVEL_FULL,
    p_list_header_idINNUMBER,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Notes

1. The p_list_header_id should be a valid list existing in the ams_list_headers table.

Table 14–4 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
p_validation_level	NUMBER	No	Level of validation required. None means no validation will be done in the API and Full means all the validations (item level and record level) will be performed.
p_list_header_id	NUMBER	Yes	List ID of the list to be generated.

Table 14–5 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 14–5 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.

14.2.2 Create List Based on Query

This procedure creates lists selections, list queries and then generates the list for a given list header ID.

Procedure Specification

```
PROCEDURE create_list_from_query (
    p_api_versionINNUMBER,
    p_init_msg_listINVARCHAR2 := FND_API.G_TRUE,
    p_commitINVARCHAR2 := FND_API.G_FALSE,
    p_validation_levelINNUMBER := FND_API.G_VALID_LEVEL_FULL,
    p_list_nameinvarchar2,
    p_list_typeinvarchar2,
    p_owner_user_idinnumber,
    p_list_header_idinnumber,
    p_sql_string_tblinAMS_List_Query_PVT.sql_string_tbl,
    p_primary_keyinvarchar2,
    p_source_object_nameinvarchar2,
    p_master_typeinvarchar2,
    x_return_statusOUTVARCHAR2,
    x_msg_countOUTNUMBER,
    x_msg_dataOUTVARCHAR2
);
```

Current Version

1.0

Parameter Descriptions

Table 14–6 IN Parameters

Parameter	Data Type	Required	Description
p_api_version	NUMBER	Yes	This must match the version number of the API. An unexpected error is returned if the calling program version number is incompatible with the current API version number.
p_init_msg_list	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: FND_API.g_false.
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_ API.g_false.
p_list_name	VARCHAR2	Yes	Name of the list.
p_list_type	VARCHAR2	Yes	There are two valid List Types: "STANDARD" and "MANUAL".
p_owner_user_ id	NUMBER	Yes	Resource ID for the user.
p_list_header_ id	NUMBER	Yes	Header ID of the list.

Table 14–7 OUT Parameters

Parameter	Data Type	Description
x_return_status	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: FND_API.G_RET_STS_SUCCESS which indicates the API call was successful. FND_API.G_RET_STS_ERROR which indicates there was a validation error or a missing data error. FND_API.G_RET_STS_UNEXP_ERROR which indicates the calling program encountered an unexpected or unhandled error.
x_msg_count	NUMBER	Holds the number of messages in the message list.

Table 14–7 OUT Parameters

Parameter	Data Type	Description
x_msg_data	VARCHAR2(2000)	Error message returned by the API. If the messages returned number more than one, this parameter will be null and the messages have to be extracted from the message stack.