

# Oracle<sup>®</sup> Marketing Online

API Reference Guide

Release 11*i*

September 2002

Part No. A97384-02

**ORACLE<sup>®</sup>**

Part No. A97384-02

Copyright © 1996, 2002, Oracle Corporation. All rights reserved.

Primary Author: James A. Hahn

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

**Restricted Rights Notice** Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation. Other names may be trademarks of their respective owners.

---

---

# Contents

<b>Send Us Your Comments .....</b>	<b>vii</b>
<b>Preface.....</b>	<b>ix</b>
Introduction.....	ix
Intended Audience .....	ix
How This Guide is Organized .....	x
How to Use This Guide.....	x
Typographic Conventions.....	x
Documentation Accessibility .....	xi
Other Information Sources .....	xi
Training and Support.....	xiv
Do Not Use Database Tools to Modify Oracle Applications Data .....	xv
About Oracle .....	xv
<b>1 Introduction</b>	
1.1 Parameter Specifications .....	1-1
1.1.1 Standard In Parameters.....	1-2
1.1.2 Standard OUT parameters.....	1-2
1.1.3 Parameter Size .....	1-3
1.1.4 Missing Parameter Attributes .....	1-3
1.1.5 Parameter Validations .....	1-4
1.1.6 Invalid Parameters .....	1-4
1.2 Version Information.....	1-4
1.3 Status Messages .....	1-5

## 2 User Hooks

2.1	Introduction.....	2-1
2.2	User Hook Registration .....	2-2
2.3	Available Campaign User Hooks.....	2-2
2.4	Available Event User Hooks.....	2-4
2.5	Parameters for User Hook Procedures.....	2-6

## 3 Campaign

3.1	User Hook Procedures.....	3-1
3.2	Type Declaration.....	3-2
3.3	Standard Parameters for Campaign APIs.....	3-5
3.4	Campaign APIs.....	3-6
3.4.1	Create Campaign.....	3-6
3.4.2	Delete Campaign.....	3-8
3.4.3	Lock Campaign.....	3-10
3.4.4	Update Campaign.....	3-11
3.4.5	Validate Campaign.....	3-13

## 4 Campaign Schedule

4.1	User Hook Procedures.....	4-2
4.2	Type Declaration.....	4-3
4.3	Standard Parameters for Campaign Schedule APIs.....	4-5
4.4	Campaign Schedule APIs.....	4-6
4.4.1	Create Campaign Schedule.....	4-6
4.4.2	Delete Campaign Schedule.....	4-8
4.4.3	Lock Campaign Schedule.....	4-10
4.4.4	Update Campaign Schedule.....	4-11
4.4.5	Validate Campaign Schedule.....	4-13
4.4.6	Copy Campaign Schedule.....	4-14

## 5 Event

5.1	Type Declaration.....	5-1
5.2	Standard Parameters for Event APIs.....	5-4
5.3	Event APIs.....	5-5

5.3.1	Create Event.....	5-5
5.3.2	Delete Event.....	5-7
5.3.3	Lock Event.....	5-9
5.3.4	Update Event.....	5-10
5.3.5	Validate Event.....	5-12

## 6 Event Schedule

6.1	Type Declaration.....	6-2
6.2	Standard Parameters for Event Schedule APIs.....	6-5
6.3	Event Schedule APIs.....	6-6
6.3.1	Create Event Schedule.....	6-6
6.3.2	Delete Event Schedule.....	6-8
6.3.3	Lock Event Schedule.....	6-10
6.3.4	Update Event Schedule.....	6-12
6.3.5	Validate Event Schedule.....	6-13

## 7 List

7.1	Type Declaration.....	7-2
7.2	Standard Parameters for List APIs.....	7-4
7.3	List APIs.....	7-5
7.3.1	Create List.....	7-5
7.3.2	Delete List Header.....	7-7
7.3.3	Lock List.....	7-9
7.3.4	Update List.....	7-10
7.3.5	Validate List.....	7-12
7.3.6	Copy List.....	7-14

## 8 List Entries

8.1	Type Declaration.....	8-2
8.2	Standard Parameters for List Entries APIs.....	8-4
8.3	List Entries APIs.....	8-5
8.3.1	Create List Entries.....	8-5
8.3.2	Delete List Entries.....	8-7
8.3.3	Lock List Entries.....	8-9

8.3.4	Update List Entries.....	8-10
8.3.5	Validate List Entries.....	8-12
8.3.6	Copy List Entries.....	8-14

## 9 List Generation

9.1	Standard Parameters for List Generation APIs.....	9-1
9.2	List Generation APIs.....	9-2
9.2.1	Generate List.....	9-2
9.2.2	Create List Based on Query.....	9-4

---

---

# Send Us Your Comments

**Oracle Marketing Online API Reference Guide, Release 11*i***

**Part No. A97384-02**

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 the postal service.

Oracle Corporation  
CRM Content Development 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 Online and provides information to help you work effectively with these APIs.

This guide includes APIs for the following components of Oracle Marketing Online:

- Campaign
- Campaign Schedule
- Event
- Event Schedule
- List
- List Entries
- List Generation

## 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 Online 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 Online.

## 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/>.

### Accessibility of Code Examples in Documentation

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

## 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

#### **OracleMetaLink**

*OracleMetaLink* 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 *OracleMetaLink*, 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 *OracleMetaLink* 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 Online Implementation Guide**

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

### **Oracle Marketing Online Concepts and Procedures**

This manual provides basic conceptual and reference information needed to understand the Oracle Marketing Online 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 Online 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 Online 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 Oracle8i or Oracle9i 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.

---

---

# Introduction

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

- Campaign
- Campaign Schedule
- Event
- Event Schedule
- 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 Online.

**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 <code>fnf_msg_pub.initialize</code> 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	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('l', '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.

---

---

**Warning:** 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	<p>Indicates that the API performed all the operations requested by its caller.</p> <ul style="list-style-type: none"> <li>■ A success return status may or may not be accompanied by messages in the API message list.</li> <li>■ Currently, the Oracle Marketing Online APIs do not provide a message for a return status of success. <i>VERIFY</i></li> </ul>
E	<p>Indicates that the API failed to perform one or more of the operations requested by its caller.</p> <p>An error return status is accompanied by one or more messages describing the error.</p>
U	<p>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.</p> <p>For example, certain programming errors such as attempting to divide by zero cause this type of error.</p> <p>These types of errors usually cannot be corrected by the user and requires a system administrator or application developer to correct.</p>

## 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.

---

# 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 Online 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**

<b>API Name</b>	<b>User Hook Procedure Name</b>
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**

<b>API Name</b>	<b>User Hook Procedure Name</b>
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 Online API.

**Table 2-2 User Hook Procedures**

<b>API Name</b>	<b>User Hook Procedure Name</b>
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Create_EventHeader_Pre
AMS_EVENTHEADER_PUB	AMS_EventHeader_CUHK.Create_EventHeader_Post

**Table 2-2 User Hook Procedures**

<b>API Name</b>	<b>User Hook Procedure Name</b>
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**

<b>API Name</b>	<b>User Hook Procedure Name</b>
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
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

## 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.

---



---

## 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**

<b>API Name</b>	<b>User Hook Procedure Name</b>
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),
dscrip_t_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	Yes	Default = FND_API.G_FALSE If set to true, then the API makes a call to <code>fnf_msg_pub.initialize</code> 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.

### 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
<code>p_api_version</code>	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.
<code>p_init_msg_list</code>	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: <code>FND_API.g_false</code> .
<code>p_commit</code>	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: <code>FND_API.g_false</code> .
<code>p_camp_id</code>	NUMBER	Yes	Campaign ID.
<code>p_object_version_number</code>	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
<code>x_return_status</code>	VARCHAR2(1)	Indicates the return status of the API. The values returned are one of the following: <code>FND_API.G_RET_STS_SUCCESS</code> which indicates the API call was successful. <code>FND_API.G_RET_STS_ERROR</code> which indicates there was a validation error or a missing data error. <code>FND_API.G_RET_STS_UNEXP_ERROR</code> which indicates the calling program encountered an unexpected or unhandled error.
<code>x_msg_count</code>	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.

---



---

## 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
<a href="#">Copy Campaign Schedule</a>	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**

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 <code>fn_msg_pub.initialize</code> 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.

**Table 4–8 OUT Parameters**

Parameter	Data Type	Description
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.

### 4.4.3 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.

**Table 4–9 IN Parameters**

Parameter	Data Type	Required	Description
p_commit	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: FND_API.g_false.
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.

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

The procedures which make up the Event APIs are:

**Table 5–1 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.

## 5.1 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_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_level             VARCHAR2(30),
    application_id          NUMBER,
    event_type_code         VARCHAR2(30),
    active_flag             VARCHAR2(1),
    private_flag            VARCHAR2(1),
    user_status_id         NUMBER,
    system_status_code      VARCHAR2(30),
    last_status_date        DATE,
    stream_type_code        VARCHAR2(30),
    source_code             VARCHAR2(30),
    event_standalone_flag   VARCHAR2(1),
    day_of_event            VARCHAR2(30),
    agenda_start_time       DATE,
    agenda_end_time         DATE,
    reg_required_flag       VARCHAR2(1),
    reg_charge_flag         VARCHAR2(1),
    reg_invited_only_flag   VARCHAR2(1),
    partner_flag           VARCHAR2(1),
    overflow_flag           VARCHAR2(1),
    parent_event_header_id  NUMBER,
    duration                NUMBER,
    duration_uom_code       VARCHAR2(3),
    active_from_date        DATE,
    active_to_date          DATE,
    reg_maximum_capacity    NUMBER,
    reg_minimum_capacity    NUMBER,
    main_language_code      VARCHAR2(4),
    cert_credit_type_code   VARCHAR2(30),
    certification_credits   NUMBER,
    inventory_item_id      NUMBER,
    organization_id         NUMBER,
    org_id                 NUMBER,
    forecasted_revenue      NUMBER,
    actual_revenue          NUMBER,

```

---

forecasted_cost	NUMBER,
actual_cost	NUMBER,
coordinator_id	NUMBER,
fund_source_type_code	VARCHAR2(30),
fund_source_id	NUMBER,
fund_amount_tc	NUMBER,
fund_amount_fc	NUMBER,
currency_code_tc	VARCHAR2(30),
currency_code_fc	VARCHAR2(30),
owner_user_id	NUMBER,
url	VARCHAR2(4000),
email	VARCHAR2(120),
phone	VARCHAR2(25),
priority_type_code	VARCHAR2(30),
cancellation_reason_code	VARCHAR2(30),
inbound_script_name	VARCHAR2(240),
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),
event_header_name	VARCHAR2(240),
event_mktg_message	VARCHAR2(4000),
description	VARCHAR2(4000),
custom_setup_id	NUMBER,
country_code	VARCHAR2(30),
business_unit_id	NUMBER,
event_calendar	VARCHAR2(15),
start_period_name	VARCHAR2(15),
end_period_name	VARCHAR2(15),
global_flag	VARCHAR2(1),
task_id	NUMBER,
program_id	NUMBER,
create_attendant_lead_flag	VARCHAR2(1),

```

create_registrant_lead_flag VARCHAR2(1),
event_purpose_code           VARCHAR2(30)
);

```

## 5.2 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 5–2 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 <code>fnd_msg_pub.initialize</code> 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 5–3 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.

## 5.3 Event APIs

### 5.3.1 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_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_evh_rec          IN      AMS_EventHeader_PVT.evh_rec_type,
    x_evh_id           OUT     NUMBER
);
```

## 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 5–4 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 5–5 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

### 5.3.2 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_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_evh_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 event to be inactive, instead of removing it from the database.

**Table 5–6 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 5–7 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 5–7 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.

### 5.3.3 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_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_evh_id           IN      NUMBER,
    p_object_version   IN      NUMBER
);
```

#### 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 5–8 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 5–8 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 5–9 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.

### 5.3.4 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_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_evh_rec          IN      AMS_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 5–10 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 5–10 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 5–11 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.

### 5.3.5 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_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_evh_rec          IN      AMS_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 5–12 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 5–13 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 5–13 OUT Parameters**

<b>Parameter</b>	<b>Data Type</b>	<b>Description</b>
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.

---



---

## 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 6–1 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 than 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.

## 6.1 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_ID                NUMBER,
    LAST_UPDATE_DATE              DATE,
    LAST_UPDATED_BY               NUMBER,
    CREATION_DATE                 DATE,
    CREATED_BY                    NUMBER,
    LAST_UPDATE_LOGIN             NUMBER,
    OBJECT_VERSION_NUMBER         NUMBER,
    APPLICATION_ID                NUMBER,
    EVENT_HEADER_ID               NUMBER,
    PRIVATE_FLAG                  VARCHAR2(1),
    ACTIVE_FLAG                   VARCHAR2(1),
    SOURCE_CODE                   VARCHAR2(30),
    EVENT_LEVEL                   VARCHAR2(30),
    USER_STATUS_ID                NUMBER,
    LAST_STATUS_DATE              DATE,
    SYSTEM_STATUS_CODE            VARCHAR2(30),
    EVENT_TYPE_CODE               VARCHAR2(30),
    EVENT_DELIVERY_METHOD_ID      NUMBER,
    EVENT_DELIVERY_METHOD_CODE    VARCHAR2(30),
    EVENT_REQUIRED_FLAG           VARCHAR2(1),
    EVENT_LANGUAGE_CODE           VARCHAR2(30),
    EVENT_LOCATION_ID             NUMBER,
    CITY                          VARCHAR2(60),
    STATE                         VARCHAR2(60),
    PROVINCE                      VARCHAR2(60),
    COUNTRY                      VARCHAR2(60),
    OVERFLOW_FLAG                 VARCHAR2(1),
    PARTNER_FLAG                  VARCHAR2(1),
    EVENT_STANDALONE_FLAG         VARCHAR2(1),
    REG_FROZEN_FLAG               VARCHAR2(1),
    REG_REQUIRED_FLAG             VARCHAR2(1),
    REG_CHARGE_FLAG               VARCHAR2(1),
    REG_INVITED_ONLY_FLAG         VARCHAR2(1),
    REG_WAITLIST_ALLOWED_FLAG     VARCHAR2(1),
    REG_OVERBOOK_ALLOWED_FLAG    VARCHAR2(1),
    PARENT_EVENT_OFFER_ID         NUMBER,
    EVENT_DURATION                 NUMBER,

```

EVENT_DURATION_UOM_CODE	VARCHAR2 (3) ,
EVENT_START_DATE	DATE,
EVENT_START_DATE_TIME	DATE,
EVENT_END_DATE	DATE,
EVENT_END_DATE_TIME	DATE,
REG_START_DATE	DATE,
REG_START_TIME	DATE,
REG_END_DATE	DATE,
REG_END_TIME	DATE,
REG_MAXIMUM_CAPACITY	NUMBER,
REG_OVERBOOK_PCT	NUMBER,
REG_EFFECTIVE_CAPACITY	NUMBER,
REG_WAITLIST_PCT	NUMBER,
REG_MINIMUM_CAPACITY	NUMBER,
REG_MINIMUM_REQ_BY_DATE	DATE,
INVENTORY_ITEM_ID	NUMBER,
INVENTORY_ITEM	VARCHAR2 (1000) ,
ORGANIZATION_ID	NUMBER,
PRICELIST_HEADER_ID	NUMBER,
PRICELIST_LINE_ID	NUMBER,
ORG_ID	NUMBER,
WAITLIST_ACTION_TYPE_CODE	VARCHAR2 (30) ,
STREAM_TYPE_CODE	VARCHAR2 (30) ,
OWNER_USER_ID	NUMBER,
EVENT_FULL_FLAG	VARCHAR2 (1) ,
FORECASTED_REVENUE	NUMBER,
ACTUAL_REVENUE	NUMBER,
FORECASTED_COST	NUMBER,
ACTUAL_COST	NUMBER,
FUND_SOURCE_TYPE_CODE	VARCHAR2 (30) ,
FUND_SOURCE_ID	NUMBER,
CERT_CREDIT_TYPE_CODE	VARCHAR2 (30) ,
CERTIFICATION_CREDITS	NUMBER,
COORDINATOR_ID	NUMBER,
PRIORITY_TYPE_CODE	VARCHAR2 (30) ,
CANCELLATION_REASON_CODE	VARCHAR2 (30) ,
AUTO_REGISTER_FLAG	VARCHAR2 (1) ,
EMAIL	VARCHAR2 (120) ,
PHONE	VARCHAR2 (25) ,
FUND_AMOUNT_TC	NUMBER,
FUND_AMOUNT_FC	NUMBER,
CURRENCY_CODE_TC	VARCHAR2 (15) ,
CURRENCY_CODE_FC	VARCHAR2 (15) ,
URL	VARCHAR2 (4000) ,
TIMEZONE_ID	NUMBER,

EVENT_VENUE_ID	NUMBER,
PRICELIST_HEADER_CURRENCY_	CODE VARCHAR2 (30) ,
PRICELIST_LIST_PRICE	NUMBER,
INBOUND_SCRIPT_NAME	VARCHAR2 (240) ,
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) ,
EVENT_OFFER_NAME	VARCHAR2 (240) ,
EVENT_MKTG_MESSAGE	VARCHAR2 (4000) ,
DESCRIPTION	VARCHAR2 (4000) ,
CUSTOM_SETUP_ID	NUMBER,
COUNTRY_CODE	VARCHAR2 (30) ,
BUSINESS_UNIT_ID	NUMBER,
EVENT_CALENDAR	VARCHAR2 (15) ,
START_PERIOD_NAME	VARCHAR2 (15) ,
END_PERIOD_NAME	VARCHAR2 (15) ,
GLOBAL_FLAG	VARCHAR2 (1) ,
TASK_ID	NUMBER,
--PROGRAM_ID	NUMBER,
PARENT_TYPE	VARCHAR2 (30) ,
PARENT_ID	NUMBER,
CREATE_ATTENDANT_LEAD_FLAG	VARCHAR2 (1) ,
CREATE_REGISTRANT_LEAD_FLAG	VARCHAR2 (1) ,
EVENT_OBJECT_TYPE	VARCHAR2 (30) ,
REG_TIMEZONE_ID	NUMBER,
EVENT_PASSWORD	VARCHAR2 (30) ,
RECORD_EVENT_FLAG	VARCHAR2 (1) ,
ALLOW_REGISTER_IN_MIDDLE_FLAG	VARCHAR2 (1) ,
PUBLISH_ATTENDEES_FLAG	VARCHAR2 (1) ,
DIRECT_JOIN_FLAG	VARCHAR2 (1) ,
EVENT_NOTIFICATION_METHOD	VARCHAR2 (30) ,
ACTUAL_START_TIME	DATE,

```

ACTUAL_END_TIME          DATE,
SERVER_ID                NUMBER,
OWNER_FND_USER_ID       NUMBER,
MEETING_DIAL_IN_INFO    VARCHAR2(4000),
MEETING_EMAIL_SUBJECT    VARCHAR2(4000),
MEETING_SCHEDULE_TYPE    VARCHAR2(30),
MEETING_STATUS          VARCHAR2(30),
MEETING_MISC_INFO       VARCHAR2(4000),
PUBLISH_FLAG            VARCHAR2(1),
MEETING_ENCRYPTION_KEY_CODE VARCHAR2(150),
NUMBER_OF_ATTENDEES     NUMBER,
EVENT_PURPOSE_CODE      VARCHAR2(30)
);

```

## 6.2 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 6–2 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 <code>fnd_msg_pub.initialize</code> 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 6–2 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 6–3 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.

## 6.3 Event Schedule APIs

### 6.3.1 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_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_evo_rec            IN       AMS_EventOffer_PVT.evo_rec_type,
x_evo_id             OUT      NUMBER
);

```

## 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 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_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 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 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

### 6.3.2 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_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_evo_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 Event Schedule to be inactive, instead of removing it from the database.

**Table 6–6 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 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.

### 6.3.3 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_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_evo_id           IN      NUMBER,
    p_object_version   IN      NUMBER
);
```

#### Current Version

1.0

## Parameter Descriptions

### Notes

1. Raise exception if the object\_version\_number doesn't match.

**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_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 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.

## 6.3.4 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_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_evo_rec          IN      AMS_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 6–10 IN Parameters**

Parameter	Data Type	Required	Description
<code>p_api_version</code>	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 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 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.

### 6.3.5 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_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_evo_rec          IN      AMS_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 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.
p_evo_rec	AMS_eventoffer_PVT.evo_rec_type	Yes	Record for the event schedule.

**Table 6–13 OUT Parameters**

<b>Parameter</b>	<b>Data Type</b>	<b>Description</b>
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.



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

The procedures which make up the List APIs are:

**Table 7-1 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 7-1 List APIs**

Procedure	Description
<a href="#">Copy List Header</a>	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 <code>ams_list_select_actions</code> , <code>ams_list_queries_all</code> , and <code>ams_list_entries</code> into a new set and associate them with a new list.

## 7.1 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_id           number,
  last_update_date        date,
  last_updated_by         number,
  creation_date           date,
  created_by              number,
  last_update_login       number,
  object_version_number   number,
  request_id              number,
  program_id              number,
  program_application_id   number,
  program_update_date     date,
  view_application_id     number,
  list_name               varchar2(240),
  list_used_by_id         number,
  arc_list_used_by        varchar2(30),
  list_type               varchar2(30),
  status_code             varchar2(30),
  status_date            date,
  generation_type         varchar2(30),
  repeat_exclude_type     varchar2(30),
  row_selection_type      varchar2(30),
  owner_user_id          number,
  access_level            varchar2(30),
  enable_log_flag         varchar2(1),
  enable_word_replacement_flag varchar2(1),
  enable_parallel_dml_flag varchar2(1),
  dedupe_during_generation_flag varchar2(1),

```

---

generate_control_group_flag	varchar2(1),
last_generation_success_flag	varchar2(1),
forecasted_start_date	date,
forecasted_end_date	date,
actual_end_date	date,
sent_out_date	date,
dedupe_start_date	date,
last_dedupe_date	date,
last_deduped_by_user_id	number,
workflow_item_key	number,
no_of_rows_duplicates	number,
no_of_rows_min_requested	number,
no_of_rows_max_requested	number,
no_of_rows_in_list	number,
no_of_rows_in_ctrl_group	number,
no_of_rows_active	number,
no_of_rows_inactive	number,
no_of_rows_manually_entered	number,
no_of_rows_do_not_call	number,
no_of_rows_do_not_mail	number,
no_of_rows_random	number,
org_id	number,
main_gen_start_time	date,
main_gen_end_time	date,
main_random_nth_row_selection	number,
main_random_pct_row_selection	number,
ctrl_random_nth_row_selection	number,
ctrl_random_pct_row_selection	number,
repeat_source_list_header_id	varchar2(4000),
result_text	varchar2(4000),
keywords	varchar2(4000),
description	varchar2(4000),
list_priority	number,
assign_person_id	number,
list_source	varchar2(240),
list_source_type	varchar2(30),
list_online_flag	varchar2(1),
random_list_id	number,
enabled_flag	varchar2(1),
assigned_to	number,
query_id	number,
owner_person_id	number,
archived_by	number,
archived_date	date,
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),
timezone_id        number,
user_entered_start_time date,
user_status_id     number,
quantum            number,
release_control_alg_id number,
dialing_method     varchar2(10),
calling_calendar_id number,
release_strategy   varchar2(10),
custom_setup_id   number,
country            number,
callback_priority_flag varchar2(1),
call_center_ready_flag varchar2(1),
language           varchar2(4),
purge_flag         varchar2(1),
public_flag        varchar2(1),
list_category      varchar2(120),
quota              number,
quota_reset        number,
recycling_alg_id   number,
source_lang        varchar2(4)
);

```

## 7.2 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 7–2 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 7–3 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.

## 7.3 List APIs

### 7.3.1 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_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_listheader_rec    IN      AMS_LISTHEADER_PVT.list_header_rec_type,
x_listheader_id     OUT     NUMBER
);

```

## 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 7-4 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.

**Table 7-4 IN Parameters**

Parameter	Data Type	Required	Description
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 7-5 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.

## 7.3.2 Delete List Header

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

### Procedure Specification

```
PROCEDURE Delete_ListHeader(
```

```

p_api_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_listheader_id    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 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 7-6 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.
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 7-7 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.

### 7.3.3 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_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_listheader_id    IN      NUMBER,
    p_object_version   IN      NUMBER
);
```

#### Current Version

1.0

## Parameter Descriptions

**Table 7–8 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 7–9 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.

### 7.3.4 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_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_listheader_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. If an attribute is passed in as FND\_API.g\_miss\_char/num/date, that column won't be updated.

**Table 7-10 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.

**Table 7-10 IN Parameters**

Parameter	Data Type	Required	Description
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 7-11 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.

### 7.3.5 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_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_listheader_rec   IN      AMS_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 7-12 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 7-13 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.

### 7.3.6 Copy List

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

#### Procedure Specification

```

PROCEDURE Copy_List (
    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_listheader_id IN          NUMBER,
    p_listheader_rec       IN          AMS_LISTHEADER_PVT.list_header_rec_type,
    p_copy_select_actions  IN          VARCHAR2 := 'Y',
    p_copy_list_queries    IN          VARCHAR2 := 'Y',
    p_copy_list_entries    IN          VARCHAR2 := 'Y',
    x_listheader_id       OUT         NUMBER
);

```

**Current Version**

1.0

**Parameter Descriptions****Notes**

1. Oracle recommends that the p-listheader\_rec should be the complete list record.

**Table 7-14 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 7–15 OUT Parameters**

<b>Parameter</b>	<b>Data Type</b>	<b>Description</b>
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.

---



---

## 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 8–1 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.

## 8.1 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_id                NUMBER := FND_API.G_MISS_NUM,
  list_header_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,
  list_select_action_id       NUMBER := FND_API.G_MISS_NUM,
  arc_list_select_action_from VARCHAR2(30) := FND_API.G_MISS_CHAR,
  list_select_action_from_name VARCHAR2(254) := FND_API.G_MISS_CHAR,
  source_code                 VARCHAR2(30) := FND_API.G_MISS_CHAR,
  arc_list_used_by_source     VARCHAR2(30) := FND_API.G_MISS_CHAR,
  source_code_for_id         NUMBER := FND_API.G_MISS_NUM,
  pin_code                   VARCHAR2(30) := FND_API.G_MISS_CHAR,
  list_entry_source_system_id NUMBER := FND_API.G_MISS_NUM,
  list_entry_source_system_type VARCHAR2(30) := FND_API.G_MISS_CHAR,
  view_application_id        NUMBER := FND_API.G_MISS_NUM,
  manually_entered_flag      VARCHAR2(1) := FND_API.G_MISS_CHAR,
  marked_as_duplicate_flag   VARCHAR2(1) := FND_API.G_MISS_CHAR,
  marked_as_random_flag     VARCHAR2(1) := FND_API.G_MISS_CHAR,
  part_of_control_group_flag VARCHAR2(1) := FND_API.G_MISS_CHAR,
  exclude_in_triggered_list_flag VARCHAR2(1) := FND_API.G_MISS_CHAR,
  enabled_flag               VARCHAR2(1) := FND_API.G_MISS_CHAR,
  cell_code                  VARCHAR2(30) := FND_API.G_MISS_CHAR,
  dedupe_key                 VARCHAR2(500) := FND_API.G_MISS_CHAR,
  randomly_generated_number  NUMBER := FND_API.G_MISS_NUM,
  campaign_id                NUMBER := FND_API.G_MISS_NUM,
  media_id                   NUMBER := FND_API.G_MISS_NUM,
  channel_id                 NUMBER := FND_API.G_MISS_NUM,
  channel_schedule_id        NUMBER := FND_API.G_MISS_NUM,
  event_offer_id             NUMBER := FND_API.G_MISS_NUM,
  customer_id                NUMBER := FND_API.G_MISS_NUM,
  market_segment_id         NUMBER := FND_API.G_MISS_NUM,
  vendor_id                  NUMBER := FND_API.G_MISS_NUM,
  transfer_flag              VARCHAR2(1) := FND_API.G_MISS_CHAR,
  transfer_status            VARCHAR2(1) := FND_API.G_MISS_CHAR,

```

list_source	VARCHAR2(240) := FND_API.G_MISS_CHAR,
duplicate_master_entry_id	NUMBER := FND_API.G_MISS_NUM,
marked_flag	VARCHAR2(1) := FND_API.G_MISS_CHAR,
lead_id	NUMBER := FND_API.G_MISS_NUM,
letter_id	NUMBER := FND_API.G_MISS_NUM,
picking_header_id	NUMBER := FND_API.G_MISS_NUM,
batch_id	NUMBER := FND_API.G_MISS_NUM,
suffix	VARCHAR2(30) := FND_API.G_MISS_CHAR,
first_name	VARCHAR2(150) := FND_API.G_MISS_CHAR,
last_name	VARCHAR2(150) := FND_API.G_MISS_CHAR,
customer_name	VARCHAR2(500) := FND_API.G_MISS_CHAR,
title	VARCHAR2(150) := FND_API.G_MISS_CHAR,
address_line1	VARCHAR2(500) := FND_API.G_MISS_CHAR,
address_line2	VARCHAR2(500) := FND_API.G_MISS_CHAR,
city	VARCHAR2(100) := FND_API.G_MISS_CHAR,
state	VARCHAR2(100) := FND_API.G_MISS_CHAR,
zipcode	VARCHAR2(100) := FND_API.G_MISS_CHAR,
country	VARCHAR2(100) := FND_API.G_MISS_CHAR,
fax	VARCHAR2(150) := FND_API.G_MISS_CHAR,
phone	VARCHAR2(150) := FND_API.G_MISS_CHAR,
email_address	VARCHAR2(500) := FND_API.G_MISS_CHAR,
COL1	VARCHAR2(500) := FND_API.G_MISS_CHAR,
COL240	VARCHAR2(500) := FND_API.G_MISS_CHAR,
COL241	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL242	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL243	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL244	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL245	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL246	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL247	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL248	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL249	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL250	VARCHAR2(4000) := FND_API.G_MISS_CHAR,
COL251	VARCHAR2(500) := FND_API.G_MISS_CHAR,
COL300	VARCHAR2(500) := FND_API.G_MISS_CHAR,
CURR_CP_COUNTRY_CODE	VARCHAR2(30) := FND_API.G_MISS_CHAR,
CURR_CP_PHONE_NUMBER	VARCHAR2(10) := FND_API.G_MISS_CHAR,
CURR_CP_RAW_PHONE_NUMBER	VARCHAR2(60) := FND_API.G_MISS_CHAR,
CURR_CP_AREA_CODE	NUMBER := FND_API.G_MISS_NUM,
CURR_CP_ID	NUMBER := FND_API.G_MISS_NUM,
CURR_CP_INDEX	NUMBER := FND_API.G_MISS_NUM,
CURR_CP_TIME_ZONE	NUMBER := FND_API.G_MISS_NUM,
CURR_CP_TIME_ZONE_AUX	NUMBER := FND_API.G_MISS_NUM,
party_id	NUMBER := FND_API.G_MISS_NUM,
parent_party_id	NUMBER := FND_API.G_MISS_NUM,

```

imp_source_line_id          NUMBER := FND_API.G_MISS_NUM,
usage_restriction          VARCHAR2(1) := FND_API.G_MISS_CHAR,
next_call_time             DATE := FND_API.G_MISS_DATE,
callback_flag              VARCHAR2(1) := FND_API.G_MISS_CHAR,
do_not_use_flag            VARCHAR2(1) := FND_API.G_MISS_CHAR,
do_not_use_reason          VARCHAR2(30) := FND_API.G_MISS_CHAR,
record_out_flag            VARCHAR2(1) := FND_API.G_MISS_CHAR,
record_release_time        DATE := FND_API.G_MISS_DATE
);

```

## 8.2 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 8–2 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 8–3 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 8–3 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.

## 8.3 List Entries APIs

### 8.3.1 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_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_list_entries_rec   IN    list_entries_rec_type := g_miss_list_entries_rec,
  x_list_entry_id     OUT   NUMBER
);
```

#### 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 8–4 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 8–5 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 8–5 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.

## 8.3.2 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_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_list_entry_id         IN    NUMBER,
    p_object_version_number 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 delete the record from the list entries table.

**Table 8–6 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 8–7 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 8–7 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.

### 8.3.3 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_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_list_entry_id     IN    NUMBER,
    p_object_version    IN    NUMBER
);
```

#### Current Version

1.0

#### Parameter Descriptions

**Table 8–8 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 8–8 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 8–9 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.

### 8.3.4 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_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_list_entries_rec     IN      list_entries_rec_type,
x_object_version_number OUT     NUMBER
);

```

## 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 8–10 IN Parameters**

Parameter	Data Type	Required	Description
<code>p_api_version</code>	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.
<code>p_init_msg_list</code>	VARCHAR2	No	Flag to indicate if the message stack should be initialized. Default: <code>FND_API.g_false</code> .
<code>p_commit</code>	VARCHAR2	No	Flag to indicate if the changes should be committed on success. Default: <code>FND_API.g_false</code> .
<code>p_list_entry_rec</code>	AMS_List_Entries_PVT.list_entries_rec_type	Yes	Record ID for the list entry.

**Table 8–11 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.

### 8.3.5 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_number IN    NUMBER,
    p_init_msg_list      IN    VARCHAR2 := FND_API.G_FALSE,
    p_validation_level   IN    NUMBER := FND_API.G_VALID_LEVEL_FULL,
    p_list_entries_rec   IN    list_entries_rec_type,
    x_return_status      OUT   VARCHAR2,
    x_msg_count          OUT   NUMBER,
    x_msg_data           OUT   VARCHAR2
);
```

#### Current Version

1.0

## Parameter Descriptions

**Table 8–12 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 8–13 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.

## 8.3.6 Copy List Entries

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

### Procedure Specification

```
PROCEDURE Copy_List_Entries(
  p_api_version_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_list_header_id     IN    NUMBER,
  p_new_list_header_id IN    NUMBER
);
```

### Current Version

1.0

### Parameter Descriptions

**Table 8–14 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 copied.

**Table 8–15 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.



---



---

## List Generation

The APIs for List Generation provides two procedures for generating lists.

The procedures which make up the List Generation APIs are:

**Table 9–1 List Generation APIs**

Procedure	Description
<a href="#">Generate List</a>	Generates a list based on the list header ID passed to the procedure.
<a href="#">Create List based on Query</a>	Creates and generates a list based on the list name and query string provided to the procedure.

### 9.1 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 9–2 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 9–3 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.

## 9.2 List Generation APIs

### 9.2.1 Generate List

This procedure generates list entries give the list header ID.

#### Procedure Specification

```
PROCEDURE Generate_List (
    p_api_version      IN      NUMBER,
    p_init_msg_list    IN      VARCHAR2 := FND_API.G_TRUE,
    p_commit           IN      VARCHAR2 := FND_API.G_FALSE,
    p_validation_level IN      NUMBER := FND_API.G_VALID_LEVEL_FULL,
    p_list_header_id   IN      NUMBER,
    x_return_status    OUT     VARCHAR2,
    x_msg_count        OUT     NUMBER,
    x_msg_data         OUT     VARCHAR2
);
```

#### 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 9–4 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 9–5 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 9–5 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.

## 9.2.2 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_version          IN    NUMBER,
    p_init_msg_list       IN    VARCHAR2 := FND_API.G_TRUE,
    p_commit              IN    VARCHAR2 := FND_API.G_FALSE,
    p_validation_level    IN    NUMBER := FND_API.G_VALID_LEVEL_FULL,
    p_list_name           in    varchar2,
    p_list_type           in    varchar2,
    p_owner_user_id      in    number,
    p_list_header_id     in    number,
    p_sql_string_tbl     in    AMS_List_Query_PVT.sql_string_tbl,
    p_primary_key        in    varchar2,
    p_source_object_name in    varchar2,
    p_master_type        in    varchar2,
    x_return_status      OUT   VARCHAR2,
    x_msg_count          OUT   NUMBER,
    x_msg_data           OUT   VARCHAR2
);

```

### Current Version

1.0

## Parameter Descriptions

**Table 9–6 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 9–7 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 9–7 OUT Parameters**

<b>Parameter</b>	<b>Data Type</b>	<b>Description</b>
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.